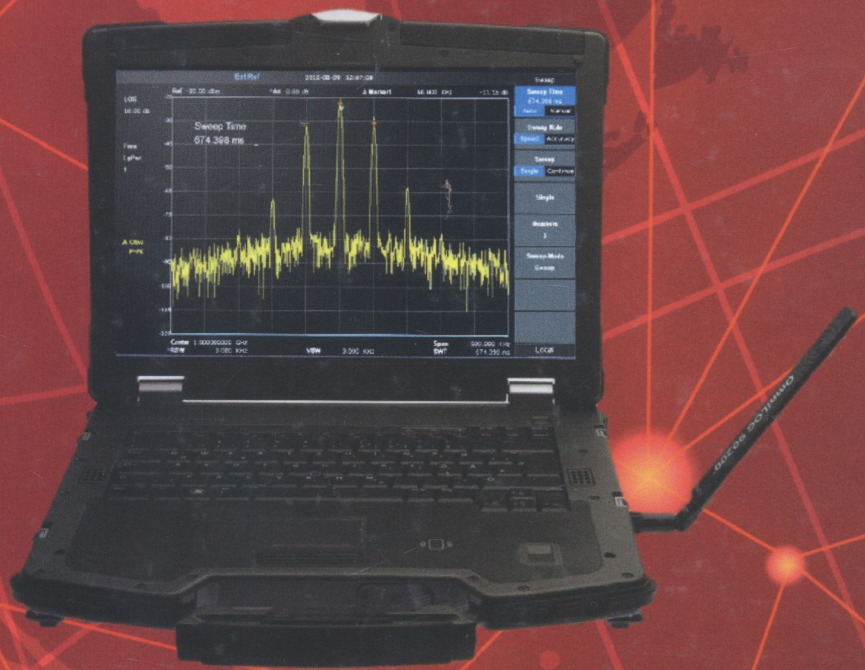
Ю. Я. Бобало, І. В. Горбатий, М. Д. Кіселичник,
А. П. Бондарєв, С. С. Войтусік, А. Я. Горпенюк,
О. А. Нємкова, І. М. Журавель, Б. М. Березюк,
Є. І. Яковенко, В. І. Отенко, І. Я. Тишик

ІНФОРМАЦІЙНА БЕЗПЕКА



Видавництво Львівської політехніки

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

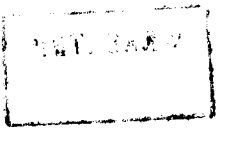
**Ю. Я. Бобало, І. В. Горбатий, М. Д. Кіселичник,
А. П. Бондарєв, С. С. Войтусік, А. Я. Горпенюк,
О. А. Нємкова, І. М. Журавель, Б. М. Березюк,
Є. І. Яковенко, В. І. Отенко, І. Я. Тишик**

ІНФОРМАЦІЙНА БЕЗПЕКА

Навчальний посібник

*За загальною редакцією д-ра техн. наук, проф. Ю. Я. Бобала
та д-ра техн. наук, доц. І. В. Горбатого*

*Рекомендувала Науково-методична рада
Національного університету "Львівська політехніка"*



Львів
Видавництво Львівської політехніки
2019

УДК 621.39+004.3
І-74

Рецензенти:

Дудикевич В. Б., доктор технічних наук, професор, завідувач кафедри захисту інформації Національного університету "Львівська політехніка";

Хорошко В. О., доктор технічних наук, професор, професор кафедри безпеки інформаційних технологій Навчально-наукового інституту інформаційно-діагностичних систем Національного авіаційного університету;

Казакова Н. Ф., доктор технічних наук, доцент, завідувач кафедри комп'ютерних та інформаційно-вимірювальних технологій Одеської державної академії технічного регулювання та якості;

Потій О. В., доктор технічних наук, професор, заступник головного конструктора Акціонерного товариства "Інститут інформаційних технологій"

Рекомендувала Науково-методична рада Національного університету "Львівська політехніка" як навчальний посібник для студентів спеціальностей 172 "Телекомунікації та радіотехніка", 125 "Кібербезпека" та 163 "Біомедична інженерія" (протокол № 35 від 3.05.2018 р.)

Бобало Ю. Я.

І-74 Інформаційна безпека : навч. посібник / Ю. Я. Бобало, І. В. Горбатий, М. Д. Кіселичник, А. П. Бондарєв, С. С. Войтусік, А. Я. Горпенюк, О. А. Нємкова, І. М. Журавель, Б. М. Березюк, Є. І. Яковенко, В. І. Отенко, І. Я. Тишик; за заг. ред. д-ра техн. наук, проф. Ю. Я. Бобала та д-ра техн. наук, доц. І. В. Горбатого. – Львів : Видавництво Львівської політехніки, 2019. – 580 с.

ISBN 978-966-941-339-0

Розглянуто основні поняття та визначення в галузі інформаційної безпеки. Описано математичні основи криптології. Розглянуто відомі та сучасні методи криптографії, криптоаналізу, стеганографії. Розглянуто питання ідентифікації, автентифікації та санкціонованого доступу.

Значну увагу в навчальному посібнику приділено практичному захисту інформації. Розглянуто питання безпеки інформаційних систем. Висвітлено особливості захисту програмного забезпечення в інформаційних системах. Розглянуто питання інформаційної безпеки підприємств та організацій. Особливу увагу приділено проектуванню, побудові та функціонуванню систем інформаційної безпеки.

Для студентів закладів вищої освіти спеціальностей 172 "Телекомунікації та радіотехніка", 125 "Кібербезпека", 163 "Біомедична інженерія" та споріднених спеціальностей, а також для тих, хто цікавиться інформаційною безпекою та захистом інформації.

УДК 621.39+004.3

ISBN 978-966-941-339-0

ЗМІСТ

ПЕРЕДМОВА

Навчальний посібник "Інформаційна безпека" спрямований на ознайомлення студентів із основними поняттями та визначеннями в галузі інформаційної безпеки. У ньому описано математичні основи криптології. Розглянуто відомі та сучасні методи криптографії, стеганографії, криптоаналізу. Висвітлено питання ідентифікації, автентифікації та санкціонованого доступу. Значну увагу в навчальному посібнику звернено на практичний захист інформації, розглянуто питання безпеки інформаційних систем, висвітлено особливості захисту програмного забезпечення в інформаційних системах. Розглянуто питання інформаційної безпеки підприємств та організацій. Особливу увагу приділено проектуванню, побудові та функціонуванню систем інформаційної безпеки.

Цей навчальний посібник відповідає навчальному плану підготовки бакалаврів і магістрів за спеціальностями 172 "Телекомунікації та радіотехніка", 125 "Кібербезпека" та 163 "Біомедична інженерія". Основою посібника є науково-методичні матеріали, напрацьовані його авторами під час викладання відповідних дисциплін у галузі інформаційної безпеки, що передбачені навчальними планами для підготовки студентів за згаданими вище спеціальностями.

Навчальний посібник "Інформаційна безпека" буде корисний усім охочим отримати або поглибити свої знання в галузі інформаційної безпеки та захисту інформації. Посібник складається з восьми розділів, що охоплюють:

– основні поняття та визначення в галузі інформаційної безпеки;
– математичні основи криптології;
– криптологію;
– стеганографію;
– ідентифікацію, автентифікацію, санкціонований доступ;
– безпеку інформаційних систем;
– захист програмного забезпечення в інформаційних системах;
– інформаційну безпеку підприємств та організацій, системи інформаційної безпеки.

Навчальний посібник підготували співробітники кафедри теоретичної радіотехніки та радіовимірювання Національного університету "Львівська політехніка" професори Ю. Я. Бобало, І. В. Горбатий, М. Д. Кіселичник, А. П. Бондарєв, співробітники кафедри безпеки інформаційних технологій Національного університету "Львівська політехніка" доценти С. С. Войтусік, І. М. Журавель, асистент О. А. Нємкова, співробітники кафедри захисту інформації Національного університету "Львівська політехніка" доценти А. Я. Горпенюк, Б. М. Березюк, В. І. Отенко, І. Я. Тишик та співробітник кафедри електронних засобів

інформаційно-комунікаційних технологій Національного університету "Львівська політехніка" доцент Є. І. Яковенко.

Авторами та співавторами окремих розділів підручника є:

загальне редагування та передмова – Ю. Я. Бобало;

загальне редагування – І. В. Горбатий;

розділ 1 – І. В. Горбатий, М. Д. Кіселичник;

розділ 2 – С. С. Войтусік;

розділ 3 – А. Я. Горпенюк, О. А. Нємкова, С. С. Войтусік;

розділ 4 – І. М. Журавель, О. А. Нємкова;

розділ 5 – С. С. Войтусік;

розділ 6 – І. В. Горбатий, А. П. Бондарєв, Б. М. Березюк, Є. І. Яковенко;

розділ 7 – В. І. Отенко, Є. І. Яковенко;

розділ 8 – І. В. Горбатий, І. Я. Тишик.

За слушні зауваження та поради, що сприяли поліпшенню змісту цього посібника, автори висловлюють подяку рецензентам: завідувачу кафедри захисту інформації Національного університету "Львівська політехніка", доктору технічних наук, професору Валерію Богдановичу Дудикевичу; професору кафедри безпеки інформаційних технологій Навчально-наукового інституту інформаційно-діагностичних систем Національного авіаційного університету, доктору технічних наук, професору Володимиру Олексійовичу Хорошку; завідувачу кафедри комп'ютерних та інформаційно-вимірювальних технологій Одеської державної академії технічного регулювання та якості, доктору технічних наук, доценту Надії Феліксовні Казаковій; заступнику головного конструктора Акціонерного товариства "Інститут інформаційних технологій", доктору технічних наук, професору Олександру Володимировичу Потію.

За сприяння в організації роботи над цим посібником автори вдячні завідувачеві кафедри безпеки інформаційних технологій Національного університету "Львівська політехніка", доктору технічних наук, професору Володимиру Миколайовичу Максимовичу.

Автори вдячні Андрію Едуардовичу Лагуну, кандидату технічних наук, доценту, доценту кафедри управління інформаційною безпекою Львівського державного університету безпеки життєдіяльності за надані матеріали з математичних основ криптології.

Відповідальний редактор –
доктор технічних наук, професор Ю. Я. Бобало

Розділ 1

ОСНОВНІ ПОНЯТТЯ ТА ВИЗНАЧЕННЯ В ГАЛУЗІ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Загалом під ***інформацією*** розуміють будь-які відомості про навколишній світ. Інформація як продукт діяльності є власністю держави, підприємств, установ, організацій, громадян і як об'єкт власності потребує забезпечення безпеки.

Безпека інформації – це такий її стан, за якого з необхідною ймовірністю забезпечують захист інформації від внутрішніх та зовнішніх загроз.

Основи забезпечення безпеки розробляють органи державної влади, враховуючи умови забезпечення національної безпеки України загалом та інформаційної безпеки зокрема.

1.1. Державна політика України в галузі інформаційної безпеки

1.1.1. Інформаційна безпека та її місце в системі національної безпеки України

Необхідною умовою нормального існування й розвитку кожного суспільства є захищеність від зовнішніх і внутрішніх загроз, стійкість до спроб зовнішнього тиску, здатність протистояти таким спробам і нейтралізовувати нові загрози, а також забезпечувати такі внутрішні й зовнішні умови існування країни, які гарантують можливість стабільного й усебічного прогресу суспільства та його громадян. Для характеристики цього стану використовують поняття національної безпеки.

Національна безпека – захищеність життєво важливих інтересів людини й громадянина, суспільства й держави, за якої забезпечують сталий розвиток суспільства, своєчасне виявлення, запобігання й нейтралізацію реальних та потенційних загроз національним інтересам.

До основних складових національної безпеки належать політична, економічна, військова, екологічна та інформаційна безпека.

Суть ***політичної безпеки*** полягає в здатності нації створити політичну систему, що забезпечує баланс інтересів різних соціальних груп; самостійно вирішувати питання державного устрою; здійснювати незалежну внутрішню і зовнішню політику.

Під ***економічною безпекою*** розуміють стан нації, за якого вона може суверенно, без зовнішнього втручання визначати шляхи й форми свого економічного розвитку.

Військова безпека полягає в можливості забезпечення національної безпеки із застосуванням військової сили. Насамперед військову безпеку характеризують здатністю нації стримувати агресію або протидіяти їй.

Екологічна безпека полягає в наявності безпечного місця існування, що забезпечує нормальну життєдіяльність людини. Баланс складових у системі "населення – навколишнє середовище – природні ресурси" є гарантом життєздатності людського суспільства.

Інформаційна безпека – стан захищеності інформаційних ресурсів від внутрішніх і зовнішніх загроз, здатних завдати збитку національним інтересам (інтересам людини й громадянина, суспільства й держави).

Оскільки ми живемо в епоху інформаційного суспільства, для якого характерні інформатизація країни, розвиток інформаційних технологій, тому інформаційні ресурси формують у всіх сферах людської діяльності, і насамперед у політичній, військовій, економічній, науково-технічній сферах. Отже, інформаційну безпеку слід розглядати як комплексний показник національної безпеки. Цим визначається її важливе місце й одна із провідних ролей у системі національної безпеки країни в сучасних умовах.

У межах забезпечення національної безпеки інформаційної безпеки досягають проведенням єдиної державної політики в галузі забезпечення безпеки, системою заходів економічного, політичного й іншого характеру, адекватних загрозам життєво важливим інтересам особи, суспільства, держави.

Політику України в галузі національної безпеки будують на основі Закону України "Про основи національної безпеки України" від 19 червня 2003 року № 964-IV [1]. Цей закон визначає основні засади державної політики, спрямованої на захист національних інтересів і гарантування в Україні безпеки особи, суспільства й держави від зовнішніх і внутрішніх загроз в усіх сферах життєдіяльності.

Правову основу у сфері національної безпеки України становлять Конституція, закони України, міжнародні договори, згоду на обов'язковість яких надала Верховна Рада України, а також видані на виконання законів інші нормативно-правові акти.

Відповідно до Закону України "Про основи національної безпеки України" розроблені й затверджені "Стратегія національної безпеки України" від 26 травня 2015 року № 287/2015 [2], "Стратегія кібербезпеки України" від 15 березня 2016 року № 96/2016 [3] і "Воєнна доктрина України" від 24 вересня 2015 року № 555/2015 [4], інші доктрини, концепції, стратегії та програми.

Згідно із цим законом ***об'єктами національної безпеки*** є:

– людина й громадянин – їхні конституційні права та свободи;

– суспільство – його духовні, морально-етичні, культурні, історичні, інтелектуальні та матеріальні цінності, інформаційне й навколишнє природне середовище та природні ресурси;

– держава – її конституційний лад, суверенітет, територіальна цілісність і недоторканність.

Суб'єктами забезпечення національної безпеки є:

– Президент України;

– Верховна Рада України;

– Кабінет Міністрів України;

– Рада національної безпеки і оборони України;

– міністерства та інші центральні органи виконавчої влади;

– Національний банк України;

– суди загальної юрисдикції;

– прокуратура України;

– Національне антикорупційне бюро України;

– місцеві державні адміністрації та органи місцевого самоврядування;

– Збройні Сили України, Служба безпеки України, Служба зовнішньої розвідки України, Державна прикордонна служба України та інші військові формування, утворені відповідно до законів України;

– органи й підрозділи цивільного захисту;

– громадяни України, об'єднання громадян.

Основними ***принципами забезпечення національної безпеки*** є:

– пріоритет прав і свобод людини й громадянина;

– верховенство права;

– пріоритет договірних (мирних) засобів у розв'язанні конфліктів;

– своєчасність і адекватність заходів захисту національних інтересів реальним і потенційним загрозам;

– чітке розмежування повноважень та взаємодія органів державної влади в забезпеченні національної безпеки;

– демократичний цивільний контроль над військовою організацією держави та іншими структурами в системі національної безпеки;

– використання в інтересах України міждержавних систем та механізмів міжнародної колективної безпеки.

Загрози національній безпеці України лежать у зовнішньополітичній сфері, у сфері державної безпеки, у воєнній сфері та сфері безпеки державного кордону України, у внутрішньополітичній, економічній, соціальній, гуманітарній, науково-технологічній сферах, сфері цивільного захисту, в екологічній та інформаційній сферах. На сучасному етапі ***основними реальними та потенційними загрозами національній безпеці України*** є:

– посягання на державний суверенітет України та її територіальну цілісність, територіальні претензії з боку інших держав;

– спроби втручання у внутрішні справи України з боку інших держав;

– воєнно-політична нестабільність, регіональні та локальні війни в різних регіонах світу, насамперед поблизу кордонів України;

– розвідувально-підривна діяльність іноземних спеціальних служб;

– загроза посягань із боку окремих груп та осіб на державний суверенітет, територіальну цілісність, економічний, науково-технічний і оборонний потенціал України, права й свободи громадян;

– поширення корупції в органах державної влади, зрощення бізнесу й політики, організованої злочинної діяльності тощо.

З урахуванням геополітичної й внутрішньої обстановки в Україні діяльність усіх державних органів має бути зосереджена на прогнозуванні, своєчасному виявленні, попередженні й нейтралізації зовнішніх і внутрішніх загроз національній безпеці, захисті суверенітету й територіальної цілісності України, безпеки її прикордонного простору, піднесенні економіки країни, забезпеченні особистої безпеки, конституційних прав і свобод людини й громадянина.

У законі наведено основні напрями державної політики з питань національної безпеки України у всіх сферах.

Також у законі розглянуто повноваження та основні функції суб'єктів забезпечення національної безпеки, визначено органи державної влади та управління, що контролюють здійснення заходів щодо забезпечення національної безпеки.

У 2018 році Верховна Рада України прийняла Закон України "Про національну безпеку України" [5]. Цей Закон відповідно до статей 1, 2, 17, 18 і 92 Конституції України визначає основи та принципи національної безпеки й оборони, цілі та основні засади державної політики, що гарантуватимуть суспільству й кожному громадянину захист від загроз. Цей Закон визначає й розмежовує повноваження державних органів у сферах національної безпеки й оборони, створює основу для інтеграції політики та процедур органів державної

влади, інших державних органів, функції яких стосуються національної безпеки й оборони, сил безпеки й сил оборони, визначає систему командування, контролю та координації операцій сил безпеки та сил оборони, запроваджує всеосяжний підхід до планування у сферах національної безпеки й оборони, забезпечуючи у такий спосіб демократичний громадський контроль над органами та формуваннями сектору безпеки й оборони.

1.1.2. Державна політика інформаційної безпеки та її здійснення в законодавстві України

Державну політику інформаційної безпеки здійснюють у межах політики національної безпеки і політики інформатизації всіх сфер діяльності держави й суспільства на основі Законів України "Про інформацію" (1992 р.), "Про Концепцію Національної програми інформатизації" (1998 р.), "Про Національну програму інформатизації" (1998 р.), "Про основи національної безпеки України" (2003 р.), які є системотвірними в національному інформаційному законодавстві [6–16].

Зокрема в Законі України "Про основи національної безпеки України" визначені основні напрями державної політики з питань національної безпеки України ***в інформаційній сфері***:

– забезпечення інформаційного суверенітету України;

– удосконалення державного регулювання розвитку інформаційної сфери створенням нормативно-правових та економічних передумов для розвитку національної інформаційної інфраструктури та ресурсів, упровадження новітніх технологій у цій сфері, наповнення внутрішнього та світового інформаційного простору достовірною інформацією про Україну;

– активне залучення засобів масової інформації до запобігання й протидії корупції, зловживанням службовим становищем, іншим явищам, які загрожують національній безпеці України;

– забезпечення неухильного дотримання конституційних прав на свободу слова, доступ до інформації, недопущення неправомірного втручання органів державної влади, органів місцевого самоврядування, їх посадових осіб у діяльність засобів масової інформації та журналістів, заборони цензури, дискримінації в інформаційній сфері й переслідування журналістів за політичні позиції, за виконання професійних обов'язків, за критику;

– вжиття комплексних заходів щодо захисту національного інформаційного простору та протидії монополізації інформаційної сфери України.

Напрямки політики захисту інформації та інформаційних ресурсів затверджено в законодавстві України, що, зокрема, містить:

– Закон України "Про інформацію, інформатизацію та захист інформації" (2005 р.);

– Закон України "Про захист інформації в інформаційно-телекомунікаційних системах" (1994 р.);

– Закон України "Про державну таємницю" (1994 р.);

– Закон України "Про основні засади забезпечення кібербезпеки України" (2017 р.);

– Закон України "Про авторське право та суміжні права" (1993 р.);

– Закон України "Про друковані засоби масової інформації (пресу) в Україні" (1993 р.);

– Закон України "Про захист персональних даних" (2010 р.);

– Цивільний кодекс України (2003 р.);

– Кримінальний кодекс України (2001 р.).

Закон "Про інформацію, інформатизацію і захист інформації" відображає основні напрями політики інформаційної безпеки, суть якої у своїй основі зводиться до захисту державних інформаційних ресурсів, регулює стосунки, що виникають при формуванні й використанні інформаційних ресурсів, створенні й використанні інформаційних технологій, захисті інформації, прав суб'єктів, що беруть участь в інформаційних процесах, а також визначає основні поняття, що використовують у законодавстві.

Закон України "Про захист інформації в інформаційно-телекомунікаційних системах" регулює відносини в сфері захисту інформації в інформаційних та телекомунікаційних системах.

Закон України "Про державну таємницю" регулює суспільні відносини, пов'язані з віднесенням інформації до державної таємниці, засекречуванням, розсекречуванням її матеріальних носіїв та охороною державної таємниці з метою захисту національної безпеки України.

Закон України "Про основні засади забезпечення кібербезпеки України" визначає правові та організаційні основи забезпечення захисту життєво важливих інтересів людини й громадянина, суспільства та держави, національних інтересів України у кіберпросторі, основні цілі, напрями та принципи державної політики у сфері кібербезпеки, повноваження державних органів, підприємств, установ, організацій, осіб та громадян у цій сфері, основні засади координації їхньої діяльності із забезпечення кібербезпеки.

Закон України "Про авторське право та суміжні права" охороняє особисті немайнові права й майнові права авторів та їх правонаступників, пов'язані зі створенням та використанням творів науки, літератури й мистецтва – авторське право, і права виконавців, виробників фонограм і відеограм та організацій мовлення – суміжні права.

Закон України "Про друковані засоби масової інформації (пресу) в Україні" створює правові основи діяльності друкованих засобів масової інформації (преси) в Україні, встановлює державні гарантії їх свободи відповідно до Конституції України, Закону України "Про інформацію" та інших актів чинного законодавства й визнаних Україною міжнародно-правових документів.

Закон України "Про захист персональних даних" регулює правові відносини, пов'язані із захистом та обробленням персональних даних, а також спрямований на захист основоположних прав і свобод людини та громадянина, зокрема права на невтручання в особисте життя, у зв'язку з обробленням персональних даних.

Цивільний кодекс України регулює особисті немайнові та майнові відносини (цивільні відносини), основані на юридичній рівності, вільному волевиявленні, майновій самостійності їх учасників.

Кримінальний кодекс України має своїм завданням правове забезпечення охорони прав і свобод людини й громадянина, власності, громадського порядку та громадської безпеки, довкілля, конституційного устрою України від злочинних посягань, забезпечення миру й безпеки людства, а також запобігання злочинам.

Загалом розвиток законодавчої бази в галузі інформаційної безпеки йде за чотирма основними напрямами:

– захист відомостей, що становлять державну таємницю;
– захист конфіденційної інформації;
– захист авторського права в сфері інформатизації;
– захист права на доступ до інформації.

1.1.3. Органи забезпечення інформаційної безпеки та захисту інформації

Органи забезпечення інформаційної безпеки та захисту інформації в сукупності із законодавством утворюють державну систему інформаційної безпеки й захисту інформації [6–16]. До складу цієї державної системи входять:

– органи законодавчої, виконавчої й судової влади;
– законодавство, що регулює відносини в галузі інформаційної безпеки, захисту інформації та інформаційних ресурсів;
– нормативна правова база із захисту інформації;
– служби (органи) захисту інформації підприємств, організацій, установ.

Верховна Рада України як орган законодавчої влади приймає закони, що регулюють стосунки в галузі інформаційної безпеки та захисту інформації.

Нормативну правову базу формують на основі нормативних правових актів у галузі захисту інформації, що видають органи різних гілок влади.

Органи виконавчої влади (Кабінет Міністрів України (Уряд), до складу якого входять міністерства з підпорядкованими їм органами, місцеві органи виконавчої влади) виконують закони. Для цього Уряд приймає відповідні ухвали в галузі захисту інформації й видає розпорядження, що є підзаконними нормативними правовими актами. Міністерства з підпорядкованими їм органами відповідно до свого призначення розробляють і приймають ухвали й рішення, що є нормативними правовими актами свого рівня. Крім того, вони розробляють і затверджують такі нормативні акти, як: положення, інструкції, правила, методичні рекомендації. До нормативних актів цього рівня також належать накази й листи керівників органів виконавчої влади.

До органів влади, що регулюють відносини в галузі інформаційної безпеки та захисту інформації, належать:

– Служба безпеки України;

– Державна служба спеціального зв'язку та захисту інформації України;

– Міністерство внутрішніх справ України, якому підпорядковані Національна поліція України, Адміністрація Державної прикордонної служби України, Державна міграційна служба України, Державна служба України з надзвичайних ситуацій;

– Державна фіскальна служба України, що підпорядкована Міністерству фінансів України;

– Державне підприємство "Український науково-дослідний і навчальний центр проблем стандартизації, сертифікації та якості" (ДП "УкрНДНЦ");

Служба безпеки України – це державний правоохоронний орган спеціального призначення, який забезпечує державну безпеку України й підпорядкований Президенту України. Служба безпеки України виконує зокрема функції захисту державної таємниці.

Державна служба спеціального зв'язку та захисту інформації України (Держспецзв'язок) є державним органом, який призначений для забезпечення функціонування й розвитку державної системи урядового зв'язку, Національної системи конфіденційного зв'язку, формування та здійснення державної політики в сферах криптографічного та технічного захисту інформації, телекомунікацій, користування радіочастотним ресурсом України, поштового зв'язку спеціального призначення, урядового фельд'єгерського зв'язку, а також інших завдань відповідно до закону.

Міністерство внутрішніх справ України веде боротьбу із правопорушниками в інформаційній сфері й із комп'ютерними злочинами.

Державна фіскальна служба України зобов'язана запобігати незаконному ввезенню й вивезенню з України "піратської" продукції, забезпечуючи тим самим захист авторських і патентних прав.

Державне підприємство "Український науково-дослідний і навчальний центр проблем стандартизації, сертифікації та якості" розробляє стандарти в галузі захисту інформації, створює передумови наближення національної системи стандартизації до міжнародних і європейських норм та правил.

Судова влада здійснює нагляд і притягання до відповідальності за порушення законодавства в інформаційній сфері. У своїй діяльності суди керуються відповідними статтями Кримінального та Цивільного кодексів України.

Керівники підприємств, організацій, установ, відповідно до своїх посадових обов'язків, при роботі з інформацією, що становить державну або іншу таємницю, створюють службу (підрозділ) із захисту інформації. Для організовування відповідної діяльності вони видають нормативні правові акти: накази, розпорядження; а також затверджують інструкції, положення, правила, методичні рекомендації, пов'язані із захистом інформації та діяльністю служб захисту інформації.

Для діяльності, пов'язаної з державною таємницею, підприємство повинне мати ліцензію на цей вид діяльності, до його структури вводять спеціальний відділ; усі засоби захисту повинні бути сертифікованими.

1.2. Інформація як об'єкт захисту

1.2.1. Поняття інформації та її властивості

Розроблення законодавчої бази в галузі інформаційної безпеки пов'язане з подоланням труднощів, зумовлених специфікою такого продукту, як інформація [6–16].

Під ***інформацією*** розуміють задокументовані або оприлюднені відомості про події та явища, які відбуваються в суспільстві, державі чи навколишньому середовищі. Інформація може бути як у матеріалізованому, так і в нематеріалізованому (нефіксованому) вигляді. Без чітких меж, що визначають інформацію як об'єкт прав, застосування стосовно неї будь-яких законодавчих норм є проблемою.

Інформація як правова категорія має такі ***особливості***, що відрізняють її від інших товарів:

– нематеріальність – інформація стає доступною людині, якщо вона міститься на матеріальному носієві або передана усно, тому об'єктами захисту є матеріальні носії інформації або люди, які є носіями інформації;

– невичерпність – інформація не зникає під час споживання й може бути використана багато разів;

– збережуваність – інформація зберігається як завгодно довго у разі правильного зберігання носія інформації, проте із часом піддається лише старінню.

Виробництво інформації, на відміну від матеріального виробництва, потребує значних фінансових затрат порівняно із затратами на її тиражування.

У разі копіювання (що не змінює інформаційних параметрів носія) кількість інформації не змінюється, а її ціна знижується.

Властивості інформації:

– конфіденційність;

– цілісність;

– доступність.

Конфіденційність (confidentiality) – лише уповноважені користувачі можуть ознайомитися з інформацією.

Цілісність (integrity) – лише уповноважені користувачі можуть змінювати інформацію.

Доступність (availability) – уповноважені користувачі можуть отримати доступ до інформації згідно із правилами, що встановлені політикою безпеки, не очікуючи довше заданого (малого) проміжку часу.

Режим доступу до інформації – це передбачений правовими нормами порядок отримання, використання, розповсюдження та зберігання інформації. За режимом доступу інформацію поділяють на:

– відкриту (публічну);

– з обмеженим доступом (конфіденційну, таємну).

Держава здійснює контроль за режимом доступу до інформації.

Доступ до ***відкритої інформації*** забезпечують шляхом:

– систематичної публікації її в офіційних друкованих виданнях, бюлетенях, збірниках;

– поширення її засобами масової інформації;

– безпосереднього її надання зацікавленим громадянам, державним органам чи юридичним особам.

Порядок і умови надання громадянам, державним органам чи юридичним особам відкритої інформації встановлює Закон України "Про доступ до публічної інформації" або договори (угоди), якщо інформацію надають на договірній основі. Обмеження права на отримання відкритої інформації заборонене законом.

Переважним правом на отримання інформації користуються громадяни, яким вона необхідна для виконання своїх професійних обов'язків.

Конфіденційна інформація – це відомості, які знаходяться у володінні, користуванні або розпорядженні окремих фізичних чи юридичних осіб і поширюються за їх бажанням відповідно до передбачених ними умов.

З метою збереження нформації, що є власністю держави та знаходиться в користуванні органів державної влади чи органів місцевого самоврядування, підприємств, установ та організацій усіх форм власності, до неї можуть відповідно до закону встановити обмежений доступ і надати їх статусу конфіденційної. Порядок обліку, зберігання й використання документів та інших носіїв інформації, що містять зазначену інформацію, визначає Кабінет Міністрів України.

До ***таємної належить інформація***, що містить відомості, які становлять державну та іншу передбачену законом таємницю, розголошення якої завдає шкоди особі, суспільству чи державі.

Державна таємниця – інформація в сфері оборони, економіки, науки й техніки, зовнішніх відносин, державної безпеки та охорони правопорядку, розголошення якої може завдати шкоди національній безпеці держави, і яку спеціально охороняє держава. Перелік відомостей, які зараховують до державної таємниці, визначають державні експерти з таємниць. Його затверджує Служба безпеки України у формі "Зводу відомостей, що становлять державну таємницю".

Носіям інформації, які містять інформацію, що становить державну таємницю, надають один із грифів обмеження доступу (за зростання рівням таємності): "таємно", "цілком таємно" або "особливої важливості". Надання грифів обмеження доступу, призначених для державної таємниці, носіям інформації, що не містять відомостей, які підпадають під "Звід відомостей, що становлять державну таємницю", заборонено.

Відомостями, що становлять державну таємницю, зазвичай володіють посадові особи, допущені до них згідно з посадовими обов'язками.

Законодавство визначає поняття форми допуску як показника повноважень посадової особи щодо доступу до державної таємниці. Існує три форми допуску:

1-ша форма – допуск до державної таємниці із грифами обмеження доступу "таємно", "цілком таємно", "особливої важливості".

2-га форма – допуск до державної таємниці із грифами обмеження доступу "таємно", "цілком таємно".

3-тя форма – допуск до державної таємниці із грифом обмеження доступу "таємно".

За розголошення державної таємниці посадові особи несуть відповідальність, зокрема кримінальну.

Право власності на інформацію містить три складові:

– право володіння;

– право використання;

– право розпорядження.

Право володіння передбачає право мати інформацію в незмінному вигляді. ***Право використання*** передбачає використання інформації у своїх інтересах. У цьому випадку окрім суб'єкта права власності до інформації можуть мати доступ й інші суб'єкти. Суб'єкт права власності на інформацію може передати частину своїх прав (***розпорядження***), не втрачаючи при цьому їх сам.

Інформація – об'єкт права власності громадян (фізичних осіб), юридичних осіб і держави. Підставами виникнення прав власності на інформацію є:

– створення інформації своїми силами та за свій рахунок;

– договір на створення інформації;

– договір, що містить умови переходу прав власності на інформацію до іншої особи.

Інформація, створена кількома громадянами або юридичними особами, є колективною власністю її творців. Порядок і правила користування такою власністю визначає договір, укладений між співвласниками. Інформація, створена юридичними особами або придбана ними іншим законним способом є їх власністю.

Інформація, створена на кошти державного бюджету, є державною власністю. Інформацію, створену на правах особистої власності, може бути віднесено до державної власності у випадках передання її на зберігання у відповідні банки даних, фонди або архіви на договірній основі.

Власник інформації має право призначати особу, яка здійснює володіння, використання й розпорядження інформацією, а також визначати правила оброблення інформації та доступ до неї, установлювати інші умови щодо інформації.

Порушення законодавства України про інформацію спричиняє дисциплінарну, цивільно-правову, адміністративну або кримінальну відповідальність згідно із законодавством України. Відповідальність за порушення законодавства про інформацію несуть особи, винні у вчиненні таких порушень як:

– необґрунтована відмова від надання відповідної інформації;

– надання інформації, що не відповідає дійсності;

– несвоєчасне надання інформації;

– навмисне приховування інформації;

– примушення до поширення або перешкоджання поширенню певної інформації, а також цензура;

– поширення відомостей, що не відповідають дійсності, ганьблять честь і гідність особи;

– безпідставна відмова від поширення певної інформації;

– використання й поширення інформації стосовно особистого життя громадянина без його згоди особою, яка є власником відповідної інформації внаслідок виконання своїх службових обов'язків;

– розголошення державної або іншої таємниці, що охороняється законом, особою, яка повинна охороняти цю таємницю;

– порушення порядку зберігання інформації;

– навмисне знищення інформації;

– необґрунтоване віднесення окремих видів інформації до категорії відомостей з обмеженим доступом;

– порушення порядку обліку, зберігання й використання документів та інших носіїв інформації, які містять конфіденційну інформацію, що є власністю держави.

Особа звільняється від відповідальності за розголошення інформації з обмеженим доступом, якщо суд установить, що ця інформація є суспільно значущою.

Актуальним завданням будь-якого підприємства, що створює або використовує у своїй діяльності продукти інтелектуальної праці, є вживання певних заходів із запобігання витоку, розкраданню, втраті, спотворенню чи іншим формам незаконного використання інформації.

Відповідно до законодавства України захисту підлягає будь-яка документована інформація, неправомірне поводження з якою може завдати збитку її власникові, користувачеві чи іншій особі.

Держава, володіючи інформацією, що представляє національне надбання або що містить відомості обмеженого доступу, неправомірне поводження з якою може завдати збитку її власникові, здійснює спеціальні заходи, що забезпечують контроль за її використанням і якістю захисту.

Режим захисту інформації встановлюють:

– відносно відомостей, віднесених до державної таємниці, – уповноваженими органами на підставі Закону України "Про державну таємницю";

– відносно конфіденційної документованої інформації – власником інформаційних ресурсів або уповноваженою особою на підставі закону;

– відносно особистих даних – законодавством України.

Організації, що обробляють інформацію з обмеженим доступом, яка є власністю держави, створюють спеціальні служби, що забезпечують захист інформації.

Власник інформаційних ресурсів має право здійснювати контроль за виконанням вимог із захисту інформації та забороняти або припиняти оброблення інформації у випадку невиконання цих вимог, а також звертатися в органи державної влади для оцінювання правильності виконання норм і вимог із захисту його інформації.

Власник документів або інформаційних систем відповідно до закону встановлює порядок надання користувачеві інформації. Власник документів забезпечує необхідний рівень захисту інформації відповідно до законодавства України.

Ризик, пов'язаний із використанням несертифікованих інформаційних систем і засобів їх забезпечення, лежить на власнику цих систем і засобів. Ризик, пов'язаний із використанням інформації, отриманої з несертифікованої системи, лежить на споживачеві інформації.

1.2.2. Загрози інформації

Основним поняттям інформаційної безпеки є ***загроза інформації*** (threat) – можливість виникнення такого явища або події, наслідком якого можуть бути небажані впливи на інформацію. ***Інцидентом інформаційної безпеки*** називають здійснену загрозу інформації [6–16].

Усі ***загрози інформації*** можна об'єднати в три групи:

– загроза розкриття – можливість того, що інформація стане відомою тому, кому не потрібно її знати;

– загроза цілісності – навмисна несанкціонована зміна інформації, яку зберігають у системі або передають з однієї системи в іншу;

– загроза відмови в обслуговуванні – небезпека блокування доступу до деякого інформаційного ресурсу.

З усієї множини способів класифікації загроз найпридатнішою для аналізу є класифікація загроз за результатами їх впливу на інформацію. У такому випадку розрізняють такі види загроз інформації:

– порушення конфіденційності (руйнування захисту, зменшення ступеня захищеності інформації):

а) загрози при керуванні потоками інформації;

б) загрози існування прихованих каналів для передавання потоків інформації;

в) порушення конфіденційності при передаванні інформації через незахищене середовище;

– порушення логічної цілісності (порушення логічних зв'язків) та фізичної цілісності (знищення, порушення елементів):

а) загрози при керуванні потоками інформації;

б) неможливість повернення захищеної інформаційної системи у вихідний стан;

в) порушення цілісності при передаванні інформації через незахищене середовище;

г) порушення змісту інформації (зміна блоків інформації, зовнішнє нав'язування помилкової інформації);

– порушення доступності чи відмовлення в обслуговуванні:

а) порушення при керуванні послугами й ресурсами користувача;

б) порушення стійкості до відмов;

в) порушення при гарячій заміні;

г) порушення при відновленні після збоїв;

– порушення спостереженості чи керованості:

а) порушення при реєструванні небезпечних дій;

б) порушення при ідентифікації та автентифікації;

– порушення прав власності на інформацію:

а) несанкціоноване копіювання інформації;

б) несанкціоноване використання інформації.

Причинами загроз інформації можуть бути певні обставини, події чи фактори, що будуть перешкоджати здійсненню конкретних захисних механізмів і заходів. До них належать навмисні та природні фактори.

Навмисні фактори такі:

– розкрадання носіїв інформації;

– підключення до каналів зв'язку;

– перехоплення електромагнітних випромінювань;

– несанкціонований доступ;

– розголошення інформації;

– копіювання інформації.

Природні фактори такі:

– нещасні випадки (пожежі, аварії, вибухи);

– стихійні лиха (урагани, повені, землетруси);

– помилки в процесі оброблення інформації (помилки користувача, помилки оператора, збої апаратури).

Отже, загрози інформації є результатом обставин, подій чи факторів, що перешкоджають захисту інформації. Надалі такі фактори називатимемо дестабілізуючими. ***Дестабілізуючі фактори*** – це такі явища чи події, що можуть з'являтися на будь-якому етапі життєвого циклу інформаційної системи, і наслідком яких можуть бути загрози інформації. При цьому інформаційна система є сукупністю телекомунікаційних мереж і засобів для накопичування, оброблення, зберігання та розповсюджування інформації (даних).

Існують такі види дестабілізуючих факторів:

– кількісна недостатність – фізична нестача компонентів інформаційної системи для забезпечення необхідного рівня захищеності оброблюваної інформації;

– якісна недостатність – недосконаліть конструкції складових інформаційної системи, внаслідок чого не забезпечено необхідного рівня захищеності оброблюваної інформації;

– відмова елементів інформаційної системи (постійна відмова) – порушення працездатності елементів, що призводить до неможливості виконання ними своїх функцій;

– збій елементів інформаційної системи (тимчасова відмова) – тимчасове порушення працездатності елементів, що призводить до неправильного виконання ними в певний момент часу своїх функцій;

– помилки елементів інформаційної системи – неправильне (одноразове чи систематичне) виконання елементами своїх функцій унаслідок специфічного (постійного й / або тимчасового) їхнього стану;

– стихійні лиха – випадкові неконтрольовані явища, що спричиняють фізичні руйнування;

– злочинні дії – дії людей, що спеціально спрямовані на порушення захищеності інформації;

– побічні явища – явища, що супроводжують виконання елементом інформаційної системи своїх функцій.

Отже, дестабілізуючі фактори можуть мати об'єктивну природу (наприклад, відмови, збої тощо) чи суб'єктивну (наприклад, дії зловмисників). В останньому випадку дестабілізуючі фактори можуть бути випадковими чи навмисними. Випадковість чи навмисність можуть бути відносними. Наприклад, іноді свідомо додані в програмне забезпечення функції можуть заздалегідь давати можливість ненавмисних дій (наприклад, при дистанційному налаштуванні системи).

Джерелами дестабілізуючих факторів можуть бути як компоненти інформаційної системи, так і зовнішнє середовище. Розрізняють такі джерела дестабілізуючих факторів:

– персонал – люди, які мають будь-який стосунок до функціонування інформаційної системи;

– технічні засоби;

– моделі, алгоритми, програми;

– технологія функціонування – сукупність засобів, прийомів, правил, заходів та угод, які використовують у процесі оброблення інформації;

– зовнішнє середовище – сукупність елементів, що не входять до складу інформаційної системи, але здатні впливати на захищеність інформації в ній, зокрема порушники.

1.2.3. Модель порушника

Серед дестабілізуючих факторів особливої уваги заслуговують такі, що спричинені діями людини. Людина може чинити несвідомі шкідливі (помилкові) дії або свідомі шкідливі (злочинні) дії. У першому випадку людину доцільно вважати ***порушником*** (violator), а в іншому – ***зловмисником*** (malefactor, intruder) [6–16].

Також існує поняття ***хакер*** (hacker) і ***кракер*** (cracker). Як хакер, так і кракер намагаються зламати захист інформаційної системи. Основна відмінність між ними полягає в постановці мети зламування системи: хакер вирішує дослідницьку задачу з оцінювання та знаходження слабких місць із метою подальшого підвищення надійності роботи та безпеки інформаційної системи. Кракер виконує вторгнення в систему з метою руйнування, крадіжки, псування, змінювання інформації та робить правопорушення з корисливими намірами швидкого збагачення.

У цьому розділі вважатимемо, що ***порушником*** (user violator) є користувач, який здійснює несанкціонований доступ до інформації. Для подальшої організації надійного захисту інформації підприємства чи організації повинні не лише оцінити весь спектр можливих загроз безпеці інформації, але й спробувати виявити категорії порушників та ті методи, які вони використовують. Для цього слід розробити модель порушника. Оскільки під порушником розуміють людину, то цілком зрозуміло, що створення його формалізованої моделі є дуже складним завданням. Тому в багатьох випадках може йтися лише про неформальну або описову модель порушника. Загалом, ***модель порушника*** – це абстрактний формалізований або неформалізований опис порушника. Розглянемо модель порушника в інформаційній системі детальніше.

Порушник – це людина, яка може отримати доступ до роботи із засобами чи ресурсами інформаційної системи. Вона може помилково, внаслідок необізнаності, цілеспрямовано, свідомо чи несвідомо, використовуючи різні можливості, методи та засоби, спробувати виконати операції, які призводять або можуть призвести до порушення властивостей інформації, що визначені політикою безпеки системи.

У кожному конкретному випадку для кожної інформаційної системи визначають імовірні загрози й моделі потенційних порушників – виконавців цих загроз, зокрема можливі сценарії здійснення загроз. Модель порушника відображає його практичні та потенційні можливості, апріорні знання, час та місце дії тощо. При розробленні моделі порушника визначають:

– припущення щодо категорії осіб, до яких може належати порушник;

– припущення щодо мотивів дій порушника (мети, яку він переслідує);

– припущення щодо рівня кваліфікації та обізнаності порушника, його технічної оснащеності (щодо методів та засобів, які він використовує для здійснення порушень);

– обмеження та припущення щодо характеру можливих дій порушників (за часом та місцем дії та інші).

Розглядають такі типи порушників:

– зовнішні;

– внутрішні.

Серед ***зовнішніх порушників*** виділяють такі:

– добре озброєна й оснащена силова група, що діє ззовні швидко й напролом;

– поодинокий порушник, який не має допуску до інформаційної системи, намагається діяти потайки й обережно, оскільки усвідомлює, що сили реагування мають над ним перевагу.

Серед потенційних ***внутрішніх порушників*** можна відзначити:

– допоміжний персонал, який є допущеним до інформаційної системи, але не допущеним до життєво її важливого центру;

– основний персонал, який є допущеним до життєво важливого центру інформаційної системи (найбільш небезпечний тип порушників);

– співробітників служби безпеки підприємства чи організації, які часто формально й не допущені до життєво важливого центру системи, але насправді мають достатньо великі можливості для збирання необхідної інформації та вчинення зловмисних дій.

Також необхідно розглядати можливість змови між порушниками різних типів, що ще більше ускладнює задачу формалізації моделі порушника.

Серед внутрішніх порушників можна виділити такі категорії персоналу:

– користувачі (оператори) інформаційної системи;

– персонал, який обслуговує технічні засоби (інженери, техніки);

– співробітники відділів розроблення та супроводження програмного забезпечення (прикладні та системні програмісти);

– технічний персонал, який обслуговує будівлю (прибиральниці, електрики, сантехніки та інші співробітники, що мають доступ до будівлі та приміщення, де розташовані компоненти інформаційної системи);

– співробітники служби безпеки підприємства чи організації;

– керівники підприємства чи організації різних рівнів.

Сторонні особи, які можуть бути порушниками:

– клієнти (представники інших підприємств чи організацій, громадяни);

– відвідувачі (запрошені з якого-небудь приводу);

– представники інших організацій, які займаються забезпеченням життєдіяльності цього підприємства чи організації (енерго-, водо-, теплопостачання тощо).

– представники конкуруючих організацій (іноземних служб) або особи, які діють за їхнім завданням;

– особи, які випадково або ненавмисно порушили пропускний режим (не маючи на меті порушити безпеку);

– будь-які особи за межами контрольованої зони.

Можна виділити також такі ***основні мотиви порушень***:

– безвідповідальність;

– самоствердження;

– підкуп;

– шантаж;

– ідеологічні мотиви.

При порушеннях, викликаних безвідповідальністю, користувач здійснює руйнівні дії, які не пов'язані зі злим умислом. У більшості випадків – це наслідок некомпетентності, недбалості або невдоволення.

Деякі користувачі вважають отримання доступу до інформації в системі значним успіхом, починаючи щось подібне на гру "користувач проти системи" заради самоствердження у власних очах або в очах колег, знайомих.

Порушення безпеки інформаційної системи може бути поясненим корисливим інтересом користувача системи, наприклад, підкупом із боку конкуруючого підприємства. У цьому випадку такий користувач цілеспрямовано намагатиметься перебороти систему захисту для доступу до інформації в системі.

У ряді випадків, коли конкуруюче підприємство не може отримати необхідну інформацію підкупом, воно вдається до шантажу працівників, які володіють необхідною інформацією.

Причиною порушень можуть бути ідеологічні мотиви, коли порушник намагається передати інформацію, до якої отримав доступ, іншим особам, підприємствам або оприлюднити її в засобах масової інформації для досягнення певної "великої мети". Також такого виду порушення можливі, якщо порушник керується настановами політичних рухів, партій чи громадських об'єднань.

За рівнем знань про інформаційну систему розрізняють порушників, які:

– знають функціональні особливості інформаційної системи, основні закономірності формування в ній масивів даних і потоків запитів до них, уміють користуватися штатними засобами;

– мають високий рівень знань, досвід роботи чи обслуговування технічних засобів інформаційної системи;

– мають високий рівень знань у галузі програмування й обчислювальної техніки, проектування й експлуатації інформаційних систем;

– знають структуру, функції та механізм дії засобів захисту, їхні сильні й слабкі сторони.

За рівнем можливостей (методами та засобами, що використовують) розрізняють порушників, які:

– застосовують лише агентурні методи отримання відомостей;

– застосовують пасивні засоби (технічні засоби перехоплення без змінювання компонентів інформаційної системи);

– використовують лише штатні засоби та недоліки системи захисту для її подолання (несанкціоновані дії з використанням дозволених засобів), а також компактні магнітні носії інформації, які можуть бути потайки пронесені через пости охорони;

– застосовують методи та засоби активного впливу (підключення додаткових технічних засобів, підключення до каналів передавання даних, впровадження програмних закладок, використання спеціального програмного забезпечення).

Порушники можуть відрізнятись за часом дії:

– у процесі функціонування інформаційної системи;

– у період неактивності інформаційної системи (у неробочий час, під час планових перерв у її роботі, перерв для обслуговування та ремонтів тощо);

– як у процесі функціонування, так і в період неактивності інформаційної системи.

Також порушники можуть відрізнятись за місцем дії:

– без доступу на контрольовану територію підприємства чи організації;

– з контрольованої території підприємства чи організації без доступу до будівель та споруд;

– усередині приміщень підприємства чи організації, але без доступу до технічних засобів інформаційної системи;

– з робочих місць користувачів (операторів) інформаційної системи;

– з доступом до зони даних (баз даних, архівів тощо) інформаційної системи;

– з доступом до зони управління засобами забезпечення безпеки інформаційної безпеки.

Розробляючи модель порушника, ураховують також такі обмеження й припущення про характер дій можливих порушників:

– робота з підбору кадрів і спеціальні заходи на підприємстві чи в організації ускладнюють можливість створення угрупувань порушників, тобто змови й цілеспрямованих дій із подолання системи захисту двох і більше порушників;

– порушник, плануючи спробу несанкціонованого доступу, приховує свої несанкціоновані дії від інших співробітників;

– несанкціонований доступ може бути наслідком помилок користувачів, системних адміністраторів, а також недоліків застосованої в інформаційній системі технології оброблення інформації тощо.

Визначення конкретних характеристик можливих порушників є значною мірою суб'єктивним. Модель порушника, яку побудовано з урахуванням особливостей конкретної предметної галузі й технології оброблення інформації, можна подати перелічуванням кількох варіантів його образу. Кожний вид порушника має бути охарактеризований згідно із класифікаціями, наведеними вище. Усі значення характеристик мають бути оцінені (наприклад, за 5-бальною системою) і зведені до відповідних форм.

Однак у результаті формування моделі порушника необхідно визначити: імовірність здійснення загрози, своєчасність виявлення порушення й відомості про таке порушення.

Необхідно зауважити, що людина (користувач інформаційної системи) є основною причиною та рушійною силою порушень і злочинів. Отже, питання безпеки захищених інформаційних систем фактично є питанням людських відносин та людської поведінки.

1.2.4. Підготовчі дії порушника перед несанкціонованим доступом до інформації

Фахівець із забезпечення безпеки інформаційної системи підприємства чи організації повинен знати, яких загрозливих дій можна очікувати від порушника, щоб змогти вчасно запобігти інцидентам інформаційної безпеки [6–16]. Перед здійсненням несанкціонованого доступу до ресурсів інформаційної системи порушнику потрібно, як правило, здійснити такі підготовчі дії:

– зібрати відомості про інформаційну систему;

– виконати пробні спроби входження до інформаційної системи.

Збирання відомостей дає можливість зловмиснику знайти недоліки захисту інформаційної системи, та надалі використати їх для несанкціонованого доступу до ресурсів інформаційної системи. Можливі різні напрямки збирання відомостей про інформаційну систему порушником:

– аналіз повідомлень у засобах масової інформації, відомчих бюлетенів і документації;

– отримання інформації від співучасників;

– підкуп, шантаж, діяльність з ідеологічних мотивів;

– організовування крадіжок;

– підглядання, підслуховування розмов;
– аналіз змісту викинутих роздруківок;
– перехоплення інформації спеціальними технічними засобами;

Порушник може почерпнути багато корисної інформації про роботу підприємства чи організації з повідомлень у засобах масової інформації, зокрема газет, інших періодичних видань, радіопередач, телепередач, а також відомчих бюлетенів, рекламних проспектів. Із цією метою також може бути корисною технічна документація на продукцію підприємства, що передана іншим підприємствам або організаціям.

Інші напрямки збирання відомостей про інформаційну систему можливо використовувати, якщо порушник працює на цьому підприємстві або знаходить співучасників, які там працюють. Для знаходження співучасників необхідно завести знайомства із працівниками підприємства. При цьому працівники можуть несвідомо або свідомо надавати інформацію, що необхідна порушнику. Наприклад, знайомлячись, порушник може: представитись менеджером; використати анкети, роздаючи їх у фойє фірми й детально розпитуючи співробітників про інформаційну систему; дзвонити системному адміністраторові в обідній час із проханням нагадати нібито забутий пароль; прогулюватись по будівлі підприємства, спостерігаючи за доступом до системи; установлювати контакти із не зайнятими в цей момент працівниками охорони, яким відвідувачі при вході в будівлю підприємства повинні пред'являти перепустки, називати ідентифікаційні коди чи паролі.

Достатньо ефективним є метод "полювання за мізками", коли на фірму приходить порушник, що видає себе за людину, яка бажає працювати системним програмістом або інженером електрозв'язку, і просить дати йому консультацію. Дуже багато інформації можна отримати від працівника, який не має перспективи росту, але вважає себе гідним більш важливої й високооплачуваної посади. Він може розкрити ідентифікатори користувачів, паролі, вказати слабкі місця в інформаційній системі.

Якщо необхідно залучити працівників підприємства чи організації для свідомого надання інформації, порушник використовує їх підкуп, шантаж або ідеологічно їх мотивує. Злочинний світ традиційно грає на людських слабкостях і нещастях, таких, як надмірне захоплення азартними іграми, сімейні негаразди, фінансові проблеми, борги, оплата медичних рахунків тощо.

Адміністратори й менеджери фірм мають можливість брати роботу додому або за необхідності передавати службову інформацію відкритими каналами електрозв'язку, зокрема мережею Інтернет. Комівояжери можуть робити угоди, використовуючи термінали в номерах готелів, або отримувати доступ до інформації безпосередньо із салону автомобіля. Це створює грунт для

здійснення крадіжок у будинках, автомобілях, готелях із метою отримання інформації для подальшого несанкціонованого доступу до ресурсів інформаційної мережі.

Важливим напрямком отримання порушником відомостей про інформаційну систему є підглядання, підслуховування розмов у барах, фойє готелів, ресторанах, таксі, вивчення вмісту загублених портфелів і документів.

Порушник може отримати конфіденційну інформацію, аналізуючи вміст викинутих роздруківок, які повинні були бути знищеними.

Перехоплення інформації спеціальними технічними засобами потребує наявності спеціальних навичок порушника та обладнання для несанкціонованого прослуховування телефонних ліній зв'язку, записування відеозображень, аналізу трафіку інформаційної мережі, аналізу протоколів, що застосовані в інформаційній мережі, перехоплення електромагнітних випромінювань тощо. Це можливо здійснити у випадку, якщо порушник має доступ до приміщень, у яких знаходяться робочі місця важливих із погляду можливості отримання інформації працівників підприємства, або приміщень, у яких розташоване мережеве обладнання або сервери.

Отримавши необхідний обсяг попередньої інформації, порушник робить наступний крок – здійснює спробу несанкціонованого доступу до інформаційної системи для перевіряння такої можливості. Використовувані ним при цьому засоби залежатимуть від кількості наявної в нього інформації. Якщо спроба є вдалою, порушник має змогу здійснити несанкціонований доступ до ресурсів інформаційної системи з метою розкриття, зміни або знищення інформації.

1.2.5. Методи та види несанкціонованого доступу. Методи захисту від несанкціонованого доступу

Несанкціонований доступ – доступ до інформації з використанням засобів, включених до складу інформаційної системи, що порушує встановлені правила розмежування доступу. Несанкціонований доступ може бути здійснений як із використанням штатних засобів, тобто сукупності програмно-апаратного забезпечення, включеного до складу інформаційної системи розробником під час розроблення або системним адміністратором у процесі експлуатації, так і з використанням програмно-апаратних засобів, включених до складу інформаційної системи зловмисником [6–16].

Під ***захистом від несанкціонованого доступу*** слід розуміти діяльність, спрямовану на забезпечення дотримання правил розмежування доступу шляхом створення й підтримування в дієздатному стані системи заходів із захисту інформації.

До основних методів несанкціонованого доступу належать:

– безпосереднє звернення до інформаційної системи з метою отримання певного виду доступу;

– створення програмно-апаратних засобів, що виконують звертання до інформаційної системи в обхід засобів захисту;

– змінювання засобів захисту, що дає змогу здійснити несанкціонований доступ;

– упровадження до інформаційної системи програмних або апаратних засобів, що порушують структуру й функції інформаційної системи, а також дають можливість здійснити несанкціонований доступ.

Розглянемо основні види несанкціонованого доступу.

Обхідний шлях (люк) – це програмний блок, вбудований у певне програмне забезпечення, який дає змогу обійти систему захисту інформації або реєстрацію в системному журналі. Зазвичай, ділянки програми, в яких здійснено обхідний шлях, вбудовують під час розроблення програмного забезпечення. Зловмисник також може здійснювати несанкціонований доступ шляхом виявлення та подальшого використання обхідних шляхів, вбудованих в операційну систему комп'ютера.

"Троянський кінь" – програма, яка має вигляд корисної для користувача програми, але додатково містить програмний блок, що несе загрозу інформаційній системі або інформації в системі. Ця програма використовує обман, щоб спонукати користувача її запустити. Цей програмний блок може спрацьовувати при настанні якої-небудь умови (дати, стану системи) або за командою ззовні. Користувач, який запустив таку програму, наражає на небезпеку не лише свої інформаційні файли, але й інформаційну систему загалом. До можливих деструктивних дій "троянського коня" належать:

– знищення інформації або програмного забезпечення, причому об'єкти і способи знищення вибирає зловмисник (автор шкідливої програми);

– перехоплення й передавання інформації до зловмисника, наприклад, перехоплення паролів, які набирають на клавіатурі;

– цілеспрямоване змінювання тексту програми, яка здійснює функції безпеки й захисту інформаційної системи.

Часто "троянський кінь" після зміни тексту програми, яка здійснює функції безпеки й захисту інформаційної системи, може подавати зловмиснику умовний сигнал про можливість доступу до файлів. Після подання сигналу "троянський кінь" деякий час очікує, а потім повертає файл у початковий стан. Отже, такий алгоритм дає змогу програмі зловмисника чинити будь-які несанкціоновані дії з файлами без їх реєстрування.

"Троянський кінь" є одним із найнебезпечніших видів несанкціонованого доступу. Радикальний спосіб захисту від нього полягає в створенні замкнутого

середовища виконання програмного забезпечення, яке необхідно зберігати й захищати від несанкціонованого доступу.

Комп'ютерний вірус – програма або програмний блок, що, на відміну від "троянського коня", крім виконання дій, що небажані для законних користувачів, додатково виробляє й розповсюджує свої копії в інформаційній системі.

Для захисту від указаних шкідливих програм уживають такі заходи:

– заборона несанкціонованого доступу до виконуваних файлів;

– старанне тестування нових програмних засобів;

– контроль цілісності файлів, що виконуються, і системних областей пам'яті;

– створення замкнутого середовища виконання програм.

Логічна бомба – програма або програмний блок, що здійснює деяку функцію у разі виконання певної умови. Логічні бомби використовують для змінювання або знищення інформації, рідше для крадіжки або шахрайства.

Атака – використання вразливостей програмного забезпечення інформаційної системи для досягнення цілей, що виходять за межі допуску певного суб'єкта в систему. За допомогою атаки, що полягає в здійсненні певних нестандартних дій, зловмисник може отримати доступ до певної інформації в системі або отримати додаткові певні права доступу для роботи в системі. У більшості інформаційних систем кожен користувач отримує певні права доступу. Несанкціоноване отримання додаткових прав доступу дає змогу зловмиснику виконувати певні дії, обходячи систему захисту. Слід зазначити, що незаконне отримання додаткових прав доступу можливе або за наявності помилок у системі захисту, або через недбалість адміністратора при керуванні системою й призначенні прав доступу користувачів.

Аналіз трафіку – аналіз частоти й методів доступу користувачів в інформаційній системі за допомогою програмно-апаратних засобів зловмисника, підключених до цієї системи. Це дає змогу зловмисникові отримати несанкціонований доступ до ресурсів інформаційної системи під виглядом законного користувача.

Між рядків – підключення обладнання зловмисника до лінії зв'язку, що дає змогу використовувати ресурси інформаційної системи під виглядом законного користувача в проміжках часу між діями такого користувача.

Розрив лінії – переключення лінії зв'язку від обладнання законного користувача до обладнання зловмисника після закінчення його сеансу зв'язку або через розрив лінії. При цьому зловмисник працює в інформаційній системі як законний користувач.

"Маскарад" – виконання яких-небудь дій зловмисником від імені іншого користувача, який володіє відповідними повноваженнями. Метою "маскараду"

є приписування яких-небудь дій зловмисника іншому користувачеві або присвоєння йому прав доступу іншого користувача, наприклад:

– вхід в інформаційну систему під іменем і паролем іншого користувача (цьому "маскараду" передує перехоплення пароля);

– передавання повідомлень у мережі від імені іншого користувача;

– здійснення банківських електронних платежів грошима з рахунків законних клієнтів банків.

"Підкладання свині" – підключення обладнання зловмисника до лінії зв'язку й імітація роботи системи з метою отримання інформації про ідентифікацію законного користувача. Наприклад, зловмисник може імітувати зависання системи й процедуру повторного входу до неї. Користувач, не здогадуючись про це, знову вводить свій ідентифікатор і пароль, після чого зловмисник повертає йому управління системою, яка нормально працює.

Повторне використання ресурсів – зчитування інформації, що призначена для знищення. Об'єктами атаки можуть бути витерті файли, що тимчасово розміщені в смітнику операційної системи, тимчасові файли, інформація в різних буферах, секторах магнітних дисків, зонах магнітних стрічок, реєстрах пам'яті тощо. Для зчитування даних безпосередньо з пам'яті іноді достатньо створити невелику програму, яка робить запит під час виконання динамічного виділення пам'яті великого об'єму. Потім у результаті навмисної помилки ця програма може аварійно завершити свою роботу й видати інформацію про вміст усіх ділянок пам'яті, які було використано перед цим.

Програма-імітатор – програма, що імітує роботу того чи іншого елемента мережі й створює в користувача інформаційної системи ілюзію взаємодії із системою з метою, наприклад, перехоплення інформації користувача. Зокрема, екранний імітатор дає можливість заволодіти паролями користувачів. Операцію перехоплення паролів здійснюють таким чином. За спроби законного користувача ввійти до системи така програма імітує на екрані дисплея введення імені та пароля користувача, які відразу пересилає зловмисникові (власникові програми-імітатора), після чого на екран виводить повідомлення про помилку й повертає управління операційній системі. Користувач вважає, що припустився помилки під час введення пароля. Він повторює введення й отримує доступ до системи. Зловмисник, який отримав ім'я й пароль законного користувача, може тепер використовувати їх зі своєю метою.

Розглянуті методи можуть бути застосовані для втілення таких найпоширеніших ***сценаріїв несанкціонованого доступу***:

– перегляд інформації;

– копіювання програм та даних;

– читання даних із лінії зв'язку;

– змінювання потоку повідомлень;

– змінювання алгоритмів програм;

– змінювання апаратної частини інформаційної системи;

– змінювання режиму обслуговування або умов експлуатації інформаційної системи;

– програмні закладки;

– переривання процесу функціонування інформаційної системи або її компонентів;

– переривання потоку повідомлень;

– переривання роботи програмного забезпечення;

– фізичне руйнування апаратних засобів системи;

– змінювання інформації;

– додавання фальшивих процесів і підмінювання справжніх процесів фальшивими;

– додавання фальшивих апаратних засобів;

– імітація роботи апаратно-програмних компонентів системи з боку суб'єктів загрози;

– знищення інформації.

Звичайно, зловмисник може використовувати різні комбінації наведених вище сценаріїв, що дуже ускладнює організацію захисту від несанкціонованого доступу.

Можна виділити такі узагальнені ***категорії методів захисту від несанкціонованого доступу***:

– правові;

– організаційні;

– технологічні;

– фізичні;

– апаратні;

– програмні.

Правові методи передбачають заходи контролю за виконанням нормативних актів загальнодержавного значення, механізми розроблення й удосконалення нормативної бази, яка регулює питання захисту інформації.

Організаційні методи базуються на раціональній організації роботи й адміністрування інформаційної системи, зокрема конфігуруванні й адмініструванні операційних систем, регламентуванні повноважень адміністратора, розробленні набору обов'язкових інструкцій, що визначають порядок доступу й роботи в мережі.

Технологічні методи грунтуються на технології виконання мережевого адміністрування, моніторингу й аудиту безпеки інформаційних ресурсів,

ведення електронних журналів реєстрування користувачів, фільтрування й антивірусного оброблення прийнятої інформації.

За допомогою фізичних методів збереження інформації забезпечують фізичну охорону носіїв інформації від викрадення, доступу, змінювання чи знищення інформації на них, а також доступу до інформаційної системи з метою отримання, змінювання чи знищення інформації, що передають, накопичують, обробляють, зберегають чи розповсюджують у такій системі.

Апаратні методи базуються на забезпеченні фізичного захисту інформаційної системи від несанкціонованого доступу, застосуванні апаратних функцій ідентифікації периферійних терміналів чи користувачів, апаратних шифраторів тощо.

Програмні методи забезпечують захист за допомогою програм ідентифікації користувачів, парольного захисту й перевірки повноважень, міжмережевих екранів, криптопротоколів тощо.

Розгляду методів забезпечення інформаційної безпеки та захисту інформації присвячені наступні розділи цього навчального посібника, зокрема розглянуто криптологію, що вивчає методи втаємничення інформації, стеганографію, що вивчає методи приховування інформації, методи ідентифікації, автентифікації, санкціонованого доступу, методи та засоби забезпечення безпеки інформаційних систем, захисту програмного забезпечення в інформаційних системах, інформаційної безпеки підприємств та організацій, принципи побудови та функціонування систем інформаційної безпеки.

Контрольні питання до розділу 1

1. Що таке інформація?
2. Що таке безпека?
3. Поняття та складові державної безпеки.
4. Поняття інформаційної безпеки.
5. Державна політика інформаційної безпеки та її здійснення в законодавстві України.
6. Напрями розвитку законодавчої бази в галузі інформаційної безпеки.
7. Органи забезпечення інформаційної безпеки та захисту інформації.
8. Властивості інформації.
9. Відповідальність за порушення законодавства України про інформацію.
10. Загрози інформації.
11. Види дестабілізуючих факторів.
12. Поняття порушника.
13. Поняття зловмисника.
14. Модель порушника.
15. Підготовчі дії порушника перед несанкціонованим доступом до інформації.
16. Методи несанкціонованого доступу.
17. Види несанкціонованого доступу.
18. Сценарії несанкціонованого доступу.
19. Категорії методів захисту від несанкціонованого доступу.

Список літератури до розділу 1

1. Про основи національної безпеки України [Електронний ресурс] : закон України № 964-IV : [прийнятий Верховною Радою України 19 червня 2003 р. : редакція від 9 травня 2018 р.]. – Режим доступу : http://zakon5.rada.gov.ua/laws/show/964-15.
2. Стратегія національної безпеки України [Електронний ресурс] : [затверджена Указом Президента України "Про рішення Ради національної безпеки і оборони України від 6 травня 2015 року "Про Стратегію національної безпеки України"" від 26 травня 2015 р. № 287/2015]. – Режим доступу : http://zakon5.rada.gov.ua/laws/show/287/2015/paran14#n14.
3. Стратегія кібербезпеки України [Електронний ресурс] : [затверджена Указом Президента України "Про рішення Ради національної безпеки і оборони України від 27 січня 2016 року "Про Стратегію кібербезпеки України"" від 15 березня 2016 р. № 96/2016]. – Режим доступу : http://zakon5.rada.gov.ua/laws/show/96/2016/paran11#n11.
4. Воєнна доктрина України [Електронний ресурс] : [затверджена Указом Президента України "Про рішення Ради національної безпеки і оборони України від 2 вересня 2015 року "Про нову редакцію Воєнної доктрини України"" від 24 вересня 2015 р. № 555/2015]. – Режим доступу : http://zakon5.rada.gov.ua/laws/show/555/2015/paran17#n17.
5. Про національну безпеку України [Електронний ресурс] : закон України № 2469-VIII : [прийнятий Верховною Радою України 21 червня 2018 р. : набрання чинності 8 липня 2018 р.]. – Режим доступу : http://zakon.rada.gov.ua/laws/show/2469-19.
6. Гарасимчук О. І. Комплексні системи санкціонованого доступу : навч. посіб. / О. І. Гарасимчук, В. Б. Дудикевич, В. А. Ромака. – Львів : Видавництво Львівської політехніки, 2010. – 212 с.
7. Горбатий І. В. Телекомунікаційні системи та мережі. Принципи функціонування, технології та протоколи : навч. посіб. / І. В. Горбатий, А. П. Бондарєв. – Львів : Видавництво Львівської політехніки, 2016. – 336 с.
8. Юдін О. К. Інформаційна безпека держави / О. К. Юдін, В. М. Богуш. – Харків: Консум, 2004. – 508 с.
9. Інформаційний ресурс. – Режим доступу : http://helpiks.org/6-26903.html/
10. Інформаційний ресурс. – Режим доступу : http://irttri.com/
11. Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу : НД ТЗІ 1.1-002-99 : [затверджений наказом Департаменту спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України від 28 квітня1999 р. № 22 із змінами згідно з наказом Адміністрації Держспецзв'язку від 28.12.2012 № 806]. – К. : ДСТСЗІ СБ України, 1999. – 21 с.
12. Основи інформаційної безпеки : навч. посіб. / В. А. Лужецький, О. П. Войнович, А. Д. Кожухівський, Л. І. Северин, І. Б. Трегубенко. – Черкаси, ЧДТУ, 2008. – 243 с.
13. Антонюк А. О. Основи захисту інформації в автоматизованих системах : навч. посіб. / А. О. Антонюк. – К. : Видавничий дім "КМ Академія", 2003. – 243 с.
14. Основы информационной безопасности : учебн. пособ. для вузов / Е. Б. Белов, В. П. Лось, Р. В. Мещеряков, А. А. Шелупанов. – М. : Горячая линия – Телеком, 2006. – 544 с. : ил.
15. Завгородний В. И. Комплексная защита информации в компьютерных системах : учеб. пособ. / В. И. Завгородний. – М. : Логос; ПБОЮЛ Н. А. Егоров, 2001. – 264 с. :ил.
16. Романец Ю. В. Защита информации в компьютерных системах и сетях / Ю. В. Романец, П. А. Тимофеев, В. Ф. Шаньгин; под ред. В. Ф. Шаньгина. – 2-е изд., перераб. и доп. – М. : Радио и связь, 2001. – 376 с. : ил.

Розділ 2
МАТЕМАТИЧНІ ОСНОВИ КРИПТОЛОГІЇ

Математичні структури є невід'ємною частиною сучасної ***криптології*** – науки, що містить ***криптографію і криптоаналіз***. Знання законів, що діють у цих структурах, необхідні для побудови стійких криптографічних систем. Для розуміння сучасних криптографічних систем необхідні знання із багатьох розділів математики, таких як абстрактна алгебра, теорія алгоритмів, теорія чисел, теорія інформації, теорія ймовірності, комбінаторики тощо. У цьому розділі буде викладено інформацію із окремих параграфів перших трьох дисциплін. Теорію інформації переважно викладають окремим курсом, а окремі розділи із теорії ймовірності та комбінаторики зазвичай викладають у курсі вищої математики.

2.1. Елементи теорії множин

Шифрування текстів по суті можна розглядати як способ відображення множини відкритих текстів у множину криптограм, який визначають доступною множиною ключів. Цей підхід дав змогу Шеннону зробити певні висновки [1, 2], на яких базується теорія сучасних шифрів і які успішно використовують у сучасних криптографічних системах [3]. У цьому підрозділі розглянемо деякі елементи теорії множин, без яких є неможливим розуміння багатьох положень сучасної криптографії.

2.1.1. Відображення

Нехай X та Y – дві множини. Припустимо, що кожному елементові x множини X відповідає деякий елемент $y = f(x)$ множини Y. В такому випадку задано відображення або функція $f : X \to Y$ із множини X у множину Y. Відображення $f : X \to Y$ називають ***ін'єктивним***, якщо воно різним аргументам зіставляє різні значення: $f(x_1) \neq f(x_2)$ для $x_1 \neq x_2$.

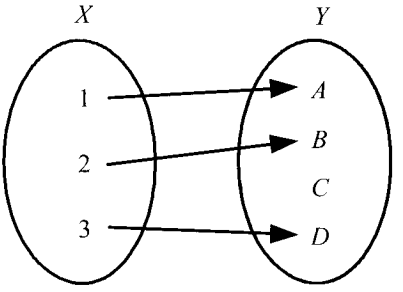


Рис. 2.1. Ін'єктивне відображення

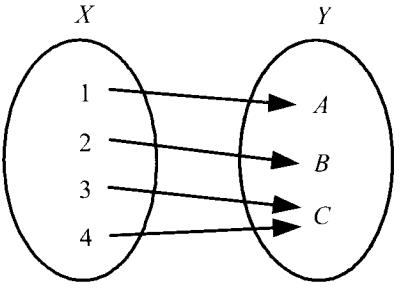


Рис. 2.2. Сюр'єктивне відображення

Відображення $f: X \to Y$ ***сюр'єктивне***, якщо кожен елемент y з множини Y має як праобраз такий елемент $x \in X$, що $f(x) = y$.

Бієктивним є відображення, яке ін'єктивне і сюр'єктивне одночасно.

На рис. 2.4 наведено приклад відображення, яке несюр'єктивне і неін'єктивне. На відміну від ін'єктивного (рис. 2.1) одному значенню множини X відповідає 2 значення множини Y (рис. 2.4), також на відміну від сюр'єктивного відображення (рис. 2.2) у множині Y є елементи, що не мають праобразу у множині X (рис. 2.4).

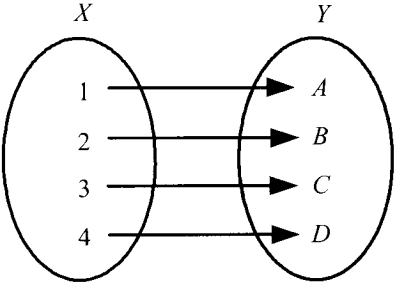


Рис. 2.3. Бієктивне відображення

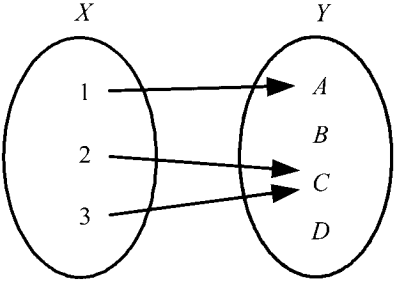


Рис. 2.4. Несюр'єктивне і неін'єктивне відображення

Для двох відображень $f: X \to Y$ і $g: Y \to Z$ їх композицію $g \circ f: X \to Z$ задають виразом $g \circ f(x): g(f(x))$ для $x \in X$, де знаком "$\circ$" позначено операцію композиції. Тотожне відображення $id_X: X \to X$ залишає елементи множини X на місці: $id_X(x) = x$. Відображення $g: Y \to X$ вважають ***лівим оберненим*** до відображення $f: X \to Y$ за умови, що їх композиція $g \circ f = id_X$;

і ***правим оберненим*** за умови, що $f \circ g = id_Y$. Відображення g називають ***оберненим*** до f, якщо воно є одночасно і лівим, і правим оберненим до f.

Властивості оберненого відображення.

1. Відображення $f: X \to Y$ ін'єктивне тоді й тільки тоді, коли до нього існує ліве обернене.

2. Відображення $f: X \to Y$ сюр'єктивне тоді і тільки тоді, коли до нього існує праве обернене.

3. Якщо відображення $f: X \to Y$ бієктивне, то його ліве обернене збігається із правим оберненим.

4. У випадку, коли множини X та Y скінченні й містять однакову кількість елементів, відображення $f: X \to Y$ має ліве обернене тоді й тільки тоді, коли воно має праве обернене.

2.1.2. Основні поняття і визначення

Напівгрупою називають множину елементів S, у якій визначено певну бінарну операцію $(*)$ із властивостями:

1. $\forall x \in S, \forall y \in S : x * y \in S$ (замкнутість).

2. $(x * y) * z = x * (y * z) : x, y, z \in S$ (асоціативність).

При тому можуть бути відсутніми для деяких елементів множини як обернений, так і нейтральний елементи. Напівгрупами є множина натуральних чисел з операцією множення "$*$", множина натуральних чисел з операцією додавання "$+$".

Моноїди. ***Моноїдом*** називають напівгрупу S з нейтральним елементом, тобто для елементів моноїда можна записати 3 діючі властивості:

1. $\forall x \in S, \forall y \in S : x * y \in S$ (замкнутість).

2. $(x * y) * z = x * (y * z) : x, y, z \in S$ (асоціативність).

3. У G існує нейтральний елемент e такий, що $x * e = e * x$ для всіх $x \in S$ (наявність нейтрального елемента).

Групи. ***Групою*** називають множину G, наділену бінарною операцією $*$ із такими властивостями:

1 $\forall x \in S, \forall y \in S : x * y \in G$ (замкнутість).

2. $(x * y) * z = x * (y * z) : x, y, z \in S$ (асоціативність).

3. У G існує нейтральний елемент e такий, що $x * e = e * x$ для всіх $x \in G$.

4. Для кожного елемента $x \in G$ в G існує обернений елемент $x^{'}$ такий, що $x * x^{'} = x^{'} * x = e$.

Якщо підмножина H множини G утворює групу відносно тієї самої операції $*$, то її називають підгрупою групи G. Так, множина раціональних чисел Q утворює групу за додаванням ($e = 0, x^{'} = -x$), а множина цілих чисел Z є її підгрупою. Множина додатних раціональних чисел Q_{+} утворює групу за множенням ($e = 1, x^{'} = 1/x$). Якщо груповою є операція множення, то групу називають мультиплікативною, а її нейтральний елемент – одиниця, якщо додавання, – то групу називають адитивною, нейтральний елемент – нуль, а обернений елемент – протилежний елемент.

Для елемента x групи G через x^{i} позначають його i-й степінь – елемент $x * x * \ldots * x$, де операцію виконано $i-1$ разів (в адитивній формі $ix = x + \cdots + x$). Для кожного елемента x скінченної групи для деякого показника m виконується рівність $x^{m} = e$. Найменше з таких m називають ***порядком елемента*** x у групі G.

Порядком скінченної групи називають кількість її елементів. Усі степені елемента групи утворюють у ній підгрупу, порядок якої дорівнює порядку елемента.

Нехай маємо дві групи G і $G^{'}$ з операціями $*$ і $\circ$ відповідно. Відображення $f : G \to G^{'}$ зберігає операцію, якщо $f(x * y) = f(x) \circ f(y)$ для всіх $x, y \in G$. Таке відображення називають ***гомоморфізмом*** з групи G у групу $G^{'}$. ***Ядром гомоморфізму*** $f : G \to G^{'}$ є множина всіх тих елементів групи G, які f відображає в нейтральний елемент групи $G^{'}$. Наприклад, відображення, яке кожному цілому числу ставить у відповідність його остачу від ділення на натуральне n, є гомоморфізмом із адитивної групи Z в адитивну групу Z_n, ядро якого утворюють цілі числа, кратні n. Ядро гомоморфізму $f : G \to G^{'}$ утворює в G підгрупу.

Гомоморфізм $f : G \to G^{'}$ є ін'єктивним тоді і тільки тоді, коли його ядро складається з нейтрального елемента групи G. Таке ядро називають тривіальним.

Гомоморфізм, який є бієктивним відображенням, називають ***ізоморфізмом***. Дві групи ізоморфні, якщо існує ізоморфізм з однієї з них на іншу. Наприклад, двійковий логарифм $\log_2 : R_{+} \to R$ задає ізоморфізм із мультиплікативної групи невід'ємних дійсних чисел R_{+} в адитивну групу всіх дійсних чисел R.

Для груп G_1 та G_2 з операціями $*$ і $\circ$ відповідно, через $G_1 \times G_2$ позначають їх прямий добуток – множину пар (x_1, x_2), де $x_1 \in G_1$, $x_2 \in G_2$, із покомпонентним виконанням операцій. Результатом виконання операцій з елементами (x_1, x_2) і (y_1, y_2) множини $G_1 \times G_2$ вважають елемент $(x_1 * y_1, x_2 \circ y_2)$.

Групу називають ***комутативною*** або ***абелевою***, якщо групова операція володіє властивістю комутативності: $x * y = y * x$ для будь-яких елементів x та y.

Якщо кожен елемент групи G є степенем її елемента g, то цей елемент називають твірним. Якщо група G має порядок n, то g є її твірним елементом тоді і тільки тоді, коли його порядок теж дорівнює n. Групу, яка має твірний елемент, називають циклічною.

Група перестановок. Іншим прикладом групи є множина Y бієктивних відображень множини X на себе з операцією композиції. Нейтральним елементом є тотожне відображення id_X, а оберненим елементом є відображення $f^{'}$, обернене до $f \in Y$. Бієкцію множини на себе називають ***перестановкою*** цієї множини. Якщо множина X налічує n елементів, то група Y ізоморфна групі $\{1,2,\ldots,n\}$. Останню називають симетричною групою степеня n і позначають через Y_n.

На власне багатократних перестановках елементів побудовано більшість сучасних симетричних блокових шифрів (див. розділ 3 "Криптологія").

Перестановку записують у вигляді таблиці

$$\begin{pmatrix} 1 & 2 & \ldots & n \\ \sigma(1) & \sigma(2) & \ldots & \sigma(n) \end{pmatrix}.$$

Цикл позначають $(i_1 i_2 \ldots i_l)$ де $i_1, i_2, \ldots, i_l$ – різні числа від 1 до n. Перестановку, яка елемент i_j, $j < l$ відображає в i_{j+1}, i_l в i_1, а всі інші елементи самі в себе, називають ***довжиною циклу***. Цикли $(i_1 i_2 \ldots i_l)$ та $(i_1 i_2 \ldots i_l)$ незалежні, якщо множини елементів $\{i_1, i_2, \ldots, i_l\}$ та $\{i_1^{'}, i_2^{'}, \ldots, i_l^{'}\}$ не перетинаються. Кожна перестановка є композицією (або добутком) попарно незалежних циклів. Наприклад, перестановка

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 5 & 2 & 6 & 3 & 1 & 4 \end{pmatrix}$$

дорівнює добутку $(15)\cdot(364)$.

Група Y_n, що складається із n елементів, має порядок $n!$

Кільця. ***Кільцем*** називають множину R із двома заданими на ній операціями + (додавання) та $\bullet$ (множення), яка має такі властивості:

1. Відносно додавання R утворює абелеву групу.
2. Операція множення асоціативна.
3. Множення дистрибутивне за додаванням, що означає виконання рівностей

$$(x+y)\bullet z = x\bullet z + y\bullet z,\ z\bullet(x+y) = z\bullet x + z\bullet y \text{ для всіх } x,y,z\in R.$$

Якщо, крім того, операція множення комутативна, то кільце називають ***комутативним***. R називають ***кільцем з одиницею***, якщо в ньому є нейтральний відносно множення елемент. Прикладом комутативного кільця з одиницею є множина Z цілих чисел зі звичайним додаванням та множенням. Прикладом некомутативного кільця з одиницею є кільце матриць.

Елемент кільця з одиницею називають ***оборотним справа [зліва]***, якщо $x\bullet x' = 1\left[x\bullet x' = 1\right]$ для деякого $x'\in R$. Елемент x' є ***правим [лівим] оберненим*** до x. Елемент x називають оборотним або дільником одиниці, якщо він оборотний і зліва, і справа. Кожен оборотний елемент має по одному лівому і правому оберненому елементу. Цей єдиний елемент називають оберненим до x і позначають x^{-1}. Для оборотних елементів x та y виконується рівність $(x\bullet y)^{-1} = x^{-1}\bullet y^{-1}$.

Відображення $f: R\to R'$ називають ***гомоморфізмом***, якщо воно зберігає операції додавання та множення. Для кілець з одиницею повинна виконуватися ще одна умова: f має відображати одиницю кільця R в одиницю кільця R'. ***Ядром гомоморфізму*** $f: R\to R'$ є множина всіх тих елементів кільця R, які f відображає в нуль кільця R'. Прикладом гомоморфізму є відображення із Z в Z_n, яке кожному цілому числу ставить у відповідність його остачу від ділення на натуральне n. Ядро утворюють цілі числа, кратні n. Як і у випадку груп, гомоморфізм $f: R\to R'$ є ін'єктивним тоді й тільки тоді, коли його ядро тривіальне, тобто складається із нуля кільця R.

Для кілець R_1 і R_2 через $R_1\otimes R_2$ позначають їх прямий добуток – множину пар (x_1,x_2), де $x_1\in R_1$, $x_2\in R_2$ із покомпонентним додаванням та множенням. Прямий добуток кілець є кільцем.

Кільце матриць. Позначимо через R кільце з одиницею. ***Матрицею*** розміру $k\times m$ в R називають індексовану сукупність (a_{ij}) елементів з R, де

$1 \le i \le k, 1 \le j \le m$. Елементи a_{ij} називають ***коефіцієнтами матриці***. Матрицю розміру $k \times 1$ називають вектором-стовпчиком, а розміру $1 \times k$ – вектором-рядком.

Сумою матриць $A = (a_{ij})$ та $B = (b_{ij})$ однакового розміру є матриця $C = (c_{ij})$ такого самого розміру з коефіцієнтами $c_{ij} = a_{ij} + b_{ij}$. Добутком матриці $A = (a_{is})$ розміру $k \times l$ на матрицю $B = (b_{sj})$ розміру $l \times m$ є матриця $C = (c_{ij})$ розміру $k \times m$ з коефіцієнтами $c_{ij} = \sum_{s=1}^{l} a_{is} b_{sj}$. Операція множення матриць є асоціативною.

Матрицю розміру $k \times k$ називають квадратною матрицею порядку k. Квадратні матриці заданого порядку k відносно операцій додавання та множення утворюють кільце, яке позначають $M_k(r)$. Цим кільцем з одиницею є одинична матриця I_k, діагональні коефіцієнти якої дорівнюють одиниці кільця R, а всі інші – нулю.

Кільця $M_k(M_l(R))$ і $M_{kl}(R)$ ізоморфні. Це означає, що матриці порядку kl можна розбити на блоки розміру l на l, після чого додавати й множити такі матриці поблоково.

Мультиплікативну групу оборотних матриць $M_k(R)^*$ називають повною лінійною групою і позначають також $GL_k(R)$.

Кільце R вважатимемо комутативним.

Визначник $\det A$ квадратної матриці $A = (a_{ij})$ порядку k дорівнює:

$$\det A = \sum_{\sigma} \varepsilon(\sigma) a_{1\sigma(1)} a_{2\sigma(2)} \cdots a_{k\sigma(k)},$$

де сумування виконують за всіма перестановками σ з Y_n, а $\varepsilon(\sigma)$ означає знак перестановки.

Визначник добутку матриць дорівнює добутку їх визначників $\det AB = \det A \det B$.

Якщо взяти в останній рівності замість B матрицю A', праву обернену до A, то отримаємо $\det A \det A' = \det I_k = 1$. Подібна рівність справедлива й для оберненої зліва матриці.

Алгебраїчним доповненням елемента a_{ij} матриці A називають значення $A_{ij} = (-1)^{i+j} M_{ij}$, де M_{ij} – визначник матриці, яку отримують із матриці A після викреслення її i-го рядка та j-го стовпчика.

Для матриці $A' = \left(a'_{ij}\right)$ з коефіцієнтами $a'_{ij} = A_{ji}$ виконується співвідношення $AA' = A'A = (\det A) I_k$. Отже, в матриці A^{-1}, оберненій до A, коефіцієнт з індексами ij дорівнює $A_{ij} (\det A)^{-1}$.

Поля. ***Полем*** називають множину із двома заданими на ній операціями + (додавання) та • (множення), яка має такі властивості:

1. Відносно додавання F утворює абелеву групу з нейтральним елементом 0.
2. Відносно множення F утворює абелеву групу з нейтральним елементом 1.
3. Множення дистрибутивне за додаванням.

Кільце многочленів. Многочлен (або поліном) степеня n від однієї змінної x над комутативним кільцем з одиницею R зображують у вигляді арифметичного виразу $a_n x^n + \cdots + a_2 x^2 + a_1 x + a_0$, де $a_n, \ldots, a_2, a_1, a_0 \in R$ – коефіцієнти многочлена, причому старший коефіцієнт відмінний від 0. Відносно стандартних операцій додавання та множення многочлени в R утворюють кільце, яке позначають $R[x]$. На операціях із многочленами побудовані різноманітні сучасні криптоалгоритми як симетричної, так і асиметричної криптографії (див. розділ 3).

У кільці $R[x]$ стандартно вводять операцію ділення з остачею. Завдяки цьому у кільці $R[x]$ аналогічно до кільця Z працює алгоритм Евкліда, який дає змогу знаходити найбільший спільний дільник двох многочленів. Простими числами є незвідні многочлени, тобто многочлени ненульового степеня, які не мають дільників меншого (але ненульового) степеня.

Нехай $p(x)$ – деякий многочлен у F. Для $b \in F$ позначимо через $p(b)$ елемент поля F, який отримують у результаті підстановки елемента b до многочлена замість змінної x і виконання операцій множення й додавання в F. Елемент $p(b)$ називають ***значенням многочлена*** $p(x)$ ***в точці*** b. Якщо $p(b) = 0$, то b називають ***коренем многочлена*** $p(x)$. Діленням з остачею многочлена $p(x)$ на многочлен $x - b$ і підстановкою $x = b$ до отриманої рівності доводимо теорему Безу: якщо b – корінь ненульового многочлена $p(x)$ в полі F, то $p(x)$ ділиться на многочлен $x - b$.

Векторні простори. Лінійним (або векторним) простором над полем F називають ***абелеву (комутативну) групу*** V із груповою операцією + та

операцією множення елемента групи (вектора) на елемент поля (скаляр), результатом виконання якої є вектор. Операції мають задовольняти такі умови:

1. Множення унітарне: $1v = v$, де 1 – одиниця поля, а v –довільний вектор з V.
2. Множення асоціативне: $a(bv) = (ab)v$ для всіх $a,b \in F$ і $v \in V$.
3. Множення і додавання, пов'язані законами дистрибутивності

$$a(v_1 + v_2) = av_1 + av_2,\ (a_1 + a_2)v = a_1v + a_2v$$

для всіх $a, a_1, a_2 \in F$ і $v, v_1, v_2 \in V$.

Наприклад, множина F^k для довільного натурального k із покомпонентним додаванням та покомпонентним множенням на елемент поля F є лінійним простором над F.

2.2. Елементи теорії чисел

На властивостях простих чисел, властивостях ділення чисел з остачею базується теорія порівнянь (п. 2.3), яка, своєю чергою, є підгрунтям для всієї несиметричної криптографії (розділ 3).

2.2.1. Ділення з остачею

Означення. Нехай $a,b, \in Z$. Число a ділиться на число b, якщо знайдеться таке число $q \in Z$, що $a = qb$. Синоніми: "a кратне b"; "b – дільник a". Запис: $a \vdots b$ чи $b \mid a$.

Відношення подільності $b \mid a$ є бінарним відношенням у множині Z. Справедлива така властивість:

Нехай $a_1 + a_2 + ... + a_n = c_1 + c_2 + ... + c_k$ – рівність сум цілих чисел. Якщо всі доданки в цій рівності, крім одного, кратні b, то і доданок, що залишився, має бути кратним b.

Теорема. Для певного цілого, відмінного від нуля, числа b довільне ціле число a єдиним чином представляється у вигляді $a = bq + r$, де $0 \le r$, $0 \le r < |b|$.

Доведення. Одне представлення числа a рівністю $a = bq + r$ одержимо, якщо вважатимемо, що bq дорівнює найбільшому кратному числу b, що не перевищує a (рис. 2.5).

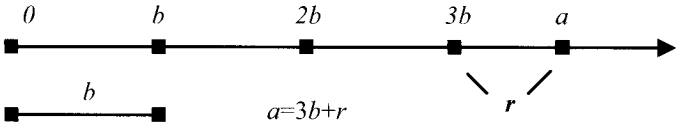


Рис. 2.5. Схема представлення числа рівністю $a=bq+r$

Тоді $0\le r<|b|$. Доведемо одиничність такого представлення. Нехай $a=bq+r$ і $a=bq_1+r_1$ – два такі представлення. Отже, $0=a-a=$ $=b(q-q_1)+(r-r_1)$. Тут 0 ділиться на b; $b(q-q_1)$ ділиться на b, отже, $(r-r_1)$ має ділитися на b. Оскільки $0\le r<b$ і $0\le r_1<b$, то $r-r_1<b$ і $r-r_1$ ділиться на b, тобто $r-r_1$ дорівнює нулю, тому $q-q_1$ дорівнює нулю, тобто два такі представлення збігаються.

Означення. Число q називають неповною часткою, а число r – залишком від ділення a на b.

Залишок завжди невід'ємне число, а неповна частка може бути яким завгодно цілим числом. Мінус п'ять поділити на три із залишком – мінус два, у залишку – один.

2.2.2. Найбільший спільний дільник і взаємно прості числа

Означення. Число $d\in Z$, на яке діляться одночасно числа $a,b,c,...,k\in Z$, називають спільним дільником цих чисел. Найбільше d із такою властивістю називають найбільшим спільним дільником. Позначення: $d=(a,b,c,...,k)$.

Розглянемо основні властивості найбільшого спільного дільника.

Теорема (Властивість 1). Якщо $(a,b)=d$, то знайдуться такі цілі числа u і v, що $d=au=bv$.

Властивість 2. Для будь-яких цілих чисел a і k справедливо:

$$(a,ka)=a;\quad (1,a)=1.$$

Властивість 3. Якщо $a=bq+c$, то сукупність спільних дільників a і b збігається із сукупністю спільних дільників b і c, зокрема,

$$(a,b)=(b,c).$$

Властивість 4. Нехай a, b і m – довільні цілі числа. Тоді

$$(am,bm)=m(a,b).$$

Властивість 5. Нехай s – дільник a і b. Тоді:

$$(a/s,b/s)=(a,b)/s.$$

Властивість 6. $(a/(a,b),b/(a,b))=1$.

Властивість 7. Якщо $(a,b)=1$, то $(ac,b)=(c,b)$.

2.2.3. Прості числа

Означення. Число $p\in N$, $p\neq 1$ називають простим, якщо p має лише два додатні дільники: 1 і p. Інші натуральні числа (крім 1) називають складеними. Число 1 за домовленістю ні просте, ні складене.

Відзначимо деякі спостереження, пов'язані з простими числами.

Спостереження 1. Найменший дільник будь-якого числа $a\in N$, відмінний від 1, є числом простим.

Доведення. Нехай $c|a$, $c\neq 1$ і c – найменше з цією властивістю. Якщо існує c_1 таке, що $c_1|c$, то $c_1\leq c$ і $c_1|a$, то $c_1=c$ або $c_1=1$.

Спостереження 2. Найменший відмінний від 1 дільник складеного числа $a\in N$ не перевищує $\sqrt{a}$.

Доведення. $c|a$, $c\neq 1$, c – найменше, отже,

$a=ca_1$, $a_1|a$, $a_1\geq c$, значить $aa_1\geq c^2a_1$, $a\geq c^2$ і $c\leq\sqrt{a}$.

Для складання таблиці простих чисел ***Ератосфен*** (Eratosthenes) придумав процедуру, яку називають "***решето Ератосфена***":

2, 3, ~~4~~, 5, ~~6~~, 7, ~~8~~, ~~9~~, ~~10~~, 11, ~~12~~, 13, ~~14~~, ~~15~~, ~~16~~, 17,...

Ідемо по натуральному ряду зліва направо. Підкреслюємо перше непідкреслене і невикреслене число, а з подальшого ряду викреслюємо кратні тільки що підкресленому. І так багато разів. Підкреслені числа – прості. Коли викреслено всі кратні до простих, менші за p, то ті, що залишилися невикресленими і менші за p^2, – прості. Це означає, що складання таблиці всіх простих чисел, менших за N, закінчено, тільки но викреслено всі кратні до простих, менших за $\sqrt{N}$.

Означення. Цілі числа a і b називають взаємно простими, якщо $(a,b)=1$. Іншими словами, два числа a і b є взаємно простими тоді й тільки тоді, коли знайдуться цілі числа u і v такі, що $au+bv=1$.

2.2.4. Алгоритм Евкліда

Слово "алгоритм" є транскрипцією латинізованого імені видатного арабського математика Аль-Хорезмі (Абу Абдулли Абу Джафар Мухаммад ібн Муса аль-Хорезмі (780–850)) і означає в сучасному розумінні деякі правила, список інструкцій чи команд, виконуючи які, можна досягти необхідного результату. Алгоритм, що дає змогу за заданими натуральними числами a і b знаходити їхній найбільший спільний дільник, називають алгоритмом ***Евкліда*** (Euclid).

Нехай дано два числа a і b; $a \geq 0$, $b \geq 0$, вважаємо, що $a > b$.

Алгоритм:

1. Ввести a і b.
2. Якщо $b = 0$, то відповідь: a. Кінець.
3. Замінити $r =$ "залишок від ділення a на b", $a := b$, $b = r$.
4. Іти на крок 2.

У сучасному буквеному записі алгоритм Евкліда виглядає так:

$a > b$; $a, b \in Z$.

$a = bq_1 + r_1$, $0 \leq r_1 < b$,

$b = r_1 q_2 + r_2$, $0 \leq r_2 < r_1$,

$r_1 = r_2 q_3 + r_3$, $0 \leq r_3 < r_2$,

$r_2 = r_3 q_4 + r_4$, $0 \leq r_4 < r_3$,

.....

$r_{n-3} = r_{n-2} q_{n-1} + r_{n-1}$, $0 \leq r_{n-1} < r_{n-2}$,

$r_{n-2} = r_{n-1} q_n + r_n$, $0 \leq r_n < r_{n-1}$,

$r_{n-1} = r_n q_{n+1}$, $r_{n+1} = 0$.

Маємо: $b > r_1 > r_2 > \ldots > r_n > 0$, отже, процес обірветься максимум через b кроків. Покажемо, що $r_n = (a, b)$. Переглянувши послідовно рівності згори донизу, бачимо, що довільний дільник a і b ділиться на $r_1, r_2, \ldots, r_n$. Якщо переглядати цей ланцюжок рівностей від останнього до першого, то видно, що $r_n \mid r_{n-1}$, $r_n \mid r_{n-2}$, і т.д. до $r_n \mid a$, $r_n \mid b$. Тому r_n – найбільший спільний дільник чисел a і b.

Сукупність дільників a і b збігається із сукупністю дільників (au, bv). Це дає практичний спосіб отримання чисел u і v з Z таких, що $r_n = au + bv = (a, b)$.

З ланцюжка рівностей маємо:

$r_n = r_{n-2} - r_{n-1} q_n = r_{n-2} - (r_{n-3} - r_{n-2} q_{n-1}) q_n = \ldots$

(ідемо ланцюжком рівностей знизу догори, виокремлюючи з кожної наступної рівності залишок і підставляючи його до виразу, що отриманий вже до цього моменту) $...=au+bv=(a,b)$.

Приклад. Нехай $a=525$, $b=231$.

$$
\begin{array}{rl}
 & \_525|\ \ 231 \\
 & \underline{462}\ \ \ |2 \\
\_231| & \underline{63} \\
\underline{189}\ \ |3 & \\
\_63\ \ \underline{42} & \\
\underline{42}\ \ |1 & \\
\_42\ \ \underline{21} & \\
\underline{42}\ \ |2 & \\
0 &
\end{array}
$$

Запис того самого у вигляді ланцюжка рівностей:

$525=231\cdot 2+63$, $231=63\cdot 3+42$, $63=42\cdot 1+21$, $42=21\cdot 2$.

Отже, $(525,231)=21$. Лінійне представлення найбільшого спільного дільника:

$$21=63-42\cdot 1=63-(231-63\cdot 3)\cdot 1=$$

$$=525-231\cdot 2-(525-231\cdot 2)\cdot 3)=525\cdot 4+231\cdot(-9)$$

і u та v з Z дорівнює, відповідно, 4 і - 9.

2.2.5. Лінійні діофантові рівняння з двома невідомими

Довільне рівняння (як правило, з цілими коефіцієнтами) називають ***діофантовим***, якщо його потрібно розв'язати в цілих числах.

Нехай потрібно розв'язати лінійне діофантове рівняння:

$$ax+by=c,$$

де $a,b,c\in Z$, a і b – не нулі.

У випадку, коли $c=1$, маємо варіант знаходження мультиплікативно оберненого значення для a за модулем b, що часто використовують у багатьох алгоритмах асиметричної криптографії.

Нехай $(a,b)=d$. Тоді $a=a_1d$; $b=b_1d$, і рівняння виглядає так:

$a_1d\cdot x+b_1d\cdot y=c$, тобто $d\cdot(a_1x+b_1y)=c$.

У такого рівняння є розв'язок (пари цілих чисел x і y) тільки тоді, коли $d\mid c$. Нехай $d\mid c$. Поділимо обидві частини рівняння на d і вважатимемо, що $(a,b)=1$.

Розглянемо кілька випадків.

Випадок 1. Нехай $c=0$, рівняння має вигляд $ax+by=0$ – "однорідне лінійне діофантове рівняння". Знаходимо, що

$x=-b/a\cdot y$.

Оскільки x має бути цілим числом, то $y=at$, де t – довільне ціле число (параметр). Отже, $x=-bt$, і розв'язками однорідного діофантового рівняння $ax+by=0$ є всі пари вигляду $\{-bt,at\}$, де $t=0;\pm1;\pm2;...$ Множину всіх таких пар називають ***загальним розв'язком лінійного однорідного діофантового рівняння***, будь-яку ж конкретну пару з цієї множини називають частковим розв'язком.

Випадок 2. Нехай тепер $c\neq0$. Для цього випадку існує теорема.

Теорема. Нехай $(a,b)=1$, $\{x_0,y_0\}$ – частковий розв'язок діофантового рівняння $ax+by=c$. Тоді його загальний розв'язок задають формулами:

$$\begin{cases}x=x_0-bt\\y=y_0+at.\end{cases}$$

Отже, загальний розв'язок неоднорідного рівняння є сумою загального розв'язку відповідного однорідного рівняння і якогось часткового розв'язку неоднорідного рівняння.

Доведення. Те, що праві частини зазначених у формулюванні теореми рівностей дійсно є розв'язками, перевіряють їх безпосередньою підстановкою у вихідне рівняння. Покажемо, що будь-який розв'язок рівняння $ax+by=c$ має вигляд, який зазначений у формулюванні теореми. Нехай $\{x^*,y^*\}$ – довільний розв'язок рівняння $ax+by=c$. Тоді $ax^*+by^*=c$, але і $ax_0+by_0=c$. Віднімемо від першої рівності другу й отримаємо:

$a(x^*-x_0)+b(y^*-y_0)=0$ – однорідне рівняння. Згідно із випадком 1, записуємо загальний розв'язок:

$x^*-x_0=-bt$, $y^*-y_0=at$, звідки отримуємо:

$$\begin{cases}x^*=x_0-bt\\y^*=y_0+at.\end{cases}$$

Частковий розв'язок $\{x_0,y_0\}$ знаходимо так. Оскільки $(a,b)=1$, то знайдуться такі u і v з Z, що $au+bv=1$, причому ці u і v отримують за допомогою алгоритму Евкліда. Помножимо тепер рівність $au+bv=1$ на c і отримаємо:

$a(uc)+b(vc)=c$, тобто $x_0=uc$, $y_0=vc$.

Приклад. Розв'язати рівняння:

$7x+12y=43$.

За алгоритмом Евкліда:

$12=7\cdot1+5$,

$7=5\cdot1+2$,

$5=2\cdot2+1$,

$2=1\cdot2$.

Отже, найбільший спільний дільник чисел 7 і 12 дорівнює 1, а його лінійний вираз такий:

$$1=5-2\cdot2=6-(7-5)\cdot=(12-7)-\big(7-(12-7)\cdot2\big)=12\cdot3+7\cdot(-5),$$

а саме $u=-5$, $v=3$. Частковий розв'язок:

$x_0=uc=(-5)\cdot43=-215$,

$y_0=vc=3\cdot43=129$.

Процедуру можна спростити, якщо записати загальний розв'язок неоднорідного діофантового рівняння:

$x=-215-12\cdot t$,

$y=129+7\cdot t$.

При $t=-18$, $x=1$, $y=3$.

2.2.6. Основна теорема арифметики

Теорема. Довільне ціле число, відмінне від -1, 0 і 1, єдиним чином (з точністю до порядку множників) розкладається за допомогою добутку простих чисел.

Доведення. Доводимо твердження теореми тільки для натуральних чисел, тому що знак мінус перед числом не впливає на суть теореми.

Нехай $a>1$, p_1 – його найменший простий дільник. Тоді, $a=p_1a_1$. Якщо далі $a_1>1$, то нехай p_2 – його найменший простий дільник і $a_1=p_2a_2$, тобто $a=p_1p_2a_2$, і так далі, поки a_n не дорівнюватиме одиниці. Це обов'язково відбудеться, тому що $a>a_1>a_2...,a_n$, а натуральні числа зі звичайним порядком задовольняють умову обриву спадних ланцюгів. Отже, $a=p_1p_2...p_n$, і можливість розкладу доведено.

Покажемо одиничність. Нехай $a=q_1q_2...q_s$ – інший розклад, тобто $p_1p_2...p_n=q_1q_2...q_s$. В останній рівності права частина ділиться на q_1, отже, ліва частина ділиться на q_1. Покажемо, що якщо добуток $p_1p_2...p_n$ ділиться на q_1, то один із множників p_k має ділитися на q_1.

Якщо $q_1 \mid p_1$, то все доведено. Нехай q_1 не ділиться на p_1. Оскільки q_1 – просте число, то $(q_1, p_1) = 1$. Тому знайдуться такі $u, v \in Z$, що $up_1 + vq_1 = 1$. Помножимо останню рівність на $p_2 \ldots p_n$, отримаємо: $p_2 \ldots p_n = p_1(p_2 \ldots p_n)u + .$ $+q_1(p_2 \ldots p_n)v$. Обидва доданки справа діляться на q_1, отже, $p_2 \ldots p_n$ ділиться на q_1.

Нехай, наприклад, $q_1 \mid p_1$. Отже, $q_1 = p_1$, тому що p_1 – просте. За рівністю $p_1 p_2 \ldots p_n = q_1 q_2 \ldots q_s$ скорочуючи, отримаємо рівність $p_2 \ldots p_n = q_2 \ldots q_s$. Оскільки $n = s$, кожен множник лівої частини рівності $p_1 p_2 \ldots p_n = q_1 q_2 \ldots q_n$ обов'язково присутній у правій і навпаки.

Наслідок 1. Довільне раціональне число можна однозначно подати у вигляді

$$p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_k^{\alpha_k},$$

де $\alpha_1, \alpha_2 \ldots, \alpha_k \in Z$.

Наслідок 2. Якщо $a = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_n^{\alpha_n}$, $b = p_1^{\beta_1} p_2^{\beta_2} \ldots p_n^{\beta_n}$ – цілі числа, то найбільший спільний дільник a і b дорівнює $p_1^{\gamma_1} p_2^{\gamma_2} \ldots p_n^{\gamma_n}$, а найменше спільне кратне a і b дорівнює $p_1^{\delta_1} p_2^{\delta_2} \ldots p_n^{\delta_n}$, де $\gamma_i = \min\{\alpha_i, \beta_i\}$, а $\delta_i = \max\{\alpha_i, \beta_i\}$.

2.3. Теорія порівнянь

Модульна арифметика надає багатьом функціям нової, дуже корисної для криптографії властивості, зокрема, вона суттєво ускладнює пошук прообразу певної функції (за відомим, значенням $Y(x)$ знайти x). Якщо для прикладу розглянути показникову функцію $Y = a^x$, то пошук прообразу цієї функції (пошук x за відомомим Y) спрощується, оскільки під час пробного розрахунку Y, підставивши якесь початкове значення x_0, ми отримаємо значення Y_0, яке ми вже можемо порівняти із відомим Y. Далі ми вже маємо напрямок пошуку, і за допомогою одного із відомих методів наближених розрахуків ми достатньо швидко можемо отримати шукане значення x. Зовсім іншу ситуацію отримаємо у випадку, якщо обчислюємо залишок від ділення результату експоненційної функції на ціле число p ($Y = a^x \bmod(p)$). У цьому випадку, діючи аналогічно, ми вже не знатимемо напрям подальшого розрахунку для отримання ефективної стратегії пошуку значення x. Тобто, отриману нами функцію $a^x \bmod(p)$ є

можливим достатньо швидко обчислити попри те, що її прообраз отримати значно складніше. Різниця в складності цих алгоритмів і дає криптографам можливість побудувати достатньо стійкі алгоритми напрямленого шифрування.

2.3.1. Означення й найпростіші властивості

Означення. Нехай $a,b \in Z$, $m \in N$. Число a порівняльне з b за модулем m, якщо a і b при діленні на m мають однакові залишки: $a \equiv b (\mathrm{mod}\, m)$.

Бінарне відношення порівняння $\equiv_m$ є відношенням еквівалентності в множині цілих чисел. Число a порівняльне з b за модулем m тоді й тільки тоді, коли $a-b$ ділиться на m без остачі. Це буває тоді й тільки тоді, коли знайдеться таке ціле число t, що $a = b + mt$. Порівняльність a і b за модулем m означає, що a і b є тим самим елементом у кільці лишків Z_m.

Класи порівняльних між собою за модулем m ілюструє рис. 2.6.

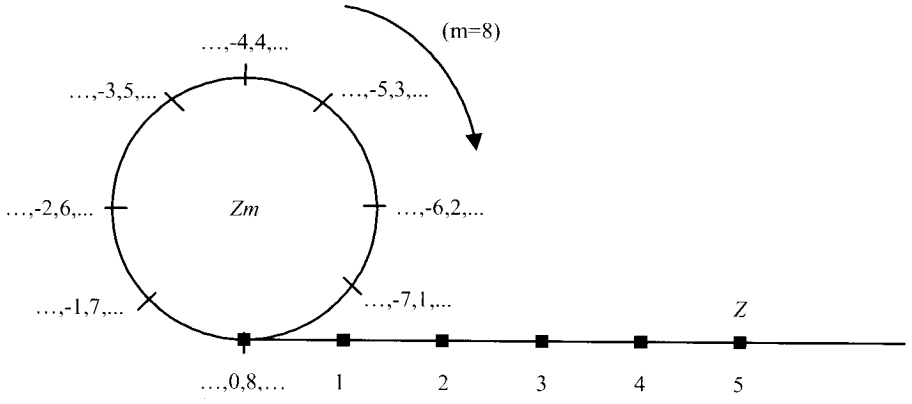


Рис. 2.6. Схема відповідності класів порівняльних між собою за модулем m чисел числовій прямій

На рис. 2.6 зображений процес "намотування" ланцюжка цілих чисел на кільце з m поділками, причому на одну поділку потрапляють порівняльні між собою числа.

Розглянемо властивості порівнянь.

Властивість 1. Порівняння за однаковим модулем можна почленно додавати.

Доведення. Нехай $a_1 \equiv b_1 (\mathrm{mod}\, m)$, $a_2 \equiv b_2 (\mathrm{mod}\, m)$. Це означає, що $a_1 \equiv b_1 + mt_1$, $a_2 \equiv b_2 + mt_2$. Після додавання останніх двох рівностей отримаємо $a_1 + a_2 \equiv b_1 + b_2 + m(t_1 + t_2)$, що означає $a_1 + a_2 \equiv b_1 + b_2 (\mathrm{mod}\, m)$.

Властивість 2. Доданок, що знаходиться в довільній частині порівняння, можна переносити в іншу частину, змінивши його знак на протилежний.

Доведення.
$$\begin{cases} a+b \equiv c(\bmod m) \\ -b = -b(\bmod m) \end{cases} +$$
$$a \equiv c-b(\bmod m).$$

Властивість 3. До будь-якої частини порівняння можна додати будь-яке число, кратне модулю.

Доведення.
$$\begin{cases} a \equiv b(\bmod m) \\ mk = 0(\bmod m) \end{cases} +$$
$$a+mk \equiv b(\bmod m).$$

Властивість 4. Порівняння за однаковим модулем можна почленно перемножувати, і обидві частини порівняння можна піднести до однакового степеня.

Доведення.
$$\begin{cases} a_1 \equiv b_1(\bmod m) \Leftrightarrow a_1 = b_1 + mt_1 \\ a_2 \equiv b_2(\bmod m) \Leftrightarrow a_2 = b_2 + mt_2 \end{cases} \times$$
$$a_1a_2 \equiv b_1b_2 + m(b_1t_2 + b_2t_1 + mt_1t_2) \Rightarrow a_1a_2 \equiv b_1b_2.$$

Наслідок перелічених властивостей розглянуто нижче.

Властивість 5. Якщо $a_0 \equiv b_0(\bmod m)$, $a_1 \equiv b_1(\bmod m)$, ... , $a_n \equiv b_n(\bmod m)$, $x \equiv y(\bmod m)$, то $a_0x^n + a_1x^{n-1} + ... + a_n \equiv b_0y^n + b_1y^{n-1} + ... + b_n(\bmod m)$.

Властивість 6. Обидві частини порівняння можна поділити на їхній спільний дільник, взаємно простий з модулем.

Доведення. Нехай $a \equiv b(\bmod m)$, $a = a_1d$, $b = b_1d$. Тоді $(a_1 - b_1)\cdot d$ ділиться на m. Оскільки d і m взаємно прості, то на m ділиться число $(a_1 - b_1)$, що означає $a_1 \equiv b_1(\bmod m)$.

Властивість 7. Обидві частини порівняння та його модуль можна помножити на те саме ціле число або розділити на їхній спільний дільник.

Доведення.

$a \equiv b(\bmod m) \Rightarrow a = b + mt \Rightarrow ak = bk + mkt \Rightarrow ak = bk + (\bmod mk)$.

Властивість 8. Якщо порівняння $a \equiv b$ існує за кількома різними модулями, то воно існує й за модулем, що дорівнює найменшому спільному кратному цих модулів.

Доведення. Якщо $a \equiv b(\bmod m_1)$ і $a \equiv b(\bmod m_2)$, то $a - b$ ділиться на m_1 і на m_2, отже, $a - b$ ділиться на найменше спільне кратне m_1 і m_2.

Властивість 9. Якщо порівняння існує за модулем m, то воно існує й за модулем d, що дорівнює будь-якому дільнику числа m.

Доведення. Якщо $a \equiv b(\mathrm{mod}\, m)$, то $a-b$ ділиться на m, отже, $a-b$ ділиться на d, де $d \mid m$.

Властивість 10. Якщо одна частина порівняння і модуль діляться на деяке число, то й інша частина порівняння має ділитися на те саме число.

Доведення. $a \equiv b(\mathrm{mod}\, m) \Rightarrow a = b + mt$.

Приклад. Довести, що для будь-якого натурального n число $37^{n+2} + 16^{n+1} + 23^n$ ділиться на 7.

Запишемо очевидні порівняння: $37 \equiv 2(\mathrm{mod}\, 7)$, $16 \equiv 2(\mathrm{mod}\, 7)$, $23 \equiv 2(\mathrm{mod}\, 7)$.

Піднесемо перше порівняння до степеня $n+2$, друге – до степеня $n+1$, третє – до степеня n і додамо:

$$\begin{cases} 37^{n+2} \equiv 2^{n+2}(\mathrm{mod}\, 7) + \\ 16^{n+1} \equiv 2^{n+1}(\mathrm{mod}\, 7) + \\ 23^{n} \equiv 2^{n}(\mathrm{mod}\, 7) \end{cases}$$

$$37^{n+2} + 16^{n+1} + 23^n \equiv 2^n \cdot 7(\mathrm{mod}\, 7),$$

тобто $37^{n+2} + 16^{n+1} + 23^n$ ділиться на 7.

2.3.2. Повна та зведена системи лишків

Відношення еквівалентності $\equiv_m$ дає змогу розбити множини цілих чисел на класи еквівалентних між собою елементів, тобто в один клас об'єднують числа, що мають у разі ділення на m однакові залишки.

Означення. Будь-яке число з класу еквівалентності $\equiv_m$ називають лишком за модулем m. Сукупність лишків, узятих по одному з кожного класу еквівалентності $\equiv_m$, називають повною системою лишків за модулем m (у повній системі лишків m чисел). Безпосередньо самі залишки ділення на m називають найменшими невід'ємними лишками. Вони утворюють ***повну систему лишків за модулем m***. Лишок ρ називають абсолютно найменшим, якщо $|\rho|$ найменший серед модулів лишків цього класу.

Приклад. Нехай $m = 5$. Тоді:

0, 1, 2, 3, 4 – найменші невід'ємні лишки;

-2, -1, 0, 1, 2 – абсолютно найменші лишки.

Обидві наведені множини чисел утворять повні системи лишків за модулем 5.

Лема 1. 1) будь-які m попарно непорівняльних за модулем m чисел утворюють повну систему лишків за модулем m.

2) якщо a і m взаємно прості, а x набуває значень з повної системи лишків за модулем m, то значення лінійної форми $ax+b$, де b – будь-яке ціле число, також набувають значень з повної системи лишків за модулем m.

Доведення. Твердження (1) випливає з означення. Доведемо твердження (2). Чисел $ax+b$ є m одиниць. Покажемо, що вони між собою непорівняльні за модулем m. Нехай для деяких різних x_1 і x_2 з повної системи лишків виявилося, що $ax_1+b=ax_2+b(\mathrm{mod}\, m)$. Тоді, за властивостями порівнянь, отримуємо:

$ax_1 \equiv ax_2(\mathrm{mod}\, m)$, $x_1 \equiv x_2(\mathrm{mod}\, m)$

– суперечить тому, що x_1 і x_2 різні та взяті з повної системи лишків.

Оскільки всі числа з цього класу еквівалентності отримують з одного числа цього класу додаванням числа, кратного m, то всі числа цього класу мають з модулем m однаковий найбільший спільний дільник.

Означення. ***Зведеною системою лишків за модулем m*** називають сукупність усіх лишків з повної системи, ***взаємно простих з модулем m.***

Зведену систему вибирають з найменших невід'ємних лишків. Зведена система лишків за модулем m містить $\varphi(m)$ лишків, де $\varphi(m)$ – функція Ейлера – кількість чисел, менших за m і взаємно простих з m.

Приклад. Нехай $m=42$. Тоді зведеною системою лишків є:

1, 5, 11, 13, 17, 19, 23, 25, 29, 31, 37, 41.

Лема 2. 1) будь-які $\varphi(m)$ числа, попарно непорівняльні за модулем m і взаємно прості з модулем, утворюють зведену систему лишків за модулем m.

2) якщо $(a,m)=1$ і x набуває значень із зведеної системи лишків за модулем m, то ax так само набуває значень із зведеної системи лишків за модулем m.

Доведення. Твердження (1) випливає з означення. Доведемо твердження (2). Числа ax попарно непорівняльні (доводимо так само, як лема 1), їх є $\varphi(m)$. Усі вони взаємно прості з модулем, оскільки $(a,m)=1$, $(x,m)=1 \Rightarrow (ax,m)=1$. Отже, числа ax утворять зведену систему лишків.

Лема 3. Нехай m_1, m_2, ..., m_k – попарно взаємно прості й $m_1+m_2+...m_k = M_1m_1 = M_2m_2 = ... = M_km_k$, де $M_j = m_1...m_{j-1}m_{j+1}...m_k$.

1) якщо $x_1, x_2, ..., x_k$ набувають значень з повних систем лишків за модулями $m_1, m_2, ..., m_k$ відповідно, то значення лінійної форми $M_1x_1 + M_2x_2 + ... + M_kx_k$ набувають значень з повної системи лишків за модулем $m = m_1m_2 ... m_k$.

2) якщо $\xi_1, \xi_2, ..., \xi_k$ набувають значень із зведених систем лишків за модулями $m_1, m_2, ..., m_k$ відповідно, то значення лінійної форми $M_1\xi_1 + M_2\xi_2 + ... + M_k\xi_k$ набувають значень із зведеної системи лишків за модулем $m = m_1 m_2 ... m_k$.

Доведення.

1) форма $M_1x_1 + M_2x_2 + ... + M_kx_k$ набуває $m_1 m_2 ... m_k = m$ значень. Покажемо, що ці значення попарно непорівнянні. Нехай

$$M_1x_1 + M_2x_2 + ... + M_kx_k \equiv M_1x_1 + M_2x_2 + ... + M_kx_k \pmod{m}.$$

Довільне M_j, відмінне від M_s є кратним до m_s. Забираючи зліва і справа в останньому порівнянні доданки, кратні до m_s, отримаємо:

$$M_sx_s \equiv M_sx_s \pmod{m_s} \Rightarrow x_s \equiv x_s \pmod{m_s}$$

– суперечить тому, що x_s набуває значень з повної системи лишків за модулем m_s.

2) форма $M_1\xi_1 + M_2\xi_2 + ... + M_k\xi_k$ набуває $\varphi(m_1)\varphi(m_2)...\varphi(m_k) = \varphi(m_1m_2 ... m_k) = \varphi(m)$ різних значень, які між собою за модулем $m = m_1m_2 ... m_k$ попарно непорівнялнні. Останнє доводимо аналогічно з доведенням твердження (1) цієї леми. Оскільки $(M_1\xi_1 + M_2\xi_2 + ... + M_k\xi_k, m_s) = (M_s\xi_s, m_s) = 1$ для кожного $1 \le s \le k$, то $(M_1\xi_1 + M_2\xi_2 + ... + M_k\xi_k, m_s) = 1$, отже, множина значень форми $M_1\xi_1 + M_2\xi_2 + ... + M_k\xi_k$ утворить зведену систему лишків за модулем m.

2.3.3. Теорема Ейлера

Нехай $m > 1$, $(a, m) = 1$, $\varphi(m)$ – функція Ейлера. Тоді:

$$a^{\varphi(m)} \equiv 1 \pmod{m}. \qquad (2.1)$$

Доведення. Нехай x набуває значення із зведеної системи лишків за $\bmod m$:

$$x = r_1, r_2, ..., r_c,$$

де $c = \varphi(m)$ їх кількість, $r_1, r_2, ..., r_c$ – найменші невід'ємні лишки за $\bmod m$. Отже, найменші невід'ємні лишки, що відповідають числам ax, є, відповідно:

$\rho_1, \rho_2, ..., \rho_c$ – також набувають значень із зведеної системи лишків, але в іншій послідовності. Тобто:

$a \cdot r_1 \equiv \rho_{\varphi 1} (\mathrm{mod}\, m)$,

$a \cdot r_2 \equiv \rho_{\varphi 2} (\mathrm{mod}\, m)$,

...

$a \cdot r_c \equiv \rho_{\varphi c} (\mathrm{mod}\, m)$.

Перемножимо ці c порівнянь. Отримаємо:

$a^c r_1 r_2 ... r_c \equiv \rho_{j1} \rho_{j2} ... \rho_{jc} (\mathrm{mod}\, m)$.

Оскільки $r_1 r_2 ... r_c = \rho_1 \rho_2 ... \rho_c \neq 0$ і взаємно просте з модулем m, то, поділивши останнє порівняння на $r_1 r_2 ... r_c$, отримаємо $a^{\varphi(m)} \equiv 1 (\mathrm{mod}\, m)$.

2.3.4. Мала теорема Ферма

Теорема. Нехай p – просте число, a не ділиться на p. Тоді:

$$a^{p-1} \equiv 1 (\mathrm{mod} p). \qquad (2.2)$$

Доведення 1. Нехай в умові теореми Ейлера $m = p$, тоді $\varphi(m) = p - 1$. Отримуємо $a^{p-1} \equiv 1 (\mathrm{mod} p)$.

У формулюваннях теорем Ейлера та Ферма умова взаємної простоти модуля i числа a є обов'язковою. Проте можна підправити формулювання теореми Ферма, щоб зняти обмеження взаємної простоти.

Наслідок 1. Без жодних обмежень на $a \in Z$,

$$a^p \equiv a (\mathrm{mod}\, p). \qquad (2.3)$$

Доведення. Помножимо обидві частини порівняння $a^{p-1} \equiv 1 (\mathrm{mod}\, p)$ на a. Отримаємо порівняння, яке справедливе при a, кратному p.

Доведення 2. Оскільки p – просте число, то всі біноміальні коефіцієнти (крім C_0^p і C_p^p) діляться на p, тому що чисельник виписаного виразу містить p, а знаменник не містить цього множника. За біномом Ньютона

$$C_n^m = \frac{n!}{m!(n-m)!}$$

$$(a+b)^n = C_n^0 a^n + C_n^1 a^{n-1} b + ... + C_n^m a^{n-m} b^m + ... + C_n^{n-1} a b^{n-1} + C_n^n b^n, (C_n^0 = C_n^n = 1).$$

Різниця

$$(A+B)^p - A^p - B^p = C_p^1 A^{p-1} B^1 + C_p^2 A^{p-2} B^2 + ... + C_p^{p-2} A^2 B^{p-2} + C_p^{p-1} A^1 B^{p-1},$$

де A і B – довільні цілі числа, завжди ділиться на p. Послідовним застосуванням цього спостереження отримуємо, що $(A+B+C)^p - A^p - B^p - C^p = \{[(A+B)+C]^p - (A+B)^p - C^p\} + (A+B)^p - A^p - B^p$ завжди ділиться на p; $(A+B+C+D)^p - A^p - B^p - C^p - D^p$ завжди ділиться на p; і взагалі, $(A+B+C+...+K)^p - A^p - B^p - C^p - ... - K^p$ завжди ділиться на p. Нехай $A=B=C=...=K=1$, а кількість цих чисел дорівнює a. Отримаємо, що $a^p - a$ ділиться на p, а це і є теорема Ферма в загальнішому формулюванні.

Наслідок 2. $(a=b)^p \equiv a^p + b^p (\mathrm{mod}\, p)$.

Приклади застосування теорем Ферма і Ейлера.

Приклад 1. Дев'ятий ступінь однозначного числа закінчується на 7. Знайти це число.

$a^9 \equiv 7(\mathrm{mod}10)$ – дано. Крім того, $(7,10)=1$ і $(a,10)=1$. За теоремою Ейлера, $a^{\varphi(10)} \equiv 1(\mathrm{mod}10)$. Отже, $a^4 \equiv 1(\mathrm{mod}10)$ і, після піднесення до квадрата, $a^8 \equiv 1(\mathrm{mod}10)$. Поділимо почленно $a^9 \equiv 7(\mathrm{mod}10)$ на $a^8 \equiv 1(\mathrm{mod}10)$ і отримаємо $a \equiv 7(\mathrm{mod}10)$. Це означає, що $a=7$.

Приклад 2. Довести, що $1^{18} + 2^{18} + 3^{18} + 4^{18} + 5^{18} + 6^{18} \equiv -1(\mathrm{mod}7)$.

Доведення. Числа 1, 2, 3, 4, 5, 6 взаємно прості з 7. За теоремою Ферма маємо:

$$\begin{cases} 1^6 \equiv 1(\mathrm{mod}7) \\ 2^6 \equiv 1(\mathrm{mod}7) \\ ... \\ 6^6 \equiv 1(\mathrm{mod}7) \end{cases}.$$

Піднесемо ці порівняння до куба і додамо:

$1^{18} + 2^{18} + 3^{18} + 4^{18} + 5^{18} + 6^{18} \equiv 6(\mathrm{mod}7) \equiv -1(\mathrm{mod}7)$.

Приклад 3. Знайти залишок ділення 7^{402} на 101.

Число 101 – просте, $(7,101)=1$, отже, за теоремою Ферма: $7^{100} \equiv 1(\mathrm{mod}101)$. Піднесемо це порівняння до четвертого степеня: $7^{400} \equiv 1(\mathrm{mod}101)$, домножимо його на очевидне порівняння $7^2 \equiv 49(\mathrm{mod}101)$, отримаємо: $7^{402} \equiv 49(\mathrm{mod}101)$. Тобто, залишок ділення 7^{402} на 101 дорівнює 49.

Приклад 4. Довести, що $(7^{32}-1)$ ділиться на 105.

Маємо: $105=3\cdot5\cdot7$, $(73,3)=(73,5)=(73,7)=1$.

За теоремою Ферма:

$73^2\equiv1(\mathrm{mod}\,3)$,

$73^4\equiv1(\mathrm{mod}\,5)$,

$73^6\equiv1(\mathrm{mod}\,7)$.

Перемножуючи, отримуємо:

$73^{12}\equiv1(\mathrm{mod}\,3),(\mathrm{mod}\,5),(\mathrm{mod}\,7)$,

звідки, за властивостями порівнянь, випливає:

$73^{12}-1\equiv0(\mathrm{mod}\,105)$.

2.3.5. Порівняння першого степеня

Розв'язуємо порівняння з одним невідомим вигляду:

$$f(x)\equiv0(\mathrm{mod}\,m),$$

де $f(x)=a_0x^n+a_1x^{n-1}+...+a_{n-1}x+a_n$ – многочлен з цілими коефіцієнтами. Якщо a_0 не ділиться на m, то n – ступінь порівняння. Якщо яке-небудь число x задовольняє порівняння, то це саме порівняння задовольняє і будь-яке інше число, порівнянне з x за $\mathrm{mod}\,m$.

Розв'язати порівняння – означає знайти всі x, які задовольняють це порівняння, при цьому весь клас чисел за $\mathrm{mod}\,m$ вважають одним розв'язком.

Отже, кількість розв'язків порівняння є кількістю лишків з повної системи, які це порівняння задовольняють.

Приклад. Для порівняння: $x^5+x+1\equiv0(\mathrm{mod}\,7)$

серед 0, 1, 2, 3, 4, 5, 6 це порівняння задовольняють два: $x_1=2$, $x_2=4$. Це означає, що у цьому порівняння два розв'язки:

$$x\equiv2(\mathrm{mod}\,7)\ \text{ і }\ x\equiv4(\mathrm{mod}\,7).$$

Порівняння називають рівносильними, якщо вони мають однакові розв'язки – аналогія з поняттям рівносильності рівнянь. Проте, на відміну від алгебраїчних рівнянь, які часто не розв'язуються радикалами, порівняння будь-якого ступеня завжди розв'язується, наприклад, перебором усіх лишків за $\mathrm{mod}\,m$.

Розглянемо порівняння першого степеня вигляду $ax\equiv b(\mathrm{mod}\,m)$ (два випадки).

Випадок 1. Нехай a і m взаємно прості. Тоді нескоротний дріб m/a розкладається в ланцюговий дріб:

$$\frac{m}{a} = q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 + \cfrac{1}{\ldots + \cfrac{1}{q_{n-1} + \cfrac{1}{q_n}}}}}.$$

Цей ланцюговий дріб скінченний, тому що m/a – раціональне число. Розглянемо два останні прямуючі дроби:

$$\delta_{n-1} = \frac{P_{n-1}}{Q_{n-1}};\ \delta_n = \frac{P_n}{Q_n} = \frac{m}{a}.$$

За властивостями чисельників і знаменників прямуючих дробів: $mQ_{n-1} - aP_{n-1} = (-1)^n$. Доданок mQ_{n-1}, кратний до m, можна відкинути з лівої частини порівняння:

$-aP_{n-1} \equiv (-1)^n (\mathrm{mod}\, m)$, тобто

$aP_{n-1} \equiv (-1)^{n-1} (\mathrm{mod}\, m)$, тобто

$a[(-1)^{n-1} P_{n-1} b] \equiv b (\mathrm{mod}\, m)$,

і єдиним розв'язком початкового порівняння є:

$x \equiv (-1)^{n-1} P_{n-1} b (\mathrm{mod}\, m)$.

Приклад. Розв'язати порівняння $111x \equiv 75 (\mathrm{mod}\, 332)$.

$(111, 332) = 1$. За алгоритмом Евкліда:

$322 = 111 \cdot 2 + 100$,

$111 = 100 \cdot 1 + 11$,

$100 = 11 \cdot 9 + 1$,

$11 = 1 \cdot 11$.

Отже, $n = 4$, а відповідний ланцюговий дріб:

$$\frac{m}{a} = \frac{322}{111} = 2 + \cfrac{1}{1 + \cfrac{1}{9 + \cfrac{1}{11}}}.$$

Обчислимо чисельники прямуючих дробів, склавши для цього таку таблицю (табл. 2.1):

Таблиця 2.1

Таблиця для знаходження чисельників прямуючих дробів

	0	2	1	9	11
P_n	1	2	3	29	322

Чисельник передостаннього прямуючого дробу дорівнює 29, отже:

$x \equiv (-1)^3 \cdot 29 \cdot 75 \equiv -2175 \equiv 79 (\mathrm{mod}\, 322)$.

Інакше кажучи, алгоритм розв'язання порівняння $ax \equiv b(\mathrm{mod}\, m)$, де a і m взаємно прості, такий. Візьмемо алгоритм Евкліда і знайдемо $u, v \in Z$ такі, що $au + vm = 1$. Помножимо цю рівність на b: $aub + vmb = b$, звідки випливає: $aub \equiv b(\mathrm{mod}\, m)$. Отже, розв'язком вихідного порівняння є $x \equiv ub(\mathrm{mod}\, m)$.

Випадок 2. Нехай $(a, m) = d$. Тоді для можливості розв'язування порівняння $ax \equiv b(\mathrm{mod}\, m)$ необхідно, щоб b ділилося на d, інакше порівняння узагалі виконуватися не може. Дійсно, $ax \equiv b(\mathrm{mod}\, m)$ буває тоді і тільки тоді, коли $ax - b$ ділиться на m без остачі, тобто $ax - b = t \cdot m$, $t \in Z$, звідки $b = ax - t \cdot m$, а права частина останньої рівності кратна до d.

Нехай $b = db_1$, $a = da_1$, $m = dm_1$. Тоді обидві частини порівняння $xa_1 d \equiv b_1 d(\mathrm{mod}\, m_1 d)$ і його модуль поділимо на d:

$xa_1 \equiv b_1 (\mathrm{mod}\, m_1)$,

де вже a_1 і m_1 взаємно прості. Згідно з випадком 1 таке порівняння має єдиний розв'язок x_0:

$$x \equiv x_0 (\mathrm{mod}\, m_1). \qquad (2.4)$$

За вихідним модулем m, числа (2.4) утворять стільки розв'язків вихідного порівняння, скільки чисел вигляду (2.4) міститься в повній системі лишків: $0, 1, 2, ..., m-2, m-1$. З чисел $x \equiv x_0 + t \cdot m$ у повну систему найменших невід'ємних лишків потрапляють лише x_0, $x_0 + m_1$, $x_0 + 2m_1$, ..., $x_0 + (d-1)m_1$, тобто лише d чисел. Отже, у початкового порівняння є d розв'язків.

Теорема 1. Нехай $(a, m) = d$. Якщо b не ділиться на d, то порівняння $ax \equiv b(\mathrm{mod}\, m)$ не має розв'язків. Якщо b кратне d, то порівняння $ax \equiv b(\mathrm{mod}\, m)$ має d розв'язків.

Приклад. Розв'язати порівняння $111x \equiv 75(\mathrm{mod}\, 321)$.

$(111, 321) = 3$, тому поділимо порівняння і його модуль на 3:

$37x \equiv 25(\mathrm{mod}\, 107)$ і $(37, 107) = 1$.

За алгоритмом Евкліда:

$107 = 37 \cdot 2 + 33$,

$37 = 33 \cdot 1 + 4$,

$33 = 4 \cdot 8 + 1$,

$4 = 1 \cdot 4$.

Маємо $n = 4$, і ланцюговий дріб такий:

$$\frac{m}{a} = \frac{107}{37} = 2 + \cfrac{1}{1 + \cfrac{1}{8 + \cfrac{1}{4}}}.$$

Складаємо таблицю для знаходження чисельників прямуючих дробів (табл. 2.2).

Таблиця 2.2

Таблиця для знаходження чисельників прямуючих дробів

q_n	0	2	1	8	4
P_n	1	2	3	26	107

Отримуємо, що

$x \equiv (-1)^3 \cdot 26 \cdot 25 \equiv -650 (\mathrm{mod}\, 107) \equiv -8 (\mathrm{mod}\, 107) \equiv 99 (\mathrm{mod}\, 107)$.

Отже, у початкового порівняння є три розв'язки:

$x \equiv 99 (\mathrm{mod}\, 321)$, $x \equiv 206 (\mathrm{mod}\, 321)$, $x \equiv 313 (\mathrm{mod}\, 321)$.

Теорема 2. Нехай $m > 1$, $(a, m) = 1$. Тоді порівняння $ax \equiv b (\mathrm{mod}\, m)$ має розв'язок:

$x \equiv b a^{\varphi(m)-1} (\mathrm{mod}\, m)$.

Доведення. За теоремою Ейлера маємо: $a^{\varphi(m)} \equiv 1 (\mathrm{mod}\, m)$, отже, $aba^{\varphi(m)-1} \equiv b (\mathrm{mod}\, m)$.

Приклад. Розв'язати порівняння $7x \equiv 3 (\mathrm{mod}\, 10)$. Обчислюємо:

$\varphi(10) = 4$; $x \equiv 3 \cdot 7^{4-1} (\mathrm{mod}\, 10) \equiv 1029 (\mathrm{mod}\, 10) \equiv 9 (\mathrm{mod}\, 10)$.

Недоліком цього способу розв'язування порівнянь є можливе піднесення числа a a до великого степеня.

2.3.6. Китайська теорема про лишки

Теорема. Нехай дано систему порівнянь першого степеня:

$$\begin{cases} x \equiv b_1 (\mathrm{mod}\, m_1) \\ x \equiv b_2 (\mathrm{mod}\, m_2) \\ \ldots \\ x \equiv b_n (\mathrm{mod}\, m_n), \end{cases} \qquad (2.5)$$

де $m_1, m_2, ..., m_k$ попарно взаємно прості. Нехай $m_1 m_2 ... m_k = M_s m_s$; $M_s \overset{`}{M_s} \equiv 1 (\mathrm{mod}\, m_s)$. Таке число $\overset{`}{M_s}$ завжди можна підібрати за допомогою алгоритму Евкліда, оскільки $(m_s, M_s) = 1$; $x_0 = M_1 \overset{`}{M_1} b_1 + M_2 \overset{`}{M_2} b_2 + ... + M_k \overset{`}{M_k} b_k$. Тоді система (2.5) рівносильна одному порівнянню $x \equiv x_0 (\mathrm{mod}\, m_1 m_2 ... m_k)$, а саме набір розв'язків (2.5) збігається із набором розв'язків порівняння $x \equiv x_0 (\mathrm{mod}\, m_1 m_2 ... m_k)$.

Доведення. Маємо: m_s дільник M_j, при $s \neq j$. Отже, $x_0 \equiv M_s \overset{`}{M_s} b_s (\mathrm{mod}\, m_s)$, звідки $x_0 \equiv b_s (\mathrm{mod}\, m_s)$. Це означає, що система (2.5) рівносильна системі

$$\begin{cases} x \equiv x_0 (\mathrm{mod}\, m_1) \\ x \equiv x_0 (\mathrm{mod}\, m_2) \\ ... \\ x \equiv x_0 (\mathrm{mod}\, m_k), \end{cases}$$

яка, своєю чергою, рівносильна одному порівнянню $x \equiv x_0 (\mathrm{mod}\, m_1 m_2 ... m_k)$.

Приклад. Знайти число, яке у разі ділення на 4 дає залишок 1, ділення на 5 дає залишок 3, а під час ділення на 7 дає залишок 2.

Складаємо систему:

$$\begin{cases} x \equiv 1 (\mathrm{mod}\, 4) \\ x \equiv 3 (\mathrm{mod}\, 5) \\ x \equiv 2 (\mathrm{mod}\, 7), \end{cases}$$

яку розв'язуємо за допомогою китайської теореми про лишки.

$b_1 = 1$; $b_2 = 3$; $b_3 = 2$; $m_1 m_2 m_3$, тобто $M_1 = 35$, $M_2 = 28$, $M_3 = 20$.

$$35 \cdot 3 \equiv 1 (\mathrm{mod}\, 4),$$
$$28 \cdot 2 \equiv 1 (\mathrm{mod}\, 5),$$
$$20 \cdot 6 \equiv 1 (\mathrm{mod}\, 7),$$

тобто $\overset{`}{M_1} = 3$, $\overset{`}{M_2} = 2$, $\overset{`}{M_3} = 6$.

Отримаємо $x_0 = 35 \cdot 3 \cdot 1 + 28 \cdot 2 \cdot 3 + 20 \cdot 6 \cdot 2 = 513$.

Після цього за китайською теоремою про лишки:

$$x \equiv 513 (\mathrm{mod}\, 140) \equiv 93 (\mathrm{mod}\, 140),$$

тобто найменше додатне число дорівнює 93.

Лема 1. Якщо $b_1, b_2, ..., b_k$ набувають значення з повних систем лишків за модулями $m_1, m_2, ..., m_k$ відповідно, то x_0 набуває значення з повної системи лишків за модулями $m_1, m_2, ..., m_k$.

Доведення. $x_0 = A_1b_1 + A_2b_2 + ... + A_kb_k$ набуває $m_1, m_2, ..., m_k$ різних значень. Покажемо, що усі вони попарно непорівняльні за модулями $m_1, m_2, ..., m_k$.

Нехай виявилося, що

$$A_1b_1 + A_2b_2 + ... + A_kb_k \equiv A_1b_1' + A_2b_2' + ... + A_kb_k' (\mathrm{mod}\, m_1m_2...m_k).$$

Тоді

$$A_1b_1 + A_2b_2 + ... + A_kb_k \equiv A_1b_1' + A_2b_2' + ... + A_kb_k' (\mathrm{mod}\, m_s)$$

для кожного s, звідки $M_sM_s'b_s \equiv M_sM_s'b_s' (\mathrm{mod}\, m_s)$.

З порівняння $M_sM_s' \equiv 1 (\mathrm{mod}\, m_s)$ випливає $M_sM_s' \equiv 1 + m_s \cdot t$, тобто $(M_s, M_s', m_s) = 1$. Поділивши обидві частини порівняння

$$M_sM_s'b_s \equiv M_sM_s'b_s' (\mathrm{mod}\, m_s)$$

на число M_sM_s', взаємно просте з модулем m_s, отримаємо, що $b_s \equiv b_s' (\mathrm{mod}\, m_s)$, тобто $b_s = b_s'$ для кожного s.

Отже, x_0 набуває $m_1, m_2, ..., m_k$ різних значень, попарно непорівняльних за модулями $m_1, m_2, ..., m_k$, а саме значення з повної системи лишків.

2.3.7. Порівняння другого степеня. Символ Лежандра

Розглянемо найпростіші двочленні порівняння другого степеня вигляду

$$x^2 \equiv a (\mathrm{mod}\, p),$$

де a і p взаємно прості, а p – непарне просте число. Необхідно звернути увагу, що умова взаємної простоти $(a, p) = 1$ виключає з нашого розгляду випадок $a = 0$.

Потрібно визначити, за яких a найпростіше двочленне порівняння другого степеня має розв'язок, а за яких – не має. Порівняння $x^2 \equiv a (\mathrm{mod}\, p)$ має розв'язок за будь-яких a, тому що замість a достатньо підставити тільки 0 або 1, а числа 0 і 1 є квадратами. Саме тому випадок $p = 2$ не є особливо цікавим і виводиться з подальшого розгляду.

Що стосується порівняння $x^2 \equiv 0 (\mathrm{mod}\, p)$, то воно завжди має розв'язок $x = 0$. Отже, цікавить нас лише ситуація з непарним простим модулем і $a \neq 0$.

Означення. Якщо порівняння $x^2 \equiv a (\mathrm{mod}\, p)$ має розв'язок, то число a називають ***квадратним лишком*** за модулем p. В іншому випадку число a називають ***квадратним нелишком*** за модулем p.

Отже, якщо a – квадрат деякого числа за модулем p, то a – "квадратний лишок", якщо жодне число в квадраті не порівняльнне з a за модулем p, то a – "квадратний нелишок".

Приклад. Число 2 є квадратом за модулем 7, бо $4^2 \equiv 16 \equiv 2 (\mathrm{mod}\, 7)$. Отже, 2 – квадратний лишок (порівняння $x^2 \equiv 2 (\mathrm{mod}\, 7)$ має ще й інший розв'язок: $3^2 \equiv 9 \equiv 2 (\mathrm{mod}\, 7)$). Навпаки, число 3 є квадратним нелишком за модулем 7, тому що порівняння $x^2 \equiv 3 (\mathrm{mod}\, 7)$ розв'язків не має, у чому неважко переконатися послідовним перебором повної системи лишків: $x = 0, 1, 2, 3, 4, 5, 6$.

Спостереження: якщо a – квадратний лишок за модулем p, то порівняння $x^2 \equiv a (\mathrm{mod}\, p)$ має лише два розв'язки. Дійсно, якщо a – квадратний лишок за модулем p, то в порівняння $x^2 \equiv a (\mathrm{mod}\, p)$ є хоча б один розв'язок $x \equiv x_1 (\mathrm{mod}\, p)$. Тоді $x_2 = -x_1$ – теж розв'язок, адже $(-x_1)^2 = x_1^2$. Ці два розв'язки непорівняльні за модулем $p > 2$, бо з $x_1 = -x_1 (\mathrm{mod}\, p)$ випливає $2x_1 \equiv 0 (\mathrm{mod}\, p)$, тобто (оскільки $p \neq 2$) $x_1 \equiv 0 (\mathrm{mod}\, p)$, що неможливо, тому що $a \neq 0$.

Оскільки порівняння $x^2 \equiv a (\mathrm{mod}\, p)$ є порівнянням другого степеня за простим модулем, то понад два розв'язки воно мати не може (див. підп. 2.3.9, лема 2).

Спостереження. Зведена (тобто без нуля) система лишків

$$-\frac{p-1}{2}, \ldots, -2, -1, 0, 1, 2, \ldots, \frac{p-1}{2}$$

за модулем p складається з $(p-1)/2$ квадратних лишків, порівняльних з числами $1 \cdot 1^2, 2^2, \ldots, ((p-1)/2)^2$ і $(p-1)/2)$ квадратних нелишків, тобто лишків і нелишків порівну.

Дійсно, квадратні лишки порівняльні з квадратами чисел

$$-\frac{p-1}{2}, \ldots, -2, -1, 0, 1, 2, \ldots, \frac{p-1}{2},$$

тобто з числами $1 \cdot 1^2, 2^2, \ldots, ((p-1)/2)^2$, при чому всі ці квадрати різні за модулем p, тому що з $k^2 \equiv l^2 (\mathrm{mod}\, p)$, де $0, k, l, (p-1)/2$; отже, нетривіальне порівняння $x^2 \equiv k^2 (\mathrm{mod}\, p)$ має аж чотири розв'язки: l, $-l$, k, $-k$, що неможливо (див. підп. 2.3.9, лема 2).

Французький математик ***Адріен-Марі Лежандр*** (Adrien-Marie Legendre) запропонував ввести зручний символ (a/p), який читають: "символ Лежандра а за пе".

Означення. Нехай a не кратне p. Тоді символ Лежандра визначають як:

$$\left(\frac{a}{p}\right)=\begin{cases}+1, & \text{якщо } a-\text{квадратний лишок за модулем } p;\\ -1, & \text{якщо } a-\text{квадратний нелишок за модулем } p;\end{cases}$$

Теорема. (Критерій Ейлера). Нехай a не кратне p. Тоді:

$$a^{(p-1)/2}\equiv(a/p)(\mathrm{mod}\,p).$$

Доведення. За теоремою Ферма, $a^{p-1}\equiv 1(\mathrm{mod}\,p)$, тобто

$$\left(a^{\frac{p-1}{2}}-1\right)\left(a^{\frac{p-1}{2}}+1\right)\equiv 0(\mathrm{mod}\,p).$$

У лівій частині останнього порівняння один співмножник ділиться на p, адже обидва співмножники на p ділитися не можуть, інакше їхня різниця, що дорівнює двом, ділилася б на $p>2$. Отже, можливе одне і тільки одне з порівнянь:

$$a^{(p-1)/2}\equiv 1(\mathrm{mod}\,p),$$

$$a^{(p-1)/2}\equiv -1(\mathrm{mod}\,p).$$

Але довільний квадратний лишок a задовольняє за деякого x порівняння $a\equiv x^2(\mathrm{mod}\,p)$ і, отже, задовольняє також отримане за ним почленним піднесенням до степеня $(p-1)/2$ порівняння $a^{(p-1)/2}\equiv x^{p-1}\equiv 1(\mathrm{mod}\,p)$ (знову теорема Ферма). Квадратні лишки є всіма розв'язками порівняння $a^{(p-1)/2}\equiv 1(\mathrm{mod}\,p)$, оскільки, будучи порівнянням степеня $(p-1)/2$, воно не може мати понад $(p-1)/2$ розв'язків. Це означає, що квадратні нелишки задовольняють порівняння $a^{(p-1)/2}\equiv -1(\mathrm{mod}\,p)$.

Властивість $a^{(p-1)/2}\equiv(a/p)(\mathrm{mod}\,p)$, що дає критерій Ейлера, можна прийняти за визначення символу Лежандра, показавши попередньо за теоремою Ферма, що $a^{(p-1)/2}\equiv\pm 1(\mathrm{mod}\,p)$.

Приклад. Чи буде число 5 квадратом за модулем 7?

$$5^{(7-1)/2}=5^3=125=18\cdot 7-1=-1(\mathrm{mod}\,7),$$

тобто порівняння $x^2\equiv 5(\mathrm{mod}\,7)$ розв'язків не має і 5 – квадратний нелишок за модулем 7. Перелічимо найпростіші властивості символу Лежандра.

Властивість 1. Якщо $a\equiv b(\mathrm{mod}\,p)$, то $(a/p)=(b/p)$.

Ця властивість випливає з того, що числа того самого класу за модулем p будуть всі одночасно квадратними лишками або квадратними нелишками.

Властивість 2. $(1/p)=1$.

Доказ очевидний, адже одиниця є квадратом.

Властивість 3. $\left(\frac{-1}{p}\right)=(-1)^{\frac{p-1}{2}}$.

Доказ цієї властивості випливає з критерію Ейлера при $a=-1$. Оскільки $(p-1/2)$ – парне, якщо p виду $4n+1$, і непарне, якщо p виду $4n+3$, то число -1 є квадратним лишком за модулем p тоді й тільки тоді, коли p виду $4n+1$.

Властивість 4. $\left(\frac{ab}{p}\right)=\left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$.

Дійсно, $\left(\frac{ab}{p}\right)\equiv(ab)^{\frac{p-1}{2}}\equiv a^{\frac{p-1}{2}}b^{\frac{p-1}{2}}\equiv\left(\frac{a}{p}\right)\left(\frac{b}{p}\right)(\mathrm{mod}\,p)$.

Властивість 4 поширюється на будь-яку кількість співмножників у чисельнику символу Лежандра, взаємно простих з p. Крім того, з неї випливає наступна влакстивість.

Властивість 5. $\left(\frac{ab^2}{p}\right)=\left(\frac{a}{p}\right)$, тобто в чисельнику символу Лежандра можна відкинути будь-який квадратний множник. Дійсно:

$$\left(\frac{ab^2}{p}\right)\equiv\left(\frac{a}{p}\right)\left(\frac{b^2}{p}\right)\equiv\left(\frac{a}{p}\right)\cdot 1\equiv\left(\frac{a}{p}\right)(\mathrm{mod}\,p).$$

2.3.8. Алгоритми перевіряння чисел на простоту

Актуальним питанням для відкритої криптографії є визначення простоти чисел. Зокрема для алгоритму RSA (див. розд. 3) необхідно згенерувати 2 великі прості числа, бітова довжина яких для забезпечення стійкості алгоритму сьогодні складає не менша за 1 кбіт. Детальний і всебічний опис математичного обґрунтування алгоритму RSA (розд. 3.) можна знайти в [4].

Для знаходження великого простого числа випадково вибирають велике непарне число і перевіряють його на простоту. У випадку невдачі повторюють операцію для наступного вибраного числа доти, поки воно не задовольнятиме

умову простоти з необхідною імовірністю. Деякі із методів перевіряння чисел на простоту ґрунтуються на уже викладеному матеріалі. Методи поділяють на детерміновані та імовірнісні. Розглянемо імовірнісні методи, що переважно застосовують для пошуку простих чисел.

Тест Ферма. Для перевірки числа P на простоту вибираємо випадкове число $a < P-1$. Якщо $a^{P-1} \neq 1(\mathrm{mod}\, P)$, число P вважаємо складеним. Провівши розрахунки для різних a n-ну кількість разів і у всіх випадках підтвердивши рівність $a^{P-1} = 1(\mathrm{mod}\, P)$, робимо висновок, що число P є імовірно простим із імовірністю помилки не більшою за $(1/2)^2$. Застосування теореми Ферма для визначення, чи є число простим, може нам дати відповідь лише з певною імовірністю, оскільки зворотне твердження до малої теореми Ферма не є вірним. Тобто існують такі числа, їх ще називають ***числами Кармайкла*** (B. D. Carmichael), які за всіх $a < P-1$ задовольнятимуть умови малої теореми Ферма.

Тест Соловея–Штрассена. Він є вільним від цієї вади і оснований на теоремі:

Теорема. Для довільного простого n і довільного $a \in Z_n^*$ виконують порівняння

$$a^{\frac{n-1}{2}} \equiv \left(\frac{a}{n}\right)(\mathrm{mod}\, n) .$$

Тобто, це є необхідний і достатній критерій того, що число n просте.

Доведення. Необхідність доводять так: якщо n просте, то враховуючи критерій Ейлера, для символу Лежандра роблять висновок про виконання такого порівняння. Достатність доводять від протилежного.

На тесті Соловея–Штрассена базується одноіменний алгоритм, який полягає у виконанні тесту k раундів для числа n. При цьому імовірність помилки менша за 1/2, і за k раундів вона буде меншою за 2^{-k}. Для визначення простоти числа в кожному раунді виконують такі дії:

– обчислюють найменший спільний дільник $НСД(a,n)$ для випадково вибраного числа $a < n$;

– якщо $НСД(a,n) > 1$, то n складене, у протилежному випадку обчислюють $a^{\frac{n-1}{2}} \equiv \left(\frac{a}{n}\right)(\mathrm{mod}\, n)$. Якщо це виконується, то число n є імовірно простим, а число a – свідком простоти числа n. У випадку знаходження після k раундів k свідків простоти вважають n імовірно простим числом. Загальна обчислювальна складність алгоритму $O(k \log_2 n)$, де "O" читається як

"O велике", асимптотичне позначення (див. розд. п. 2.4.). Ефективнішим тестом вважають тест Міллера–Рабіна з коефіцієнтом імовірної похибки 4^{-k} після k раундів.

Базою ***тесту Міллера–Рабіна*** є наступна теорема (критерій непростоти).

Теорема. Нехай p – непарне ціле число, тоді можна записати $p-1=2^j q$, де q – непарне ціле. У випадку, якщо справджується нерівність $a^q \neq \pm 1 (\mathrm{mod}\, p)$ або $a^{2^j q} \neq -1 (\mathrm{mod}\, p)$, де $a \in Z_p^*$, то число є p складеним, у протилежному випадку – імовірно просте.

Алгоритм Міллера–Рабіна, так само, як і алгоритм Соловея–Штрассена, є імовірнісним, обчислення в якому здійснюють k раундів. Для визначення непростоти непарного цілого числа p:

– записують парне число $p-1$ у вигляді добутку степені двійки і непарного числа q ($p-1=2^j q$);

– вибирають випадково число $1<a<p-2$ і обчислюють $a^q (\mathrm{mod}\, p)$, $a^{2^j q} (\mathrm{mod}\, p)$. Якщо знайдеться таке число a, за якого перше значення не дорівнюватиме 1, а друге не дорівнювати –1, роблять висновок, що число p складене. Після цього вибирають наступного кандидата для перевіряння простоти. У протилежному випадку число a є свідком імовірної простоти числа p. Отже, якщо визначають підряд k свідків простоти, то роблять висновок, що число p є імовірно простим з імовірністю, не меншою за $1-4^{-k}$.

Крім того, знайдені великі прості числа мають бути ***сильними*** з погляду криптографії [5]. Це зменшить можливості для факторизації RSA-ключів сучасними криптоаналітичними методами. Для визначення сильних простих чисел у криптографії згідно з [5] використовують такі правила:

– число p – достатньо велике (у [5] бітова довжина $p \geq 256$ р);

– найбільший дільник $p-1$ (його позначимо p^-) – достатньо великий (достатньо великим надалі, поки не буде зазначено додатково, вважатимемо число згідно з [5], бітова довжина якого ≥ 100), тобто можна записати $p = a^- p^- + 1$ для деякого цілого a^- і великого простого p^-;

– найбільший дільник $p^- - 1$ (його позначимо p^{--}) – достатньо великий, тобто можна записати $p^- = a^{--} p^{--} + 1$ для деякого цілого a^{--} і великого простого p^{--};

– найбільший дільник $p+1$ (його позначимо p^+) – достатньо великий, тобто можна записати $p = a^+ p^+ - 1$ для деякого цілого a^+ і великого простого p^+.

У деяких випадках просте число вважають сильним, якщо воно задовольняє лише підмножину наведених умов, а саме:

– p^- сильне, якщо p^- достатньо велике;

– p^{--} сильне, якщо p^{--} достатньо велике;

– p^+ сильне, якщо p^+ достатньо велике;

– (p^-, p^+) сильні, якщо p^- і p^+ достатньо великі;

– (p^-, p^{--}, p^+) дійсно сильні, якщо всі p^-, p^{--} і p^+ достатньо великі.

Аналогічні правила дійсні відповідно і для другого простого числа q. Існують алгоритми знаходження сильних простих чисел.

Алгоритм Гордона

1. Знаходимо p^{--} і p^+ великі прості числа (130 бітів кожне).

2. Обчислюємо p^- як найменше просте за формулою $p^- = a^{--} p^{--} + 1$ для деякого цілого a^{--}.

3. Нехай $p_0 = ((p^+)^{p^- - 1} - (p^-)^{p^+ - 1}) \bmod (p^- p^+)$.

Для коректності алгоритму зазначимо, що згідно із малою теоремою Ферма $p_0 \equiv 1 (\bmod p^-)$ і $p_0 \equiv -1 (\bmod p^+)$.

4. Обчислюємо p^- як найменше просте за формулою $p = p_0 + a p^- p^+$ для деякого цілого a.

2.3.9. Порівняння будь-якого степеня за простим модулем

Розглянемо порівняння вигляду $f(x) \equiv 0 (\bmod p)$, де p – просте число; $f(x) = ax^n + a_1 x^{n-1} + ... + a_n$ – многочлен із цілими коефіцієнтами.

Лема 1. Довільне порівняння $f(x) \equiv 0 (\bmod p)$, де p – просте число, рівносильне деякому порівнянню степеня, не вищого за $p-1$.

Доведення. Розділимо $f(x)$ на многочлен $x^p - x$ із залишком:

$$f(x) = (x^p - x) \cdot Q(x) + R(x),$$

де степінь залишку $R(x)$ не перевищує $p-1$. Але за теоремою Ферма, $x^p - x \equiv 0(\mathrm{mod}\, p)$. Це означає, що $f(x) \equiv R(x)(\mathrm{mod}\, p)$, а вихідне порівняння рівносильне порівнянню $R(x) \equiv 0(\mathrm{mod}\, p)$.

За допомогою доведеної леми можна звести розв'язування порівняння високого степеня до розв'язування порівняння меншого степеня.

Лема 2. Якщо порівняння $a_0x^n + a_1x^{n-1} + ... + a_n \equiv 0(\mathrm{mod}\, p)$ степеня n за модулем p, де p – просте число, є більше n різних розв'язків, то всі коефіцієнти $a_0, a_1, ..., a_n$ кратні p.

Доведення. Нехай порівняння $a_0x^n + a_1x^{n-1} + ... + a_n \equiv 0(\mathrm{mod}\, p)$, має $n+1$ розв'язок і $x_1, x_2, ..., x_n, x_{n+1}$ – найменші невід'ємні лишки цих розв'язків. Тоді многочлен $f(x)$ подамо у вигляді:

$$\begin{aligned} f(x) = {} & a(x-x_1)(x-x_2)...(x-x_{n-2})(x-x_{n-1})(x-x_n) + \\ & +b(x-x_1)(x-x_2)...(x-x_{n-2})(x-x_{n-1}) + \\ & +c(x-x_1)(x-x_2)...(x-x_{n-2}) + \\ & +... + \\ & +k(x-x_1)(x-x_2) + \\ & +l(x-x_1) + \\ & +m. \end{aligned}$$

Коефіцієнт b має дорівнювати коефіцієнту при x^{n-1} за різниці

$$f(x) - a(x-x_1)(x-x_2)...(x-x_n);$$

коефіцієнт с – це коефіцієнт перед x^{n-2} за різниці

$$f(x) - a(x-x_1)(x-x_2)...(x-x_n) - b(x-x_1)(x-x_2)...(x-x_{n-1}).$$

Підставивши послідовно $x = x_1$, x_2, …, x_n, x_{n+1} ..., отримуємо:

1) $f(x_1) = m \equiv 0(\mathrm{mod}\, p)$, отже, m ділиться на p.

2) $f(x_2) = m + l(x_2 - x_1) \equiv l(x_2 - x_1) \equiv 0(\mathrm{mod}\, p)$, отже, l ділиться на p, тому що $x_2 - x_1$ не може ділитися на p, оскільки $x_2 < p$, $x_1 < p$.

3) $f(x_3) = k(x_3 - x_1)(x_3 - x_2) \equiv 0(\mathrm{mod}\, p)$, отже, k ділиться на p.

Отже, усі коефіцієнти a, b, c,..., k, l кратні p. Це означає, що всі коефіцієнти a, a_1,…, a_n теж кратні p, адже вони є сумами чисел, кратних p.

Довільне нетривіальне порівняння за $\mathrm{mod}\, p$ рівносильне порівнянню степеня, не вищого за $p-1$, і має не більше $p-1$ розв'язків.

Теорема Вільсона. Порівняння $(p-1)!+1 \equiv 0(\mathrm{mod}\, p)$ виконується тоді й тільки тоді, коли p – просте число.

Доведення. Нехай p – просте число. Якщо $p=2$, то $1!+1 \equiv 0(\mathrm{mod}\, 2)$. Якщо $p>2$, то розглянемо порівняння: $[(x-1)(x-2)...(x-(p-1))]-$ $-(x^{p-1}-1) \equiv 0(\mathrm{mod}\, p)$.

Це порівняння є степеня не вищого за $p-2$ і має $p-1$ розв'язок: 1, 2, 3, ... , $p-1$, тому що при підстановці кожного з цих чисел доданок у квадратних дужках дорівнюватиме нулю, а x^{p-1} порівняльне з нулем за теоремою Ферма (x і p взаємно прості, тому що $x<p$). Це означає, за лемою 2, що всі коефіцієнти виписаного порівняння кратні p, зокрема, на p ділиться його вільний член, який дорівнює $1 \cdot 2 \cdot 3 \cdot ... \cdot (p-1)+1$.

Якщо p – не просте, то знайдеться дільник d числа p, $1<d<p$. Тоді $(p-1)!$ ділиться на d, тому $(p-1)!+1$ не може ділитися на d, і тому не може ділитися також i на p. Отже, порівняння $(p-1)!+1 \equiv 0(\mathrm{mod}\, 2)$ не виконується.

Приклад. $1 \cdot 2 \cdot 3 \cdot ... \cdot 10+1=3628800+1=3628801$ – ділиться на 11 (ознака ділення на 11: якщо сума цифр у десятковому записі числа на парних позиціях збігається із сумою цифр на непарних позиціях, то число кратне 11).

2.3.10. Порівняння будь-якого степеня за складним модулем

Теорема 1. Якщо числа $m_1, m_2, ..., m_k$ попарно взаємно прості, то порівняння $f(x) \equiv 0(\mathrm{mod}\, m_1 m_1 ... m_k)$ рівносильне системі порівнянь:

$$\begin{cases} f(x) \equiv 0(\mathrm{mod}\, m_1) \\ f(x) \equiv 0(\mathrm{mod}\, m_2) \\ ... \\ f(x) \equiv 0(\mathrm{mod}\, m_k) \end{cases}.$$

При цьому, якщо $T_1, T_2, ..., T_k$ – кількість розв'язків окремих порівнянь цієї системи за відповідними модулями, то кількість розв'язків початкового порівняння дорівнює $T_1 T_2 ... T_k$.

Доведення. Перше твердження теореми (про рівносильність системи і порівняння) очевидне, оскільки якщо $a \equiv b(\mathrm{mod}\, m)$, то $a \equiv b(\mathrm{mod}\, d)$, де m не

ділиться на d. Якщо ж $a \equiv b(\mathrm{mod}\, m_1)$ і $a \equiv b(\mathrm{mod}\, m_2)$, то $a \equiv b(\mathrm{mod}\, НСК(m_1, m_2))$, де $НСК(m_1, m_2)$ – найменше спільне кратне m_1 і m_2 (з властивостей порівнянь).

Перейдемо до другого твердження теореми (про кількість розв'язків порівняння).

Кожне порівняння $f(x) \equiv 0(\mathrm{mod}\, m_s)$ виконується тоді й тільки тоді, коли виконується одне з T_s порівнянь вигляду $x \equiv b_s(\mathrm{mod}\, m_s)$, де b_s набувають значення лишків розв'язків порівняння $f(x) \equiv 0(\mathrm{mod}\, m_s)$. Загалом різних комбінацій таких найпростіших порівнянь

$$\begin{cases} x \equiv b_1(\mathrm{mod}\, m_1) \\ x \equiv b_2(\mathrm{mod}\, m_2) \\ \ldots \\ x \equiv b_k(\mathrm{mod}\, m_k) \end{cases}$$

$T_1 T_2 \ldots T_k$ різновидів. Усі ці комбінації призводять до різних класів лишків за $\mathrm{mod}(m_1 m_2 \ldots m_k)$...

Отже, розв'язування порівняння $f(x) \equiv 0(\mathrm{mod}\, p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_k^{\alpha_k})$ зводиться до розв'язування порівнянь вигляду $f(x) \equiv 0(\mathrm{mod}\, p^a)$. Розв'язування цього останнього порівняння, своєю чергою, зводиться до розв'язування деякого порівняння $g(x) \equiv 0(\mathrm{mod}\, p)$ з іншим многочленом у лівій частині, але вже з простим модулем, що було розглянуто раніше. Зведемо розв'язування порівняння $f(x) \equiv 0(\mathrm{mod}\, p^s)$ до розв'язування порівняння $g(x) \equiv 0(\mathrm{mod}\, p)$.

Процес зведення.

З виконання порівняння $f(x) \equiv 0(\mathrm{mod}\, p^a)$ випливає те, що x задовольняє порівняння $f(x) \equiv 0(\mathrm{mod}\, p)$. Нехай $x \equiv x_1(\mathrm{mod}\, p)$ – який-небудь розв'язок порівняння $f(x) \equiv 0(\mathrm{mod}\, p)$. Це означає, що

$$x = x_1 + p \cdot t_1, \text{ де } t_1 \in Z.$$

Підставимо це x до порівняння $f(x) \equiv 0(\mathrm{mod}\, p^2)$. Отримаємо порівняння

$$f(x_1 + p \cdot t_1) \equiv 0(\mathrm{mod}\, p^2),$$

яке також виконується.

Розкладемо ліву частину отриманого порівняння за формулою Тейлора за степенями $(x - x_1)$:

$$f(x)=f(x_1)+\frac{f'(x_1)}{1!}(x-x_1)+\frac{f''(x_1)}{2!}(x-x_1)^2+\ldots$$

Оскільки $x=x_1+p\cdot t_1$, то

$$f(x_1+p\cdot t_1)=f(x_1)+\frac{f'(x_1)}{1!}p\cdot t_1+\frac{f''(x_1)}{2!}p^2\cdot t_1^2+\ldots.$$

Зауважимо, що число $f^{(k)}(x_1)/k!$ завжди ціле, бо $f(x_1+p\cdot t_1)$ – многочлен з цілими коефіцієнтами. Тепер у порівнянні

$$f(x_1+p\cdot t_1)\equiv 0(\mathrm{mod}\,p^2)$$

можна зліва відкинути члени, кратні p^2:

$$f(x_1)+\frac{f'(x_1)}{1!}p\cdot t_1\equiv 0(\mathrm{mod}\,p^2).$$

Розділимо останнє порівняння і його модуль на p:

$$\frac{f(x_1)}{p}+\frac{f'(x_1)}{1!}t_1\equiv 0(\mathrm{mod}\,p).$$

Зауважимо, що $f(x_1)/p$ – ціле число, тому що $f(x_1)\equiv 0(\mathrm{mod}\,p)$. Обмежимося випадком, коли значення похідної $f'(x_1)$ не ділиться на p. У цьому випадку існує єдиний розв'язок порівняння першого степеня

$$\frac{f(x_1)}{p}+\frac{f'(x_1)}{1!}t_1\equiv 0(\mathrm{mod}\,p)\ \text{відносно}\ t_1:\ t_1\equiv t_1'(\mathrm{mod}\,p).$$

Це означає, що $t_1=t_1'+p\cdot t_2$, де $t_2\in Z$, і

$$x=x_1+p\cdot t_1=\underbrace{x_1+p\cdot t_1'}_{x_2}+p^2t_2=x_2+p^2t_2.$$

Підставимо це $x=x_2+p^2t_2$ у порівняння $f(x)\equiv 0(\mathrm{mod}\,p^3)$ (але тепер це порівняння за $\mathrm{mod}\,p^3$), розкладемо його ліву частину за формулою Тейлора за степенями $(x-x_2)$ і відкинемо члени, кратні p^3:

$$f(x_2)+(f'(x_2)/1!)\cdot p^2t_2\equiv 0(\mathrm{mod}\,p^3).$$

Поділимо це порівняння і його модуль на p^2:

$$f(x_2)/p^2+f'(x_2)\cdot t_2\equiv 0(\mathrm{mod}\,p).$$

$f(x_2)/p^2$ – ціле число, адже число t_1' таке, що $f(x_1+p\cdot t_1)\equiv 0(\mathrm{mod}\,p^2)$. Крім того, $x_2\equiv x_1(\mathrm{mod}\,p)$, отже, $f'(x_2)\equiv f'(x_1)(\mathrm{mod}\,p)$, тобто $f'(x_2)$, як і $f'(x_1)$, не

ділиться на p. Маємо єдиний розв'язок порівняння першого степеня $f(x_2)/p^2 + f'(x_2)\cdot t_2 \equiv 0(\mathrm{mod}\, p)$ відносно t_2:

$$t_2 \equiv t_2'(\mathrm{mod}\, p).$$

Це означає, що $t_2 \equiv t_2' + p\cdot t_3$, де $t_3 \in Z$, і

$$x = \underbrace{x_2 + p^2\cdot t_2'}_{x_3} + p^3 t_3 = x_3 + p^3 t_3,$$

і процес продовжується далі, до досягнення степеня p^a, у якому стоїть просте число p у модулі початкового порівняння $f(x)\equiv 0(\mathrm{mod}\, p^a)$.

Отже, довільний розв'язок $x \equiv x_1(\mathrm{mod}\, p)$ порівняння $f(x)\equiv 0(\mathrm{mod}\, p)$, за умови $p/f'(x_1)$ не ділиться на p, дає один розв'язок порівняння $f(x)\equiv 0(\mathrm{mod}\, p^a)$ вигляду $x = x_a + p^a\cdot t_a$, тобто $x \equiv x_a(\mathrm{mod}\, p^a)$.

Приклад. Розв'язати порівняння $x^4 + 7x + 4 \equiv 0(\mathrm{mod}\, 27)$.

$27 = 3^3$. За допомогою перебору повної системи лишків за mod3 отримаємо, що порівняння $x^4 + 7x + 4 \equiv 0(\mathrm{mod}\, 3)$ має єдиний розв'язок $x \equiv 1(\mathrm{mod}\, 3)$.

$$f'(x) = (4x^3 + 7)|_{x=1} \equiv 2(\mathrm{mod}\, 3),$$

тобто не ділиться на p=3.

$$x_1 = 1 + 3\cdot t_1,$$

$$f(1) + f'(1)\cdot 3\cdot t_1 \equiv 0(\mathrm{mod}\, 3^2).$$

Знаходимо t_1:

$$3 + 3\cdot t_1\cdot 2 \equiv 0(\mathrm{mod}\, 9),$$

після ділення на $p = 3$:

$$1 + 2\cdot t_1 \equiv 0(\mathrm{mod}\, 3),$$

$$t_1 \equiv 1(\mathrm{mod}\, 3) \text{ – єдиний розв'язок.}$$

$$t_1 = 1 + 3t_2,\ x = 1 + 3t_1 = 4 + 9t_2,$$

$$f(4) + 9\cdot t_2\cdot f'(4) \equiv 0(\mathrm{mod}\, p^3 = 27),\ 18 + 9\cdot 20\cdot t_2\ 0(\mathrm{mod}\, 27),$$

і після ділення на $p^2 = 9$ обчислюємо t_2:

$$2 + 20t_2 \equiv 0(\mathrm{mod}\, 3),$$

$$t_2 \equiv 2(\mathrm{mod}\, 3),\ t_2 = 2 + 3\cdot t_3,$$

звідки

$$x = 4 + 9\cdot(2 + 3t_3) = 22 + 27t_3.$$

Отже, єдиним розв'язком вихідного порівняння є $x \equiv 22(\mathrm{mod}\,27)$.

Наступна теорема належить до специфічного виду порівнянь.

Теорема 2. Нехай A, m, n – натуральні числа; $(A,m)=1$, $x \equiv x_0(\mathrm{mod}\,m)$ – один із розв'язків порівняння $x^n \equiv A(\mathrm{mod}\,m)$.

Тоді всі розв'язки цього порівняння отримують множенням x_0 на лишки розв'язків порівняння $y^n \equiv 1(\mathrm{mod}\,m)$.

Доведення. Перемножимо порівняння:

$$\begin{array}{l} x_0^n \equiv A(\mathrm{mod}m) \\ \underline{y^n \equiv 1(\mathrm{mod}m)} \quad \times \\ (x_0 y)^n \equiv A(\mathrm{mod}m), \end{array}$$

звідки видно, що $x_0 y$ – розв'язок порівняння $x^n \equiv A(\mathrm{mod}\,m)$.

Доведемо далі, що за різних y_1 та y_2 розв'язки є різні, тобто $y_1 \neq y_2(\mathrm{mod}\,m)$ і $x_0 y_1 \neq x_0 y_2(\mathrm{mod}\,m)$. Припустімо, що $x_0 y_1 = x_0 y_2(\mathrm{mod}\,m)$. Очевидно, що $(x_0, m)=1$, тому що інакше було б:

$$x_0 = d \cdot x_0^{'},\ m = d \cdot m^{'},$$

$$x_0 = d^n \cdot (x_0^{'})^n \equiv A(\mathrm{mod}\, d\, m^{'}),$$

Отже, A ділиться на d і m ділиться на d, що суперечить взаємній простоті A і m. Отже, $(x_0, m) \equiv 1$, і порівняння $x_0 y_1 = x_0 y_2(\mathrm{mod}\,m)$ можна поділити на x_0: $y_1 \equiv y_2(\mathrm{mod}\,m)$, а це суперечить початковому припущенню. Отже, для різних y_1 та y_2 отримуємо різні розв'язки.

Залишилося переконатися, що кожний розв'язок порівняння $x^n \equiv A(\mathrm{mod}\,m)$ одержуємо саме таким способом. Маємо:

$$x^n \equiv A(\mathrm{mod}\,m),\ x_0^n \equiv A(\mathrm{mod}\,m),$$

отже, $x^n \equiv x_0^n(\mathrm{mod}\,m)$. Візьмемо число y таке, що $x \equiv y\,x_0(\mathrm{mod}\,m)$. Тоді $y^n x_0^n \equiv x_0^n(\mathrm{mod}\,m)$, тобто $y^n \equiv 1(\mathrm{mod}\,m)$.

2.4. Алгоритми та їх складність

У практиці програмування розрізняють швидкі та повільні алгоритми. Ці характеристики є важливими для криптографії. З одного боку, користувачі криптосистеми повинні володіти швидкими алгоритмами шифрування та дешифрування, а з іншого – їх суперник не повинен мати швидкого алгоритму

розкриття криптотексту без знання ключа. В ідеалі, суперник не має швидкого алгоритму розкриття не тому, що він не може створити такий алгоритм, а тому, що останнього взагалі не існує – усі алгоритми зламу криптосистем є повільними.

2.4.1. Задачі й алгоритми

Розглянемо функції вигляду $f: A^* \to B^*$, де A і B – деякі алфавіти. Цей клас функцій виникає для функцій, які можна ефективно обчислювати. Наприклад, функцію піднесення до квадрата у множині невід'ємних цілих чисел з обчислювального погляду природно трактувати як функцію із множини $\{0,1,...,9\}^*$ в себе. Слід лише домовитись, як бути з десятковими словами, що починаються цифрою 0. Їх можна утотожнити з невід'ємними цілими десятковими числами, що отримують після відкидання перших нулів. Також можна вважати, що функція відображає слова, які не є десятковим записом натурального числа, в 0.

Задача обчислення функції $f: A^* \to B^*$ полягає у знаходженні для вказаного слова $w \in A^*$ значення функції $f(w)$. Задачі обчислення описують так:

Задано: $w \in A^*$.

Обчислити: $f(w)$.

Якщо не буде важливо, в якому алфавіті і як саме кодують аргументи та значення функції, допустимими є й менш формальні описи задач, наприклад:

Задано: $x \in \mathrm{N}$.

Обчислити: x^2.

Іноді задачу у вищеозначеному сенсі називають ***масовою***. Конкретне задане $w \in A^*$ – це ***індивідуальна*** задача. Значення $f(w)$ називають розв'язком індивідуальної задачі w. Масову задачу можна вважати нескінченним набором індивідуальних задач.

Нехай $L \subseteq A^*$ – множина слів у алфавіті A^*. L називають мовою.

Задача розпізнавання мови L – визначити, чи належить задане слово $w \in A^*$ цій мові, чи ні:

Задано: $w \in A^*$.

Розпізнати: $w \in L$?

Наприклад,

Задано: $x \in \mathrm{N}$.

Розпізнати: чи є x повним квадратом ?

Індивідуальною задачею є будь-яке задання слова w. Розв'язком індивідуальної задачі є відповідь "так" чи "ні", яку прийнято кодувати символами 1 та 0, відповідно.

Ще один різновид задачі називають задачею пошуку елемента із заданою властивістю. Нехай алфавіт A не містить символу # і $P \subseteq (A \cup \{\#\})^*$. Для кожного заданого $w \in A^*$ задача полягає або у знаходженні хоча б одного $u \in A^*$ такого, що $w\#u \in P$, або у констатації, що елемента u з такою властивістю немає:

Задано: $w \in A^*$.

Знайти: u таке, що $w\#u \in P$.

Індивідуальною задачею є довільно задане слово $w \in A^*$, а її розв'язком є або відповідне u, або відповідь "не існує", яку зручно кодувати символом #. Наприклад,

Задано: $x \in \mathrm{N}$.

Знайти: $y \in \mathrm{Z}$ таке, що $x = y^2$.

В обчислювальному сенсі всі три типи задач рівносильні між собою – кожну задачу одного типу можна переформулювати як задачу будь-якого з двох інших типів.

Наприклад, задача розпізнавання мови $L \subseteq A^*$ є задачею обчислення її характеристичної функції $\chi_L : A^* \to \{0,1\}$, що набуває значення 1 на аргументах з L і лише на них. Задачу обчислення функції $f : A^* \to B^*$ легко подати як задачу пошуку, взявши $P = \{w\#f(w) : w \in A^*\}$, де P — множина слів у алфавіті $A \cup B \cup \{\#\}$. Своєю чергою, кожну задачу пошуку можна звести до деякої задачі розпізнавання.

Для алгоритмів вважають зафіксованими два алфавіти: вхідний A та вихідний B. Робота алгоритму полягає в тому, що він отримує на вхід слово у вхідному алфавіті – вхід, і як результат виконання послідовності елементарних операцій подає на вихід слово у вихідному алфавіті – вихід. Поняття елементарної операції, або кроку роботи, є складовою формального означення алгоритму.

Алгоритм розв'язує масову задачу, якщо, отримавши на вхід будь-яку індивідуальну задачу $w \in A^*$, він за скінченну кількість кроків подає на вихід її розв'язок. Залежно від типу задачі алгоритм обчислює функцію, розпізнає множину чи мову, знаходить елемент із певною властивістю. У випадку задачі розпізнавання, якщо алгоритм подає на вихід 1, він приймає вхід, а якщо 0, то відхиляє вхід.

Довжиною входу $w \in A^*$ називають кількість букв у слові w, яку позначаємо $|w|$. Нехай $t : \mathrm{N} \to \mathrm{N}$ – деяка функція. Алгоритм розв'язує задачу протягом часу t, якщо на кожному вході w він робить не більше ніж $t(|w|)$ кроків. Якщо $t(n) \le cn^c$ для деякої константи c, то алгоритм розв'язує задачу протягом поліномного від довжини входу часу. Такий алгоритм називають ***поліномним***, а задачу – ***поліномно розв'язною***.

Поняття поліномного часу є основною концепцією теорії складності обчислень. Згідно з цією концепцією поліномні алгоритми відповідають швидким, ефективним на практиці алгоритмам, а поліномно розв'язні задачі є легкими задачами, які можна розв'язати за прийнятний час на входах довжини, що має практичний інтерес. Часто поліномний алгоритм задовольняє всі практичні проблеми. У гіршому випадку його можна вважати швидким лише асимптотично, а на відносно невеликій кількості входів алгоритм може працювати довго.

Клас поліномно розв'язних задач не залежить від способу формулювання задачі. Наприклад, задачу множення натуральних чисел ефективно розв'язують як у двійковій, так і в десятковій системах числення. Використання іншого алфавіту не здатне суттєво прискорити обчислення (тобто зменшити час як функцію від довжини входу). Винятком є формулювання задачі в односимвольному алфавіті. Цей випадок виключають як такий, що не має застосувань.

Алгоритм, який на нескінченній послідовності входів робить понад 2^{n^c} кроків, де n – довжина входу, а $c > 0$ – деяка константа – називають ***експонентним***. Такий алгоритм потребує експонентного часу. Експонентні алгоритми є повільними, неефективними на практиці алгоритмами. Прикладом є алгоритм зламування шифру повним перебором ключів. Задача, яку можна розв'язати лише експонентним алгоритмом, є важкою. Можлива ситуація, коли для задачі відомі лише експонентні алгоритми, але не вдається довести, що кращих не існує. На практиці таку задачу вважають важкою (доки для неї не буде знайдено ефективного алгоритму). Прикладом може бути задача пошуку, на важкості якої грунтуються сучасні криптосистеми:

Задано: $x \in \mathrm{N}$.

Знайти: нетривіальний дільник числа x.

Взагалі кажучи, експонентний алгоритм є повільним у найгіршому випадку – достатньо, щоб він затрачав час 2^{n^c} хоча б на одному з k^n входів довжини n, де k – кількість букв у вхідному алфавіті. Можлива ситуація, коли повільний у найгіршому випадку алгоритм є доволі швидким у середньому, тобто для переважної більшості входів довжини n. Тоді такий експонентний алгоритм може бути придатний для практичних потреб.

Множина індивідуальних задач є множиною всіх слів у деякому алфавіті A. Інколи розглядають звуження масової задачі на деяку підмножину індивідуальних задач $S \subset A^*$. Алгоритм розв'язує звужену задачу, якщо він видає правильний розв'язок для індивідуальних задач із множини S.

2.4.2. Асимптотичні позначення

Оцінювання складності виконання алгоритмів має велике значення для криптографії. Так, для операцій шифрування/дешифруваня алгоритми повинні бути швидкими, водночас не має існувати швидкого алгоритму для криптоаналізу зашифрованої інформації.

Для порівняння складності різних алгоритмів існує стандартна нотація, що характеризує час виконання алгоритму, який, своєю чергою, прямо пропорційний до кількості необхідних операцій алгоритму залежно від величини входу n [6, 7]. Ефективність алгоритмів описують множиною асимптотичних функцій, визначених на множині невід'ємних цілих чисел. Так, для функції $g(x)$ ***асимптотичне наближення*** Θ (тета) визначає множину функцій $f(n)$, що задовольняють умову:

$$\Theta(g(x)) = \{f(n) : \exists c1, c2, n_0, 0 \le c1 g(n) \le f(n) \le c2 g(n), \forall n \ge n_0\}.$$

Асимптотичне наближення O ("О" велике) визначає множину функцій $f(n)$, що задовольняють умову:

$$\mathrm{O}(g(x)) = \{f(n) : \exists c,\ n_0, 0 \le f(n) \le cg(n), \forall n \ge n_0\}.$$

Отже, "О" велике дає нам верхню границю для функції, яку оцінюють. Тобто можна записати $f(x) = \mathrm{O}(g(x))$, якщо існують додатні константи n_0 і c такі, що значення $f(x)$ завжди лежить нижче значення $cg(n)$.

Асимптотичне наближення Ω (омега) визначає множину функцій $f(n)$, що задовольняють умову:

$$\Omega(g(x)) = \{f(n) : \exists c, n_0,\ 0 \le cg(n) \le f(n), \forall n \ge n_0\}.$$

І для нижньої границі функцій можна записати $f(x)=\Omega(g(x))$, якщо існують додатні константи n_0 і c такі, що значення $f(x)$ завжди лежить вище значення $cg(n)$.

Теорема. Для двох функцій $f(n)$ і $g(n)$ маємо рівність $f(n)=\Theta(g(n))$ тоді і тільки тоді, коли $f(n)=O(g(n))$ і $f(n)=\Omega(g(n))$.

Надалі будемо застосовувати лише $O(g(n))$, тому наведемо їх властивості, згідно із якими ми можемо виконувати операції [7]:

1. $n^m=O(n^{m`})$, якщо m ≤ m`;
2. $O(f(n))+O(g(n))=O(|f(n)|+|g(n)|)$;
3. $f(n)=O(f(n))$;
4. $cO(f(n))=O(f(n))$, якщо c – константа;
5. $O(f(n))+O(f(n))=O(f(n))$;
6. $O(O(f(n)))=O(f(n))$;
7. $O(f(n))O(g(n))=O(f(n)g(n))$;
8. $O(f(n)(g(n))=f(n)O(g(n))$.

Прямолінійні програми

Поняття прямолінійної програми можна вводити для довільної алгебраїчної структури. Зробимо це для кільця з одиницею. Елементарними операціями є множення та додавання в кільці.

Нехай задано такі об'єкти:

– символ 1, який називають символом предметної константи, і який позначає одиницю кільця;

– алфавіт $X=\{x_1,...,x_m\}$, елементи якого називають вхідними змінними;

– алфавіт $Z=\{z_1,...,z_l\}$, елементи якого називають службовими змінними;

– три символи операцій •, +, – для множення, додавання та віднімання.

Прямолінійною програмою називають послідовність із l слів у алфавіті $X\cup Z\cup\{1,\cdot,+,-,=\}$, в якій i-те слово має вигляд $z_i=z'\circ z''$, де $\circ$ є одним із символів операцій, а z', як і z'' є або символом предметної константи, або вхідною змінною, або однією зі службових змінних $z_1,...,z_{i-1}$.

Слова, з яких складається прямолінійна програма, називають її ***командами***. Кожна команда виконує операцію з операндами, які можуть бути лише результатами виконання попередніх команд. Кількість команд прямолінійної програми (l) називають її ***довжиною***, або складністю. Кількість команд множення називають мультиплікативною складністю, а додавання та віднімання – адитивною складністю програми.

Припустимо, що в нас є прямолінійна програма і нехай R – деяке кільце з одиницею. Кожна команда прямолінійної програми визначає деяку функцію $f: R^m \to R$, яка є поліномом від змінних $x_1,...,x_m$. Перша команда задає поліном вигляду $1, x_j \pm 1, x_j \pm x_i$ або $x_j \cdot x_i$. Кожна наступна команда задає функцію, що є добутком, сумою або різницею функцій, визначених якось із попередніх команд.

Функція $F: R^m \to R^k$ є набором k функцій-проекцій $f_j: R^m \to R$, де $F(x_1,...,x_m) = f_1(x_1,...,x_m),...,f_k(x_1,...,x_m)$. Прямолінійна програма обчислює функцію $F: R^m \to R^k$, якщо кожну з функцій f_j, де $1 < j < k$ визначають деякою командою цієї програми. ***Складністю обчислення функції*** F прямолінійними програмами називають найменшу довжину прямолінійної програми, що обчислює F. Подібно визначають мультиплікативну та адитивну складність функції.

Приклад. Функцію $F(x) = x^4 + x^2$ у будь-якому кільці обчислюють за такою програмою:

$$z_1 = x \cdot x,$$
$$z_2 = z_1 \cdot x,$$
$$z_3 = z_2 \cdot x,$$
$$z_4 = z_3 + z_1.$$

Отже, складність цієї функції не перевищує 4. Насправді вона менша, тому що для обчислення функції є коротша програма:

$$z_1 = x \cdot x,$$
$$z_2 = z_1 + 1,$$
$$z_3 = z_1 \cdot z_2.$$

Піднесення до степеня. Розглянемо задачу обчислення функції $f(x) = x^d$ у довільному кільці. Прямолінійна програма для цієї функції має складність $d-1$:

$$z_1 = x \cdot x,$$
$$z_2 = z_1 \cdot x,$$
$$z_3 = z_2 \cdot x,$$
$$...$$
$$z_{d-1} = z_{d-2} \cdot x.$$

Піднесення до степеня за модулем *n*

Задано: $x \in Z_n$, $d \in N$.

Обчислити: $x^d \bmod n$.

Вважаємо, що $d < n$. Якщо використати для розв'язання цієї задачі наведену вище прямолінійну програму (з множенням в кільці Z_n), то отримаємо експонентний алгоритм, тому що коли d не набагато менше, ніж n, то довжина входу має порядок $\log n$, а кількість операцій множення має порядок n.

Існує алгоритм, який називають ***бінарним методом***. Опишемо цей метод у вигляді прямолінійної програми для обчислення функції $f(x) = x^d$.

Подамо показник d у двійковій системі числення: $d = (d_l ... d_1 d_0)_2$, де $d_i \in \{0,1\}$ і $d = \sum_{i=0}^{l} d_i 2^i$. Для спрощення запису дозволимо одній команді програми виконувати до двох множень і задамо $z_0 = 1$. Тоді i-ту команду задають так:

$$z_i = \begin{cases} z_{i-1} \cdot z_{i-1}, & \text{якщо} \quad d_{l+1-i} = 0, \\ z_{i-1} \cdot z_{i-1} \cdot x, & \text{якщо} \quad d_{l+1-i} = 1. \end{cases}$$

Загалом команд $l+1$. Результатом виконання останньої є

$$(...((x^{d_l})^2 x^{d_{l-1}})^2 x^{d_{l-2}} ...)^2 x^{d_0} =$$

$$= x^{d_l 2^l + d_{l-1} 2^{l-1} + d_{l-2} 2^{l-2} + ... + 2^0 d^0} = x^d.$$

Разом витрачають $l + \sum_{i=0}^{l-1} d_i \le 2l \le 2\log_2 d$ множень (множення першої команди не є необхідним – вона включена для однорідності). Наведену прямолінійну програму можна конвертувати в алгоритм чи в програму будь-якою мовою програмування, що розв'язує задачу піднесення до степеня за модулем n.

2.4.3. Рандомізація, іморівнісні алгоритми

Алгоритми, які раніше були розглянуті, називають ***детермінованими***. Більші обчислювальні можливості мають ймовірнісні алгоритми. Окрім входу w, ймовірнісний алгоритм отримує випадкову двійкову послідовність $r \in \{0,1\}^l$, далі працює як звичайний детермінований алгоритм, і результат роботи u подає на вихід. Довжина випадкової послідовності l залежить від довжини входу.

Вихід $u = u(w,r)$ ймовірнісного алгоритму залежить не лише від входу, а й від випадкової послідовності. Випадкову послідовність вважають рівномірно розподіленою в $\{0,1\}^l$, тобто кожне r вибирають з ймовірністю 2^{-l}. Ймовірнісний алгоритм розв'язує масову задачу з ймовірністю помилки ε, якщо, отримавши на вхід індивідуальну задачу w, він подає на вихід її правильний розв'язок з ймовірністю, не меншою за $1-\varepsilon$. Інакше кажучи, вихід алгоритму $u = u(w,r)$ є розв'язком індивідуальної задачі w для всіх, окрім щонайбільше $\varepsilon 2^l$ випадкових послідовностей r.

Імовірнісні алгоритми називають ***алгоритмами Монте-Карло***. Лас-Вегас-алгоритми є підкласом алгоритмів Монте-Карло. ***Лас-Вегас-алгоритм*** розв'язує задачу з ймовірністю невдачі $\varepsilon \in \{0,1\}$, якщо на кожному вході він видає або правильний розв'язок індивідуальної задачі, або повідомлення про неспроможність такий розв'язок знайти, причому повідомлення з'являється з ймовірністю щонайбільше ε. Можна вважати, що Лас-Вегас-алгоритм взагалі не помиляється, лише часом утримується від подання на вихід розв'язку.

Час роботи ймовірнісного алгоритму на вході w означують як максимальну кількість кроків, які алгоритм робить на цьому вході з використанням випадкової послідовності r (іноді час роботи ймовірнісного алгоритму означують як середню, а не максимальну кількість кроків). Отже, поняття поліномного часу для детерміновиних алгоритмів переносять на клас імовірнісних алгоритмів.

Стосовно задач розпізнавання ймовірнісний алгоритм A розпізнає мову L з однобічною помилкою ε, де $0 < \varepsilon < 1$, якщо дотримано таких умов:

1) якщо $w \in L$, то A приймає вхід w з імовірністю 1;

2) якщо $w \notin L$, то A відхиляє вхід w з імовірністю принаймні $1-\varepsilon$.

Нехай імовірнісний алгоритм A розпізнає мову L з однобічною помилкою ε, використовуючи на вході довжини n випадкову послідовність довжини $l = l(n)$. У цьому випадку є ефективний спосіб зменшення помилки. Розглянемо ймовірнісний алгоритм A^t, який на вході w працює так:

– використовує випадкову послідовність довжини lt, де $l = l(|w|)$, розбивши її на t частин $r_1,...,r_t$ довжини l кожна;

– моделює роботу алгоритму A на вході w з використанням кожної із випадкових послідовностей $r_1,...,r_t$, і отримує t результатів моделювання $u_1,...,u_t$, де $u_i \in \{0,1\}$;

– подає на вихід 1, якщо серед всіх u_i є хоча б один 0.

Перевіримо, чи алгоритм A^t розпізнає ту ж мову L і оцінимо ймовірність помилки. Якщо $w \in L$, то алгоритм A видає 1 незалежно від набору випадкової послідовності. Тому $u_i = 1$ для всіх $i \le t$, і в цьому випадку A^t приймає w з ймовірністю 1. Якщо $w \notin L$, то $u_i = 1$ з ймовірністю щонайбільше ε для кожного i. Оскільки $r_1, ..., r_t$ є випадковими і незалежними, то рівність $u_i = 1$ виконується для всіх $i \le t$ з ймовірністю, не більшою ніж ε^t. Тому A^t приймає w не з L із ймовірністю щонайбільше ε^t і, таким чином, розпізнає L з однобічною помилкою ε^t.

Якщо алгоритм A поліномний, то таким є і алгоритм A^t, причому не лише для t константи, а й для будь-якого $t(n)$-полінома від довжини входу. Приймаючи $t(n) = cn$ для відповідної константи c, отримаємо, що якщо множину розпізнає поліномний імовірнісний алгоритм із однобічною помилкою ε, то цю ж множину розпізнає деякий поліномний алгоритм із однобічною помилкою 2^{-n}. Ймовірність такого порядку називають ***експонентно низькою***. Отже, експонентною зниження помилки (до ε^t) досягають завдяки лінійному збільшенню часу роботи (у t разів).

Вважаємо, що ймовірнісний алгоритм здатен отримати як завгодно довгу послідовність випадкових бітів, причому на отримання одного біта йде один крок роботи алгоритму.

Вибрати випадковий елемент x зі скінченної множини X означає, що елемент x має бути сконструйованим певним чином із випадкової послідовності r, причому кожен з елементів множини X мусить бути отриманий з однаковою ймовірністю.

Найпростіше влаштувати випадковий вибір елемента із множини Z_n для $n = 2^l$, або із Z_p^* для простого $p = 2^l + 1$. Для цього випадкову двійкову послідовність r довжини l розглядають як двійковий запис елемента такої множини.

Щоб отримати випадковий елемент із Z_n для довільного n, можна вибрати випадковий елемент x із Z_{2^l}, де $l = [\log_2 n]$, і якщо $x < n$, то використовувати цей елемент, якщо ж $x \ge n$, то вибір слід повторити ще раз. Внаслідок цієї процедури буде отримано $x \in Z_n$ з ймовірністю $\alpha > 1/2$. У випадку невдачі повторювати вибір доведеться не надто довго. Оскільки ймовірність того, що x потрапляє у Z_n лише за i-м разом, дорівнює $(1-\alpha)^{i-1}\alpha$.

Так само вибирають випадковий елемент із Z_p^* для довільного простого p. Щоб вибрати випадковий елемент з множини Z_{pq}^*, де p і q прості, можна скористатися наслідком з Китайської теореми про лишки, згідно з яким достатньо вибрати x_1 і x_2 – випадкові елементи множин Z_p^* і Z_{pq}^*, і обчислити $x \in Z_{pq}^*$, для якого $x_1 = x \bmod p$ і $x_2 = x \bmod q$. Такий елемент x рівномірно розподілений на Z_{pq}^*.

Існує ефективніший для великих p і q спосіб породження випадкового елемента з Z_{pq}^*. Вибираємо випадковий елемент x з Z_{pq}^* і перевіряємо, чи він ділиться на якесь із чисел p і q. Якщо ні, то $x \in Z_{pq}^*$, що й потрібно, якщо ж ділиться, то процедуру вибору повторюємо ще раз. Ймовірність того, що вибір буде успішно здійснено за один раз, дорівнює $(1-1/p)(1-1/q)$. Цей спосіб придатний для породження випадкового елемента множини Z_n^* для довільного n. А саме, вибираємо випадковий елемент x з Z_n і обчислюємо $НСД(x,n)$. Якщо останній дорівнює 1, то вибір здійснено, інакше вибір слід повторити.

2.4.4. Односторонні функції

Одностороння функція (one-way function) – це функція, яку легко обчислити, проте для неї не існує ефективного алгоритму ***інвертування***, тобто обчислення за значенням функції довільного (хоча б одного) значення з прообразу. Використовують термін інвертування, а не обертання, оскільки ***обертання*** цієї функції – це задача пошуку зворотної функції. Наприклад, щоб обернути лінійне перетворення, задане невиродженою квадратною матрицею, потрібно обчислити обернену матрицю. Інвертування – це масова задача обчислення за заданим значенням функції значення з прообразу. При цьому обернена функція може не існувати. Крім того, вважають, що цей алгоритм успішно інвертує функцію, якщо він робить це для деякої, достатньо великої частини значень.

Моделі обчислень. Ефективними алгоритмами є поліномні алгоритми. Останні можна визначити за двома моделями обчислень.

За ***однорідною*** (uniform) моделлю обчислень поліномний алгоритм – це ймовірнісна машина Тюрінга, час роботи якої обмежено поліномом від довжини вхідного слова.

За ***неоднорідною*** (nonuniform) моделлю алгоритм – це сімейство булевих схем $\{C_n\}$. Схема $\{C_n\}$ має n входів і побудована, скажімо, для визначеності, з функціональних елементів І, АБО і НЕ. Вважають, що всі елементи І і АБО двовходові. Алгоритм $\{C_n\}$ називають ***поліномним***, якщо існує поліном, що обмежує згори кількість елементів у схемі C_n для всіх достатньо великих n.

Алгоритм за неоднорідною моделлю можна визначити як сімейство машин Тюрінга $\{T_n\}$, де T_n працює на вхідних словах довжини n. Алгоритм $\{T_n\}$ називають поліномним, якщо існує поліном, що обмежує згори час роботи машини T_n для всіх достатньо великих n.

Більшість визначень і результатів теоретичної криптографії можна сформулювати за обома моделями обчислень. Вважатимемо, що маємо неоднорідну модель, яка ближча до потреб криптографії: необхідно, щоб супротивник не міг знайти ефективний алгоритм для кожного окремого достатньо великого n, тоді як відсутність у супротивника ефективних алгоритмів в однорідній моделі гарантує лише високу обчислювальну складність переходу від n до $n+1$.

Для забезпечення стійкості криптографічних схем, побудованих на основі односторонніх функцій, необхідно, щоб останні було важко інвертувати майже завжди. Тобто, потрібно використовувати міру складності – складність "у середньому".

Неформально таку односторонню функцію можна визначити такими умовами:

– функцію можливо ефективно обчислити;

– будь-який ефективний алгоритм інвертує цю функцію лише для дуже малої частини значень.

За теорією складності односторонні функції, що важко інвертувати, називають ***криптографічними***.

Нехай $\Sigma = \{0,1\}^*$, $\Sigma^n = \{0,1\}^n$. Вважаємо, що множина Σ є областю визначення всіх функцій, які розглядатимуть. Довжину слова x позначимо через $|x|$. Для деякої події α $P_r(\alpha)$ – це ймовірність цієї події.

Зазначимо, що функцію можливо важко інвертувати лише тому, що вона дуже сильно "стискає" вхідні значення. Жоден поліномний алгоритм не зможе її інвертувати, оскільки протягом поліномного (від довжини значення функції) часу він не встигне виписати знайдене значення з прообразу. Цей випадок виключають за допомогою вимоги чесності.

Означення 1. Функцію f називають чесною, якщо існує поліном p такий, що $|x| < p((|f(x)|)$ для будь-якого $x \in \Sigma$.

Означення 2. Чесну функцію f називають односторонньою в сильному значенні, якщо:

– існує поліномна машина Тюрінга T, яка за будь-яким значенням на вході $x \in \Sigma$ обчислює $f(x)$;

– для будь-якого поліномного алгоритму A справедливе наступне твердження.

Твердження. Нехай $x \in \Sigma^n$ і слово $y = f(x)$ подано на вхід алгоритму A. Тоді для будь-якого полінома p і для всіх достатньо великих n

$$P_r(f(A(y)) = y) < 1/p(n).$$

Імовірність беруть за вибором значення x з Σ^n, і якщо A – це ймовірнісна машина Тюрінга, то за виробленими нею випадковими величинами.

Якщо в означенні 2 обмеження на ймовірність замінити слабшою умовою:

$$P_r(f(A(y)) = y) < 1 - 1/p(n)$$

для деякого фіксованого полінома p, то отримаємо визначення функції, односторонньої в слабкому значенні. Таку функцію важко інвертувати принаймні на поліномній частині значень. Надалі функції, які односторонні в сильному значенні, називаємо ***односторонніми***.

Гіпотеза про існування односторонніх функцій. У теоретичній криптографії розглядають два типи стійкості криптографічних схем. Теоретико-інформаційна стійкість означає, що супротивник, який атакує схему, не отримує достатньої інформації для того, щоб загрожувати безпеці її використання. У випадку, якщо задача супротивника може бути вирішеною, але є, ймовірно, обчислювально складною, то говорять про теоретико-складну стійкість. Усі криптографічні схеми з відкритим ключем можуть бути стійкими лише в теоретико-складному значенні.

Якщо під стійкістю криптографічної схеми розуміти відсутність у супротивника ефективного алгоритму здійснення такої загрози (на основі такої атаки), то з припущення про існування таких схем випливає існування односторонніх функцій. Це показують криптографічні схеми з відкритим ключем. Одним з компонентів останніх (наприклад, криптосистем із відкритим ключем або схем електронного підпису) є алгоритм генерації ключів G. На вході $r \in \Sigma^n$ алгоритм G обчислює пару $(x, y) \in \Sigma \times \Sigma$, де x – секретний ключ, а y – відкритий ключ. Тоді функція $f : \Sigma \to \Sigma$, де $f(r) = y$, якщо $G(r) = (x, y)$ для деякого $x \in \Sigma$, має бути односторонньою. Інакше алгоритм, який інвертує функцію f, можна використовувати для обчислення значення r' з прообразу y. Але $G(r') = (x', y)$ для деякого x'. Отже, супротивник може знайти таємний ключ x' (що не обов'язково збігається з x), який відповідає відкритому ключу y.

2.4.5. Функції з секретом

Функції з секретом (trapdoor functions) використовують у більшості криптосистем з відкритим ключем, схемах електронного підпису і для побудови різних криптографічних протоколів.

Неформально функція з секретом – це одностороння, а саме така, яку важко інвертувати, і яка містить деякий секрет, знання якого дає змогу виконувати операцію інвертування ефективно. Прикладом функції з секретом є функція RSA. Для її інвертування потрібен алгоритм обчислення коренів за складеним модулем, що вважають обчислювально складною задачею. Проте, знання розкладу модуля на прості множники дає можливість розв'язувати цю задачу ефективно. Супротивник, який не знає секрету, може спробувати знайти його, розв'язуючи задачу факторизації. Проте останню також вважають обчислювально складною.

Означення. Нехай G – поліномна ймовірнісна машина Тюрінга, а алгоритм обчислення функції E та алгоритм інвертування функції I – поліномні машини Тюрінга. Трійку (G,E,I) називають генератором функцій із секретом, якщо:

– для довільного $n>0$ на вході 1^n (рядок з n двійкових одиниць) машина G видає пару n-бітових рядків (f,g) (f – параметр, що вибирає функцію з сімейства; g – секрет);

– для довільної такої пари (f,g) і для будь-якого $x\in\Sigma^n$

$$E(f,I(g,E(f,x)))=E(f,x);$$

– для будь-якого поліномного алгоритму A, будь-якого полінома p і всіх достатньо великих n

$$P_r(E(f,A(1^n,f,y))=E(f,x))<1/p(n),$$

де $y=E(f,x)$ для $x\in\Sigma^n$, 1^n – рядок з n двійкових одиниць.

Імовірність беруть за вибором пари (f,g) алгоритмом G, за вибором x і за випадковими величинами алгоритму A, якщо останній є ймовірнісною машиною Тюрінга.

Псевдовипадкові генератори. Псевдовипадкові генератори в криптографії використовують переважно для побудови потокових криптосистем з секретним ключем (шифраторів гамування).

Послідовність є псевдовипадковою, якщо вона проходить всі статистичні тести, які можуть бути виконані протягом поліномного часу. Інакше кажучи, послідовність називають ***псевдовипадковою***, якщо будь-який поліномний

алгоритм може відрізнити її від випадкової послідовності лише з нехтувально малою ймовірністю.

Означення. Генератором називають поліномний алгоритм (машина Тюрінга) g, який на вході s довжини n видає послідовність $g(s)$ довжини $q(n)$. Тут q – деякий поліном, s і $g(s)$ розглядають як двійкові послідовності.

Для довільного поліномного алгоритму A, який на всіх входах видає або 0, або 1, нехай $p_1 = P_r(A(x)=1)$, $p_2 = P_r(A(g(s))=1)$, де x – рядок, випадково вибраний зі всіх рядків довжини $q(n)$, а s – рядок, що випадково вибраний зі всіх рядків довжини n. Генератор g називають ***псевдовипадковим***, якщо для будь-якого поліномного алгоритму A, для будь-якого полінома p і для всіх достатньо великих n

$$|p_1 - p_2| < 1/p(n).$$

В означенні вважають, що $q(n) < n$. На практиці інтерес становлять генератори, які отримують на вході випадкове джерело s завдовжки, наприклад, декілька кілобітів і "розтягують" його в псевдовипадкову послідовність, довжина якої перевищує довжину s на декілька порядків.

Псевдовипадковий генератор можна означити ще так. Вибираємо довільний поліномний алгоритм A із двійковим виходом. Потім виконуємо два експерименти. Під час першого вибираємо випадкове джерело s, даємо генератору g і отриману послідовність $g(s)$ подаємо на вхід алгоритму A. Під час другого експерименту вибираємо випадкову послідовність такої самої довжини, як $g(s)$, і подаємо на вхід A. Результатами цих експериментів є ймовірність появи 1 на виході A. Генератор називають псевдовипадковим, якщо різниця цієї ймовірності є нехтівно малою величиною.

У наведеному означенні накладають дуже сильні умови на послідовності, які видає псевдовипадковий генератор. Ослабити вимогу невідрізняльності псевдовипадкової послідовності від випадкової будь-яким поліномним алгоритмом і розглядати тільки алгоритми з деякого вужчого класу не можна.

Розглянемо один з тестів, що називають ***тестом наступного біта***. Говорять, що послідовність не проходить тест наступного біта, якщо існує поліномний алгоритм, який за початковим відрізком послідовності передбачає наступний біт з ймовірністю, не меншою за $1/2+1/p(n)$. Тут p – деякий поліном, n – довжина джерела s. Тест наступного біта є загальноприйнятим критерієм якості псевдовипадкових послідовностей у криптографії. ***Ендрю Цічжи Яо*** (Yao A.C.) довів, що генератор є псевдовипадковим тоді й тільки тоді, коли породжена ним послідовність проходить тест наступного біта. Інакше

кажучи, довільний ефективний алгоритм, що відрізняє псевдовипадкову послідовність від випадкової, можна перетворити на ефективний алгоритм прогнозу наступного біта в цій псевдовипадковій послідовності.

2.5. Алгоритми виконання операцій із довгими числами

Для стійкості несиметричних алгоритмів як елементи криптографічних алгоритмів використовують достатньо великі числа, оскільки шифрування в групі точок еліптичних кривих бітова довжина чисел має бути не меншою за 160 бітів (≈45 десяткових цифр), при цьому для отримання аналогічної криптографічної стійкості з використанням алгоритму RSA вже потрібно використовувати не менше за 2 кбітів (≈ 564 десяткових цифри).

2.5.1. Розміщення в пам'яті комп'ютера довгих чисел та аналіз типів даних для виконання арифметичних операцій з ними

Арифметичні дії, що виконують за допомогою комп'ютера в обмеженій кількості розрядів, не завжди дають змогу отримати точний результат. Також існують обмеження на розмір (величину) чисел, з якими можна працювати. Прикладом може бути виконання дій із дуже великими числами, наприклад, 30!=265252859812191058636308480000000.

У таких випадках необхідно певним чином представити числа в комп'ютері та точно виконати арифметичні операції з ними.

Число 30!=265252859812191058636308480000000

можна подати у вигляді:

$30!=2\cdot(10^4)^8+6525\cdot(10^4)^7+2859\cdot(10^4)+8121\cdot(10^4)^5+9105\times$

$\times(10^4)^4+8636\cdot(10^4)^3+3084\cdot(10^4)^2+8000\cdot(10^4)^1+0000\cdot(10^4)^0$.

Це представлення записують у вигляді масиву (табл. 2.3). Можна вважати, що наведене "довге" число представлене в 10000 – 10-й системі числення, а "цифрами" числа є чотиризначні числа.

Таблиця 2.3

Представлення довгого числа в 10000 – 10-й системі числення

Номер елемента в масиві *A*	0	1	2	3	4	5	6	7	8	9
Значення	9	0	8000	3084	8636	9105	8121	2859	6525	2

Числа, для представлення яких у стандартних комп'ютерних типах даних не вистачає кількості двійкових розрядів, називають ***довгими***, а алгоритми виконання арифметичних операцій із довгими числами – ***довгою арифметикою***.

Алгоритми роботи з довгими числами залежать від представлення користувачем цих чисел у комп'ютері. Довге число можна записати, наприклад, за допомогою масиву десяткових цифр. Кількість елементів такого масиву дорівнює кількості значущих цифр у довгому числі. Проте, якщо виконати арифметичні операції з цим числом, то розмір масиву має бути достатнім, щоб розмістити в ньому і результат, наприклад, множення.

У десятковій та інших позиційних системах числення декілька записаних поруч цифр формують число. Множина можливих значень кожної цифри обмежена, проте завдяки позиційній вазі, яка залежить від положення цифри, за допомогою короткого запису представляють достатньо великі числа. Цей алгоритм можна використати для побудови довгих чисел.

Якщо взяти масив звичайних цілих і вважати його позиційним записом довгого числа у системі числення з деякою основою, наприклад B, то кожен елемент масиву набуває значення в діапазоні від 0 до $B-1$, а N таких елементів дадуть змогу представити числа від 0 до B^N-1.

Наступним кроком алгоритму є виділення місця для запису довгого числа. Насамперед потрібно визначитися із типом запису довгого числа в масив, а саме, як записати довге число в масив: з початку чи з кінця масиву, з початку чи з кінця числа. Варіанти розміщення числа одночасно не з кінця і не з початку масиву не розглядаємо. Наприклад, для $N=6$, $B=10$ розмістимо число 2000. Можливі чотири варіанти (табл. 2.4):

Таблиця 2.4

Варіанти розміщення числа 2000 у виділених 6-ти комірках

		2	0	0	0
		0	0	0	2
2	0	0	0		
0	0	0	2		

Ще однією проблемою є заповнення невикористаних розрядів. Під час кодування цих розрядів, переважно, використовують один з таких підходів:

1) кожному довгому числу відповідає змінна цілого типу – лічильник, що показує, скільки елементів масиву реально використано;

2) невикористані розряди заповнюють значенням, яке наперед не може зустрітися в числі, наприклад, –1;

3) невикористані розряди заповнюють нулями і обробляють так само, як і ті, що використовують.

Останній підхід можливий лише для першого та четвертого варіантів розміщення довгого числа.

Під час написання програм на основі алгоритмів роботи з довгими цілими числами необхідно враховувати виконання операцій з цілими числами різними мовами програмування.

До переліку допустимих операцій для цілих чисел входять 5 арифметичних операцій, 6 порівнянь і присвоєння.

Включення присвоєння до переліку операцій використовують у мові "С", проте в мовах Паскаль та Бейсік чинять інакше. Операції порівняння для цілих чисел – це порівняння на рівність, нерівність, менше, більше, менше дорівнює, більше дорівнює. Синтаксис операцій порівняння в різних мовах може відрізнятися, наприклад, "не дорівнює" в мові Паскаль записують "<>", у "С" – "!=", у мові Фортран – ".NE.", проте зміст від запису не залежить. Порівняння – це операція, яка так само, як і арифметичні операції, має операнди; лише результатом порівняння є не цілий, а логічний тип.

Під час виконання арифметичних операцій підсумовування, віднімання і множення в різних мовах проблем не виникає, за винятком ситуації переповнення.

Для операції ділення звичайне математичне ділення двох цілих може дати дробовий результат. Для того, щоб залишитися в діапазоні множини цілих чисел, потрібно використати цілочислове ділення, а саме вважати часткою цілу частину результату, а дробову відкидати.

У багатьох мовах цілочислове ділення є окремою операцією, для якої існує спеціальне позначення. Наприклад, в багатьох версіях мови Бейсік для цього використовують зворотну похилу лінію, а в Паскалі – службове слово **div.** Запис операції цілочисельного ділення за допомогою похилої лінії в мові Паскаль недопустимий, оскільки він дає результат дійсного типу.

У мові "С" використовують інший підхід. Спеціальної операції цілочислового ділення тут немає, проте звичайне ділення виконують як цілочислове, якщо обидва операнди цілого типу.

Для операції остачі цілочислового ділення в різних мовах також існують свої позначення, наприклад, у мовах Бейсік та Паскаль – службове слово **mod**, а в мові "С" – знак %.

2.5.2. Здійснення алгоритму множення довгого числа на коротке

Для виконання операції множення довгих чисел можна скористатися моделюванням стовпчика, виконуючи на кожному кроці перенесення в наступний розряд.

Коротке число є беззнаковим цілим (unsigned int). Блок-схему цього алгоритму наведено на рис. 2.7.

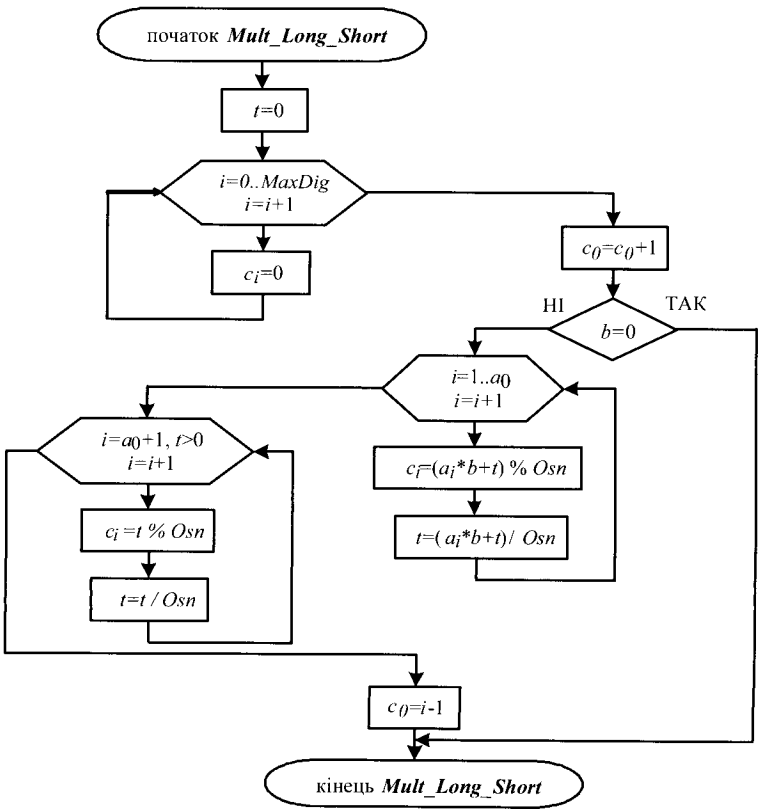


Рис. 2.7. Алгоритм множення довгого числа на коротке

Використовують додаткову змінну t для визначення кількості розрядів результату і формування перенесення в наступний розряд. Довге число записано в масиві a, а коротке – у змінній b.

Насамперед значення результату обнулюють. Потім перевіряють, чи короткий множник дорівнює 0. Якщо так, то результатом буде нуль. Інакше здійснюють формування розрядів результату множення, які дорівнюють остачі від ділення на основу системи числення добутку розряду довгого множника на короткий множник плюс розряд перенесення.

Значення перенесення в наступний розряд результату є цілою частиною від ділення добутку $a_i \cdot b + t$ на основу системи числення.

На останньому кроці формують значення кількості розрядів результату, старші розряди результату – цілочисловим діленням та остачі цілочислового ділення значення перенесення t на основу системи числення.

2.5.3. Множення довгих чисел із використанням стовпчика

Алгоритм множення довгого числа на довге із використанням стовпчика (рис. 2.8), подібний до описаного вище алгоритму множення за винятком того, що під час формування розрядів результату перемножують відповідні розряди довгих чисел a_i та b_i.

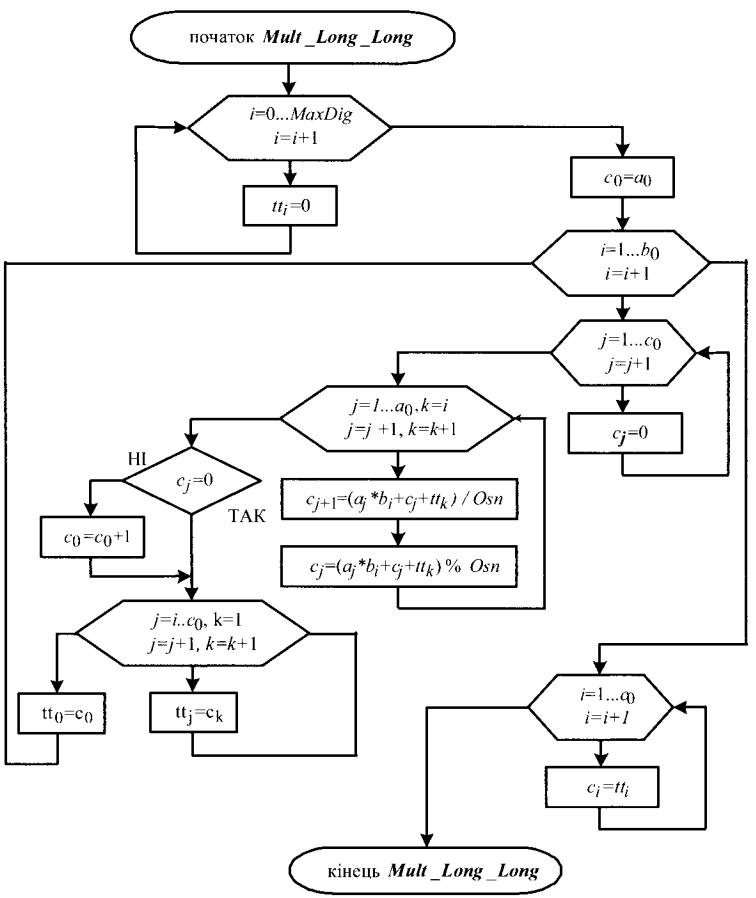


Рис. 2.8. Алгоритм множення довгих чисел способом стовпчика

Ще однією відмінністю розробленого алгоритму множення довгих чисел є додавання до кожного розряду результату множення значення проміжного множення на окремий розряд tt_i.

2.5.4. Алгоритм швидкого множення

Маємо два довгі числа в канонічній формі:

$$x = \sum_{i=0}^{m} x_i B^i \ , \ y = \sum_{i=0}^{m} y_i B^i \ .$$

Алгоритм множення стовпчиком працює дуже повільно.

Набагато ефективніший алгоритм – алгоритм "швидкого множення" і він дає змогу замінити множення натуральних чисел невеликою кількістю підсумовувань.

Алгоритм швидкого множення полягає у використанні співвідношень:

якщо b – парне, то $a \cdot b = (2a) \cdot (b/2)$,

якщо b – непарне, то $a \cdot b = a + a \cdot (b-1)$.

Блок-схему цього алгоритму наведено на рис. 2.9.

Після завершення роботи алгоритму b дорівнюватиме нулю, а p – добутку.

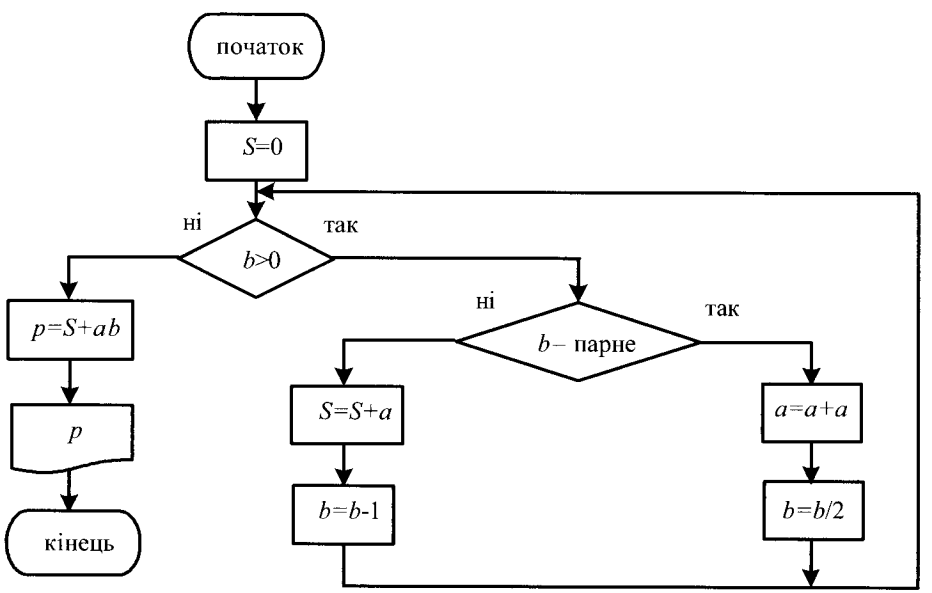


Рис. 2.9. Алгоритм швидкого множення довгих чисел

2.5.5. Множення з використанням швидкого перетворення Фур'є

Для множення дуже довгих чисел використовують алгоритми, що працюють за допомогою згортки Фур'є:

$$c_i = \sum_{k+l=i} a_k b_l$$

з подальшим перетворенням. Підсумовування здійснюють за k-ми та l-ми номерами відповідних розрядів чисел a та b.

У коефіцієнтів згортки є практичний зміст – вони дають результат множення многочлена $a_0 + a_1 x^1 + ... + a_{n-1} x^{n-1}$ на многочлен $b_0 + b_1 x^1 + ... + b_{m-1} x^{m-1}$, де степені m та n – довільні натуральні числа.

$$(a_0 + a_1 x^1 + ... + a_{n-1} x^{n-1})(b_0 + b_1 x^1 + ... + a_{bn-1} x^{n-1}) = c_0 + c_1 x^1 + ... c_{n+m-1} x^{n+m-1}.$$

Числа в записі за основою Osn є многочленами, де x – це основа, тому згортка може бути проінтерпретована як результат множення числа $A = a_0 + a_1 Osn^1 + ... + a_{n-1} Osn^{n-1}$ на число $B = b_0 + b_1 Osn^1 + ... + b_{m-1} Osn^{m-1}$ без обчислення перенесень:

$$A \oplus B = c_0 + c_1 Osn^1 + ... + c_{n+m-1} Osn^{n+m-1}.$$

Конструкцію, яку утворюють коефіцієнти c_i, називають пірамідою множення, оскільки довжина виразу для коефіцієнтів спочатку зростає, досягаючи максимуму в середині, а потім спадає, перетворюючись в кінці на нуль:

$$c_0 = a_0 \cdot b_0;$$
$$c_1 = a_0 \cdot b_1 + a_1 \cdot b_0;$$
$$c_2 = a_0 \cdot b_2 + a_1 \cdot b_1 + a_2 \cdot b_0;$$
$$\ldots$$
$$c_i = \sum_{k+l=i} a_k b_l;$$
$$\ldots$$
$$c_{n+m-3} = a_{n-1} \cdot b_{m-2} + a_{n-2} \cdot b_{m-1};$$
$$c_{n+m-2} = a_{n-1} \cdot b_{m-1};$$
$$c_{n+m-1} = 0.$$

У цьому випадку базовий тип, що використовують, повинен мати достатній об'єм для зберігання коефіцієнтів порядку $(n+m-1) \cdot Osn^2$, які знаходяться на вершині піраміди.

Отже, для множення довгих чисел достатньо:

1) обчислити коефіцієнти згортки c_i, $i = 0...n = m-1$;

2) зробити необхідні перенесення розрядів, тобто згрупувати коефіцієнти за Osn .

Усі перенесення можна обчислити протягом $O(n+m)$ кроків, переходячи від молодших розрядів до старших. Блок-схема фрагменту алгоритму кроку 2 зображена на рис. 2.10.

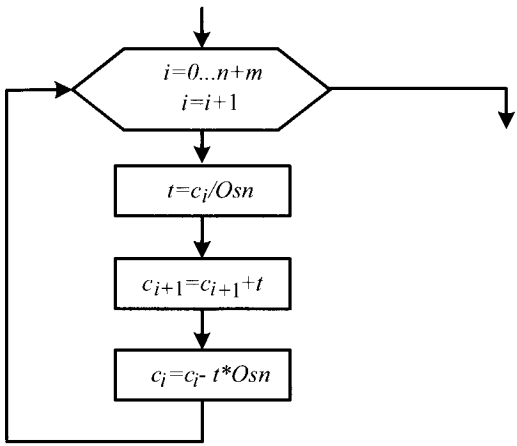


Рис. 2.10. Фрагмент алгоритму перенесення в наступний розряд

Отже, основна складність алгоритму множення полягає в обчисленні коефіцієнтів згортки на кроці 1. Для цього можна використати швидке перетворення Фур'є (ШПФ) і швидке перетворення Хартлі (ШПХ).

ШПФ комплексного вектора $(a_0, a_1, ..., a_{N-1})$ обчислюють як комплексний вектор з координатами $(y_0, y_1, ..., y_{N-1})$:

$$y_k = \sum_{j=0}^{N-1} a_j \omega^{kj},$$

де ω – комплексний корінь N -го степеня з одиниці, тобто

$$\omega = \omega_N = e^{\frac{2\pi}{N} i} = \cos\frac{2\pi}{N} + i\sin\frac{2\pi}{N}.$$

Індекс степеня N може бути відсутній, тоді степінь кореня дорівнює кількості координат вектора, що перетворюють.

До дискретного перетворення Фур'є існує обернене, яке обчислюють за формулою

$$a_k = \frac{1}{N}\sum_{j=0}^{N-1} y_j \omega^{-kj}.$$

2.5.6. Застосування швидкого перетворення Фур'є для обчислення згортки a⊗b

У ШПФ використовують теорему про згортку: перетворення Фур'є від згортки двох векторів є скалярним добутком Фур'є-образів цих векторів:

$$c = a \otimes b \Leftrightarrow ШПФ(c) = ШПФ(a) * ШПФ(b) ,$$

звідки $c = ШПФ^{-1}(ШПФ(a) * ШПФ(b))$.

Отже, алгоритм обчислення згортки складається з трьох кроків:

1) обчислити ШПФ(a) і ШПФ(b);
2) скалярно перемножити отримані вектори;
3) обчислити зворотне перетворення Фур'є від скалярного добутку.

Перемноження многочленів зводиться до скалярного перемноження відповідних векторів.

Вважаємо, що на кожному кроці розміри векторів однакові й дорівнюють N, оскільки не можна скалярно перемножити короткий вектор на довший, тому загальний час має порядок $O(N \log N)$. Найвищої швидкодії досягають за варіантами ШПФ, що працюють з векторами розміру 2^k, тому вектори за необхідності потрібно доповнити нулями.

Як приклад візьмемо $A = (3,4)$, $B = (5,4,3,2,1)$. Числа зберігають задом наперед, а саме старший розряд йде останнім. Тоді алгоритм швидкого множення виглядає так:

1) знайти найменше число Len – степінь двійки: $Len \geq a_0 + b_0$. Для розглянутих чисел $Len = 8$;
2) доповнити A і B нулями до Len: $A = (3,4,0,0,0,0,0,0)$, $B = (5,4,3,2,1,0,0,0)$;
3) обчислити ШПФ дійсних векторів на обидвох масивах цифр;
4) скалярно перемножити перетворені вектори, отримавши вектор розміру Len;
5) застосувати зворотне перетворення Фур'є, результатом якого буде згортка;
6) перетворити згортку на масив цілих чисел, зробити перенесення.

Цифри для довгих чисел зберігають у цілочисловому форматі. Тому для ШПФ їх необхідно скопіювати в тимчасові масиви типу з плаваючою точкою. Якщо необхідно отримувати результат максимальної довжини N, то необхідно виділити для довгих чисел пам'ять розміру не менше $MaxLen = 2^k$, де $MaxLen$ – мінімальний степінь двійки, який більший за N. Наприклад, якщо

максимальний результат складатиметься із 1000 цифр за основою Osn, то мінімальний об'єм пам'яті $MaxLen = 1024$, оскільки будемо рахувати ШПФ вектора саме такої довжини.

Усі обчислення виконують у форматі з плаваючою крапкою і використовують ірраціональні числа, тому результат буде не набором цілих чисел, а наближеним до нього. Для отримання результату кожну координату вектора необхідно округлити до найближчого цілого числа.

Проблема полягає в тому, що якщо точність обчислень недостатньо висока, то округлення можливо здійснювати не до потрібного числа. Тоді алгоритм завершиться, проте результат буде неправильний. Оскільки арифметичні операції з дійсними числами не можливо здійснювати абсолютно точно, то розмір використаного типу даних повинен бути доволі великим, щоб помилка на кожному знаку була меншою за 0,5.

Наприклад, з використанням типу даних розміру один дріб 1/3 буде представлений у вигляді 0,3. Під час додавання трьох дробів одержимо 1/3+1/3+1/3=(у форматі для чисел з плаваючою крапкою) =0,9.

Якщо взяти дві цифри, то 1/3=0,33 і 0,33+0,33+0,33=0,99. При цьому точність обчислень значно зросла.

Взагалі кажучи, є два шляхи підвищення точності обчислень, один з яких пов'язаний зі збільшенням довжини типу, що використовують, – наприклад, у мові C від float до double, від double до double^2 тощо. Другий підхід гнучкіший і передбачає зменшення довжини основи Osn. Отже, число стане довшим, займатиме більше пам'яті, але завдяки цьому підвищиться точність.

2.5.7. Обмеження швидкого перетворення Фур'є множення

Нехай потрібно перемножити вектори з 2^n координат з використанням ШПФ для векторів з дійсними координатами. Тоді похибку множення err оцінюють зверху виразом

$$err < 2^n Osn^2 (\varepsilon \cdot 3n + 3\sqrt{5}(3n+4) + \beta(3n+3)),$$

де ε – точність додавання (віднімання); β – точність тригонометричних обчислень. За цією формулою можна знайти мінімально можливе значення основи Osn. Наприклад, для типу *double* (53 біти) $\varepsilon = 2^{-53}$. Похибки тригонометрії обмежено величиною $\beta = \varepsilon / 2$.

Обмежимо верхню межу похибок числом 1/2. Приблизно порахувавши константи, отримаємо $2^n Osn^2 2^{-53} \cdot (11{,}83n + 11{,}07) < 1/2$. Для чисел завдовжки 2^{20} отримаємо значення основи $Osn < 4100$. На практиці, якщо вибрати основу

10000, то множення на основі ШПФ працюватиме для набагато більших довгих чисел. Під час округлення необхідно враховувати різницю між округленим значенням і результатом округлення. Якщо вона менша за 0,2, то множення швидше за все дає правильний результат, якщо більша, то рекомендують зменшити основу або скористатися іншим базовим типом для зберігання коефіцієнтів.

Після виконання п'ятого кроку немає готового добутку, а є лише згортка – результат без перенесень. Як було зазначено під час розгляду піраміди множення, значення коефіцієнтів згортки можуть бути набагато більшими за основу, досягаючи максимального значення $2NOsn^2$. Якщо згадати, що під час зворотного перетворення Фур'є виконують ділення результатів на N, то максимальний розмір цифри дорівнює $2NOsn^2$, тому для запобігання переповненню основу системи числення слід вибирати не більшою за чотири десяткові цифри.

Отже, існує три проблеми виконання операції множення:

1. Точність тригонометрії.
2. Точність під час обчислення ШПФ.
3. Переповнення базового типу.

Другу і третю проблеми вирішують зменшенням основи системи числення або збільшенням базового типу. При цьому ефективність алгоритму спадає, оскільки менша основа означає збільшення кількості цифр, а більший базовий тип не завжди доступний.

На останньому кроці необхідно перетворити згортку на довге число.

2.5.8. Використання швидкого перетворення Хартлі для обчислення згортки

Множення використовує дійсні числа. Кращий результат може забезпечити перетворення Хартлі. Алгоритм швидкого множення можна вдосконалити на основі цього перетворення.

Для коефіцієнтів дискретного перетворення Хартлі $ДПХ(a \otimes b)$ теорема про згортку має вигляд:

$$ДПХ(a \otimes b)_k = 1/2(c_k(d_k + d_{N-k}) + c_{N-k}(d_k - d_{N-k})), \qquad (2.5.1)$$

де $c = ДПХ(a)$, $d = ДПХ(b)$, $k = 0...N-1$.

Індекси обчислюють за модулем N, тобто замість елементів з індексом N потрібно брати елементи з нульовим індексом. Усі кроки попереднього алгоритму залишаються без змін за винятком кроку 4, де координати вектора обчислюють за формулою (2.5.1).

Завдяки двом стилям *ШПХ* та ШПФ (розбиття за частотою (*ШПХ_ЧТ*) і часом (*ШПХ_ЧС*)) дає змогу уникнути виклику підпрограми перестановки в алгоритмі *ШПФ*, виконання якої займає приблизно 10 % часу виконання алгоритму множення. *ШПХ_ЧС* має на вході перетворений вектор, а в кінці роботи повертає звичайний, а *ШПХ_ЧТ* – навпаки.

Тому кроки попередньо розглянутого алгоритму можна змінити так:

1) вектори зберігають зі звичайним порядком індексів. Обчислюють *ШПХ_ЧТ* у двох масивах цифр;

2) вектор на виході в перетвореному вигляді. Обчислюють коефіцієнти $ШПХ(a \otimes b)$ за формулою (2.5.1), враховуючи зміни в порядку індексів;

3) вектор ще у перетвореному вигляді. Обчислюють ШПХ_ЧС, результатом якого буде ненормалізована згортка.

Остаточний вектор має звичайний порядок індексів.

Розглянемо як приклад, обчислення у псевдокоді $ШПХ(a \otimes b)$ для $a = ШПХ\_ЧТ(a_1, a_2, ...)$, $b = ШПХ\_ЧТ(b_1, b_2, ...)$ замість вектора b. Елементи з індексами 0 і $N/2$ обробляють окремо, оскільки для них формула набуває вигляду $a_0 = a_0 b_0$, $a_{n/2} = a_{n/2} b_{n/2}$. Тут знак "=" позначає операцію присвоєння. Для інших елементів необхідно врахувати, що на вхід формули числа подають із двох позицій, а вихід записують в одну. Щоб уникнути затирання, можна обчислювати по два значення водночас, використовуючи симетричність виразу.

$$a_k = 1/2(a_k(b_k + b_{N-k}) + a_{N-k}(b_k - b_{N-k})),$$
$$a_{N-k} = 1/2(a_{N-k}(b_{N-k} + b_k) - a_k(b_k - b_{N-k})).$$

Елементи вектора a із сумою індексів N перетворюватимуть попарно і на одному місці. Утворюють так званий "метелик згортки", виконання якого для $k = 1...n/2 - 1$ дасть необхідний результат.

Якщо порядок індексів звичайний, то обчислення виконують безпосередньо за формулами. Для 8-елементних векторів схему обчислень зображено на рис. 2.11.

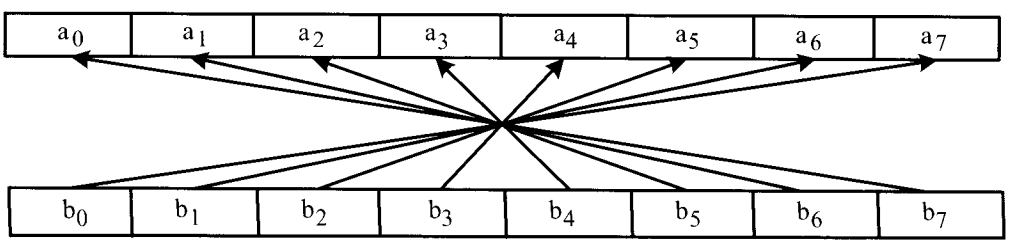


Рис. 2.11. Обчислення ДПХ згортки через ДПХ векторів. Індекси в звичайному порядку

Щоб врахувати бітовий порядок, для перетворених векторів можна виконати метелики спочатку для 2 елементів, потім для 4, для 8 тощо з кроком, помноженим на 2 (рис. 2.12).

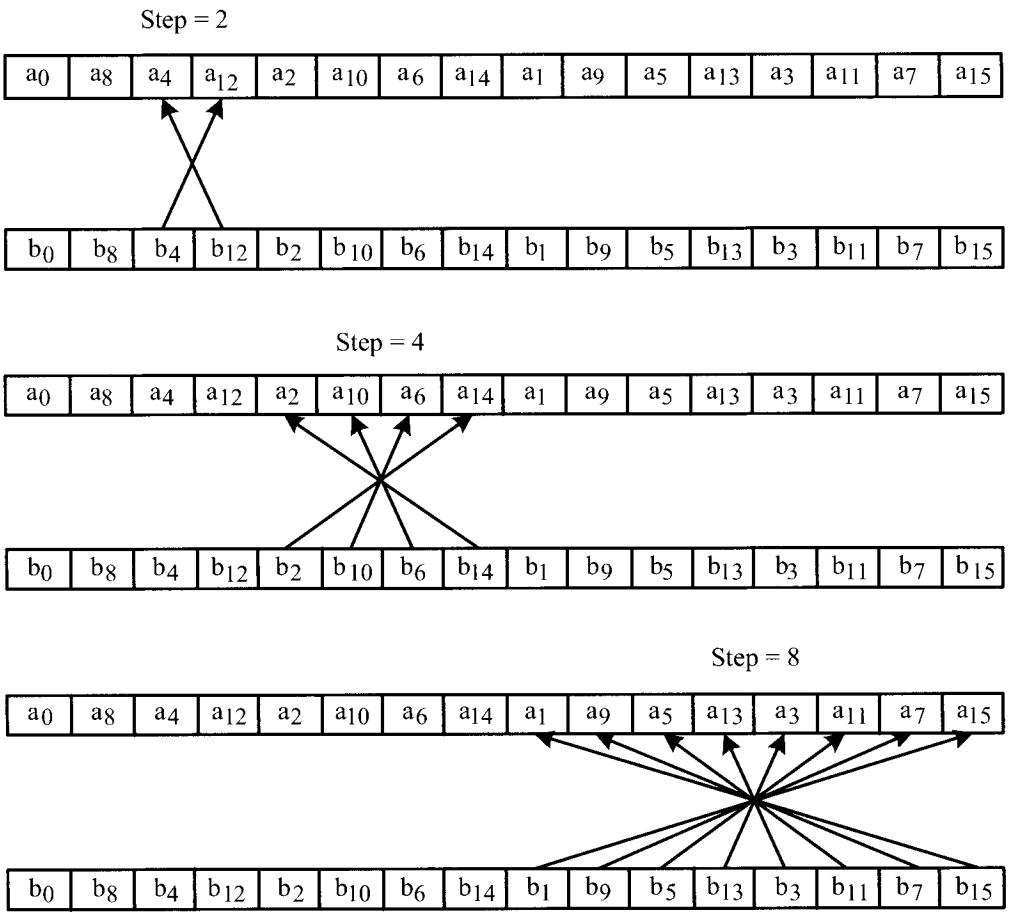


Рис. 2.12. Обчислення ДПХ згортки через ДПХ векторів. Індекси в зворотному бітовому порядку. N=16

Обчислення пари коефіцієнтів *ДПХ* згортки потребує 4 додавань і 4 множень, а для перетворення Фур'є кількість операцій становить 2 комплексні множення (8 звичайних) і 4 додавання. Проте комплексний вектор удвічі коротший, тому *ДПХ* потребує на 2 додавання більше для пари елементів.

2.5.9. Порівняльна характеристика алгоритмів множення довгих чисел

Перемножимо два числа однакового розміру. Залежності часу виконання операції множення від довжини числа для алгоритмів на основі алгоритму множення стовпчиком (пунктирна лінія) та алгоритму множення з використанням ШПХ (суцільна лінія) зображено на рис. 2.13.

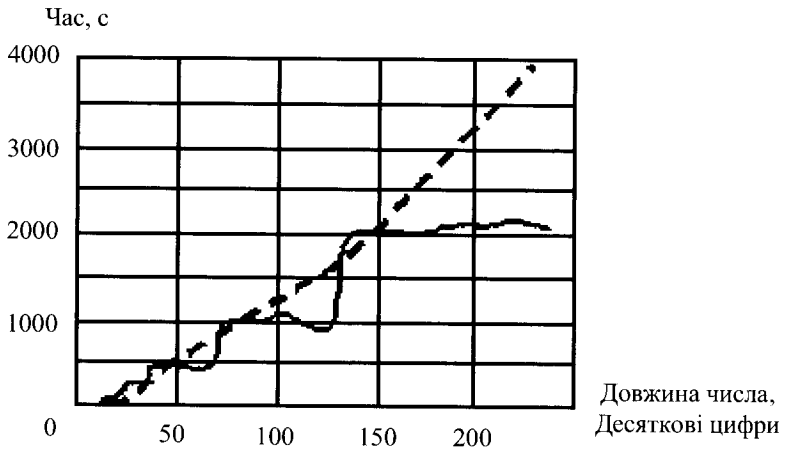


Рис. 2.13. Порівняльна характеристика алгоритмів множення на основі стовпчика та ШПХ

Як видно з рис. 2.13, скачок в алгоритмі на основі *ШПХ* виникає під час переходу через степінь двійки. Наприклад, при перемноженні 128-значних чисел обчислюють *ШПХ* для 128-координатних векторів, а для 129-значних векторів необхідно працювати з 256-значними векторами, оскільки зростає розрядність.

Починаючи зі 150-розрядних десяткових цифр, алгоритм множення на основі ШПХ працює значно швидше.

Узагалі кажучи, стандартні алгоритми додавання, віднімання, множення та ділення на мале ціле число мають складність $O(n)$. Множення потребує складності $O(n^2)$ для двох чисел розмірності n, що значно сповільнює алгоритми.

Для прискорення процесу множення довгих чисел використовують ***метод Карацуби***. Асимптотично він є гірший за множення на основі швидкого перетворення Фур'є, проте дає змогу значно збільшити швидкодію за допомогою операцій низького рівня. Пам'яті такі алгоритми потребують трохи більше, ніж O (*довжини чисел*).

Завдяки швидкому перетворенню Фур'є можна здійснити множення протягом $O(n\log(n)\log(n))$ операцій, що становить приблизно $O(n\log(n))$, оскільки $\log(\log(n))$ зростає дуже повільно.

Складність алгоритму множення на основі швидкого перетворення Фур'є насправді є більшою за $n\log(n)$. Проаналізуємо складність алгоритмів множення довгих чисел.

Для перемноження двох чисел по N цифр у кожному ці числа записують за основою B, яка містить k цифр ($B = 10^k$). Тому ці числа мають коефіцієнти $n = N / k$. Алгоритм множення на основі швидкого перетворення Фур'є працює протягом $O(n\log(n))$ операцій з числами за основою B. Через похибки обчислень для типів з плаваючою точкою ці числа мають мати точність для представлення цілих чисел до $6n^2B^2\log(n)$. Кількість цифр у довгому числі потребує порядку $\log(B)+\log(n)$. Тоді складність основних операцій з такими числами становитиме $O((\log(B)+\log(n))^2)$, а остаточна складність – $O(n\log(n)(\log(B)+\log(n))^2)$. Оскільки основу B вибрано так, що $k = \log_{10}(B)$ має такий самий порядок, як і $\log(n)$, то складність перемноження двох довгих чисел розміру N становить $O(n\log(n)^3) = O(N\log(N)^2)$.

Найкращу теоретичну верхню межу складності можна отримати завдяки множенню Шенхаге Штрассена [8], для якого процес узагальнюють за допомогою роботи зі скінченними кільцями.

2.6. Елементи теорії еліптичних кривих

Пошук більш ефективних і надійних алгоритмів для напрямленого шифрування спричинив розгляд таких математичних об'єктів як групи точок на еліптичних кривих [9]. Їх застосування в асиметричній криптографії (див. розділ 3) дало можливість істотно зменшити довжину чисел, що використовують як ключі, із збереженням рівня стійкості алгоритму.

2.6.1. Способи побудови еліптичних кривих

Криву третього порядку E, що задають рівнянням вигляду

$$E:\ y^2 = x^3 + ax + b, \qquad (2.6.1)$$

називають ***еліптичною кривою*** (насправді рівняння (2.6.1) отримано заміною змінних з більш загального рівняння, яке нас не цікавитиме).

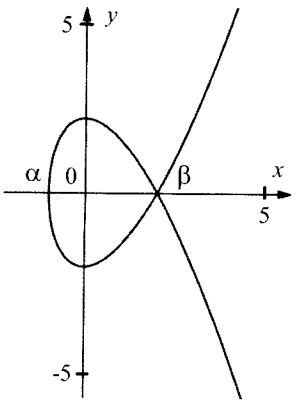


Рис. 2.14. Еліптична крива при $D = 0$

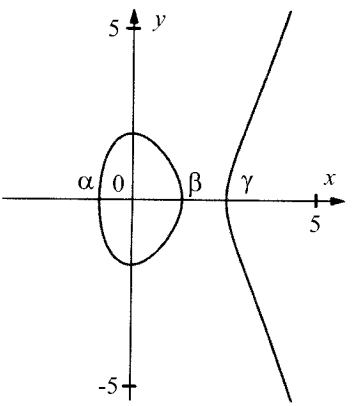


Рис. 2.15. Еліптична крива при $D < 0$

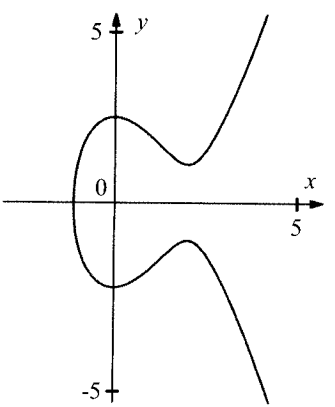


Рис. 2.16. Еліптична крива при $D > 0$

Оскільки $y = \pm\sqrt{x^3 + ax + b}$, графік кривої симетричний щодо осі абсцис. Щоб знайти точки його перетину з віссю абсцис, необхідно розв'язати кубічне рівняння

$$x^3 + ax + b = 0. \qquad (2.6.2)$$

Це можна зробити за допомогою відомих формул Кардано. Дискримінант цього рівняння

$$D = \left(\frac{a}{3}\right)^3 + \left(\frac{b}{2}\right)^2. \qquad (2.6.3)$$

Якщо $D < 0$, то (2.6.2) має три різні дійсні корені α, β, γ; якщо $D = 0$, то (2.6.2) має три дійсні корені, наприклад, α, β, β, принаймні два з яких однакові; нарешті, якщо $D > 0$, рівняння (2.6.2) має один дійсний корінь α і два комплексно спряжені. Криві у всіх трьох випадках зображено на рис. 2.14–2.16.

Криву, представлену на рис. 2.14, називають ***сингулярною***. В її точці сингулярності (β,0) є дві дотичні. Сингулярні криві виключають з розгляду. Тому при заданні кривої за допомогою параметрів a і b вимагатимемо виконання умови $D \neq 0$, що еквівалентно

$$4a^3 + 27b^2 \neq 0. \qquad (2.6.4)$$

2.6.2. Композиція точок еліптичних кривих

Нехай еліптичну криву E задано рівнянням (2.6.1) з обмеженням на коефіцієнти (2.6.4). Визначимо операцію композиції точок на кривій. Візьмемо які-небудь дві точки $P = (x_1, y_1) \in E$, $Q = (x_2, y_2) \in E$ і проведемо через них

пряму (рис. 2.17). Ця пряма обов'язково перетне криву в третій точці, яку позначимо R'. Третя точка обов'язково існує тому, що кубічне рівняння, яке отримують після підстановки рівняння прямої в (2.6.1), має два дійсні корені, відповідні точкам P і Q, отже, його третій корінь, що відповідає R', також дійсний. Точку $R = (x_3, y_3)$ отримаємо зміною знаку ординати точки R'.

Позначимо описану операцію композиції точок: $R = P + Q$.

Нехай точка $P \in E$ має координати (x, y). Тоді точку з координатами $(x, -y)$ позначатимемо $-P$. Вважатимемо, що вертикальна пряма, що проходить через P і $-P$ перетинає криву в нескінченно віддаленій точці O, тобто $P + (-P) = O$. Точка O відіграє роль нуля в операціях на еліптичній кривій.

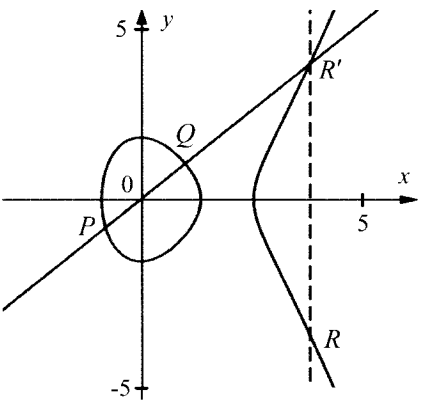


Рис. 2.17. Композиція точок $R = P + Q$

Тепер уявимо, що точки P і Q (рис. 2.17) зближуються і, нарешті, зливаються в одну точку $P = Q = (x_1, y_1)$. Тоді композицію $R = (x_3, y_3) = P + Q = P + P$ буде отримано проведенням дотичної в точці P і віддзеркаленням її другого перетину з кривою R' щодо осі абсцис (див. рис. 2.17). Використовуватимемо таке позначення: $R = P + P = [2]P$.

Виведемо формули для визначення координат результуючої точки $R = (x_3, y_3)$ на основі координат початкових точок $P = (x_1, y_1)$ і $Q = (x_2, y_2)$. Розглянемо спочатку випадок, коли $P \neq \pm Q$, $R = P + Q$ (рис. 2.17). Позначимо через k кутовий коефіцієнт прямої, що проходить через P і Q. Очевидно, що

$$k = \frac{y_2 - y_1}{x_2 - x_1}. \tag{2.6.5}$$

Тоді рівняння прямої матиме вигляд $y - y_1 = k(x - x_1)$, звідки

$$y = y_1 + k(x - x_1). \tag{2.6.6}$$

Підставимо знайдений вираз для змінної Y до рівняння кривої (2.6.1). Отримаємо $(y_1 + k(x - x_1))^2 = x^3 + ax + b$.

Підносячи до квадрата і групуючи подібні члени, отримаємо кубічне рівняння $x^3 - k^2x^2 + \ldots = 0$.

Відомо, що сума коренів кубічного рівняння дорівнює коефіцієнту при x^2, узятому з протилежним знаком (***теорема Вієта*** для кубічних рівнянь), тобто

$$x_1 + x_2 + x_3 = k^2,$$

звідки

$$x_3 = k^2 - x_1 - x_2. \qquad (2.6.7)$$

Підставивши знайдене значення x_3 до рівняння прямої (2.6.6), знайдемо ординату точки R', $y_3' = y_1 + k(x_3 - x_1)$ і, змінивши знак, отримаємо

$$y_3' = k(x_1 - x_3) - y_1. \qquad (2.6.8)$$

Отже, ми знайшли координати точки R.

Тепер розглянемо випадок, коли $P = Q$ і результуюча точка $R = [2]P$ (рис. 2.18). Диференціюючи обидві частини (2.6.1) за x, отримаємо

$$2yy' = 3x^2 + a.$$

Кутовий коефіцієнт дотичної дорівнює значенню похідної в точці P

$$k = \frac{3x_1^2 + a}{2y_1}. \qquad (2.6.9)$$

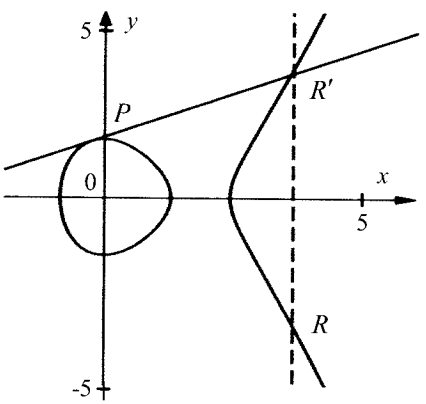


Рис. 2.18. Подвоєння точки $R = P + P = [2]P$

Подальші міркування такі самі, як у першому випадку, і координати точки R визначають за формулами (2.6.7) і (2.6.8). Якщо ордината точки P дорівнює нулю, то дотична проходить паралельно осі ординат і $[2]P = 0$.

Використовуючи отримані формули для обчислення композиції та прийняті угоди щодо точки в нескінченності, можна довести такі властивості точок еліптичної кривої:

1) $P + Q = Q + P$ для всіх точок $P, Q \in E$;

2) $P + (Q + S) = (Q + P) + S$ для всіх точок $P, Q, S \in E$;

3) існує нульовий елемент O (точка в нескінченності) такий, що $P + O = O + P = P$ для всіх $P \in E$;

4) для кожної точки $P \in E$ існує точка $-P \in E$, така, що $P + (-P) = 0$.

Перераховані властивості точок збігаються з властивостями цілих чисел при використанні операції додавання. Тому композицію точок часто називають ***додаванням точок***, а операцію $[2]P$ – ***подвоєнням точки***.

Продовжуючи аналогію із додаванням чисел, зручно ввести такі позначення. Для цілого m

$$[m]P = \underbrace{P + P + ... + P}_{m},$$

$$[0]P = 0,$$

$$[-m]P = -(\underbrace{P + P + ... + P}_{m}).$$

Під час обчислення композиції точок на кривій (формули (2.6.5), (2.6.9), (2.6.7) і (2.6.8)) використовують лише операції додавання, віднімання, множення й ділення чисел. Це означає, що всі наведені вище тотожності збережуться, якщо виконувати обчислення з цілими числами за модулем простого числа p. У цьому випадку додають й множать числа за модулем p, різницю $u - i$, де u і i цілі числа, обчислюють як $u + (p - v) \bmod p$, а ділення u / v виконують множенням u на $v^{-1} \bmod p$ (простота модуля гарантує, що для будь-якого додатного числа $v < p$ існує число v^{-1}, таке, що $vv^{-1} \bmod p = 1$).

У результаті отримуємо криву

$$E: \ y^2 = x^3 + ax + b (\bmod p). \qquad (2.6.10)$$

У рівнянні (2.6.10) змінні x, y і коефіцієнти a, b набувають цілочислових значень, а всі обчислення виконують за модулем p. Відповідно до (2.6.4) на a, b накладають обмеження

$$(4a^3 + 27b^2) \bmod p \neq 0. \qquad (2.6.11)$$

Множина $E_p(a,b)$ складається зі всіх точок (x,y), $0 \le x, y < p$, що задовольняють рівняння (2.6.10), і точки в нескінченності O. Кількість точок у $E_p(a,b)$ позначаємо $\#E_p(a,b)$. Ця величина має важливе значення для криптографічних додатків еліптичних кривих.

Наприклад, розглянемо криву

$$E_7(2,6): \ y^2 = x^3 + 2x + 6 (\bmod 7). \qquad (2.6.12)$$

Перевіримо умову (2.6.11):

$4 \cdot 2^3 + 27 \cdot 6^2 = 4 \cdot 1 + 6 \cdot 1 = 3 \neq 0 \pmod 7$.

Отже, крива несингулярна. Знайдемо яку-небудь (випадкову) точку в $E_7(2,6)$. Нехай $x = 5$. Тоді

$$y^2 = 5^3 + 2 \cdot 5 + 6 = 6 + 3 + 6 = 1 \pmod 7$$

і $y = 1 (\mathrm{mod}\ 7)$ або $y = -1 = 6 (\mathrm{mod}\ 7)$. Ми знайшли відразу дві точки: (5,1) і (5,6). Знайдемо ще пару точок обчисленням композиції. Спочатку знайдемо (5,1) [2]. Використовуючи (2.6.9), (2.6.7) і (2.6.8), обчислюємо

$$k = \frac{3 \cdot 5^2 + 2}{2 \cdot 1} = \frac{0}{2} = 0 (\mathrm{mod}\ 7),$$

$$x_3 = 0 - 2 \cdot 5 = 4 \pmod 7,$$

$$y_3 = 0 \cdot (5 - 4) - 1 = 6 \pmod 7.$$

Ми отримали $[2](5,1) = (4,6)$ (можна переконатися, що отримана точка лежить на кривій, підставивши її координати в рівняння (2.6.12)). Знайдемо ще одну точку $[3](5,1) = (5,1) + (4,6)$. Використовуючи (2.6.5), (2.6.7) і (2.6.8), обчислюємо

$$k = \frac{6 - 1}{4 - 5} = \frac{5}{6} = 5 \cdot 6 = 2 (\mathrm{mod}\ 7),$$

$$x_3 = 2^2 - 5 - 4 = 2 (\mathrm{mod}\ 7),$$

$$y_3 = 2 \cdot (5 - 2) - 1 = 2 \cdot 3 - 1 = 5 \pmod 7.$$

Ми отримали $[3](5,1) = (2,5)$. Отже, ми знайшли чотири точки. Для криптографічного використання кривої важливо знати, скільки загалом точок у множині $E_7(2,6)$.

2.6.3. Властивості множини точок на еліптичній кривій

Існують такі властивості множини точок $E_p(a,b)$. Ця множина скінченна, оскільки до неї входять лише точки з цілочисловими координатами $0 \le x, y < p$. Існує пряма аналогія між $E_p(a,b)$ і множиною степенів цілих чисел, що обчислюють за модулем p. Крім того, $E_p(a,b)$ має генератор, тобто таку точку G, що ряд G, $[2]G$, $[3]G$, ..., $[n]G$, де $n = \#E_p(a,b)$, містить всі точки множини $E_p(a,b)$, причому $[n]G = O$. Кількість точок на кривій за належного вибору параметрів p, a і b може бути простим числом q: $\#E_p(a,b) = q$. У цьому випадку будь-яка точка (окрім O) є генератором всієї множини точок. Така крива краща з багатьох аспектів і завжди може бути знайдена за практично прийнятний час. Якщо з якихось причин таку криву знайти не вдалося, і $\#E_p(a,b) = hq$, де q – знову просте число, то в $E_p(a,b)$ існує підмножина з q точок (тобто потужності q), генератором якої може бути будь-яка точка $G \neq O$, така, що $[n]G = O$.

2.6.4. Криптографічні операції на еліптичній кривій

Основна криптографічна операція на еліптичній кривій – m-кратна композиція, тобто обчислення

$$Q = [m]P = \underbrace{P + P + ... + P}_{m}. \qquad (2.6.13)$$

Цю операцію виконують дуже ефективно, вона вимагає не більше $2\log m$ композицій точок. Підходи до її здійснення такі самі, як і до піднесення до степеня. Наприклад, щоб отримати точку $Q = [21]P$, обчислюємо $[2]P$, $[4]P$, $[8]P$, $[16]P$, кожного разу подвоюючи попередню точку, і додають $P + [4]P + [16]P = Q$ (усього 4 подвоєння і 2 додавання).

Зворотну задачу – ***дискретне логарифмування на еліптичній кривій*** – формулюють так. Знаючи точки P і Q, знайти таке число m, що $[m]P = Q$. Ця задача є дуже складною. Якщо ретельно вибрати параметри кривої, то якнайкращі відомі на цей час алгоритми для знаходження m потребують $O\sqrt{q}$ операцій на кривій, де q – потужність підмножини точок, якій належать точки P і Q. Усі обчислення на кривій виконують за модулем p, тобто з числами завдовжки $t \approx \log p$ бітів. Для криптографічних додатків $\log q \approx \log p$, тому $O\sqrt{q} = O(2^{t/2})$ означає експонентне зростання трудомісткості зі збільшенням довжини чисел.

Контрольні питання до розділу 2

1. Чи для всіх елементів множини Y у відображенні $f : X \to Y$ повинні існувати прообрази, щоби це відображення вважалось ін'єктивним?
2. Чи може бути сюр'єктивне відображення бієктивним?
3. Яке відображення називають оберненим?
4. Навести властивості оберненого відображення.
5. Дати визначення напівгрупи.
6. Якою властивістю моноїд відрізняється від групи?
7. Якою властивістю моноїд відрізняється від напівгрупи?
8. Дати визначення порядку елемента групи.
9. Дати визначення гомоморфізму в групах.
10. Дати визначення ізоморфізму в групах.
11. Дати визначення абелевої групи.
12. Назвіть порядок перестановки із n елементів.
13. Що є нейтральним елементом групи перестановок?
14. Дати визначення кільця.
15. Навести властивості кільця.

16. Дати визначення поля.
17. Дати визначення кільця многочленів.
18. Дати визначення векторного простору над полем F.
19. Дати означення неповної частки й залишку від ділення двох чисел.
20. Дати означення найбільшого спільного дільника двох чисел.
21. Дати означення взаємно простих чисел.
22. Дати опис алгоритму Евкліда.
23. Дати означення лінійних діофантових рівнянь з двома невідомими.
24. Навести алгоритм розв'язання лінійних діофантових рівнянь із двома невідомими.
25. Сформулювати й довести основну теорему арифметики.
26. Дати визначення чисел порівняльних за модулем.
27. Сформулювати й довести властивості порівнянь.
28. Дати визначення й навести приклади класів еквівалентності.
29. Дати означення зведеної системи лишків за модудем m.
30. Сформулювати й довести теорему Ейлера.
31. Сформулювати й довести малу теорему Ферма.
32. Сформулювати й довести китайську теорему про лишки.
33. Дати означення квадратних лишків і квадратних нелишків за модулем p.
34. Дати означення символу Лежандра.
35. Сформулювати й довести теорему критерій Ейлера для символу Лежандра.
36. Перелічити й довести найпростіші властивості символу Лежандра.
37. Сформулювати тест чисел на простоту Ферма.
38. Сформулювати тест чисел на простоту Соловея–Штрассена.
39. Сформулювати тест чисел на простоту Міллера–Рабіна.
40. Навести алгоритм перевірки чисел на простоту за малою теоремою Ферма.
41. Сформулювати правила визначення сильних простих чисел.
42. Навести алгоритм Гордона для визначення сильних простих чисел.
43. Сформулювати та довести теорему 1 для порівнянь будь-якого степеня за складним модулем.
44. Дати визначення понять масової та індивідуальної задач.
45. Дати визначення поняття алгоритму.
46. Дати визначення поліномного алгоритму.
47. Дати означення для асимптотичного наближення Θ (тета).
48. Дати означення для асимптотичного наближення O ("О велике").
49. Дати означення для асимптотичного наближення Ω (омега).
50. Навести основні властивості асимптотичного наближення O ("О велике").
51. Дати означення прямолінійної програми.
52. Описати процес визначення складності прямолінійної програми.
53. Описати бінарний алгоритм піднесення до степеня за модулем.
54. Навести приклади ймовірнісних алгоритмів.
55. Дати означення односторонніх функцій.
56. Дати означення функції з секретом.
57. Дати означення псевдовипадкового генератора.
58. Навести приклади розміщення довгих чисел в пам'яті комп'ютера.
59. Описати алгоритм множення довгого числа на коротке.
60. Описати алгоритм множення довгих чисел з використанням стовпчика.
61. Описати алгоритм швидкого множення довгих чисел.
62. Описати алгоритм множення довгих чисел з використанням швидкого перетворення Фур'є.
63. Використання алгоритму швидкого перетворення Хартлі для обчислення згортки.

64. Дати порівняльну характеристику алгоритмів множення довгих чисел.
65. Дати визначення поняття еліптичної кривої.
66. Які еліптичні криві називають сингулярними?
67. Дати визначення композиції двох точок на еліптичній кривій.
68. Вивести формули для визначення результуючої кординати при композиції двох точок еліптичної кривої.
69. Описати алгоритм m -кратної композиції точок еліптичної кривої.

Список літератури до розділу 2

1. Shannon C. E. Communication Theory of Secrecy Systems / C. E. Shannon // Bell System Technical Journal. – 1949. – Vol. 28, n. 4. – P. 656–715.
2. Шеннон К. Работы по теории информации и кибернетике : [пер. с англ.] / К. Шеннон; под ред. Р. Л. Добрушина и О. Б. Лупанова; с предисловием А. Н. Колмогорова. – М. : Изд-во иностранной литературы, 1963. – 830 с. Теория связи в секретных системах. – С. 333–402.
3. Горбенко І. Д. // Прикладна криптологія: Теорія. Практика. Застосування / І. Д. Горбенко, Ю. І. Горбенко. – Харків : Форт, 2013. – 880 с.
4. Coutinho S. C. The Mathematics of Cipher: number theory and RSA cryptography / S. C. Coutinho, A. K. Peters/CRC Press, 1999. – 196 p.
5. Rivest R. Are 'Strong' Primes Needed for RSA? : Cryptology ePrint Archive / R. Rivest, R. Silverman // Report 2001/007. – Режим доступу : http://eprint.iacr.org/2001/007/
6. Cormen T. H. Introduction to algorithms / T. H. Cormen et al. – 3rd ed. – Massahusetts Institute of Technology, 2009. – 1313 p.
7. Грэхем Р. Конкретная математика. Основания математики : [пер. с англ.] / Р. Грэхем, Д. Кнут, О. Паташник. – М. : Мир, 1998. – 793 с., ил.
8. Fürer M. Faster integer multiplication / M. Fürer // STOC 2007 Proceedings. – 2007. – P. 57–66.
9. Koblitz N. Elliptic Curve Cryptosystems / N. Koblitz // Mathematics of Computation. – No. 48. – 1987. – P. 203–209.
10. Василенко О. Н. Теоретико-числовые алгоритмы в криптографии / О. Н. Василенко. – М. : Изд-во МГУ, 2000ю – 328 с.
11. Ященко В. В. Введение в криптологию / В. В. Ященко. – СПб. : Питер, 2000. – 288 с.
12. Ноден П., Китте К. Алгебраическая алгоритмика. – М. : Мир, 1999. – 720 с.
13. Проценко В. С., Чаленко П. Й., Ставровський А. Б. Техніка програмування мовою Сі: навч. посіб. – К. : Либідь, 1993. – 223 с.
14. Miller V. S. Use of Elliptic Curves in Cryptography / V. S. Miller // Advances in Cryptology – Crypto'85. – LNCS 218. – 1986. – P. 417–426.

Розділ 3
КРИПТОЛОГІЯ

Криптологія [1, 2] – це наука, яка охоплює дві взаємопов'язані частини: криптографію та криптоаналіз. ***Криптографія*** – це наука про зовнішню зміну інформації (без зміни її змісту) до такого вигляду, щоб зміст цієї інформації став недоступний для сторонніх читачів (таке перетворення називають ще ***шифруванням,*** і воно є частиною ***шифру***). При цьому сам факт існування інформації, як правило, не приховують, що принципово відрізняє криптографію від іншої науки – ***стеганографії***, основне завдання якої – розроблення методів приховування таємної інформації під час її передавання. Тобто, криптографія – один зі способів втаємничення інформації.

Важливо розуміти, що в криптографії інформацію перетворюють із метою погіршення її розуміння. Це принципово відрізняє криптографію від ***кодування***, мета якого протилежна – покращення розуміння інформації навіть за впливу різноманітних завад. Разом із тим, термін "кодування" часом уживають як синонім до шифрування, що пов'язано з деякими військовими застосуваннями. Однак зараз термін кодування як синонім до шифрування є архаїчним саме через протилежні цілі цих, часом дуже схожих, перетворень.

Криптоаналіз – це наука про методи розкриття (зламу) шифрів, тобто про методи розкриття змісту повідомлень, змінених методами криптографії. Криптографія та криптоаналіз взаємопов'язані, оскільки часто досягнення криптоаналізу є єдиним обґрунтуванням надійності шифру.

3.1. Історія криптології

Сучасний історичний етап розвитку людства характеризується кризами донедавна стабільних міждержавних об'єднань, встановленням деяких державах тоталітарних режимів, загостренням існуючих та виникненням нових локальних військових конфліктів, посиленням тероризму, поширенням методів гібридних та інформаційних воєн, загостренням конкурентної боротьби в умовах глобалізації. У таких умовах ще більше зростає роль інформаційної безпеки, якій після світових воєн і періоду холодної війни провідні країни світу традиційно приділяли значну увагу.

Загалом людство дбало про інформаційну безпеку ще від часу виникнення писемності. У писемних джерелах майже всіх давніх цивілізацій є згадки про ті чи інші методи та засоби захисту інформації. Винятком є ті давні цивілізації, які культивували складне письмо, наприклад, Давній Китай. Таке письмо саме по собі було своєрідним інструментом захисту інформації. Давній і сталий інтерес до інформаційної безпеки дослідники пояснюють такими властивостями людської натури, як допитливість, з одного боку, і прагнення до втаємниченості – з іншого. Також стимулами розвитку методів захисту інформації завжди були загальний культурний розвиток, зокрема розвиток писемності, та активність військових дій.

Здавна відомі три основні способи захисту інформації [1, 2]. Перший із них передбачав захист інформації лише силовими методами. На ранніх етапах розвитку інформаційної безпеки це була звичайна фізична охорона документа або носія інформації. Сьогодні цей спосіб охоплює різноманітні системи сигналізації та охорони периметра, розмежування доступу, методи захисту від витоку інформації побічними каналами, організаційні та правові методи захисту інформації.

Другий спосіб отримав назву "стеганографія". Це латино-грецьке поєднання слів, які в сукупності означають "тайнопис". Суть цього способу в приховуванні самого факту наявності інформації. Наприклад, застосовували симпатичне чорнило, тобто речовину, напис якою був невидимий і проявлявся при певному хімічному обробленні. Наприклад, напис молоком є невидимий, але проявляється за нагрівання. Один із відомих прикладів приховування інформації наведено в працях давньогрецького історика Геродота. Таємне повідомлення записували на поголеній голові раба, після чого давали волоссю відрости, і відсилали раба до адресата. Поголивши раба, можна було прочитати повідомлення. Проте метод "голова раба" швидко застарів. Натомість тайнопис симпатичними чорнилами й різноманітні "наколювання" букв (спосіб тайнопису, у якому букви таємного повідомлення малопомітним способом позначають в іншому, нетаємному повідомленні), набули розповсюдження та активно застосовувалися аж до середини XX століття. Наприкінці XX століття стався справжній "вибух" стеганографічних методів, зумовлений розвитком інформаційних та мережевих технологій. Суть цих сучасних стеганографічних методів полягає в приховуванні таємної інформації в іншій "видимій" електронній інформації. Наприклад, можна бітами таємного повідомлення замінити молодші біти однієї зі складових кольору пікселів нетаємного графічного зображення. Сучасні стеганографічні методи є доступними, тобто не є ресурсомісткими, їх складно виявити й нейтралізувати. Такими методами часто користуються різноманітні терористичні організації.

Третій спосіб захисту інформації полягає у перетворенні змістовного тексту в деякий хаотичний набір знаків, причому отримувач повідомлення знає спосіб зворотного перетворення. Такий спосіб захисту інформації називають криптографією, або шифруванням. Зупинимося детальніше на історії криптографії як одного з найдавніших методів забезпечення інформаційної безпеки й водночас одного з основних інструментів сучасної інформаційної безпеки.

У документах давніх цивілізацій – Індії, Єгипту, Месопотамії – є відомості про способи складання шифрованих повідомлень. Один із найдавніших відомих шифрованих текстів походить із Месопотамії – це написаний клинописом на табличці рецепт виготовлення глазурі для гончарних виробів. При написанні цього рецепту було застосовано рідкісні клинописні знаки, ігнорували деякі голосні й приголосні, а також застосовували числа замість імен. Відомі шифровані тексти давнього Єгипту: релігійні тексти, медичні рецепти, надгробні епітафії. При написанні останніх давні єгиптяни експлуатували людську допитливість. Метою шифрування надгробних епітафій, як пише Девід Кан [3], було "підсилення таємниці, а тому й чарівної сили поминальних текстів". Єгиптяни вірили, що прочитання надгробних епітафій висловлює благословення, яке містилося в надгробних написах. Але зростаюча кількість таких написів послабила інтерес до їх прочитання. Щоб відродити інтерес, писці навмисне робили написи незрозумілими. Це привертало увагу читача, примушувало його задуматися й викликало бажання розгадати зміст написів.

Найвідоміші та достовірні дані про шифри походять із Давніх Греції та Риму. Найвідомішим історичним прикладом шифру є ***шифр Юлія Цезаря*** (Гай Юлій Цезар, Imperator Gaius Iulius Caesar, 1 ст. до н. е.), описаний істориком Давнього Риму ***Свєтонієм*** (Гай Светоній Транквілл, Gaius Suetonius Tranquillus). Цезар застосовував шифр зсуву. Кожну букву таємного повідомлення він заміняв іншою, а саме такою, яка розміщена в алфавіті через три позиції. ***Шифр зсуву*** є звуженням ***шифру заміни або підстановки***, який до сьогодні є базовим перетворенням симетричної криптографії.

Іншим відомим шифром тих часів був шифр і шифрувальний пристрій ***"Скитала"***.

Скиталу було винайдено в давній Спарті. Рим також швидко скористався цим пристроєм. Скиталою (див. рис. 3.1) був циліндр певного діаметра. На нього намотували смужку із пергаменту, після чого повідомлення писали на ній вздовж осі скитали. Потім смужку розмотували й відправляли адресату. Якщо супротивник перехоплював повідомлення, він читав на ньому незрозумілий набір букв. Натомість адресат намотував смужку на Скиталу такого самого діаметра й читав таємне повідомлення. Скитала є класичним прикладом шифру

перестановки – іншого із двох базових перетворень сучасних симетричних шифрів. Цікаво, що винайдення дешифрувального пристрою з умовною назвою ***"Антискитала"*** приписують ***Арістотелю*** (Aristotle). Він запропонував для розшифрування застосовувати конусоподібний спис, на який намотували перехоплену смужку, пересовували її вздовж осі, змінюючи діаметр, поки не прочитувати зв'язний текст.

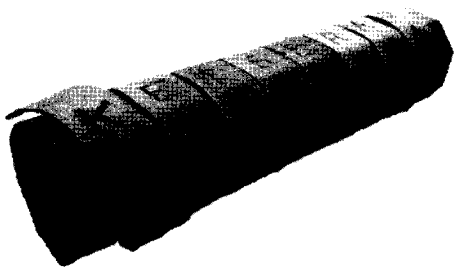


Рис. 3.1. Скитала

Шифри Цезаря та Скитала найвідоміші, але не найдавніші з ранніх шифрів. Ще задовго до них було розроблено й поширено способи шифрування. Деякі з них дійшли до наших днів. Наприклад, ще один винахід давніх греків – ***квадрат Полібія*** (Polybius square) – у якому букви латинського алфавіту розміщували в квадратній таблиці, рядки й стовпці якої було позначено першими п'ятьма буквами алфавіту. Кожну букву повідомлення шифрували двома: перша позначала рядок, а друга – стовпчик у квадраті Полібія. Якщо рядки й стовпці квадрата позначити цифрами, то квадрат Полібія перетвориться на сучасний "тюремний" шифр, у якому інформацію передають перестуком. Спочатку кількість ударів, яка позначає рядок, а потім кількість ударів, яка позначає стовпчик.

Після розвалу Священної Римської імперії почався період раннього середньовіччя – період "темних віків". Він характеризувався занепадом культури й науки, майже абсолютною неграмотністю. Навіть тогочасні королі бували неграмотні, тому в ті часи не було особливої необхідності додатково ускладнювати шифруванням письмо, яке й так мало хто розумів. Відповідно методи шифрування не зазнали помітного розвитку в цей історичний період. Разом з тим, деякі держави близького сходу в цей час культивували елітну науку й зробили помітний внесок у розвиток криптографії.

Період середньовіччя змінився періодом Відродження, який характеризувався активним розвитком культури, науки, активізацією військових дій. Відповідно знову активно розвивали криптографію, яку почали застосовувати не лише в політиці й військовій справі, а також для захисту інтелектуальної

власності від викрадення іншими вченими, що наблизило тодішню криптографію до сучасної. Архітектор ***Леон Батіста Альберті*** (Leon Battista degli Alberti) винайшов ***метод багатоалфавітної заміни***. Цей метод потім удосконалив і дав йому своє ім'я ***Блейз де Віженер*** (Blaise de Vigenere) – дипломат XIV століття. Удосконалена багатоалфавітна заміна до сьогодні залишається основою криптографічної стійкості симетричних шифрів.

Особливого практичного значення криптографія набула в часи Першої й Другої світових воєн [2]. Зараз, із відстані десятиліть, історики та військові аналітики схиляються до думки, що саме сила чи слабкість застосованих сторонами криптографічних методів визначили стратегічні успіхи або невдачі цих сторін, а тому фактично визначили й наслідки воєн. Зокрема розгром російської армії у Східній Прусії в Першій світовій війні стався завдяки зламу капітаном Австро-Угорської армії Германом Покірним застосованого росіянами шифру. А наслідки Другої світової війни значною мірою були зумовлені зламом німецького шифру ***Енігма*** (Enigma) англійцями за допомогою ***машини Тюрінга (Turing machine)***.

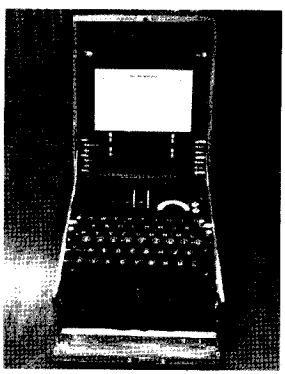


Рис. 3.2. Шифрувальна машина Енігма

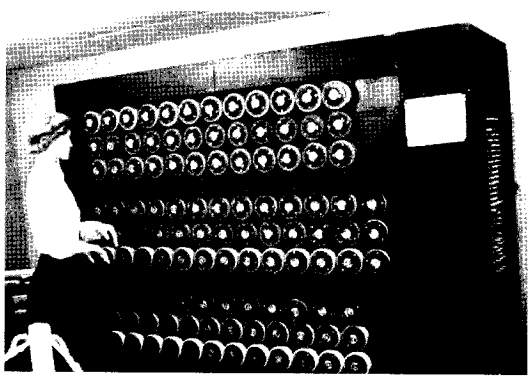


Рис. 3.3. Дешифрувальна машина "Bombe" Тюрінга

Ще одним історичним поворотом у розвитку криптографії в XX столітті стала фундаментальна праця ***Клода Елвуда Шеннона*** (Claude Elwood Shannon) "Теорія зв'язку в секретних системах" [4] – таємна доповідь, яку автор виконав у 1945 році та опублікував у "Bell System Technical Journal" в 1949 році. У своїх працях [4, 5] Шеннон, який одночасно працював у двох областях – в області передавання інформації та в області секретного зв'язку – сформулював наукові принципи криптографії з погляду теорії інформації. Саме праці Шеннона поклали початок криптографії як сучасної математичної науки.

Після Другої світової війни, з усвідомленням ролі криптографії та криптоаналізу у військовій справі, криптографія перетворилася на секретну, зосереджену у військових відомствах, науку. Цей стан речей ще більше загострився в роки Холодної війни. У результаті весь цивільний науковий та освітній світи були відрізані від цієї науки. Разом із тим у післявоєнні роки, разом із відбудовою й початком бурхливого розвитку економіки, сформувався ринок, і зріс попит на криптографічні методи та засоби для захисту комерційної інформації. У цих умовах споживачі таких методів та засобів виявилися незахищеними. Цивільні науковці не могли підтвердити надійність криптографічних засобів, – вони не були сертифікованими і стандартизованими. Відповідно зростав інтерес до вивчення криптографії як науки. Ситуація різко змінилася в 70-ті роки XX століття. Спочатку Національне бюро стандартів США добилося від ***Агентства національної безпеки*** (АНБ) США дозволу на впровадження та відкрите опублікування стандартного симетричного шифру ***DES***. Проведення конкурсу на стандартний шифр, поява перших відкритих публікацій із криптографії, публікація стандартного симетричного шифру, удосконалення та вихід із-під контролю АНБ програмних реалізацій DES, а також винайдення в 70-ті роки принципово нового підходу до шифрування – асиметричного, або двоключового шифрування – характеризують сучасний етап розвитку криптографії як відкритої науки. Сучасна криптографія не тільки один зі способів захисту інформації, але й один з основних інструментів найрізноманітніших напрямків інформаційної безпеки.

Основна задача сучасної інформаційної безпеки – забезпечення конфіденційності. Крім того, за допомогою методів інформаційної безпеки, зокрема криптографічних методів, вирішують також завдання перевірки оригінальності (забезпечення автентичності), забезпечення цілісності інформації та задачі неспростовності (незаперечення авторства).

Сьогодні системи інформаційної безпеки вивчають та розробляють як складні системи соціально-технологічного типу. Такі системи повинні відповідати цілому ряду вимог, основна з яких – адаптованість, тобто система інформаційної безпеки не повинна бути статичною. Вона має бути гнучкою й адаптуватися до змінних умов захисту інформації, наприклад, модифікуватися у разі заміни чи оновлення програмних чи апаратних засобів оброблення й передавання інформації, зміни будівельних чи комунікаційних особливостей об'єкта, де зберігають, обробляють та передають таємну інформацію, чи навколишніх об'єктів. Разом із тим сьогодні не достатньо програмно-технічних засобів для ефективного захисту інформації. Навіть у технічно добре захищених системах ефективними залишаються такі методи добування інформації як підкуп, шантаж, тортури, експлуатація таких людських рис, як комуніка-

бельність, співчуття, симпатія, честолюбство. Особливості застосування таких методів добування інформації та способи захисту від них вивчають у межах такої нетехнічної дисципліни як соціальна інженерія.

Отже, для володіння сучасними методами забезпечення інформаційної безпеки необхідне вивчення нормативної та правової бази, організаційних методів та методів управління захистом інформації, методів соціальної інженерії, основ документаційного забезпечення захисту інформації, технічних засобів охорони об'єктів, методів і засобів захисту інформації від витоку основними й побічними каналами, криптографічних та стеганографічних методів захисту інформації, методів побудови й застосування захищених протоколів обміну інформацією, методів захисту інформації в комп'ютерних системах та мережах різного типу.

3.2. Основні поняття та визначення криптології

Класична задача криптології [2] виникає тоді, коли двоє людей, а в загальному випадку дві сторони (наприклад, комп'ютери), мають намір чи завдання обмінятися таємною інформацією за присутності третьої, недружньої сторони, яка намагається заволодіти цією інформацією. Цю сторону називають ***суперником***.

Припустімо, ***користувач А*** передає конфіденційну інформацію, а ***користувач В*** її приймає. Суперник може бути пасивним (перехоплювач повідомлень) або активним (зловмисний активний зламувач шифрів). Як бачимо, сторони обміну інформацією, названі тут користувач А і користувач В, відповідно до цих імен наділені типовими цілями та можливостями. Саме тому в криптографічній літературі (особливо в описах криптографічних протоколів) часто застосовують умовні імена – щоб скоротити і спростити складні описи процесів інформаційного обміну завдяки вилучення з них докладних характеристик сторін обміну інформацією.

З погляду сучасних інформаційних і телекомунікаційних технологій, користувач А і користувач В є законними користувачами каналу зв'язку або легальними користувачами інформаційної мережі. Суперник є несанкціонованим користувачем, який намагається перехопити (підслухати) або зламати таємне повідомлення. Щоб зберегти таємницю, користувач А ***зашифровує*** своє повідомлення – тобто, застосувавши якийсь ***алгоритм зашифрування***, перетворює його до незрозумілого для суперника вигляду. У результаті з повідомлення, яке ще називають ***відкритим повідомленням*** або ***відкритим текстом***, користувач А отримує ***шифртекст*** (***шифротекст, криптотекст***).

Часто поняття відкритого повідомлення й відкритого тексту розрізняють. Відкрите повідомлення позначають буквою M (від англ. message). Під цим поняттям розуміють будь-яку відкриту інформацію (текст, потік бітів, оцифрований звук, цифрове відеозображення тощо). Тобто для комп'ютера M – це набір двійкових даних. Якщо необхідно підкреслити, що відкрите повідомлення є насправді текстом деякою мовою, його позначають буквою P (від англ. *Plain Text* – звичайний текст). Шифртекст прийнято позначати буквою C (від англ. *Ciphertext*).

Означення. Процес зміни форми повідомлення способом, що дає змогу приховати його зміст, називають ***зашифруванням***. Зашифроване повідомлення називають шифртекстом (шифротекстом, криптотекстом).

Означення. Процедуру зворотного перетворення ***шифртексту*** у відкритий текст називають ***розшифруванням.***

Щодо терміна ***шифрування***, то ним іноді позначають пару перетворень: зашифрування та розшифрування, а в інших випадках уживають як синонім до зашифрування.

Отже, користувач В отримує шифртекст і розшифровує його за допомогою ***алгоритму розшифрування***. У результаті він отримує початкове відкрите повідомлення.

Означення. Алгоритми зашифрування та розшифрування разом складають ***алгоритм шифрування*** або частину шифру.

Науку й мистецтво створення шифрів називають ***криптографією***. ***Криптоаналіз*** – наука й мистецтво зламу або розкриття шифрів. Криптоаналітики намагаються здійснити ***дешифрування*** – тобто відновити зміст перехопленого ***шифртексту*** без знання таємних параметрів шифру (***ключа*** шифру). Тобто, розшифрування та дешифрування – два різні поняття.

Розділ математики, що охоплює криптографію та криптоаналіз, називають ***криптологією,*** а спеціалістів, які нею займаються – криптологами.

Обмежені та відкриті алгоритми шифрування. Якщо рівень захисту інформації, який забезпечує алгоритм шифрування, залежить від збереження цього алгоритму в таємниці, такий алгоритм називають ***обмеженим***. Обмежені алгоритми не витримують зміни користувачів, не допускають ефективного контролю та стандартизації, застосування відкритих програмних чи апаратних продуктів.

Надійнішим і зручнішим є застосування ***відкритих алгоритмів шифрування***. Такі алгоритми відомі всім, тобто не тільки легальним користувачам, але й супернику. Застосовуючи відкриті алгоритми шифрування, конфіденційність забезпечують застосуванням таємного параметра шифру. Такий па-

раметр називають ***ключем*** і позначають буквою K. Наприклад, у шифрі Цезаря ключем є величина зсуву в алфавіті, у шифрі Скитала ключем є її діаметр. Легальні користувачі домовляються про таємний ключ наперед (обмінюються ключем) або застосовують кожен свій таємний ключ. Цей ключ вони зберігають у таємниці й часто змінюють. Множину всіх можливих ключів називають ***простором ключів***. Аналогічно визначають ***простори відкритих повідомлень та шифртекстів.***

Означення. ***Криптосистема,*** або ***шифр*** – це алгоритм шифрування (який складається з алгоритмів зашифрування й розшифрування), а також усі можливі відкриті тексти, шифртексти та ключі.

Із застосуванням розглянутих понять класичну криптографічну схему зображають так (рис. 3.4):

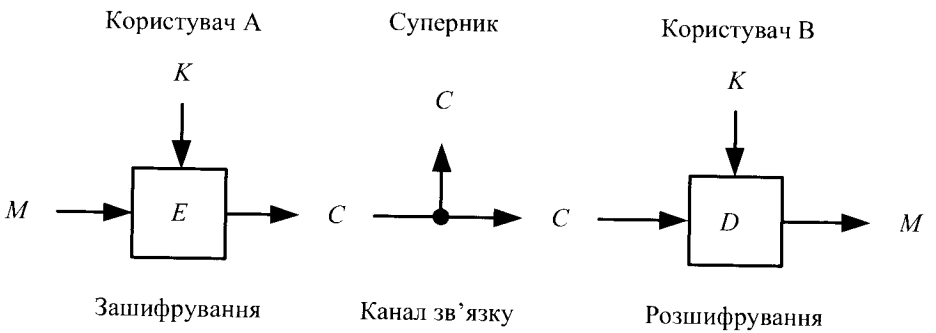


Рис. 3.4. Класична криптографічна схема

На схемі буквами E та D позначено алгоритми зашифрування та розшифрування відповідно. Ключ K позначено на обох сторонах схеми однаково, хоча ключі зашифрування та розшифрування не завжди однакові.

Те, що шифртекст C є результатом застосування алгоритму E до відкритого тексту M із ключем K, символічно записують так:

$$C = E_K(M).$$

Так само на приймальному боці M – це результат застосування алгоритму розшифрування D до шифртексту C із ключем K:

$$M = D_K(C).$$

При цьому має справджуватися рівняння:

$$M = D_K(E_K(M)).$$

Аналізуючи надійність шифрів, побудованих на відкритих алгоритмах, припускають, що суперник не лише здатний підслухати шифртекст C, але й знає алгоритми зашифрування й розшифрування E та D. І лише ключ K йому невідомий.

Симетричні та асиметричні алгоритми шифрування. Відомо два основні типи алгоритмів шифрування, заснованих на застосуванні ключів: симетричні алгоритми шифрування та асиметричні алгоритми шифрування.

Симетричні алгоритми шифрування називають інакше алгоритмами із секретним або таємним ключем або алгоритмами з єдиним (спільним) ключем. Такі алгоритми потребують, щоб користувачі перед початком передавання секретних даних узгодили єдиний, таємний для суперника, ключ. Якщо в симетричному алгоритмі ключі зашифрування та розшифрування різні, то ключ розшифрування можна легко обчислити із ключа зашифрування.

Симетричні алгоритми поділяють на 2 категорії:

– ***Потокові алгоритми*** (шифри), що обробляють відкритий текст побітово;

– ***Блокові алгоритми*** (шифри), що обробляють групи бітів відкритого тексту. Такі групи називають блоками.

До появи комп'ютерів алгоритми обробляли текст посимвольно.

Асиметричні алгоритми шифрування (алгоритми з відкритим, або публічним ключем) передбачають наявність двох різних ключів: відкритого (публічного) ключа зашифрування та закритого (таємного) ключа розшифрування. Причому на відміну від симетричних алгоритмів ключ розшифрування дуже важко обчислити за ключем зашифрування. У таких системах користувач В (тобто отримувач таємного повідомлення) відкрито публікує свій ключ зашифрування, і будь-хто може надіслати користувачеві В повідомлення, зашифроване його відкритим ключем. Але таємний ключ розшифрування знає тільки користувач В, і тільки він може розшифрувати й прочитати повідомлення, надіслані йому.

Часто асиметричні алгоритми застосовують у режимі, коли зашифрування здійснюють таємним ключем відправника, а розшифрування – відкритим ключем відправника. Такий режим називають режимом ***цифрового підпису***.

Задачі криптографії. Основна задача криптографії – забезпечення конфіденційності. Крім цього, криптографічні алгоритми застосовують для вирішення інших завдань, зокрема:

– перевіряння оригінальності (автентифікація) – отримувач повинен мати можливість установити джерело повідомлень, а суперник не здатний замаскуватися під когось іншого;

– забезпечення цілісності – отримувач повинен мати можливість перевірити, чи не було повідомлення змінене в процесі доставлення, а суперник не здатний видати змінене повідомлення за справжнє;

– незаперечення авторства – відправник не повинен мати можливості заперечувати відправлення свого повідомлення або своє авторство.

Основна задача криптоаналітика – відновити зміст повідомлення M. Для цього, відповідно до традиційної криптографічної термінології, криптоаналітик здійснює ***атаку на шифр***. Успішний криптоаналіз дає змогу відновити відкритий текст або ключ, а також виявити слабкі місця в криптосистемі.

Означення. Розкриття ключа без застосування методів криптоаналізу називають ***компрометацією ключа***.

Існує чотири основні типи криптоаналітичних атак [1] у припущенні, що криптоаналітик детально знає алгоритм шифрування:

1. ***Атака на основі лише шифртексту***. Тут криптоаналітик має у своєму розпорядженні шифртексти одного або декількох повідомлень, зашифрованих однаковим ключем. Його задача – дешифрування якомога більшої кількості повідомлень. Ще краще, якщо він установить ключ зашифрування. Це дасть змогу йому розшифровувати також інші повідомлення, зашифровані цим ключем. Отже:

   Дано: $C_1 = E_k(M_1)$; $C_2 = E_k(M_2)$; …, $C_i = E_k(M_i)$.

   Знайти: відкриті тексти M_i, або ключ K, або альтернативний алгоритм відновлення C_{i+1} з M_{i+1} без знання ключа K.

2. ***Атака з відомим відкритим текстом*** – криптоаналітик має доступ не лише до шифртекстів декількох повідомлень, але й до відкритих текстів цих повідомлень. Його задача полягає у визначенні застосованого ключа з метою дешифрування інших повідомлень, зашифрованих цим ключем.

   Дано: $C_1 = E_k(M_1), M_1$; $C_2 = E_k(M_2), M_2$; $C_i = E_k(M_i), M_i$.

   Знайти: ключ K або альтернативний алгоритм відновлення C_{i+1} з M_{i+1} без знання ключа K.

3. ***Атака з вибраним відкритим текстом*** – криптоаналітик має не лише шифртексти й відповідні відкриті тексти, але й можливість вибирати відкритий текст для зашифрування. Задача полягає в розкритті ключа або знаходженні альтернативного алгоритму, який дає змогу читати шифртексти без знання ключа. Такий тип атаки відповідає мінімальному рівню можливостей суперника в асиметричних криптосистемах. Адже в таких системах ключ зашифрування супернику відомий, і він може зашифровувати ним будь-які повідомлення.

   Дано: $C_1 = E_k(M_1), M_1; C_2 = E_k(M_2), M_2; \ldots, C_i = E_k(M_i), M_i$,

   де $M_1 \ldots M_i$ криптоаналітик може вибирати.

   Знайти: ключ K або альтернативний алгоритм відновлення C_{i+1} з M_{i+1} без знання ключа K.

4. ***Атака з адаптивно підібраним відкритим текстом***. Це особливий випадок атаки з вибраним відкритим текстом. Криптоаналітик може не

лише вибирати відкритий текст для зашифрування, але й уточнювати кожен свій наступний вибір, грунтуючись на попередніх результатах шифрування.

Крім цих чотирьох основних, є щонайменше ще три типи атак:

5. ***Атака з вибраним шифртекстом*** – криптоаналітик може вибирати різні шифртексти для розшифрування, а також має доступ до розшифрованих відкритих текстів. Наприклад, на якийсь час він отримав доступ до обладнання, яке автоматично здійснює розшифрування. Задача полягає в розкритті ключа.
6. ***Атака з підібраним ключем*** – криптоаналітику відомо про зв'язки між різними ключами.
7. ***Бандитський криптоаналіз***. Для отримання ключа суперник вдається до погроз, шантажу, тортур, підкупу та інших методів компрометації ключа.

Стійкість алгоритмів шифрування. Залежно від складності зламу, різні криптоалгоритми забезпечують різні ступені захисту [1]. Якщо вартість зламу більша за вартість зашифрованих даних, тоді дані, очевидно, у безпеці. Так само якщо час, необхідний для зламу алгоритму, більший за час, протягом якого треба зберігати таємницю, зашифрована інформація також у безпеці. Якщо обсяг даних, зашифрованих одним ключем, менший за мінімальний обсяг даних, необхідний для зламу алгоритму, дані також у безпеці.

Відома така класифікація складності зламу (у порядку спадання значущості) [1]:

1. ***Повне розкриття.*** Криптоаналітик знаходить ключ K, такий, що $D_K(C) = M$.
2. ***Глобальна дедукція.*** Криптоаналітик знаходить альтернативний алгоритм A, еквівалентний $D_K(C)$, без знання ключа K.
3. ***Випадкова (або часткова) дедукція***. Криптоаналітик знаходить (наприклад, викрадає) відкритий текст для перехопленого шифртексту.
4. ***Інформаційна дедукція.*** Криптоаналітик добуває деяку інформацію про ключ або про відкритий текст. Це можуть бути, наприклад, декілька бітів ключа або відомості про форму відкритого тексту.

Алгоритм шифрування ***безумовно стійкий*** або, інакше, ***стійкий у теоретико-інформаційному сенсі***, якщо відновлення відкритого тексту неможливе за будь-якого обсягу перехопленого шифртексту. Існує (довів Шеннон) єдиний абсолютно стійкий шифр. Усі інші шифри можна розкрити з використанням тільки шифртексту простим перебором можливих ключів і перевірянням осмисленості отриманого відкритого тексту. Таку атаку називають ***брутальною (лобовою, brute force) атакою***.

Алгоритм шифрування називають ***стійким в обчислювальному сенсі***, якщо його не можна розкрити за прийнятний час із застосуванням доступних обчислювальних потужностей.

Складність тієї чи іншої атаки оцінюють різними способами [1]:

1. За складністю даних, тобто за обсягом вихідних даних, необхідних для успішної атаки.
2. За складністю оброблення, тобто за часом, необхідним для атаки. Часто таку оцінку називають фактором затрат праці.
3. За вимогами до пам'яті комп'ютера, тобто за обсягом пам'яті, необхідним для успішної атаки.

Приблизну складність атаки визначають максимальною оцінкою, обчисленою за кожним із цих трьох факторів.

Означення. Довільну скінченну непусту множину називають ***алфавітом***.

Означення. Елементи алфавіту називають ***буквами***.

Означення. Будь-які кінцеві послідовності елементів алфавіту (букв) називають ***словами***.

Означення. Слово, що не містить букв, називають пустим і позначають λ.

Означення. Довжина слова – це кількість букв у ньому, причому однакові букви рахують стільки разів, скільки вони входять до слова.

Множину всіх слів над алфавітом A позначають A^*.

Підмножини множини A^* називають ***мовами (формальними мовами)*** над алфавітом A.

Приклади алфавітів:

{а, б, в, г, ..., ь, ю, я} – український алфавіт;

{а, б, в, г, ..., ь, ю, я,-,?,(,),…} – український алфавіт з розділовими знаками;

Z_{33} = {0, 1, 2, 3, …, 32}– кільце лишків за модулем 33 – цифровий аналог українського алфавіту.

Отже, коли говорять про криптосистему або шифр, мають на увазі такі об'єкти:

1. ***Простір повідомлень.*** Нехай маємо алфавіт A, у якому записують відкриті повідомлення. Відкрите повідомлення M є словом у цьому алфавіті (це слово може складатися з багатьох слів у звичайному лінгвістичному розумінні): $M \in A^*$. Множину A^* називають ***простором повідомлень,*** або відкритих текстів.
2. ***Простір шифртекстів.*** Шифртексти записують в алфавіті B. Множину B^* називають простором шифртекстів. Часто $A = B$.
3. ***Простір ключів*** K. Складається зі слів у деякому алфавіті. Ці слова називають ***ключами***.

4. ***Зашифровуюче відображення*** E. Простір відкритих текстів A^* за допомогою простору ключів відображають у простір шифртекстів: $K \times A^* \to B^*$.
5. ***Розшифровуюче відображення*** D: $K \times B^* \to A^*$.

Відображення E і D повинні мати властивість, яка гарантує можливість розшифрування інформації:

$$D(K, E(K, M)) = M \text{, для всіх } M \in A^*, k \in K .$$

3.3. Класичні криптосистеми та їхній криптоаналіз

3.3.1. Шифри простої заміни

Такі шифри перетворюють відкритий текст так, що кожен його символ замінюють на якийсь інший символ [2]. При цьому однаковим символам у відкритому тексті відповідають однакові символи в шифртексті. Ключем шифру простої заміни є ***таблиця підстановок***, яка вказує, у який саме символ шифртексту переходить кожен символ відкритого тексту.

Наприклад, шифр Цезаря в українському алфавіті задають такою таблицею підстановок:

А	Б	В	Г	Ґ	Д	Е	Є	Ж	З	И	І	Ї	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ю	Я
Г	Ґ	Д	Е	Є	Ж	З	И	І	Ї	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ю	Я	А	Б	В

Під час шифрування кожну букву відкритого тексту відшукують у верхньому рядку й заміняють на відповідну букву з нижнього рядка. Під час розшифрування – навпаки. У цьому випадку букви в нижньому рядку таблиці підстановок також розташовані в алфавітному порядку. Тобто шифр Цезаря є ***шифром зсуву*** – звуженням ***шифру простої заміни***. У загальному випадку в шифрі простої заміни букви в нижньому рядку таблиці підстановок можуть бути розташовані в довільному порядку (саме їх розміщення і є ключем шифру). Це можуть бути взагалі не букви, а цифри, символи чи, для прикладу, танцюючі чоловічки з повісті Артура Конан Дойля. Це не змінює суті шифру, а також не впливає на стійкість шифру. Тому вважатимемо, що відкритий текст і шифртекст записують в однаковому алфавіті.

Слід сказати, що запам'ятати ключ шифру простої заміни непросто, а зберігати записаним ризиковано через небезпеку компрометації ключа. Тому на практиці для формування ключа часто застосовують ключове слово. Такий шифр називають ***шифром Цезаря із ключовим словом***. Ключовими пара-

метрами такого шифру є число від 0 до $n-1$ (де n – кількість букв алфавіту) і слово в цьому алфавіті. Розглянемо приклад складання таблиці підстановок у шифрі Цезаря із ключовим словом. Для прикладу виберемо ключові параметри: число ***3***, ключове слово ***хешування***.

Насамперед усуваємо із ключового слова повтори букв – отримаємо ***хешуваня***. Далі записуємо верхній рядок таблиці підстановок – український алфавіт. Букви алфавіту нумеруємо, починаючи з нуля. У нижній рядок таблиці записуємо ключове слово, починаючи з позиції ***3***, яка є першим ключовим параметром. Після ключового слова записуємо решту букв алфавіту в алфавітному порядку. У результаті отримаємо таку таблицю підстановок:

0	1	2	3	4	5	6	7	8	9	10	11	12	13		29	30	31	32
А	Б	В	Г	Ґ	Д	Е	Є	Ж	З	И	І	Ї	Й	...	Щ	Ь	Ю	Я
Щ	**Ь**	**Ю**	**Х**	**Е**	**Ш**	**У**	**В**	**А**	**Н**	**Я**	Б	Г	Ґ	...	Т	Ф	Ц	Ч

У такому шифрі ключові параметри запам'ятати нескладно, а таблицю підстановок за потреби можна легко створити, та після шифрування / розшифрування знищити. Разом із тим ключовий простір такого шифру є значно вужчим за ключовий простір загального шифру простої заміни.

Криптоаналіз шифру простої заміни. Оцінимо, скільки може бути різних ключів у шифрі простої заміни [2]. Для прикладу обмежимося українським алфавітом без розділових знаків – разом 33 букви. Отже, при складанні ключа – таблиці підстановок – букву на першу позицію в нижньому рядку таблиці ми можемо вибрати з 33 доступних букв алфавіту. Для вибору букви на другу позицію в нас залишається 32 букви, на третю позицію – 31 і так далі. Нарешті, для останньої, 33 позиції в нас залишається єдина буква. Отже, загальна кількість варіантів розміщення букв у нижньому рядку дорівнює:

$$33 \cdot 32 \cdot 31 \cdot \cdots \cdot 1 = 33!$$

Якщо ж алфавіт налічує n букв, кількість ключів для такого алфавіту – $n!$. Щоправда, деякі із цих ключів непридатні, наприклад, такий, де нижній рядок таблиці підстановок збігається з верхнім.

Простір ключів шифру зсуву є значно меншим і має $n-1$ різних ключів.

Об'єм простору ключів шифру простої заміни для українського й латинського алфавітів ($n=33$ та $n=26$ відповідно) оцінюють надзвичайно великими числами ($33! > 10^{33}$ та $26! > 10^{26}$ відповідно). Це означає, що брутальна атака (перебором ключів) на такий шифр є безперспективною. Але це не означає, що шифр загалом є стійкий. Шифр простої заміни легко зламують за допомогою методу, який був основою криптоаналізу до середини XX століття. Цей метод називають частотним криптоаналізом.

Частотний криптоаналіз.

Означення. ***Частота символу*** в **тексті** дорівнює кількості його входжень у цей текст, поділеній на загальну кількість символів у тексті.

Наприклад, маємо відкритий текст "криптологічні_перетворення". Загальна кількість букв – 26. Частота букви "о" у цьому тексті – 3/26. Натомість частота букви "і" у цьому самому тексті становить 2/26.

Основою частотного криптоаналізу [2] є такий ***емпіричний факт***: у достатньо довгих текстах кожна буква зустрічається із приблизно однаковою частотою, залежною від букви й не залежною від тексту. Отже, з кожним символом пов'язують деяке число – частоту цього символу в мові. Частоти букв для різних мов є відомі (табл. 3.1).

Таблиця 3.1

Частоти букв у текстах українською та англійською мовами

В українській мові						В англійській мові					
Буква	Частота	Буква	Частота	Буква	Частота	Буква	Частота	Буква	Частота	Буква	Частота
а	0,062	л	0,035	ц	0,004	a	0,0804	b	0,0154	c	0,0306
б	0,014	м	0,026	ч	0,012	d	0,0399	e	0,1251	f	0,0230
в	0,038	н	0,053	ш	0,006	g	0,0196	h	0,0549	i	0,0726
г	0.013	о	0,090	щ	0,003	j	0,0016	k	0,0067	l	0,0414
д	0,025	п	0,023	и	0,016	m	0,0253	n	0,0709	o	0,0760
е	0,072	р	0,040	ь	0,014	p	0,0200	q	0,0011	r	0,0612
ж	0,007	с	0,045	є	0,003	s	0,0654	t	0,0925	u	0,0271
з	0,016	т	0,053	ю	0.006	v	0,0099	w	0,0192	x	0,0019
і	0,062	у	0,021	я	0.018	y	0,0173	z	0,0009		
й	0,010	ф	0,002	пр.	0,174						
к	0,028	х	0,009								

За допомогою таких таблиць шифр простої заміни зламують просто, адже в шифрі простої заміни кожну букву відкритого тексту завжди заміняють однаково. Очевидно, що за такої заміни частотні характеристики символів не приховують. Криптоаналітику достатньо підрахувати частоти символів у перехопленому шифртексті й замінити ці символи такими, яким більше "пасують" підраховані частоти. І шифр зламано. Звичайно, за таким методом зворотна заміна під час частотного аналізу не завжди правильна. Але тут на допомогу криптоаналітику приходить інший феномен людських мов – феномен надлишковості. Більше того, за результатами частотного аналізу різних мов відомо, що в кожній мові є невелика група букв (8–9), які зустрічаються найчастіше й заповнюють близько 3/4 текстів цією мовою. Завдяки феномену надлишковості текст, у якому розкрито 3/4 букв, уже можна сподіватися

прочитати. Тому за частотного аналізу достатньо правильно встановити ці найуживаніші букви.

На практиці при зламуванні шифру простої заміни криптоаналітики одночасно із частотним аналізом застосовують інші прийоми, які дають можливість спростити аналіз. Наприклад, якщо перед шифруванням із відкритого тексту не вилучено пропусків, криптоаналітик може легко встановити слова з одної-двох букв, яких в кожній мові є небагато. Застосовують також ***протягування ймовірного слова***. Наприклад, криптоаналітик знає, що зашифроване повідомлення є листом і припускає наявність у документі звертання "Шановний". Припускаючи, що лист починається із цього звертання, він робить зворотну заміну й перевіряє осмисленість дешифрованого документа. Якщо результат його не задовольняє, він "посуває" імовірне слово на одну позицію й так далі, аж поки лист не буде дешифрованим.

Додаткові прийоми дають змогу підвищити ефективність частотного аналізу й так скоротити об'єм даних, необхідних для успішного аналізу. Відомо, що за допомогою частотного аналізу й різних допоміжних прийомів шифр простої заміни можна зламати, перехопивши шифртекст, обсяг якого не менший за кількість букв у цьому алфавіті.

Зрозуміло, що через ефективність частотного криптоаналізу шифр простої заміни не є стійким, тому його сьогодні не застосовують. Разом із тим частотний аналіз до сьогодні є потужним криптографічним інструментом, але його застосовують не для криптоаналізу, а з іншою метою. Частотний аналіз дає можливість комп'ютеру відрізняти осмислений текст від хаотичного набору символів. Завдяки цьому на комп'ютер можна перекласти здійснення брутальної атаки на різні шифри. Тобто сьогодні частотний аналіз дає змогу автоматизувати процедуру криптоаналізу шифрів.

3.3.2. Гомофонний шифр заміни

Шифр заміни, стійкий до класичного однолітерного частотного аналізу, винайшов ***Йоган Карл Фрідріх Гаусс*** (Johann Carl Friedrich Gauss) [2]. У цьому шифрі кожну букву відкритого тексту заміняють не єдиним символом, а будь-яким символом із декількох можливих. Кількість варіантів заміни при цьому пропорційна до емпіричної частоти використання літери у цій мові. Вибір варіанта заміни щоразу випадковий. Обов'язкова умова: замість різних букв необхідно підставляти різні символи. Тобто множини символів для заміни різних літер не повинні перетинатися. Це означає, що алфавіт шифртексту має бути більшим за алфавіт відкритого тексту. Тому часто в цьому шифрі букви відкритого тексту заміняють на числа. Наприклад, замість "а" підставляють одне із

чисел: 10, 17, 23, 46, 55. Натомість замість "б" – одне із двох: 12, 71 (насправді частота букви "а" у сім разів більша за частоту букви "б", тому кількість чисел для заміни "а" має також бути в сім разів більша за кількість чисел для заміни "б").

Оскільки кількість варіантів заміни для кожної букви пропорційна до її частоти, у доволі довгому шифртексті однакові символи зустрічаються із приблизно однаковою частотою, що робить класичний однолітерний частотний аналіз безперспективним. Однак результативним є інший варіант частотного аналізу, який ураховує частоти пар символів (біграм), які теж є приблизно однаковими в різних текстах однією мовою.

3.3.3. Поліграмні шифри

Означення. Послідовність кількох букв тексту називають ***поліграмою***. Послідовність із двох букв називають ***біграмою*** або ***диграфом***. Послідовність з L букв називають L-грамою.

Ідея поліграмних шифрів полягає в тому, що заміняють не окремі символи відкритого тексту, а послідовності кількох символів, або, інакше, L-грами. Тобто відкритий текст при шифруванні розбивають на L-грами, кожну з яких заміняють на якийсь символ чи групу символів.

Шифр Плейфейра (Plaifer algoritm). Це біграмний шифр [2]. Розглянемо його на прикладі шифрування тексту латинкою. Для шифрування в якості ключа використовують квадрат 5×5, у якому перемішано всі букви латинського алфавіту, крім букви j (j найрідше зустрічається в текстах латинкою). Наприклад, такий квадрат:

S	Y	D	W	Z
R	I	P	U	L
H	C	A	X	F
T	N	O	G	E
B	K	M	Q	V

Під час шифрування відкритий текст розбивають на біграми так, щоб вони не містили однакових букв. Якщо такі біграми утворюються, текст модифікують, але так, щоб не змінити зміст повідомлення.

Кожну біграму зашифровують за допомогою такого квадрата у двох таких випадках:

1. Якщо дві букви біграми не потрапляють до одного рядка чи стовпчика квадрата, то вони визначають (умовно) прямокутник, і їх заміняють буквами із протилежних вершин цього прямокутника.

Наприклад, біграма CE визначає прямокутник із вершинами на буквах CFEN. Тому букву C біграми заміняють на F, а букву E – на букву N. Отже, маємо заміну: $CE \rightarrow FN$. Інші приклади замін за цим правилом:

$SG \rightarrow WT$; $UK \rightarrow IQ$; $TL \rightarrow ER$.

2. Якщо дві букви біграми потрапляють до одного рядка чи стовпчика квадрата, тоді їх циклічно зсувають на одну позицію праворуч або донизу відповідно.

Наприклад: $CX \rightarrow AF$; $NE \rightarrow OT$; $PO \rightarrow AM$; $SB \rightarrow RS$.

Ключем шифру є розташування букв у квадраті. Для формування квадрата Плейфейра часто застосовували ключове слово, яке після вилучення повторів букв записували до квадрата зліва направо й згори донизу. Потім записували решту букв за алфавітом.

Відомі різні модифікації шифру Плейфейра. Наприклад, ключ можна сформувати у вигляді чотирьох квадратів, розмістивши їх теж у вигляді квадрата (схематично показано на рис. 3.5). Тоді першу букву біграми, що шифрують, фіксують у верхньому лівому квадраті, а другу – у нижньому правому. Ці дві букви (умовно 1, 2 – див. рис. 3.5) заміняють буквами з інших двох квадратів. Отримують заміну $12 \rightarrow 1'2'$.

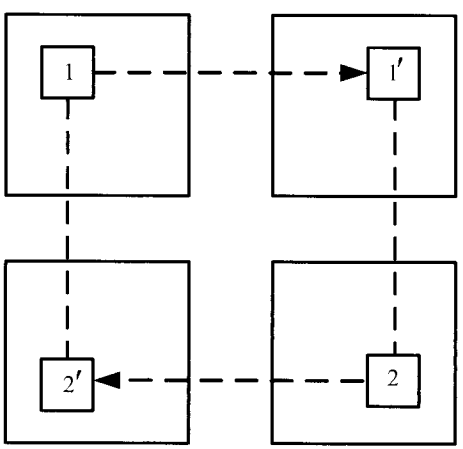


Рис. 3.5. Схема модифікованого шифру Плейфейра

Шифр Плейфейра розкривають аналізом частот біграм.

Поліграмні шифри з L=2,3,4 вразливі до частотного аналізу біграм, триграм, тетраграм відповідно.

3.3.4. Поліалфавітні криптосистеми

Усі розглянуті раніше шифри були одноалфавітними. У таких шифрах застосовують одну таблицю підстановок і завжди під час шифрування кожну символ чи групу символів замінюють однаково, не залежно від позиції цього символу в тексті.

Якщо для шифрування застосовують різні таблиці підстановок, застосування яких визначається позицією букви в тексті, такі шифри називають ***поліалфавітними*** [2]. Однією з найдавніших та найвідоміших поліалфавітних криптосистем є ***шифр Віженера*** (Vigenere cipher).

Шифр Віженера. Цей шифр подібний до шифру Цезаря, у якому ключ (величина зсуву) змінюють від кроку до кроку. Під час шифрування під відкритим текстом записують ключове слово, яке повторюють потрібну кількість разів. У результаті під усіма буквами відкритого тексту опиняються букви ключового слова. Номер букви ключового слова в алфавіті визначає кількість позицій, на які буде зсунуто відповідну букву відкритого тексту під час шифрування. Розглянемо приклад:

Відкритий текст: ПАРОЛЬАДЛЯАЛІСИДЛЯБОБАВ

Ключове слово: КЛЮЧ

Позиція букви К в алфавіті – 14, Л – 15, Ю – 31, Ч – 27. Тому букви відкритого тексту, які опинилися над буквою К ключового слова, будуть зсунуті на 14 позицій; ті, які над буквою Л – на 15; букви над буквою Ю – на 31 позицію, а ті, які над буквою Ч – на 27 позицій. Зсув здійснюють циклічно. Принцип шифрування та отриманий шифртекст для цього прикладу показано на рис. 3.6.

	П	А	Р	О	Л	Ь	А	Д	Л	Я	А	Л	І	С	И	Д	Л	Я	Б	О	Б	А	В
+	**К**	**Л**	**Ю**	**Ч**	К	Л	Ю	Ч	К	Л	Ю	Ч	К	Л	Ю	Ч	К	Л	Ю	Ч	К	Л	Ю
	А	Л	О	Ї	Щ	Ї	Ю	Я	Щ	К	Ю	З	Х	Г	Ж	Я	Щ	К	Я	Ї	Л	Л	А

Рис. 3.6. Приклад шифрування класичним методом Віженера

Шифрування за Віженером за описаним підходом є доволі трудомістким. Зручніше для шифрування користуватися таблицею Віженера (рис. 3.7).

Під час шифрування за допомогою таблиці Віженера буква ***відкритого тексту*** визначає рядок таблиці, а буква ***ключа*** – стовпчик (або навпаки). На перетині знаходять відповідну букву шифртексту. Розшифрування здійснюють так само. Фактично, таблиця Віженера для українського алфавіту відображає всі 33 варіанти таблиць підстановок шифру зсуву для всіх можливих ключів (величин зсуву).

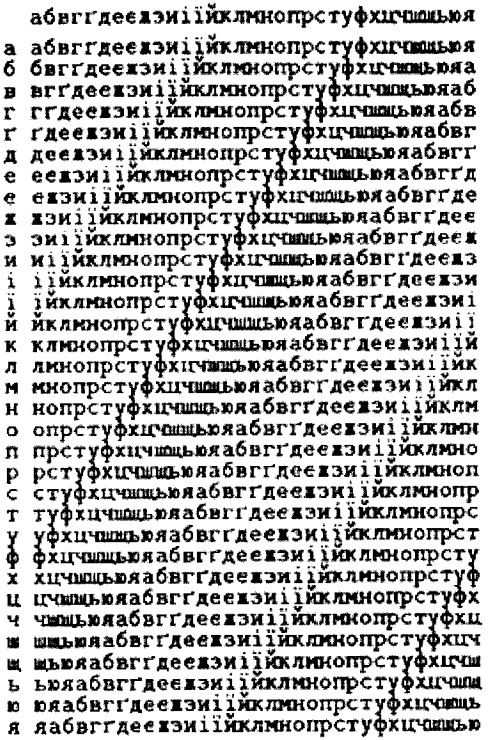


Рис. 3.7. Таблиця Віженера

Криптоаналіз шифру Віженера. Шифр Віженера можна розкрити за допомогою частотного аналізу, якщо відома довжина ключа [2]. Для цього шифртекст необхідно розбити на частини, кожну з яких шифрували своєю величиною зсуву. Це можна зробити, записавши шифртекст зліва направо згори до низу у таблицю, кількість колонок якої дорівнює довжині ключового слова. Кожну колонку такої таблиці шифрували своєю величиною зсуву, і її можна розкрити частотним аналізом.

Спосіб визначення невідомого періоду запропонував в 60-ті роки XIX століття німецький криптоаналітик ***Фрідріх Вільгельм Казіскі*** (Friedrich Wilhelm Kasiski). Він слідкував за появою в шифртексті однакових L-грам і вважав це ознакою того, що відстань між ними кратна довжині ключа (адже в такому випадку через повтори ключового слова в ключовій послідовності однаковим L-грамам шифртексту відповідають однакові L-грами відкритого тексту – див. рис. 3.6). Звичайно, це не завжди так. Поява однакових L-грам у шифртексті може бути випадковою, але з високою ймовірністю в доволі довгому шифртексті поява однакових L-грам насправді можна пояснити згаданими причинами.

Отже, відшукуючи в шифртексті однакові L-грами й припускаючи, що довжина ключа – це числа, які ділять націло відстань між цими L-грамами, випробовуючи частотний аналіз для кожного із цих чисел, шифр Віженера розкривають. Зокрема, у прикладі на рис. 3.6 криптолітик визначає відстань між однаковими триграмами перехопленого шифртексту "ЯЩК" – 8. Потім пробує частотний аналіз для довжини ключового слова 2 і результату не отримує. Потім пробує частотний аналіз для довжини ключового слова 4 і зламує шифр.

Шифр Віженера з автоключем. Шифрування здійснюють так само, як у шифрі Віженера, але ключову послідовність формують інакше. Ключове слово записують один раз, а після нього дописують початковий відрізок відкритого тексту [2].

Розглянемо приклад шифрування для того самого відкритого тексту й ключового слова, що й на рис. 3.7, але ключову послідовність сформуємо інакше (рис. 3.8):

	П	А	Р	О	Л	Ь	А	Д	Л	Я	А	Л	І	С	И	Д	Л	Я	Б	О	Б	А	В
+	**К**	**Л**	**Ю**	**Ч**	П	А	Р	О	Л	Ь	А	Д	Л	Я	А	Л	І	С	И	Д	Л	Я	Б
	А	Л	О	Ї	Б	Ь	Р	У	Ь	Щ	А	Р	Ц	Р	И	Р	Ц	Р	І	У	М	Я	Г

Рис. 3.8. Приклад шифрування методом Віженера з автоключем

На перший погляд метод Казізкі для цього шифру не працює, адже ключове слово не повторюється. Проте метод Казізкі й тут працює завдяки тому, що деякі слова в кожній мові є часто вживаними. Тому можна сподіватися, що відстань між ними в доволі довгому тексті хоча б раз виявиться кратною періоду, а це призведе до появи однакових L-грам у шифртексті, відстань між якими кратна довжині ключа. У прикладі на рис. 3.8 такими однаковими L-грамами є триграми "ДЛЯ". Посередині між ними розміщено триграму "ЛІС". Через особливості формування ключової послідовності під час шифрування під цією триграмою опиняється перша триграма "ДЛЯ", а під другою триграмою "ДЛЯ" відкритого тексту в ключовій послідовності розташується триграма "ЛІС". У шифрах Віженера триграму "ЛІС" із ключем "ДЛЯ" шифрують так само, як триграму "ДЛЯ" із ключем "ЛІС". Результатом шифрування в обох випадках є триграма "РЦР". Відстань між ними в шифртексті насправді кратна довжині ключа. Отже, шифр з автоключем також піддається криптоаналізу за методом Казізкі. За цим методом визначають довжину ключа, після чого ключове слово знаходять перебором (повний перебір піддається вдосконаленню).

Зауваження. Якщо застосовують шифрування блоками (тобто відкритий текст розбивають на блоки по L символів у кожному, які поодинці шифрують, то такі шифри називають ***блоковими з періодом*** L.

Шифр Віженера також можна розглядати як блоковий шифр із періодом, що дорівнює довжині ключового слова.

3.3.5. Шифри перестановки

Означення. Шифри перестановки зберігають усі букви відкритого тексту, розміщуючи їх в іншій послідовності [2].

Прикладом шифрів перестановки є шифр Скитала. Такі шифри називають інакше шифрами обходу.

Матричний шифр обходу. Повідомлення записують рядками у вигляді прямокутної таблиці, кількість колонок якої дорівнює довжині ключа. Шифртекст формують зчитуванням букв стовпчиками в послідовності, яку визначає ключ (відносна послідовність букв ключового слова визначає послідовність зчитування стовпчиків).

Приклад:

Відкритий текст: ОХОРОНУСКЛАДІВПОСЛАБЛЕНО

Ключ: КЛЮЧ

Зашифрування:

К	Л	Ю	Ч
1	2	4	3
О	Х	О	Р
О	Н	У	С
К	Л	А	Д
І	В	П	О
С	Л	А	Б
Л	Е	Н	О

Шифртекст: ООКІСЛХНЛВЛЕРСДОБООУАПАН

Загальний шифр перестановки з періодом L переставляє L букв у довільному порядку [1, 2]. Цей порядок визначає ключ. Ключ зручно задавати у вигляді таблиці. Таку таблицю називають ***таблицею перестановок***.

Наприклад, таблиця перестановок:

$$\begin{pmatrix} 1 & 2 & 3 & \cdots & L \\ i_1 & i_2 & i_3 & \cdots & i_L \end{pmatrix}$$

показує, що перша буква блока відкритого тексту займає позицію i_1 у відповідному блоці шифртексту, другу букву перемістять на позицію i_2 й так далі. Наприклад, якщо $i_2 = 5$, другу букву кожного блока відкритого тексту

перемістять на п'яту позицію у відповідних блоках шифртексту. Отже, на відміну від таблиць підстановок, у таблиці перестановок фігурують номери позицій символів, а не самі символи.

За невеликого періоду L шифр перестановки легко розкривають спеціально організованим аналізом частот біграм.

Приклад. Нехай перехоплено шифртекст [2]:

ІЦКАЗИВИМЯИЛКИНОЙРЕСРПІНЗМЕЛБОПИРПИИТИНИИВІСВИТАЛП

Відомо, що він отриманий шифром перестановки з періодом 5. Отже, його можна дешифрувати, розбивши на блоки по п'ять букв, записавши їх до таблиці й переставивши колонки таблиці в правильній послідовності (відповідно до невідомої таблиці перестановок). Отже, розіб'ємо перехоплений шифртекст на блоки по п'ять букв і запишемо ці блоки в прямокутну таблицю:

1	2	3	4	5
І	Ц	К	А	З
И	В	**И**	М	Я
И	Л	К	**И**	Н
О	Й	Р	Е	С
Р	П	І	Н	З
М	Е	Л	Б	О
П	**И**	Р	П	**И**
И	Т	**И**	Н	**И**
И	В	І	С	В
И	Т	А	Л	П

Для того щоб визначити правильну послідовність розміщення стовпчиків, можна використати інформацію про частоти біграм в українській мові. Наприклад, біграма "ИИ" ніколи в українській мові не зустрічається (є один виняток). Отже, під час перестановки стовпчиків таблиці не можна допустити, щоб букви "И" опинилися поруч. Тому при перестановці не можуть бути поруч перший та третій стовпчики таблиці (цей висновок робимо із другого рядка), перший і четвертий стовпчики (див. третій рядок), другий і п'ятий стовпчики (див. сьомий рядок), а також будь-які два з першого, третього та п'ятого стовпчиків (восьмий рядок). Таким обмеженням відповідають тільки дві послідовності цілих чисел від 1 до 5. Це послідовність 1, 2, 3, 4, 5 (яку маємо в проаналізованій таблиці), а також послідовність 5, 4, 3, 2, 1. Отже, остання послідовність і є правильним порядком розміщення стовпчиків. Переставляємо стовпчики й читаємо дешифрований текст зліва направо згори донизу:

1	2	3	4	5
І	Ц	К	А	З
И	В	**И**	М	Я
И	Л	К	**И**	Н
О	Й	Р	Е	С
Р	П	І	Н	З
М	Е	Л	Б	О
П	**И**	Р	П	**И**
И	Т	**И**	Н	**И**
И	В	І	С	В
И	Т	А	Л	П

$\Rightarrow$

5	4	3	2	1
З	А	К	Ц	І
Я	М	И	В	И
Н	И	К	Л	И
С	Е	Р	Й	О
З	Н	І	П	Р
О	Б	Л	Е	М
И	П	Р	И	П
И	Н	И	Т	И
В	С	І	В	И
П	Л	А	Т	И

Отже, у цьому прикладі розкритим ключем є така таблиця перестановок:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 \end{pmatrix}.$$

У загальному випадку беруть до уваги різні малоймовірні буквосполучення й формують систему обмежень на ключ. Якщо період невідомий, описаний метод пробують для різних періодів, поки не досягнуть успіху.

3.3.6. Кількаразове шифрування

Означення. Алгоритм, який шифрує повідомлення одним шифром, а потім результат шифрування – ще одним шифром, називають ***композицією*** або ***добутком*** двох шифрів [2].

Відомо, що композиція матричних шифрів обходу призводить до перестановки, яку важко задати саму по собі. Відомо також, що при компонуванні двох блокових шифрів із взаємно простими періодами, період добутку шифрів збільшується.

У роки Першої Світової війни з'явилися подрібнювальні шифрувальні системи. Найвідоміша з них – це шифр із назвою ADFGVX. Цей шифр застосовували в німецькій армії. Він є композицією двох шифрів: підстановки та перестановки. Спочатку кожен символ відкритого тексту зашифровували шифром заміни. Але заміняли його не одним, а двома символами. Тобто застосовували ***таблицю білітеральної підстановки-подрібнення***, наприклад, таку таблицю:

	A	D	F	G	V	X
A	C	O	8	X	F	4
D	M	K	3	A	Z	9
F	N	W	L	0	J	D
G	5	S	I	Y	H	U
V	P	1	V	B	6	R
X	E	Q	7	T	2	G

За допомогою такої таблиці кожен символ відкритого тексту – букву або десяткову цифру – замінювали на біграму, перша буква якої позначала рядок таблиці, а друга – стовпчик, на перетині яких знаходився символ, що шифрували. У результаті отримували проміжний шифртекст, удвічі довший за відкритий текст. До проміжного шифртексту застосовували звичайний матричний шифр обходу й отримували остаточний шифртекст.

Для свого часу це був дуже складний композиційний шифр, але й його зламав французький криптоаналітик Жорж Пенвен.

3.3.7. Роторні шифрувальні машини

У докомп'ютерні часи для автоматизації процесу шифрування застосовували різні механічні або електромеханічні шифрувальні засоби. В основу конструкції більшості з них покладено концепцію ротора – механічного колеса, що використовували для виконання підстановок.

Роторна машина складається із клавіатури й набору зв'язаних між собою роторів. Вона здійснює варіант шифру Віженера. На кожному роторі розміщено букви алфавіту. Він має 26 фіксованих позицій і може виконувати просту підстановку. Вихідні штирі одного ротора з'єднано з вхідними штирями наступного. Взаємне розміщення роторів можна змінювати.

Найвідоміший роторний пристрій – машина Енігма (див. рис. 3.2), яку застосовувала Німеччина під час Другої Світової війни. Вона мала 3 ротори, які вибирали з набору п'яти роторів. Крім того, вона мала комутатор, який перетасовував (перестановка) відкритий текст, а також відображальний ротор, який примушував кожен ротор двічі обробляти вхідний текст (див. рис. 3.9). Як відомо, незважаючи на складність

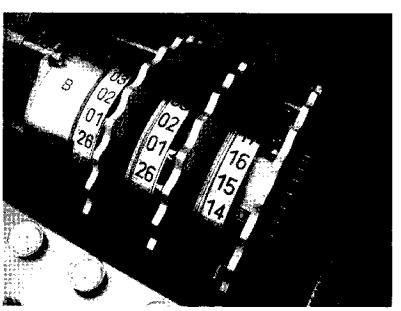


Рис. 3.9. Три з'єднаних ротори Енігми та відображальний ротор (позначений буквою B)

задачі, під час війни шифр було зламано (дешифрувальна машина "Bombe" Тюрінга, див. рис. 3.3).

3.4. Афінні шифри

Подання тексту в цифровій формі. У докомп'ютерні часи під час шифрування не було необхідності змінювати алфавіт відкритого тексту. Шифрування здійснювали в символьному вигляді, або вручну, або за допомогою механічних шифрувальних засобів. Якщо ж спробувати скористатися сучасними засобами передавання даних або здійснити шифрування за допомогою комп'ютера, виявиться, що звичайний алфавіт є незручною формою представлення, оброблення й передавання інформації. Зручнішим є подання інформації в цифровій формі. Існують різні способи подання інформації в цифровій формі (способи первинного кодування). У криптології найпростішим і достатньо розповсюдженим є спосіб, за яким кожну букву заміняють її номером за алфавітом. Нумерацію, як правило, починають із нуля. Наприклад, слово "банан" у цифровій формі за допомогою десяткового коду (два розряди на букву) можна представити як 0100170017. Номери букв можна записати не в десятковій, а в двійковій системі числення. У цьому випадку для кодування всіх 33 букв українського алфавіту необхідні шість бітів. За таких умов двійковий код слова "банан" буде таким: 000001000000010001000000010001. Тобто кожен блок із шести бітів є номером відповідної букви в алфавіті. Усю послідовність бітів називають двійковим словом. Нарешті, подібно можна закодувати не лише букви, але й розділові знаки, букви інших алфавітів, службові символи тощо. Наприклад, двійковим 8-бітним блоком можна закодувати 256 різних символів, достатніх для передавання практично будь-якої інформації (ASCII-кодування).

Афінні шифри. Афінні шифри – це підклас шифрів заміни, що містить як часткові випадки шифри зсуву, Віженера, перестановки з фіксованим періодом тощо [2].

Зауваження. Моноалфавітні k-грамні шифри заміни можна означити як блокові шифри з періодом k. Відповідно шифр простої заміни можна трактувати як блоковий шифр із періодом 1.

Правило. n-символьний алфавіт ототожнюють із кільцем цілих чисел Z_n.

Отже, кожну букву алфавіту відкритого тексту заміняють її номером за алфавітом. Нумерацію починають із нуля. Наприклад, український алфавіт ототожнюють з Z_{33}, латинський – із Z_{26}.

Після прийняття такого правила до букв відкритого тексту можна застосовувати операції додавання та множення за відповідним модулем.

Отже, далі n – кількість букв в алфавіті відкритого тексту.

Афінний шифр зсуву першого порядку

Ключ: число s таке, що $0 \le s < n$.

Зашифрування: Кожну букву x відкритого тексту (нагадаємо, що тут і далі під поняттям "буква" маємо на увазі номер букви за алфавітом) заміщують буквою y шифртексту ($x \to y$) за таким правилом:

$$y = E(x) = (x + s) \bmod n .$$

Розшифрування: $y \to x$:

$$x = D(y) = (y + s') \bmod n .$$

Тут $s' = n - s$ (величина зворотного зсуву) – ключ розшифрування.

Тобто шифрування, яке в символьному варіанті шифру зсуву виконували пересовуванням букв по алфавіту або за допомогою таблиці підстановок, за цим цифровим варіантом шифру зсуву виконують додаванням до номера букви відкритого тексту величини зсуву. Додавання модульне, тобто циклічне. Якщо в результаті додавання отримують число, яке виходить за межі кільця, виконують модульну редукцію (переходять на початок алфавіту).

Афінний лінійний шифр першого порядку

Ключ: число a таке, що $0 < a < n$, $НСД(a,n) = 1$.

Виконання вимоги до ***найбільшого спільного дільника*** (НСД) чисел a і n гарантує наявність числа, оберненого до a за модулем n, а отже, гарантує можливість розшифрування інформації.

Зашифрування: $x \to y$:

$$y = E(x) = (a \cdot x) \bmod n .$$

Розшифрування: $y \to x$:

$$x = D(y) = (a' \cdot y) \bmod n ,$$

де $a' = a^{-1} \bmod n$ – ключ розшифрування.

Зауваження: НСД та обернене число за модулем ($a' = a^{-1} \bmod n$) обчислюють за класичним та розширеним алгоритмом Евкліда (див. розділ 2 посібника).

Узагальнений афінний шифр першого порядку

Цей шифр є узагальненням шифру зсуву й лінійного шифру.

Ключ: числа a та s такі, що $0 \le s < n$, $0 < a < n$, $НСД(a,n) = 1$.

Зашифрування: $x \to y$:

$$y = E(x) = (a \cdot x + s) \bmod n .$$

Розшифрування: $y \to x$:

$$x = D(y) = (a' \cdot y + s') \bmod n,$$

де $a' = a^{-1} \bmod n$, і $s' = (-a' \cdot s) \bmod n$ – ключі розшифрування.

Афінні шифри вищих порядків

Розглянуті афінні шифри першого порядку були монограмними. Аргументом функції шифрування був номер однієї букви. Розширимо тепер розглянуті монограмні афінні шифри на шифрування k-грам.

Введемо операцію додавання в Z_n^k, що є множиною векторів розміру k із коефіцієнтами з кільця Z_n. Такі вектори є цифровими відображеннями k-грам у n-буквеному алфавіті.

Сумою векторів $X = (x_1, x_2, \cdots, x_k)$ і $S = (s_1, s_2, \cdots, s_k)$, які належать Z_n^k, є вектор $X + S$ із такими коефіцієнтами:

$$X + S = ((x_1 + s_1) \bmod n, \cdots, (x_k + s_k) \bmod n).$$

Вектор $-S = (n - s_1, n - s_2, \cdots, n - s_k)$ є оберненим до S відносно операції додавання.

Афінний шифр зсуву *k*-го порядку (або шифр Віженера з періодом *k*)

Ключ: вектор $S \in Z_n^k$.

Зашифрування: відкритий текст розбивають на k-грами. Кожну k-граму X заміщують k-грамою Y ($X \to Y$):

$$Y = E(X) = X + S.$$

Розшифрування: $Y \to X$:

$$X = D(Y) = Y + S',$$

де $S' = -S$ – ключ розшифрування.

Цей шифр є цифровим аналогом класичного шифру Віженера.

Лінійний афінний шифр *k*-го порядку (криптосистема Хілла)

Через $M_k(Z_n)$ позначимо множину всіх можливих матриць розміру $k \times k$ із коефіцієнтами з кільця Z_n. Через $GL_k(Z_n)$ позначимо підмножину оборотних матриць, яка об'єднує тільки ті матриці з $M_k(Z_n)$, до яких існують обернені.

Для деякої матриці A, яка належить $GL_k(Z_n)$, обернену до неї матрицю позначимо A^{-1}.

Добутком AX матриці $A = (a_{ij}) \in M_k(Z_n)$ на вектор-стовпчик $X = (x_1, x_2, \cdots, x_k) \in Z_n^k$ є вектор-стовпчик:

$$A \cdot X = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ a_{k1} & a_{k2} & & a_{kk} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{pmatrix} = \begin{pmatrix} (a_{11}x_1 + a_{12}x_2 + \cdots a_{1k}x_k) \bmod n \\ (a_{21}x_1 + a_{22}x_2 + \cdots a_{2k}x_k) \bmod n \\ \vdots \\ (a_{k1}x_1 + a_{k2}x_2 + \cdots a_{kk}x_k) \bmod n \end{pmatrix}.$$

Ключ: матриця $A \in GL_k(Z_n)$ така, що $НСД(\omega, n) = 1$ (умова оборотності матриці; ω – визначник матриці).

Зашифрування: $X \to Y$:

$$Y = E(X) = A \cdot X.$$

Розшифрування: $Y \to X$:

$$X = D(Y) = A' \cdot Y',$$

де $A' = A^{-1}$ – ключ розшифрування.

Узагальнений афінний шифр *k*-го порядку

Ключ: матриця $A \in GL_k(Z_n)$ така, що $НСД(\omega, n) = 1$; вектор $S \in Z_n^k$.

Зашифрування: $X \to Y$:

$$Y = E(X) = A \cdot X + S.$$

Розшифрування: $Y \to X$:

$$X = D(Y) = A' \cdot Y + S',$$

де $A' = A^{-1}$, $S' = -A' \cdot S$ – ключі розшифрування.

3.5. Потокові симетричні шифри

Алгоритм звичайного XOR. XOR – це логічна операція "сума за модулем 2". Результати такої операції для двох бітів є такі:

$$0 \oplus 0 = 0,$$
$$0 \oplus 1 = 1,$$
$$1 \oplus 0 = 1,$$
$$1 \oplus 1 = 0.$$

Для такої операції справедливі, зокрема, такі тотожності:

$$a \oplus a = 0; a \oplus b \oplus b = a.$$

Алгоритм XOR [1] часто застосовували в комерційних програмних продуктах, зокрема в мережах мобільного зв'язку. Проте згодом було виявлено, що рівень захисту алгоритму практично дорівнює нулю.

При ***генеруванні*** шифртексту відкритий текст за допомогою операції XOR додають побітово із ключем. Якщо ключ коротший за повідомлення, то його повторюють необхідну кількість разів. Оскільки повторне застосування операції

XOR відновлює оригінал, зашифрування та розшифрування виконують однією й тією самою програмою:

$$C = M \oplus K,$$
$$M = C \oplus K = (M \oplus K) \oplus K = M \oplus K \oplus K = M.$$

Криптоаналіз алгоритму XOR

1. Визначають довжину ключа за допомогою процедури, яку називають ***"підрахунок збігів"***. Для цього операцію XOR застосовують до шифртексту, використовуючи замість ключа сам шифртекст із різними зсувами.
2. Підраховують кількість байтів, які збігаються з байтами шифртексту (у випадку застосування ASCII кодування). Якщо величина зсуву кратна довжині ключа, збігається понад 6% байтів (***індекс збігу***). Якщо ні, тоді менше ніж 0,4%. Мінімальне зміщення, яке забезпечує високий індекс збігу, і є довжиною ключа.
3. Після визначення довжини ключа криптоаналітик зміщує шифртекст на величину ключа й додає за модулем 2 до незміщеного шифртексту. Результатом є ліквідація ключа й отримання відкритого тексту, на який операцією XOR накладено той самий, але зміщений відкритий текст.

Для інших кодувань процедура й значення індексу збігу дещо інші, але принцип розкриття аналогічний.

Шифр одноразового блокнота. Абсолютно стійкий шифр. Якщо в алгоритмі XOR замість ключа у вигляді повтореного n разів ключового слова взяти випадкову послідовність бітів такої самої довжини, що й відкритий текст, отримаємо варіант ***алгоритму одноразового блокнота*** [1, 2]. Тобто в бінарному варіанті шифрування одноразовим блокнотом здійснюють накладанням на відкритий текст деякої довжини ключової послідовності строго випадкових бітів такої самої довжини. Накладання здійснюють за допомогою операції XOR. Розшифрування виконують накладанням на шифртекст тієї самої ключової послідовності:

$$C = M \oplus K,$$
$$M = C \oplus K = (M \oplus K) \oplus K = M \oplus K \oplus K = M.$$

Операцію зашифрування можна здійснювати й над символьною інформацією. У цьому випадку додавання здійснюють за модулем n (n – кількість букв в алфавіті).

Зауваження. Якщо в шифрі Віженера вибрати як ключ строго випадкову послідовність букв такої самої довжини, що й повідомлення, тоді також отримаємо шифр одноразового блокнота.

Шифр одноразового блокнота винайшли у 1917 році ***Гілберт Вернам*** (Gilbert Sandford Vernam) та ***Мейджор Джозеф Моборн*** (Major Joseph Mauborn). Цей шифр ***абсолютно стійкий*** (або, інакше, ***стійкий у теоретико-інформаційному сенсі***). Його теоретично неможливо розкрити жодних відомими й поки що не відомими методами. Гарантією абсолютної стійкості, як показав ***Клод Елвуд Шеннон*** (Claude Elwood Shannon), є ***довжина ключа***, його ***строга випадковість***, а також ***однократність застосування***. Шеннон порівнював застосування цього шифру з накладанням на корисний сигнал у лінії зв'язку білого шуму (тобто випадкового шуму такої самої інтенсивності, що й корисний сигнал), після чого стає неможливим виділення корисного сигналу на приймальному боці.

Отже, шифр одноразового блокнота неможливо розкрити навіть брутальною атакою. Адже якщо ключ такий самий завдовжки, що й повідомлення, то випробовуючи всі можливі ключі, ми отримаємо повну множину всіх можливих відкритих текстів, зокрема підмножину всіх можливих ***змістовних відкритих текстів***. Причому криптоаналітик не має жодної додаткової інформації про те, яким саме змістовний відкритий текст був під час зашифрування.

Наприклад, зашифруємо слово ***"банан"*** шифром одноразового блокнота.

Подамо слово "банан" двійковим словом M:

$$\text{банан} \to M = 000001\ 000000\ 010001\ 000000\ 010001.$$

Далі вибираємо випадковий двійковий ключ K такої самої довжини. Наприклад:

$$K = 001101\ 110101\ 100010\ 011000\ 111010.$$

Додаємо M і K за модулем 2 і отримуємо шифртекст:

$$C = 001100\ 110101\ 110011\ 011000\ 101011.$$

Перехопивши шифртекст C, криптоаналітик методом брутальної атаки рано чи пізно знайде ключ K і прочитає перехоплений текст "банан". Але з такою імовірністю він може натрапити на ключ K':

$$K' = 001111\ 100001\ 100100\ 000100\ 101011$$

і прочитати перехоплений текст як "груша". Так само з іншими ключами він може отримати інші змістовні слова з п'яти букв. Криптоаналітик не має додаткової інформації про те, який розшифрований текст є більш імовірним, ніж інші.

Недоліком шифру одноразового блокнота є складність процедури обміну ключем. Ключ довгий, таємний, і його необхідно застосовувати лише одного разу. Відповідно має існувати надійний канал передавання таємного ключа. Сама назва шифру пішла від того, що дві сторони каналу зв'язку мали однаковий таємний блокнот із ключовою послідовністю. В одному сеансі

зв'язку вони використовували необхідну кількість бітів з однієї й тієї самої сторінки блокнота в якості ключової послідовності. Після завершення сеансу відповідний листок блокнота знищували й у наступному сеансі для формування ключа використовували наступний листок блокнота.

Іншим недоліком є необхідність синхронізації приймання та передавання. Якщо приймач втратить хоча б один біт шифртексту, він не зможе розшифрувати текст.

Ще одним недоліком шифру є те, що він не забезпечує захист даних від спотворення.

Незважаючи на ці недоліки, абсолютна стійкість є унікальною властивістю цього шифру.

Потокові шифри. Шифр звичайного XOR та бінарний варіант шифру одноразового блокнота належать до класу ***потокових шифрів*** [1, 2].

Означення. Потоковим називають шифр, який двійкове повідомлення $M = m_1 m_2 \cdots m_i \cdots m_l$ (де m_i – i-й біт повідомлення), з використанням двійкового ключа такої ж довжини $K = k_1 k_2 \cdots k_i \cdots k_l$ перетворює в шифртекст $C = c_1 c_2 \cdots c_i \cdots c_l$ побітовим додаванням M та K за модулем 2:

$$c_i = (m_i + k_i) \bmod 2 \text{ або } c_i = m_i \oplus k_i, \text{ або } C = M \oplus K .$$

Різновиди потокових шифрів відрізняються за способом формування ключа. Наприклад, у шифрі одноразового блокнота необхідно згенерувати довгу послідовність дійсно випадкових бітів ключа. Процес генерування такої послідовності є складним і тривалим. Тому через складність процедури генерування та обміну великих випадкових таємних чисел замість випадкової на практиці часто застосовують так звану ***псевдовипадкову ключову послідовність***.

Ключову послідовність потокових шифрів часто називають ***гама-послідовністю***, а потокове шифрування інакше називають ***гамуванням***.

Означення. ***Генератор псевдовипадкових бітів*** – це пристрій або алгоритм [2], який за заданими параметрами генерує послідовність бітів. При цьому значення кожного чергового i-го біта, не знаючи параметрів, неможливо передбачити з імовірністю, більшою за 0,5. Знаючи параметри, значення кожного чергового i-го біта можна передбачити з імовірністю 1.

Отже, у випадку застосування псевдовипадкової ключової послідовності користувачам не потрібно обмінюватися довгим ключем. Вони обмінюються таємними параметрами однакових генераторів. Після цього вони можуть згенерувати однакові псевдовипадкові ключові послідовності. Таємні значення параметрів власне і є ключем. Це, як правило, справжня випадкова послідовність бітів, однак значно коротша за можливі довжини повідомлень. Таку випадкову послідовність називають ***паростком***, який породжує гаму.

Стійкість потокових шифрів із псевдовипадковою гамою повністю залежить від якості й криптографічної стійкості ***генератора псевдовипадкових чисел*** (ГПВЧ).

Основні показники роботи генераторів псевдовипадкових чисел. До таких показників належать ***період генератора***, його ***лінійна складність*** та ***кореляційний імунітет***.

Більшість генераторів псевдовипадкових чисел є скінченними автоматами, які, пройшовши через усі свої можливі стани (у кращому випадку), починають генерувати вже згенеровану послідовність спочатку. Винятком є генератори на основі обчислювачів ірраціональних чисел, послідовність цифр яких, як вважають, є неперіодичною псевдовипадковою послідовністю. Псевдовипадкову послідовність можна використовувати як ключ тільки до повтору, інакше потоковий шифр виродиться в нестійкий алгоритм звичайного XOR. Тобто, бажано, щоб період генератора був якомога більшим.

Існує алгоритм, який після аналізу певного скінченного числа бітів перехопленої гами дає змогу побудувати генератор на базі одного регістра зсуву з лінійним зворотним зв'язком, який генерує таку саму послідовність. Довжину такого еквівалентного генератора називають ***лінійною складністю*** ГПВЧ. Якщо довжина еквівалентного регістра виявиться невеликою, генератор можна легко зламати. Тому при проектуванні генераторів необхідно забезпечити достатню лінійну складність.

З метою покращення характеристик генераторів часто застосовують різні способи комбінування простіших генераторів. Ризики такого підходу полягають у тому, що згенерована псевдовипадкова послідовність може виявитися залежною від виходу одного із блоків генератора. Якщо криптоаналітик цю залежність виявить, він зламає спочатку простіший блок, а потім і весь генератор. Тому при проектуванні комбінованих генераторів необхідно забезпечити їх кореляційний імунітет.

Способи проектування генераторів псевдовипадкових чисел. Існують чотири основні підходи до проектування ГПВЧ [1]:

– ***системно-теоретичний підхід***, за яким, використовуючи низку фундаментальних критеріїв і законів проектування, криптограф намагається впевнитися, що розроблений генератор створює складну проблему для криптоаналізу;

– ***інформаційно-теоретичний підхід,*** має на меті, що буде мати місце невпевненість криптоаналітика відносно відкритого тексту. Криптоаналітик не може отримати однозначного результату дешифрування. Тобто тут ідеться, очевидно, про шифр одноразового блокнота;

– ***складнісно-теоретичний підхід.*** Тут як основу для побудови генератора використовують деяку відому й складну математичну задачу (наприклад, розкладання на множники або обчислення дискретних логарифмів);

– ***рандомізований підхід.*** В межах цього підходу намагаються створити проблему криптоаналітику, примушуючи його перевіряти в ході криптоаналізу надто великий об'єм даних.

Системно-теоретичний підхід до проектування потокових шифрів. Результатом застосування цього підходу є більшість реальних потокових шифрів, наприклад, шифри на базі регістрів зсуву з лінійними зворотними зв'язками. Тут криптограф розробляє генератори, які володіють конкретними характеристиками, які можна перевірити: періодом, статистичним розподілом згенерованих бітів, лінійною складністю тощо. Такі шифри, як правило, не основані на складних математичних задачах. Криптограф також вивчає різні методи криптоаналізу розроблених генераторів і перевіряє, чи стійкі генератори щодо відомих способів розкриття.

У межах цього підходу було сформульовано критерії проектування потокових шифрів:

– довгий період без повторень;

– критерій лінійної складності: велика лінійна складність, лінійний профіль складності, локальна лінійна складність тощо;

– статистичні критерії, наприклад, ідеальні k -вимірні розподіли бітів;

– критерій перемішування: кожен біт гами має бути складним перетворенням усіх або більшості бітів ключа;

– критерій розсіювання: надлишковість у підмножинах бітів повинна розсіюватися на всю згенеровану послідовність;

– критерії нелінійності для логічних функцій, такі як відсутність кореляції m -го порядку, відстань до лінійних функцій, лавинний критерій тощо.

Цей перелік критеріїв проектування не унікальний для потокових шифрів, розроблених за допомогою системно-теоретичного підходу. Він стосується всіх потокових шифрів. Він також справедливий і для всіх блокових шифрів. Особливістю системно-теоретичного підходу є те, що потокові шифри безпосередньо розробляють, щоб задовольнити ці критерії.

Проблемою такого підходу є неможливість довести стійкість розроблених генераторів. Тобто не доведено, що перелічених критеріїв проектування необхідно й достатньо для безпеки генератора. Генератор гами може задовольняти всі критерії і все ж виявитися нестійким. З іншого боку, розкриття будь-якого з таких генераторів є окремою проблемою для криптоаналітика.

Поширеними ГПВЧ, які розробляють у межах системно-теоретичного підходу, є генератори на ***регістрах зсуву з лінійним зворотним зв'язком*** (linear

feedback shift register – LFSR) [1] (рис. 3.10). Зворотний зв'язок є функцією XOR деяких бітів регістра. Перелік цих бітів називають ***послідовністю відводів*** (tap sequence). Завдяки простоті схеми зворотного зв'язку для аналізу LFSR можна використовувати доволі розвинену математичну теорію.

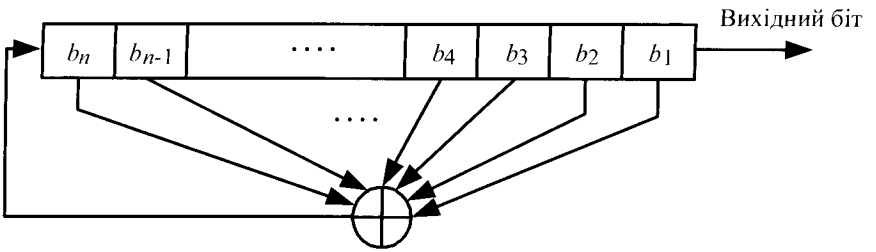


Рис. 3.10. Регістр зсуву з лінійним зворотним зв'язком

n-бітовий LFSR може знаходитися в одному із $2^n - 1$ внутрішніх станів. Це означає, що теоретично такий регістр може генерувати псевдовипадкову послідовність із періодом $2^n - 1$ бітів. Однак відомо, що тільки за деяких послідовностей відводів LFSR циклічно пройде через усі $2^n - 1$ внутрішні стани. Послідовність максимального періоду, згенеровану LFSR, називають ***M-послідовністю***.

Для того щоб конкретний LFSR мав максимальний період, многочлен, утворений із послідовності відводів регістра, має бути примітивним за модулем 2. Степінь многочлена збігається з довжиною регістра. У загальному випадку не існує простого способу генерування примітивних многочленів заданого степеня за модулем 2. Тому многочлен вибирають випадково і перевіряють, чи не є він примітивним. Існують також таблиці многочленів різних степенів, примітивних за модулем 2. Наприклад, запис у такій таблиці (32, 7, 5, 3, 2, 1, 0) означає, що многочлен $x^{32} + x^7 + x^5 + x^3 + x^2 + x + 1$ є примітивним за модулем 2. Тому він може задати послідовність відводів LFSR, який генерує послідовність бітів максимального періоду. При цьому перше число (старший степінь многочлена) указує на довжину LFSR. Останнє число завжди дорівнює 0, і його можна ігнорувати. Усі числа, за винятком 0, задають послідовність відводів, що відлічують від лівого краю регістра. Тобто члени многочлена з меншим степенем відповідають позиціям ближче до правого краю регістра.

Отже, запис (32, 7, 5, 3, 2, 1, 0) означає, що у вибраному 32-бітовому регістрі зсуву новий біт генерують за допомогою XOR тридцять другого, сьомого, п'ятого, третього, другого й першого бітів (рис. 3.11) [1]. Такий LFSR матиме максимальний період, циклічно проходячи до повторення через 2^{32}-1 станів.

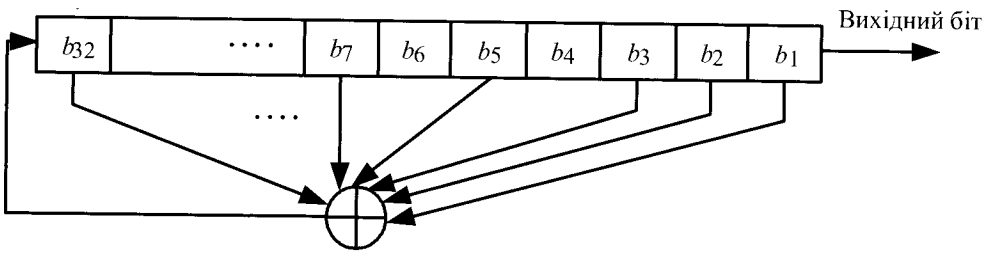


Рис. 3.11. 32-бітний LFSR із максимальним періодом

Самі по собі LFSR є статистично добрими генераторами псевдовипадкових послідовностей. Однак, з погляду криптографічної стійкості, вони мають низки небажаних властивостей, зокрема: послідовні біти лінійні, що робить їх не придатними для шифрування; для LFSR довжини n його внутрішній стан визначає наступні n вихідних бітів генератора; навіть якщо схему зворотного зв'язку зберігають у секреті, її можна визначити за допомогою ***високоефективного алгоритму Берлекемпа–Мессі*** (Berlekamp–Massey algorithm).

Разом із тим більшість реальних потокових шифрів основана на LFSR. Але такі шифри будують із використанням кількох LFSR, комбінуючи їх за певними правилами. Є два основні способи комбінування LFSR. Перший із них полягає в нелінійному об'єднанні виходів кількох LFSR (***нелінійне змішування***), а другий називають ***керованим тактуванням***. Суть другого способу в тому, що генератор будують на декількох LFSR, причому виходи (або відводи) одних LFSR є входами інших, або один LFSR може керувати власним тактуванням (фільтрувальні генератори). Два згадані способи, своєю чергою, також можна комбінувати під час проектування ГПВЧ.

Складнісно-теоретичний підхід до проектування потокових шифрів. За цим підходом криптограф використовує теорію складності, щоб довести, що його генератори стійкі [1]. Отже, генератори мають бути якомога складнішими, ґрунтуючись на тих самих складних математичних проблемах, що й асиметрична криптографія. І подібно до асиметричних шифрів, такі генератори, як правило, є повільними й складними.

Розглянемо як приклад використання складнісно-теоретичного підходу простий та ефективний ***алгоритм Блюма – Блюма – Шуба*** (Algorithm Blum – Blum – Shub – BBS), який називають генератором квадратичних лишків [1].

Під час побудови генератора спершу вибирають два прості числа p і q, які конгруентні (тобто порівнянні – див. розділ 2) з $3 \bmod 4$. Добуток цих чисел n є цілим числом Блюма. Потім вибирають інше випадкове ціле число x, взаємно просте з n. Обчислюють стартове число генератора:

$$x_0 = x^2 \bmod n.$$

Далі можна починати генерувати псевдовипадкові біти. i-м бітом є молодший значущий біт числа x_i:

$$x_i = x_{i-1}^2 \bmod n.$$

Безпека цієї схеми основана на складності розкладання n на множники. Можна опублікувати n, так що хто завгодно може генерувати біти за допомогою генератора. Проте поки криптоаналітик не розкладе n на множники, він не зможе передбачити вихід генератора. Більш того, генератор BBS ***непередбачуваний ліворуч*** й ***непередбачуваний праворуч***. Це означає, що, отримавши частину згенерованої послідовності, криптоаналітик не зможе передбачити ні наступного, ні попереднього біта послідовності.

Як і більшість генераторів, розроблених за складнісно-теоретичним підходом, цей алгоритм повільний, але є способи його прискорити. Виявляється, що як псевдовипадкові біти можна використовувати декілька бітів кожного x_i. Якщо n – довжина числа x_i у бітах, можна використовувати $\log_2 n$ молодших значущих бітів x_i. Через свою повільність генератор BBS не підходить для потокових шифрів. Проте для таких застосувань, як генерування ключів, цей генератор один із кращих.

3.6. Блокові симетричні шифри

3.6.1. Методи компонування сучасних блокових симетричних шифрів

Двома основними методами компонування сучасних симетричних шифрів є методи маскування надлишковості відкритого тексту, якими, за Шенноном, є перемішування й розсіювання [1].

Перемішування маскує зв'язок між відкритим текстом і шифртекстом. Воно ускладнює спроби знайти у шифртексті надлишковість і статистичні закономірності. Простим способом перемішування є ***підстановка***. У шифрі простої підстановки, наприклад, шифрі Цезаря, усі однакові букви відкритого тексту замінюють іншими однаковими буквами шифртексту. У сучасних шифрах підстановка є складнішою (багатоалфавітною): довгий блок відкритого тексту замінюють блоком шифртексту, а спосіб заміни блока змінюють із кожним бітом відкритого тексту або ключа.

Розсіювання (дифузія) розсіює надлишковість відкритого тексту, поширюючи її по всьому шифртексту. Простим способом здійснити розсіювання є ***перестановка*** (транспозиція). Простий перестановочний шифр тільки пере-

ставляє букви відкритого тексту. Сучасні шифри також виконують таку перестановку, але вони також використовують інші форми розсіювання, які дають можливість розкидати частини повідомлення по всьому повідомленню, що сприяє виникненню так званого ***лавинного ефекту***.

Потокові шифри, як правило, використовують тільки перемішування. Блокові алгоритми застосовують і перемішування, і розсіювання. Переважно, розсіювання саме по собі нескладно зламати. Проте в поєднанні (особливо багатократному) з перемішуванням воно покращує характеристики шифру.

Розглянуті шеннонівські принципи перемішування й розсіювання до сьогодні залишаються наріжним каменем проектування якісного блокового шифру.

Отже, перемішування слугує для маскування взаємозв'язків між відкритим текстом, шифртекстом і ключем. Навіть незначну залежність між цими трьома поняттями можна використати під час диференційного й лінійного криптоаналізу.

Розсіювання поширює вплив окремих бітів відкритого тексту на якомога більшу частину шифртексту. Це також маскує статистичні взаємозв'язки й ускладнює криптоаналіз.

Узагалі кажучи [1], для безпеки шифру достатньо одного перемішування. Алгоритм, що складається з єдиної, залежної від ключа таблиці відповідності 128 бітів відкритого тексту 128 бітам шифртексту, був би достатньо стійким. Проблема в тому, що для зберігання такої таблиці необхідно надто багато пам'яті: більше за 10^{40} байтів. Тому сучасні симетричні шифри структурно є значно складнішими за таку таблицю. Вони поєднують принципи перемішування та розсіювання, використовують багатократне оброблення (принцип ітеративного шифру), але завдяки цьому дають змогу обмежитися доступними ресурсами, зберігаючи при цьому достатню стійкість.

Отже, основний прийом при проектуванні сучасних симетричних блокових шифрів полягає в тому, щоб в одному шифрі в різних комбінаціях періодично чергувати перемішування (з порівняно невеликими таблицями підстановок) і розсіювання. Такий шифр називають ***складеним шифром***. Інколи блоковий шифр, який містить послідовні перестановки й підстановки, називають ***мережею перестановок – підстановок*** (substitution – permutation network – SP) або ***SP-мережею***.

Ще одним прийомом проектування є принцип ***ітеративного блокового шифру***. Цей принцип передбачає, що порівняно просту функцію одного етапу перетворення буде послідовно використано декілька разів.

Мережі Фейстеля. Одним з найпоширеніших способів побудови блокових шифрів є використання ***мереж Фейстеля*** (Feistel network), названих

іменем дослідника, який працював свого часу в корпорації ***IBM*** (International Business Machines) і був одним з авторів стандарту ***DES*** (Data Encryption Standard). Мережею Фейстеля є загальний метод перетворення за допомогою довільної функції (яку позначають ***F-функцією)*** у перестановку на множині блоків. Цю конструкцію, яку винайшов ***Хорст Фейстель***, було використано в багатьох шифрах, зокрема в DES і ГОСТ 28147-89. F-функцію, що є основним будівельним блоком мережі Фейстеля, завжди вибирають нелінійною й практично у всіх випадках необоротною.

Розглянемо принцип побудови мережі (петлі) Фейстеля. Візьмемо блок завдовжки n та поділимо його на дві половини L і R (ліву та праву) завдовжки $n/2$. Результат перетворення мережею Фейстеля на i-му етапі можна визначити так:

$$L_i = R_{i-1},$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K),$$

де K – підключ, який застосовують на i-му етапі, а f – довільна функція етапу.

Структуру однієї ітерації мережі Фейстеля подано на рис. 3.12.

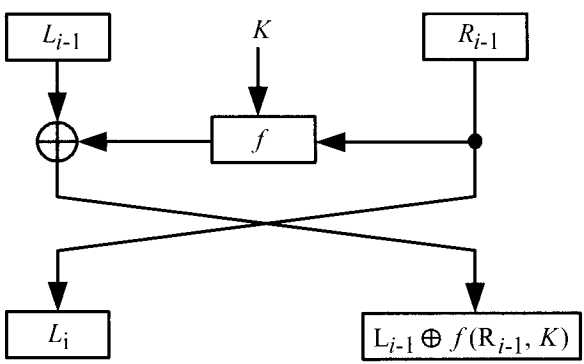


Рис. 3.12. Структура мережі Фейстеля

Подібна структура шифрів має важливу перевагу, а саме: процедури шифрування й розшифрування збігаються, за винятком того, що ключову інформацію при розшифруванні використовують у зворотній послідовності. Відповідно, будуючи пристрої шифрування, можна використовувати однакові блоки в колах шифрування й розшифрування.

Недоліком є те, що на кожній ітерації перетворюють лише половину блока оброблюваного тексту, що зумовлює необхідність збільшувати кількість ітерацій для досягнення необхідної стійкості.

Стосовно вибору F-функції якихось чітких стандартів не існує, проте, як правило, ця функція є послідовністю залежних від ключа нелінійних підстановок, розсіювальних перестановок і зсувів.

SP-мережі. Іншим підходом до побудови блокових шифрів є використання оборотних, залежних від ключа перетворень. У цьому випадку на кожній ітерації змінюють весь блок і, відповідно, загальну кількість ітерацій можна скоротити. Кожна ітерація є послідовністю перетворень ("шарів"), кожне з яких виконує свою функцію. Зазвичай використовують шар нелінійної оборотної підстановки, шар лінійного перемішування (перестановка) і один або два шари підмішування ключа. Прикладом здійснення такого підходу до побудови шифру є сучасний стандарт блокового симетричного шифрування ***AES*** (Advanced Encryption Standard).

До недоліків цього підходу можна зарахувати те, що для процедур шифрування й розшифрування в загальному випадку не можна використовувати одні й ті самі блоки, що збільшує апаратні й/або програмні затрати на здійснення. Проте через прогрес інтегральних технологій цей недолік втрачає свою вагу, через що виникає тенденція до розширення області застосування таких шифрів.

3.6.2. Стандарт блокового симетричного шифрування DES

Стандарт шифрування даних (Data Encryption Standard – DES), за допомогою якого можливо здійснити блокове симетричне шифрування, став без перебільшення цілою епохою в сучасній історії розвитку криптографії. Історія його розроблення, прийняття, багаторазових перезатверджень, спроб його криптоаналізу стала зразком здійснення перехідного етапу від розвитку криптографії в закритому режимі до новітнього розвитку криптографії у відкритому (публічному) режимі [1].

Отже, усе почалося 15 травня 1973 року, коли ***Національне бюро стандартів США*** (United States. National Bureau of Standards – NBS) опублікувало вимоги до криптографічного алгоритму, який можна було б прийняти як стандарт. Такий алгоритм можна було б сертифікувати, різні криптографічні засоби на його основі могли б взаємодіяти один з одним. Крім того, алгоритм мав бути відносно недорогим та доступним. Отже, опубліковані вимоги були такими (значною мірою вони актуальні до цього часу):

– алгоритм повинен забезпечувати високий рівень безпеки;

– алгоритм має бути детально описаний і зрозумілий;

– безпека алгоритму спирається на ключ, а не на таємність алгоритму;

– алгоритм має бути доступний усім користувачам;
– алгоритм повинен бути адаптований до різних застосувань;
– економічна вигідність апаратного здійснення;
– алгоритм має бути ефективним у використанні (велика швидкодія);
– алгоритм повинен надавати можливість перевіряння;
– алгоритм має бути дозволений для експорту.

Через низький рівень цивільної криптографічної науки початку 70-х років XX століття не відразу вдалося знайти алгоритм, який повною мірою відповідав цим вимогам. Проте нарешті обрали алгоритм, який згодом ліг в основу стандарту симетричного блокового шифрування DES. Більше того, стандарт DES протримався багато десятиліть аж до початку третього тисячоліття, незважаючи на постійну критику, нові спроби розкриття, нові конкурси на новий стандарт блокового симетричного шифрування. У стандарті DES здійснено більшість сучасних підходів до проектування блокового симетричного шифру, а критика DES стала ґрунтом для розроблення решти таких підходів. Отже, розглянемо основні принципи побудови DES.

DES є симетричним блоковим шифром, який шифрує дані 64-бітними блоками. На вході алгоритму є 64-бітний блок відкритого тексту, а на виході – 64-бітний блок шифртексту. Ключем також є 64-бітне число, проте кожен восьмий біт у ньому використовують для контролю парності. Тому фактично довжина ключа становить 56 бітів. Рівень стійкості DES повністю залежить від ключа. Деякі ключі вважають слабкими, проте цей недолік можна легко виявити, і не використовувати такі ключі.

На спрощеному рівні алгоритм є комбінацією двох основних (стародавніх) методів шифрування: перемішування й розсіювання (або підстановки й перестановки). Фундаментальним будівельним блоком DES є комбінація цих методів (особлива комбінація алгоритмів підстановки й перестановки). Такий блок називають етапом, або ***раундом***. DES складається з 16 раундів. Тобто одну й ту саму комбінацію методів застосовують до відкритого тексту та проміжних шифртекстів 16 разів.

Загальну схему DES зображають такою структурою (рис. 3.13).

Спочатку над 64-бітним блоком відкритого тексту здійснюють ***початкову перестановку*** (initial permutation – IP). Потім блок із переставленими бітами розбивають на ліву L_0 й праву R_0 половини завдовжки по 32 біти. Далі здійснюють перший раунд зашифрування. При цьому перетворюють тільки ліву частину блока L_0 Праву половину R_0 не змінюють, а переміщують на місце лівої в наступному раунді ($L_1 = R_0$). Ліву половину перетворюють за допомогою підключа K_1, правої частини R_0 і деякої функції f. Після перетворення її

переміщують на місце правої частини в наступному раунді ($R_1 = L_0 \oplus f(R_0, K_1)$). Однакові раунди зашифрування повторюють 16 разів, для чого застосовують 16 підключів раундів. В останньому раунді підблоки місцями не міняють. Після шістнадцятого раунду праву й ліву половини об'єднують, і алгоритм завершують кінцевою перестановкою (final permutation – FP або IP^{-1}), оберненою до початкової.

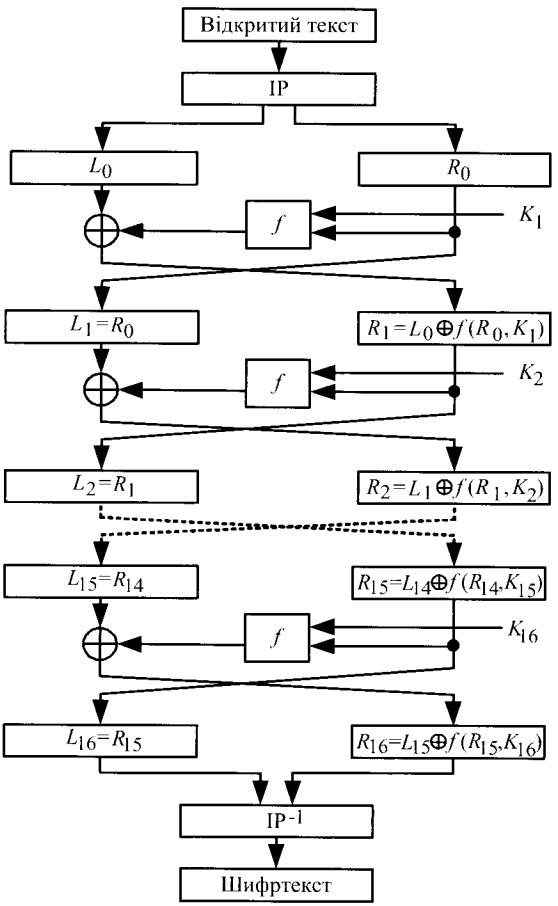


Рис. 3.13. Загальна структура DES

Алгоритм використовує лише стандартну арифметику 64-бітних чисел і логічні операції, тому його легко й швидко виконували в апаратурі другої половини 70-х років XX століття. Перші програми для здійснення алгоритму були доволі незграбні (через велику кількість операцій над окремими бітами), але сучасні програми значно досконаліші.

Початкова й кінцева перестановки. Початкова перестановка (табл. 3.2) і відповідна кінцева перестановка (табл. 3.3) не впливають на стійкість DES (ця перестановка сприяла полегшенню побайтного завантаження відкритого тексту й шифртексту в мікросхему DES). Проте ці перестановки стандартизовані й підлягають виконанню навіть сьогодні, хоча через прогрес інтегральних технологій прямої необхідності їх виконання (крім зазначеної) немає.

Таблиця 3.2

Таблиця початкової перестановки блока відкритого тексту

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

У цій та інших таблицях перестановок показано, яку фактичну позицію займе біт вхідного блока із заданим номером. Наприклад, у таблиці початкової перестановки бачимо, що 58-й біт вхідного блока стане першим бітом вихідного блока, 50-й біт – другим тощо.

Кінцева перестановка обернена до початкової.

Таблиця 3.3

Таблиця кінцевої перестановки блока відкритого тексту

40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

Схема раунду DES. Схему одного раунду DES подано на рис. 3.14.

У кожному раунді біти ключа зсувають, після чого з 56 бітів ключа вибирають 48 (стискаюча перестановка). Праву половину даних збільшують до 48 бітів внаслідок перестановки з розширенням. Потім її об'єднують операцією XOR з 48 бітами зміщеного та переставленого ключа, передають через 8 S-блоків підстановки, утворюючи 32 нові бітів, і переставляють знову. Ці чотири операції і є раундовою функцією f. Результат виконання функції f об'єднують із лівою половиною за допомогою XOR. У результаті формують нову праву половину, а стара права половина стає новою лівою. Усе повторюють 16 разів, утворюючи 16 раундів DES. Розглянемо детальніше зміст цих операцій.

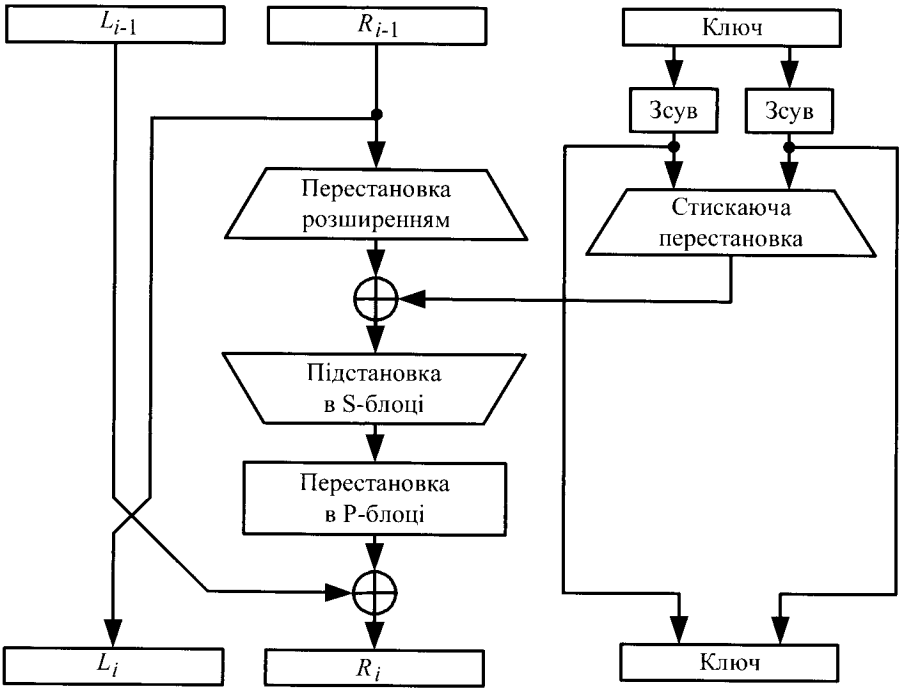


Рис. 3.14. Схема раунду DES

Перетворення ключа. Спочатку 64-бітовий ключ DES зменшують до 56-бітового відкиданням кожного восьмого біта. Ці біти використовують для контролю за парністю кількості одиниць, виявляючи ймовірне спотворення ключа під час передавання. Потім виконують початкову перестановку ключа (табл. 3.4), що обумовлена способом побайтного введення даних до мікросхеми DES.

Таблиця 3.4

Таблиця початкової перестановки ключа

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

Далі з переставленого 56-бітного ключа формують 16 48-бітних раундових підключів. Ці підключі K_i формують так. Спочатку 56-бітний ключ ділять на дві 28-бітні половини. Потім ці половини циклічно зсувають уліво на один або два біти залежно від номера раунду (табл. 3.5).

Таблиця 3.5

Величини зсуву половинок ключа

Раунд	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Зсув	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Після зсуву половинки об'єднують і з 56 бітів вибирають 48 бітів. Оскільки при цьому не лише вибирають підмножину бітів, але й змінюють їх послідовність, цю операцію називають ***стискною перестановкою*** або ***перестановочною вибіркою*** (табл. 3.6). 48-бітний результат стискної перестановки і є раундовим підключем. Завдяки зсуву для формування кожного підключа використовують іншу підмножину бітів первинного ключа. Кожен біт первинного ключа використовують приблизно в 14 з 16 підключів.

Таблиця 3.6

Таблиця стискної перестановки

14	17	11	24	1	5	3	28	15	6	21	10
23	19	11	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

Перестановка з розширенням. Ця операція розширює праву половину даних R_i з 32 до 48 бітів. Таку операцію називають ***перестановкою з розширенням.*** За її допомогою вирішують два завдання: приводять розмір правої половини у відповідність із розміром ключа для виконання операції XOR, а також викликають ***лавинний ефект***. Суть цього явища в тому, що мала зміна відкритого тексту або ключа призводять до значних змін шифртексту. За допомогою розширюючої перестановки біти переставляють так, що деякі з них впливають потім не на одну, а на дві підстановки в S-блоках, що призводить до зміни відразу 8 бітів шифртексту.

Принцип здійснення перестановки з розширенням (інколи її називають ***E-блоком*** (від expansion)) показано на рис. 3.15 (показано перестановку половини з 32 вхідних бітів).

Таким чином, з кожного 4-бітного вхідного підблока перший і четвертий біти формують два біти вихідного блока, а другий і третій біти – по одному біту вихідного блока. Повну перестановку з розширенням, як правило, задають таблицею розширюючої перестановки (табл. 3.7).

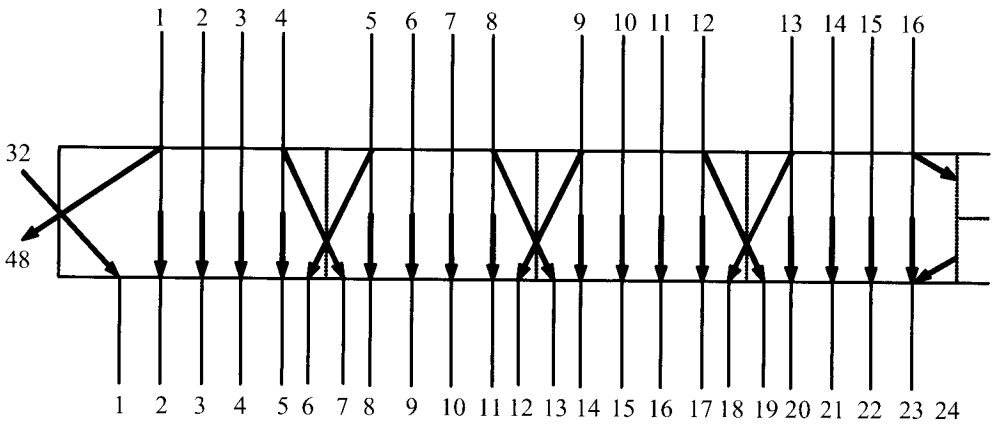


Рис. 3.15. Перестановка з розширенням

Таблиця 3.7

Таблиця розширюючої перестановки

32	1	2	3	4	5	4	5	6	7	8	9
8	9	10	11	12	13	12	13	14	15	16	17
16	17	18	19	20	21	20	21	22	23	24	25
24	25	26	27	28	29	28	29	30	31	32	1

Підстановка за допомогою S-блоків. Після об'єднання стисненого підключа з розширеним блоком за допомогою операції XOR над 48-бітним результатом виконують операцію підстановки. Підстановку (тобто заміну) здійснюють за допомогою восьми блоків підстановки, або S-блоків (від substitution). У кожного S-блока є 6-бітний вхід і 4-бітний вихід. Усі вісім блоків різні. Під час виконання підстановки 48 вхідних бітів ділять на вісім 6-бітних підблоків. Кожен підблок обробляють своїм S-блоком: замість 6-бітного підблока підставляють 4-бітний (рис. 3.16).

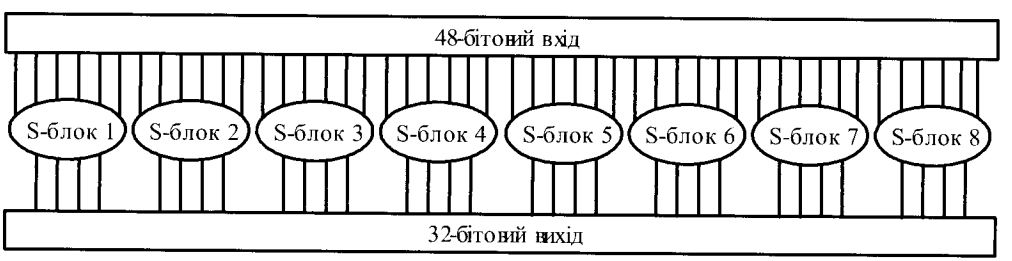


Рис. 3.16. Схема включення S-блоків

S-блоки DES подано в табл. 3.8.

Таблиця 3.8

S-блоки DES

		Номер стовпця																
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Номер рядка	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	S_1
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	S_2
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	S_3
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	S_4
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	S_5
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	S_6
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	
	0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	S_7
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
	0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	S_8
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

Кожен із восьми S-блоків (табл. 3.8) є таблицею з 4 рядків і 16 стовпців. Нумерацію рядків і стовпців починають із нуля. У таблицях певним чином розташовано 4-бітні числа. Шість вхідних бітів S-блока визначають номери рядка й стовпця, на перетині яких зчитують 4-бітне число, яке підставляють замість вхідного 6-бітного.

Нехай маємо 6-бітний вхід S-блока: *b1*, *b2*, *b3*, *b4*, *b5* і *b6*. Біти *b1* і *b6* об'єднують, утворюючи 2-бітне число від 0 до 3, яке визначає номер рядка таблиці. Середні 4 біти, від *b2* до *b5*, – об'єднують, утворюючи 4-бітне число від 0 до 15, яке визначає номер стовпця таблиці. Наприклад, нехай на вхід четвертого S-блока (тобто біти функції XOR від 19 до 24) подають 101011. Перший і останній біти, об'єднавшись, утворюють двійкове число 11, що відповідає третьому рядку четвертого S-блока. Середні 4 біти утворять 0101, що відповідає п'ятому стовпчику того самого S-блока. У четвертому S-блоці (див. табл. 3.8) на перетині рядка з номером 3 і стовпця з номером 5 зчитуємо число 1. Двійкове 4-бітне число 0001, яке відповідає одиниці, і підставляємо замість вхідного 6-бітного блока 101011.

Підстановка за допомогою S-блоків є ключовим кроком алгоритму DES. Саме ця підстановка визначає стійкість алгоритму. Інші операції алгоритму лінійні й легко піддаються аналізу. S-блоки нелінійні, і саме вони визначають безпеку DES.

Результатом підстановки є вісім 4-бітних блоків, які знов об'єднують у єдиний 32-бітний блок. Цей блок находить на вхід наступного кроку алгоритму – перестановки за допомогою P-блока.

Перестановка за допомогою P-блока. Тут здійснюють пряму перестановку 32-х вхідних бітів. Жоден біт не використовують двічі й жоден не відкидають (табл. 3.9).

Таблиця 3.9

Таблиця перестановки в P-блоці

16	7	20	21	29	12	28	17	1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

Результат перестановки за допомогою P-блока підсумовують побітово за модулем 2 із лівою половиною вхідного 64-бітного блока раунду. Потім ліву й праву половини міняють місцями та починають наступний раунд.

Розшифрування в DES. *Після всіх підстановок, перестановок, операцій XOR і циклічних зсувів можна подумати, що алгоритм розшифрування різко відрізняється від алгоритму шифрування і є таким самим заплутаним. Проте, різні компоненти DES завдяки мережі Фейстеля, за якою його побудовано,* підібрано так, що для зашифрування й розшифрування використовують один і той самий алгоритм. Єдина відмінність у тому, що підключі раундів застосовують у зворотній послідовності. Тобто, якщо для зашифрування використовували ключі K_1, K_2, K_3 ..., K_{16}, то ключами розшифрування будуть K_{16}, K_{15}, K_{14} ..., K_1.

Потрійний DES. Упродовж останніх десятиліть DES інтенсивно досліджували. Зараз його вже не вважають стійким, в основному через недостатню довжину ключа. Існує декілька модифікацій, направлених на вдосконалення DES. Найвідоміші з них полягають у потрійному застосуванні алгоритму. Одну з можливих схем такого ***потрійного DES*** (Triple DES) показано на рис. 3.17.

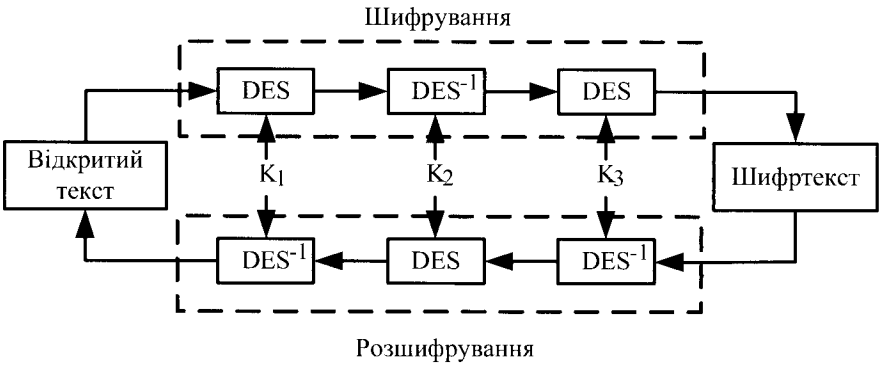
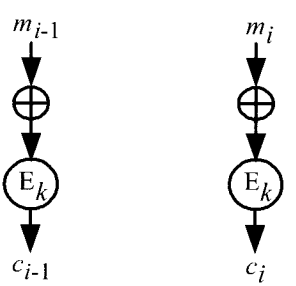


Рис. 3.17. Схема потрійного DES

Режими застосування блокових симетричних шифрів. Для шифрування відкритого тексту довільної довжини блокові шифри можна використати в декількох режимах. Проаналізуємо чотири основні режими застосування блокових шифрів, що найчастіше зустрічаються в системах криптографічного захисту інформації [1, 2], а саме режими ***електронної кодової книги*** (Electronic Code Book – ECB), ***зчеплення блоків шифртексту*** (Cipher Block Chaining – CBC), ***зворотного зв'язку за шифртекстом*** (Cipher Feedback – CFB) і ***зворотного зв'язку за виходом*** (Output Feedback – OFB).

Розгляд усіх попередніх шифрів, зокрема DES, стосувався ***режиму простої заміни,*** або інакше, режиму ***електронної кодової книги.*** У цьому режимі кожен i-й блок відкритого тексту m_i шифрують блоковим шифром з ключем k (E_k) незалежно від інших (рис. 3.18). Результатом зашифрування є i-й блок шифртексту c_i. Подібним чином здійснюють розшифрування за допомогою алгоритму розшифрування D_k.

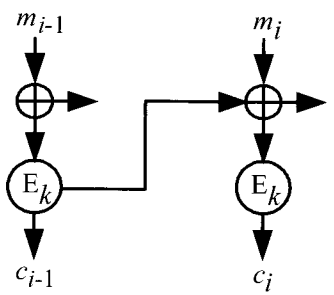


$$c_i = E_k(m_i);\ m_i = D_k(c_i).$$

Рис. 3.18. Режим електронної кодової книги

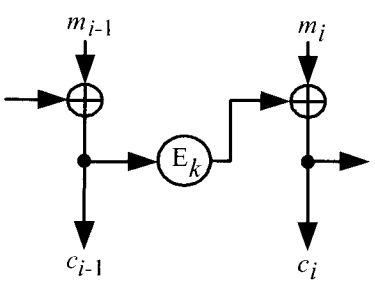
Цей режим має такі недоліки: 1) однакові блоки у відкритому тексті переходять в однакові блоки шифртексту; 2) поширення помилки. Ці недоліки призводять до того, що криптоаналітик, знаючи кілька блоків відкритого тексту (наприклад, типові звертання в листах) і перехопивши шифртекст, може спростити криптоаналіз. Крім того, суперник може модифікувати повідомлення. Наприклад, легально переслати на власний банківський рахунок невелику суму грошей і перехопити шифртекст, який відповідає зашифрованому номеру власного рахунку. Потім йому достатньо вставити цей шифртекст замість іншого зашифрованого номера рахунку в чужому переказі.

Подібних ризиків можна уникнути, здійснюючи зашифрування в інших режимах. Ці режими передбачено стандартом DES, але придатні також і для інших блокових шифрів.



$c_i = E_k(m_i \oplus c_{i-1}); m_i = D_k(c_i) \oplus c_{i-1}.$

Рис. 3.19. Режим зчеплення зашифрованих блоків



$c_i = E_k(c_{i-1}); \; m_i = E_k(c_{i-1}) \oplus m_i.$

Рис. 3.20. Режим зворотного зв'язку за шифртекстом

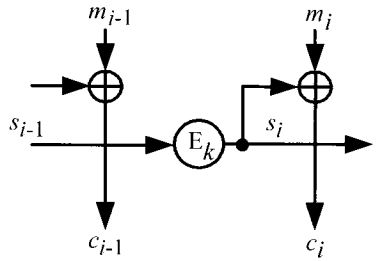
У ***режимі зчеплення зашифрованих блоків*** перед зашифруванням i-го блока відкритого тексту m_i до нього додають побітово за модулем 2 попередній $(i-1)$-й блок шифртексту c_{i-1} (рис. 3.19).

Для початку процесу шифрування в якості нульового блока шифртексту використовують ***вектор ініціалізації*** (або початковий вектор), який передають у канал зв'язку у відкритому вигляді.

У цьому режимі однаковим блокам відкритого тексту відповідають різні блоки шифртексту. Причому кожен блок шифртексту залежить від усіх попередніх, а тому підміну якогось блока буде відразу виявлено.

У режимі зворотного зв'язку за шифртекстом попередній блок шифртексту зашифровують ще раз, і для отримання чергового блока шифрованого тексту результат додають побітово за модулем 2 із блоком відкритого тексту. Для початку процесу зашифрування також використовують вектор ініціалізації (рис. 3.20).

У цьому режимі, якщо два блоки шифрованого тексту ідентичні, то результати їх шифрування на наступному кроці також будуть ідентичні, що створює можливість просочування інформації про початковий текст. Цей режим відповідає режиму гамування зі зворотним зв'язком алгоритму ГОСТ 28147-89.



$c_i = m_i \oplus s_i;\ m_i = c_i \oplus s_i;\ s_i = E_k(s_{i-1})$.

Рис. 3.21. Режим зворотного зв'язку за виходом

Режим зворотного зв'язку за виходом подібний до режиму зворотного зв'язку за шифртекстом за винятком того, що блоки s_i, які додають за модулем 2 із блоками відкритого тексту, генерують незалежно від відкритого або шифрованого тексту (рис. 3.21).

Для початку процесу зашифрування також використовують початковий вектор ініціалізації. Цей режим володіє перевагою перед режимом зворотного зв'язку за шифртекстом у тому сенсі, що будь-які бітові помилки, що виникли в процесі передавання, не впливають на розшифрування подальших блоків. Проте тут можлива маніпуляція початковим текстом шляхом зміни шифрованого тексту.

3.6.3. Шифр ГОСТ 28147-89

Конкурентом DES був алгоритм шифрування ***ГОСТ 28147-89***, який застосовували в СРСР, а також в Україні до прийняття нового стандартного алгоритму симетричного блокового шифрування "Калина". Розглянемо коротко його особливості.

Шифр не накладає обмежень на ступінь конфіденційності оброблюваної інформації. Його можна здійснити порівняно нескладними апаратними й програмними засобами.

Це блоковий алгоритм [1], який шифрує 64-бітні блоки за допомогою 256-бітного ключа. Шифр є 32-раундовою мережею Фейстеля із тридцятьма двома раундовими підключами. Схема одного раунду алгоритму ГОСТ 28147-89 подана на рис. 3.22.

Раундові підключі формують так. Початковий таємний ключ завдовжки 256 бітів ділять на вісім 32-бітні підключів K_0, $K_1, \ldots K_7$. Ці підключі в 32-х раундах шифрування використовують у такій послідовності: K_0, $K_1, \ldots K_7$; K_0, $K_1, \ldots K_7$; K_0, $K_1, \ldots K_7$; K_7, $K_6, \ldots K_0$.

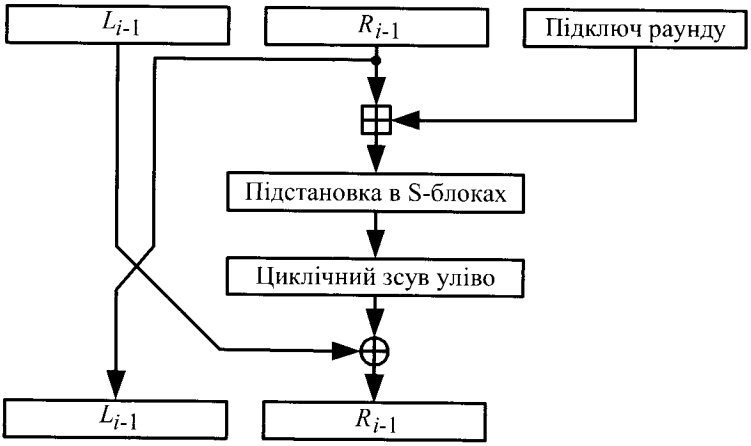


Рис. 3.22. Схема одного раунду ГОСТ 28147-89

На i-му раунді алгоритму, згідно зі схемою на рис. 3.22, виконують такі перетворення:

$$L_i = R_{i-1},$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K).$$

Раундова функція f проста. Спочатку праву половину вхідного блока раунду та підключ раунду додають за модулем 2^{32}. Результат розбивають на вісім 4-бітних частин, кожну з яких подають на вхід свого S-блока. У ГОСТі є вісім різних S-блоків. Перші 4 біти потрапляють на перший S-блок, наступні 4 біти – на другий S-блок тощо. Кожен S-блок є перестановкою чисел від 0 до 15. Наприклад, S-блок може виглядати так:

7, 10, 2, 4, 15, 9, 0, 3, 6, 12, 5, 13, 1, 8, 11.

У цьому випадку вхідне 4-бітове число цього S-блока визначає порядковий номер числа (починаючи з нуля), яке буде вибрано під час заміни. Тобто, якщо на вході S-блока 0, то на виході 7, якщо на вході 1, на виході 10 тощо. Усі вісім S-блоків є різними. Фактично вони є частиною ключа, тому S-блоки потрібно зберігати в таємниці.

Виходи всіх восьми S-блоків об'єднують у 32-бітне слово, яке далі циклічно зсувають уліво на 11 бітів. Нарешті, результат об'єднують операцією XOR із лівою половиною. Так отримують нову праву половину, а вхідна права половина стає новою лівою половиною. Усе це повторюють 32 рази й отримують результат зашифрування одного блока.

Алгоритм передбачає криптографічне перетворення даних у таких режимах роботи:

– простої заміни, який збігається з режимом електронної кодової книги (Electronic Code Book – ECB);

– гамування, який збігається з режимом зворотного зв'язку за шифртекстом (Cipher Feedback – CFB);

– гамування зі зворотним зв'язком, який збігається з режимом зчеплення блоків шифртексту (Cipher Block Chaining – CBC);

– вироблення імітовставки.

Особливості перших трьох режимів ми вже розглянули в попередньому пункті. ***Режим вироблення імітовставки*** призначений для виявлення випадкових і навмисних помилок (спотворень) під час передавання зашифрованих даних і є однаковим для будь-якого з режимів шифрування відкритої інформації, які було розглянуто вище. Імітовставкою є додатковий l-бітний блок інформації U, який формують або перед шифруванням усього повідомлення, або одночасно із шифруванням окремих блоків. Кількість бітів l визначають, ураховуючи імовірність виникнення помилкової інформації $p = 2^{-l}$. Процес формування імітовставки відбувається так. Перший блок відкритої інформації шифрують 16-ма раундами, в режимі простої заміни. Результат шифрування додають за модулем 2 із другим блоком відкритої інформації. Потім результат додавання знову шифрують 16-ма раундами, додають за модулем 2 із третім блоком відкритої інформації й так далі. Із отриманого після останнього кроку криптоперетворення блока шифртексту вибирають імітовставку завдовжки l бітів (зазвичай 32 біти).

Імітовставку передають наприкінці зашифрованих даних або після кожного зашифрованого блока інформації. На приймальному боці після розшифрування шифртексту з отриманих блоків інформації формують імітовставку, яку порівнюють з імітовставкою, отриманою з каналу зв'язку. Якщо вони різні, розшифровану інформацію вважають помилковою.

3.6.4. Стандарт блокового симетричного шифрування AES

У 1997 р. ***Національний інститут стандартів і технологій*** США (National Institute of Standards and Technology – NIST) оголосив про початок програми з ухвалення ***нового стандарту шифрування*** (Advanced Encryption Standard – AES) – стандарту XXI ст. для закриття важливої інформації урядового рівня на заміну алгоритму DES, що існував ще з 1974 р.

Вимоги до кандидатів були такі:

– криптоалгоритм має бути відкрито опублікований;

– криптоалгоритм має бути симетричним блоковим шифром, що допускає розміри ключів 128, 192 і 256 бітів;

– криптоалгоритм має бути зручним як для апаратного, так і для програмного здійснення;

– криптоалгоритм має бути доступний для відкритого застосування в будь-яких продуктах, а отже, не може бути запатентований, інакше патентні права мають бути анульовані;

– криптоалгоритм аналізують за такими параметрами: стійкості, вартості, гнучкості, можливості здійснення в smart-картах.

Стійкість – це найважливіший критерій при оцінюванні алгоритму. Оцінювали: здатність шифру протистояти різним методам криптоаналізу, статистична безпека й відносна захищеність порівняно з іншими кандидатами.

Вартість – це не менш важливий критерій, ураховуючи одну з основних цілей NIST – широка область використання й доступність AES. Вартість залежить від обчислювальної ефективності (насамперед швидкодії) на різних платформах, зручності програмного й апаратного здійснення, низьких вимог до пам'яті, простоти (прості алгоритми легко здійснювати, вони прозоріші для аналізу).

Гнучкість – передбачає здатність алгоритму обробляти ключі більше обумовленого розміру (128 бітів), надійність і ефективність виконання в різному середовищі, можливість здійснення інших криптографічних функцій: комбінованого шифрування, хешування тощо.

Інакше кажучи, AES мав бути істотно ефективнішим із погляду практичного здійснення (насамперед швидкості шифрування й формування ключів), мати більший запас міцності, ніж потрійний DES, при цьому не поступаючись йому в стійкості.

Можливість здійснення в smart-картах. Передбачалося, що важлива область використання AES в майбутньому – smart-карти, коли головною проблемою є невеликий обсяг доступної пам'яті. NIST виходив із припущення, що деякі дешеві карти можуть мати лише 256 байтів оперативної пам'яті (Random Access Memory – RAM) для обчислюваних даних і 2000 байт постійної пам'яті (Read Only Memory – ROM) для зберігання алгоритмів і констант. Існує два основні методи формування раундових ключів і (або) базової операції підстановки:

– обчислення перед початком шифрування та зберігання в пам'яті;

– обчислення раундових ключів і нелінійної функції підстановки "на льоту".

Зрозуміло, що другий варіант зменшує витрати RAM, і тому наявність такої можливості в криптоалгоритмі є його безперечною перевагою.

На відкритий конкурс було прийнято 15 алгоритмів, які розробили криптографи з 12 країн: Австралії, Бельгії, Великобританії, Німеччини, Ізраїлю, Канади, Коста-Ріки, Норвегії, США, Франції, Південної Кореї та Японії. У фінал конкурсу вийшли такі алгоритми: ***MARS***, ***TWOFISH*** і ***RC6*** (США), ***RIJNDAEL*** (Бельгія), ***SERPENT*** (Великобританія, Ізраїль, Норвегія). За своєю структурою TWOFISH є класичним шифром Фейстеля; MARS і RC6 можна зарахувати до модифікованих шифрів Фейстеля, у них використовують нову маловивчену операцію циклічного "прокручування" бітів слова на кількість позицій, що змінюють залежно від шифрованих даних і секретного ключа; RIJNDAEL і SERPENT є класичними SP-мережами. MARS і TWOFISH мають найскладнішу конструкцію, RIJNDAEL і RC6 – найпростішу.

У процесі конкурсу шифр RIJNDAEL більшість учасників ***визнали кращим***, якщо не буде обрано їхнього власного шифру. В алгоритмі не було виявлено слабких місць у захисті.

Переваги RIJNDAEL:

– висока ефективність на будь-яких платформах;

– високий рівень захищеності;

– зручний для здійснення в smart-картах завдяки низьким вимогам до пам'яті;

– швидка процедура формування ключа;

– добра підтримка паралелізму на рівні інструкцій;

– підтримка різних довжин ключа із кроком 32 біти.

Недоліки RIJNDAEL:

– уразливий до аналізу потужності.

У жовтні 2000 р. конкурс завершився. Переможцем визнано бельгійський шифр RIJNDAEL як такий, що має якнайкраще поєднання стійкості, продуктивності, ефективності здійснення та гнучкості. Його низькі вимоги до обсягу пам'яті роблять його ідеально зручним для вбудованих систем. Авторами шифру є ***Йоан Даймен*** (Joan Daemen) та ***Вінсент Реймен*** (Vincent Rijmen), початкові букви прізвищ яких і утворюють назву алгоритму – RIJNDAEL.

Після цього NIST розпочав підготовку попередньої версії Федерального стандарту обробки інформації (Federal Information Processing Standart – FIPS) і в лютому 2001 р. опублікував його на сайті http://csrc.nist.gov/encryption/aes/. Протягом 90-денного періоду відкритого обговорення попередню версію FIPS переглядали, враховуючи коментарі, після чого розпочався процес виправлень і затвердження. Нарешті, 26 листопада 2001 р. було опубліковано остаточну версію стандарту FIPS-197, що описує новий американський стандарт шифрування AES. Стандарт набув чинності від 26 травня 2002 р.

RIJNDAEL – криптоалгоритм нефейстелівського типу. Це типова SP-мережа, процедура розшифрування в якій відрізняється від процедури зашифрування не тільки порядком раундових ключів. Алгоритм орієнтований на роботу з байтами. Жодної операції, окрім XOR, у якій би дії виконували над окремими бітами, у ньому немає. Це забезпечує зручність при здійсненні та збільшує швидкодію.

Шифр має змінні параметри:

– довжина блока ВТ (відкритого тексту) та ШТ (шифртексту) може набувати значень 128, 192 та 256 бітів (у FIPS-197 довжину блока ВТ визначено тільки у 128 бітів);

– довжина ключа також може набувати значень 128, 192 та 256 бітів;

– кількість раундів набуває значень від 10 до 14 залежно від перших двох параметрів (табл. 3.10).

Таблиця 3.10

Визначення кількості раундів у шифрі RIJNDAEL

		Nk		
		4	6	8
Nb	4	10	12	14
	6	12	12	14
	8	14	14	14

Блок ВТ, а також ключ зручно подавати записаними в таблиці із чотирма рядками та N_b і N_k стовпцями відповідно:

Блок ВТ

a_{00}	a_{01}	...
a_{10}	a_{11}	...
a_{20}	a_{21}	...
a_{30}	a_{31}	...

$N_b \in \{4,6,8\}$

Ключ

k_{00}	k_{01}	...
k_{10}	k_{11}	...
k_{20}	k_{21}	...
k_{30}	k_{31}	...

$N_k \in \{4,6,8\}$

У кожній клітинці такої таблиці записано один байт. Таблиці заповнюють згори донизу за стовпцями. Відповідно, один стовпець таблиці є 4-байтовим словом (W). N_b та N_k – кількість слів у відповідній таблиці. Наприклад, якщо довжина блока ВТ дорівнює 128, то $N_b = 4$, якщо довжина блока ВТ дорівнює 192, то $N_b = 6$ тощо. Параметри N_b та N_k обирає користувач незалежно один від одного.

Проміжні результати перетворень, що виконують у криптоалгоритмі, називають станами (State). Стан на кожному кроці алгоритму також подають у вигляді таблиці з N_b слів. Кількість раундів N_r визначають за вибраними довжинами блока ВТ і ключа згідно з табл. 3.10.

Перетворення в одному раунді. Раунд складається із чотирьох різних перетворень:

1. SubBytes (підстановка байтів).
2. ShiftRows (зсув рядків).
3. MixColumns (перемішування стовпців).
4. AddKey (додавання ключа).

Останній N_r-й раунд містить тільки 1, 2 і 4 операції.

Розглянемо кожне із цих перетворень окремо.

Через a або b будемо позначати байти (в окремих випадках – біти), що подають на вхід операції, а через a', b' – байти на виході операції.

1. Операція SubBytes

Перетворення виконують над кожним байтом стану окремо. Воно складається із двох етапів:

а) кожен байт можна представити як елемент поля $GF(2^8)$, тобто у вигляді многочлена:

$$a(x) = a_7 x^7 + \ldots + a_1 x + a_0$$

або у вигляді вектора $a = (a_7, \ldots, a_0)$. Як породжувальний многочлен поля $GF(2^8)$ взято незвідний многочлен $f(x) = x^8 + x^4 + x^3 + x + 1$ над полем $GF(2)$. Тобто дії над байтами виконують як дії над многочленами за $\bmod f(x)$.

Нехай a' – байт, отриманий у результаті цієї операції. Тоді $a' = a^{-1}$ – обернений до a елемент у полі $GF(2^8)$.

Це єдине нелінійне перетворення в алгоритмі.

б) афінне перетворення:

Нехай $(a_0, a_1, \ldots, a_7)$ – байт на вході операції, тоді:

$$\begin{pmatrix} a'_0 \\ a'_1 \\ \vdots \\ a'_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_7 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

Як ми вже зазначали, нелінійні перетворення забезпечують перемішування, а лінійні – розсіювання.

2. Операція ShiftRows

У таблиці стану (відповідно в клітинках таблиці записано байти стану) виконують циклічний зсув байтів у рядках на C_0, C_1, C_2, C_3 позицій уліво відповідно (рис. 3.23). C_0 завжди дорівнює 0. Величини зсувів C_1, C_2, C_3 визначають за вибраною довжиною блока відкритого тексту відповідно до табл. 3.11.

Таблиця 3.11

Визначення величини зсуву рядків у шифрі RIJNDAEL

	Nb		
	4	6	8
C_1	1	2	2
C_2	1	2	3
C_3	1	3	4

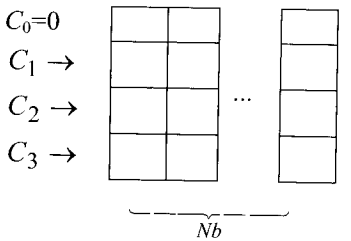


Рис. 3.23. Схема зсуву рядків стану

3. Операція MixColumns

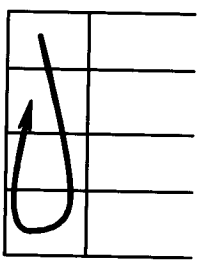


Рис. 3.24. Схема перемішування стовпців в алгоритмі RIJNDAEL

Ця операція діє на кожний стовпець стану окремо. Вона полягає у такому перемішуванні байтів стовпця (рис. 3.24).

Кожен стовпець можна подати як поліном степеня не вищого за 3 над $GF(2^8)$. Наприклад, перший стовпець:

$$a(x) = a_{00} + a_{10}x + a_{20}x^2 + a_{30}x^3, \quad a_{ij} \in GF(2^8).$$

Під час перемішування цей поліном множать на фіксований многочлен

$$g(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$$

за модулем $x^4 + 1$. (Тут $\{03\}$, $\{01\}$, $\{02\}$ – байти, представлені як числа 16-ої системи, тобто $\{03\} = (0000\ 0011)$ тощо). Отже, результатом перемішування стовпця є стовпець коефіцієнтів многочлена $a'(x)$, де:

$$a'(x) = a(x)g(x) \bmod (x^4 + 1).$$

Коефіцієнти многочленів $a(x)$ і $g(x)$ перемножують як елементи поля $GF\left(2^8\right)$, тобто за $\bmod f(x)$. Очевидно, степінь многочлена $a'(x)$ також не перевищує 3. Многочлен x^4+1 узятий за модуль завдяки такій його властивості:

$$x^i \bmod(x^4+1) = x^{i \bmod 4}.$$

Це полегшує зведення результату множення многочленів за модулем. Дійсно, якщо перемножити два многочлени третього степеня $a(x) = a_3x^3 + a_2x^2 + a_1x + a_0$ і $b(x) = b_3x^3 + b_2x^2 + b_1x + b_0$, то в результаті отримаємо многочлен:

$$a(x)b(x) = c(x) = c_6x^6 + c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0,$$

де

$$\begin{aligned}
c_0 &= a_0b_0,\\
c_1 &= a_1b_0 \oplus a_0b_1,\\
c_2 &= a_2b_0 \oplus a_1b_1 \oplus a_0b_2,\\
c_3 &= a_3b_0 \oplus a_2b_1 \oplus a_1b_2 \oplus a_0b_3,\\
c_4 &= a_3b_1 \oplus a_2b_2 \oplus a_1b_3,\\
c_5 &= a_3b_2 \oplus a_2b_3,\\
c_6 &= a_3b_3.
\end{aligned}$$

Після зведення $c(x)$ за $\bmod(x^4+1)$ отримаємо многочлен:

$$d(x) = d_3x^3 + d_2x^2 + d_1x + d_0,$$

де

$$\begin{aligned}
d_0 &= a_0b_0 \oplus a_3b_1 \oplus a_2b_2 \oplus a_1b_3,\\
d_1 &= a_1b_0 \oplus a_0b_1 \oplus a_3b_2 \oplus a_2b_3,\\
d_2 &= a_2b_0 \oplus a_1b_1 \oplus a_0b_2 \oplus a_3b_3,\\
d_3 &= a_3b_0 \oplus a_2b_1 \oplus a_1b_2 \oplus a_0b_3,
\end{aligned}$$

що в матричній формі записують так:

$$\begin{pmatrix} d_0\\ d_1\\ d_2\\ d_3 \end{pmatrix} = \begin{bmatrix} a_0 & a_3 & a_2 & a_1\\ a_1 & a_0 & a_3 & a_2\\ a_2 & a_1 & a_0 & a_3\\ a_3 & a_2 & a_1 & a_0 \end{bmatrix} \begin{pmatrix} b_0\\ b_1\\ b_2\\ b_3 \end{pmatrix}.$$

4. Операція AddKey

Операція полягає в побітовому додаванні раундового ключа й блока стану за $\bmod 2$.

Алгоритм формування ключів (Key Shedule). Алгоритм формування ключів складається із двох частин: розширення ключа (***Key Expansion***) та вибору раундового ключа (***Round Key Selection***). Розширений ключ можна представити як таблицю з одним рядком з $N_b(N_r+1)$ 4-байтових слів (рис. 3.25).

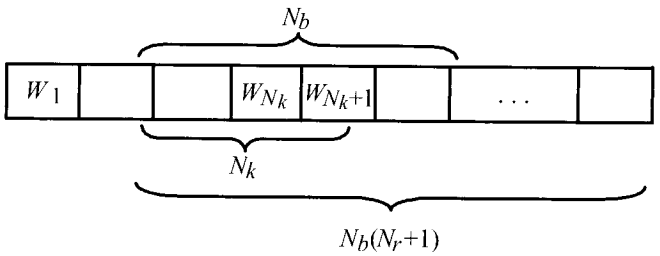


Рис. 3.25. Схема формування раундових ключів в алгоритмі RIJNDAEL

Перші N_k слів заповнюють словами ключа шифрування. Наступні слова отримують із попередніх за допомогою рекурентної процедури:

$$W_i = W_{i-1} \oplus W_{i-N_k},$$

якщо i не кратне N_k. При i-кратному N_k процедура обчислення слова дещо складніша, а саме:

$$W_i = SubWord(RotW_{i-1}) \oplus C(i/N_k) \oplus W_{i-N_k}.$$

Тут операція $RotW$ здійснює побайтовий циклічний зсув слова W на один байт уліво. Операція $SubWord$ здійснює побайтову заміну з використанням функції ***SubBytes***. Константа $C(t) = 2^{t-1}$.

Так отримують смугу завдовжки $N_b(N_r+1)$ слів. Цю смугу поділяють на блоки завдовжки N_b слів, перший з яких використовують у початковій операції ***AddKey***, а решту – як раундові ключі. З метою економії пам'яті можна не формувати всю смугу одразу, а робити це поступово, на кожному раунді шифрування, запам'ятовуючи лише N_k останніх слів (формування ключів "на льоту").

Функція зашифрування. Зашифрування в ***RIJNDAEL*** складається з:

– початкового додавання до блока BT першого раундового ключа (***AddKey***);

– $N_r - 1$ раунду;

– завершального раунду, у якому відсутня операція ***MixColumns***.

Функція розшифрування. При розшифруванні виконують операції, обернені до операцій ***SubBytes***, ***ShiftRows***, ***MixColumns*** та ***AddKey*** у зворотній послідовності. При цьому послідовність використання раундових ключів також змінюють на протилежну.

Сформулюємо основні особливості RIJNDAEL:

– нова архітектура "Квадрат", яка забезпечує швидке розсіювання й перемішування інформації, коли за один раунд перетворюють весь вхідний блок;

– байт-орієнтована структура, зручна для здійснення на 8-розрядних мікроконтролерах;

– усі раундові перетворення є операціями в скінчених полях, відповідно допускають ефективне апаратне й програмне здійснення на різних платформах.

3.6.5. Національний стандарт блокового симетричного шифрування ДСТУ 7624:2014

У результаті успішного проведення Державною службою спеціального зв'язку та захисту інформації України національного відкритого конкурсу симетричних блокових криптографічних алгоритмів [6], у 2015 році в Україні було затверджено новий стандарт симетричного блокового шифрування ДСТУ 7624:2014 [7], основою якого є алгоритм симетричного блокового шифрування "Калина" [8].

Алгоритм виконує шифрування блоків даних із використанням ключів. Розміри блока даних та ключів можуть бути такими [8]:

– блок розміром 128 бітів (ключ завдовжки 128 і 256 бітів);

– блок розміром 256 бітів (ключ завдовжки 256 і 512 бітів);

– блок розміром 512 бітів (ключ завдовжки 512 бітів).

Представлення вхідних та вихідних даних. Вхідними даними алгоритму "Калина" є відкритий текст та ключ, які подають у вигляді рядків байтів завдовжки $8 \times N_b$ та $8 \times N_k$ байтів відповідно. Вихідними даними є шифртекст, який також подають у вигляді рядка завдовжки $8 \times N_b$ байтів. Один байтовий рядок завдовжки $n = 8 \times N_b$ байтів подають у такій формі: $B_0 B_1 B_2 \ldots B_{n-1}$.

Поточний стан шифру. Перед зашифруванням або розшифруванням вхідні дані подають у вигляді двовимірного масиву байтів, який називають поточним станом шифру.

Поточний стан даних, які шифрують, подають як матрицю розміру $8 \times N_b$ байтів (тобто вісім рядків завдовжки N_b байтів – вісім рядків, N_b стовпців; коефіцієнтами матриці є байти). Кожен байт у поточному стані має два індекси: номер рядка $(r, 0 \le r < 8)$ і номер стовпця $(c, 0 \le c < N_b)$. Відповідно, байт вхідних та проміжних даних адресують як $s_{r,c}$ або $s[r,c]$.

Крім того, у деяких операціях шифру поточний стан шифру $S = (s_{i,j})$ розглядають як послідовність 64-бітових слів $S = (S_i)$ – тобто послідовність стовпців. У цьому випадку кожен стовпець із номером i, $0 \le i < N_b$ розглядають як окреме 64-бітове слово, що має значення $S_i = s_{0,i} \cdot 2^{0 \times 8} + s_{1,i} \cdot 2^{1 \times 8} + ... + s_{7,i} \cdot 2^{7 \times 8}$. Відповідно, молодшим бітом 64-бітового блока є молодший біт байта рядка з найменшим номером, старшим бітом 64-бітового блока – старший біт байта з рядка з найбільшим номером.

Приклад переформатування в поточний стан шифру для довжини блока 128 бітів. При зашифруванні відкритий текст подають байтовою послідовністю $in_0, in_1, ... in_{15}$ – рядком з 16-ти байтів. Відповідно, обчислений шифртекст формують як послідовність 16-ти байтів $out_0, out_1, ... out_{15}$. Заповнення поточного стану шифру перед початком зашифрування та після його закінчення подано в табл. 3.12.

Аналогічно, під час розшифрування шифртекст подають байтовою послідовністю $in_0, in_1, ... in_{15}$; обчислений відкритий текст формують у послідовність байтів $out_0, out_1, ... out_{15}$. При цьому заповнення поточного стану також відповідає табл. 3.12.

Таблиця 3.12

Таблиці заповнення поточного стану для довжини блоку 128 бітів

Вхідна послідовність	
in_0	in_8
in_1	in_9
in_2	in_{10}
in_3	in_{11}
in_4	in_{12}
in_5	in_{13}
in_6	in_{14}
in_7	in_{15}

→

Поточний стан	
$s_{0,0}$	$s_{0,1}$
$s_{1,0}$	$s_{1,1}$
$s_{2,0}$	$s_{2,1}$
$s_{3,0}$	$s_{3,1}$
$s_{4,0}$	$s_{4,1}$
$s_{5,0}$	$s_{5,1}$
$s_{6,0}$	$s_{6,1}$
$s_{7,0}$	$s_{7,1}$

→

Вихідна послідовність	
out_0	out_8
out_1	out_9
out_2	out_{10}
out_3	out_{11}
out_4	out_{12}
out_5	out_{13}
out_6	out_{14}
out_7	out_{15}

Для інших довжин блока іншою є кількість стовпців N_b.

Незвідний поліном шифру. Операція лінійного розсіювання (перемішування в стовпці) алгоритму "Калина" використовує поліноміальне представлення байтів у полі Галуа $GF\left(2^8\right)$. Таке поле формують незвідним поліномом. В алгоритмі "Калина" застосовують такий незвідний поліном:

$$f(x) = x^8 + x^4 + x^3 + x^2 + 1,$$

або $\{0\,1\}\ \{0\,\mathrm{d}\}$ в шістнадцятковому поданні.

Розмір блока даних та довжина ключа шифрування. Алгоритм шифрування "Калина" шифрує блоки даних розміром 128, 256 і 512 біти із використанням ключа шифрування завдовжки 128, 256 і 512 біти. Довжина ключа збігається з розміром блока або удвічі більша за нього. Допустимі комбінації розміру блока та довжини ключа шифрування подано в табл. 3.13.

Таблиця 3.13

Комбінації розміру блока та довжини ключа шифрування

Розмір блока, бітів	Довжина ключа, бітів
128 ($N_b = 2$)	128 ($N_k = 2$), 256 ($N_k = 4$)
256 ($N_b = 4$)	256 ($N_k = 4$), 512 ($N_k = 8$)
512 ($N_b = 8$)	512 ($N_k = 8$)

Кількість раундів шифрування. Алгоритм шифрування є процедурою, що складається з попереднього й прикінцевого забілювання (яке здійснюють додаванням за модулем 2^{64}) та ітеративного раундового перетворення. На вхід кожного раундового перетворення подають поточний стан, а також необхідну кількість ключової інформації (раундовий ключ). Кількість раундів шифрування (N_r) залежить від довжини ключа (табл. 3.14). На початку та в кінці процедур зашифрування / розшифрування виконують додаткові операції забілювання.

Таблиця 3.14

Кількість раундів шифрування алгоритму "Калина" (N_r)

Розмір блока, бітів	Кількість раундів шифрування для різних довжин ключа		
	Довжина ключа 128 бітів ($N_k = 2$)	Довжина ключа 256 бітів ($N_k = 4$)	Довжина ключа 512 бітів ($N_k = 8$)
128 ($N_b = 2$)	10	14	–
256 ($N_b = 4$)	–	14	18
512 ($N_b = 8$)	–	–	18

Зашифрування. На вхід процедури зашифрування подають відкритий текст і раундові ключі. Відкритий текст подають у вигляді поточного стану шифру. Після закінчення зашифрування отриманий шифртекст видають як послідовність байтів.

Перед здійсненням раундів зашифрування в алгоритмі "Калина" виконують перетворення ***Add64RoundKey*** з ключем 0-го раунду. Потім виконують N_{r-1} раундів зашифрування, кожен з яких містить такі перетворення:

– ***Kalyna_S_boxes;***
– ***ShiftRows;***
– ***MixColumns;***
– ***XORRoundKey.***

Після цього виконують останній, N_r-й раунд, який відрізняється від попередніх тільки застосуванням перетворення ***Add64RoundKey*** замість перетворення ***XORRoundKey***.

Розглянемо особливості згаданих перетворень шифру.

XORRoundKey. Тут виконують побітове додавання за модулем 2 (XOR) поточного стану та раундового ключа. Результатом операції є новий поточний стан. Раундовий ключ подають у вигляді матриці, аналогічно до подання поточного стану шифру.

Для поточного стану $A=(a_{i,j})$ та циклового ключа $K=(k_{i,j})$ результат цього перетворення $B=(b_{i,j})=A\oplus K$ обчислюють за формулою $b_{i,j}=a_{i,j}\oplus k_{i,j}$, $0\le i<8$, $0\le j<N_b$. Тобто відповідні байти поточного стану та раундового ключа додають за модулем 2. Приклад цього перетворення для розміру блока 256 бітів подано на рис. 3.26.

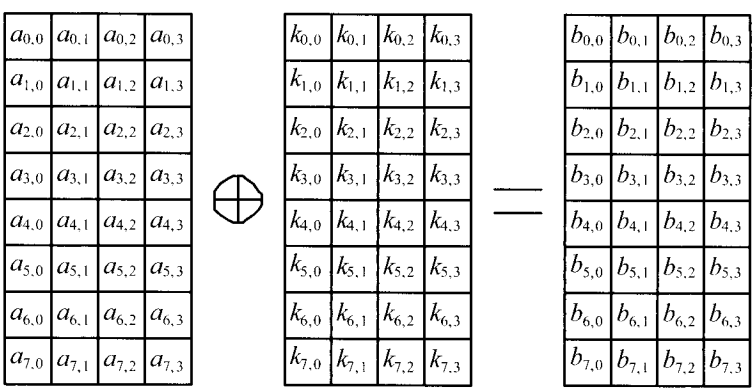


*Рис. 3.26. Перетворення **XORRoundKey** для розміру блока 256 бітів*

Add64RoundKey. Тут виконують додавання за модулем 2^{64} 64-бітних слів поточного стану та раундового ключа. Результатом операції є новий поточний стан. Фактично тут додають за модулем 2 відповідні стовпці поточного стану шифру та раундового ключа. При цьому молодшим бітом 64-бітового блока буде молодший біт байта рядка з найменшим номером, старшим бітом 64-бітового блока – старший біт байта рядка з найбільшим номером (формат little endian).

Для поточного стану $A=(a_i)$ (де i – кількість стовпців матриці поточного стану), де кожен стовпчик має значення: $a_i = a_{0,i} \cdot 2^{0\times 8} + a_{1,i} \cdot 2^{1\times 8} + \\ + ... + a_{7,i} \cdot 2^{7\times 8}$ $(0 \le i < N_b)$, а також для раундового ключа $K=(k_i)$ в аналогічному представленні, результат перетворення $B=(b_i)$ обчислюють за формулою: $b_i = (a_i + k_i) \bmod 2^{64}$.

Схему розбиття на 64-бітові блоки 128-бітового поточного стану та раундового ключа такого самого розміру з подальшим перетворенням ***Add64RoundKey*** подано на рис. 3.27.

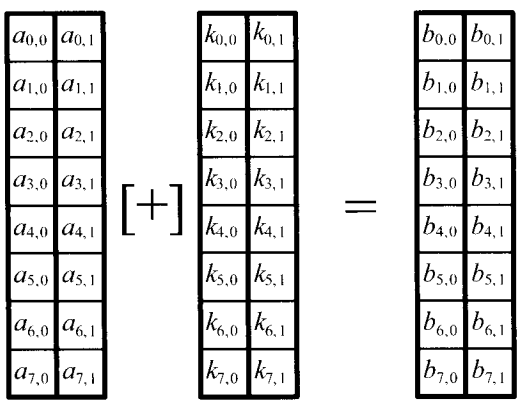


*Рис. 3.27. Перетворення **Add64RoundKey** для величини блока 128 бітів*

Перетворення ***Add64RoundKey*** для станів розміром 256 і 512 біти відрізнятиметься лише кількістю 64-бітових блоків, на які розбивають стан та цикловий ключ (кількістю стовпчиків).

Kalyna_S_boxes – це основний етап алгоритму, на якому замінюють кожний байт поточного стану відповідно до заданої або згенерованої таблиці підстановки. Задано (рекомендовано) чотири таблиці підстановок "байт-у-байт". Причому для байтів одного рядка поточного стану шифру застосовано одну й ту саму підстановку:

для байтів 0-го рядка (елементи $s_{0,i}$) – підстановка $S0$;

для байтів 1-го рядка (елементи $s_{1,i}$) – підстановка $S1$;

для байтів 2-го рядка (елементи $s_{2,i}$) – підстановка $S2$;

для байтів 3-го рядка (елементи $s_{3,i}$) – підстановка $S3$;

для байтів 4-го рядка (елементи $s_{4,i}$) – підстановка $S0$;

для байтів 5-го рядка (елементи $s_{5,i}$) – підстановка $S1$;

для байтів 6-го рядка (елементи $s_{6,i}$) – підстановка $S2$;

для байтів 7-го рядка (елементи $s_{7,i}$) – підстановка $S3$.

Стандартом рекомендовано такі таблиці підстановки (табл. 3.15):

Таблиця 3.15

S-блоки алгоритму "Калина"

Підстановка S0															
A8	43	5F	06	6B	75	6C	59	71	DF	87	95	17	F0	D8	09
6D	F3	1D	CB	C9	4D	2C	AF	79	E0	97	FD	6F	4B	45	39
3E	DD	A3	4F	B4	B6	9A	0E	1F	BF	15	E1	49	D2	93	C6
92	72	9E	61	D1	63	FA	EE	F4	19	D5	AD	58	A4	BB	A1
DC	F2	83	37	42	E4	7A	32	9C	CC	AB	4A	8F	6E	04	27
2E	E7	E2	5A	96	16	23	2B	C2	65	66	0F	BC	A9	47	41
34	48	FC	B7	6A	88	A5	53	86	F9	5B	DB	38	7B	C3	1E
22	33	24	28	36	C7	B2	3B	8E	77	BA	F5	14	9F	08	55
9B	4C	FE	60	5C	DA	18	46	CD	7D	21	B0	3F	1B	89	FF
EB	84	69	3A	9D	D7	D3	70	67	40	B5	DE	5D	30	91	B1
78	11	01	E5	00	68	98	A0	C5	02	A6	74	2D	0B	A2	76
B3	BE	CE	BD	AE	E9	8A	31	1C	EC	F1	99	94	AA	F6	26
2F	EF	E8	8C	35	03	D4	7F	FB	05	C1	5E	90	20	3D	82
F7	EA	0A	0D	7E	F8	50	1A	C4	07	57	B8	3C	62	E3	C8
AC	52	64	10	D0	D9	13	0C	12	29	51	B9	CF	D6	73	8D
81	54	C0	ED	4E	44	A7	2A	85	25	E6	CA	7C	8B	56	80
Підстановка S1															
CE	BB	EB	92	EA	CB	13	C1	E9	3A	D6	B2	D2	90	17	F8
42	15	56	B4	65	1C	88	43	C5	5C	36	BA	F5	57	67	8D
31	F6	64	58	9E	F4	22	AA	75	0F	02	B1	DF	6D	73	4D
7C	26	2E	F7	08	5D	44	3E	9F	14	C8	AE	54	10	D8	BC
1A	6B	69	F3	BD	33	AB	FA	D1	9B	68	4E	16	95	91	EE
4C	63	8E	5B	CC	3C	19	A1	81	49	7B	D9	6F	37	60	CA
E7	2B	48	FD	96	45	FC	41	12	0D	79	E5	89	8C	E3	20
30	DC	B7	6C	4A	B5	3F	97	D4	62	2D	06	A4	A5	83	5F
2A	DA	C9	00	7E	A2	55	BF	11	D5	9C	CF	0E	0A	3D	51
7D	93	1B	FE	C4	47	09	86	0B	8F	9D	6A	07	B9	B0	98
18	32	71	4B	EF	3B	70	A0	E4	40	FF	C3	A9	E6	78	F9
8B	46	80	1E	38	E1	B8	A8	E0	0C	23	76	1D	25	24	05
F1	6E	94	28	9A	84	E8	A3	4F	77	D3	85	E2	52	F2	82
50	7A	2F	74	53	B3	61	AF	39	35	DE	CD	1F	99	AC	AD
72	2C	DD	D0	87	BE	5E	A6	EC	04	C6	03	34	FB	DB	59
B6	C2	01	F0	5A	ED	A7	66	21	7F	8A	27	C7	C0	29	D7

Продовження табл. 3.15

Підстановка S2															
93	D9	9A	B5	98	22	45	FC	BA	6A	DF	02	9F	DC	51	59
4A	17	2B	C2	94	F4	BB	A3	62	E4	71	D4	CD	70	16	E1
49	3C	C0	D8	5C	9B	AD	85	53	A1	7A	C8	2D	E0	D1	72
A6	2C	C4	E3	76	78	B7	B4	09	3B	0E	41	4C	DE	B2	90
25	A5	D7	03	11	00	C3	2E	92	EF	4E	12	9D	7D	CB	35
10	D5	4F	9E	4D	A9	55	C6	D0	7B	18	97	D3	36	E6	48
56	81	8F	77	CC	9C	B9	E2	AC	B8	2F	15	A4	7C	DA	38
1E	0B	05	D6	14	6E	6C	7E	66	FD	B1	E5	60	AF	5E	33
87	C9	F0	5D	6D	3F	88	8D	C7	F7	1D	E9	EC	ED	80	29
27	CF	99	A8	50	0F	37	24	28	30	95	D2	3E	5B	40	83
B3	69	57	1F	07	1C	8A	BC	20	EB	CE	8E	AB	EE	31	A2
73	F9	CA	3A	1A	FB	0D	C1	FE	FA	F2	6F	BD	96	DD	43
52	B6	08	F3	AE	BE	19	89	32	26	B0	EA	4B	64	84	82
6B	F5	79	BF	01	5F	75	63	1B	23	3D	68	2A	65	E8	91
F6	FF	13	58	F1	47	0A	7F	C5	A7	E7	61	5A	06	46	44
42	04	A0	DB	39	86	54	AA	8C	34	21	8B	F8	0C	74	67
Підстановка S3															
68	8D	CA	4D	73	4B	4E	2A	D4	52	26	B3	54	1E	19	1F
22	03	46	3D	2D	4A	53	83	13	8A	B7	D5	25	79	F5	BD
58	2F	0D	02	ED	51	9E	11	F2	3E	55	5E	D1	16	3C	66
70	5D	F3	45	40	CC	E8	94	56	08	CE	1A	3A	D2	E1	DF
B5	38	6E	0E	E5	F4	F9	86	E9	4F	D6	85	23	CF	32	99
31	14	AE	EE	C8	48	D3	30	A1	92	41	B1	18	C4	2C	71
72	44	15	FD	37	BE	5F	AA	9B	88	D8	AB	89	9C	FA	60
EA	BC	62	0C	24	A6	A8	EC	67	20	DB	7C	28	DD	AC	5B
34	7E	10	F1	7B	8F	63	A0	05	9A	43	77	21	BF	27	09
C3	9F	B6	D7	29	C2	EB	C0	A4	8B	8C	1D	FB	FF	C1	B2
97	2E	F8	65	F6	75	07	04	49	33	E4	D9	B9	D0	42	C7
6C	90	00	8E	6F	50	01	C5	DA	47	3F	CD	69	A2	E2	7A
A7	C6	93	0F	0A	06	E6	2B	96	A3	1C	AF	6A	12	84	39
E7	B0	82	F7	FE	9D	87	5C	81	35	DE	B4	A5	FC	80	EF
CB	BB	6B	76	BA	5A	7D	78	0B	95	E3	AD	74	98	3B	36
64	6D	DC	F0	59	A9	4C	17	7F	91	B8	C9	57	1B	E0	61

На рис. 3.28 подано приклад заміни байта під час перетворення.

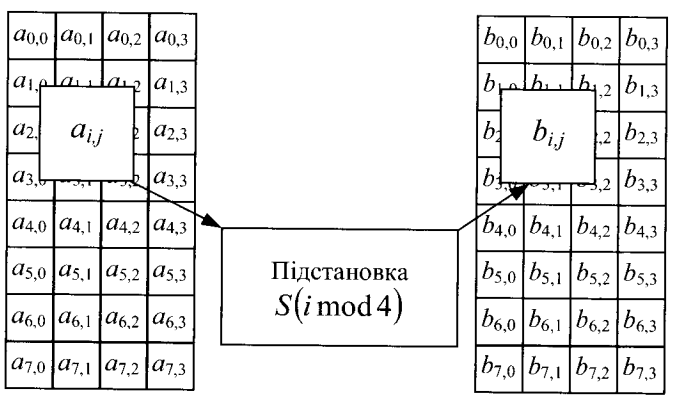


Рис. 3.28. Приклад підстановки байта для розміру блока 256 бітів

Заміна одного байта полягає у виборі з таблиці підстановки нового значення за адресою, яку задає поточне значення байта. Нове вибране значення і є результатом підстановки для поточного байта.

Приклад підстановки (за допомогою таблиці $S0$) для байта із шістнадцятковим значенням 5A подано в табл. 3.16.

Таблиця 3.16

Приклад підстановки за допомогою таблиці $S0$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	A8	43	5F	06	6B	75	6C	59	71	DF	87	95	17	F0	D8	09
1	6D	F3	1D	CB	C9	4D	2C	AF	79	E0	97	FD	6F	4B	45	39
2	3E	DD	A3	4F	B4	B6	9A	0E	1F	BF	15	E1	49	D2	93	C6
3	92	72	9E	61	D1	63	FA	EE	F4	19	D5	AD	58	A4	BB	A1
4	DC	F2	83	37	42	E4	7A	32	9C	CC	AB	4A	8F	6E	04	27
5	2E	E7	E2	5A	96	16	23	2B	C2	65	66	0F	BC	A9	47	41
6	34	48	FC	B7	6A	88	A5	53	86	F9	5B	DB	38	7B	C3	1E
7	22	33	24	28	36	C7	B2	3B	8E	77	BA	F5	14	9F	08	55
8	9B	4C	FE	60	5C	DA	18	46	CD	7D	21	B0	3F	1B	89	FF
9	EB	84	69	3A	9D	D7	D3	70	67	40	B5	DE	5D	30	91	B1
A	78	11	01	E5	00	68	98	A0	C5	02	A6	74	2D	0B	A2	76
B	B3	BE	CE	BD	AE	E9	8A	31	1C	EC	F1	99	94	AA	F6	26
C	2F	EF	E8	8C	35	03	D4	7F	FB	05	C1	5E	90	20	3D	82
D	F7	EA	0A	0D	7E	F8	50	1A	C4	07	57	B8	3C	62	E3	C8
E	AC	52	64	10	D0	D9	13	0C	12	29	51	B9	CF	D6	73	8D
F	81	54	C0	ED	4E	44	A7	2A	85	25	E6	CA	7C	8B	56	80

Отже, старші 4 біти вхідного байта визначають номер рядка, молодші 4 біти – номер стовпця. Результат підстановки для значення 5A – це шістнадцяткове число 66, що знаходиться в таблиці на перетині 6-го рядка (з шістнадцятковим номером 5) та 11-го стовпця (з номером A).

Для перетворення можна використовувати інший набір підстановок (від 1 до 8 таблиць), відмінний від рекомендованих. У цьому випадку набір підстановок має постачатися в установленому порядку й може бути додатковим секретним параметром шифру, як і ключ шифрування.

ShiftRows. Під час цього перетворення рівномірно розподілюють байти кожного 64-бітового стовпця серед інших стовпців. Цього досягають циклічним зсувом рядків стану праворуч на різну кількість байтів. Значення зсувів залежать від розміру блока й подані в табл. 3.17. Рис. 3.29 пояснює порядок виконання перетворення ***ShiftRows*** для різних розмірів блока.

Таблиця 3.17

Значення циклічних зсувів рядків для різних розмірів блока

Номер рядка	Значення зсуву, байтів		
	Довжина блока 128 бітів	Довжина блока 256 бітів	Довжина блока 512 бітів
0	0	0	0
1	0	0	1
2	0	1	2
3	0	1	3
4	1	2	4
5	1	2	5
6	1	3	6
7	1	3	7

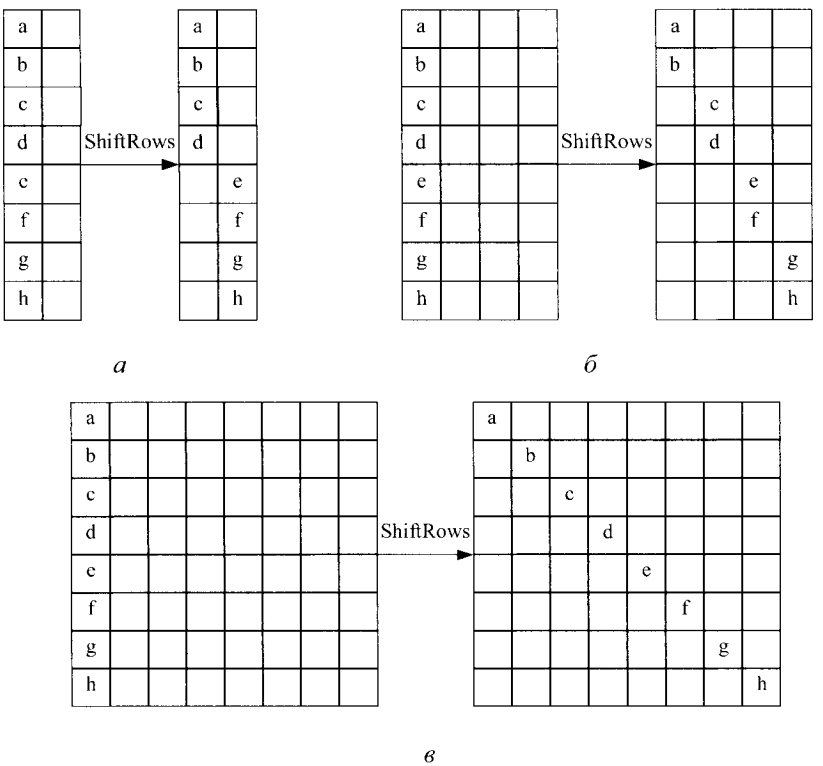


Рис. 3.29. Порядок розподілу байтів першого стовпця під час перетворення ShiftRows: а – 128-бітовий блок; б – 256-бітовий блок; в – 512-бітовий блок

MixColumns. Під час цього перетворення виконують послідовно оброблення усіх стовпців поточного стану. Кожен 8-байтовий стовпець розглядають як поліном над полем $GF(2^8)$, що складається із суми 8 одночленів (кожен байт представляють у вигляді елемента поля $GF(2^8)$, і він є коефіцієнтом перед змінною степеня, що дорівнює індексу байта в стовпці). Під час перетворення цей поліном множать за модулем x^8+1 на фіксований поліном $C(x)$, де:

$$C(x)=\{01\}x^7+\{05\}x^6+\{01\}x^5+\{08\}x^4+\{06\}x^3+\{07\}x^2+\{04\}x+\{01\}.$$

Ця операція еквівалентна матричному множенню над $GF(2^8)$ початкового 8-байтного вектора a на фіксовану матрицю, результат зберігають у 8-байтний вектор b:

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} = \begin{bmatrix} 01 & 01 & 05 & 01 & 08 & 06 & 07 & 04 \\ 04 & 01 & 01 & 05 & 01 & 08 & 06 & 07 \\ 07 & 04 & 01 & 01 & 05 & 01 & 08 & 06 \\ 06 & 07 & 04 & 01 & 01 & 05 & 01 & 08 \\ 08 & 06 & 07 & 04 & 01 & 01 & 05 & 01 \\ 01 & 08 & 06 & 07 & 04 & 01 & 01 & 05 \\ 05 & 01 & 08 & 06 & 07 & 04 & 01 & 01 \\ 01 & 05 & 01 & 08 & 06 & 07 & 04 & 01 \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix}.$$

Порядок обчислення елементів результуючого вектора b пояснено наступним співвідношенням, при цьому всі операції множення на байт виконують у полі $GF(2^8)$.

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} = \begin{bmatrix} 01\cdot a_0 \oplus 01\cdot a_1 \oplus 05\cdot a_2 \oplus 01\cdot a_3 \oplus 08\cdot a_4 \oplus 06\cdot a_5 \oplus 07\cdot a_6 \oplus 04\cdot a_7 \\ 04\cdot a_0 \oplus 01\cdot a_1 \oplus 01\cdot a_2 \oplus 05\cdot a_3 \oplus 01\cdot a_4 \oplus 08\cdot a_5 \oplus 06\cdot a_6 \oplus 07\cdot a_7 \\ 07\cdot a_0 \oplus 04\cdot a_1 \oplus 01\cdot a_2 \oplus 01\cdot a_3 \oplus 05\cdot a_4 \oplus 01\cdot a_5 \oplus 08\cdot a_6 \oplus 06\cdot a_7 \\ 06\cdot a_0 \oplus 07\cdot a_1 \oplus 04\cdot a_2 \oplus 01\cdot a_3 \oplus 01\cdot a_4 \oplus 05\cdot a_5 \oplus 01\cdot a_6 \oplus 08\cdot a_7 \\ 08\cdot a_0 \oplus 06\cdot a_1 \oplus 07\cdot a_2 \oplus 04\cdot a_3 \oplus 01\cdot a_4 \oplus 01\cdot a_5 \oplus 05\cdot a_6 \oplus 01\cdot a_7 \\ 01\cdot a_0 \oplus 08\cdot a_1 \oplus 06\cdot a_2 \oplus 07\cdot a_3 \oplus 04\cdot a_4 \oplus 01\cdot a_5 \oplus 01\cdot a_6 \oplus 05\cdot a_7 \\ 05\cdot a_0 \oplus 01\cdot a_1 \oplus 08\cdot a_2 \oplus 06\cdot a_3 \oplus 07\cdot a_4 \oplus 04\cdot a_5 \oplus 01\cdot a_6 \oplus 01\cdot a_7 \\ 01\cdot a_0 \oplus 05\cdot a_1 \oplus 01\cdot a_2 \oplus 08\cdot a_3 \oplus 06\cdot a_4 \oplus 07\cdot a_5 \oplus 04\cdot a_6 \oplus 01\cdot a_7 \end{bmatrix}.$$

На рис. 3.30 зображено порядок виконання перетворення ***MixColumns*** для поточного стану шифру.

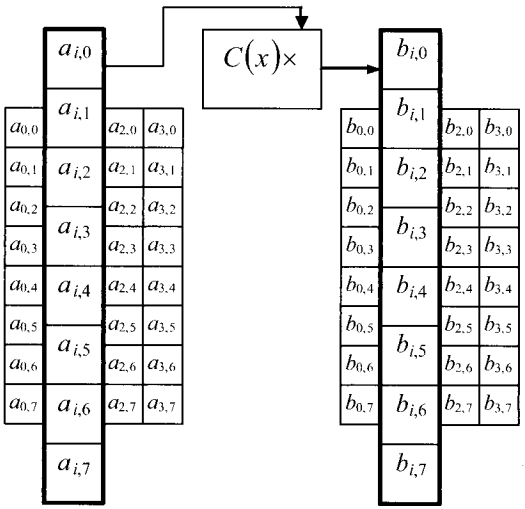


*Рис. 3.30. Порядок перетворення **MixColumns** для поточного стану шифру з розміром блока 256 бітів*

Розшифрування. Процедура розшифрування є оберненою до зашифрування. На вхід подають шифртекст та раундові ключі. Після закінчення розшифрування отриманий відкритий текст формують у вигляді послідовності байтів.

У початковому раунді розшифрування виконують таку послідовність перетворень (із ключем N_r-го раунду):

– ***Sub64RoundKey***;
– ***InvMixColumns***;
– ***InvShiftRows***;
– ***Kalyna_InvS_boxes***.

Потім виконують наступні N_{r-1} раундів зашифрування (з раундовими ключами від N_{r-1} до 1), кожен з яких містить такі перетворення:

XORRoundKey;
InvMixColumns;
InvShiftRows;
Kalyna_InvS_boxes.

Наприкінці розшифрування виконують перетворення ***Sub64RoundKey*** із ключем нульового раунду.

Розглянемо особливості перетворення розшифрування.

Операція ***XORRoundKey*** є оберненою сама до себе (подвійне застосування дає початкове значення). Отже, як під час зашифрування, так і розшифрування виконують операцію ***XORRoundKey***.

Перетворення ***Sub64RoundKey*** є оберненим до ***Add64RoundKey***. Це перетворення полягає в аналогічному розбитті стану та раундового ключа на 64-бітові блоки та відніманні за модулем 2^{64} від блоків стану $B=(b_i)$ відповідних блоків раундового ключа $k=(k_i)$ для отримання стану результату $A=(a_i)$: $a_i=(b_i-k_i)\bmod 2^{64}$, $0\le i<N_b$.

Після виконання операції результат записують на місце першого аргументу. Правила переходу від байтів поточного стану до 64-бітових блоків та навпаки ідентичні до перетворень, що виконують під час ***Add64RoundKey***.

Перетворення ***Kalyna_InvS_boxes*** є оберненим до перетворення ***Kalyna_S_boxes***. Тут замінюють кожний байт поточного стану шифру згідно з таблицями обернених підстановок. Отже, обернена процедура відрізняється від прямої лише таблицями підстановок, а послідовність оберненого перетворення для кожного байта є такою самою, як і прямого перетворення.

Якщо для зашифрування використовували набір підстановок, відмінних від рекомендованого, то для розшифрування застосовують відповідний набір таблиць оберненої підстановки.

Перетворення ***InvShiftRows*** є оберненим до ***ShiftRows***. Здійснюють це перетворення виконанням циклічних зсувів рядків стану на ту саму кількість байтів (табл. 3.17), але ліворуч. На рис. 3.31 зображено послідовність виконання перетворення ***InvShiftRows***.

Перетворення ***InvMixColumns*** є оберненим до ***MixColumns***. Перетворення полягає в перемноженні кожного стовпця, що представлений у вигляді полінома над полем $GF(2^8)$, на обернений для $C(x)$ поліном $D(x)$:

$$D(x)=\{95\}x^7+\{76\}x^6+\{A8\}x^5+\{2F\}x^4+\{49\}x^3+\{D7\}x^2+\{CA\}x+\{AD\}.$$

Ця операція еквівалентна матричному множенню над $GF(2^8)$ початкового 8-байтного вектора a на фіксовану матрицю. Результат заносять до 8-байтового вектора b:

$$\begin{bmatrix}b_0\\b_1\\b_2\\b_3\\b_4\\b_5\\b_6\\b_7\end{bmatrix}=\begin{bmatrix}AD&95&76&A8&2F&49&D7&CA\\CA&AD&95&76&A8&2F&49&D7\\D7&CA&AD&95&76&A8&2F&49\\49&D7&CA&AD&95&76&A8&2F\\2F&49&D7&CA&AD&95&76&A8\\A8&2F&49&D7&CA&AD&95&76\\76&A8&2F&49&D7&CA&AD&95\\95&76&A8&2F&49&D7&CA&AD\end{bmatrix}\times\begin{bmatrix}a_0\\a_1\\a_2\\a_3\\a_4\\a_5\\a_6\\a_7\end{bmatrix}.$$

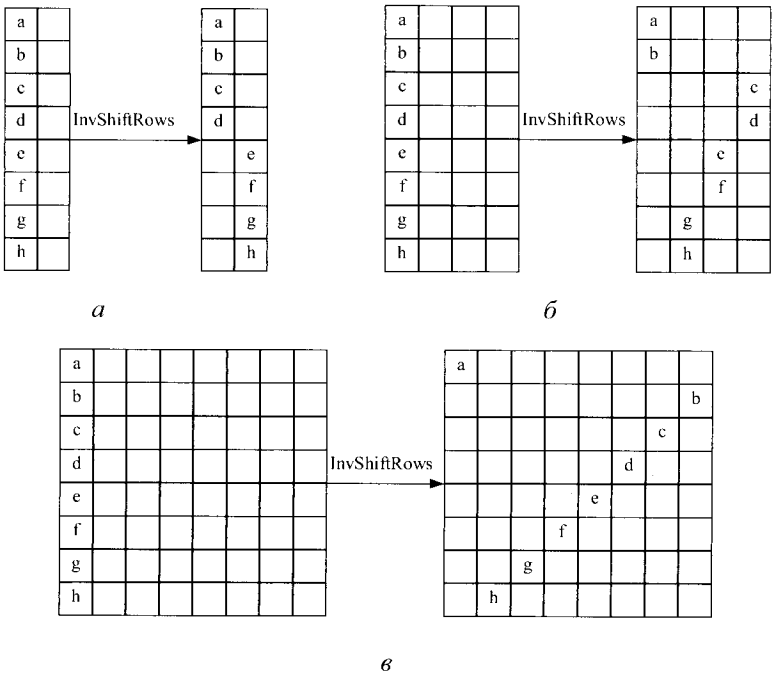


Рис. 3.31. Порядок розподілу байтів першого стовпця при виконанні перетворення ***InvShiftRows:*** *а – 128-бітний блок; б – 256-бітний блок; в – 512-бітний блок*

Порядок обчислення елементів результуючого вектора b пояснює наступне співвідношення:

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} = \begin{bmatrix} AD \cdot a_0 \oplus 95 \cdot a_1 \oplus 76 \cdot a_2 \oplus A8 \cdot a_3 \oplus 2F \cdot a_4 \oplus 49 \cdot a_5 \oplus D7 \cdot a_6 \oplus CA \cdot a_7 \\ CA \cdot a_0 \oplus AD \cdot a_1 \oplus 95 \cdot a_2 \oplus 76 \cdot a_3 \oplus A8 \cdot a_4 \oplus 2F \cdot a_5 \oplus 49 \cdot a_6 \oplus D7 \cdot a_7 \\ D7 \cdot a_0 \oplus CA \cdot a_1 \oplus AD \cdot a_2 \oplus 95 \cdot a_3 \oplus 76 \cdot a_4 \oplus A8 \cdot a_5 \oplus 2F \cdot a_6 \oplus 49 \cdot a_7 \\ 49 \cdot a_0 \oplus D7 \cdot a_1 \oplus CA \cdot a_2 \oplus AD \cdot a_3 \oplus 95 \cdot a_4 \oplus 76 \cdot a_5 \oplus A8 \cdot a_6 \oplus 2F \cdot a_7 \\ 2F \cdot a_0 \oplus 49 \cdot a_1 \oplus D7 \cdot a_2 \oplus CA \cdot a_3 \oplus AD \cdot a_4 \oplus 95 \cdot a_5 \oplus 76 \cdot a_6 \oplus A8 \cdot a_7 \\ A8 \cdot a_0 \oplus 2F \cdot a_1 \oplus 49 \cdot a_2 \oplus D7 \cdot a_3 \oplus CA \cdot a_4 \oplus AD \cdot a_5 \oplus 95 \cdot a_6 \oplus 76 \cdot a_7 \\ 76 \cdot a_0 \oplus A8 \cdot a_1 \oplus 2F \cdot a_2 \oplus 49 \cdot a_3 \oplus D7 \cdot a_4 \oplus CA \cdot a_5 \oplus AD \cdot a_6 \oplus 95 \cdot a_7 \\ 95 \cdot a_0 \oplus 76 \cdot a_1 \oplus A8 \cdot a_2 \oplus 2F \cdot a_3 \oplus 49 \cdot a_4 \oplus D7 \cdot a_5 \oplus CA \cdot a_6 \oplus AD \cdot a_7 \end{bmatrix}.$$

3.7. Асиметричні криптосистеми

Концепцію ***асиметричної криптографії*** [1] (або криптографії з відкритим (публічним) ключем) запропонували ***Уітфілд Діффі*** (Bailey Whitfield 'Whit' Diffie) і ***Мартін Хелман*** (Гелман, Martin E. Hellman), а також,

незалежно, ***Ральф Меркл*** (Ralph Charles Merkle). Діффі та Хелман представили цю ідею в 1976 р. на Національній комп'ютерній конференції США. Через декілька місяців вийшла друком їх базова праця "Нові напрямки в криптографії".

Перша праця Меркла вийшла друком аж в 1978 р., однак саме Мерклу деякі дослідники віддають першість стосовно ідеї асиметричного шифрування (причини цієї ситуації розглянемо нижче).

Ідея відкритих ключів проста, проте знайдена вона була лише наприкінці XX століття (на відміну від тисячолітніх ідей симетричних методів шифрування). Можливо, це сталося через те, що раніше просто не було гострої необхідності в застосуванні альтернативних, відмінних від симетричних, методів шифрування. Основною проблемою симетричної криптографії є необхідність обміну спільним таємним ключем. Якщо сторони, які обмінюються таємними повідомленнями, це люди (а саме так було до другої половини XX століття), проблему ще можна вирішити, і симетричної криптографії достатньо. Якщо ж в інформаційний обмін як сторони вступають машини (комп'ютери), а сам інформаційний обмін здійснюють за допомогою стрімко прогресуючих мережевих технологій, починають виникати непереборні проблеми. Перша проблема – проблема довіри. Адресата-людину ми знаємо, тому довіряємо. Якщо ж адресат – комп'ютер, загублений десь у мережі, довіра до такого адресата стрімко слабне. Друга проблема – щораз більша кількість користувачів комп'ютерних мереж. Якщо кожна пара користувачів захоче обмінюватися таємною інформацією, вони мусять обмінятися своїм унікальним таємним ключем. Для іншої пари ключ має бути інший. Кількість ключів стрімко зростає, і кожен треба передати таємно. Нарешті ці проблеми стають непереборними в умовах сучасних електронних комунікаційних можливостей і починають різко обмежувати сферу застосування симетричної криптографії.

Математичний апарат сучасних асиметричних криптосистем, особливо без спеціальної й ґрунтовної математичної підготовки, є складним. Тому основи криптографії з відкритим ключем пояснюють на простих прикладах.

На примітивному рівні симетричне та асиметричне шифрування порівнюють із сейфом. Користувач А купує сейф із двома однаковими ключами. Один ключ він надсилає користувачеві В. Після цього вони можуть обмінюватися таємними повідомленнями, надсилаючи їх один одному в закритому сейфі. Це і є модель симетричного шифрування.

Припустимо тепер, що сейф обладнано замком, який автоматично закривається. Ситуація кардинально змінюється. Тепер користувач В може купити такий сейф з одним ключем, ключ залишити в себе, а відкритий сейф занести на пошту й повідомити всім охочим, де він є. Після цього кожен, не

тільки користувач А, може покласти своє таємне повідомлення в сейф, закрити його й надіслати користувачеві В. Після закриття сейфа ніхто, крім користувача В, відчинити сейф не зможе (навіть відправник – користувач А), адже тільки в користувача В є ключ. Замок, який автоматично закривається, має ту властивість, що закрити його (втаємничити своє повідомлення) легко може кожен, а от відкрити його – значно складніше завдання. Однак для того, хто має ключ, це завдання є простим.

Уникнути необхідності передавання таємного ключа можна й без складного замка. Проте доведеться збільшити видатки на пересилання сейфа. Користувач А може купити сейф без замка й навісний замок з одним ключем (для себе). Користувач В купує свій навісний замок з одним ключем. Користувач А кладе своє повідомлення в сейф, закриває своїм замком і надсилає користувачеві В. Користувач В відкрити замок користувача А не може. Натомість він додатково закриває сейф своїм замком і повертає сейф користувачеві А. Користувач А знімає свій замок і знову надсилає сейф користувачеві В, якому залишається зняти свій замок і відкрити сейф. Так, проблему передавання ключа усунуто.

Отже, криптографія з відкритим ключем дає змогу уникнути необхідності обміну таємним ключем. Розглянемо просту асиметричну криптосистему, на прикладі якої сформулюємо основні поняття й принципи шифрування з відкритим ключем.

3.7.1. Криптосистема на основі телефонного довідника

Нехай користувач В має намір отримувати таємні повідомлення від користувача А, друзів, колег тощо [1]. Він купує багатотомний телефонний довідник великого міста й виконує достатньо складну й тривалу роботу: пересортовує його за номерами телефонів (пересортований довідник він ховає подалі). Потім користувач В повідомляє користувачеві А та всім охочим, що вони можуть купити телефонний довідник і скористатися ним для шифрування своїх повідомлень. Шифрування просте: кожну букву повідомлення необхідно замінити на номер телефону абонента, прізвище якого починається із цієї букви. Причому ключ зашифрування (звичайний телефонний довідник) продається в кожному кіоску (він не є таємним, навпаки – він публічний).

Припустимо, суперник перехопив зашифроване повідомлення. Метод і ключ зашифрування йому відомі. Він купує собі довідник, однак стикається із проблемою: відшукати в багатотомному довіднику номери телефонів, які фігурують у шифртексті, дуже складно. Тому дешифрувати повідомлення

суперник не може. Натомість користувач В, отримавши повідомлення, дістає свій таємний, пересортований за номерами телефонів, довідник, і за його допомогою легко розшифровує повідомлення.

Отже, у криптосистемі є два різні ключі. Один із них – ***ключ зашифрування – є публічним*** (звичайний, посортований за прізвищами абонентів, довідник). Другий ключ – ***ключ розшифрування – є таємним*** (пересортований за номерами телефонів довідник). Його має тільки користувач В і нікому його не передає. Причому визначити цей ключ, знаючи публічний ключ зашифрування, дуже непросто (треба пересортувати довідник). ***Функція зашифрування проста***: вибір із довідника номера телефону за буквою, з якої починається прізвище абонента. Для суперника ***функція дешифрування складна***: пошук номерів телефону в довіднику, посортованому за прізвищами. Натомість користувач В має таємний ключ (пересортований довідник), який дає могу йому застосувати ***просту функцію розшифрування*** – пошук номерів телефону в довіднику, посортованому за цими номерами. В основу процесу шифрування покладено ***односторонню (one way)***, або ***важкооборотну функцію***. Її особливість у тому, що обчислити значення функції для заданого значення аргументу просто, однак дуже складно дізнатися значення аргументу, маючи значення функції. У розглянутому прикладі з телефонним довідником у якості такої функції є вибір із довідника номера телефону за буквою, з якої починається прізвище абонента. Такий вибір здійснити легко. Проте зворотна задача – знайти заданий номер телефону в довіднику – є складною. Разом із тим, для користувача В існує можливість обійти цю складну проблему. Тобто важкооборотна функція зашифрування має секрет, який знає користувач В. Цей секрет – пересортований довідник (таємний ключ). Володіння цим секретом дає можливість користувачеві В не обертати функцію зашифрування (що мусить робити суперник), а скористатися альтернативним простим алгоритмом розшифрування.

3.7.2. Головоломки Меркла

Ральф Меркл [1] на початку 70-х років XX століття слухав курс ***Ланса Хоффмана*** (Lance Hoffman) з комп'ютерної безпеки в Каліфорнійському університеті. У 1974 р. Меркл написав курсову роботу "Секретний зв'язок у відкритих системах" і подав її на захист Хофману. Хофман ідеї Меркла не зрозумів і роботу не зарахував. Спроби Меркла відстояти свої ідеї успіху не мали, і він припинив відвідувати курс Хофмана. Проте Меркл продовжував просувати свою роботу, однак тільки після опублікування статті Діффі та Хелмана роботу Меркла визнали. Саме ця історія призвела до того, що Мерклу

віддають першість у винайденні принципів асиметричної криптографії, хоча його праця була опублікована вже після роботи Діффі та Хелмана.

Система шифрування, яку запропонував Меркл, відома за назвою ***Головоломки Меркла*** [1]. Узагалі кажучи, ця система більше схожа на протокол обміну ключем, проте ідеї асиметричного шифрування тут простежуються.

Нехай на цей момент надійна довжина ключа симетричної криптосистеми становить 40 бітів. Як користувачам обмінятись таким ключем?

Користувач А генерує 2^{20} (близько мільйона) повідомлень виду "Це повідомлення №X: ключ шифрування – Y". Тобто в кожному повідомлені вказано його номер, а також варіант надійного 40-бітного ключа симетричної криптосистеми. Усі повідомлення користувач А зашифровує 20-бітним ключем і надсилає користувачеві В. Той обирає одне повідомлення й зламує його брутальною атакою (це йому зробити складно, але можливо, адже повідомлення зашифроване ключем, довжина якого значно менша від необхідної для стійкого шифрування). За зламаним повідомленням користувач В дізнається його номер і варіант надійного 40-бітного ключа. Потім користувач В відкрито посилає користувачеві А номер прочитаного повідомлення, за яким користувач А встановлює той самий 40-бітний ключ, який отримав користувач В. Із цього моменту користувачі мають той самий ключ і можуть обмінюватися повідомленнями за допомогою симетричної криптосистеми з 40-бітновим ключем.

Суперник може легко перехопити будь-яке з мільйона повідомлень та зламати його. Проте він не знає, яке повідомлення вибрав користувач В. Щоправда, він має також номер обраного повідомлення, який було відкрито надіслано користувачеві А. Але суперник не знає, у якому саме із зашифрованих повідомлень цей номер містився. Щоб дізнатися це, Супернику в найгіршому випадку треба зламати всі 2^{20} повідомлень, кожне з яких зашифровано 20-бітним ключем. А це еквівалентне зламу криптосистеми з 40-бітним ключем, яка є стійкою.

3.7.3. Важкооборотні функції

В основу всіх асиметричних криптосистем покладено ***важкооборотні функції***. Ідеальні важкооборотні функції називають ***криптографічними***. На сьогодні не відомо жодної криптографічної функції [2].

Означення 1. Важкооборотною (односторонньою), або криптографічною називають функцію, для якої ***не існує*** простих (поліноміальних за складністю) алгоритмів обчислення оберненої функції.

Оскільки невідомо жодної функції, яка відповідає означенню 1, реальні асиметричні криптосистеми будують на основі важкооборотних функцій, які відповідають "пом'якшеному" означенню 2:

Означення 2. Важкоборотною (односторонньою, one way) називають функцію, для якої ***не відомо*** простих алгоритмів обчислення оберненої функції.

Заміна поняття "не існує" на "не відомо" означає, що над усіма асиметричними шифрами висить загроза їх розкриття. Сьогодні асиметричні шифри є шифрами з недоведеною стійкістю. Тобто їх обчислювальна стійкість теоретично не доведена. Стійкість сучасних асиметричних криптосистем доводять емпірично, як правило, через проведення конкурсу на злам шифру. Показники найшвидшого в конкурсі алгоритму зламу шифру і є емпіричною оцінкою стійкості цього шифру. Тому впровадження практично значущих асиметричних шифрів є вартісним і тривалим процесом. Причому великі разові витрати (наприклад, на винагороду переможцю конкурсу на злам шифру) і добра емпірична оцінка в окремому конкурсі ще не гарантують практичної придатності шифру. Для отримання довіри споживачів необхідна тривала історія криптоаналізу шифру.

Важкоборотних функцій, які відповідають означенню 2, є багато, проте далеко не всі ці функції придатні для побудови асиметричної криптосистеми. Потрібно, щоб така функція мала "секретний хід", знання якого дає змогу легальному користувачу обійти необхідність обчислення оберненої функції й скористатися простішим способом розшифрування інформації. Пошук цього (або іншого, не відомого раніше) секрету – ще одна додаткова ціль для криптоаналітика.

Незважаючи на велику кількість відомих важкоборотних функцій, сьогодні є лише три асиметричні криптосистеми (і відповідно важкооборотні функції), які вважають надійними. Це криптосистеми RSA, Ель-Гамала та Рабіна. Причому на поширений сьогодні математичний апарат еліптичних кривих ефективно перенесено лише одну з них – систему Ель-Гамала (на основі показникової модульної важкооборотної функції). Важкоборотні функції дають можливість розв'язати не лише задачу конфіденційності зв'язку, а й інші задачі захисту інформації, зокрема такі, для яких раніше не існувало розв'язку. Наприклад, цифровий підпис, надійні протоколи аутентифікації, електронне жеребкування, електронні дистанційні вибори тощо.

3.7.4. Ранцеві криптосистеми

На побутовому рівні, напевно, кожен знає, що задача пакування рюкзака є непростою, особливо коли різних предметів хочеться взяти багато, а вага рюкзака обмежена. Подібна задача відома також у математиці й належить до

класу важких задач. Час розв'язування цієї задачі росте експоненційно відносно кількості предметів, з яких можна вибирати [1].

Алгоритм рюкзака (ранця) для шифрування розробили ***Ральф Меркл*** та ***Мартін Хелман***. Цей алгоритм став першим алгоритмом шифрування з відкритим ключем, призначеним для широкого вжитку. Спочатку алгоритм використовували лише для шифрування. Згодом ***Аді Шамір*** модифікував його так, щоб алгоритм підтримував також цифровий підпис. Безпека цього, а також інших ранцевих алгоритмів, грунтується на складній задачі пакування рюкзака. Із часом було виявлено, що алгоритм рюкзака Меркла–Хелмана є нестійкий. Проте цей алгоритм є ілюстративний і на його прикладі, як правило, розглядають основні поняття та інструменти асиметричної криптографії.

Задачу, або проблему пакування рюкзака формулюють так: дано множину предметів різної ваги. Визначити, чи можна покласти деякі із цих предметів у рюкзак так, щоб його вага дорівнювала заданому значенню. Формалізують задачу так: дано набір значень $M_1, M_2, \cdots M_n$ (ваги предметів), а також підсумкове значення S. Необхідно визначити такі b_i, що:

$$S = b_1 M_1 + b_2 M_2 + \cdots + b_n M_n,$$

де коефіцієнти b_i можуть бути нулем або одиницею. Значення $b_i = 1$ означає, що i-й предмет кладуть до ранця, а $b_i = 0$ – не кладуть.

Нехай, для прикладу, ваги предметів мають значення: 1, 5, 6, 11, 14, 20.

Можна спакувати рюкзак так, щоб його вага дорівнювала 22. Для цього беремо предмети з вагами 5, 6, 11. Разом із тим неможливо спакувати рюкзак, щоб його вага дорівнювала 24. Тобто задача пакування рюкзака не завжди має розв'язок.

Час, необхідний для розв'язання задачі пакування рюкзака, у загальному випадку зростає експоненційно зі збільшенням кількості предметів, з яких вибирають.

В основу алгоритму, який запропонували Меркл та Хелман, покладено ідею шифрування на основі розв'язання серії задач пакування рюкзака. При цьому при зашифруванні предмети до рюкзака вибирають за допомогою блока відкритого тексту. Довжина блока в бітах дорівнює кількості предметів, з яких вибирають. Біти блока відкритого тексту відповідають значенням b_i, а шифртекстом кожного блока є результуюча вага.

Розглянемо приклад зашифрування [1] чотирьох блоків відкритого тексту по шість бітів у кожному, якщо ваги предметів, з яких деякі відбирають у рюкзак, мають значення: 1, 5, 6, 11, 14, 20.

Приклад.

Відкритий текст	010110	100100	110001	000000
Рюкзак	1 5 6 11 14 20	1 5 6 11 14 20	1 5 6 11 14 20	1 5 6 11 14 20
Шифртекст	5+11+14=30	1+11=12	1+5+20=26	0

Таким чином, у розглянутому прикладі відкритому бінарному тексту 010110100100110001000000 відповідає шифртекст: 30, 12, 26, 0.

Очевидно, що зашифрування тут не є проблемою. А от під час розшифрування для кожного блока шифртексту потрібно розв'язувати складну задачу пакування рюкзака. Така ситуація є типовою для асиметричних криптосистем. Звичайно, у розглянутому простому прикладі цю задачу розв'язати відносно просто. Адже предметів для вибору є небагато, і перебором швидко можна встановити набір предметів, сума ваг яких становитиме задану вагу рюкзака. Але якщо кількість предметів для вибору велика, процедура дешифрування різко ускладнюється. Причому з додаванням одного предмета складність задачі зростає вдвічі.

Отже, маємо типову для асиметричних криптосистем ситуацію: зашифрування просте, дешифрування – складне. Як можна спростити розшифрування для легального користувача (отримувача повідомлення)? Для легального користувача будують інший рюкзак. Існує дві задачі пакування рюкзака – одна з них складна, іншу розв'язати легко, і вона має лінійну трудомісткість. При цьому легкий для пакування рюкзак можна перетворити на еквівалентний складний рюкзак. У рюкзачній криптосистемі секретним ключем є легкий для пакування рюкзак. За ним користувач може легко розшифрувати повідомлення. На основі цього легкого рюкзака користувач синтезує еквівалентний складний і публікує його. Це і є відкритий ключ зашифрування.

Надзростаючі рюкзаки. Якщо перелік ваг предметів рюкзака є надзростаючою послідовністю, задачу пакування рюкзака розв'язати легко.

Надзростаючою називають послідовність, у якій кожен елемент більший за суму всіх попередніх елементів. Наприклад, послідовність {1, 3, 6, 13, 27, 52} є надзростаючою, а послідовність {1, 3, 4, 9, 15, 25} такою не є.

Задачу пакування надзростаюкого рюкзака розв'язати просто. Спочатку беруть як поточну повну вагу рюкзака. Порівнюють її з найбільшим елементом послідовності. Якщо цей елемент менший або дорівнює поточній вазі, то його беруть обов'язково. Адже в іншому випадку сума всіх інших елементів послідовності не дасть нам навіть такого числа. Якщо ж вага найбільшого елемента більша за поточну, цей елемент у рюкзак не кладуть.

Якщо найбільший елемент вибрано, поточну вагу зменшують на вагу цього елемента й порівнюють із наступним елементом рюкзака за аналогічним

алгоритмом вибору елемента. І так продовжують, поки поточна вага не перетвориться на нуль. Якщо в процесі таких перетворень поточну вагу вдалося зменшити до 0, розв'язок задачі рюкзака знайдено. Якщо ні – його не існує. Причому за такого алгоритму додавання до набору елементів одного елемента додає до алгоритму лише один крок. Тобто це є ***лінійний ріст трудомісткості***.

Приклад. Нехай задано надзростаючий рюкзак: {2, 3, 6, 13, 27, 52}. Розв'язати задачу рюкзака, якщо його вага дорівнює 70.

Розв'язок. Найбільша вага предмета 52. Це менше ніж 70, тому кладемо цей предмет у рюкзак, а біжучу вагу зменшуємо на 52: 70-52=18. Наступний предмет – 27 – важчий за 18, тому його не беремо, а поточна вага залишається 18. Аналізуємо наступний предмет, вага якого 13. Це менше за 18, тому цей елемент кладемо в рюкзак, а поточну вагу зменшуємо на вагу обраного предмета: 18–13=5. Вага чергового предмета 6. Це більше за 5, тому цей предмет не обираємо, а поточну вагу залишаємо 5. Наступний предмет: $3 \le 5$. Обираємо цей предмет. Нова поточна вага: 5–3=2. І останній предмет: $2 \le 2$. Обираємо цей предмет. Нова поточна вага: 2–2=0. Отже, задачу розв'язано. За заданої ваги рюкзака він містить такі предмети: {2, 3, 13, 52}.

Якщо змінити розглянутий приклад, додавши до початкового набору предметів один елемент, потрібно буде виконати додатково один крок описаного алгоритму. Тобто маємо лінійний ріст складності алгоритму розв'язання.

Для нормальних, або, інакше, складних рюкзаків не відомо жодного подібного ефективного алгоритму розв'язання. Для таких рюкзаків найшвидшим алгоритмом є перебір можливих комбінацій предметів у рюкзаку. У цьому випадку додавання одного предмета до їх повного переліку вдвічі ускладнює задачу пакування рюкзака.

Алгоритм Меркла–Хелмана грунтується на розглянутих властивостях. ***Закритим ключем*** є послідовність ваг для пакування надзростаючого рюкзака. ***Відкритим ключем*** є послідовність ваг для задачі пакування нормального рюкзака, прицьому обидві задачі мають однаковий розв'язок.

Меркл та Хелман розробили спосіб перетворення задачі пакування надзростаючого рюкзака на еквівалентну задачу пакування нормального рюкзака. Для цього вони застосували арифметику модулярних операцій.

Обчислення відкритого ключа за закритим. Нехай задано закритий ключ – надзростаючу послідовність {2, 3, 6, 13, 27, 52}. Для її перетворення на еквівалентну нормальну послідовність необхідно помножити всі значення послідовності на число $n \bmod m$, причому значення модуля m має бути більшим за суму всіх чисел надзростаючої послідовності. Для прикладу ви-

беремо $m = 105$. Множник n повинен бути взаємно простим числом із модулем. Для прикладу вибираємо $n = 31$. За допомогою алгоритму Евкліда переконуємося, що $НСД(31,105) = 1$. За таких обраних параметрів еквівалентною нормальною послідовністю буде:

$$2 \cdot 31 \bmod 105 = 62,$$
$$3 \cdot 31 \bmod 105 = 93,$$
$$6 \cdot 31 \bmod 105 = 81,$$
$$13 \cdot 31 \bmod 105 = 88,$$
$$27 \cdot 31 \bmod 105 = 102,$$
$$52 \cdot 31 \bmod 105 = 37.$$

Отже, маємо еквівалентну нормальну послідовність {62, 93, 81, 88, 102, 37} – ***відкритий нетаємний ключ зашифрування***.

Зашифрування. Зашифрування здійснюють за допомогою відкритого ключа – нормальної послідовності для пакування рюкзака. Для зашифрування повідомлення спершу розбивають на блоки, які за розміром відповідають кількості елементів послідовності. Вважаючи, що одиниця в блоці вказує на наявність відповідного елемента в рюкзаку, а нуль – на його відсутність, обчислюють повні ваги рюкзаків – по одному для кожного блока повідомлення.

Приклад. Нехай повідомлення в бінарному вигляді є таким:

011000110101101110.

Зашифруємо його за допомогою нормальної послідовності, обчисленої в попередньому пункті: {62,93,81,88,102,37}.

У послідовності є шість елементів, тому розбиваємо повідомлення на блоки по шість бітів у кожному:

011000 | 110101 | 101110.

Зашифровуємо кожен блок окремо за допомогою нормальної послідовності:

$$011000 \Rightarrow 93+81=174,$$
$$110101 \Rightarrow 62+93+88+37=280,$$
$$101110 \Rightarrow 62+81+88+102=333.$$

Отже, шифртекстом буде послідовність чисел: 174, 280, 333.

Розшифрування. Законний отримувач повідомлення знає закритий ключ (надзростаючу послідовність). Знає він також числа m та n, які використовували для її перетворення на еквівалентну нормальну послідовність.

Для розшифрування отриманого повідомлення він спочатку обчислює (за розширеним алгоритмом Евкліда – див. розділ 2 посібника) $n^{-1} \bmod m$, тобто таке число, що: $n \cdot n^{-1} \bmod m \equiv 1 \bmod m$.

Потім кожне число шифртексту множить на $n^{-1} \bmod m$ і розділяє на біти за допомогою таємного ключа – надзростаючої послідовності, тобто розв'язує задачу пакування надзростаючого рюкзака.

Приклад. У прикладі зашифрування надзростаюча послідовність була така: {2, 3, 6, 13, 27, 52}; $m = 105$; $n = 31$; шифртекст: 174, 280, 333.

Обчислюємо (за розширеним алгоритмом Евкліда) $31^{-1} \bmod 105 = 61$. Усі числа шифртексту множимо на 61 за модулем 105. Результати розділяємо на біти, розв'язуючи задачу пакування надзростаючого рюкзака:

$$174 \cdot 61 \bmod 105 = 9 = 6 + 3 \Rightarrow 011000,$$
$$280 \cdot 61 \bmod 105 = 70 = 52 + 13 + 3 + 2 \Rightarrow 110101,$$
$$333 \cdot 61 \bmod 105 = 48 = 27 + 13 + 6 + 2 \Rightarrow 101110,$$

звідки отримуємо повідомлення 011000110101101110.

Отже, ми розшифрували шифртекст і отримали вихідне відкрите бінарне повідомлення.

Слід сказати, що рюкзаки, придатні до практичного застосування, містили не менше за 250 елементів. Довжина кожного елемента надзростаючої послідовності знаходилася в діапазоні між 200 і 400 бітами. Зламувати такі рюкзаки "прямим перебором" було безнадійно. Разом із тим ранцеву криптосистему було зламано. ***Аді Шамір*** (Adi Shamir) та ***Ціппел*** (Zippel) змогли відновити надзростаючу послідовність із нормальної, тобто обчислити таємний ключ розшифрування за публічним ключем зашифрування. Після цього було розроблено інші модифікації ранцевих криптосистем. Деякі з них теж було зламано. Інші залишаються незламаними до сьогодні. Однак загалом довіру до ранцевих криптосистем було підірвано, і на практиці такі криптосистеми сьогодні не застосовують.

3.7.5. Алгоритм RSA

Невдовзі після появи алгоритму рюкзака Меркла було створено перший повноцінний алгоритм із відкритим ключем, який можна застосовувати для шифрування й для цифрового підпису [1]. Цим алгоритмом був алгоритм ***RSA***. Назву системи утворено з перших літер прізвищ її винахідників: ***Рональда Лінна Райвеста*** (Ronald L. Rivest), ***Аді Шаміра*** (Adi Shamir) та ***Леонарда Макса Адлемана*** (Leonard Adleman). Алгоритм RSA протягом багатьох років залишався найпопулярнішим алгоритмом асиметричної криптографії. Алгоритм було запропоновано в 1977 році. Протягом багатьох років алгоритм RSA успішно протистояв спробам криптоаналітичного зламу. За ці роки не було доведено обчислювальної стійкості алгоритму, але також і не було знайдено

ефективних алгоритмів зламу системи. Тобто, що й відповідає загалом концепції стійкості асиметричних криптосистем, алгоритм сьогодні вважають емпірично стійким.

Безпека алгоритму RSA ґрунтується на трудомісткості факторизації (тобто розкладу на прості множники) великих чисел. Відкритий та закритий ключі RSA є великими числами і містять сотні десяткових цифр. Вважають, що відновлення відкритого тексту за шифртекстом і відкритим ключем рівносильне розкладанню числа на два великі прості множники.

Генерування ключів у системі RSA. Для генерування двох ключів системи RSA (відкритого та закритого) застосовують два великі випадкові прості числа, які позначають p та q. Для максимальної безпеки ці числа повинні мати однакову довжину. Потім виконують такі дії:

1. Обчислюють добуток цих двох чисел: $n = p \cdot q$.
2. Для добутку двох простих чисел функція Ейлера дорівнює:
$$\Phi(n) = (p-1)(q-1) = n - p - q + 1.$$

Нагадаємо (див. розділ 2), що значення функції Ейлера вказує на кількість чисел у кільці Z_n, до яких існують обернені. Тобто це є обсяг мультиплікативної групи.

3. Випадково вибирають число e, яке не перевищує значення $\Phi(n)$ та взаємно просте з ним.
4. Потім за розширеним алгоритмом Евкліда (див. Розділ 2) знаходять число d, обернене до e в мультиплікативній групі $Z^*_{\Phi(n)}$. Тобто це таке число, що:
$$d < \Phi(n) \text{ і } ed \equiv 1 \bmod \Phi(n).$$

У результаті отримують:

відкритий ключ зашифрування e, n.

Таємний ключ розшифрування: d.

Зашифрування в системі RSA. Зашифрування здійснюють блоками. Для цього повідомлення записують у цифровій формі й розбивають на блоки так, щоб кожен блок був числом, яке не перевищує n. Наприклад, якщо блок записано у вигляді двійкового слова завдовжки m бітів, необхідне виконання нерівності: $2^m < n$. Кожен блок M розглядають як елемент кільця Z_n і як такий, що можна підносити до степеня за модулем n.

Алгоритм зашифрування в системі RSA полягає в піднесенні блока відкритого тексту M до степеня e:

$$C = E(M) = M^e \bmod n.$$

У результаті отримують блок шифртексту $C = E(M)$, який також є цифровим записом якогось елемента кільця Z_n.

Розшифрування в системі RSA. Алгоритм розшифрування блока шифртексту C полягає в піднесенні C до степеня d:

$$M = D(C) = C^d \bmod n,$$

де d – таємний ключ розшифрування.

Приклад. Нехай при генеруванні ключів RSA обрано: $p = 53$, $q = 67$. Обчислюємо ключі:

$$n = 53 \cdot 67 = 3551;\ \Phi(n) = n - p - q + 1 = 3551 - 53 - 67 + 1 = 3432.$$

Виберемо $e = 1021$ (за допомогою алгоритму Евкліда перевіряємо, що $НСД(3432, 1021) = 1$).

За розширеним алгоритмом Евкліда обчислюємо:

$$d = e^{-1} \bmod 3432 = 1021^{-1} \bmod 3432 = 1237.$$

Отже, ключі обрано.

Відкритий ключ: $e = 1021$ і $n = 3551$ оприлюднюємо.

Таємний ключ: $d = 1237$ зберігаємо в таємниці.

Припустимо, що необхідно зашифрувати алгоритмом RSA слово "продай". Спочатку перетворимо повідомлення на цифрову форму. Отримаємо таке повідомлення в десятковій формі (за умови первинного кодування букв повідомлення їх двоцифровими десятковими номерами в алфавіті):

19 20 18 05 00 13.

Розбиваємо повідомлення на блоки так, щоб кожен блок, прочитаний як число, не перевищував значення $n = 3551$. Отже, це буде 3 блоки по 4 десяткові цифри в кожному.

1920 | 1805 | 0013.

Потім зашифровуємо кожен блок окремо (для обчислення великого степеня від великого числа застосовуємо бінарний алгоритм, описаний нижче):

$$C_1 = 1920^{1021} \bmod 3551 = 2393,$$

$$C_2 = 1805^{1021} \bmod 3551 = 1788,$$

$$C_3 = 0013^{1021} \bmod 3551 = 2188,$$

Отже, шифртекст обчислено:

2393 | 1788 | 2188.

Розшифрувати цей шифртекст можна, піднісши блоки шифртексту до степеня $d = 1237$ за модулем 3551.

Коректність алгоритму RSA. Коректність алгоритму обґрунтовує можливість розшифрування. Коректність RSA доводять так (усі операції виконують за модулем n):

$$(C_i)^d = \left(M_i^e\right)^d = M_i^{ed} = M_i^{k(p-1)(q-1)+1} = M_i \cdot M_i^{k(p-1)(q-1)} = M_i \cdot 1 = M_i .$$

Надійність RSA. Безпека RSA грунтується на складності задачі факторизації. У практичному плані надійність RSA, як і більшості інших асиметричних криптосистем, грунтується на довжині ключа. Адже довжина ключа визначає довжину блока. А якщо довжина блока невелика, криптоаналітик, маючи ключ зашифрування, може зашифрувати всі можливі блоки відкритого тексту й скласти таблицю з відповідними блоками шифртексту. Якщо він зможе обчислити таку таблицю й має достатньо пам'яті для її зберігання, злам будь-якого перехопленого блока шифртексту для нього не проблема. Подібна атака, нагадаємо, називається атакою з вибраним відкритим текстом. Отже, довжина ключа має бути така, щоб криптоаналітик за розумний час не зміг обчислити такої таблиці й не мав достатніх обсягів пам'яті для її зберігання.

Вимоги до довжини ключа постійно зростають разом зі зростанням доступних обчислювальних потужностей, технологій та обсягів пам'яті. Сьогодні для RSA це близько 2000–3000 бітів.

Ефективність. За алгоритмом RSA виконують такі операції:

1. Генерування великих простих чисел.

Для виконання цієї операції використовують декілька ефективних алгоритмів. Працюють вони за такою схемою:

а) генерування великого випадкового числа;

б) перевіряння згенерованого числа на простоту.

2. Генерування випадкового числа, взаємно простого із заданим.
3. Обчислення оберненого числа за заданим модулем.

Операції 2–3 ефективно виконують за допомогою розширеного алгоритму Евкліда.

4. Піднесення великого числа до великого степеня за заданим модулем.

Для виконання операції 4 є алгоритм, значно ефективніший за тривіальне багатократне домноження на основу. Цей алгоритм був відомий в Індії ще до нашої ери. Його називають ***бінарним алгоритмом,*** або ***алгоритмом послідовного піднесення до квадрата***. Часом цей алгоритм називають також ***адитивним ланцюжком.***

Загалом обчислювальна складність алгоритму RSA, як і інших асиметричних криптосистем, є дуже великою і на порядки перевищує обчислювальну складність симетричних криптосистем. А тому роль ефективних обчислю-

вальних алгоритмів постійно зростає. На сучасному етапі розвитку криптології прикладну криптологію, яка вивчає такі алгоритми, виділяють в окремий науковий напрям [9]. У межах цього напряму досліджують, зокрема, бінарний алгоритм піднесення до степеня і множення точок еліптичних кривих, ефективні методи модульної редукції [10], різноманітні спеціалізовані методи швидких обчислень [11] тощо.

3.7.6. Бінарний алгоритм піднесення до степеня

Алгоритм ґрунтується на розкладі показника степеня за цілими степенями двійки. Розглянемо основні ідеї алгоритму на прикладі [1].

Приклад. Нехай необхідно обчислити a^{25}. Тривіальний алгоритм вимагає для цього 24 множення числа a саме на себе: $a \cdot a \cdot a \cdot \cdots \cdot a$.

Бінарний алгоритм полягає в наступному. Подамо показник 25 двійковим числом і розпишемо це число за цілими степенями основи (двійки):

$$25 = (11001)_2 = 1 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 = 1 \cdot 2^0 + 1 \cdot 2^3 + 1 \cdot 2^4.$$

Тепер наша обчислювальна задача набуде такого вигляду:

$$a^{25} = a^{(11001)_2} = a^{\left(1 \cdot 2^0 + 1 \cdot 2^3 + 1 \cdot 2^4\right)} = a^{2^0} \cdot a^{2^3} \cdot a^{2^4} = a \cdot a^8 \cdot a^{16}.$$

Щоб обчислити a^{16}, необхідно виконати 4 множення:

$$\left(\left(\left((a)^2\right)^2\right)^2\right)^2,$$

причому зауважимо, що в процесі такого обчислення ми отримаємо також і a^8. Отже, для обчислення a^{25} залишається виконати 2 множення: $a^{25} = a \cdot a^8 \cdot a^{16}$. Тобто, за бінарним алгоритмом для обчислення a^{25} необхідно виконати лише 6 множень на відміну від тривіального алгоритму, який потребує для цього аж 24 множення. Цей приклад ілюструє ефективність давнього бінарного алгоритму. Причому ефективність цього алгоритму то вища, що більші числа, з якими працюємо.

Загалом за бінарним методом піднесення до степеня спочатку послідовним піднесенням до квадрата обчислюють степені основи, які дорівнюють цілим степеням двійки. Потім, відповідно до двійкового представлення показника степеня, перемножують обрані проміжні результати.

Розглянемо тепер деталізований приклад.

Приклад. Нехай у попередньому прикладі $a = 4$. Показник степеня 25. Складемо таблицю для степенів 4, що дорівнюють цілим степеням 2 (значення

цілого степеня двійки в таблиці відповідають значенням змінної j). При цьому кількість цілих степенів двійки дорівнює двійковому представленню показника степеня. Двійковий запис показника степеня (25=11001) запишемо під таблицею, ***починаючи з молодшого розряду***.

j	0	1	2	3	4
a^{2^j}	4	16	256	65536	4294967296
	1	0	0	1	1

Зауважимо тут, що $a^{2^0} = a^1 = a = 4$ – основа. Наступні степені основи отримуємо послідовним піднесенням до квадрата попереднього результату.

Для обчислення остаточного результату 4^{25} перемножуємо ті a^{2^j}, яким відповідають одиниці двійкового представлення показника степеня:

$$4^{25} = 4 \cdot 65536 \cdot 4294967296 = 1125899906842624.$$

У поданому прикладі розв'язано класичну задачу піднесення до степеня. Якщо ж необхідно виконати піднесення до степеня за заданим модулем, як у криптосистемі RSA, достатньо всі операції алгоритму виконувати за заданим модулем.

Приклад. Обчислити за бінарним алгоритмом $4^{25} \bmod 5$:

j	0	1	2	3	4
$a^{2^j} \bmod 5$	4	1	1	1	1
	1	0	0	1	1

Остаточний результат: $4^{25} \bmod 5 = (((4 \cdot 1) \bmod 5) \cdot 1) \bmod 5 = 4$.

Ефективний бінарний алгоритм піднесення до степеня. Розглянутий раніше табличний приклад піднесення до степеня за бінарним алгоритмом є ілюстративним, проте незручним для програмного здійснення. Зручнішим та економнішим є алгоритм, сформульований нижче.

Нехай потрібно обчислити: $z = f(x) = x^d$.

Подамо показник степеня d двійковим числом: $d = (d_l \cdots d_1 d_0)_2$, де $d_i \in \{0,1\}$.

Отже: $d = \sum_{i=0}^{l} d_i \cdot 2^i$.

Формулювання алгоритму:

1) $z_0 = 1; i = \overline{1, l+1}$;

2) $z_i = z_{i-1}^2$;

3) $z_i = z_i \cdot x$ якщо $d_{l+1-i} = 1$.

Тобто в цьому алгоритмі, на відміну від табличного варіанта, аналіз бітів показника степеня починають від старшого біта. Незалежно від значення проаналізованого біта попередній результат підносять до квадрата. Якщо ж проаналізований біт рівний одиниці, поточний результат домножують на основу.

3.7.7. Криптосистема Рабіна

Безпека ***системи Рабіна*** (Міхаель Ошер Рабін, Michael O. Rabin) грунтується на складності пошуку квадратних коренів за модулем складеного числа. За складністю ця проблема подібна до розкладу на множники [1, 2].

Генерування ключів. Вибирають два великі прості числа p, q. Ці числа (задля гарантування коректного розшифрування) мають бути порівнянні з $3 \bmod 4$. Обчислюють добуток цих чисел: $n = p \cdot q$.

Відкритий ключ: n.

Таємний ключ: p, q.

Зашифрування. Зашифрування здійснюють блоками, подібно до системи RSA. Зашифрування блока відкритого тексту здійснюють за формулою:

$$C = E(M) = M^2 \bmod n.$$

Отже, зашифрування в системі Рабіна є значно простішим порівняно з RSA. Натомість розшифрування є істотно складнішим.

Розшифрування. Отримувач повідомлення знає таємний ключ – числа p та q. Він обчислює:

$$m_1 = C^{(p+1)/4} \bmod p,$$
$$m_2 = \left(p - C^{(p+1)/4}\right) \bmod p,$$
$$m_3 = C^{(q+1)/4} \bmod q,$$
$$m_4 = \left(q - C^{(q+1)/4}\right) \bmod q.$$

Потім обчислює два цілі числа a і b:

$$a = q\left(q^{-1} \bmod p\right);\ b = p\left(p^{-1} \bmod q\right).$$

Чотирма можливими розв'язками є:

$$M_1 = \left(am_1 + bm_3\right) \bmod n,$$
$$M_2 = \left(am_1 + bm_4\right) \bmod n,$$

$$M_3 = (am_2 + bm_3) \bmod n,$$
$$M_4 = (am_2 + bm_4) \bmod n.$$

Один із цих чотирьох розв'язків і є шуканим блоком відкритого тексту M. Якщо відкрите повідомлення є текстом якоюсь мовою, із цих чотирьох варіантів вибирають змістовний текст. Якщо ж повідомленням є числова інформація, то перед зашифруванням необхідно додати до повідомлення якийсь відомий заголовок, за яким ідентифікують правильність одного з обчислених варіантів.

3.7.8. Система Ель-Гамала

Генерування ключів. У системі ***Ель-Гамала*** (ElGamal) вибирають [1, 2] велике просте число p, а також число g: $1 < g < p-1$, яке має в мультиплікативній групі Z_p^* великий порядок. В ідеальному випадку g є первісним коренем за модулем p. Нагадаємо, що порядком числа g в мультиплікативній групі Z_p^* є кількість різних цілих чисел з кільця Z_p, які можна отримати при піднесенні числа g до всіх степенів з кільця Z_p. Число g є первісним коренем за модулем p, якщо при обчисленні всіх степенів цього числа ($g^0 \bmod p, g^1 \bmod p, g^2 \bmod p, \ldots, g^{p-1} \bmod p$) всі результати будуть різні і, відповідно, ми отримаємо кільце цілих чисел Z_p.

Числа p і g не є таємними, – їх опубліковують. Потім кожен користувач вибирає собі випадкове число a в проміжку від 1 до $p-1$ і обчислює:

$$h = g^a \bmod p.$$

Відкритий ключ: p, g, h.

Таємний ключ: a.

Зашифрування. Зашифрування здійснюють блоками. Блок відкритого тексту M перетворюють на шифртекст C так:

1. Вибирають випадкове число r: $1 \le r \le p-1$.
2. Обчислюють $C = (c_1, c_2)$:

$$c_1 = g^r \bmod p,\ c_2 = Mh^r \bmod p.$$

Розшифрування. Маючи таємний ключ a і шифртекст $C = (c_1, c_2)$, обчислюють блок відкритого тексту:

$$M = D(C) = c_2 \left(c_1^a\right)^{-1} \bmod p.$$

3.8. Асиметричні криптосистеми на еліптичних кривих

Багато асиметричних криптосистем грунтуються на складній задачі дискретного логарифмування в обраній скінченій групі (див. розділ 2). У традиційних криптосистемах як скінчену групу використовують мультиплікативну групу скінченого поля. Для такої групи на сьогоднішній день відомі алгоритми розв'язку задачі дискретного логарифмування з субекспоненційною складністю. На практиці це означає, що основні параметри таких криптосистем мають бути дуже великими числами. Обчислювальні операції над такими числами збільшують обчислювальну складність традиційних асиметричних криптосистем. Але задачу дискретного логарифмування формулюють для будь-якої скінченої групи, зокрема й для групи точок ***еліптичної кривої*** (ЕК). Причому для таких криптосистем сьогодні не відомо субекспоненційних алгоритмів розкриття. А це означає, що за однакової з традиційними криптосистемами стійкості, розрядність чисел, якими оперують в криптосистемах на еліптичних кривих, є істотно меншою. Для прикладу, система на ЕК із довжиною ключа 160 бітів має приблизно таку саму стійкість, як і традиційна система із ключем у 1024 біти. А 320-бітному ключу в системі на ЕК відповідає ключ довжиною 5600 бітів у традиційній системі.

Уперше криптографічні алгоритми на групах точок еліптичних кривих запропонували незалежно один від одного ***Ніл Коблiц*** (Neal Koblitz) та ***Віктор Міллер*** (Viktor S. Miller) у 1985 році. Спочатку ці алгоритми сприйняли з недовірою, але із часом емпірично було обґрунтовано стійкість систем на ЕК та можливість їх ефективного здійснення, що й зумовило бурхливий розвиток цього напрямку криптології сьогодні.

Криптографічні алгоритми, що грунтуються на задачі дискретного логарифмування, майже дослівно переформульовують на групи точок на ЕК із заміною операції множення на додавання й операції піднесення до степеня – на скалярне множення. Зауважимо, що криптосистеми, які використовують інші складні задачі, наприклад, задачу факторизації великих чисел (система RSA), не переносяться на ЕК ефективно (наприклад, з виграшем у довжині ключа).

У розділі 2 було розглянуто теоретичні основи криптосистем на еліптичних кривих, способи виконання базових операцій таких криптосистем, а також особливості основного обчислювального алгоритму криптосистем на еліптичних кривих – бінарного алгоритму множення точнок ЕК – аналогу бінарного алгоритму піднесення до степеня в традиційних асиметричних криптосистемах.

Розглянемо деякі приклади систем на ЕК.

Схема відкритого розподілу ключів Діффі–Хеллмана. Нехай абоненти А і В хочуть створити спільний секретний ключ (наприклад, для шифрування симетричною системою), користуючись лише відкритим каналом. Вони обирають скінченне поле F_q (розширене поле Галуа, де $q = p^r$ – велике число, p – просте число – див. розділ 2) та ЕК, визначену над цим полем, а також базову точку P на цій кривій. Базова точка відіграє роль твірного елемента g мультиплікативної групи скінченного поля в класичному варіанті ***схеми Діффі–Хеллмана*** (Вітфілд Діффі, Bailey Whitfield "Whit" Diffie – Мартін Хеллман, Martin E. Hellman). Але цього разу ми не вимагаємо, щоб P породжувала групу точок ЕК, адже остання може взагалі не бути циклічною. Необхідно лише, щоб порядок P був великим: збігався з порядком N ЕК або був великим простим дільником N.

Поле F_q, обрана еліптична крива над цим полем E/F_q та базова точка на кривій P – відомі параметри системи.

Абоненти А і В вибирають великі випадкові числа k_A і k_B відповідно, які вони тримають у секреті, обчислюють $k_A P$ (відповідно $k_B P$) і обмінюються цими значеннями за допомогою відкритого каналу. Абонент А, отримавши $k_B P$, обчислює $k_A k_B P$; так само абонент В обчислює $k_B k_A P$. У результаті вони отримують спільний секретний ключ $Q = k_A k_B P$. Щоправда, ключ має бути числом, а не точкою ЕК. Але за точкою ЕК над скінченним полем F_q легко можна отримати число, наприклад, так. Нехай $q = p^r$. Елемент α поля F_q можна представити як многочлен степеня, не вищого за $r-1$ над F_p:

$$\alpha = \sum_{i=0}^{r-1} a_i x^i, \quad a_i \in F_p .$$

Поставимо у відповідність α число у системі числення з основою p:

$$\alpha \sim \sum_{i=0}^{r-1} a_i p^i .$$

Взявши x-координату точки $Q = (x, y)$, $x \in F_q$, поставимо у відповідність їй число (як показано вище) і отримаємо секретний ключ k.

Суперник, перехопивши $k_A P$ та $k_B P$ – інформацію, що передають відкритим каналом – не в змозі знайти k, оскільки не може обчислити k_A або k_B внаслідок складності розв'язання аналога задачі дискретного логарифмування в групі точок ЕК.

Система шифрування Мессі–Омури. Відкриті параметри ***системи шифрування Мессі – Омури*** (Джеймс Мессі, James Lee Massey – Джим К. Омура, Jim K. Omura): розширене поле Галуа F_q ($q = p^r$ – велике число), еліптична крива E/F_q, N – порядок еліптичної кривої.

Нехай абонент А хоче передати абоненту В секретне повідомлення M за допомогою відкритого каналу. Вважаємо, що числу M відповідає точка ЕК P_M так, наприклад, як це вказано вище.

Абонент А обирає випадкове велике число k_A таке, що $(k_A, N) = 1$ (тобто існує $k_A^{-1} \bmod N$), обчислює $k_A P_M$ і відсилає це значення абоненту В. Абонент В вибирає випадкове велике число k_B, $(k_B, N) = 1$, обчислює $k_B k_B P_M$ і відсилає назад абоненту А. Абонент А знаходить точку $k_A^{-1} k_B k_A P_M = k_B P_M$, відсилає її абоненту В, який обчислює $k_B^{-1} k_B P_M = P_M$ і так отримує розшифроване повідомлення M.

Стійкість системи грунтується на складності розв'язання задачі дискретного логарифмування в групі точок ЕК.

Система шифрування Ель-Гамала. Відкриті загальні параметри системи: F_q, E/F_q, базова точка $P \in E/F_q$ великого порядку. Знати порядок ЕК у цьому випадку непотрібно.

Кожен абонент, крім того, вибирає випадкове ціле число a, яке тримає в секреті (закритий ключ), а точку aP оголошує як свій відкритий ключ.

Щоб зашифрувати й відіслати абонентові В таємне повідомлення P_M, абонент А вибирає випадкове число k і обчислює пару $(c_1, c_2) = (kP, P_M + k(a_B \cdot P))$, де $a_B \cdot P$ – відкритий ключ абонента В. Ця пара і є шифрованим повідомленням, яке абонент А насилає абоненту В. Для того щоб прочитати повідомлення, абонент В обчислює $c_2 - a_B c_1 = P_M + k(a_B \cdot P) - a_B(kP) = P_M$.

Отже, абонент А разом із зашифрованим повідомленням посилає "ключ" kP для його розшифрування, яким, однак, може скористатися лише абонент В, який знає свій секретний ключ a_B.

Подібним чином на групи точок на ЕК переносять і алгоритми цифрового підпису Ель–Гамала та Шнорра.

Загалом, недовіра до стійкості асиметричних криптосистем на еліптичних кривих ще зберігається, оскільки надто коротким є період емпіричного аналізу їх криптографічної стійкості. Разом з тим, очевидна перевага таких крипто-

систем над традиційними в обчислювальній складності призвела до їх активного впровадження в багатьох країнах, зокрема й в Україні.

3.9. Альтернативна криптографія

Шифрування даних, а також безпеку даних при передаванні можна забезпечити, використовуючи нові підходи в інших галузях науки. У багатьох випадках результати проведених досліджень є дуже перспективними й уже знаходять практичне застосування.

Квантова криптографія. Теоретичні засади квантової криптографії [10, 11] розробив ***Стефан Вейснер*** (Stephen J. Wiesner), фізик з університету Колумбії. Він запропонував використовувати поляризацію фотонів для створення банкнот, які не можна підробити. У його моделі кожна банкнота мала бути оснащена певною кількістю (що більше, то більша безпека) елементів, які містять фотони певної поляризації. Банк мав би перелік банкнот, кожній банкноті було б надано серійний номер, а також параметри поляризації кожного окремого фотона. Тест на автентичність полягав би в перевірянні відповідності поляризації фотонів пред'явленої банкноти з відомими для банкноти з цим серійним номером параметрами поляризації. Таке перевіряння можна виконати за допомогою спеціального фільтра поляризації (поляризатора), установленого під відповідним кутом. Такий фільтр пропускає тільки фотони з відповідною поляризацією. Якщо поляризація фотонів має певний кут відносно поляризатора, можна говорити про пропускання тільки з певною ймовірністю.

Створена Вейснером модель передбачала комфортну ситуацію для банку. Банк знав поляризацію окремих фотонів, міг відповідно підбирати позицію поляризатора. Під час проби кожен фотон мав пройти через поляризатор. Якщо якийсь із них не пройшов через поляризатор, це означало би, що банкнота фальшива. Такі банкноти було б практично неможливо підробити.

Пізніше ідею Вейснера адаптували Беннет і Брассард до проблеми безпечного передавання інформації. У 1984 році ***Чарльз Беннет*** (Charles H. Bennett) з фірми IBM і ***Жиль Брассард*** (Gilles Brassard) з Університету Монреаля запропонували просту схему захищеного квантового розподілу ключів шифрування. Дані пересилають як послідовність фотонів із різними поляризаціями, які відповідають нулям або одиницям. Їх зчитують набором поляризаторів. Послідовність фотонів відіграє роль ключа. Кожна несанкціонована спроба зчитування інформації призводить до зміни поляризації як мінімум деяких фотонів (відповідно до принципу невизначеності Гейзенберга, згідно з яким спроба провести виміри в квантовій системі спотворює

її стан, і отримана в результаті таких вимірювань інформація не повністю відповідає стану до початку вимірювань). Отже, спроба перехоплення інформації із квантового каналу зв'язку неминуче призводить до внесення завад, що легко виявляє отримувач повідомлення. Для передавання ключа Беннет і Брассард запропонували такий протокол:

1) Користувач А пересилає користувачу В послідовність фотонів із різною поляризацією.

2) Користувач В виконує вимірювання поляризації, вибираючи установки фільтра на свій розсуд. У зв'язку з тим деякі виміри будуть помилковими. Далі користувач В інформує користувача А, які установки він вибрав для окремих фотонів.

3) Користувач А передає користувачу В номери правильно розкодованих фотонів, які становлять безпечно переданий потік даних, і які можна використовувати для подальшого шифрування.

Квантова криптографія зробила можливим пересилання інформації цілком безпечним каналом. Кожна спроба несанкціонованого вимірювання поляризації фотонів призводить до зміни поляризації. Тому спробу несанкціонованого доступу завжди можна виявити.

Протокол, який запропонували Беннет і Брассард, було практично впроваджено. У 1999 році цим способом удалося переслати дані на відстань 48 км. У червні 2017 року супутник "Мо-цзи", призначений для квантового передавання інформації каналом зв'язку, гарантовано захищеним від зловмисників, зміг транслювати фотони на станції, розташовані в китайських містах Делінха та Ліцзян, фізична відстань між якими 1203 км. Успіхи в цій галузі наближають настання періоду ідеальної безпеки.

Квантовий комп'ютер. Ідея конструювання комп'ютера, принцип дії якого грунтувався на квантовій фізиці, виникла у 1980 році. Спочатку це була тільки теоретична модель. Усе змінилося після того, як було опрацьовано алгоритм, що давав змогу миттєво розкладати великі числа на прості множники з використанням такого комп'ютера. Не заглиблюючись у деталі такого алгоритму, достатньо сказати, що його застосування зробило б можливим злом шифру RSA і, відповідно, поставило б під загрозу безпеку всіх асиметричних шифрів.

Основним поняттям, пов'язаним із квантовим комп'ютером, є ***кубіт*** (qubit). Це є квантовий відповідник біта в звичайних комп'ютерах. Для представлення кубіта можна використати елементарні частинки із двома різними спінами (спін – квантова характеристика, власний момент імпульсу квантової частинки). Один спін відповідав би логічному нулю, а другий – одиниці. Серія

таких частинок дала б можливість записувати потік нулів та одиниць, а, отже, програмувати квантовий комп'ютер. Спіни частинок можна змінювати за допомогою сильних енергетичних імпульсів, модифікуючи збережену в комп'ютері інформацію.

Суперпозицію спінів можна отримати, надаючи частинкам менший імпульс енергії. У цьому випадку їхні спіни можуть або змінитися, або ні. Інакше кажучи, існує певна ймовірність, що відбудеться зміна спінів. У цій ситуації частинки знаходяться в обох станах одночасно, їхні спіни набувають одночасно значень нуля й одиниці (реально кубіт може набувати нескінченної кількості станів залежно від розподілу ймовірності). Тільки виконання вимірювання дає можливість установити дійсний стан частинок.

Розглянемо квантовий комп'ютер, який складається з 200 кубітів. На початку кубіти мають визначені спіни. Це означає, що вміст пам'яті комп'ютера є повністю відомим. Застосування невеликого імпульсу може змінити їхні спіни з певною ймовірністю. У такий спосіб досягають суперпозиції двохсот кубітів, а, отже, 2^{200} можливих станів одночасно. Обчислювальна потужність, якої можна досягти цим методом, є дуже великою. Квантовий комп'ютер можна використати для практично миттєвого знаходження ключа симетричних та асиметричних шифрів, а також розв'язання інших задач, які вимагають великих обчислювальних потужностей. Наприклад, завдяки алгоритму, який розробив у 1996 році ***Лов Кумар Гровер*** (Lov Kumar Grover) став можливим перебір і пошук у базах даних зі швидкістю, яка в 500 тисяч разів більша, ніж у звичайного комп'ютера.

Перешкод на шляху до конструювання квантового комп'ютера є багато. У 2000 році вдалося сконструювати перший примітивний квантовий комп'ютер. Він містив 5 кубітів і був оснований на дії електромагнітних імпульсів, які одночасно використовували для читання вхідних даних. На цьому комп'ютері здійснено короткі алгоритми. Роком пізніше групі вчених з IBM та Стенфордського університету вдалося сконструювати комп'ютер, який містив 7 кубітів, до того ж виконував ***алгоритм Шора*** (Пітер Шор, Piter Shor), з використанням якого було здійснено розклад числа 15 на множники 5 і 3. При цьому кубітами були спіни ядер атомів відокремленої складної органічної молекули $(^{19}\mathbf{F})_2\text{-}C=^{13}\mathbf{C}(^{19}\mathbf{F})\text{-}^{13}\mathbf{C}[Fe(CO)_2(C_5H_5)]=C(^{19}\mathbf{F})_2$. (рис. 3.41).

У листопаді 2017 року компанія IBM оголосила про розроблення прототипу квантового комп'ютера на 50 кубітів. Планують, що комп'ютер буде доступний у хмарі. Компанія вже роздає в хмарі в межах проекту IBM Q обчислювальні потужності квантових комп'ютерів на 17 кубітів.

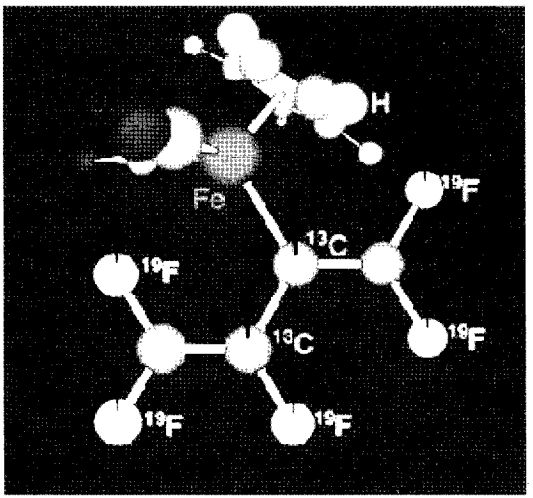


Рис. 3.41. Органічна семикубітна молекула

Криптографія ДНК. У 1962 році ***Френсіс Крік*** (Francis Harry Compton Crick), ***Джеймс Дьюї Ватсон*** (James Dewey Watson) і ***Моріс Хью Фредерік Вілкінс*** (Maurice Hugh Frederick Wilkins) отримали Нобелівську премію за відкриття основ будови ***ДНК*** (дезоксирибонуклеїнова кислота). З того часу інтенсивно виконують дослідження структур ДНК, зокрема ДНК людини, завдяки чому знання основ життя живих організмів значно поглибилося. Також виявилось, що ДНК можна використовувати для передавання зашифрованої інформації [12, 13].

ДНК разом із ***РНК*** (рибонуклеїнова кислота) створює основу для кодування генетичної інформації живих організмів. Молекула ДНК складається із двох спірально скручених полінуклеїдних ланцюгів, які утворюють подвійну спіраль (рис. 3.42). Ланцюги сполучені між собою водневими зв'язками.

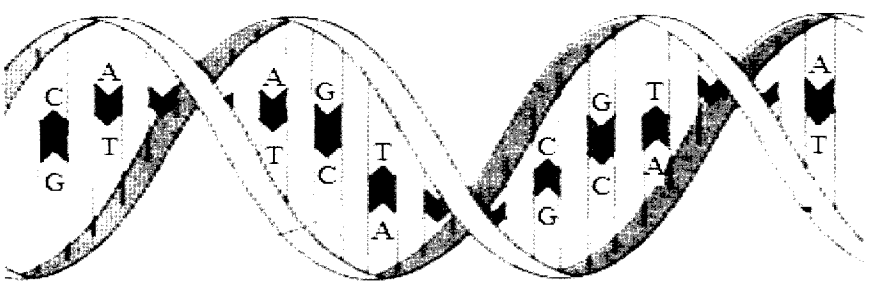


Рис. 3.42. Просторова модель ДНК

Кожен ланцюг складається з нуклеотидів, які є основними частинками ДНК. Нуклеотиди мають подібну будову, відрізняються тільки залишками азотистих основ, які мають назву: аденін (A), гуанін (G), цитозин (C), а також тимін (T). Нуклеотиди, які містять ці основи, позначають відповідно як A, G, C та T. Дрібніші послідовності нуклеотидів утворюють гени, які є носіями спадковості. Поєднання генів батьків визначає риси їхніх нащадків.

Зв'язки між нитками спіралі утворені виключено між нуклеотидами A та T (два водневі зв'язки), а також G та C (три водневі зв'язки). Зв'язки виникають між нуклеотидами, які ніби доповнюють один одного за розмірами – це правило називають комплементарністю. На цьому правилі грунтується відновлення ДНК у клітинах організму. Якщо є одна з ниток ДНК, можна за правилом комплементарності добудувати другу.

Молекулярна криптографія. У 1994 році ***Леонард Адлеман***, один з авторів криптографічного алгоритму RSA, запропонував використовувати молекули ДНК для розв'язання математичної проблеми, відомої як задача комівояжера. У 2001 році сконструйовано перший комп'ютер із використанням ДНК. Теоретичні моделі таких комп'ютерів з'явилися значно раніше, але тоді не було технічних можливостей для їх конструювання. Згадані комп'ютери використовували ДНК не лише для програмування та створення входу–виходу, але й як джерела живлення. Завдяки цьому такі комп'ютери є дуже енергоощадними, мають відносно велику обчислювальну потужність, а додатковою їх перевагою є невеликі розміри. Крім того, комп'ютери на ДНК мають величезну місткість пам'яті. Тільки 1 кубічний міліметр розчину ДНК здатний вмістити 10 петабайтів (1 петабайт дорівнює 1024 терабайт) інформації.

Зчитування даних здійснюють за спеціальною методикою (1997 рік), яка дає можливість виокремлення фрагмента молекули ДНК і зчитування послідовності, яку створюють нуклеотиди. Цю послідовність можна потім використовувати для запису й зчитування інформації.

Беручи до уваги, що в комп'ютерах усі дані записують як послідовності нулів та одиниць, нуклеотиди G і C трактують як нуль, а A і T – як одиницю. Можна, зрозуміло, і навпаки.

Інший спосіб – це створення спеціального алфавіту шифрування, у якому комбінації нуклеотидів використовують для записування літер, цифр і спеціальних символів. Кількість символів, які закодовано в такий спосіб, залежить від кількості нуклеотидів, використаних в окремій комбінації. У випадку подвійних комбінацій (AT, AC, AG, CT тощо) можливо закодувати 16 символів. Використання потрійних комбінацій дає можливість закодувати 64 різні символи. Для четвірок нуклеотидів є 256 різних конфігурацій, що дає можли-

вість закодувати всі літери, цифри й спеціальні символи. Установлений алфавіт має бути відомим обом сторонам зв'язку перед початком обміну зашифрованою інформацією.

Крім зашифрування даних, ДНК можна також використати для здійснення стеганографічних методів. В обох випадках застосовують ***ланцюгову реакцію полімеризації*** (polymerases chain reaction – PCR). Вона надає можливість збільшення кількості однакових ланцюгів нуклеотидів. Ця реакція складається із трьох етапів:

– розрив водневих зв'язків між комплементарними азотистими основами (денатурація); при цьому двоспіральна ДНК розпадається на окремі ланцюги;

– під'єднання короткого ланцюга ДНК до закінчень (праймер або затравка) кожного з ланцюгів;

– приєднання комплементарного ланцюга до кожного відокремленого; на цьому етапі використовують спеціальний ензим – полімеразу ДНК.

Кожна ланцюгова реакція полімеризації подвоює кількість ДНК у розчині. Це робить можливим застосування праймера як ключа шифрування, або параметра, який дає можливість виявити таємну інформацію серед інших ланцюгів ДНК.

Починають з установлення алфавіту, яким користуватимуться під час запису даних. Потім за допомогою цього алфавіту записують дані, а також ключ, який буде використано для їх шифрування. Тобто, виникають дві окремі послідовності нуклеотидів. Ключ має бути відповідно довгим, щоб зробити неможливим його підбір методом брутальної атаки. Мінімальна рекомендована довжина становить 10 знаків, а, отже, 40 нуклеотидів для алфавіту, який ґрунтується на четвертних комбінаціях нуклеотидів. Довжина ключа є важливою також під час проведення ланцюгової реакції полімеризації в процесі розшифрування.

Шифрують повідомлення так:

– утворюють два ланцюги нуклеотидів – повідомлення й ключ;

– до ланцюга повідомлення долучають з обох сторін послідовність нуклеотидів, які становлять гасло;

– утворену послідовність гасло–повідомлення–гасло піддають реакції полімеризації з використанням ключа як праймера, так отримують молекули ДНК, які складаються із двох ланцюгів;

– приховують повідомлення серед великої кількості інших ланцюгів ДНК.

Отримувач повідомлення використовує ключ як праймер для проведення ланцюгової реакції полімеризації. Цю реакцію потрібно повторювати багаторазово, і кожного разу буде подвоюватись кількість ланцюгів ДНК, які містять приховані дані. Після виділення відповідної частини послідовності нуклеотидів,

що утворює таємне повідомлення, його можна читати за допомогою операції ***секвенціювання*** (операції розділення на послідовності). Потім відновлюють відкритий текст, користуючись раніше встановленим алфавітом шифрування.

Шифрування з використанням молекул ДНК дає можливість проведення тільки основних операцій шифрування, таких як метод підстановки, або XOR. Було також досліджено виконання операцій асиметричного шифрування з використанням ДНК.

У березні 2017 року з'явилось повідомлення, що вченим із Колумбійського університету й Нью-Йоркського центру з вивчення генома вже вдалося зберегти на ДНК операційну систему, фільм та інші файли. Однією з перешкод для широкого впровадження цих технологій є їхня вартість. Дослідники витратили $7000, щоб синтезувати ДНК, яку вони використовували для архівування 2 Мбайт даних, і ще $2000, щоби потім прочитати цю інформацію.

Проведені дослідження доводять, що в ДНК прихований величезний потенціал для застосування в галузі інформаційної безпеки. Молекулярні комп'ютери можна використовувати в криптоаналізі. Крім описаних вище криптографічних і стеганографічних методів, можна також здійснювати біометричні техніки. Вони дають можливість ідентифікувати особу на підставі проби ДНК.

ДНК можна використовувати як унікальне значення, яке дає можливість ідентифікувати предмети. Це дасть змогу в майбутньому замінити голограми біохімічними мітками, які значно складніше підробити. Отже, молекулярна інформатика може в майбутньому стати невід'ємною частиною систем захисту інформації.

Візуальна криптографія. Концепція візуальної криптографії [14] з'явилась на конференції Eurocrypt 94. Під час своєї вступної лекції ***Аді Шамір*** і ***Моні Наор*** (Moni Naor) запропонували новий спосіб графічного кодування інформації. У його основу було використання обмеженої кількості підзображень. Накладання підзображень одне на одного давало зображення, яке містило таємне повідомлення.

Кожне цифрове зображення складається з ***пікселів*** (pixels). Це найменші графічні елементи, заповнені одним кольором. При розташуванні один біля одного на площині вони утворюють зображення, яке ми бачимо на екрані комп'ютера. Концепція візуальної криптографії полягає в поділі пікселів на ще менші частинки – субпікселі. Пікселі оригінального зображення, яке містить таємне повідомлення, поділяють на субпікселі. Їх кількість залежить від кількості підзображень. На рис. 3.43 наведено найпростішу можливу схему поділу на субпікселі для чорно-білого зображення.

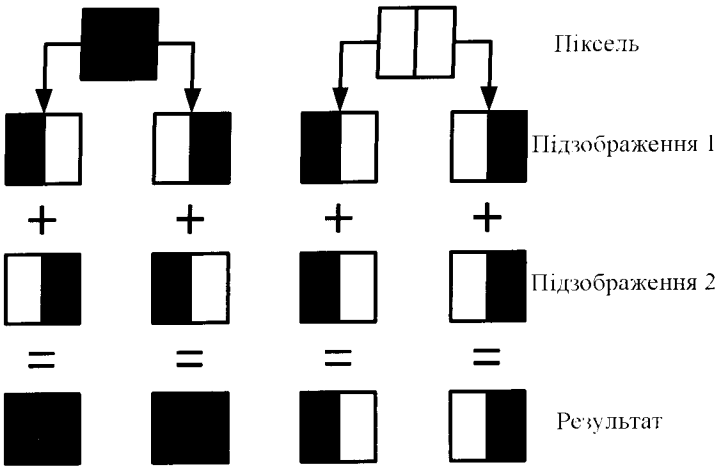


Рис. 3.43. Субпікселі для чорно-білого зображення та їх накладання

У цьому прикладі кожен піксель складається із двох субпікселів. У результаті накладання субпікселів білого пікселя утворюють зображення сірого кольору. Субпікселі чорного пікселя утворюють після накладання зображення чорного кольору. Підзображення, що складені із субпікселів, не дають жодної інформації про зашифроване повідомлення, оскільки кожна комбінація субпікселів з'являється з імовірністю 50 %, що подібно до випадкового розподілу. Коли обидва підзображення будуть накладені одне на одне, з'явиться первинне зображення з тією різницею, що його білі області будуть тепер виглядати як сірі. Можна, вочевидь, поділяти пікселі на більшу кількість субпікселів, утворюючи тим самим більшу кількість підзображень.

Описану техніку використовують для здійснення ***порогових схем***. Це є спосіб розподілу секрету, який полягає в поділі таємного повідомлення на n підзображень. Візуалізація повідомлення стає можливою в разі накладання всіх його частин. Таку схему описують за допомогою матриць. Кожен рядок матриці представляє одне підзображення й складається із m субпікселів. Наприклад, матриці, що представляють порогову схему із двома підзображеннями й складаються із двох субпікселів, мають вигляд:

$$M_0 = \left\{ \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \right\},$$

$$M_1 = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right\}.$$

У цьому записі одиниця означає чорний колір, нуль – білий. Для зашифрування білого пікселя таємної інформації вибирають довільну матрицю з набору M_0, а для чорного пікселя – довільну матрицю з набору M_1. Поділяють рядки підзображень між двома користувачами, після чого знищують первинний образ. Тепер його відновлення буде можливим тільки за згодою обох користувачів.

Кодування зображення з використанням двох субпікселів спричиняє значне його спотворення. Тому застосовують порогові схеми, які основані на чотирьох субпікселях. У такому випадку набір матриць подають так:

$$M_0 = \left\{ \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \right\},$$

$$M_1 = \left\{ \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \right\}.$$

За допомогою побудованих у такий спосіб матриць можна здійснити довільні порогові схеми типу (n, r), де n означає загальну кількість підзображень, а r – кількість підзображень, необхідних для розшифрування повідомлення.

Можна шифрувати зображення за шкалою інтенсивності кольору. Піксель такого зображення може набувати значень від 0 до 255, 0 означає відсутність кольору, а 255 – максимальну інтенсивність. Утворення матриці підзображень розпочинають від поділу кожного пікселя на 256 субпікселів, їхні взаємні пропорції залежать від рівня інтенсивності цього пікселя. Якщо прийняти, що цей рівень становить p, то в матриці підзображень мали би знаходитись p субпікселів без кольору й (256–p) субпікселів із максимальною інтенсивністю кольору. Основною проблемою тут є розмір утворених матриць підзображень. Для одного пікселя необхідна матриця 16x16, що призводить до значного зростання обсягу пам'яті для зашифрованих зображень.

Візуальна криптографія є ідеальним інструментом здійснення протоколів поділу секрету, а спосіб утворення підзображень на основі ділення пікселів гарантує їх повністю випадковий характер.

3.10. Елементи криптоаналізу

Криптоаналіз – це наука про методи розкриття (злому) шифрів, тобто про методи розкриття змісту повідомлень, змінених методами криптографії. Криптографія та криптоаналіз взаємопов'язані, оскільки часто досягнення криптоаналізу є єдиним обґрунтуванням надійності шифру.

Криптоаналіз, як і криптографія, є складовою криптології. Якщо зміст криптографії полягає в збереженні відкритого тексту в секреті, то зміст криптоаналізу – в отриманні відкритого тексту без знання ключової інформації. Інакше кажучи, мета криптоаналізу – дослідження можливості розшифрування інформації без знання ключів. Успішно проведеним криптоаналізом можна розкрити відкритий текст або ключ. Також можна знайти слабкі місця в криптосистемах, що в результаті може призвести до розкриття зашифрованого тексту.

Основне положення криптографії, яке сформулював уперше ***Огюст Керкгоффс*** (Auguste Kerckhoffs) у 1883 році, полягає в припущенні, що безпека (неможливість розкриття) зашифрованого тексту повністю визначається ключем. Керкгоффс припускав, що в криптоаналітика є алгоритм шифрування. Це є справедливим для сучасної криптографії. Усі сучасні алгоритми розробляють за принципом Керкгоффса, тобто знання алгоритму шифрування не повинно давати криптоаналітику якихось значних переваг для розкриття відкритого тексту або ключової інформації.

У теоретичному криптоаналізі злом шифру не обов'язково передбачає виявлення практичного способу для відновлення відкритого тексту за перехопленим зашифрованим повідомленням. Шифр вважають зламаним, якщо в системі шифрування виявлено слабке місце, яке можна використати для ефективнішого злому, ніж ***метод повного перебору ключів*** (brute-force approach). Такі способи можуть вимагати нереально великих обсягів підібраного відкритого тексту або пам'яті комп'ютера. Під зломом розуміють лише підтвердження наявності слабких місць шифру, тобто що властивості надійності шифру не відповідають оголошеним характеристикам. Як правило, криптоаналіз починають зі спроб злому спрощеної модифікації алгоритму, після чого результати поширюють на повноцінну версію.

3.10.1. Типи розкриття

Розглядають ***4 типи розкриття*** залежно від того, яка інформація є в криптоаналітика. Для кожного з них припускають, що криптоаналітик має доступ до алгоритму шифрування.

1. Розкриття з використанням лише шифртексту. У криптоаналітика є шифртексти декількох повідомлень, зашифрованих одним і тим самим алгоритмом шифрування. У цьому випадку задача криптоаналітика полягає в розкритті відкритого тексту якомога більшої кількості повідомлень, або, що ще краще, в отриманні ключа (ключів), що застосовували для зашифрування повідомлень, з метою дешифрування інших повідомлень, зашифрованих тими

самими ключами. У цьому випадку не обов'язково розглядати якісь зовсім окремі зашифровані тексти у вигляді окремих файлів. Більшість сучасних шифрів є блоковими, тобто вони обробляють текст блоками, і в такому випадку відкритий текст M можна представити у вигляді конкатенації (об'єднання) блоків:

$$M = M_1 \| M_2 \| ... M_i \| ... M_n.$$

Для DES, Triple DES це 64 біти, для AES – 128, 192 або 256 бітів. Отже, процес зашифрування й дешифрування можна подати формулами

$$C_i = E_k(M_i),$$

де C_i – блок зашифрованого тексту; k – ключ; E_k – алгоритм зашифрування; M_i – блок відкритого тексту;

$$M_i = \tilde{D}_k(C_i),$$

де $\tilde{D}_k$ – алгоритм дешифрування.

Для задач криптоаналізу кількість блоків можна вважати за кількість текстів, і у випадку тексту розміром 1 Мбайт, зашифрованого шифром DES, маємо 131072 зашифрованих текстів для аналізу. Тобто, для розкриття з використанням лише шифртексту маємо такі вхідні дані:

$$C_1 = E_k(M_1),\ C_2 = E_k(M_2),...,C_i = E_k(M_i).$$

При цьому необхідно отримати: $M_1, M_2, ..., M_i, k$ або алгоритм отримання C_i з $C_i = E_k(M_i)$.

2. Розкриття з використанням відкритого тексту. У криптоаналітика може бути доступ не лише до певної кількості шифртекстів, але й до відкритих текстів, що їм відповідають. Це припущення не є настільки неможливим, що його не можна було б розглядати. Пари відкритий текст – шифртекст може створювати певна службова інформація, присутня в сеансі зв'язку. Тому задачею цього розкриття є отримання ключа за відомих пар значень відкритий текст – зашифрований текст.

Дано: $M_1, C_1 = E_k(M_1), M_2, C_2 = E_k(M_2), ..., M_i, C_i = E_k(M_i)$.

Необхідно отримати: k або алгоритм отримання C_i із $C_i = E_k(M_i)$.

3. Розкриття з використанням вибраного відкритого тексту. У криптоаналітика є не лише доступ до шифртекстів та їх відкритих текстів, але також є можливість вибирати відкритий текст для зашифрування. Це дає більше варіантів, ніж розкриття з використанням відкритого тексту, оскільки аналітик може вибирати блоки відкритого тексту для зашифрування, що, своєю чергою, може дати більше інформації про ключ. Задача полягає в отриманні ключа або

алгоритму, що дасть можливість дешифрувати повідомлення, зашифровані тим самим ключем.

Дано: $M_1, C_1 = E_k(M_1), M_2, C_2 = E_k(M_2), \ldots, M_i, C_i = E_k(M_i)$.

Необхідно отримати: k або алгоритм отримання C_i із $C_i = E_k(M_i)$.

4. Адаптивне розкриття з використанням відкритого тексту. Це частковий випадок розкриття з використанням вибраного відкритого тексту. Криптоаналітик може не лише вибирати відкритий текст для зашифрування, але й грунтувати свій подальший вибір на результатах шифрування попереднього тексту. Під час розкриття з використанням вибраного відкритого тексту криптоаналітик міг вибрати для зашифрування лише один великий блок відкритого тексту, під час адаптивного розкриття з використанням вибраного відкритого тексту він може вибрати менший блок відкритого тексту, після чого вибрати наступний блок, використовуючи результати першого вибору і так далі.

Також згідно зі Шнайером існують принаймі ще три типи розкриття.

5. Розкриття з використанням вибраного шифртексту.

Криптоаналітик може вибрати різні шифртексти для дешифрування й має доступ до розшифрованих відкритих текстів. Тобто це виглядає так, що криптоаналітик має доступ до "чорної скриньки", яка виконує автоматичне розшифрування. Задача криптоаналітика полягає в отриманні ключа.

Дано: $M_1, C_1 = E_k(M_1), M_2, C_2 = E_k(M_2), \ldots, M_i, C_i = E_k(M_i)$.

Необхідно отримати: k.

Такий тип розкриття зазвичай використовують для алгоритмів із відкритим ключем.

6. Розкриття з використанням вибраного криптоключа. Цей тип розкриття ґрунтується на знанні деяких взаємозалежностей між ключами. Тобто криптоаналітик знає про певний взаємозв'язок між ключами, який він використовує для отримання самих ключів. Цей тип розкриття має, переважно, теоретичне значення. Здійснити на практиці його фактично не можливо.

7. Бандитський криптоаналіз. Соціальна інженерія. Його практично вважають найдієвішим. Він полягає в певному впливі на суб'єкти, що є носіями ключа, з метою отримати його значення.

3.10.2. Криптоаналіз класичних алгоритмів

Класичні методи аналізу зашифрованого тексту ґрунтуються на певних статистичних властивостях кожної мови.

Хоча поняття "криптоаналізу" використовують від недавна, деякі методи злому було винайдено десятки століть тому. Першою відомою писемною

згадкою про криптоаналіз є "Манускрипт про розшифрування криптографічних повідомлень", який написав арабський вчений ***Аль-Кінді*** (Абу Юсуф Якуб ібн Ісхак ібн Саббах аль-Кінді, уживане скорочення імені – Аль-Кінді, Abu Yusuf Yaqub ibn Ishaq al-Kindi) ще у IX столітті. У цій науковій праці міститься опис методу частотного аналізу.

Частотний метод. Частотний аналіз – основний інструмент для злому більшості класичних шифрів перестановки чи заміни. Цей метод ґрунтується на припущенні про існування нетривіального статистичного розподілу символів, а також їх послідовностей одночасно й у відкритому тексті, і у шифртексті. Причому цей розподіл зберігатиметься з точністю до заміни символів у процесі як шифрування, так і дешифрування. За умови доволі великої довжини зашифрованого повідомлення моноалфавітні шифри легко піддаються частотному аналізу: якщо частота появи літери в мові та частота появи деякого присутнього в шифртексті символу приблизно однакові, то в цьому випадку з великою ймовірністю можна припустити, що цей символ і буде цією самою буквою. Найпростішим прикладом частотного аналізу може бути підрахунок кількості кожного із символів шифртексту, потім слідують процедури ділення отриманого числа символів на кількість усіх символів у тексті та множення результату на сто, щоб представити остаточну відповідь у відсотках. Потім отримані відсотки порівнюють із таблицею ймовірнісного розподілу букв для мови передбачуваного оригіналу.

Впродовж XV–XVI століть у Європі створювали й розвивали поліалфавітні шифри заміни. Найвідомішим є шифр французького дипломата ***Блеза де Віженера*** (Blaise de Vigenere), в основу якого покладено використання послідовності декількох шифрів ***Цезаря*** (Гай Юлій Цезар, Imperator Gaius Iulius Caesar) з різними значеннями зсуву. Протягом трьох століть шифр Віженера вважали повністю криптографічно стійким, поки в 1863 році ***Фрідріх Вільгельм Казіскі*** (Friedrich Wilhelm Kasiski) не запропонував свою методику злому цього шифру.

Основна ідея методу Казіскі є такою: якщо у відкритому тексті між двома однаковими наборами символів знаходиться такий блок тексту, що його довжина кратна довжині ключового слова, то ці однакові набори символів відкритого тексту під час шифрування перейдуть в однакові відрізки шифртексту. На практиці це означає, що за наявності в шифртексті однакових відрізків завдовжки три й більше символів велика ймовірність того, що ці відрізки відповідають однаковим відрізкам відкритого тексту.

Метод Казіскі застосовують так: у шифртексті шукають пари однакових відрізків довжиною три або більше символів, потім обчислюють відстань між ними, тобто кількість символів, які розділяють стартові позиції парних відрізків. У результаті аналізу всіх пар однакових відрізків отримують

сукупність відстаней $d_1, d_2, d_3, \ldots$ Очевидно, що довжина ключового слова буде дільником для кожної з відстаней і, отже, для їх найбільшого загального дільника.

Метод частотного аналізу відомий уже понад тисячу років. Винахідником його є знаменитий учений арабського світу IX століття ***Аль-Кінді***. Стверджують, що ймовірність появи окремих букв, а також їх послідовність у словах і фразах природної мови підкоряються статистичним закономірностям: наприклад, серед пар букв, що стоять поряд, "ця" в українській мові ймовірніша, ніж "ци", а "оь" у російській чи в українській мовах не зустрічається зовсім (зате часто зустрічається, наприклад, у чеченській). Аналізуючи достатньо довгий текст, зашифрований методом заміни, можна за частотою появи символів провести зворотну заміну й відновити початковий текст. Важливими характеристиками тексту є повторюваність букв (кількість букв у кожній мові обмежена), пари букв, тобто (m-грам), сполучуваність букв одна з однією, чергування голосних і приголосних, а також деякі інші особливості. Важливо зазначити, що ці характеристики є достатньо стійкими.

Ідея криптоаналізу полягає в підрахунку кількості входжень кожної n_m можливих m-грам у достатньо довгих відкритих текстах $T = t_1, t_2, \ldots, t_l$, складених з букв алфавіту $\{a_1, a_2, \ldots, a_n\}$. При цьому є видимими m-грами тексту, що йдуть підряд:

$$t_1 t_2 \ldots t_m, \quad t_2 t_3 \ldots t_{m+1}, \ldots, \quad t_{l-m+1} t_{l-m+2} t_l.$$

Якщо $L(a_{i1} a_{i2} \ldots a_{im})$ – кількість поява m-грами $a_{i1} a_{i2} \ldots a_{im}$ у тексті T, а $\tilde{L}$ – загальна кількість підрахованих m-грам, то за достатньо великих $\tilde{L}$ частоти $L(a_{i1} a_{i2} \ldots a_{im}) / \tilde{L}$, для цієї m-грами мало відрізняються одна від однієї.

Через це відносну частоту вважають наближенням імовірності $P(a_{i1} a_{i2} \ldots a_{im})$ появи цієї m-грами у випадково вибраному місці тексту (такий підхід прийнято для статистичного визначення ймовірності).

Загалом частоту букв у процентному відношенні можна визначити так: підраховують, скільки разів вона зустрічається в шифртексті, потім отриману кількість ділять на загальну кількість символів шифртексту; для отримання процентного відношення її ще множать на 100.

Але існує деяка різниця значень частот, яка пояснюється тим, що частоти істотно залежать не лише від довжини тексту, але й від його характеру. Наприклад, текст може бути технічний, де рідкісна буква Ф може стати доволі частою. Тому для надійного визначення середньої частоти появи букв бажано мати набір різних текстів.

Силові методи. Повний перебір (або метод брутальної атаки, метод "грубої сили", bruteforce) – метод розв'язання задачі шляхом перебору всіх можливих варіантів. Складність повного перебору залежить від кількості всіх можливих розв'язків задачі. Якщо простір розв'язків дуже великий, то повний перебір може не дати результатів протягом декількох років або навіть століть.

Будь-яке завдання з класу ***NP*** – non-determinigtic polynomial (клас містить завдання, які можна "швидко" вирішити на ***недетермінованій машині Тюрінга*** (абстрактна обчислювальна машина)) можна вирішити повним перебором. При цьому, навіть якщо обчислення цільової функції від кожного конкретного можливого розв'язку задачі можна здійснити за поліноміальний час (відносно довжини шифртексту), залежно від кількості всіх можливих розв'язків повний перебір може потребувати експоненційного (відносно довжини шифртексту) часу роботи.

У криптографії на обчислювальній складності повного перебору ґрунтується оцінювання криптостійкості шифрів. Зокрема, шифр вважають криптостійким, якщо не існує методу "злому" істотно швидшого, ніж повний перебір усіх ключів. Криптографічні атаки, основані на методі повного перебору, є найбільш універсальними, але й найдовшими. Цю атаку, зазвичай, використовують лише тоді, коли всі інші були неуспішними.

3.10.3. Криптоаналіз симетричних шифрів

Найбільшого прогресу в розробленні методів розкриття блокових шифрів було досягнуто в самому кінці XX століття, що пов'язано із появою двох методів – диференційного та лінійного криптоаналізу.

Метод диференційного криптоаналізу поєднує узагальнення ідеї загальної лінійної структури із застосуванням імовірнісно-статистичних методів дослідження. Цей метод належить до атак за обраним відкрим текстом. Хоча ***Дон Копперсміт*** (Don Coppersmith) стверджує, що диференційний криптоаналіз був відомий команді розробників алгоритму DES ще на початку 70-х років, офіційною датою появи цього методу вважають 1990 рік, а першість у розробленні визнано за ізраїльськими математиками ***Елі Біхамом*** (Eli Biham) та ***Аді Шаміром*** (Adi Shamir). Диференційний аналіз оснований на використанні нерівноймовірності в розподілі значень різниці двох шифртекстів, отриманих із пари відкритих текстів, що мають деяку фіксовану різницю. Зазначимо, що диференційний аналіз можна застосовувати й для злому хеш-функцій.

Як і диференційний криптоаналіз, ***лінійний криптоаналіз*** є комбінованим методом, що поєднує пошук лінійних статистичних аналогів для рівнянь

шифрування, статистичний аналіз наявних відкритих і зашифрованих текстів, а також методи узгодження й перебору. За цим методом досліджують статистичні лінійні співвідношення між окремими координатами векторів відкритого тексту, відповідного шифртексту й ключа, а також використовують ці співвідношення для визначення за статистичними методами окремих координат ключового вектора.

Сьогодні за методом лінійного криптоаналізу отримано найбільш суттєві результати з розкриття ряду ітераційних систем блокового шифрування, зокрема й системи DES. Метод лінійного криптоаналізу в неявному вигляді з'явився ще в роботі ***Шона Мерфі*** (Sean Murphy) в 1990 році, де його було успішно застосовано під час аналізу системи блокового шифрування ***FEAL***. У 1992 році ***Міцуру Мацуі*** (Mitsuru Matsui) формалізував цей підхід, а пізніше успішно застосував його до аналізу криптоалгоритму DES.

Диференційний (різницевий) криптоаналіз та його модифікації. Диференційний криптоаналіз DES. У 1990 році ***Елі Біхам*** і ***Аді Шамір***, використовуючи метод диференційного криптоаналізу, знайшли спосіб злому DES, ефективніший, аніж злом методом брутальної атаки. Працюючи з парами шифртексту, відкриті тексти яких мають певні відмінності, учені аналізували еволюцію цих відмінностей під час проходження відкритих текстів через етапи DES.

На схемі (рис. 3.44) наведено проходження одного з етапів DES.

Нехай X і X' – пара входів, що відрізняються на ΔX. Відповідні їм виходи відомі й дорівнюють Y і Y', різниця між ними – ΔY. Також відомі перестановка з розширенням і P-блок, тому відомі ΔA і ΔC. B і B' невідомі, але їх різниця дорівнює ΔA, оскільки ключ не змінюється і при обчисленні ΔB байти ключа взаємовиключаються. Для заданого ΔA не всі значення ΔC рівноймовірні, а комбінація ΔA і ΔC дає можливість припустити значення $(A\ XOR\ K_i)$ і $(A'\ XOR\ K_i)$. За відомих A і A' це дає інформацію про K_i.

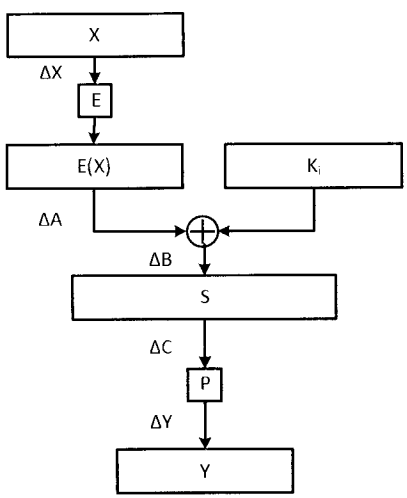


Рис. 3.44. Схема злому DES

Отже, певні відмінності пар відкритих текстів із високою ймовірністю спричиняють певні відмінності отримуваних шифртекстів. Такі відмінності називають характеристиками. Для знаходження характеристик складають таблицю, в якій рядки –

це можливі ΔX, стовпці – можливі ΔY, а на перетині рядка і стовпця пишуть, скільки разів певний ΔY зустрічається для певного ΔX. Різні характеристики можна об'єднувати, і за умови незалежності етапів перемножувати їхні ймовірності.

Пару відкритих текстів, яка відповідає характеристиці, називають ***правильною парою***, а пара, яка не відповідає характеристиці – ***неправильною парою.*** Правильна пара вказує на правильний ключ розглянутого етапу, неправильна – на випадковий ключ. Для знаходження правильного ключа етапу потрібно зібрати достатню кількість припущень: один із ключів зустрічатиметься серед правильних частіше, ніж інші. Знаючи ймовірне значення K_i, отримуємо 48 бітів ключа шифрування K. Решту 8 бітів можна визначити перебором.

Ефективність злому. У найпростішому вигляді диференційний криптоаналіз має низку проблем. Для успішного визначення ключа необхідно набрати деяку порогову кількість припущень, інакше ймовірність успіху мала, або нею можна знехтувати (неможливо виділити правильний ключ із шуму). При цьому зберігання ймовірностей 248 можливих ключів (тобто зберігання лічильника для кожного ключа) потребує великого обсягу пам'яті.

Біхам і Шамір запропонували замість 15-етапної характеристики 16-етапного DES використовувати 13-етапну характеристику й за допомогою низки прийомів отримувати дані для решти етапів. Крім того, вони використовували спеціальні оптимізації для отримання ймовірних 56-бітних ключів, що давало можливість перевіряти їх негайно. Це вирішувало проблему з пороговим характером злому й усувало необхідність у зберіганні лічильників.

Найефективніший алгоритм диференційного розкриття повного 16-етапного DES потребує 247 обраних відкритих текстів. При цьому часова складність становить 237 операцій DES.

Диференційний криптоаналіз застосовують для злому алгоритмів із постійними S-блоками, наприклад, як DES. При цьому його ефективність значно залежить від структури S-блоків. Виявилося, що S-блоки DES оптимізовані проти диференційного криптоаналізу.

Підвищення стійкості до злому. Стійкість DES до диференційного криптоаналізу можна підвищити збільшенням кількості етапів. Диференційний криптоаналіз для DES із 17 або 18 етапами потребує стільки ж часу, скільки й повний перебір, а у разі 19 або більше етапів диференційний криптоаналіз неможливий (тому що для нього буде потрібно більш ніж 264 відкритих текстів, що неможливо за довжини блока 64 біти).

Недоліки методу. Зазначають, що метод диференційного криптоаналізу більшою мірою є теоретичним досягненням. Його застосування на практиці

обмежене високими вимогами до часу й обсягу даних. Крім того, цей метод, насамперед, призначений для розкриття за обраним відкритим текстом. Його можна використати для розкриття з відомим відкритим текстом, але в разі повного 16-етапного DES це робить його навіть менш ефективним, ніж злом брутальною атакою.

Отже, правильно здійснений алгоритм DES зберігає стійкість до диференційного криптоаналізу.

Лінійний криптоаналіз. У криптографії ***лінійним криптоаналізом*** називають метод криптоаналітичного розкриття з використанням лінійних наближень для опису роботи шифру.

Лінійний криптоаналіз винайшов японський криптолог ***Міцуру Мацуї*** (Mitsuru Matsui). Запропонований ним у 1993 році алгоритм був спочатку спрямований на розкриття DES і FEAL. Згодом лінійний криптоаналіз поширився і на інші алгоритми. Розроблено метод атак на блокові й потокові шифри.

Відкриття лінійного криптоаналізу стало поштовхом до побудови нових криптографічних схем.

Принцип роботи. Криптоаналіз здійснюють у два кроки. Перший – побудова співвідношень між відкритим текстом, шифртекстом і ключем, які справедливі з високою ймовірністю. Другий – використання цих співвідношень разом із відомими парами відкритий текст – шифртекст для отримання бітів ключа.

Побудова лінійних рівнянь. Сенс алгоритму полягає в отриманні співвідношень такого вигляду:

$$P_{i1} \oplus P_{i2} \oplus ... \oplus P_{ia} \oplus C_{j1} \oplus C_{j2} \oplus ... \oplus C_{jb} = K_{k1} \oplus K_{k2} \oplus ... \oplus K_{kc},$$

де $P_n, C_n, K_n - n$-ні біти тексту, шифртексту й ключа.

Ці співвідношення називають лінійними апроксимаціями. Для довільно обраних бітів відкритого тексту, шифртексту й ключа ймовірність справедливості такого співвідношення P приблизно дорівнює 1/2. Такими співвідношеннями, ймовірність яких помітно відрізняється від 1/2, можна користуватися для розкриття алгоритму.

Як під час диференційного криптоаналізу, спочатку криптоаналітик знаходить якесь одноразундове співвідношення, потім намагається поширити його на весь алгоритм. На відміну від диференційного криптоаналізу, існують алгоритми пошуку корисних співвідношень. Два алгоритми описав Міцуру Мацуї, інші з'явилися пізніше.

У блокових шифрах аналіз переважно концентрується на S-блоках, оскільки вони є нелінійною частиною шифру. Найефективніше одноразундове співвідношення для алгоритму DES використовує властивість таблиці S5.

Другий вхідний біт таблиці дорівнює результату операції *XOR* над усіма вихідними бітами з імовірністю 3/16 (зміщення в 5/16 щодо 1/2).

Лінійний криптоаналіз має одну дуже корисну властивість: за певних умов можна звести попереднє співвідношення до рівняння вигляду:

$$C_{j1} \oplus C_{j2} \oplus ... \oplus C_{jb} = K_{k1} \oplus K_{k2} \oplus ... \oplus K_{kc}.$$

Тут відсутні біти відкритого тексту, тобто можна побудувати атаку на основі тільки шифртексту, така атака є найефективнішою.

Хоча апроксимацію з найбільшим відхиленням від 1/2 знайти не складно, виникає ряд проблем при екстраполюванні методу на повнораундовий шифр. Перша стосується обчислення ймовірності лінійної апроксимації. У принципі це потребувало б від криптоаналітика переглянути всі можливі комбінації відкритих текстів та ключів, що практично неможливо. Вирішення цієї проблеми в тому, щоб зробити деякі припущення і наблизити ймовірність, використовуючи ***лему про накопичування знаків*** (piling-up lemma). З використанням цієї леми лінійну апроксимацію представляють у вигляді ланцюжка апроксимацій, причому кожна з них охоплює лише невелику частину шифру. Такий ланцюжок називають лінійною характеристикою. Імовірність знаходження комбінації дорівнює

$$P = \frac{1}{2} + 2^{n-1} \prod_{i=1}^{n} (p_i - \frac{1}{2}).$$

Отримавши лінійну апроксимацію, можна застосувати прямий алгоритм і, використовуючи пари відкритий текст – шифртекст, припустити значення бітів ключа. При цьому логічно використовувати максимально ефективне співвідношення, тобто таке, для якого відхилення ймовірності P від 1/2 є максимальним.

Для кожного набору значень бітів ключа в правій частині рівняння імовірності знаходження комбінації P обчислюють кількість пар відкритий текст – шифртекст T, для яких справедлива рівність (лема). Набір, для якого відхилення T від половини кількості пар відкритий текст – шифртекст є найбільшим за абсолютним значенням, вважають найімовірнішим набором значень бітів ключа.

Застосування. Лінійний криптоаналіз спочатку розробляли для атак на блокові шифри, але він може бути застосований і до потокових. Сам розробник детально вивчив його застосування до DES.

Експерименти Міцуру Мацуї здійснення атак із використанням відкритого тексту (обчислення виконували на комп'ютері HP9750 66 МГц) дали наступні результати:

– 8-раундовий DES зламували за допомогою 221 відомого відкритого тексту. Це зайняло в Мацуї 40 секунд;

– 12-раундовий DES зламували за допомогою 233 відомих відкритих текстів. На це знадобилося 50 годин.

– на 16-раундовий DES потрібно 247 відомих відкритих текстів. Ця атака зазвичай не практична. Однак метод дає результати швидше, ніж брутальна атака 56-бітового ключа.

Сучасна обчислювальна техніка здатна виконувати злом швидше.

Доволі часто лінійний криптоаналіз використовують разом із методом брутальної атаки (брутфорсу) після того, як певні біти ключа знайдено за допомогою лінійного криптоаналізу, виконують вичерпний пошук за можливими значеннями інших бітів. Для такого злому DES потрібно 243 відкриті тексти.

На відміну від диференційного криптоаналізу, коли потрібні вибрані відкриті тексти, для застосування лінійного криптоаналізу достатня наявність відомих відкритих текстів, що істотно збільшує область його застосування. Однак, і в лінійному криптоаналізі іноді буває корисно використовувати вибрані відкриті тексти замість відомих. Наприклад, для DES існує методика, що значно зменшує кількість необхідних даних і обчислень, використовуючи вибрані відкриті тексти.

Що стосується атак з використанням лише шифртексту, Мацуї були отримані наступні результати:

– якщо відкритий текст складається з англійських речень у кодуванні ASCII, то 8-раундовий DES можна зламати, використовуючи лише 229 шифртекстів;

– якщо відкритий текст складається з будь-яких символів ASCII, то 8-раундовий DES зламують за допомогою 237 шифртекстів.

Сьогодні від новостворених шифрів вимагають доказ стійкості до лінійного криптоаналізу. Для підтвердження можливості використання нових шифрів потрібно довести їх стійкість до лінійного криптоаналізу.

Захист від лінійного криптоаналізу. Для атаки на блоковий шифр за допомогою лінійного криптоаналізу достатньо, як було описано вище, отримати лінійне співвідношення, істотно зміщене за ймовірністю від 1/2. Відповідно, перша мета проектування шифру, стійкого до атаки, – мінімізувати ймовірні зсуви та переконатися, що подібного співвідношення не існуватиме. Інакше кажучи, необхідно зробити так, щоб за будь-якої зміни тексту або ключа в отриманому шифртексті рівно половина бітів змінювала своє значення на протилежне, причому кожен біт змінювався з імовірністю 1/2. Зазвичай цього досягають вибором високонелінійних S-блоків і посиленням дифузії.

Цей підхід забезпечує добре обґрунтування стійкості шифру, але щоб строго довести захищеність від лінійного криптоаналізу, розробникам шифрів необхідно враховувати складніше явище – ***ефект лінійних оболонок*** (linear hull effect).

Слід зауважити, що шифри, які є оптимальними проти деякого вузького класу атак, зазвичай слабкі проти інших типів атак.

3.10.4. Криптоаналіз асиметричних шифрів

Практично всі алгоритми асиметричної криптографії основані на задачах факторизації (наприклад, відома криптосистема RSA) і дискретного логарифмування в різних алгебраїчних структурах (схема електронного цифрового підпису Ель-Гамала). Незважаючи на те, що приналежність цих задач до класу NP-повних задач не доведено, на сьогодні не знайдено поліноміального алгоритму їхнього розв'язання. Для криптоаналізу асиметричних криптосистем можна застосовувати універсальні методи, наприклад, метод "зустрічі посередині". Інший підхід полягає в розв'язанні математичної задачі, покладеної в основу асиметричного шифрування. З того моменту, як У. Діффі та М. Хеллман у 1976 році запропонували концепцію криптографії з відкритим ключем, проблеми факторизації цілих чисел і дискретного логарифмування стали об'єктом пильного вивчення математиків усього світу. За останні роки в цій галузі спостерігають значний прогрес. Підтвердженням тому може слугувати такий казус: у 1977 році ***Рональд Ривест*** (Ronald Linn Rivest) оголосив, що розкладання на множники 125-розрядного двійкового числа потребує 40 квадрильйонів років, проте вже в 1994 році було факторизовано число, що складається з 129 двійкових розрядів.

Задачу дискретного логарифма вважають складнішою, ніж задачу факторизації. Якщо буде знайдений поліноміальний алгоритм її розв'язання, стане можливим і розкладання на множники (протилежне не доведено).

Останні досягнення теорії обчислювальної складності показали, що загальна проблема логарифмування в кінцевих полях уже не є достатньо міцним фундаментом. Найефективніші сьогодні алгоритми дискретного логарифмування мають уже не експонеційну, а субекспоненційну складність. Це алгоритми "index-calculus", що використовують факторну базу. Перший такий алгоритм для обчислення дискретного логарифма в простому полі Z_p запропонував ***Леонард Адлеман*** (Leonard Adleman). На практиці алгоритм Адлемана виявився недостатньо ефективним; ***Дон Копперсміт*** (Don Cop-

persmith), ***Ендрю Одлижко*** (Andrew Odlyzko) та ***Річард Шреппель*** (Richard Schroeppel) запропонували свою версію ***субекспоненційного алгоритму дискретного логарифмування*** – COS (Coppersmith, Odlyzko, Schroeppel). Алгоритм решета числового поля, який запропонував ***Олівер Шірокауер*** (Oliver Schirokauer) при $p > 10100$ (p – максимальне просте число з факторної бази), працює ефективніше за різні модифікації методу COS.

Низка успішних атак на системи, основані на складності дискретного логарифмування в кінцевих полях, спричинила те, що стандарти ***електронного цифрового підпису*** (ЕЦП), які були прийняті в 1994 році і базувалися на схемі Ель-Гамала, у 2001 році оновлено – переведені на еліптичні криві. Схеми ЕЦП при цьому залишилися, але числа, якими вони оперують, тепер є не елементами кінцевого поля $GF(2n)$ або $GF(p)$, а еліптичними числами – розв'язками рівняння еліптичних кривих над зазначеними кінцевими полями. Алгоритмів, що виконують дискретне логарифмування на еліптичних кривих у загальному випадку хоча б із субекспоненційною складністю, сьогодні не існує, хоча роботи в цьому напрямку здійснюють.

Асиметрична криптографія, яку винайдено й розвинуто за останні два десятиліття минулого століття, обіймала за цей період приблизно таке саме місце, як і блокове симетричне шифрування, яке нараховує півстолітню історію. Асиметричне шифрування (шифрування з використанням відкритого і закритого ключів) представляє собою зовсім іншу технологію.

Кожен користувач володіє двома ключами – відкритим (відомим всім іншим користувачам) та закритим (відомим тільки йому, і який він зберігає в секреті). Відкритий ключ використовують для відправлення повідомлень: за його допомогою на основі спеціального алгоритму будь-який бажаючий може здійснити асиметричне шифрування – необоротне перетворення документа. Але розшифрувати повідомлення може лише власник закритого ключа, тобто законний отримувач.

Ця сама асиметрична схема, використана навпаки (коли для шифрування задіяний закритий ключ), з невеликими доповненнями породила ще одну величезну галузь сучасної криптографії – ЕЦП. Насправді користувач обчислює контрольну суму повідомлення, шифрує її закритим ключем і приєднує шифрограму до повідомлення, проте дієслово "підписує" дуже вдало вписалося в нове значення й тепер невіддільне від поняття ЕЦП.

Слід зазначити, що асиметричне шифрування й ЕЦП вирішують зовсім різні завдання: перше – забезпечення конфіденційності послання, друге – автентичність відправника й цілісність повідомлення.

В ідеальному випадку за наявності закритого ключа й відповідного вибору всіх параметрів задачі процедура шифрування або підписання документа триває частки секунди, а ось на зламування їх без знання закритого ключа потрібні десятиліття.

В асиметричній криптографії й ЕЦП ключ є складнішим компонентом, ніж просто 128-бітовий блок даних симетричної криптографії. Найчастіше й відкритий, і закритий ключі складаються із двох або трьох дуже великих (сотні й навіть тисячі бітів) натуральних чисел, проте є випадки, коли ключ складається із сотень натуральних чисел – певної послідовності або двовимірної матриці. Тому не слід дивуватися тому, що як ключ або цифровий підпис використовуватимуть набори чисел – усе це нероздільний ключ, який має сенс тільки в поєднанні всіх компонентів. Не варто дивуватися і їх розмірам (є схеми із ключами розміром сотні кілобайтів).

Криптоаналіз алгоритму RSA. Вважають, що абсолютну частку обчислювальної складності криптоаналізу RSA становить розв'язання задачі факторизації модуля перетворення. Тому задачу криптоаналізу RSA зводять до задачі розкладання (факторизації) модуля $N = PQ$ на два прості числа: P та Q.

Задача факторизації великих чисел. Факторизацією цілого числа називають його розкладання в добуток простих співмножників. Таке розкладання, згідно з основною теоремою арифметики, завжди існує і є єдиним (з точністю до порядку проходження множників). Усі методи факторизації залежно від їх продуктивності можна поділити на дві групи: експоненційні методи й субекспоненційні методи. Ці методи доволі трудомісткі та потребують значних обчислювальних ресурсів для чисел великої довжини. Проте теоретичне обґрунтування необхідної складності таких обчислень або, інакше кажучи, існування високих нижніх оцінок не доведено, тому питання про існування алгоритму факторизації з поліноміальною складністю на класичному комп'ютері для виконання факторизації є однією з важливих відкритих проблем сучасної теорії чисел. Водночас факторизація з поліноміальною складністю можлива на квантовому комп'ютері за допомогою алгоритму Шора. У цьому розділі розглянемо найвідоміші алгоритми факторизації, що мають експоненційну оцінку збіжності.

$(p+1)$-метод Вільямса. ***$(p+1)$-метод Вільямса*** (Williams) схожий на ***$(p-1)$-метод Поларда*** (John Pollard) й оснований на припущенні гладкості числа $(p+1)$ (число називають гладким, якщо всі його прості множники не перевищують числа 7). Цей метод має добрі показники продуктивності тільки у випадку, коли число $(p+1)$ легко факторизується. Як правило, на практиці цей

метод використовують нечасто через невисокий відсоток подібних випадків. Розглянемо послідовність чисел Люка:

$u_0 = 0;, u_1 = u; u_{n+1} = P^* u_n - Q^* u_{n-1}$, де P, Q – фіксовані цілі числа.

Нехай p – простий дільник числа n, яке факторизують, і виконано розкладання $(p+1)$:

$$p+1 = \prod_{i=1}^{k} q_i^{a_i} .$$

Нехай $B = \max\{q_i^{a_i} \mid 1 \le i \le k\}$. Назвемо натуральне число r B-степенево-гладким, якщо найбільший степінь співмножника $p_i^{a_i}$ у розкладанні r на прості множники не перевищує B. Отже, визначене вище число B є найменшим числом, для якого $p+1$ є B-степенево-гладким. Зазначимо, що оскільки p невідоме, то й B так само не відоме.

Алгоритм Вільямса полягає у такому:

1. Вибираємо деяке число B як верхню межу для розглянутих простих чисел та їхні степенів.

2. Будуємо послідовність $p_i^{a_i}$ простих чисел $2 < 3 < 5 < ... < p_m$, менших за B, і послідовність степенів a_i таку, що $p_i^{a_i} < B$.

3. Покладемо число $R = \sum_{i=1}^{m} q_i^{a_i}$. Якщо p є B-степенево-гладким, то R ділиться на p.

4. Вибираємо випадково числа P та Q і будуємо послідовність чисел Люка, поки не обчислимо u_R.

5. Потім обчислимо $НСД(n, u_R) = d$. Якщо $1 < d < n$, то задачу розв'язано.

Доведено, якщо P і Q взаємно прості, $(P^2 - 4Q)/P = -1$, то властивості послідовності Люка забезпечують знаходження нетривіального дільника для числа n.

Метод факторизації ρ-Полларда. Метод факторизації ρ-Полларда ґрунтується на створенні колізії (співпадінні за модулем) елементів кільця ZN. Його сутність полягає в знаходженні цілих чисел X і Y, які порівнюють за $\mathrm{mod}(N)$, тобто

$$X \equiv Y (\mathrm{mod}\, N).$$

Цю тотожність можна подати у вигляді

$$X - Y = 0 (\mathrm{mod}\, N),$$

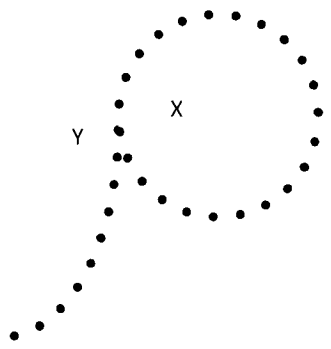


Рис. 3.45. Крива ρ-Полларда створення колізії

а також у вигляді рівняння:

$$(X - Y) = kN.$$

Суть здійснення методу можна звести до такого. За допомогою деякої функції $f(x)$ будують послідовність точок згідно з рис. 3.45 (крива ρ – Полларда).

Після здійснення колізії знаходять значення одного із простих чисел, обчислюючи найбільший спільний дільник між $(X - Y)$ та N, тобто $НСД(X - Y, N)$.

Застосування цього методу суттєво залежить від вибору функції f, згідно з якою будують криву ρ – Полларда. Вочевидь, єдиного підходу тут немає, метод ρ – Полларда ґрунтується скоріше на інтуїції чи досвіді. Для реальних значень N його зараз не використовують. Його єдиною перевагою є можливість розпаралелювання обчислень. Але для дискретного логарифмування в групах точок еліптичних кривих його вважають як найбільш ефективним.

Метод факторизації "квадратичне решето". До 1994 року для розкладання на множники застосовували підхід, відомий як метод квадратичного решета. Загроза ключам великої довжини тут подвійна: безупинне зростання обчислювальної потужності сучасних комп'ютерів і безупинне вдосконалення алгоритмів розкладання на множники.

Розглянемо двійкове решето, яке, відповідно до сучасних поглядів, є найшвидшим за довжини модуля не більшої ніж 120 десяткових цифр.

Необхідно знайти два випадкові цілі числа x та y – такі, що:

$$x^2 = y^2 \bmod N.$$

Представимо цю рівність у вигляді

$$x^2 - y^2 \equiv 0 \bmod N.$$

З урахуванням того, що в тотожності останнього співвідношення операції виконують за модулем N, його можна подати у вигляді рівняння:

$$x^2 - y^2 = kN, k = 1,2,...$$

Якщо розкласти ліву частину цього рівняння як різницю квадратів, то отримаємо, що

$$(x - y)(x + y) = kN, k = 1,2,...$$

причому $N = PQ$.

Отриманий вираз доцільно застосовувати в таких випадках:

1. $(x-y)/N$;
2. $(x+y)/N$;
3. $(x-y)/P \vee (x+y)/Q$;
4. $(x-y)/Q \wedge (x+y)/P$.

У випадках 1 і 2 P або Q знайти не можна, оскільки модуль N не можна розкласти на співмножники. У випадках 3 та 4 маємо розв'язок.

Далі, для третього випадку можемо скористатися алгоритмом Евкліда та обчислити найбільший спільний дільник:

$$\begin{cases} НСД(x-y,N); \\ НСД(x+y,N). \end{cases}$$

Ураховуючи це, можна обчислити P або Q.

Практично факторизацію модуля N із використанням двійкового решета можна здійснити в такій послідовності:

1. Нехай N – число, яке необхідно факторизувати. Побудуємо деяку базу $Z = P_1 P_2 P_3 ... P_K$ з таким значенням Z, щоб $Z \approx N$, де $P_1, P_2, P_3, ... P_K$ – прості числа, краще невеликого розміру; Z – база двійкового решета.

2. Знайдемо $\sqrt{N}$, заокругливши знизу. Потім побудуємо числа вигляду

$$\left(i + \sqrt{N}\right)$$

і знайдемо

$$\left(i + \sqrt{N}\right)^2 \bmod N = S^2.$$

У результаті отримаємо тотожність:

$$\left(i + \sqrt{N}\right)^2 \equiv S^2 \bmod N.$$

Тобто

$$X^2 = Y^2 \bmod N.$$

Пошук дискретного логарифма. Існує велика кількість алгоритмів, основаних на схемі Ель-Гамала. Їх криптостійкість грунтується на обчислювальній складності проблеми дискретного логарифмування, тобто за відомими p, g та y необхідно обчислити x, який задовольняє рівняння

$$Y \equiv g^x \bmod p.$$

Ефективний алгоритм знаходження дискретного логарифма значною мірою знизив би безпеку систем, основаних на схемі Ель-Гамала. Серед найвідоміших алгоритмів пошуку дискретного логарифма слід назвати ***алгоритм малого та великого кроків*** (Baby-step giant-step) та ρ – Полларда. Детальний

розгляд першого алгоритму виходить за межі цього викладення. Слід зазначити, що названі алгоритми мають часову складність $O\left(\sqrt{n}\right)$. Baby-step giant-step неможливо використовувати на практиці через величезні вимоги до пам'яті. Щодо алгоритму ρ – Полларда, то у 1998 році компанія Certicom розпочала змагання з обчислення дискретних логарифмів на еліптичних кривих з бітовою довжиною від 109 до 369. Сьогодні успішно зламано лише криві завдовжки 109 бітів. Останню успішну спробу здійснили в 2004 році.

3.10.5. Силові методи криптоаналізу

Нижче опишемо відомі методи атаки на функції хешування.

Атаку "грубою силою" (брутфорс, брутальна атака, атака на знаходження другого прообразу) можуть використовувати для знаходження прообразу за заданим хеш-значенням або для знаходження прообразу, що дає задане хеш-значення. Суть атаки полягає в послідовному або випадковому переборі вхідних повідомлень і порівнянні результату виконання функції хешування із заданим. Успіх атаки за рівноймовірних випадань хеш-результатів дорівнюватиме 2^{-n}, де n – довжина хеш-значення в бітах. Складність такої атаки оцінюють у кількості виконання $2^n - 1$ операцій обчислення хеш-значень.

Атаку методом "дня народження" виконують для знаходження двох різних повідомлень з однаковими хеш-значеннями. Ця атака основана на парадоксі "дня народження" і полягає в тому, що в двох згенерованих множинах хеш-значень, що містять n_1 і n_2 елементів відповідно, імовірність знаходження збіжних елементів між цими множинами оцінюють співвідношенням: $P \approx 1 - \exp\left(-n_1 n_2 / 2^n\right)$. При $n_1 = n_2 = 2^{n/2}$ імовірність успіху дорівнює $P \approx 1 - 1/e \approx 0{,}63$.

Атака "зустріч посередині" є модифікацією атаки методом "дня народження" і за складністю може бути порівняна з нею.

Атаку з корекцією блока використовують у випадку, якщо хакер має повідомлення й хоче змінити в ньому один або більше блоків без зміни хеш-значення. Один цикл алгоритму хеш-функції MD5, вразливий до цієї атаки: хакер бере блок повідомлення M_i (16 слів по 32 біти), залишає 11 слів, модифікує одне слово й обчислює 4 слова, що залишилися. У результаті виходить блок M_i', що відображається в те саме хеш-значення, що M_i. Повна версія MD5 не вразлива до цієї атаки.

Атака з фіксованою точкою може бути застосовувана за умови, що циклова функція f має одну або декілька фіксованих точок. Фіксованою точкою називають блок повідомлення X_i, для якого виконується

$f(H_{i-1}, X_i) = H_{i-1}$, тобто існує блок повідомлення X_i, що не змінює проміжного результату H_{i-1}. Отже, до повідомлення X можна додавати або видаляти блоки X_i без зміни хеш-значення. Захистом від таких атак слугує обчислення довжини повідомлення й додавання її наприкінці повідомлення.

Атаку на базовий алгоритм шифрування використовують для атаки на функції хешування, що основані на блокових симетричних шифрах. Оскільки алгоритми шифрування розробляли як двоспрямовані (підтримують зворотне перетворення), то це може збільшити вразливість внаслідок застосування функції стискання.

Диференційні атаки. Диференційний криптоаналіз є потужним інструментом для аналізу не лише блокових шифрів, але також і функцій хешування. За такої атаки досліджують вплив різниці вхідних даних функцій стискання на відповідні вихідні різниці. Колізія (співпадіння значень хеш-функції від різних аргументів) виникає, якщо вихідна різниця дорівнює нулю.

Силові методи криптоаналізу на основі паралельних обчислень на основі FPGA (COPACOBANA – Cost Optimized Parallel Code Breaker). Криптоаналіз симетричних та асиметричних шифрів надзвичайно вимогливий до ресурсів обчислень. Оскільки параметри безпеки (зокрема, довжина ключа) майже всіх практичних криптоалгоритмів вибирають так, що атаки з використанням звичайних комп'ютерів є практично нездійсненними, єдиний перспективний спосіб розв'язання існуючих задач криптоаналізу (припускаючи відсутність математичного прориву) полягає в створенні спеціального обладнання. Для цього розроблено та здійснено COPACOBANA (Cost-Optimized Parallel Code Breaker) – машину, яка оптимізована для виконання криптоаналітичних алгоритмів і здійснена за ціну, не більшу за 10000 доларів США. COPACOBANA має 120 недорогих FPGA (Field-Programmable Gate Array, програмована користувачем вентильна матриця) і здатна, наприклад, виконувати вичерпний пошук ключа у DES у середньому протягом дев'яти днів.

3.10.6. Криптоаналіз за побічними каналами

Атаки за побічними (сторонніми) ***каналами*** використовують інформацію, яку можна отримати із пристрою шифрування і яка не є при цьому ні відкритим текстом, ні шифртекстом. Цей підхід є менш узагальненим, але найчастіше більш потужним, ніж класичний криптоаналіз.

Атаки на криптосистеми за побічними каналами використовують слабкості в здійсненні й розміщенні механізмів криптоалгоритму. Такі атаки основані на кореляції між значеннями фізичних параметрів, що вимірюють у

різні моменти часу під час обчислень (споживання енергії, час обчислень, електромагнітне випромінювання тощо), і внутрішнім станом обчислювального пристрою (криптографічного модуля), що має стосунок до секретного ключа. На практиці атаки за побічними каналами на багато порядків ефективніші, ніж традиційні атаки, основані лише на математичному аналізі. При цьому атаки за побічними каналами використовують особливості здійснення (тому їх називають також атаками за здійсненням – implementation attacks) для вилучення секретних параметрів, задіяних в обчисленнях. Такий підхід менш узагальнений, оскільки прив'язаний до конкретної ситуації, але найчастіше потужніший, ніж класичний криптоаналіз.

В останні роки кількість подібних криптографічних атак різко зросла. Наприклад, криптоаналітик може відстежувати енергію споживання смарт-карти, коли вона виконує операції із закритим ключем. Інколи криптоаналітику вдається виміряти час, що витрачають на виконання криптографічних операцій, або аналізувати поведінку криптографічного пристрою у разі виникнення певних помилок. Побічну інформацію на практиці зібрати часом нескладно, тому потрібно обов'язково враховувати таку загрозу під час оцінювання захищеності системи (важливо, наприклад, екранувати серверні приміщення в банках).

Атаки за побічними каналами класифікують за:

– контролем над обчислювальним процесом: ***пасивні*** та ***активні***;

– способом доступу до модуля: ***агресивні*** (invasive), ***напівагресивні*** (semiinvasive) і ***неагресивні*** (non-invasive);

– методом, що застосовують у процесі аналізу: ***прості*** (simple side channel attack – SSCA) і ***різницеві*** (differential side channel attack – DSCA).

За видом побічного каналу атаки поділяють на: ***атаки за часом виконання*** (Timing Attacks); ***атаки за енергоспоживанням*** (Power Analysis Attacks); ***атаки за помилками обчислень*** (Fault Attacks); ***атаки за електромагнітним випромінюванням*** (ElectroMagnetic Analysis); ***атаки за помилками в каналі зв'язку*** (Error Message Attacks). Існують і більш витончені види атак: ***атаки за станом кеш-пам'яті*** (Cache-based Attacks), ***акустичні атаки*** (Acoustic Attacks), ***атаки за світловим випромінюванням*** (Visible Light Attacks).

Розглянемо деякі з названих атак детальніше.

Атака за часом. Атака за часом є способом отримання будь-якої прихованої інформації точним вимірюванням часу, який потрібний користувачу для виконання криптографічних операцій. Це найпоширеніша з атак за побічними каналами. Найчастіше час оброблення даних у криптографічному

модулі трохи змінюється залежно від вхідних значень (наприклад, відкритого тексту або шифртексту). Це є наслідком оптимізації продуктивності й ряду інших причин. Атака за часом заснована на вимірюванні часу, необхідного криптографічному модулю для виконання операції шифрування. Ця інформація може сприяти розкриттю інформації про секретний ключ. Наприклад, ретельно вимірюючи час, необхідний для виконання операцій із секретним ключем, криптоаналітик може знайти точне значення експоненти в алгоритмі Діффі-Хелмана.

У 1995 році інформація про атаки за часом з'явилася в пресі. Закриті ключі RSA можна відновити, вимірюючи відносні інтервали часу, витрачені на виконання криптографічних операцій. Ці атаки було успішно застосовано до карток із мікропроцесорами та інших засобів надійної ідентифікації, а також до серверів електронної комерції в мережі Інтернет.

Атака за потужністю. Атака з аналізу потужності придатна переважно до апаратного здійснення криптографічних засобів. Її успішно застосовують при зламі смарт-карт та інших систем, у яких зберігають секретний ключ. Сучасне обладнання здатне забезпечити вимірювання напруги (для наступного розрахунку потужності) у широкому діапазоні частот із високою точністю.

Атаки за потужністю можна поділити на прості (Simple Power Analysis – SPA) і різницеві (Differential Power Analysis – DPA). Метою SPA є отримання інформації про конкретні інструкції в системі та про конкретні дані, які обробляють у цій системі. У загальному випадку SPA може дати як відомості про роботу криптографічного модуля, так і інформацію про ключі. Для здійснення цієї атаки криптоаналітик повинен мати у своєму розпорядженні точні дані про здійснення криптографічного модуля. Цей метод використовує безпосередні дані вимірювань, зібрані під час виконання криптографічних операцій. Відомо, що проста атака за потужністю для смарт-карт зазвичай займає декілька секунд, тоді як різницева атака може зайняти декілька годин.

DPA, як правило, не потребує даних про конкретне здійснення криптографічного модуля й у якості альтернативи використовує статистичні методи аналізу. Різницевий аналіз сьогодні є одним із найпотужніших засобів для проведення атак, що використовують побічні канали, причому ця атака характеризується дуже малими витратами.

Атаки за помилками обчислень. Помилки апаратного забезпечення, що з'являються під час роботи відповідного криптографічного модуля, або помилкові вихідні дані можуть стати важливими побічними каналами й іноді істотно збільшують уразливість шифру до криптоаналізу. Криптоаналіз на основі формування випадкових апаратних помилок – це вид атаки на шифри, коли криптоаналітик має можливість зовнішньої фізичної дії на крипто-

графічний модуль, яка б спричиняла поодинокі помилки в процесі шифрування одного блока даних. Атаки за помилками на криптографічні алгоритми вивчають від 1996 року. З того часу майже всі криптографічні алгоритми були вразливими до атак такого виду.

Можливість здійснення атаки за помилками залежить від можливостей криптоаналітика спеціально спричиняти помилки в криптографічному модулі або користуватися збоями природного походження. Помилки найчастіше виникають через стрибки напруги, збої часу або через випромінювання різних типів. Розгляд питання стійкості до цього методу є особливо актуальним для криптографічних модулів, що застосовують у смарт-картках.

Здебільшого помилки класифікують за такими критеріями:

– точність, якої криптоаналітик може досягти, вибираючи час й місце, де з'являється помилка під час роботи криптографічного модуля;

– кількість бітів даних, на які впливає помилка, наприклад, лише один біт;

– сталість помилки (помилка є короткочасною чи постійною);

– тип помилки: зміна одного біта; зміна одного біта, але тільки в одному напрямку (наприклад, з 1 на 0); зміна біта на випадкове значення тощо.

Загалом успішна атака за помилками на криптографічні модулі або пристрої потребують двох кроків: крок створення помилки й крок використання помилки. Помилки можуть виникати в смарт-картках внаслідок зовнішнього впливу на них, наприклад, штучне створення неправильних зовнішніх умов. Деякі з них виникають внаслідок аномального й раптового зниження або підвищення напруги, частоти, температури, випромінювання, освітлення тощо.

Різницевий аналіз за помилками полягає у вивченні результату роботи алгоритму шифрування в нормальних і ненормальних умовах за одного й того самого входу (відкритого тексту). Ненормальні умови зазвичай спричиняють помилки в процесі (короткочасна помилка) або перед процесом (постійна помилка) роботи. Різницевий аналіз за помилками глибоко теоретично вивчений й може бути застосовний майже до всіх симетричних криптосистем.

Для ГОСТ 28147-89 було показано можливість розкриття ключа й таблиць підстановок за допомогою криптоаналізу на основі формування випадкових апаратних помилок. DES, RC5 та інші шифри також є уразливими до цього виду криптоаналізу, тому в разі їх використання необхідно забезпечити захист апаратури від нав'язування збоїв.

Атаки за електромагнітним випромінюванням. Виконання обчислювальних операцій на комп'ютері пов'язано з електромагнітним випромінюванням. Криптоаналітик може отримати значну інформацію про обчислення й дані, якщо буде вимірювати й аналізувати це випромінювання. Атаки за

електромагнітним аналізом можна розділити на дві великі категорії: ***прості*** (simple electromagnetic analysis – SEMA) і ***різницеві*** (differential electromagnetic analysis – DEMA).

3.10.7. Нові методи криптоаналізу

Сьогодні існує багато алгоритмів шифрування, які можна умовно поділити на симетричні й асиметричні. Для можливості застосування криптоаналізу бажано знати, за яким алгоритмом відбулося шифрування. Методи диференційного й лінійного криптоаналізу ефективні, якщо алгоритм шифрування є відомим. Наприклад, у попередніх розділах наведено методи криптоаналізу алгоритму DES. На практиці не лише ключова інформація, але й алгоритм шифрування залишаються в таємниці.

Методи криптоаналізу за побічними каналами демонструють ефективність злому, який здійснюють не за логікою алгоритму шифрування, а за іншими підходами.

У цьому параграфі наведено нові методи в криптоаналізі, які демонструють застосування евристичних алгоритмів для злому криптографічних систем. Офіційно не відомо про скільки-небудь серйозні прориви в зломі шифрів за допомогою евристичних алгоритмів. Вважають, що ці методи представляють більш академічний, ніж практичний інтерес. Проте не виключено, що із часом їхнє значення в криптології зросте, як це сталося з алгоритмом Шора. Нижче розглянуто такі методи: нейронні мережі, генетичні алгоритми, квантовий криптоаналіз.

Нейронні мережі. Криптосистему розглядають як "чорну скриню", тобто пристрій або програму, про внутрішню структуру якої нічого невідомо. Подаючи дані на вхід, можна отримати реакцію на виході. Завдання криптоаналізу – ідентифікація цієї системи, тобто визначення її структури на основі вхідних і вихідних сигналів.

Штучна нейронна мережа – це математична модель, а також пристрої паралельних обчислень, що є системою зі з'єднаних і взаємодіючих штучних нейронів – простих процесорів. Кожен штучний нейрон оперує лише зі вхідними сигналами й сигналами, які він надсилає іншим нейронам. У разі з'єднання в доволі велику мережу з керованою взаємодією такі штучні нейрони разом здатні виконувати достатньо складні завдання. Отримані моделі називають ***штучними нейронними мережами*** (ШНМ). Схему ШНМ зображено на рис. 3.46. Чорним позначено два вхідні елементи і один вихідний. Білим позначено шар прихованих нейронів.

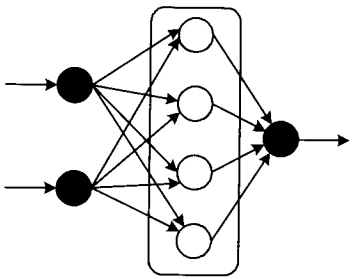


Рис. 3.46. Штучна нейронна мережа із двома входами, прихованим шаром штучних нейронів та одним виходом

Спочатку будь-яка нейромережа має пройти навчання на прикладах. Нейронну мережу потрібно навчити давати правильний відгук на вхідні дані. У процесі навчання здійснюють підбір параметрів штучних нейронів.

Брюс Шнайер (Bruce Schneier) у своїй книзі "Прикладна криптографія" песимістично ставиться до застосування нейронних мереж у криптоаналізі: "Процес злому не залишає місця для навчання: ви або розкриваєте ключ, або ні. Нейронні мережі добре працюють у структурованих середовищах, що допускають навчання, але не у високоентропійному ймовірнісному світі криптографії". Тим не менш, дослідження в цьому напрямку тривають.

Розроблено "атаку чорної скрині" на класичні й потокові криптосистеми, основану на побудові моделі ***"нейророзпізнавач чорної скрині"*** (Black-Box Neuro Identifier). Є дві мети: визначення ключа на основі відкритих і відповідних їм зашифрованих текстів, а також створення нейромоделі досліджуваної криптосистеми.

Ідентифікація системи полягає у визначенні правил функціонування системи "чорної скрині" за вхідними та вихідними даними й апроксимації невідомої функції моделі нейронної мережі. Першим етапом є вибір нейронної моделі, її архітектури й алгоритму навчання. Множину відомих пар відкритий текст – шифртекст поділяють на дві підмножини, одну з яких використовують для навчання мережі, а іншу для перевіряння відповідності отриманої моделі заданому критерію точності. Оптимальною вважають модель, що має мінімальну кількість нейронів і при цьому задовольняє критерій точності.

Є відомими приклади практичного здійснення описаної системи для класичного поліалфавітного шифру Віжинера та потокових шифрів. Удалося отримати модель криптосистеми, яка видає результат зі 100 % точністю для порогу похибки 10^{-5}. Перевагами нейромережевого методу є незалежність результату від мови повідомлення та його статистичних характеристик, оскільки для отримання знань система використовує адаптивний навчальний процес.

Нейронні мережі було застосовано для злому DES. У процесі навчання було використано 2240 пар відкритих текстів і шифртекстів, що дало змогу отримати результати з точністю до 98 %.

Генетичні алгоритми. Шифри на основі завдання про укладання ранця були одними з перших спроб створення системи шифрування з відкритим ключем (***Меркл*** (Merkle) і ***Хеллман*** (Hellman), 1978). У 1983 році ***Брикель*** (Brickell) запропонував спосіб злому криптосистеми на основі ранця низької

щільності. Рік по тому ***Шамір*** (Shamir) розробив поліноміальний алгоритм для атаки на вихідну "ранцеву" криптосистему. Після цього було запропоновано багато інших систем на основі алгоритму укладання ранця: декілька послідовних ранців, ранці ***Грем–Шаміра*** (Garham – Shamir) тощо. Для всіх цих систем було розроблено методи злому. Універсальним методом для криптоаналізу шифрів на основі алгоритму укладання ранця є метод генетичних алгоритмів (далі в тексті застосовані терміни методу генетичних алгоритмів).

Генетичні алгоритми працюють із набором бінарних рядків, що складають популяцію (рис. 3.47). Кожний рядок – особина – є конкретним рішенням. Якість особини (рішення) визначають фітнес-функцією (fitness function). Для операцій над бітовими рядками використовують такі оператори: оператор відбору, оператор схрещування (кросинговер), оператор мутації. Першою стадією генетичного алгоритму є відбір.

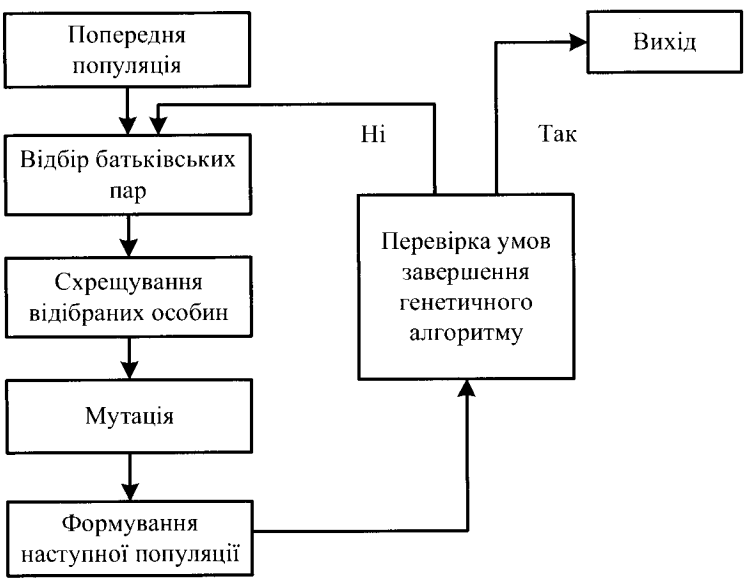


Рис. 3.47. Схема еволюції популяції

У процесі відбору визначають особини, яких використовуватимуть для створення нового покоління. "Батьків" вибирають за методом рулетки (ймовірність відбору особин пропорційна їх якості), що забезпечує більші шанси для якісніших особин, проте кожна особина має шанс бути обраною. Тобто, алгоритм просувається в найперспективнішому напрямі пошуку. Наступна стадія – схрещування. При цьому здійснюють обмін бітовими підрядками завдовжки r; їх довжина та розташування є характеристиками конкретного алгоритму. У результаті роботи оператора схрещування кількість особин у популяції може суттєво збільшитись завдяки новоутвореним особинам –

"нащадкам". Потім працює оператор мутації, який виконує інвертування випадкового біта випадково обраної особини. На початку роботи алгоритму встановлюють фіксовану малу ймовірність мутації. Завершальна стадія – зменшення кількості особин у популяції до їхньої початкової кількості, при цьому в новій популяції залишають кращі особини.

Формально "ранцеву" задачу формулюють так. Дано набір значень $M_1, M_2, ..., M_n$ (n – довжина блоку відкритого тексту в бітах) і сумарне значення S (цільове значення). Потрібно знайти значення b_i таке, що $S = b_1 M_1 + b_2 M_2 + ... + b_n M_n$. Тут b_i може бути нулем (значення M_i немає в ранці) або одиницею (значення M_i є в ранці). Тоді представлення вмісту ранця є особиною, біти якої відповідають значенням b_i.

Фітнес-функція "кращих" особин оцінює близькість ваги конкретного ранця до заданого числа. Значення фітнес-функції набувають значення з діапазону [0, 1], де 1 означає точний збіг із шуканою вагою. Якщо вага одного ранця перевищує цільове значення S на деяке число x, а вага іншого, навпаки, менша за необхідну на те ж число x, то "кращим" вважають останній ранець.

Загальний алгоритм для криптоаналізу завдання про укладання ранця має вигляд:

1. Створюють випадкову популяцію двійкових особин.
2. Для кожної особини обчислюють значення фітнес-функції.
3. На основі отриманих значень фітнес-функції відбирають батьківські пари.
4. До обраних пар застосовують оператор кросинговер.
5. Нащадків із певною ймовірністю піддають мутації.
6. Застосовують оператор відбору "кращих особин" і формують нову популяцію.

Процес завершать, коли кількість поколінь перевищить певну задану кількість або буде досягнуто необхідної загальної якості популяції. Кращих особин буде використано для злому шифру.

Для простого шифру підстановки алгоритм показав добрі результати: для досягнення оптимальної точки, тобто секретного ключа, алгоритму знадобилось досліджувати в середньому не більше 2 % всього ключового простору. Генетичні алгоритми успішно застосовують для криптоаналізу шифрів перестановки та підстановки.

Квантовий криптоаналіз. За допомогою квантового комп'ютера можна виконувати обчислення, що не здійснені на сьогоднішніх (класичних) комп'ютерах. У 1994 році ***Пітер Віллістон Шор*** (Peter Williston Shor) відкрив "обмежено-імовірнісний" алгоритм факторизації, який дає змогу розкласти на

множники число N за поліноміальний від розмірності задачі час $O\left((\log N)^3\right)$. Алгоритм Шора розкладання чисел на множники є головним досягненням в області квантових обчислювальних алгоритмів. Саме із цього моменту почали активно фінансувати роботи зі створення квантових комп'ютерів.

У класичних обчисленнях одиницею інформації є біт. Кожен біт може перебувати тільки в одному із двох можливих станів: 0 або 1. Регістр з N бітів може містити одну з 2^N можливих комбінацій станів. Для оброблення й перетворення інформації використовують побітові логічні операції – однобітну НЕ (NOT) і двобітні І (AND) та АБО (OR). Бітові операції описують через таблиці істинності. У них наведено відповідність вхідних і вихідних аргументів. Алгоритм класичних обчислень – це набір послідовних побітових операцій.

У класичних обчисленнях до пам'яті комп'ютера завантажують лише один з 2^N варіантів даних, і для цього варіанта обчислюють значення функції. У результаті одночасно обробляють лише один з 2^N можливих наборів даних.

У пам'яті квантового комп'ютера одночасно представлені всі 2^N комбінацій вихідних даних. Перетворення застосовують до всіх цих комбінацій відразу. У результаті за одну операцію обчислюють функцію для всіх 2^N можливих варіантів набору даних.

У більшості алгоритмів, зокрема алгоритмі Шора, використовують стандартний спосіб зведення задачі розкладання до задачі пошуку періоду функції. В алгоритмі Шора використано квантовий паралелізм для отримання суперпозиції всіх значень функції за один крок. Потім виконують квантове перетворення Фур'є, результатом якого, як і для класичного перетворення, є Фур'є-образ. Аргумент Фур'є-образу кратний величині, яка є оберненою до періоду. З високою ймовірністю вимір стану повертає період, який, своєю чергою, використовують для розкладання цілого числа N.

Алгоритм Шора простий і потребує набагато простішого апаратного забезпечення ніж те, яке потрібно для універсального квантового комп'ютера. Фактично можна говорити про спеціалізований квантовий пристрій для розкладання на множники числа N. Очікують, що такий спеціалізований квантовий пристрій буде побудовано задовго до того, як весь діапазон квантових обчислень стане технологічно здійсненним.

Контрольні питання до розділу 3

1. Історія криптології. Основні етапи розвитку.
2. Шифр Цезаря та Скитала. Квадрат Полібія. Особливості цих шифрів.
3. Основні поняття та визначення криптології. Класична криптографічна схема.
4. Задачі криптології.

5. Типи криптоаналітичних атак.
6. Стійкість алгоритмів шифрування.
7. Алфавіти. Простори повідомлень, криптотекстів, ключів. Зашифровувальне та розшифровувальне відображення.
8. Шифри простої заміни: означення, приклади, об'єм простору ключів. Шифр Цезаря із ключовим словом.
9. Криптоаналіз шифру простої заміни. Частотний криптоаналіз.
10. Гомофонний шифр заміни та його криптоаналіз.
11. Поліграмні шифри. Шифр Плейфейра та його різновиди.
12. Поліалфавітні криптосистеми. Шифр Віженера та його криптоаналіз.
13. Шифр Віженера з автоключем та його криптоаналіз.
14. Шифри перестановки. Матричний шифр обходу та його криптоаналіз.
15. Кількаразове шифрування. Шифр ADFGVX.
16. Роторні машини.
17. Подання тексту в цифровій формі. Афінний шифр зсуву 1-го порядку. Приклад зашифрування.
18. Лінійний та узагальнений афінні шифри 1-го порядку. Приклади зашифрування.
19. Афінні шифри вищих порядків. Шифр зсуву k-го порядку. Приклад зашифрування.
20. Лінійний та узагальнений афінні шифри k-го порядку. Приклади зашифрування.
21. Алгоритм звичайного XOR. Приклад зашифрування.
22. Криптоаналіз шифру звичайного XOR.
23. Шифр одноразового блокнота. Абсолютна стійкість. Приклад зашифрування.
24. Потокові шифри. Генератори псевдовипадкових чисел.
25. Способи проектування генераторів псевдовипадкових чисел.
26. Системно-теоретичний підхід до проектування генераторів псевдовипадкових чисел (ГПВЧ) на регістрах зсуву зі зворотними зв'язками.
27. Складнісно-теоретичний підхід до проектування ГПВЧ.
28. Класичний алгоритм Евкліда. Приклади застосування.
29. Розширений алгоритм Евкліда. Приклади застосування.
30. Методи компонування сучасних блокових симетричних шифрів.
31. Мережі Фейстеля.
32. SP-мережі.
33. Історія розроблення й упровадження DES. Загальна схема DES із поясненнями.
34. Схема раунду DES із поясненнями.
35. Перетворення ключа в DES.
36. Розширююча перестановка. Лавинний ефект.
37. S-блоки – схема включення та алгоритм підстановки. Приклади.
38. Потрійний DES.
39. Режими застосування блокових симетричних шифрів. Недоліки режиму простого заміщення. Режим зчеплення зашифрованих блоків.
40. Режими застосування блокових симетричних шифрів: режим зворотного зв'язку за виходом, режим зворотного зв'язку за криптотекстом та режим зчеплення блоків відкритого тексту.
41. Шифр ГОСТ 28147-89.
42. Стандарт AES.
43. Національний стандарт блокового симетричного шифрування ДСТУ 7624:2014 – алгоритм "Калина".
44. Криптографія з відкритим ключем. Концепція асиметричної криптографії, порівняння із симетричною: переваги, недоліки та приклади.
45. Криптографічна система на основі телефонного довідника та демонстрація на її основі основних особливостей асиметричних криптосистем.

46. Історія асиметричної криптографії. Основні поняття та інструменти асиметричної криптографії.
47. Головоломки Меркла.
48. Криптографічні та важкооборотні функції. Приклади.
49. Рюкзаки. Проблема пакування рюкзака. Застосування для шифрування. Приклад зашифрування.
50. Надзростаючі рюкзаки. Приклади. Задача пакування надзростаючого рюкзака. Алгоритм та приклад розв'язання задачі.
51. Алгоритм Меркла–Хелмана перетворення надзростаючого рюкзака на нормальний. Приклад перетворення.
52. Зашифрування та розшифрування в рюкзачній криптосистемі Меркла–Хелмана. Приклади.
53. Алгоритм RSA. Опис системи. Генерування ключів системи RSA. Приклади.
54. Зашифрування та розшифрування в системі RSA. Приклади.
55. Коректність, надійність та ефективність системи RSA.
56. Бінарний алгоритм піднесення до степеня: концепція, табличний варіант. Приклади застосування.
57. Загальний опис бінарного алгоритму піднесення до степеня. Приклад застосування.
58. Система Рабіна. Генерування ключів та зашифрування в системі Рабіна.
59. Розшифрування в асиметричній криптосистемі Рабіна.
60. Криптографічна система Ель Гамала. Генерування ключів. Зашифрування. Розшифрування.
61. Асиметричні криптосистеми на еліптичних кривих.
62. Альтернативна криптографія.
63. Поняття криптоаналізу.
64. Які типи розкриття розглядають у криптоаналізі?
65. Для розкриття яких шифрів призначений диференційний аналіз?
66. Який метод є більш дієвим для розкриття блокових шифрів?
67. Які методи застосовують для злому хеш-функцій?
68. Що таке криптоаналіз за побічними каналами?
69. Для яких задач можна використати алгоритм Шора?
70. Які основні задачі виникають під час криптоаналізу асиметричних шифрів?

Список літератури до розділу 3

1. Bruce Schneier. Applied cryptography: protocols, algorithms, and source code in C / 2nd ed. – New York : John Wiley & Sons, Inc.,1995. – 792 pages.
2. Вербіцький О. В. Вступ до криптології / О. В. Вербіцький. – Львів : Вид-во наук.-техн. літ., 1998. – 247 с.
3. Kahn D. The Codebreakers: The Story of Secret Writing / D. Kahn. – New York : Maemillan Publishing Co., 1967. – 1164 p.
4. Shannon C. E. Communication Theory of Secrecy Systems / C. E. Shannon // Bell System Technical Journal. – 1949. – Vol. 28, n. 4. – P. 656–715.
5. Шеннон К. Работы по теории информации и кибернетике : [пер. с англ.] / К. Шеннон; под ред. Р. Л. Добрушина и О. Б. Лупанова; с предисловием А. Н. Колмогорова. – М. : Изд-во иностранной литературы, 1963. – 830 с. Теория связи в секретных системах. – С. 333–402.
6. Державна служба спеціального зв'язку та захисту інформації України, Інститут кібернетики імені В.М. Глушкова Національної академії наук України. Положення про проведення відкритого конкурсу криптографічних алгоритмів [Електронний ресурс]. – Режим доступу : www. URL: http://www.dstszi.gov.ua/dstszi/control/uk/publish/printable_article?art_id=48383/

7. ДСТУ 7624:2014. Інформаційні технології. Криптографічний захист інформації. Алгоритм симетричного блокового перетворення [Текст]. – Введ. 01–07–2015. – К. : Мінекономрозвитку України, 2015.
8. Горбенко І. Д. Симетричний блоковий шифр "Калина" – новий національний стандарт України / І. Д. Горбенко, Р. В. Олійников, О. В. Казимиров, В. І. Руженцев, О. О. Кузнєцов, Ю. І. Горбенко, О. В. Дирда, В. І. Долгов, А. І. Пушкарьов, Р. І. Мордвінов // Радиотехника. – 2015. – Вып. 181. – С. 5–22.
9. Лунин А. В. Перспективы развития и использования асимметричных криптоалгоритмов в криптографии / А. В. Лунин, А. А. Сальников // Конфидент. – 1998. – № 6. – С. 15–23.
10. Montgomery P. L. Modular Multiplication without Trial Division / P. L. Montgomery // Mathematics of Computation. – 1985. – Vol. 44, No. 170. (Apr., 1985). – P. 519–521.
11. Horpenyuk A. Fast algorithms and computing means of cryptological functions / A. Horpenyuk // Computing. – 2005. – Vol. 4, Issue 2. – P. 69–76.
10. Aubum B. R. Quantum Encryption – A Means to Perfect Security? / B. R. Aubum. – SANS Institute. – 2003. – P. 1–16. – Режим доступу : www.sans.org/reading-room/whitepapers/vpns/quantum-encryption-means-perfect-security-986/
11. Hughes R.J. Quantum Cryptography / Richard J. Hughes D. M. Alde, P. Dyer, G. G. Luther, G. L. Morgan and M. Schauer. – University of California, Physics Division, Los Alamos National Laboratory, Los Alamos, 1994. – P. 1–44. – Режим доступу: https://arxiv.org/abs/quant-ph/9504002/
12. Xiao G. New field of cryptography: DNA cryptography / G. Xiao, L. Qin, M. Lu, X. Lai. – Chinese Science Bulletin 51(12). – June 2006. – P. 1413–1420. – Режим доступу : www.researchgate.net/publication/227268565_New_field_of_cryptography_DNA_cryptography/
13. Salehi S. A. Computing Mathematical Functions using DNA via Fractional Coding / S. A. Salehi, X. Liu, M. D. Riedel, K. K. Parhi, 29 May 2018. – P. 1–14. – Режим доступу : www.nature.com/articles/s41598-018-26709-6.pdf/
14. Naor M. Visual cryptography / M. Naor, A. Shamir // In Proc. Eurocrypt 94, Perugia, Italy, May 9–12, LNCS 950, Springer Verlag, 1994. – P. 1–14. – Режим доступу:http://www.fe.infn.it/u/filimanto/scienza/webkrypto/visualdecryption.pdf/
15. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. – М. : Триумф, 2003. – 806 с. – Режим доступу : http://www.dut.edu.ua/uploads/l_1134_27449793.pdf/
16. Сингх С. Книга шифров. Тайная история шифров и их расшифровки / С. Сингх. – М. : Аст: Астрель, 2006. – 447 с.
17. Бауэр Ф. Расшифрованные секреты. Методы и принципы криптологии / Ф. Бауэр. – М. : Мир, 2007. – 550 с.
18. Biham E. Differential Cryptanalysis of the Data Encryption Standard / E. Biham, A. Shamir. – 2009, 188 p. – Режим доступу: http://www.cs.technion.ac.il/~biham/Reports/differential-cryptanalysis-of-the-data-encryption-standard-biham-shamir-authors-latex-version.pdf/
19. Matsui M. Linear Cryptanalysis Method for DES Cipher / M. Matsui // Advances in Cryptology, EUROCRYPT '93 Proceedings, Springer-Verlag, 1994. – P. 386–397.
20. Moriai S. Improving the Search Algorithm for the Best Linear Ex-pression // S. Moriai, K. Aoki, K. Ohta // Advances in Cryptology, CRYPTO '95 Proceedings, Springer-Verlag, 1995. – P. 157–170.
21. Langford S. Diferential-Linear Cryptanalysis / S. Langford, M. Hellman // Advances in Cryptology, CRYPTO '94 Proceedings, Springer-Verlag, 1994. – P. 17–26.
22. Chabaud F. Links Between Differential and Linear Cryptanalysis / F. Chabaud, S. Vaudenay // Advances in Cryptology, EUROCRYPT '94 Proceedings, Springer-Verlag, 1995. – P. 356–365.
23. Smith C. Basic Cryptanalysis Techniques / C. Smith. – 2001, 10 p. – Режим доступу : https://www.sans.org/reading-room/whitepapers/vpns/basic-cryptanalysis-techniques-752/
24. Druin J. Cryptanalysis of the Vigenère Cipher / J. Druin. – 2018, 33 p. – Режим доступу : https://pen-testing.sans.org/resources/papers/gcih/learning-cryptography-wrong-cryptanalysis-vigenere-cipher-126917/

Розділ 4
СТЕГАНОГРАФІЯ

4.1. Стеганографічні системи

4.1.1. Сфери застосування стеганографії

Розвиток засобів обчислювальної техніки дав новий поштовх для розвитку комп'ютерної стеганографії. З'явилося багато нових галузей її застосування. Більшість поточних досліджень у галузі стеганографії так чи інакше пов'язані із цифровим обробленням сигналів. Це дає змогу говорити про цифрову стеганографію [1]. За допомогою стеганографії приховують сам факт існування таємного повідомлення, на відміну від криптографії, метою якої є приховування вмісту повідомлень шифруванням.

Предмет стеганографії, основні терміни та визначення. Отримання доступу до інформації з появою глобальних комп'ютерних мереж стало неймовірно простим. Це значно підвищує загрозу порушення безпеки інформації за відсутності засобів щодо її захисту, а саме загрозу неавторизованого доступу до інформації. Тому актуальним є питання розроблення методів і засобів захисту інформації, зокрема методів криптографії та стеганографії.

Криптографічний захист інформації – система зміни останньої з метою зробити її незрозумілою, приховати зміст повідомлень шифруванням. Цей захист не вирішує згаданої вище проблеми повністю, оскільки наявність шифрованого повідомлення привертає увагу, і зловмисник, заволодівши криптографічно захищеним файлом, здогадується про розміщення в ньому секретної інформації й теоретично може її дешифрувати.

Приховування ж самого факту існування секретних даних під час їх передавання, зберігання або оброблення є завданням ***стеганографії*** – науки, що вивчає методи та способи приховання конфіденційних відомостей.

Стеганографування здійснюють різними способами. Загальною ж ознакою таких способів є те, що приховуване повідомлення вбудовують в об'єкт, що не привертає увагу. Цей об'єкт потім відкрито передають отримувачеві. Історично напрям стеганографічного приховування інформації був першим, але

згодом багато в чому був витіснений криптографією. Інтерес до стеганографії відродився в останні десятиліття й пояснюється поширенням технологій мультимедіа.

Методи стеганографії дають можливість не тільки приховано передавати інформацію (класична стеганографія), але й успішно вирішувати завдання завадостійкої автентифікації, захисту інформації від несанкціонованого копіювання, відстежування поширення інформації телекомунікаційними мережами, пошуку інформації в мультимедійних базах даних тощо.

Сфери застосування. Практичні аспекти побудови стеганографічних систем. Цифрова стеганографія як наука зародилася в останні десятиріччя. Вона містить такі напрямки:

1) вбудовування інформації з метою її прихованого передавання;
2) вбудовування ***цифрових водяних знаків*** (ЦВЗ, watermarking);
3) вбудовування ***ідентифікаційних номерів*** (fingerprinting);
4) вбудовування ***заголовків*** (captioning).

ЦВЗ можна застосовувати, переважно, для захисту від копіювання та несанкціонованого використання. У зв'язку з бурхливим розвитком технологій мультимедіа гостро постало питання щодо інформації, представленої в цифровому вигляді. Назву цей метод отримав від усім відомого способу захисту цінних паперів, зокрема грошей, від підроблення. На відміну від звичайних водяних знаків ЦВЗ можуть бути не лише видимими, але, як правило, невидимими. Невидимі ЦВЗ аналізують спеціальним декодером, що приймає рішення про їхню коректність. ЦВЗ можуть містити деякий автентичний код, інформацію про власника або яку-небудь керуючу інформацію. Найбільш придатними об'єктами захисту за допомогою ЦВЗ є нерухомі зображення, відеофайли тощо.

Завдання вбудовування й виділення повідомлень з іншої інформації виконує стеганографічна система. Узагальнену модель стеганографічної системи (стегосистеми) наведено на рис. 4.1.

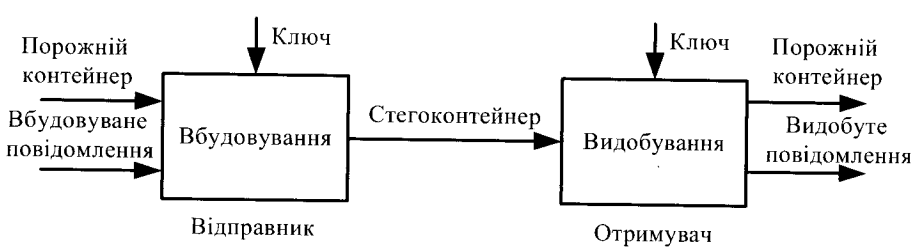


Рис. 4.1. Узагальнена модель стеганографічної системи

Стеганографічна система складається з таких основних елементів:

– вбудовуване (приховуване) повідомлення (стегоповідомлення) – інформація (текст, зображення, відеофайл, аудіофайл тощо), що підлягає приховуваному передаванню за допомогою вбудовування в контейнер;

– контейнер – будь-яка інформація, що призначена для приховування вбудованих повідомлень;

– порожній контейнер – контейнер без вбудованого повідомлення;

– стегоконтейнер (заповнений контейнер, стего) – контейнер, який містить вбудовану інформацію;

– ключ (стегоключ) – секретний ключ, який необхідний для приховування інформації; залежно від кількості рівнів захисту (наприклад, вбудовування попередньо зашифрованого повідомлення) у стегосистемі може бути один чи декілька стегоключів;

– видобуте повідомлення – інформація (текст, зображення, відеофайл, аудіофайл тощо), що була приховано передана шляхом вбудовування в контейнер;

– стеганографічний канал (стегоканал) – канал приховуваного передавання повідомлень (канал передавання стего).

4.1.2. Атаки на стеганографічні системи та протидія їм

Відомо, що зловмисник може бути пасивним, активним і злочинним. Залежно від цього він може створювати різні загрози.

Пасивний зловмисник може лише виявити факт наявності стегоканалу й, можливо, читати повідомлення. Чи зможе він прочитати повідомлення після його виявлення, залежить від стійкості системи шифрування, і це питання, як правило, не розглядають у стеганографії. Якщо є можливість виявити факт наявності стегоканалу, то стеганографічну систему зазвичай вважають нестійкою. Виявлення стегаканалу є найбільш трудомістким завданням, а захист від виявлення (за визначенням) вважають основним завданням стеганографії. Діапазон дій активного зловмисника значно ширший. Приховане повідомлення він може вилучити або зруйнувати. Дії злочинного зловмисника найнебезпечніші. Він здатний не лише руйнувати, але й створювати помилкові стегоконтейнери. Це може призводити до катастрофічних наслідків.

Для здійснення тієї або іншої загрози зловмисник застосовує атаки.

Найпростіша атака – суб'єктивна. Зловмисник уважно розглядає зображення чи відеозапис (слухає аудіозапис), намагаючись визначити "на око", чи є в ньому приховане повідомлення. Зрозуміло, що таку атаку можна здійснити лише проти зовсім незахищених стеганографічних систем. Проте

вона, напевно, найпоширеніша на практиці, принаймні, на початковому етапі розкриття стеганографічної системи.

Класифікація атак на стеганографічні системи цифрових водяних знаків (ЦВЗ). Відомо, що цифрові водяні знаки повинні відповідати вимогам візуальної (аудіо) непомітності. Надалі як контейнер використовуватимемо файл зображення, але все сказане може стосуватися також відеофайлів та аудіофайлів.

Звернемося до системи вбудовування повідомлень змінюванням ***найменшого значущого біта*** (Least Significant Bit – LSB) пікселів. Практично будь-який спосіб оброблення зображень може зруйнувати значну частину вбудованого повідомлення. Наприклад, розглянемо операцію обчислення ковзного середнього по двох сусідніх пікселах $(a+b)/2$, що є найпростішим прикладом низькочастотної фільтрації. Нехай значення яскравості пікселів a і b можуть бути парними або непарними числами з імовірністю $p=1/2$. Тоді й значення найменшого значущого біта зміниться після усереднення в половині випадків. До того ж ефекту може призвести й зміна шкали квантування, скажімо, від 8 до 7 бітів. Аналогічно впливає і стискання зображень із втратами. Більше того, застосування методів очищення сигналів від шумів, що використовують оцінювання й усунення шуму, призведе до перекручування переважної більшості бітів прихованого повідомлення.

Існують також і набагато більш згубні для ЦВЗ операції оброблення зображень, наприклад, масштабування, повороти, усікання, перестановка пікселів. Ситуація ускладнюється ще й тим, що перетворення вбудовуваного повідомлення може здійснювати не тільки зловмисник, але й законний користувач. Таке перетворення також може бути наслідком помилок під час передавання каналом зв'язку.

Незначне зміщення пікселів під час вбудовування повідомлення відправником або передавання стегоконтейнера каналом зв'язку може призвести до невиявлення ЦВЗ у разі його видобування отримувачем.

Можлива різна класифікація атак на стеганографічні системи. Розглянемо атаки, специфічні для систем ЦВЗ. Можна виділити наступні категорії атак проти таких стеганографічних систем:

1. Атаки проти вбудовуваного повідомлення – спрямовані на видалення або пошкодження ЦВЗ шляхом маніпулювання стегоконтейнером. За допомогою методів, що входять до цієї категорії, не намагаються оцінити чи виділити водяний знак. Прикладами таких атак можуть бути лінійна фільтрація, стискання зображень, додавання шуму, вирівнювання гістограми зображення, зміна контрастності тощо.

2. Атаки проти стеганодетектора, який використовують для видобування прихованого повідомлення і виявлення наявності такого повідомлення в стегоконтейнері, спрямовані на те, щоб утруднити або унеможливити правильну роботу детектора. При цьому ЦВЗ у зображенні залишають, але губиться можливість його розпізнавання. До цієї категорії входять такі атаки, як афінні перетворення (тобто масштабування, зсуви, повороти), усікання зображення, перестановка пікселів тощо.

3. Атаки проти протоколу використання ЦВЗ – переважно пов'язані зі створенням помилкових ЦВЗ, та стегоконтейнерів, інверсією ЦВЗ, додаванням декількох ЦВЗ.

4. Атаки проти самого ЦВЗ – спрямовані на оцінювання й витягнення ЦВЗ зі стегоконтейнера, за можливості без перекручування контейнера. До цієї групи входять такі атаки, як атаки змови, статистичного усереднення, методи очищення сигналів від шумів, деякі види нелінійної фільтрації тощо.

Треба зазначити, що розглянута класифікація атак не є єдино можливою й повною. Крім того, деякі атаки (наприклад, видалення шуму) можна зарахувати до декількох категорій.

Відповідно до цієї класифікації всі атаки на системи вбудовування ЦВЗ можна поділити на чотири групи:

1) атаки, спрямовані на видалення ЦВЗ;

2) геометричні атаки, спрямовані на перекручування стегоконтейнера;

3) криптографічні атаки;

4) атаки проти використовуваного протоколу вбудовування й перевіряння ЦВЗ.

Методи протидії атакам на системи ЦВЗ.

Для визначення прихованого повідомлення в імовірному контейнері (зображення, відеофайли, аудіофайли) зловмисники застосовують ***статистичний стеганоаналіз*** (стегоаналіз), що ґрунтується на вивченні статистичних властивостей файлу. Наприклад, розподіл значень молодших бітів цифрових зображень, відеофайлів чи аудіофайлів має, переважно, характер, подібний до шуму (помилки квантування). Вони несуть найменшу кількість інформації, і їх можна використовувати для вбудовування прихованого повідомлення. При вбудовуванні прихованого повідомлення, можливо, зміняться статистичні властивості файла, що й буде слугувати під час стеганоаналізу ознакою наявності стегоканалу.

Для непомітного вбудовування повідомлень необхідно вирішити три завдання: виділити підмножину бітів, модифікація яких мало впливає на якість, вибрати із цієї підмножини потрібну кількість бітів відповідно до розміру приховуваного повідомлення та їх відредагувати. Якщо статистичні властивості

контейнера не змінилися, то вбудовування повідомлення можна вважати успішним. Оскільки розподіл імовірності значень найменших значущих бітів найчастіше близький до білого шуму, вбудовані дані повинні мати подібний розподіл. Цього досягають попереднім шифруванням повідомлення або його стисненням.

У найпростіших стеганографічних системах ЦВЗ для вбудовування використовують псевдовипадкову послідовність, що є реалізацією білого шуму з гауссовим розподілом амплітуд. Такі системи практично нестійкі до більшості розглянутих вище атак. Для підвищення робастності (здатності системи відновлюватись) стеганографічних систем можна запропонувати низку покращень.

У робастній стеганографічній системі необхідно правильно вибрати параметри псевдовипадкової послідовності. Відомо, що при цьому системи з розширенням спектра можуть бути достатньо робастними стосовно атак типу додавання шуму, стиску тощо. Вважають, що ЦВЗ повинен не зазнавати спотворень навіть за доволі сильної низькочастотної фільтрації.

Є декілька ***способів підтвердження прав власності на цифрові дані за допомогою ЦВЗ***. Один зі способів полягає у вбудовуванні в ЦВЗ деякої часової позначки, що надає третя, довірена сторона. У разі виникнення конфлікту особу, яка має на зображенні більш ранню часову позначку, вважають справжнім власником.

Один із принципів побудови робастного ЦВЗ полягає в адаптації його спектра. У деяких роботах показано, що обвідна спектра ідеального ЦВЗ має повторювати обвідну спектра контейнера.

Для захисту від ***атак типу афінного перетворення*** можна використовувати додатковий (опорний) ЦВЗ. Цей ЦВЗ не містить повідомлення, але його використовують для "реєстрації" виконуваного зловмисником перетворення. На приймальній стороні застосовують схему попереднього деформування, що виконує обернене до афінного перетворення. Однак у цьому випадку атака може бути спрямована саме проти опорного ЦВЗ. Іншою альтернативою є вкладення ЦВЗ у візуально значущі області зображення, які не можна вилучити з нього без суттєвого його спотворення. Нарешті, можна розмістити ЦВЗ у коефіцієнтах, що є незмінними під час афінних перетворень. Наприклад, амплітуда перетворення Фур'є зображення є незмінною під час зсуву зображення (при цьому змінюється тільки фаза).

Іншим методом захисту від таких атак є ***блоковий детектор***. Модифіковане зображення розбивають на блоки розміром 12×12 або 16×16 пікселів, і для кожного блока аналізують усі можливі спотворення. Тобто пікселі в блоці піддають повороту, перестановці тощо. Для кожної зміни визначають

коефіцієнт кореляції ЦВЗ. Перетворення, після якого коефіцієнт кореляції виявився найбільшим, вважають реально виконаним зловмисником. Отже, з'являється можливість виявити внесені зловмисником спотворення стегофайлу. Можливість такого підходу ґрунтується на припущенні про те, що зловмисник не буде значно спотворювати контейнер (це не в його інтересах).

4.1.3. Пропускна здатність каналів приховуваного передавання повідомлень

Для більшості сучасних стеганографічних систем характерна обернено пропорційна залежність їх ***стеганостійкості*** (стійкості до спотворення вбудованого повідомлення) від обсягу вбудовуваних даних. Можна виділити задачі, в яких основною метою є передавання невеликих обсягів даних (коротке секретне повідомлення-пароль) із забезпеченням максимальної стеганостійкості. З іншого боку, існують задачі, в яких необхідно передавати максимально можливий обсяг даних.

Найпоширеніший метод стеганографічного вбудовування змінюванням найменших значущих бітів (LSB метод) забезпечує найбільшу швидкість вбудовування даних, що робить його найуживанішим як на локальних робочих станціях, так і в режимах реального часу у високошвидкісних каналах передавання даних. У разі вбудовування повідомлень у растрові зображення з 24-бітним кодуванням кольору пікселів заміною лише наймолодших бітів максимальний обсяг вбудовуваних даних становить 12,5 % від розміру контейнера. Оскільки більшість зображень, відеофайлів та аудіофайлів у цифровому представленні володіють суттєвою надлишковістю, виникає питання про можливість змінювання не лише молодших бітів, але й старших, включно до четвертого. Інакше кажучи, йдеться про граничне перевищення рівня мультиплексування пропускної здатності стеганографічного каналу. Під мультиплексуванням (multiplexing – багатоканальне передавання) розуміють розміщення в межах ширини смуги пропускання одного каналу декількох каналів із меншою шириною.

Для вирішення цієї задачі як контейнери можна взяти, наприклад, аудіофайли у wav-форматі, які не було стиснено. Нехай j – відлік аудіосигналу. Згідно з моделлю представлення аудіофайлів відлік j може містити 1 байт (за 8-бітового кодування) чи 2 байти (за 16-бітового кодування) інформації. Нехай i – номер біта в j-му відліку аудіосигналу. Тоді i набуває таких значень у межах одного відліку:

$$i = \begin{cases} \overline{0,7}, \text{ за 8-бітового кодування}; \\ \overline{0,15}, \text{ за 16-бітового кодування}, \end{cases} \quad (4.1)$$

де номеру молодшого біта відповідає значення $i = 0$.

На слух установлено, що простір стеганографічного каналу SC у випадку максимально допустимих стеганографічних змін аудіо-контейнера має вигляд:

$$SC \subset j_{mx}, \quad (4.2)$$

де mx – кількість змінених бітів у відліку чи рівень мультиплексування:

$$mx = \begin{cases} \overline{0,1}, \text{ за 8-бітового кодування}; \\ \overline{0,3}, \text{ за 16-бітового кодування}. \end{cases} \quad (4.3)$$

Як контейнери-зображення можна вибрати контейнери в BMP-форматі, в якому кожна із трьох (червона, зелена, синя) колірна складова пікселя складається з 8 бітів. Вбудовування можна виконувати і у кожну колірну компоненту, і вибірково. Установлено візуально, що коли вбудовування виконують у кожну колірну компоненту контейнера-зображення, простір стеганографічного каналу під час максимального змінювання (мультиплексування) контейнера має вигляд:

$$SC \subset j_{R_{mx}, G_{mx}, B_{mx}}, \text{ де } mx = \overline{0,3}. \quad (4.4)$$

де $j_{R_{mx}, G_{mx}, B_{mx}}$ – контейнер-зображення зі зміненими колірними складовими пікселя.

На рис. 4.2, *а* показано незаповнений контейнер-зображення. Рис. 4.2, *б* відповідає рівню мультиплексування $mx = 1$, що означає, що у вихідному контейнері заміщено лише один найменший значущий біт кожного відліку (кожної колірної компоненти). На рис. 4.2, *в, г* заміщено відповідно 2 і 3 молодші біти відліків контейнера бітами приховуваного повідомлення.

З рис. 4.2 видно, що використання другого та третього рівня мультиплексування є недопустимим, оскільки призводить до помітних візуальних спотворень на зображеннях (рис. 4,2, *в, г*). Отже, для зображень, які містять великі області, що заповнені одним кольором, використання мультиплексування вищих рівнів є неприпустимим, оскільки вносить візуально помітні втрати в заповнений контейнер.

Для оцінювання спотворень на LSB-модифікованих контейнерах із різним рівнем мультиплексування використовують різноманітні підходи. Найуживаніші є ***середньоквадратичне відхилення*** (root mean square – RMS), ***пікове відношення сигнал/шум*** (peak signal-to-noise ratio – PSNR) та ***кількісне оцінювання візуальної якості зображень*** [2].

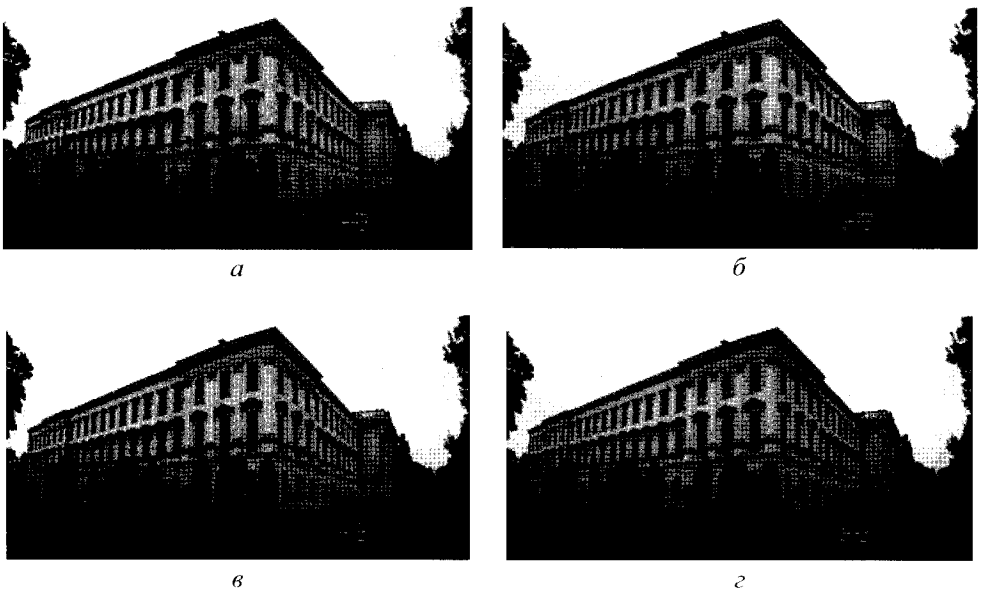


Рис. 4.2. Мультиплексування пропускної здатності контейнера-зображення: а – порожній контейнер-зображення; б – тх = 1; в – тх = 2; г – тх = 3

4.1.4. Оцінювання стійкості стеганографічних систем

На відміну від криптографічних систем, поняття та оцінювання безпеки стеганографічних систем є складнішими та припускають велику кількість їх визначень. Це зумовлено і складністю задач стеганографічного захисту інформації, і недоопрацюваннями теоретичного характеру [3].

Під стійкістю різноманітних систем розуміють їхню здатність приховувати від кваліфікованого порушника факт передавання даних, здатність протистояти спробам порушника спотворити чи знищити приховане повідомлення, а також здатність підтвердити чи спростувати оригінальність переданої інформації.

Розглянемо класифікацію атак порушника, який намагається визначити факт передавання приховуваного повідомлення й при встановленні цього факту робить спробу переглядати їх.

Стійкість різних стеганографічних систем можна поділити на стійкість до виявлення факту передавання (існування) приховуваного повідомлення, стійкість до добування приховуваного повідомлення, стійкість до нав'язування помилкових повідомлень каналом прихованого зв'язку (імітостійкість), стійкість до відновлення стегоключа стеганографічної системи.

Для аналізу стійкості стеганографічних систем до виявлення факту передавання приховуваних повідомлень використовують теоретико-інформаційну модель стеганографічної системи із пасивним порушником.

На основі аналізу розподілів імовірності значень бітів у порожніх та заповнених контейнерах виявляють факт використання стеганографічної системи. Для цього в розглянутій теоретико-інформаційній моделі передбачають, що порушник знає точні ймовірнісні характеристики контейнерів, стегоконтейнерів, приховуваних повідомлень і ключів. Також у моделі передбачають, що передані стегоконтейнери й порожні контейнери не зазнають жодних спотворень у процесі їх доставлення каналом зв'язку, а відправник приховуваних повідомлень вибирає тільки такі контейнери, характеристики яких збігаються з характеристиками всієї множини контейнерів. У підсумку будь-яке відхилення статистичних характеристик спостережуваного порушником у каналі зв'язку повідомлення від середньостатистичних характеристик порожніх контейнерів має кваліфікуватися як факт виявлення стегоканалу.

Очевидно, що така ідеальна модель не зовсім відповідає реаліям. Насправді зловмисник не може мати стільки інформації про контейнери, приховувану інформацію, вибирати контейнери тощо. Окрім того, розглянута вище модель не враховує завад, які вносить сам канал. Тому існує також інший підхід до визначення стійкості стеганографічної системи [4]. Отже, стеганографічну систему називають стійкою, якщо порушник не здатний отримати жодної інформації про вбудоване повідомлення, аналізуючи перехоплені стегоконтейнери за умови знання статистичних характеристик порожніх контейнерів.

4.2. Методи стеганографії

4.2.1. Приховування даних у нерухомих цифрових зображеннях, відео- та аудіофайлах

Одна з областей застосування методів приховування даних у нерухомих цифрових зображеннях, відео- та аудіофайлах – передавання через Інтернет даних, секретність яких потрібно зберігати протягом нетривалого проміжку часу (наприклад, декілька днів) [5]. Це можуть бути новини, переписка, документи, аудіофайли тощо. Розглянемо такі методи.

Приховування даних у зображення. Для забезпечення великої кількості вбудовуваних даних (повідомлень) можна використовувати, наприклад, контейнер-зображення в 24-бітовому BMP-форматі. Найпоширенішим серед методів вбудовування в просторовій області є ***метод змінювання найменшого значущого біта*** (LSB-метод) пікселів зображення.

Найменший значущий біт зображення містить найменше інформації. Відомо, що людина у більшості випадків не здатна помітити зміну в цьому біті. Фактично найменший значущий біт є шумом, тому для вбудовування інформації можна замінювати найменші значущі біти пікселів зображення бітами секретного повідомлення. При цьому для зображення в градаціях сірого (кожен піксель зображення кодують одним байтом) обсяг вбудовуваних даних може становити 1/8 від загального обсягу контейнера. Наприклад, у зображення розміром 512×512 пікселів можна вбудувати приблизно 32 кБ інформації. Якщо змінювати два молодші біти (що також практично непомітно), то пропускну здатність каналу приховуваного передавання повідомлень можна збільшити ще вдвічі.

Визначення кількості бітів для вбудовування обчислюють для кожного пікселя контейнера-зображення на підставі його значення, причому областями з максимально можливою кількістю даних, які можна вбудувати, є граничні області, які, своєю чергою, визначають за допомогою фільтра виділення границь (наприклад, Собела, Превіт тощо). Розподіляють вбудований файл, по контейнеру, використовуючи генератор псевдовипадкових послідовностей.

У вищерозглянутому найпростішому випадку змінюють найменші значущі біти всіх послідовно розташованих пікселів зображення. Інший підхід – ***метод псевдовипадкового інтервалу*** – полягає у випадковому розподілі бітів секретного повідомлення по контейнеру, у результаті чого відстань між двома вбудованими бітами визначають псевдовипадково. Ця методика особливо ефективна у випадку, коли бітова довжина секретного повідомлення є істотно меншою за кількість пікселів зображення. За найпростішим варіантом цього методу інтервал між двома послідовними вбудовуваннями бітів повідомлення є функцією координат попередньо модифікованого пікселя.

Приховування даних у відеофайлах. Найпопулярнішими стандартами кодування відео є MPEG-2 і MPEG-4.

Стеганографічні методи, які застосовують для вбудовування даних у відеофайли, сформовані за стандартами MPEG-2 і MPEG-4, мають працювати в реальному часі. Способи вбудовування даних, які працюють у реальному часі, мають відповідати декільком вимогам, насамперед, вони повинні характеризуватися малою обчислювальною складністю. Отже, єдино прийнятними є методи, за якими вбудовують дані безпосередньо в потік стиснених даних, щоб уникнути зайвих обчислень.

Окрім того, операція вбудовування даних не повинна збільшувати розміру стиснених відеофайлів, оскільки можуть виникнути проблеми під час передавання відеопотоку каналом із фіксованою швидкістю.

Приховування даних в аудіофайлах. Найпростішим серед методів приховування даних в аудіофайлах є ***метод змінювання найменшого зна-***

чущого біта (LSB-метод) у кожній точці вибірки із аудіосигналу, представленого в двійковому вигляді, на біт приховуваного повідомлення. Використання цього методу характеризується високою пропускною здатністю каналу, платою за що є ледь відчутний низькочастотний шум.

Метод розширення спектра майже ідентичний до методу приховування даних у нерухомих зображеннях. Секретне повідомлення рівномірно розподіляють у межах ширини спектра носійного сигналу так, щоб співвідношення сигнал (повідомлення) / шум у каналі було дуже низьким і не викликало підозр щодо наявності повідомлення. Сигнал, з якого формують файл-контейнер, у цьому випадку обирають набагато більшим за амплітудою порівняно із секретним повідомленням.

Метод приховування даних із використанням сигналу луни (echo signal) передбачає вбудовування повідомлення в аудіофайл-контейнер додаванням до основного аудіосигналу додаткового сигналу луни. Дані приховують за допомогою зміни параметрів сигналу луни: початкової амплітуди, швидкості загасання та зсуву. Коли зсув між оригінальним сигналом та сигналом луни зменшувати, то починаючи з певного значення слухова система людини стає не здатною виявити різницю між двома сигналами, а сигнал луни сприймає лише як додатковий резонанс. Цей метод непростий у здійсненні, тому що описане значення зсуву дуже важко визначити. Воно значною мірою залежить від якості початкового сигналу і, зрозуміло, від слухача.

Метод фазового кодування. Існують різні варіанти методів вбудовування інформації на основі фазового кодування, що полягають у зміні фази кожної спектральної складової дискретного сигналу. Для цього вихідний сигнал розбивають на серію коротких сегментів, що містять однакову кількість елементів (відліків). Кількість елементів повинна бути вдвічі більша, ніж кількість бітів у переданому повідомленні. До кожного сегмента застосовують дискретне перетворення Фур'є, у результаті якого для кожного сегмента утворюють масиви фаз і амплітуд. Для збереження прихованості повідомлення необхідно запам'ятати різницю фаз між сусідніми сегментами, оскільки слухова система людини більш чутлива до різниці фаз, ніж до абсолютних значень фази. Приховують повідомлення, змінюючи фази першого сегмента. Вбудовують інформацію заміною вихідного значення фази на значення, що дорівнює мінус $\pi/2$, якщо приховуваний біт повідомлення є нулем, і на значення $\pi/2$, якщо біт повідомлення дорівнює одиниці. Щоб зберегти первинну різницю фаз, необхідно до отриманого масиву фаз першого сегмента додати раніше обчислений масив різниць між першим і другим масивом фаз, і так далі для кожного масиву фаз. Для відновлення звукового сигналу слід виконати зворотне дискретне перетворення Фур'є для масивів амплітуд і модифікованих масивів фаз.

4.2.2. Текстова стеганографія

Стеганографію, у якій для приховування повідомлень використовують текстові контейнери, називають текстовою [6]. Ідея текстової стеганографії полягає в тому, що таємне повідомлення в будь-який спосіб замінюють двійковим кодом – або з використанням таблиці кодів, або за власним правилом. Далі за отриманою послідовністю нулів та одиниць відбуваються заміни в текстовому контейнері (модуляція вибраного символу послідовністю нулів та одиниць).

Для приховування повідомлень використовують можливості розташування та зміни кількості символів у тексті, які не враховують під час читання людиною й комп'ютерного аналізу текстового файла. Це може бути додаткова кількість пропусків і знаків табуляції в різних частинах рядка, чергування деяких службових символів, великих і маленьких літер, букв із різних алфавітів, які є подібними. До методів текстової стеганографії належать: форматування, зміна порядку проходження символів кінця рядка, метод хвостових пропусків, метод знаків однакового відображення та зміни коду пробілу тощо.

Розглянемо декілька методів текстової стеганографії детальніше.

Застосовуючи ***метод форматування (вирівнювання) тексту за допомогою пробілів,*** залишають один пробіл між словами, якщо в двійковому представленні таємного повідомлення зустрічається біт зі значенням 0, а якщо зі значенням 1 – дописують ще один пробіл. Але безпосереднє застосування цього підходу хоча й можливе, але на практиці породжує чимало незручностей, зокрема, оформлення тексту стає неохайним (текст ніби набрано початківцем), що дає змогу легко запідозрити в ньому наявність приховуваних даних. Цю проблему можна вирішити, перерозподіляючи пробіли в межах поточної довжини рядка, переносячи за можливості довгі пробіли в її кінець. У результаті рядки вихідного тексту мають акуратний вигляд, що утруднює виявлення стегоповідомлення.

У ***методі зміни порядку проходження символів кінця рядка*** використовують нечутливість переважної більшості засобів відображення текстової інформації до порядку проходження символів повернення каретки (Carriage Return – CR) і зміни рядка (Line Feed – LF), що обмежують рядок тексту. Традиційний порядок проходження CR/LF відповідає біту таємного повідомлення зі значенням 0, а інвертований LF/CR – зі значенням 1.

Метод хвостових пробілів передбачає дописування в кінці коротких рядків (менше 225 символів; значення 225 вибрано досить довільно) від 0 до 15 пробілів, що кодують біти таємного повідомлення.

Метод знаків однакового відображення передбачає підміну (якщо біт таємного повідомлення має значення 1) або відмову від такої підміни (якщо біт таємного повідомлення має значення 0), наприклад, українського символу латинським, які мають одне й те саме візуальне представлення.

Метод двійкових нулів є різновидом методу знаків однакового представлення й передбачає або заміну першого в групі із двох або більше внутрішніх пробілів двійковим нулем (якщо біт таємного повідомлення має значення 1), або відмову від такої заміни (якщо біт таємного повідомлення має значення 0).

Незважаючи на простоту здійснення текстової стеганографії та її можливе поширення, сьогодні практично не розроблено методик її виявлення. З автоматичних методів текстової стеганографії у відкритій літературі згадують здебільшого один: форматування (тобто вирівнювання) тексту за допомогою пробілів. При цьому серед поширених програмних засобів, що здійснюють методи текстової стеганографії, відомий метод однакового відображення знаків.

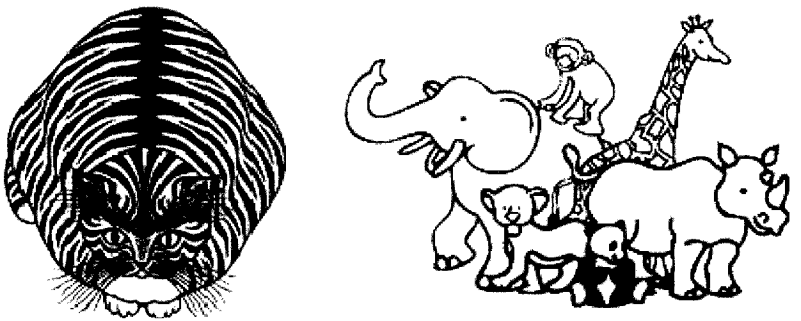
4.2.3. Практичне застосування стеганографії

Процес приховування інформації в мультимедійних файлах-контейнерах розглянемо на прикладі програми ***Steganography-Tools*** (скорочена назва S-Tools, автор Andrew Brown).

Як файл-контейнер програма використовує як графічні, так і аудіо-файли. Графічні файли повинні мати формат bmp (24-бітовий рисунок) або gif, звукові – формат wav. Ця програма спочатку стискає текст повідомлення, далі шифрує його методами криптографії й лише після цього вкладає його у файл-контейнер, при цьому приховану інформацію рівномірно розподіляє по всьому файлу.

Криптографічне закриття інформації здійснюють згідно з одним з алгоритмів: IDEA (International Data Encryption Algorithm – симетричний міжнародний алгоритм шифрування даних), DES (Data Encryption Standard – стандарт шифрування даних), 3DES (потрійний DES) або MDC (Manipulation Detection Code – код виявлення маніпуляцій) з 128-бітовим ключем. Ключ формують із символів пароля, який надає користувач.

Працюють із програмою за принципом Drag and Drop (перенести та покласти – "перетягнути"). На початку потрібно розгорнути вікно програми так, щоб воно займало частину екрана. На вільній частині екрана розгорнути папку ***Провідник*** або папку ***Мій комп'ютер*** із піктограмою файла-контейнера. Як контейнер візьмемо гравюру відомого українського художника-графіка Яківа Гніздовського "The Cat" (файл Tomcat.bmp). Розмір файла-контейнера 522 кБ (24-бітовий рисунок). Як файл-повідомлення візьмемо зображення з ресурсу kleurplaat dieren afrika (файл zoo.bmp) розміром 115 кБ і приховаємо його у файлі Tomcat.bmp (рис. 4.3).



а *б*

Рис. 4.3. Зображення файла-контейнера Tomcat.bmp (а) і зображення файла-повідомлення zoo.bmp (б)

Для приховання даних у файлі-контейнері слід виконати такі дії:

1. "Перетягнути" мишкою піктограму файла-контейнера всередину вікна програми S-Tools. У правому нижньому куті вікна програми (рис. 4.4) з'явиться інформація про максимально можливий об'єм файла-повідомлення (66,595 кБ).

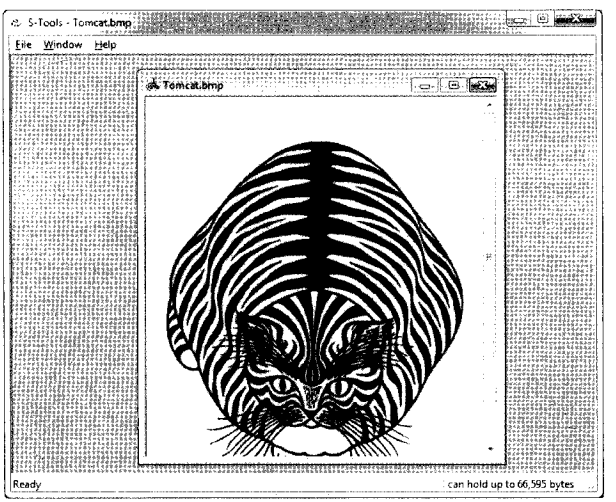


Рис. 4.4. Вікно програми S-Tools із перетягнутим файлом-контейнером

2. За допомогою програми ***WinRAR*** стиснути файл-повідомлення й записати його у вигляді zoo.rar, при цьому файл зменшиться до розміру 26,715 кБ. "Перетягнути" мишкою піктограму файла-повідомлення всередину вікна програми S-Tools.

3. Якщо розмір контейнера є достатнім для приховування даних, з'явиться наступне вікно. У нього ввести пароль (***Passphrase***), підтвердити його

(***Verify Passphrase***), а також вибрати один із запропонованих алгоритмів шифрування (рис. 4.5). Зазначимо, що у разі введення пароля програма S-Tools розрізняє літери різних регістрів.

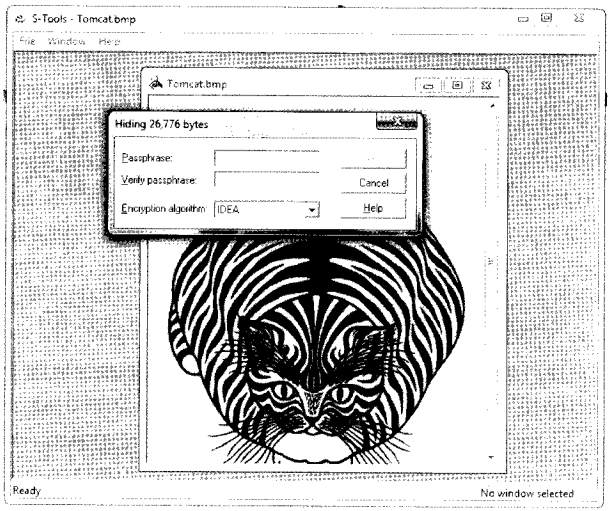


Рис. 4.5. Вибір пароля та типу шифрування

4. Заповнений контейнер з'явиться в новому вікні ***Hidden data*** (Приховані дані). Результат зберегти, вибравши в контекстному меню опцію ***Save as***, при цьому вказати ім'я отриманого файла з розширенням (Tomcat1.bmp), а також місце збереження (рис. 4.6).

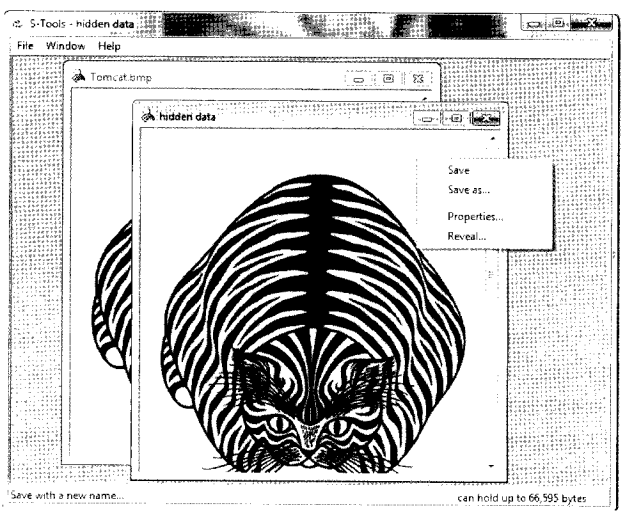


Рис. 4.6. Виклик контекстного меню для збереження стегофайла

Видобувати приховане повідомлення зі стегофайла слід у зворотній послідовності:

1. "Перетягнути" мишкою піктограму стегофайла всередину вікна програми S-Tools.

2. Правою кнопкою миші викликати контекстне меню, вибрати опцію ***Reveal*** (рис. 4.7).

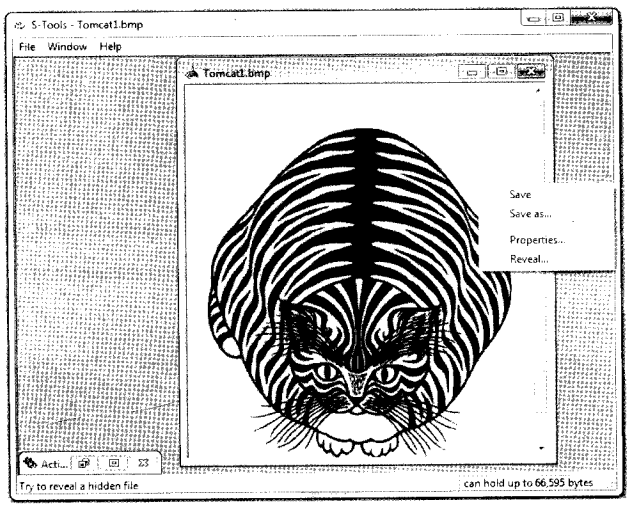


Рис. 4.7. Виклик контекстного меню для видобування прихованого файла

3. Набрати пароль у пунктах ***Passphrase*** та ***Verify Passphrase***, зазначити алгоритм шифрування, який було використано при шифруванні даних.

4. Якщо все введено правильно, з'явиться вікно ***Revealed Archive***, в якому знаходиться прихований файл. Клікнути (натиснути й відпустити) правою кнопкою миші по прихованому файлу, у контекстному меню вибрати опцію ***Save as*** та зазначити місце на диску, де треба зберегти файл (рис. 4.8).

Порівнюючи результати роботи програми S-Tools – стегофайла Tomcat1.bmp (рис. 4.9, *а*) і файла-контейнера Tomcat.bmp (рис. 4.9, *б*), бачимо, що розміри обох файлів Tomcat1.bmp і Tomcat.bmp збігаються як у бітах, так і в пікселях.

За допомогою програми ***HEdit32*** (шістнадцятковий редактор) можна побачити різницю між вмістом файла-контейнера та стегофайла. Для цього створимо за допомогою графічного редактора Paint зображення прямокутника чорного кольору з розмірами 20 на 20 пікселів.

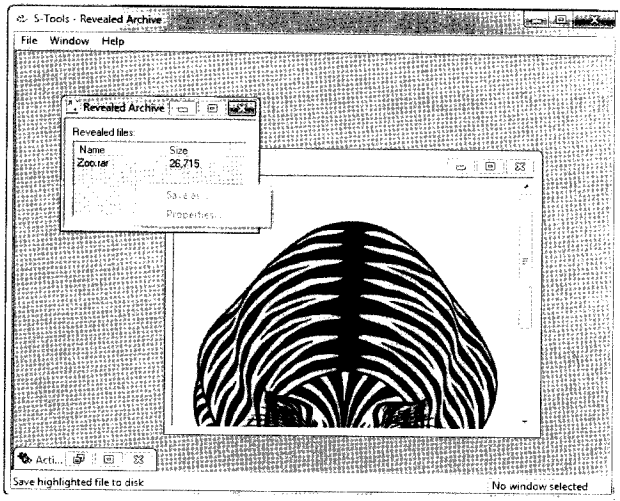
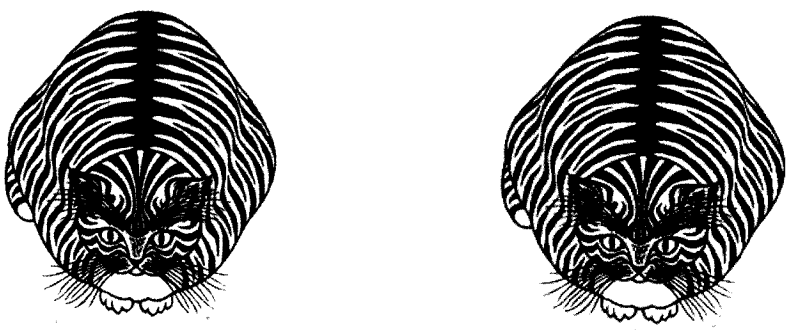


Рис 4.8. Збереження прихованого файла в каталозі



а *б*

Рис. 4.9. Зображення стегофайла (а) і файла-контейнера (б)

Збережемо цей файл під назвою black_20_20.bmp. У текстовому редакторі Блокнот наберемо літери FGR і збережемо цей файл за назвою FGR.txt. Виконаємо приховання текстового файла в графічний файл за допомогою програми S-Tools, як було описано вище, і збережемо його за назвою black_20_20_FGR.bmp.

Тепер відкриємо файл black_20_20.bmp у редакторі HEdit32. На рис. 4.10 показано вміст файла для чорного квадрата розміром 20 на 20 пікселів. Перші три рядки містять службову інформацію (заголовок). Оскільки квадрат чорний, то байти, що відображають колір квадрата, містять нулі (00).

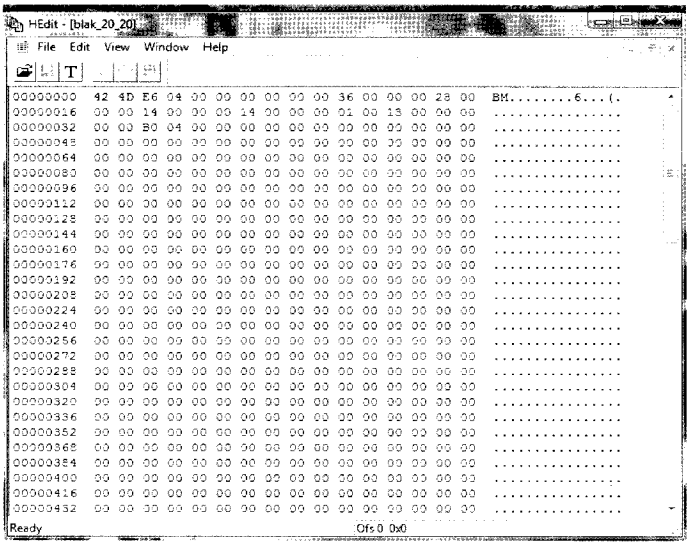


Рис 4.10. Вміст файла-контейнера black_20_20.bmp

У лівому стовпці зазначено восьмирозрядні шістнадцяткові числа – адреси комірок пам'яті. У правому стовпці наведено вміст комірок пам'яті в шістнадцятковій системі числення.

На рис. 4.11 показано вміст стегофайла black_20_20_FGR.bmp (файла-контейнера із вбудованими в нього літерами FGR).

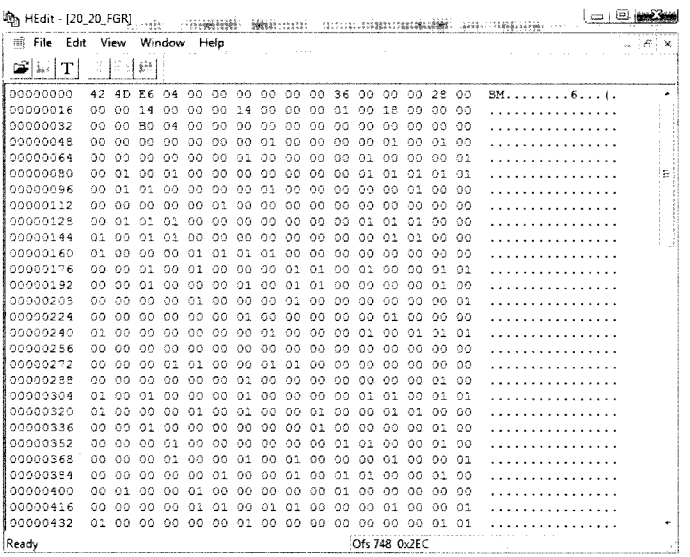


Рис 4.11. Вміст стегофайла

У перших рядках міститься службова інформація (комірки 0...32). Вміст комірок, які відповідають за зображення, змінено. Як і згадувалося вище, інформацію прихованого файла рівномірно розподілено по всьому контейнеру. Візуально різниці між зображеннями файла-контейнера й стегофайла не спостерігається.

Музичні файли дають змогу приховувати великий обсяг інформації. Якщо перетворюють аналоговий сигнал на цифровий із частотою дискретизації 44,1 кГц, то кожну секунду можливо зберігати 44100 бітів інформації для монофонічного сигналу та 88200 бітів – відповідно для стереофонічного (у разі застосування LSB-методу). Отже, у звуковому файлі, який відображає звук впродовж 1 секунди, можна вмістити текст обсягом, більшим за 10 кБ.

У мережі Інтернет можна знайти велику кількість програм, що дають можливість приховувати файли одних форматів у файлах інших форматів. Назвемо деякі з них.

Програма ***BMP Secrets*** дає змогу приховувати всередині файлів формату bmp дуже великі обсяги інформації, а саме 63 % від розміру файла-контейнера. Якщо ж здійснити стиснення фотографії й приховувати не сам файл, а його архів, то у фотографії меншого розміру можна приховати фотографію більшого розміру. Другою відмінністю цієї програми є можливість вкладання більш таємного файла до менш таємного, який, своєю чергою, вкладений у файл-контейнер. Експерименти із застосуванням стиснутого файла показали можливість чотирикратного вкладання. Зазначимо, що в цій програмі використовують алгоритм стиснення для знаходження в рисунку надлишковості, після чого її замінюють корисною інформацією, яку потрібно приховати.

Secret Letter – це додаток для платформи ***Android***, що є стеганографічним інструментом. Він дає можливість зашифровувати й розшифровувати текстові повідомлення в зображеннях або фотографіях, зроблених за допомогою камери мобільного пристрою. Програма дає змогу приховано передати необхідну інформацію, зокрема приховати сам факт цього передавання.

Mr. Crypto – програма для приховування будь-яких типів даних у зображеннях без їх візуальних змін. Усі дані за замовчуванням шифрують; програма використовує ***AES*** (Advanced Encryption Standard – симетричний алгоритм блокового шифрування) або 3DES, що ускладнює їх виявлення.

Steganos Privacy Suite 16 містить практично всі інструменти шифрування й захисту конфіденційності, а також вбудований ***VPN-сервіс*** (Virtual Private Network — віртуальна приватна мережа). Цей продукт є скоріше менеджером окремих інструментів і програм, ніж повністю інтегрованим рішенням. Деякі представлені компоненти Steganos Privacy Suite 16 можна використовувати як окремі програми.

Контрольні питання до розділу 4

1. Що таке стеганографія?
2. Методи приховування даних у нерухомих зображеннях.
3. Методи приховування даних у відеофайлах.
4. Методи приховування даних у аудіофайлах.
5. Методи текстової стеганографії.
6. Як оцінюють стійкість стеганографічних систем?
7. Який формат повинен мати графічний файл-контейнер у разі його використання в програмі S-Tools?
8. Як визначити максимальний обсяг файла-повідомлення, який можна приховати в зображенні за допомогою програми S-Tools?
9. На підставі чого можна зробити припущення, що в програмі S-Tools застосовано стеганографічний метод заміни найменшого значущого біта?
10. Обґрунтуйте доцільність попереднього шифрування даних із застосуванням стеганографічних методів.
11. На стегозображенні провели невелику лінію (за допомогою програми Paint). Чи можна відновити після цього приховані дані?

Список літератури до розділу 4

1. Кузнецов О. О. Стеганографія : навч. посіб. / О. О. Кузнецов, С. П. Євсеєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2011. – 232 с.
2. Журавель І. М. Оцінка спотворень модифікованих контейнерів у стеганографії з використанням технологій обробки цифрових зображень / І. М. Журавель // Вісник Національного університету "Львівська політехніка" "Автоматика, вимірювання та керування". – Львів, 2015. – № 821. – С. 119–122.
3. Моденова О.В. Стеганография и стегоанализ в видеофайлах // О. В. Моденова / Прикладная дискретная математика. Приложение. – 2010. – №. 3. – С. 37–39.
4. Грибунин В. Г. Цифровая стеганография / В. Г. Грибунин, И. Н. Оков, И. В. Туринцев. – М. : Солон-Пресс, 2009. – 265 с.
5. Юдін О. К. Захист інформації в мережах передачі даних : підручник / О. К. Юдін, Г. Ф. Конахович, О. Г. Корченко. – К. : Видавництво ТОВ НВП "ІНТЕРСЕРВІС", 2009. – 716 с.
6. Конахович Г. Ф. Компьютерная стеганография. Теория и практика / Г. Ф. Конахович, А. Ю. Пузыренко. – К. : МК-Пресс, 2006. – 288 с.

Розділ 5

ІДЕНТИФІКАЦІЯ. АВТЕНТИФІКАЦІЯ. САНКЦІОНОВАНИЙ ДОСТУП

Розвиток мережевих технологій, електронної торгівлі з використанням мережі Інтернет, електронного документообігу та подальша інформатизація всіх областей діяльності суспільства призводять до того, що послуги із забезпечення цілісності та автентичності інформації сьогодні виходять на передові позиції. У цьому розділі наведено матеріал, що стосується питань ідентифікації, автентифкації та санкціонованого доступу – ключових процесів взаємодії інформаційних систем.

У розділі наведено структурну схему захищеної інформаційної системи, визначено задачі, які розв'язують за допомогою цих процесів. Описано класифікацію систем автентифікації за степенем стійкості. Розглянуто методи захисту цілісності даних та MAC-коди, методи та механізми їх утворення за допомогою ключових хеш-функцій та хеш-функцій, побудованих на основі алгоритмів блокового шифрування та з використанням потокового шифру. Частина матеріалу розділу присвячена протоколам автентифікації, їх стандартним варіантам. На основі стандартних стратегій розглянуто переваги та недоліки дійсних протоколів автентифікації. Наведені типові атаки на протоколи автентифікації. Як практичний протокол автентифікації розглянуто протокол "Kerberos".

5.1. Автентифікація

5.1.1. Основні визначення (термінологія)

У нашій (україномовній) літературі використовують два типи написання терміна – "аВтентифікація" та "аУтентифікація", які насправді описують одне й те саме поняття. Ураховуючи появу нормативного документа технічного захисту інформації "Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу" (НД ТЗІ 1.1-003-99) [1], де стандартизовано поняття у формі слова "автентифікація", будемо використо-

вувати його згідно із цим документом. Документ розглядає термінологію захисту інформації в комп'ютерних системах, проте узагальнимо її на інформаційні системи (ІС).

З урахуванням цього документа вважатимемо, що ***автентифікація*** (authentication) – процедура перевіряння відповідності пред'явленого ідентифікатора об'єкта інформаційної системи (ІС) на предмет належності його цьому об'єкту; установлення або підтвердження автентичності. І відповідно ***ідентифікація*** (identification) – процедура присвоєння ідентифікатора об'єкту ІС або встановлення відповідності між об'єктом і його ідентифікатором – впізнання.

5.1.2. Ідентифікація та автентифікація об'єктів

Якщо розглядати процедуру початку взаємодії об'єкта з інформаційною системою (далі ІС), яка містить інформацію, захищену від несанкціонованого доступу, то в ній можна виділити такі операції, що виконує система захисту цієї ІС: це ідентифікація, автентифікація та авторизація об'єкта (рис. 5.1).

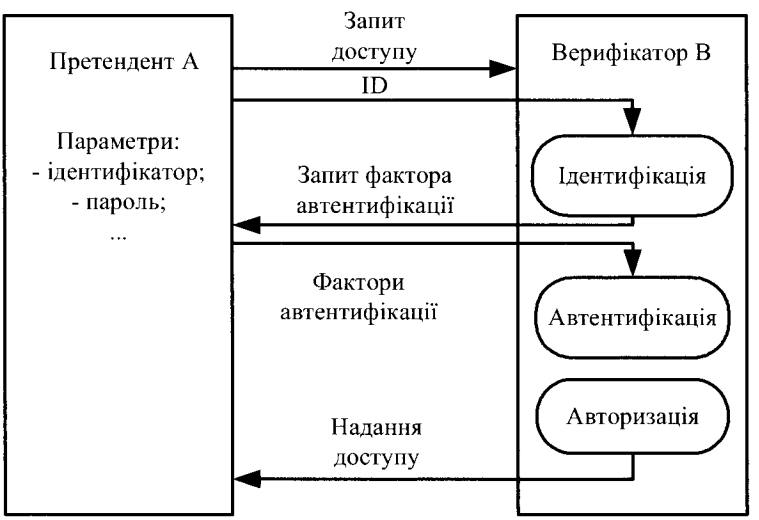


Рис. 5.1. Процеси взаємодії об'єкта з ІС

Ідентифікацію об'єкта виконує насамперед система захисту ІС. Перш ніж отримати доступ до ІС, об'єкт повинен назвати (ідентифікувати) себе. Для цього в загальному випадку ІС запитує в об'єкта його ім'я та його ідентифікаційний номер (ідентифікатор). Ім'я об'єкта використовують у ІС із метою ведення діалогу з об'єктом (імена різних об'єктів можуть збігатися).

Ідентифікатор об'єкта є унікальним, його використовують як позначення цього об'єкта. ІС перевіряє, чи є вказаний ідентифікатор зареєстрованим. Якщо це так, то ідентифікацію вважають успішною, і ІС ставить у відповідність цьому об'єкту всі його ідентифікаційні дані в повному обсязі. Після цього ІС переходить до виконання наступних захисних операцій, пов'язаних із цим об'єктом. Якщо ж об'єкт виявляється нелегальним, то залежно від політики захисту в ІС цьому об'єкту може бути відмовлено в доступі або запропоновано перейти до процедури реєстрації.

Автентифікацію об'єкта система захисту ІС виконує з метою встановлення, чи є цей об'єкт дійсно тим, за кого він себе видає. Перевірити істинність об'єкта найпростішим способом ІС може, запитуючи в нього фактор автентифікації (пароль, ключ, біометричні дані тощо). Проте використання лише одного фактора автентифікації не є безпечним; для систем із підвищеним захистом використовують два й більше факторів (багатофакторна автентифікація). У звичайній ситуації, коли є впевненість, що ІС є саме тією, за кого себе видає, об'єкт просто повідомляє їй свій фактор автентифікації. Після цього ІС, порівнявши отримане значення з еталонним, переконується в правах об'єкта.

Але в ситуації, коли є невпевненість у повноваженнях ІС, такі дії об'єкта неприпустимі, оскільки ІС, якщо її насправді створив зловмисник, може скомпрометувати цей фактор автентифікації. Але процедури автентифікації потребують не лише об'єкти, що перебувають у процесі обміну інформацією, а й сама інформація, яку передають (повідомлення). Передавати її можна як у просторі, так і в часі, тобто існує задача перевіряння незмінності інформації. Також існує й задача автентифікації мережі, якою передають інформацію.

Авторизацію об'єкта виконує система захисту ІС із метою встановлення допустимих дій для об'єкта в ІС, а також призначення доступних йому ресурсів ІС. Саме тому цю операцію називають також наданням повноважень об'єктові. Усі три розглянуті операції становлять разом процедуру ***ініціалізації*** й стосуються якогось єдиного об'єкта ІС, тобто мають односторонній характер. Але в багатьох випадках виявляється потрібним взаємне, двостороннє встановлення істинності об'єктів, що зв'язуються між собою каналом передавання даних.

Аудит об'єкта здійснюють після авторизації з метою отримання оперативної інформації щодо легітимності виконуваних об'єктом дій. В аудиті може бути налаштована сигналізація системи при настанні певних подій, що можуть свідчити про порушення політики безпеки або про невідповідність налаштованої політики безпеки заданій меті тощо. Аудит дає можливість проаналізувати поточний стан безпеки системи й відповідно його покращення.

У наведеної вище схеми односторонньої автентифікації є такі вади:

– об'єкт, що автентифікує ІС, не може бути впевненим в автентичності самої ІС, тобто він може вести діалог із фальшивою ІС, завдяки чому зловмисники отримають усі дані об'єкта, необхідні їм для авторизації на справжній ІС із правами цього об'єкта;

– фактор автентифікації передають відкритим каналом, тому він може бути перехоплений, модифікований; у зв'язку із цим в умовах повної недовіри виникає потреба у взаємній автентифікації.

5.1.3. Системи захисту цілісності даних

Вразливість відкритих мереж полягає в тому, що всі повідомлення можуть проходити через зловмисника, який може їх перехоплювати, передавати, модифікувати, фальсифікувати або спотворювати. Якщо зловмисник модифікує або фальсифікує повідомлення, він прагне запевнити отримувача в тому, що їх надіслав законний користувач. Для того щоб захистити відкриті системи зв'язку й зробити їх придатними для бізнесу та електронної комерції, недостатньо застосовувати криптографічні механізми, призначені для збереження конфіденційності повідомлень (тобто для запобігання їх перехопленню). Для цього необхідні засоби, що дадуть отримувачу змогу переконатися в тому, що повідомлення надіслано із законного джерела й не було змінене в процесі передавання. Саме для цього призначені ***методи захисту цілісності даних*** (data integrity), що запобігають несанкціонованій модифікації повідомлень.

Цілісність даних у сучасній криптографії тісно пов'язана із класичним поняттям теорії зв'язку – ***кодом контролю помилок*** (error-detection code), що є процедурою для виявлення помилок, які могли виникнути внаслідок неякісного зв'язку. Вважають, що шкоду від використання повідомлень, які містять помилки внаслідок недосконалості засобів зв'язку, можна порівняти зі шкодою, що виникає у разі використання навмисно спотвореної інформації. Із цієї причини принципи захисту цілісності даних і методи розпізнавання помилок, по суті, збігаються: відправник повідомлення приєднує до нього "контрольне значення", а отримувач обробляє це значення за певними правилами, узгодженими з відправником.

Мабуть, найпростішим прикладом методу розпізнавання помилок є ***циклічний надлишковий код*** (Cyclic Redundancy Code – CRC), що є значенням, яке за певним алгоритмом розраховують для деякого блока даних. CRC використовують в інформаційних системах (наприклад, контроль жорстких дисків, архіватори).

У кодах із контролем помилок надлишкову інформацію кодують так, щоб отримувач міг розпізнати можливі помилки, застосувавши детектор, який працює за методом максимальної правдоподібності.

Розглянемо приклад системи передавання даних від відправника А до отримувача В із використанням циклічного надлишкового коду. Дані можна передавати як у просторі, так і в часі. На рис. 5.2 наведено приклад передавання даних і перевіряння їх цілісності із використанням цього методу.

Відправник А перед передаванням визначає CRC-код повідомлення й у вигляді певної добавки передає її разом із цим повідомленням отримувачу В. Отримувач В, отримавши повідомлення й добавку до нього, повторно вираховує CRC-код і, у випадку збігу розрахованого й отриманого приймає рішення щодо цілісності повідомлення. Як бачимо, CRC-код не може забезпечити захисту цілісності даних за двома причинами:

– CRC-код розроблено для виявлення випадкових помилок, але в жодному випадку не зловмисних;

– зловмисник може легко повторно згенерувати CRC-код і передати його разом зі зміненим повідомленням отримувачу В.

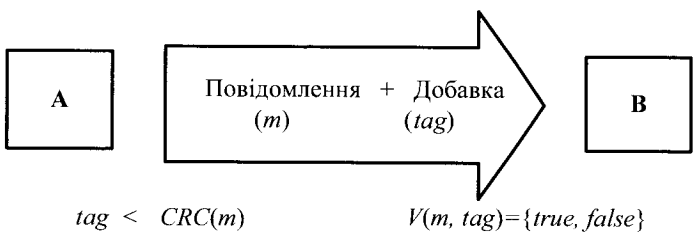


Рис. 5.2. Використання циклічного надлишкового коду в процесі передавання даних

У системах захисту цілісності даних надлишкову інформацію кодують так, щоб додане контрольне значення було якомога рівномірніше розподіленим по всьому простору повідомлень, а ймовірність його навмисного спотворення була мінімальною. Криптографічне перетворення, яке при цьому використовують, дуже схоже на перемішування під час шифрування, хоча перемішування під час шифрування ніяк не пов'язано з додаванням надлишкової інформації, призначеної для верифікації (перевіряння) повідомлень.

Подібно до алгоритмів шифрування, криптографічні перетворення, що призначені для забезпечення цілісності даних, параметризують ключами. Отже, результат верифікації цілісності даних надає перевіряючому інформацію про джерело повідомлення, тобто про користувача, який захистив ці дані. У процесі захисту цілісності даних можна виокремити такі етапи [2]:

– створення за допомогою деякої криптографічної функції f надлишкової інформації до даних tag, яку також у літературних джерелах [2] називають ***MDC*** (manipulation data cod), вона унікально характеризує їх і параметризується ключем автентифікації K_a: $tag \leftarrow f(data, K_a)$;

– перевіряння відповідності надлишкової інформації tag даним, що перевіряють: $Ver = V(data, tag, Kv) = \{1, 0\}$.

Отже, з деякою ймовірністю отримаємо відповідь про цілісність даних.

Схему системи захисту цілісності даних в умовах повної недовіри наведено на рис. 5.3.

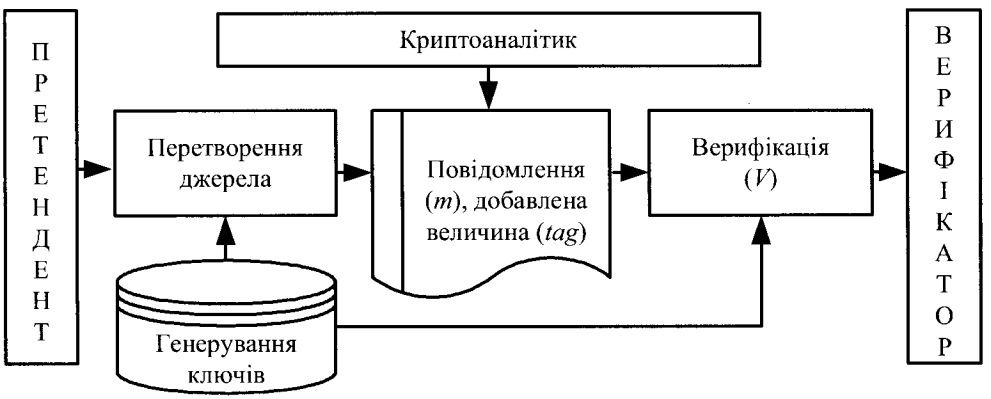


Рис. 5.3. Система захисту цілісності даних

Розглядаючи систему, наведену на рис. 5.3, можна виокремити такі загрози, які може здійснити відправник [3]:

– відправник формує та передає деяке повідомлення M_i у бік отримувача, а потім відмовляється від факту передавання M_i;

– відправник не передавав у бік отримувача повідомлення, але потім стверджує, що передав M_i повідомлення;

– відправник формує в момент часу t_i деяке повідомлення M_i і передає його отримувачу, а потім стверджує, що передав його в інший час t_j.

– відправник формує й передає отримувачу повідомлення M_i, а потім стверджує, що передавав інше повідомлення M_j.

Відповідно, загрози, які може здійснити отримувач:

– отримання від відправника повідомлення M_i і відмова від факту його отримання;

– отримувач формує деяке хибне повідомлення M_j, а потім стверджує, що отримав його від відправника;

– отримувач приймає повідомлення M_i у час t_i , а стверджує, що отримав в час t_j;

– отримувач отримує деяке повідомлення M_i, модифікує його до вигляду $M`_i$ та стверджує, що отримав $M`_i$.

Отже, відправник та отримувач можуть виступати як злочинці. Тому основним завданням автентифікації є захист користувачів від їх взаємного обману.

І, нарешті, загрози, які може здійснити криптоаналітик (зловмисник):

– модифікувати повідомлення;

– створювати хибні повідомлення;

– нав'язувати раніше передані повідомлення;

– формувати та передавати команди керування.

5.1.4. Задачі автентифікації

За всіх переваг сучасних систем шифрування самі по собі вони не забезпечують автентифікації даних. Тому засоби автентифікації необхідно використовувати в комплексі із криптографічними алгоритмами.

Сьогодні загальновизнаними вважають ***три задачі автентифікації***:

– автентифікація даних (перевіряння того, що масив даних не змінено протягом часу, коли він був поза контролем);

– автентифікація повідомлення (встановлення достовірності повідомлення, яке один абонент надсилає іншому відкритим каналом зв'язку);

– автентифікація користувача або в загальнішому випадку – сутності (встановлення достовірності користувача / сутності, яким потрібний доступ до захищеної інформації або які хочуть налагодити обмін інформації мережею).

Важливими є задачі забезпечення захисту від обману, оцінювання якості захисту від обману, оцінювання втрат, які можуть бути нанесені в результаті обману.

Для кращого розуміння принципів, які покладено в основу систем автентифікації, розглянемо особливості використання симетричного шифрування з метою автентифікації [4].

Випадок, коли розшифруванням шифрованого повідомлення отримують осмислений відкритий текст, доводить отримувачу, що повідомлення не було змінено чи підроблено (зазвичай, мають на увазі стійкий надійний алгоритм криптування й повну довіру між двома користувачами). Тобто за стійкого шифрування користувач, який не має секретного ключа, не може скласти бажаного тексту. З цього погляду шифрування забезпечує не лише конфіденційність повідомлень, але й їхню автентифікацію. Однак це твердження потребує

уточнення. Річ у тім, що отримувач повинен мати надійний критерій відкритого тексту. Якщо шифрують довільну (випадкову) послідовність знаків, то без використання додаткових засобів неможливо розпізнати відкритий текст у пункті призначення автоматично. Цю проблему вирішують введенням певної надлишковості таким чином, щоби лише порівняно невелику підмножину всіх можливих послідовностей знаків розглядали як множину відкритих текстів. Тоді вибрана випадкова послідовність знаків збігається із шифрованим текстом деякого повідомлення лише із близькою до нуля ймовірністю. Це не дає можливості зловмиснику сформувати підробку.

Одним зі способів виділення множини відкритих текстів є таке їх структурування, щоб відповідну структуру можна було легко розпізнавати, але не можна було відтворити без знання секретного ключа.

Спосіб, за якого спочатку розраховують контрольну суму, а після цього виконують шифрування, називають ***внутрішнім контролем помилок***. А спосіб, за якого спочатку здійснюють шифрування, а після цього додають контрольну суму, розраховану за шифртекстом, називають ***зовнішнім контролем помилок***. У цьому випадку зловмисник отримує можливість створювати повідомлення із правильними кодами розпізнавання помилок. І хоча при цьому йому не вдасться отримати відкритий текст повідомлення, він зможе створювати помилкові або спотворені повідомлення, оскільки спосіб вирахування контрольної суми йому відомий.

5.2. Класифікація систем автентифікації за ступенем стійкості

За ступенем стійкості системи автентифікації поділяють на обчислювально стійкі, доказово стійкі та безумовно стійкі. Якщо успіх активних атак визначають складністю деякого алгоритму, то кажуть, що така система автентифікації є ***обчислювально стійкою***. Мають на увазі, що алгоритм існує, але має надто велику часову складність. Наприклад, для систем автентифікації, що використовують MAC-коди на базі DES шифрування, для криптоаналізу методом повного перебору необхідно перебрати 2^{56} ключів.

Кажуть, що система автентифікації є ***доказово стійкою***, якщо можна довести, що успішне нав'язування зловмисником помилкової інформації еквівалентне вирішенню деякої відомої своєю складністю задачі, наприклад, таких, як розкладання на множники великого цілого числа або обчислення дискретного логарифму в скінченному полі $GF(q)$, за правильного вибору q.

Різниця між обчислювально та доказово стійкими системами не значна. В обох випадках стійкість залежить від обчислювальної складності вирішення

деякої задачі. Різниця полягає лише в тому, що в першому випадку існують підстави вірити, а в другому відомо, що досягнення обману еквівалентне вирішенню деякої складної задачі. Тому схеми цих двох типів часто об'єднують в одну групу.

Кажуть, що система автентифікації є ***безумовно стійкою***, якщо вона не залежить від обчислювальних ресурсів або часу, які може мати зловмисник. Це означає, що зловмиснику не залишається нічого іншого, як випадково вибрати повідомлення в надії, що воно буде проінтерпретовано отримувачем як істинне, незалежно від стратегії обману.

5.2.1. Поняття безумовно безпечних кодів автентифікації

Жодна система автентифікації не може дати повної гарантії, що отримане повідомлення прийшло саме від санкціонованого відправника. Завжди є якась мала ймовірність того, що згенеровану послідовність створив не справжній кореспондент, а зловмисник (криптоаналітик). Розглянемо питання ймовірності успішного обману за відсутності припущень щодо обчислюваної стійкості, тобто розглянемо питання безумовної безпеки або абсолютної стійкості згідно з [5].

Розглянено приклад передавання інформації від відправника А отримувачу В, припустивши їхню повну довіру один до одного та наявність спільного ключа. Розглянемо приклад.

Приклад. Відправник А хоче передати 1 біт інформації отримувачу В ("так" або "ні") у вигляді 2-бітового слова. Відправнику А і отримувачу В доступні 4 можливі ключі, причому вони використовують матрицю (табл. 5.1). Так, при застосуванні 3-го ключа повідомлення 1 буде надіслане як 11. Імовірність того, що хтось інший може видати себе за відправника А, дорівнює 1/2, оскільки відносно загального секретного ключа відправника й отримувача лише 2 з 4-х слів 00, 01, 11, 10 можна насправді передати.

Таблиця 5.1

Матриця шифрування

Ключ/код	00	01	10	11
1	0	1	–	–
2	1	–	0	–
3	–	0	–	1
4	–	–	1	0

Супротивник (криптоаналітик), намагаючись підмінити передане повідомлення іншим знає, що можуть бути використані лише 2 ключі, але не

знає, який саме. Наприклад, якщо криптоаналітик перехоплює 01, то він знає, що було надіслане повіомлення 1 (відносно ключа 1) або повідомлення 0 (відносно ключа 3). У першому випадку він повинен передати 00, а в другому – 11, отже його успіх має ймовірність 1/2.

Ця схема забезпечує також і секретність, тому що будь-яке передане слово може породжуватись як повідомленням 0, так і повідомленням 1 (обидві події мають імовірність 1/2).

Загальне означення коду автентифікації в термінах теорії безумовної безпеки виглядає так.

Означення. ***Код автентифікації*** – це трійка (M, K, C) разом із відображенням

$f : M \times K \to C$, таким, що для всіх m, $m \in M$ і всіх $k \in K$

$$f(m) = f(m') \to m = m', \tag{5.1}$$

де M – множина повідомлень, K – множина ключів; C – множина кодових слів.

Код автентифікації можна записати у вигляді матриці, у якій стрічки індексовані k з K, стовпці – кодовими словами c з C, а в клітинці з індексом (k,c) стоїть таке $m \in m'$, що $f_k(m) = c$ (якщо воно існує, то згідно зі співвідношенням (5.1) воно єдине), якщо ні, то стоїть прочерк. Таку таблицю називають матрицею автентифікації цього коду.

Так, у наведеному прикладі $M=\{0, 1\}$, $K=\{1, 2, 3, 4\}$, $C=\{00, 01, 10, 11\}$, а матрицю автентифікації задано таблицею.

Сімонс (Gustavus J. Simmons) у своїй теорії [3, 6] грунтується на схемі (рис. 5.3), ураховуючи, що для захисту від обману формують ключ автентифікації, що є відомим відправнику та отримувачу:

– відправник може формувати $M_i, i = 1,..., n_M$;

– M_i відображають у захищене повідомлення $`C_i, i = 1,..., n_M$.

При цьому і криптоаналітик, і відправник, а також і отримувач можуть здійснити загрози, розглянуті нижче.

Відправник може здійснити такі загрози:

– відправник формує повідомлення M_i, а потім відмовляється від самого факту його передавання;

– відправник стверджує, що він сформував деяку інформацію M_i і передав її, хоча в дійсності він її не передавав;

– відправник стверджує, що він сформував і передав деяку інформацію M_i у певний час, хоча в дійсності він її передав у інший час.

– відправник формує й передає інформацію M'_i і при цьому стверджує, що сформував і передав інформацію M_i.

Отримувач відповідно може здійснити такі загрози:

– отримувач сам формує деяку інформацію M'_i, а потім стверджує що він її отримав від відправника;

– отримувач отримує інформацію M_i від відправника, модифікує її в M'_i, а потім стверджує, що саме її він отримав від відправника;

– отримувач стверджує, що отримав інформацію M_i від відправника в момент часу, який відрізняється від часу, коли було насправді сформовано й передано цю інформацію;

– отримувач отримує інформацію M_i і стверджує, що її не отримував.

І нарешті – основні загрози, що може здійснити криптоаналітик:

– криптоаналітик створює інформацію M'_i і передає її отримувачу в момент часу, коли відправник перебуває в пасивному стані (атака імітації);

– криптоаналітик перехоплює інформацію M_i, модифікує її в M'_i і передає отримувачу (атака підміни);

– криптоаналітик передає повторно інформацію M_i отримувачу в момент часу, коли відправник пасивний;

– криптоаналітик передає хибні команди керування мережевими службами, помилкові команди керування ключами, підміняє сертифікати.

На підставі розглянутого можливо зробити наступні висновки.

1. Очевидно, що якщо криптоаналітик сформує одне з повідомлень та введе його в систему, то ймовірність обману можна оцінити як:

$$P_{обм} = \frac{n_M}{n_C}, \tag{5.2}$$

де n_C, n_M – розміри множин криптограм і повідомлень.

Якщо $n_M = n_C$, то криптоаналітик нав'яже повідомлення з імовірністю $P_{обм}=1$.

2. Для забезпечення захисту від обману треба збільшити n_C.

3. У такій системі $P_{обм} \neq 0$. Ідеальних систем захисту від обману не існує.

4. Без надлишковості забезпечити захист від обману неможливо.

Криптоаналітик може використовувати інформацію ΔI, яка є різницею між ентропією джерела шифртексту та умовною ентропією джерела шифртексту після використання ключів.

$$\Delta I = H(C) - H\left(C/K\right), \tag{5.3}$$

$$H(C) = -\sum_{i=1}^{n_C} P(C_i)\log P(C_i), \tag{5.4}$$

$$\log_2 P_{обм} \geq -\Delta I, \tag{5.5}$$

$$P_{обм} \geq 2^{-\Delta I(C,K)}, \tag{5.6}$$

де ΔI – це кількість інформації, яку використано для захисту від обману.

Співвідношення (5.6) у теорії Сімонса визначає нижню межу ймовірності обману в системі. Згідно з (5.6) що більше значення ΔI, то захищеність від обману є кращою. Тобто що менше $H\left(C/K\right)$, то рівень захищеності вищий. При цьому ключ автентифікації повинен бути конфіденційним. Мінімальної ймовірності $P_{обм}$ досягають у випадку, коли $P_{обм} = 2^{-\Delta I(C,K)}$. Криптосистему, в якій це забезпечують, називають ***досконалою***. За співвідношенням (5.6) також з'ясовано, що в жодній системі $P_{обм}$ не може дорівнювати 0, і в дійсних системах забезпечують грубе оцінювання обману.

При створенні дійсних криптосистем імітозахисту, як правило:

$$\Delta I(C,K) \Rightarrow l_{ka}, \tag{5.7}$$

де l_{ka} – бітова довжина ключа автентифікації, тоді

$$P_{обм} \geq 2^{-l_{ka}}, \tag{5.8}$$

$$P_{обм} \geq 2^{-l_{ім}}, \tag{5.9}$$

де $l_{ім}$ – бітова довжина повідомлення,

$$P_{обм} \geq 2^{-l_{ЕЦП}}, \tag{5.10}$$

де $l_{ЕЦП}$ – довжина ключа повідомлення.

У процесі розроблення або оцінювання систем автентифікації потрібно враховувати ступінь різних загроз. Оцінку ступеня впливу загрози криптоаналітиком можна виразити як максимальну з-поміж інших:

$$P_{обм} = \left\{\max\ P_i, P_n, P_{рп}, P_{ку}\right\}, \tag{5.11}$$

де P_i – імовірність імітації; P_n – імовірність підміни; $P_{рп}$ – імовірність раніше переданого повідомлення; $P_{ку}$ – імовірність команд управління.

5.3. Криптографічні хеш-функції

Як і криптосистеми, механізми захисту цілісності даних бувають симетричними й асиметричними. Проте слід підкреслити різницю між ними, що виявляється при використанні відкритого ключа. У криптосистемах, основаних на асиметричних методах, відкритий і закритий ключі мають фіксоване застосування: відкритий ключ призначений для шифрування, а закритий – для розшифрування. У системах захисту цілісності даних, що використовують асиметричні методи, відкритий і закритий ключі можна застосовувати як для шифрування, так і для розшифрування.

У симетричних методах захисту цілісності даних криптографічні перетворення створення автентифікаційного коду f і його перевірка V є симетричними криптографічними алгоритмами, тобто $f = V$ і $Ke = Kv$. Інакше кажучи, створення й перевіряння відповідності між даними *Data* і кодом MDC забезпечують одними й тими самими криптографічними операціями.

Завдяки тісному зв'язку між цілісністю даних і автентифікацією повідомлень код MDC, створений за допомогою симетричного криптографічного методу, часто називають ***кодом автентифікації повідомлень*** (message authentication code – MAC). Такий код можна створювати й верифікувати за допомогою або ключової хеш-функції, або алгоритму блокового шифрування.

Загальний спосіб здійснення MAC-коду полягає в застосуванні ***ключової хеш-функції*** (keyed hash function). Введемо поняття криптографічної функції хешування [2].

Функція хешування – це детермінована функція, що відображає рядок бітів довільної довжини в хешоване значення, що є рядком бітів фіксованої довжини.

Формально для виразу хеш-функції можна записати:

$$h: \{0, 1\}^* \times \{0, 1\}^{\ell} \rightarrow \{0, 1\}^{n}.$$

Позначимо через h хеш-функцію, що повертає рядок фіксованої довжини. Ця функція h повинна мати такі властивості:

– ***перетворення перемішування;*** при будь-якому аргументі хешоване значення не повинно з обчислювального погляду відрізнятися від рядка бітів, рівномірно розподілених в інтервалі $[0,2^{|n|}]$;

– ***запобігання колізіям;*** пошук двох величин x та y, що задовольняють умови $x \neq y$ і $h(x)=h(y)$, повинен бути нерозв'язним завданням; для того, щоб це припущення було обґрунтованим, необхідно, щоб область значень функції h була великою; число $|n|$ не повинно бути меншим за 128 і, як правило, дорівнювало 160;

– **складність обчислення прообразів;** завдання обчислення вхідного рядка x за заданим хешованим значенням h – $h(x)$ має бути нерозв'язним обчислювальним завданням. У цьому випадку також необхідно, щоб область значень функції h була достатньо великою;

– **практична ефективність;** обчислення значення $h(x)$ має обмежуватися поліномом невеликого степеня (в ідеальному випадку – лінійним), залежним від розміру рядка x.

Властивості перемішування й запобігання колізіям можна забезпечити за допомогою тих самих функцій, які використовують в алгоритмах блокового шифрування.

Складності обчислення прообразів досягають завдяки стисненню даних, за якого втрачають частину початкової інформації, й, отже, важко обчислити зворотну функцію. Розглянемо приклади реальних хеш-функцій, що використовують у системах автентифікації.

5.3.1. Алгоритм MD5

Сімейство алгоритмів ***MD*** (message digest algorithm) розробив ***Рональд Лін Рівест*** (Ronald L. Rivest). Алгоритм ***MD2***, розроблений у 1989 р. Він орієнтований на 8-розрядні процесори й відрізняється від від *MD4* і *MD5* (орієнтованих на 32-розрядні процесори) значенням стартового вектора хешування (нульового вектора) і використанням 256-байтової перестановки, а оброблення 2-го блока виконують у ньому за 18 циклів.

В алгоритмі *MD4*, розробленому в 1990 р., один блок обробляють за 3 цикли, кожен з яких містить 16 операцій. Сьогодні у багатьох системах використовують ***MD5***. Входом алгоритму є повідомлення довільної довжини. Виходом є коротке повідомлення (дайджест, хеш) довжиною 128 бітів. Загальна схема алгоритму наведена на рис. 5.4.

Поширення повідомлення. Повідомлення завдовжки K бітів доповнюють так, щоб його довжина L у бітах стала конгруентною 448 mod 512 (L = 448 mod 512). Якщо довжина повідомлення вже така, то все одно додають 512 біти. Інформація, що додають, складається з одиниці та нулів за нею (тобто 100...0). Додавати можуть від 1 до 512 біти.

Додавання довжини. До повідомлення додають довжину вихідного повідомлення (число K) як 64-бітове число. Якщо довжина вихідного повідомлення більша за 2^{64}, то додають число K mod 2^{64}.

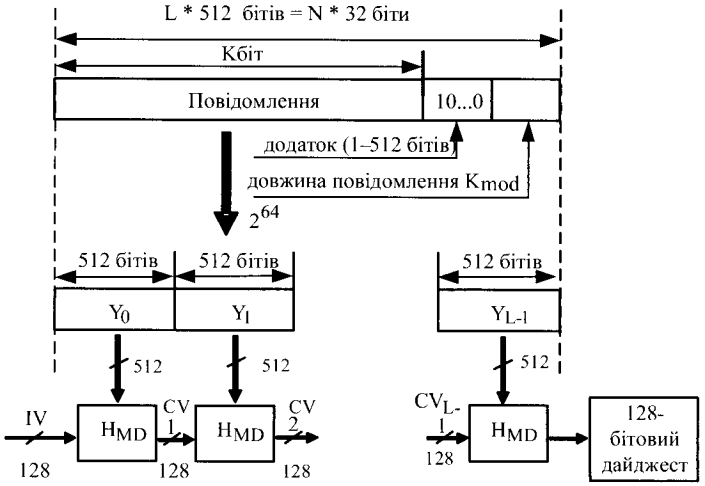


Рис. 5.4. Генерація короткого повідомлення (дайджеста)

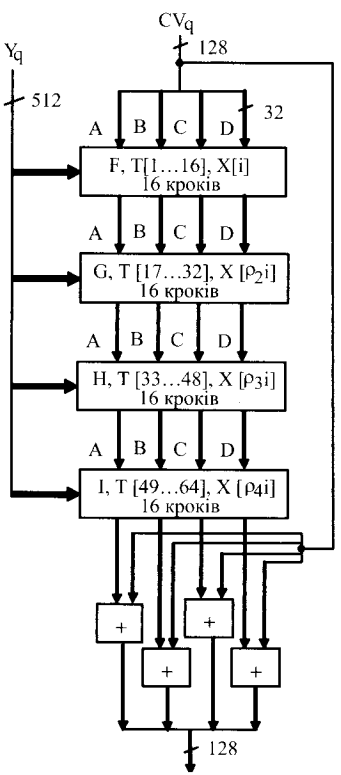


Рис. 5.5. Оброблення одного 512-бітового блока

Ініціалізація MD-буфера. 128-бітовий буфер використовують для зберігання проміжних та остаточних результатів хеш-функції. Буфер містить чотири 32-бітові регістри A, B, C, D. Регістри у вигляді вектора ініціалізації (IV) попередньо заповнюють такими значеннями:

A = 67452301,
B = EFCDAB89,
C = 98BADCFE,
D = 10325476.

І далі, після обробки в модулі $\mathbf{H_{MD}}$ 512 бітів інформації повідомлення, на виході блоку $\mathbf{H_{MD}}$ отримують наступний вектор $\mathbf{CV_1}$, який уже використовують як вектор ініціалізації для оброблення модулем $\mathbf{H_{MD}}$ наступного 512 бітового блока інформації.

Оброблення 512-бітового повідомлення. Структуру модуля $\mathbf{H_{MD5}}$ зображено на рис. 5.5. Функція стиснення H_{MD} складається із чотирьох раундів подібної структури, але кожний з них має свою власну логічну функцію – F, G, H та I (табл. 5.2). Значення цих функцій у залежності від параметрів b, c, d наведені в табл. 5.3.

Таблиця 5.2

Примітивні функції F, G, H, I

Раунд	Примітивна функція g	g(b, c, d)
1	F(b, c, d)	(b and c) or ((not b) and d)
2	G(b, c, d)	(b and d) or (c and (not d))
3	H(b, c, d)	B xor c xor d
4	I(b, c, d)	c xor (b or (not d))

Таблиця 5.3

Значення примітивних функцій F, G, H, I

b	c	d	F	G	H	I
0	0	0	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	1	1	0
0	1	1	1	0	0	1
1	0	0	0	0	1	1
1	0	1	0	1	0	1
1	1	0	1	1	0	0
1	1	1	1	1	1	0

Кожний раунд приймає на вхід 512-бітовий блок Y_q та 128-бітове значення буфера ABCD. Кожний раунд також використовує четверту частину 64-елементної таблиці T[1...64], яку побудовано за допомогою функції синуса (T[i] дорівнює цілій частині значення $2^{32} \cdot abs(\sin(i))$, де значення i задають у радіанах. Оскільки значення $abs(\sin(i))$ знаходиться у проміжку від 0 до 1, то кожний елемент таблиці T є 32-бітовим числом.

Результат четвертого раунду додають до входу першого раунду (CV_q), у результаті чого отримують CV_{q+1}. Додавання здійснюють за модулем 2^{32}.

Формування результату. Після оброблення всіх L 512-бітових блоків результатом алгоритму MD5 (128-бітовим коротким повідомленням або дайджестом) є вихід L-го блока.

Кожний раунд обробки одного 512-бітового блока складається з послідовності 16 кроків, кожен з яких є наступною операцією над ABCD-буфером:

$$a = b + ((a + g(b, c, d) + X[k] + T[i]) <<< s),$$

де <<< s – операція циклічного зсуву вліво на s бітів; CX[k] = M[q * 16 + k] – k-те 32-бітове слово в q-му 512-бітовому блоці вхідного повідомлення.

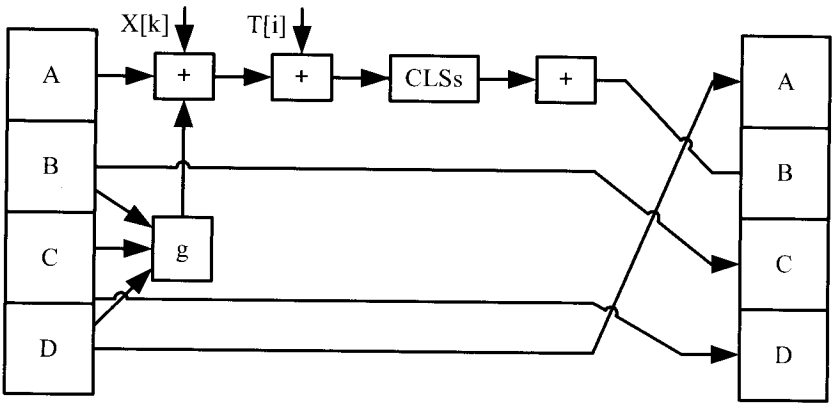


Рис. 5.6. Елементарна операція

У кожному раунді значення X[i] використовують лише один раз. До першого раунду біти надходять у тій самій послідовності, як вони стоять на вході. При надходженні бітів до 2, 3 та 4 раундів використовують такі перестановки:

ρ_2 (i) = (1 + 5i)mod 16,

ρ_3 (i) = (5 + 3i)mod 16,

ρ_4 (i) = 7i mod 16,

```
/* обробити кожне 16-бітове слово 512-бітового блока */
for q = 0 to (N/16) – 1 do
  /* скопіювати блок q у X */
  for j = 0 to 15 do
          X[j] = M[q * 16 + j]
  end /* циклу по j */

AA = A
BB = B
CC = C
DD = D

/* Раунд 1 */
/* [abcd k s i] визначає операцію a = b + ((a + F(b, c, d) + X[k] + T[i]) <<< s) */
[ABCD  0      7      1]
[DABC  1      12     2]
[CDAB  2      17     3]
[BCDA  3      22     4]
```

```
[ABCD  4     7     5]
[DABC  5     12    6]
[CDAB  6     17    7]
[BCDA  7     22    8]
[ABCD  8     7     9]
[DABC  9     12    10]
[CDAB  10    17    11]
[BCDA  11    22    12]
[ABCD  12    7     13]
[DABC  13    12    14]
[CDAB  14    17    15]
[BCDA  15    22    16]

/* Раунд 2 */
/* [abcd k s i] визначає операцію a = b + ((a +  G(b, c, d) + X[k] + T[i]) <<< s) */
[ABCD  1     5     17]
[DABC  6     9     18]
[CDAB  11    14    19]
[BCDA  0     20    20]
[ABCD  5     5     21]
[DABC  10    9     22]
[CDAB  15    14    23]
[BCDA  4     20    24]
[ABCD  9     5     25]
[DABC  14    9     26]
[CDAB  3     14    27]
[BCDA  8     20    28]
[ABCD  13    5     29]
[DABC  2     9     30]
[CDAB  7     14    31]
[BCDA  12    20    32]

/* Раунд 3 */
/* [abcd k s i] визначає операцію a = b + ((a +  H(b, c, d) + X[k] + T[i]) <<< s) */
[ABCD  5     4     33]
[DABC  8     11    34]
[CDAB  11    16    35]
[BCDA  14    23    36]
```

[ABCD 1 4 37]
[DABC 4 11 38]
[CDAB 7 16 39]
[BCDA 10 23 40]
[ABCD 13 4 41]
[DABC 0 11 42]
[CDAB 3 16 43]
[BCDA 6 23 44]
[ABCD 9 4 45]
[DABC 12 11 46]
[CDAB 15 16 47]
[BCDA 2 23 48]

/* Раунд 4 */
/* [abcd k s i] визначає операцію a = b + ((a + I(b, c, d) + X[k] + T[i]) <<< s) */
[ABCD 0 6 49]
[DABC 7 10 50]
[CDAB 14 15 51]
[BCDA 5 21 52]
[ABCD 12 6 53]
[DABC 3 10 54]
[CDAB 10 15 55]
[BCDA 1 21 56]
[ABCD 8 6 57]
[DABC 15 10 58]
[CDAB 6 15 59]
[BCDA 13 21 60]
[ABCD 4 6 61]
[DABC 11 10 62]
[CDAB 2 15 63]
[BCDA 9 21 64]

A = A + AA
B = B + BB
C = C + CC
D = D + DD
end /* циклу по q */

5.3.2. Алгоритм SHA

Національний інститут стандартів і технології США (National Institute of Standards and Technology – NIST), разом з ***Агентством національної безпеки США*** (National Security Agency – NSA) для ***стандарту цифрового підпису*** (Digital Signature Standard) розробив ***стандарт безпечного хешування*** (Secure Hash Standard – SHS), у якому використано ***алгоритм безпечного хешування*** (Secure Hash Algorithm – SHA). Цей алгоритм необхідний для забезпечення безпеки ***алгоритму цифрового підпису*** (Digital Signature Algorithm – DSA).

Для будь-якого вхідного повідомлення довжиною менше 2^{64} бітів SHA видає 160-бітовий результат, який називають ***коротким змістом повідомлення***. Далі, короткий зміст повідомлення стає входом DSA, який обчислює підпис для повідомлення. ***Підпис короткого змісту*** замість усього повідомлення часто підвищує ефективність процесу, оскільки короткий зміст повідомлення набагато менший, ніж саме повідомлення. Тож короткий зміст повідомлення повинен бути отриманий тим, хто перевіряє підпис, якщо прийняту ним версію повідомлення використовують як вхід SHA. SHA називають безпечним, оскільки він розроблений так, щоб було обчислювально неможливо знайти повідомлення, відповідне певному короткому змісту повідомлення, або знайти два різні повідомлення з однаковим коротким змістом повідомлення. Будь-які зміни, події при передаванні повідомлення з дуже високою ймовірністю спричинять зміну короткого змісту повідомлення, і підпис не пройде перевіряння. Принципи, що лежать в основі SHA, аналогічні використаним професором Рональдом Л. Рівестом при проектуванні алгоритму короткого змісту повідомлення MD4. SHA розроблений за зразком цього алгоритму.

SHA видає 160-бітове хеш-значення, довше, ніж у MD5. Повідомлення доповнюють, щоб воно було завдовжки кратним 512 бітам. Використовують те саме доповнення, що й у MD5: спочатку додають 1, а потім нулі так, щоб довжина отриманого повідомлення була на 64 біти меншого за число, кратне 512, а потім додають 64-бітове представлення довжини оригінального повідомлення.

Задають початкові значення п'яти 32-бітових регістрів:

A =0x67452301,

B = 0xefcdab89,

C = 0x98badcfe,

D =0x10325476,

E = 0xc3d2e1f0.

Потім починають головний цикл алгоритму. Він обробляє повідомлення 512-бітовими блоками, поки не будуть вичерпані всі блоки повідомлення.

Спочатку п'ять змінних копіюють в інші змінні: A в a, B в b, C в c, D в d і E в e.

Головний цикл складається із чотирьох етапів по 20 операцій у кожному (у MD5 чотири етапи по 16 операцій у кожному). Кожна операція є нелінійною функцією над трьома з a, b, c, d і e, а потім виконує зсув і додавання аналогічно до MD5. У SHA використовують такий набір нелінійних функцій:

$f_t(x,y,z) = (x \wedge y) \vee (\neg x \wedge z)$ для t=0 до 19,

$f_t(x,y,z) = (x \oplus y \oplus z)$ для t=20 до 39,

$f_t(x,y,z) = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$ для t=40 до 59,

$f_t(x,y,z) = (x \oplus y \oplus z)$ для t=60 до 79.

В алгоритмі використовують такі чотири константи:

Kt = 0x5a827999 для t=0 до 19,

Kt = 0x6ed9eba1 для t=20 до 39,

Kt = 0x8flbbcdc для t=40 до 59,

Kt = 0xca62c1d6 для t=60 до 79.

(Якщо цікаво, як отримано ці числа, то: 0x5a827999 = $\sqrt{2}/4$, 0x6ed9eba1 = = $\sqrt{3}/4$, 0x8flbbcdc = $\sqrt{5}/4$, 0xca62c1d6 = $\sqrt{10}/4$.)

Блок повідомлення перетворюють з 16 32-бітових слів (від M0 до M15) на 80 32-бітових слів (від W0 до W79) за допомогою такого алгоритму:

W = Mt, для t = 0 до 15

W = (Wt-3 ⊕ W-8 ⊕ Wt-14 ⊕ Wt-16) <<< 1, для t = 16 до 79.

Якщо t – це номер операції (від 1 до 80), Wt – t-ій підблок розширеного повідомлення, а <<<s – циклічний зсув ліворуч на s бітів, то головний цикл виглядає так:

FOR t = 0 to 79

TEMP = (a <<< 5) + f_t(b,c,d)+ e + Wt + K

e = d

d = c

c = b <<< 30

b = a

a = TEMP

На рис. 5.7 показано одну операцію. Зміщення змінних виконує ту саму функцію, яку в MD5 виконує використання в різних місцях різних змінних.

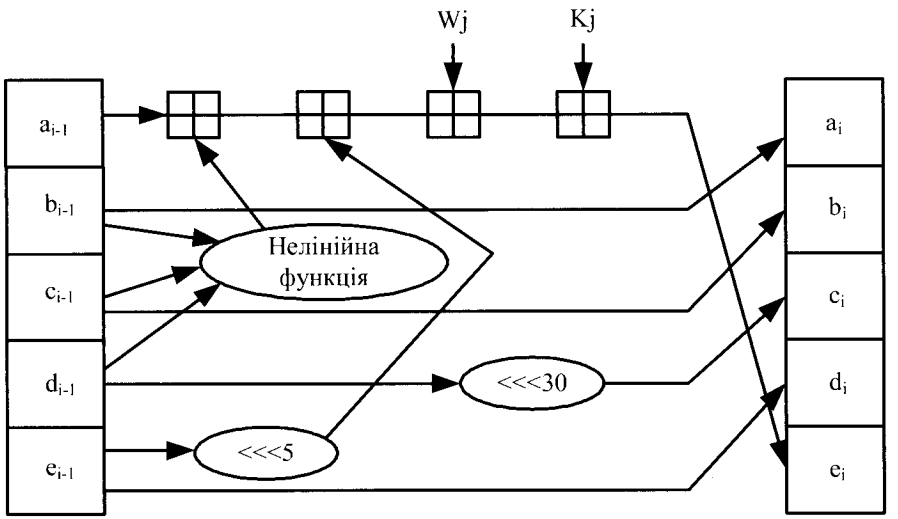


Рис. 5.7. Ітерація алгоритму SHA1

Після всього цього a, b, c, d і e додають до A, B, C, D і E відповідно, і алгоритм продовжують виконувати для наступного блоку даних. Остаточний результат отримують після об'єднання A, B, C, D і E.

5.3.3. Алгоритм SHA3

Серед п'яти фіналістів конкурсу NIST SHA-3 Competition, оголошеного у 2012 році, алгоритм JH не відповідав критерію стійкості до знаходження прообразу. Серед решти чотирьох алгоритмів Skein, Blake, Keccak і Grøstl переміг алгоритм Keccak. Крім того, алгоритм Keccak, так само як і Skein, виявився універсальним алгоритмом, але на відміну від Skein він не містить блокового шифру. Замість функції стиснення автори Keccak використали ***псевдовипадкові перестановки*** (Pseudo Random Permutation – PRP). Загальну схему алгоритму наведено на рис. 6.8. В основу алгоритму покладено ***конструкцію губки*** (spong) [7]. Він має дві фази: фазу ***всмоктування*** (absorbing) і фазу ***віджимання*** (squeesing). У фазі всмоктування відкриту інформацію, розбиту на блоки m_i, подають у блок стану розміром 1600 бітів, після цього блок m_0 за допомогою операції Xor додають до фрагмента блока стану розміром r. Частину, що залишилась, об'ємом c залишають не заповненою. Результат подають на вхід багатораундової безключової функції f, що здійснює псевдовипадкову перестановку. Операцію повторюють, поки не будуть вичерпані блоки інформації, які хешують. Після цього наступає фаза віджимання губки, під час якої отримують хеш довільної довжини. Перед

початком оброблення повідомлення його доповнюють до довжини, кратної *r*. Доповнення останнього блока здійснюють так:

– до повідомлення додають 1, після неї 0 або більше нульових бітів (до *r*-1), у кінці 1;

– може бути додано *r-1* бітів, коли останній блок має довжину *r-1* бітів; до цього блока додають одиницю, наступний блок міститиме *r-1* нулів і одиницю.

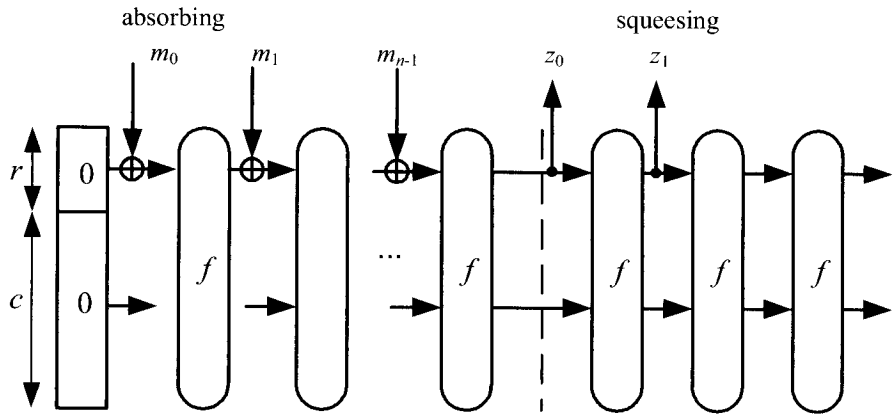


Рис. 5.8. Криптографічна конструкція "губки"

Функцію *f* будують на послідовному застосуванні перетворень Chi, Theta, Pi, Rho, Iota, які містять операції виняткове "або" (**Xor**), побітове "і" (AND), побітове "не" (NOT). Як блок обробки даних, або ***масив стану*** вибирають тривимірний масив A розміром 5 × 5 x *w*. Тоді елементом масиву A[*i*], [*j*], [*k*] є (5*i*+*j*) × *w*+*k* бітів рядка стану S. У табл. 5.4 наведено можливі значення масивів стану, де $w= b/25$, $l=log_2(b/25)$.

Таблиця 5.4

Можливі параметри масиву стану

b	25	50	100	200	400	800	1600
w	1	2	4	8	16	32	64
l	0	1	2	3	4	5	6

У цьому алгоритмі розрізняють такі фрагменти масиву стану: двовимірні – площина, шар, сторінка. Їхніми елементами є відповідні одновимірні масиви: рядок, колонка й лінія. Найменший елемент (0-вимірний) – біт.

Наведемо описи алгоритмів для перетворень, що використовують у функції *f*.

Алгоритм 1: θ(A) –"Theta".

Вхід: state array A.

Вихід: state array A′.

Крок 1. Для всіх пар (i,k), $0 \le i < 5$ і $0 \le k < w$,

$$C[i, k] = A[i, 0, k] \oplus A[i, 1, k] \oplus A[i, 2, k] \oplus A[i, 3, k] \oplus A[i, 4, k]$$

Крок 2. For all pairs such as (i,k), $0 \le i < 5$ і $0 \le k < w$,

$$D[i, k] = C[(i\text{-}1), \text{ mod } 5, k] \oplus C[(i+1) \text{ mod } 5, (k\text{-}1)\text{mod } w]$$

Крок 3. For all such as (i,k), $0 \le i < 5$ і $0 \le k < w$,

$$A'[i, j, k] = A[i, j, k] \oplus D[i, k]$$

Алгоритм 2: ρ(A) – "Rho".

Вхід: масив стану A.

Вихід: масив стану A′.

Крок 1. For all such as k, $0 \le k < w$, $A'[0, 0, k] = A[0, 0, k]$

Крок 2. $(i, j) = (1, 0)$.

Крок 3. For all t from 0 to 23

a. for all k such that $0 \le k < w$, $A'[i, j, k] = A[i, j,(k-(t+1)(t+2)/2)\text{mod } w]$;

b. set $(i,j) = (j,(2i=3j) \text{ mod } 5)/$

Крок 4. Return A'.

Алгоритм 3; π(A) – "Pi"

Вхід: стан масиву A.

Вихід: стан масиву A′.

Крок 1. For all triples (i, j, k) such that $0 \le i < 5$, $0 \le j < 5$, і $0 \le k < w$,

Set $A'[i, j, k] = A[(i+3j)\text{mod } 5, i, k]$.

Крок 2. Return A'.

Алгоритм 4: χ(A) – "Chi",

Вхід: state array A.

Вихід: state array A′.

Крок 1. For all triples (i, j, k) such that $0 \le i < 5$, $0 \le j < 5$, і $0 \le k < w$,

$$A'[i, j, k] = A[i, j, k] \oplus ((A[(i+1) \text{ mod } 5, j, k] \oplus 1) \cdot A[(i+2) \text{ mod } 5, j, k]).$$

Крок 2.Return A.

Алгоритм 5: rc(t).

Вхід: integer t.

Вихід: bit rc(t).

Крок 1. If t mod 255 = 0, return 1.

Крок 2. Let R = 10000000.

Крок 3. For i from 1 to t mod 255,

set: a. R = 0 || R;

b. R[0] = R[0] ⊕ R[8];

c. R[4] = R[4] ⊕ R[8];

d. R[5] = R[5] ⊕ R[8];

e. R[6] = R[6] ⊕ R[8];

f. R =Trunc8[R].

Крок 4. Return R[0].

Алгоритм 6: ι(A, i_r) – "Iota"

Вхід: state array A.

round index i_r.

Вихід: state array A′.

Крок 1. For all triples (i, j, k) such that $0 \leq i$.

Крок 2. Let RC=0 w.

Крок 3. For j from 0 to l, let RC[$2j$ –1]=rc(j+7ir).

Крок 4. For all z such that 0≤z.

Крок 5. Return A′.

5.3.4. Застосування функції хешування в криптографії

Хеш-функції дуже поширені в криптографії, зокрема в:

– алгоритмах цифрового підпису хеш-функції зазвичай використовують для створення "конспекту повідомлення", або "відбитку повідомлення"; це забезпечує деяку надлишковість повідомлення, підписаного так, що після хешування воно містить розпізнавану інформацію; стійкість схеми цифрового підпису до фальсифікацій сильно залежить від надмірної інформації, що міститься в підписаному повідомленні;

– прикладних криптосистемах із відкритим ключем хеш-функції широко використовують для здійснення механізму верифікації (перевіряння) правильності зашифрованих текстів; такий механізм необхідний для того, щоб забезпечити доказову стійкість до активних атак;

– широкому колі криптографічних додатків функції хешування використовують як псевдовипадкові функції; до таких застосувань належать узгодження ключів (наприклад, коли два користувачі використовують загальне початкове значення як аргумент функції хешування для набуття розділеного значення ключа), протоколи автентифікації (наприклад, коли два учасники

протоколу підтверджують виконання протоколу, обмінюючись хешованими значеннями), протоколи електронної комерції (наприклад, для накопичення платежів для біржової гри), протоколи доказу знання (наприклад, для забезпечення автономного режиму доказу).

5.3.5. Хеш-функції, що використовують симетричні блокові алгоритми

Як однонапрямлені хеш-функції можна використовувати симетричні блокові алгоритми шифрування. Ідея полягає в тому, що якщо блоковий алгоритм є безпечним, то й однонапрямлена хеш-функція буде безпечною.

Найочевиднішим способом є шифрування повідомлення в ***режимі зчеплення блоків*** (cipher-block chaining – CBC) ***або шифрувальної книги*** (elecrton codebook – ECB) за допомогою фіксованого ключа й ***вектора ініціалізації*** (initialization vector – IV), хеш-значенням буде останній блок шифртексту. Цей спосіб не дуже підходить для однонаправлених хеш-функцій, хоча він працюватиме для MAC-кодів.

Кращим способом є використання як ключа блоку повідомлення, попереднє хеш-значення як вхід, а поточне хеш-значення є виходом. Дійсні хеш-функції ще складніші. Розмір блока зазвичай збігається з довжиною ключа, і розміром хеш-значення буде довжина блока. Оскільки більшість блокових алгоритмів 64-бітові, спроектовано низку схем, що дають хеш-значення, вдвічі більше від довжини блока.

За умови, що хеш-функція правильна, безпека цієї схеми основана на безпеці використовуваної блокової функції. Проте є й винятки. Диференційний криптоаналіз краще працює проти блокових функцій у хеш-функціях, ніж проти блокових функцій, що використовують для шифрування: ключ відомий, тому можна використовувати різні прийоми. Для успіху потрібна тільки одна правильна пара, і можна генерувати стільки вибраного відкритого тексту, скільки потрібно.

Нижче наведено огляд різних хеш-функцій, описаних у літературі.

Висновки про можливість успішної атаки припускають, що використовуваний блоковий алгоритм безпечний, і кращою атакою є атака грубою силою.

Корисною мірою для хеш-функцій, основаних на блокових шифрах, є швидкість хешування, або кількість n-бітових блоків повідомлення (n – це розмір блока алгоритму), що обробляють при шифруванні. Що вища швидкість хешування, то швидший алгоритм.

Розглянемо схеми, в яких довжина хеш-значення дорівнює довжині блока.

Ось загальна схема (рис. 5.9):

$$H_0 = IH,$$

де IH – випадкове початкове значення,

$$H = E_A(B) \oplus C,$$

де A, B і C можуть бути або M_i, H_{i-1}, ($M_i \oplus H_{i-1}$, або константи (можливо, дорівнюють 0), $E_A(B)$ – функція блокового шифрування інформації B за допомогою ключа A. H_0 – це деяке випадкове початкове число IH. Повідомлення розбивають на частини відповідно до розміру блока M_i, що обробляють окремо. Крім того, використовують варіант MD-підсилення, можлива та сама процедура доповнення, що й у MD5 і SHA.

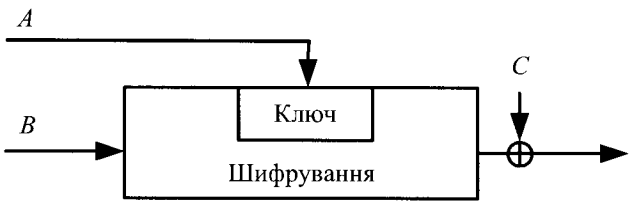


Рис. 5.9. Схема формування хеш-послідовності на основі блокового шифру

Розглядаючи цю схему й ураховуючи те, що три різні змінні можуть набувати одного з 4-х можливих значень, отримуємо 64 можливі варіанти схем цього типу. Усі вони були проаналізовані, і в результаті з них було виділено 12 безпечних схем:

1. $H_i = E_{Hi-1}(M_i) \oplus M_i$.
2. $H_i = E_{Hi-1}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1}$.
3. $H_i = E_{Hi-1}(M_i) \oplus H_{i-1} \oplus M_i$.
4. $H_i = E_{Hi-1}(M_i \oplus H_{i-1}) \oplus M_i$.
5. $H_i = E_{Mi}(H_{i-1}) \oplus H_{i-11}$.
6. $H_i = E_{Mi}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-11}$.
7. $H_i = E_{Mi}(H_{i-1}) \oplus M_i \oplus H_{i-1}$.
8. $H_i = E_{Mi}(M_i \oplus H_{i-1}) \oplus H_{i-1}$.
9. $H_i = E_{Mi \oplus Hi-1}(M_i) \oplus M_i$.
10. $H_i = E_{Mi \oplus Hi-1}(H_{i-1}) \oplus H_{i-1}$.
11. $H_i = E_{Mi \oplus Hi-1}(M_i) \oplus H_{i-1}$.
12. $H_i = E_{Mi \oplus Hi-1}(H_{i-1}) \oplus M_i$.

Перші чотири – стійкі до всіх атак, решта вісім стійкі до всіх типів атак, окрім атаки з фіксованою точкою (метод Полларда).

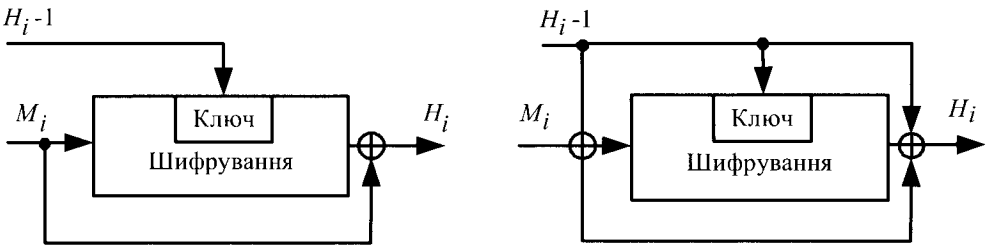


Рис. 5.10. Приклад зображення функцій формування хеш-значень на основі функцій $H_i = E_{Hi-1}(M_i) \oplus M_i$, $H_i = E_{Hi-1}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1}$

Модифікація схеми Девіса – Маєра. ***Лей*** (Lai) і ***Мессі*** (Massey) модифікували метод ***Девіса – Маєра*** (Davies – Meyer) (5-та схема), щоби можна було використовувати шифр IDEA. IDEA використовує 64-бітовий блок і 128-бітовий ключ. На рис. 5.11 зображена запропонована ними схема.

$H_0 = I_H$, де I_H – випадкове початкове значення, $H_i = E_{Hi-1,Mi}(H_{i-1})$.

Ця функція хешує повідомлення 64-бітовими блоками й видає 64-бітове значення.

Якоїсь іншої атаки цієї схеми, ніж метод грубої сили, немає.

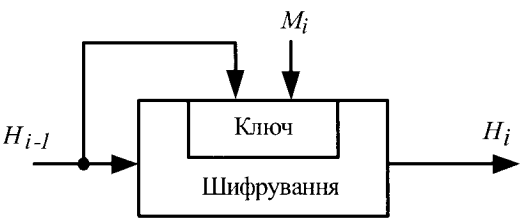


Рис. 5.11. Схема Лея–Мессі

Прикладом збільшення значення хеш-функції на основі блокових шифрів з 64-бітовим ключем є використання блокового IDEA-подібного алгоритму з 64-бітовим блоком і 128-бітовим ключем.

5.3.6. Хеш-функція ГОСТ

Ця хеш-функція з'явилася в Росії й визначена в стандарті ГОСТ Р 34.11. 94. Також її використовували в Україні згідно з міждержавним стандартом ГОСТ 34.311-95. У ній застосовують блоковий алгоритм ГОСТ (див. розділ 3), хоча теоретично можна використовувати будь-який блоковий алгоритм з 64-бітовим блоком і 256-бітовим ключем. Функція видає 256-бітове хеш-значенння.

Функцію стиснення, $H_i = f(M_i, H_i\text{-}1)$ (обидва операнди – 256-бітові величини) – визначають так:

1) за допомогою лінійного змішування M_i, H_i-1 і деяких констант генерують чотири ключі шифрування ГОСТ;

2) кожен ключ використовують для шифрування відмінних 64 бітів H_i-1 у режимі ECB. Отримані 256 бітів зберігають у тимчасовій змінній S;

3) H_i є складною, хоча й лінійною функцією S, M_i і H_i-1.

Хеш-значення останнього блока повідомлення не є його остаточним хеш-значенням. Насправді використовують три змінні зчеплення: Hn – це хеш-значення останнього блока; Z – це XOR усіх блоків повідомлення, а L – довжина повідомлення. З використанням цих змінних і доповненого останнього блоку M' остаточне хеш-значення дорівнює:

$$H = f(Z \oplus M', f(L, f(M', H_n))).$$

5.3.7. Функція хешування "Купина" – національний стандарт України ДСТУ 7564:2014

На зміну міждержавному стандарту ГОСТ 34.311-95, який використовували в Україні майже двадцять років (російський ГОСТ Р 34.11-94), у 2014 році прийнято новий стандарт ДСТУ 7624:2014, що базується на криптографічній функції хешування "Купина". У цьому стандарті передбачено додатковий режим її застосування для формування коду автентифікації повідомлення (імітовставки). У стандарті також наведено значення для перевіряння реалізацій.

Опис стандарту наведемо згідно з публікаціями авторів хеш-функції "Купина" [8]. Цей стандарт визначає дві функції хешування: "Купина-256" та "Купина-512", які забезпечують обчислення хеш-значень із довжинами 256 та 512 біти відповідно. Ці значення можна скоротити до величин від 8 до 248 бітів у першому та від 504 до 264 біти в другому випадку із кроком у 8 бітів. Такі режими отримання хеш-значень позначають як "Купина-n".

Основними рекомендованими режимами роботи функції хешування, є режими "Купина-256", "Купина-384" і "Купина-512". Усі вони можуть формувати хеш-значення інформації завдовжки від 0 до $2^{96} - 1$ біт. Перед початком формування хеш-значення інформацію поділяють на блоки m_1, m_2, ..., m_t завдовжки l бітів кожний. Розмір блока l становить 512 і 1024 біти для режимів "Купина-256" та "Купина-512" відповідно. Останній блок доповнюють до l бітів так. Повідомлення доповнюють також у випадку, коли воно кратне l. До повідомлення додають допоміжну інформацію, яка містить одиничний біт, певну кількість нульових бітів та довжину повідомлення, яке обробляє функція хешування так, щоб доповнена бітова послідовність була завдовжки, кратною розміру внутрішнього стану l. Кількість нульових бітів можна визначити з формули $w = (-N\text{-}97) \bmod l$. Після цього додають 96 бітів, в яких міститься значення довжини повідомлення N, записаного у форматі *little endian*, тобто найменш значущі байти мають менший номер.

Потім кожний блок ітеративно обробляють функцією стиснення φ за формулою $h_i = \varphi(h_{i-1}, m_i)$ де $i=1,...,t$ а $h_0 = IV$ – початкове значення (вектор ініціалізації). Структуру функції хешування "Купина" зображено на рис. 5.12.

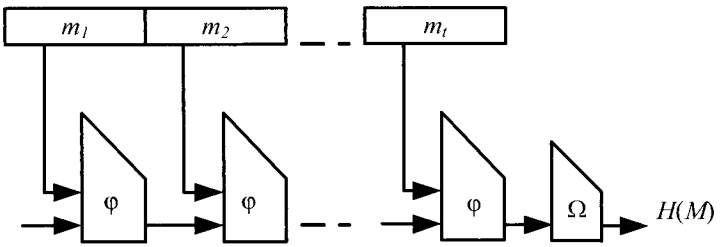


Рис. 5.12. Структура функції хешування "Купина"

Залежність між розміром внутрішнього стану l, кількістю ітерацій r, кількістю стовпців внутрішнього стану c у матричному поданні та значенням векторів ініціалізації від розміру хеш-значення n наведено в табл. 5.5.

Таблиця 5.5

Розмір хеш-значення n	Розмір внутрішнього стану l	Кількість ітерацій r	Кількість стовпців у матриці c	Вектор ініціалізації (IV)
$8 \le n \le 256$	512	10	8	0x4000…00
$256 \le n \le 512$	1024	14	16	0x8000…00

Функція стиснення φ складається з перетворень l-бітового блока P і Q. Її визначають так: $\varphi(h,m) = P(h \oplus m) \oplus Q(m) \oplus h$.

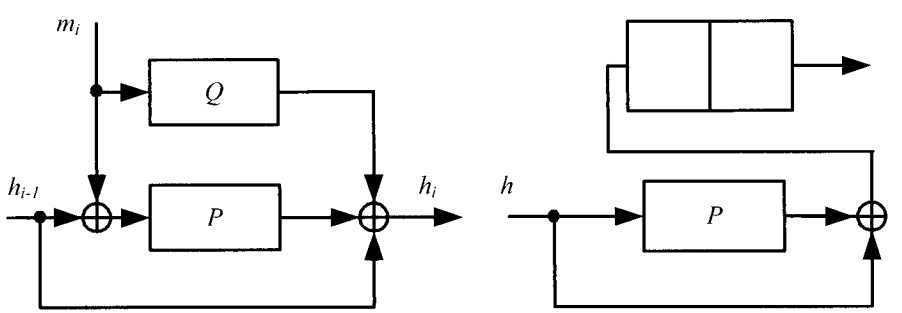


Рис. 5.13. Схема функції стиснення φ та завершальне перетворення Ω

Завершальне перетворення Ω задане формулою:

$$\Omega(x) = trunc_n\left(P(h) \oplus h\right),$$

де функція $trunc_n\left(x\right)$ відсікає всі біти аргументу h, крім останніх (старших) n бітів.

Перед виконанням перетворень P і Q вхідну послідовність представляють як внутрішній стан функції хешування завдовжки l бітів ($l = 512$ або $l = 1024$ залежно від розміру внутрішнього стану). Після завершення виконання перетворень P і Q внутрішній стан знову трансформують у послідовність байтів, яку подають на вхід наступної ітерації функції стиснення φ або на завершальне перетворення для формування остаточного хеш-значення. У режимі хешування "Купина-256" вхідну послідовність байтів позначають як $in_0, in_1, \ldots, in_{63}$. Остаточну послідовність байтів після оброблення позначають як $out_0, out_1, \ldots, out_{63}$. Процес заповнення внутрішнього 512-бітового стану S-функції хешування перед початком перетворень та зворотний процес після їх завершення наведено в табл. 5.6–5.8. Відображення 1024-бітової вхідної послідовності для режиму "Купина-512" у внутрішній стан функції хешування є аналогічним.

Таблиця 5.6

Вхідна послідовність байтів

Вхідна послідовність							
In_0	In_8	In_{16}	In_{24}	In_{32}	In_{40}	In_{48}	In_{56}
In_1	In_9	In_{17}	In_{25}	In_{33}	In_{41}	In_{49}	In_{57}
In_2	In_{10}	In_{18}	In_{26}	In_{34}	In_{42}	In_{50}	In_{58}
In_3	In_{11}	In_{19}	In_{27}	In_{35}	In_{43}	In_{51}	In_{59}
In_4	In_{12}	In_{20}	In_{28}	In_{36}	In_{44}	In_{52}	In_{60}
In_5	In_{13}	In_{21}	In_{29}	In_{37}	In_{45}	In_{53}	In_{61}
In_6	In_{14}	In_{22}	In_{30}	In_{38}	In_{46}	In_{54}	In_{62}
In_7	In_{15}	In_{23}	In_{31}	In_{39}	In_{47}	In_{55}	In_{53}

Таблиця 5.7

Внутрішній стан функції хешування

Внутрішній стан функції хешування							
$S_{0,0}$	S_{012}	$S_{0,2}$	$S_{0,3}$	$S_{0,4}$	$S_{0,5}$	$S_{0,6}$	$S_{0,7}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,4}$	$S_{1,5}$	$S_{1,6}$	$S_{1,7}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$	$S_{2,4}$	$S_{2,5}$	$S_{2,6}$	$S_{2,7}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$	$S_{3,4}$	$S_{3,5}$	$S_{3,6}$	$S_{3,7}$
$S_{4,0}$	$S_{4,1}$	$S_{4,2}$	$S_{4,3}$	$S_{4,4}$	$S_{4,5}$	$S_{4,6}$	$S_{4,7}$
$S_{5,0}$	$S_{5,1}$	$S_{5,2}$	$S_{5,3}$	$S_{5,4}$	$S_{5,5}$	$S_{5,6}$	$S_{5,7}$
$S_{6,0}$	$S_{6,1}$	$S_{6,2}$	$S_{6,3}$	$S_{6,4}$	$S_{6,5}$	$S_{6,6}$	$S_{6,7}$
$S_{7,0}$	$S_{7,1}$	$S_{7,2}$	$S_{7,3}$	$S_{7,4}$	$S_{7,5}$	$S_{7,6}$	$S_{7,7}$

Таблиця 5.8

Вихідна послідовність байтів

Вихідна послідовність							
Out_0	Out_8	Out_{16}	Out_{24}	Out_{32}	Out_{40}	Out_{48}	Out_{56}
Out_1	Out_9	Out_{17}	Out_{25}	Out_{33}	Out_{41}	Out_{49}	Out_{57}
Out_2	Out_{10}	Out_{18}	Out_{26}	Out_{34}	Out_{42}	Out_{50}	Out_{58}
Out_3	Out_{11}	Out_{19}	Out_{27}	Out_{35}	Out_{43}	Out_{51}	Out_{59}
Out_4	Out_{12}	Out_{20}	Out_{28}	Out_{36}	Out_{44}	Out_{52}	Out_{60}
Out_5	Out_{13}	Out_{21}	Out_{29}	Out_{37}	Out_{45}	Out_{53}	Out_{61}
Out_6	Out_{14}	Out_{22}	Out_{30}	Out_{38}	Out_{46}	Out_{54}	Out_{62}
Out_7	Out_{15}	Out_{23}	Out_{31}	Out_{39}	Out_{47}	Out_{55}	Out_{53}

Під час обчислення хеш-значення внутрішній стан функції стиснення має розмір 512 або 1024 бітів. Для кожного варіанта розміру внутрішнього стану використовують власну пару перетворень: P512, Q512 або P1024, Q1024.

Перетворення і є варіантами блокового шифру, в яких замість циклових ключів використовують визначені константи (див. нижче). Кількість циклів r залежить від розміру внутрішнього стану та наведена в табл. 5.6. Згідно з [8] перетворення P і Q можуть бути виражені програмно так (рис. 5.15):

```
void P_Hash(byte in[ 8 * c ], byte out[ 8 * c ] ) {
      byte state[ 8, c ] = in
      for(round = 0 to r-1 step 1) {
            XORRoundKey( state, roundconstP[ round ] )
            Kalyna_S_boxes( state )
            KupynaShiftRows( state )
            MixColumns( state )
      }
      Out = state
}
void Q_Hash(byte in[ 8 * c ], byte out[ 8 * c ]) {
      byte state[ 8, c ] = in
      for(round = 0 to r-1 step 1) {
            Add64RoundKey( state, roundconstQ[ round ] )
            Kalyna_S_boxes( state )
            KupynaShiftRows( state )
            MixColumns( state )
      }
out = state
}
```

Операції ***XORRoundKey***, ***Add64RoundKey***, ***Kalyna_S_boxes***, ***MixColumns*** відповідають аналогічним операціям блокового шифру "Калина" і визначені в ДСТУ 7624:2014 (відповідно як $K_l^{(K_v)}$, $\eta_l^{(K_v)}$, π_l', ψ_l).

Циклові константи roundconstP[round] і roundconstQ[round] (що формують на основі значень $\omega_j^{(v)}$ та $\varsigma_j^{(v)}$ з ДСТУ 7564:2014) залежно від номера циклу перетворення round (i) та розміру внутрішнього стану (l) задають такими матрицями (у шістнадцятковому вигляді):

$$P_{512}:C^i = \begin{bmatrix} 00\oplus i & 10\oplus i & 20\oplus i & 30\oplus i & 40\oplus i & 50\oplus i & 60\oplus i & 70\oplus i \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \end{bmatrix},$$

$$Q_{512}:C^i = \begin{bmatrix} f3 & f3 & f3 & f3 & f3 & f3 & f3 & f3 \\ f0 & f0 & f0 & f0 & f0 & f0 & f0 & f0 \\ f0 & f0 & f0 & f0 & f0 & f0 & f0 & f0 \\ f0 & f0 & f0 & f0 & f0 & f0 & f0 & f0 \\ f0 & f0 & f0 & f0 & f0 & f0 & f0 & f0 \\ f0 & f0 & f0 & f0 & f0 & f0 & f0 & f0 \\ f0 & f0 & f0 & f0 & f0 & f0 & f0 & f0 \\ 70\oplus i & 60\oplus i & 50\oplus i & 40\oplus i & 30\oplus i & 20\oplus i & 10\oplus i & 00\oplus i \end{bmatrix},$$

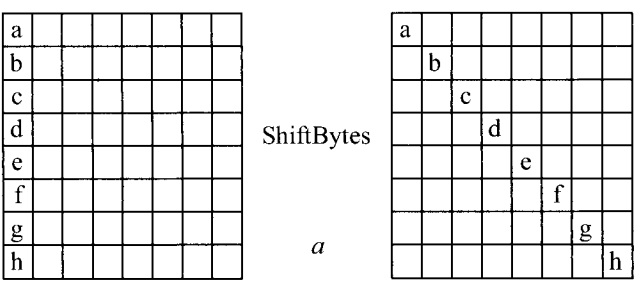
$$P_{1024}:C^i = \begin{bmatrix} 00\oplus i & 10\oplus i & 20\oplus i & 30\oplus i & 40\oplus i & 50\oplus i & \dots & f0\oplus i \\ 00 & 00 & 00 & 00 & 00 & 00 & \dots & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & \dots & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & \dots & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & \dots & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & \dots & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & \dots & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & \dots & 00 \end{bmatrix},$$

$$Q_{1024}: C^i = \begin{bmatrix} f3 & f3 & f3 & f3 & f3 & f3 & \dots & f3 \\ f0 & f0 & f0 & f0 & f0 & f0 & \dots & f0 \\ f0 & f0 & f0 & f0 & f0 & f0 & \dots & f0 \\ f0 & f0 & f0 & f0 & f0 & f0 & \dots & f0 \\ f0 & f0 & f0 & f0 & f0 & f0 & \dots & f0 \\ f0 & f0 & f0 & f0 & f0 & f0 & \dots & f0 \\ f0 & f0 & f0 & f0 & f0 & f0 & \dots & f0 \\ f0 \oplus i & e0 \oplus i & d0 \oplus i & c0 \oplus i & b0 \oplus i & a0 \oplus i & \dots & 00 \oplus i \end{bmatrix}.$$

Таблиця 5.9

Значення циклічних зсувів рядків залежно від розміру внутрішнього стану

	Значення зсуву байтів	
	Внутрішній стан 512 біти	Внутрішній стан 1024 біти
0	0	0
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	11



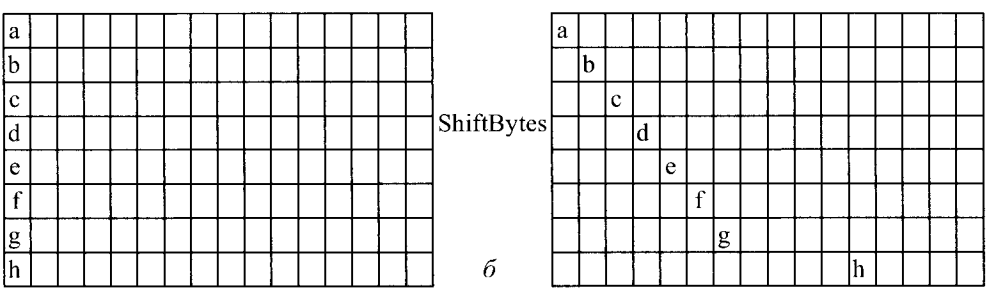


Рис. 5.14. Схема перетворення КирупаShiftRows для: а – 512-бітового та б – 1024-бітового внутрішніх станів

Перетворення ***KupynaShiftRows*** (функція в ДСТУ 7564:2014) виконує розподілення байтів кожного 64-бітового стовпця серед інших стовпців циклічним зсувом рядків внутрішнього стану праворуч на різну кількість байтів. Значення зсувів залежать від розміру внутрішнього стану функції хешування та наведені в табл. 5.9.

Схему перетворення ***KupynaShiftRows*** для різних розмірів внутрішнього стану наведено на рис. 5.14.

Описану хеш-функцію розроблено за консервативним підходом із залученням відомих і добре досліджених конструкцій на основі нового блокового шифру "Калина" (ДСТУ 7624:2014).

5.3.8. Коди автентифікації повідомлень, що використовують функції хешування із ключем

Криптографічні функції хешування є природним засобом для захисту цілісності даних. У сценарії з розподіленими ключами на вхід хеш-функції надходить ключ. Решта вхідної інформації складається з повідомлення, яке необхідно автентифікувати. Так, щоб автентифікувати повідомлення M, відправник обчислює значення:

$$\text{MAC} = h\,(k \,\|\, M),$$

де k – секретний ключ, розподілений між відправником і отримувачем, а символ $\|$ позначає операцію зчеплення бітових рядків.

Враховуючи властивості функції хешування, перераховані вище, можна припустити, що для створення правильного MAC-коду за допомогою хеш-функції, отриманої як аргументи ключ k і повідомлення M, користувач повинен володіти правильними ключем і повідомленням. Отримувач, який володіє загальним із відправником ключем, повинен наново обчислити MAC-код на основі отриманого повідомлення M і перевірити, чи відповідає він отриманому MAC-коду. Якщо ці коди збігаються, повідомлення надійшло від законного відправника. Оскільки такі MAC-коди обчислюють за допомогою функції хешування, їх називають ***кодами HMAC*** (hash message authentication code). Доволі часто коди HMAC обчислюють за формулою:

$$\text{HMAC} = h(k \,\|\, M \,\|\, k),$$

тобто ключ приєднують до повідомлення M як префікс і суфікс. Це не дає змоги супротивникові змоги використовувати циклічну структуру деяких функцій хешування. Якщо повідомлення не захищене ключем з обох кінців, така структура деяких функцій хешування дає змогу супротивникові, який навіть не знає ключа, модифікувати повідомлення за допомогою довільного префікса або суфікса.

Як правило, функції хешування із ключем утворюють за допомогою алгоритмів блокового шифрування в режимі зчеплення блоків зашифрованого тексту. Функцію хешування із ключем, побудована таким чином, називають функцією MAC.

Нехай $E_k(m)$ – алгоритм блокового шифрування повідомлення M із ключем k. Для того щоб автентифікувати повідомлення M, відправник спочатку розбиває його на блоки:

$$M = m_1, m_2 ... m_l,$$

де розмір кожного блока m_i (i = 1, 2, ..., l) дорівнює розміру вхідних даних алгоритму блокового шифрування. Якщо розмір останнього блока ml не дорівнює розміру повного блока, його доповнюють випадковою величиною.

Нехай $C_0 = IV$ – випадковий вектор ініціалізації. Відправник застосовує алгоритм блокового шифрування в режимі зчеплення блоків зашифрованого тексту:

$$C_i \leftarrow E_k(m_i \oplus C_{i-1}),\ i = 1,2,.., l.$$

Тепер пару (IV, Ci) можна використовувати як код автентифікації повідомлень, який приєднують до повідомлення M перед відправленням.

Очевидно, що обчислення коду автентифікації повідомлень у режимі зчеплення блоків зашифрованого тексту CBC-MAC передбачає необоротне стиснення даних (по суті, кодом CBC-MAC є "конспект повідомлення"). Із цієї причини перетворення CBC-MAC є однонапрямленим. Більше того, властивість перемішування відповідного алгоритму блокового шифрування надасть цьому перетворенню характеру хешування (тобто розподіляє код MAC по всьому простору кодів так само рівномірно, як початковий алгоритм блокового шифрування розподіляє повідомлення по всьому простору зашифрованих текстів). Отже, можна припустити, що для створення коректного коду CBC-MAC користувач дійсно повинен володіти ключем k відповідного алгоритму блокового шифрування. Отримувач, який використовує разом із відправником ключ, повинен наново обчислити код MAC на основі отриманого повідомлення й перевірити, чи збігається він з отриманим варіантом. Якщо збігається, то повідомлення надіслав законний відправник.

Код автентифікації повідомлень, що забезпечує цілісність даних у повідомленні M для користувачів, які володіють ключем k, іноді позначають як MAC(k, M). У цьому позначенні ігнорують деталі здійснення, наприклад, однонапрямлене перетворення, використане під час обчислення коду.

Коди MAC можуть бути використані для перевіряння достовірності файлів, якими обмінюються користувачі. Також їх може використати один користувач для перевіряння, чи не змінилися його файли, скажімо, через вірус.

Користувач може обчислити MAC його файлів і зберегти ці значення в таблиці. Якщо користувач скористається замість MAC однонапрямленою хеш-функцією, то вірус може обчислити нові хеш-значення після зараження файлів і замінити елементи таблиці. З MAC вірус не зможе цього добитися, оскільки ключ вірусу невідомий. Простим способом перетворення однонапрямленої хеш-функції на MAC є шифрування хеш-значення симетричним алгоритмом. Будь-який MAC можна перетворити на однонапрямлену хеш-функцію за допомогою розкриття ключа.

5.3.9. CBC-MAC

Простий спосіб створення залежної від ключа однонапрямленої хеш-функції – шифрування повідомлення блоковим алгоритмом у режимах CBC або ***CFB*** (cipher feedback). Хеш-значенням є останній блок, зашифрований у режимах CBC або CFB. Потенційна проблема, пов'язана з безпекою цього методу, полягає в тому, що отримувач повинен знати ключ, і цей ключ дає змогу йому генерувати повідомлення з тим самим хеш-значенням, що й у присланого повідомлення, за допомогою дешифрування в зворотному напрямі.

Алгоритм перевіряння достовірності повідомлення (Message Authenticator Algorithm – MAA). Цей алгоритм входить у стандарти ISO [19–21]. Він видає 32-бітове хеш-значення й спроектований для мейнфреймів зі швидкими інструкціями множення.

$v = v \lll 1$

$e = v \oplus w$

$X = ((((e + y) \bmod 2^{32}) \vee A \wedge C) * (x \oplus Mi)) \bmod 2^{32}\text{-}1$

$Y = ((((e + x) \bmod 2^{32}) \vee B \wedge D) * (y \oplus M)) \bmod 2^{32}\text{-}1$

Ці дії повторюють для кожного блока повідомлення, *Mi*, і остаточне хеш-значення отримують за допомогою XOR *X* і *Y*. Змінні v і e залежать від ключа. *A, B, C* і *D* є константами.

5.4. Протоколи автентифікації

Якщо визначати поняття ***автентифікації*** більш стисло, то згідно з [5] його можна визначити як процедуру, що дає змогу одній сутності перевірити оголошені властивості іншої. Тобто автентифікація дає першій сутності змогу перевірити, чи є друга сутність законною, а другій сутності – довести свою законність. У цьому випадку другу сторону називають претендентом, а першу – перевірником. Це в односторонній автентифікації. У випадку взаємної

автентифікації й перша, і друга сторони набувають одночасно обох якостей: і претендента/пред'явника, й перевірника. Процес спілкування двох сторін під час їх взаємної або односторонньої автентифікації називають ***протоколом автентифікації***.

Існують різні схеми класифікації протоколів автентифікації, але насамперед, мабуть, слід зазначити існування протоколів, які залучають третю сторону, яку називають ***третьою довірчою стороною*** (ТДС), і таких, які обходяться без неї.

У протоколах автентифікації можна виділити три важливі процеси: автентифікацію даних, які передають, автентифікацію сутностей і генерацію автентифікаційних ключів. Перший процес автентифікації означає перевіряння оголошеної властивості повідомлення, другий передбачає доведення властивостей відправника повідомлення (претендента), а третій призначений для організації захищеного каналу для обміну секретними повідомленнями.

5.4.1. Автентифікація джерела даних

Автентифікація джерела даних (що також називають "автентифікацією повідомлення" (message authentication)) тісно пов'язана із захистом цілісності даних. Проте насправді автентифікація джерела даних і захист цілісності даних є дещо різними поняттями.

На відміну від інформації, що захищають лише методом захисту цілісності даних за допомогою вже нам відомих MDC або MAC-кодів, у повідомленні, що передають, необхідно захистити й перевірити ще й інші його властивості, а саме: приналежність цього повідомлення відправнику, свіжість повідомлення (тобто відправник у цей момент передав саме це повідомлення, а не створене ним раніше). На цій можливості ґрунтується атака "повтор".

Отже, автентифікація джерела даних передбачає такі дії:

– передавання повідомлення від відправника до отримувача, який перевіряє достовірність повідомлення перед його ухваленням;

– ідентифікація відправника повідомлення;

– перевіряння цілісності даних, отриманих від відправника;

– перевіряння справжності відправника повідомлення.

5.4.2. Автентифікація сутності

Автентифікація сутності – це процес обміну інформацією (тобто протокол), під час якого користувач установлює ***достовірність*** (lively correspondence) іншого користувача. Як правило, у ході протоколу автентифікації

з'ясовують і достовірність повідомлення. У таких ситуаціях, щоб переконатися в достовірності повідомлення та його автора, слід скористатися механізмом автентифікації джерела даних. Існує декілька сценаріїв автентифікації сутності в розподілених системах. Наведемо деякі з них.

Обмін повідомленнями між двома головними комп'ютерами (host-host type). Учасниками протоколу є комп'ютери, названі вузлами або платформами розподіленої системи. Робота комп'ютерів, як правило, повинна бути узгодженою. Наприклад, якщо одна з віддалених платформ "перезавантажується", вона повинна ідентифікувати достовірний сервер і передати йому необхідну інформацію, наприклад, достовірну копію операційної системи, достовірні установки таймера або достовірні установки оточення. Достовірність інформації зазвичай визначають за допомогою протоколу автентифікації. Як правило, протокол обміну повідомленнями між двома комп'ютерами є системою клієнт-сервер, в якій один із комп'ютерів (клієнт) обслуговується іншим комп'ютером (сервером).

Обмін повідомленнями між користувачем і головним комп'ютером (userhost type). Користувач отримує доступ до комп'ютерної системи, реєструючись у головному комп'ютері. У простому випадку користувач реєструється в головному комп'ютері через ***мережевий доступ*** (telnet) або передає файл відповідно до ***протоколу передавання файлів*** (file transfer protocol – FTP). В обох ситуаціях запускають протокол автентифікації пароля. У багатьох важливих ситуаціях, в яких скомпрометований головний комп'ютер може призвести до великих втрат, потрібна взаємна автентифікація.

Обмін повідомленнями між процесом і головним комп'ютером (process-host type). Сьогодні розвиток методів розподілених обчислень надав користувачам широкі функційні можливості. Головний комп'ютер може надавати зовнішнім процесам широкі права. Наприклад, на віддалений комп'ютер можна передавати й запускати на ньому фрагменти "коду для мобільного телефону" або "застосування, написане мовою Java". У секретних системах необхідно передбачати механізми автентіфікації розробника, щоб не допустити виконання ворожих аплетів і не надавати права доступу не авторизованим застосуванням.

Члени клубу (member-club type). Доказ членства в клубі є узагальненням способу, основаного на обміні повідомленнями між користувачем і головним комп'ютером. У цьому випадку клуб цікавить тільки мандат його члена, а не інформація про нього. Зокрема, клуб не цікавиться достовірністю особи члена клубу, який має мандат. Цей сценарій здійснюють у ***протоколах ідентифікації з нульовим розголошуванням*** (zero-knowledge identification protocol) і ***схемах незаперечного цифрового підпису*** (undeniable signature schemes).

Як правило, сторони, що обмінюються інформацією, запускають протокол автентифікації сутності для того, щоб надалі перевести спілкування на вищий рівень. У сучасній криптографії в основу організації захищених каналів зв'язку лежать криптографічні ключі. Таким чином, протоколи автентифікації сутності для подальшого обміну інформацією по захищених каналах в якості складової частини повинні містити механізм ***генерації автентифікаційних ключів*** (authenticated key establishment), або ***обміну ключами*** (key exchange), або ***узгодження ключів*** (key agreement).

Нагадаємо, що протоколи автентифікації сутності можуть передбачати автентифікацію джерела даних. Аналогічно, у протоколах генерації автентифікаційних ключів протокольні повідомлення містять параметри ключів, джерело яких також підлягає автентифікації.

У літературі протоколи автентифікації (сутності), протоколи генерації автентифікаційних ключів (обміну ключами, узгодження ключів), протоколи для захисту даних і навіть криптографічні протоколи часто називають ***протоколами зв'язку*** (communication protocols).

5.4.3. Атаки на протоколи автентифікації

Оскільки метою протоколу автентифікації (джерела даних, сутності, ключа) є перевіряння оголошеної властивості, необхідно застосувати криптографічні методи. Атака на протокол має протилежну мету. Атаку на протокол автентифікації організовує супротивник або коаліція супротивників (за загальною назвою "зловмисник"), які мають незаконну мету. Мета зловмисника може бути різною, наприклад, розкриття секретного повідомлення або ключа, або менш серйозною, наприклад, обман отримувача повідомлення. Як правило, протокол автентифікації вважають некоректним, якщо користувач вважає, що протокол виконується правильно й зв'язок встановлений зі справжнім партнером, тоді як справжній партнер доходить протилежного висновку.

Слід підкреслити, що атаки на протоколи автентифікації, як правило, не пов'язані зі зламом криптографічних алгоритмів. Звичайні протоколи автентифікації небезпечні не тому, що в них застосовано слабкі криптографічні алгоритми, а тому, що мають недоліки, які дають змогу зловмисникові пройти автентифікацію взагалі без зламу криптографічного алгоритму. Із цієї причини, аналізуючи протоколи автентифікації, зазвичай припускають, що криптографічний алгоритм, який покладено в його основу, є "досконалим", і не розглядають його потенційні слабкості.

5.4.4. Основні протоколи автентифікації

Існує багато методів здійснення протоколів автентифікації (джерела даних, сутності) і операції автентифікаційних ключів. Проте основних протокольних конструкцій, особливо вдалих, не так багато.

Почнемо з основних протокольних конструкцій, приділяючи особливу увагу конструкціям, прийнятим як міжнародні стандарти. Ці конструкції можуть і повинні бути використані для розроблення нових протоколів автентифікації. Спробуємо розібратись, чому деякі із цих конструкцій привабливіші, а інші мають недоліки.

Перерахуємо основні методи автентифікації:

– стандартні механізми визначення "свіжості" повідомлення та існування користувача;

– взаємна автентифікація й одностороння автентифікація;

– автентифікація із залученням довіреного посередника;

– "свіжість" повідомлення та існування користувача.

Перевіряння "свіжості" повідомлення – невід'ємна частина автентифікації джерела даних, а також автентифікації сутності, у процесі якої користувач повинен жваво обмінюватися інформацією зі справжнім партнером. Отже, механізми, що дають змогу встановити "свіжість" повідомлення або існування користувача, належать до основних компонентів протоколу автентифікації.

Розглянемо стандартні механізми, що дають змогу вирішити поставлене завдання.

5.4.5. Стратегія "виклик – відгук"

У ***стратегії "виклик – відгук"*** (challenge-responce mechanism) перевірник отримує суміш, що складається із протокольного повідомлення й криптографічної операції, виконаної претендентом так, щоб перевірник міг переконатися в її існуванні, перевіривши "свіжість" отриманої інформації. Позначимо в описі схем автентифікації перевірника як користувача B, а пред'явника – як користувача A. Розглянемо стандартний варіант стратегії, що називають ***двопрохідним одностороннім протоколом автентифікації ISO*** (ISO Two-pass Unilateral Authentication Protocol) [10]. Він виглядає так. Перевірник отримує одноразове випадкове число (R_b), що завчасно було згенероване ним і надіслано претенденту.

Позначимо це одноразове число символом R_b. Стратегія перевіряння "свіжості" повідомлення виглядає так:

1. B → A: R_b || Text1,
2. A → B: Token AB,

де – Token AB = Text3 || $E_{K_{AB}}$ (R_b || B || Text2).

Користувач B розшифровує порцію шифру і або приймає її, якщо дізнається число R_b, або відмовляється приймати в протилежному випадку.

Тут і далі для опису стандартів ISO/IEC використовують позначення, прийняті в специфікаціях стандартних протоколів, де Textl, Text2 тощо – необов'язкові поля, символ || означає операцію зчеплення, а R_b – одноразове випадкове число, яке згенерував в цьому випадку абонент B.

Перше повідомлення, яке надіслав користувач B (перевіряючий) користувачу A (пред'явнику), називають ***викликом*** (challenge), а друге повідомлення, надіслане від A до B, називають ***відгуком*** (response). Користувач B є ***ініціатором*** (initiator), а користувач A – ***відповідачем*** (responder).

В описаній вище стратегії використано метод симетричної криптографії, а саме симетричне шифрування.

Отже, отримавши відповідь від претендента перевірник повинен розшифрувати порцію шифрованого тексту, використовуючи спільний ключ K_{AB}. Якщо розшифрування правильно відновлює випадкове число, користувач B має підстави вважати, що користувач A дійсно зашифрував його після отримання виклику. Якщо інтервал між викликом і відгуком достатньо малий, повідомлення вважають "свіжим". Ця стратегія основана на впевненості користувача B в тому, що користувач A виконує шифрування після отримання виклику від користувача B. Оскільки випадкове число, надіслане B, вибране на достатньо великому просторі, немає жодної можливості передбачити його заздалегідь.

Зазначимо також, що цю стратегію застосовують для автентифікації сутності. Отже, введення в повідомлення імені користувача B замість повідомлення *M* за цією стратегією є дуже важливим моментом: це підкреслює, що механізм ISO/IEC призначений для перевіряння існування користувача B і є протоколом автентифікації, в якому користувач B є суб'єктом.

Насправді правильною й стандартною стратегією, що гарантує цілісність даних за використання симетричних методів шифрування, є застосування коду розпізнавання маніпуляцій MDC. Отже, за наведеною вище стратегією шифрування слід супроводжувати кодом MDC, зашифрованим за допомогою загального ключа й таким, що є частиною зашифрованого тексту. Це забезпечує захист цілісності повідомлення. Якщо повідомлення *M* не потребує захисту секретності, для перевіряння його "свіжості" можна застосувати стратегію, що називають ***двопрохідним одностороннім протоколом автентифікації з використанням криптографічної тестової функції*** (ISO Two-pass Unilateral Authentication Protocol Using a Cryptographic Check Function (CCF)) [11]:

1. B → A: Rb || Textl;
2. A → B:||TokenAB.

Тут TokenAB = Text2 $\| f_{K_{AB}}(R_B) \|$ B || Text2), де f – криптографічна функція хешування із ключем.

Отримавши повідомлення TokenAB, користувач B повинен реконструювати ключову функцію CCF, використовуючи загальний ключ, своє випадкове число, своє ім'я й поле Text2. Якщо в результаті розшифрування відновлений блок функції CCF збігається з отриманим, автентифікацію вважають успішною, а якщо ні – невдалою.

У стратегії "виклик – відгук" можна використовувати й методи асиметричного шифрування. Таким є стандартний ***двопрохідний односторонній протокол автентифікації з відкритим ключем ISO*** (ISO Public Key Two-pass Unilateral Authentication Protocol) [10]. Він виглядає так:

1. B → A: Rn || Text 1;
2. A → B: *CertA* || TokenAB.

Тут TokenAB = (R_A ||(R_B|| B || Text3 $\|sig_A(R_A$ ||R_B|| B || Text2), де sig_A – сертифікат відкритого ключа користувача A.

Отримавши повідомлення TokenAB, користувач B повинен перевірити його цифровий підпис. Якщо підпис проходить перевірку, автентифікацію вважають успішною, а якщо ні – невдалою. За цим протоколом ISO/IEC користувач A може вільно вибирати випадкове число R_A для того, щоб запобігти ненавмисному підписанню повідомлення, підготовленого користувачем B.

5.4.6. Мітки часу

Іншим ідентифікатором свіжості повідомлення разом із генерацією випадкових чисел користувачами протоколу є мітка часу. Використовуючи ***механізм мітки часу*** (timestamp mechanism), користувачі зазначають час створення свого повідомлення, використовуючи криптографічну операцію. Тобто час створення повідомлення стає невід'ємною частиною повідомлення.

Розглянемо стандартні варіанти протоколів автентифікації за допомогою механізму міток часу.

Однопрохідний односторонній протокол автентифікації із симетричним ключем ISO (ISO Symmetric Key One-pass Unilateral Authentication Protocol) [9] виглядає так:

1. A → B: Token AB;

де Token AB = Text2|| $E_{K_{AB}}$ ($\begin{matrix} T_A \\ N_A \end{matrix}$ B||Text1).

Оскільки в цій стратегії використовують простий механізм шифрування-розшифрування, необхідно нагадати, що алгоритм шифрування має забезпечувати захист цілісності даних.

Тут $\begin{matrix} T_A \\ N_A \end{matrix}$ позначає вибір між застосуванням мітки часу T_A і випадкового числа N_A, яке називають ***порядковим номером*** (sequence number). Якщо застосовують порядковий номер, користувачі повинні синхронізувати його (наприклад, використовуючи загальний лічильник) так, щоби користувач В знав про кожне збільшення порядкового номера N_A. Після успішного отримання й перевіряння порядкового номера кожен із двох учасників протоколу має відновити лічильник.

Механізм, що використовує порядковий номер, має два недоліки. По-перше, кожен із партнерів повинен зберігати інформацію про стан лічильника. У відкритих мережах, де кожен користувач взаємодіє з багатьма партнерами, це може бути доволі складно. Отже, застосування порядкового номера недостатньо добре масштабувати. По-друге, управління порядковим номером може спричиняти ускладнення за наявності завад на лінії зв'язку, – як випадкових, так і навмисних (атака на основі відмови в обслуговуванні (denial-of-service attack)). Але протокол автентифікації також не повинен мати історії, оскільки протокол з історією не може правильно функціонувати у ворожому середовищі. Отже, механізм, оснований на застосуванні порядкового номера, не можна рекомендувати для застосування, хоча він і стандартизований.

Наступний стандартний протокол називають ***однопрохідним односторонній протоколом автентифікації з використанням криптографічної тестової функції*** (CCF) (ISO One-pass Unilateral Authentication with Cryptographic Check Function) [11]. Він виглядає так:

1. A → B: Token AB.;

де Token AB = $\begin{matrix} T_A \\ N_A \end{matrix}$ || B ||Text2|| $f_{K_{AB}}$ ($\begin{matrix} T_A \\ N_A \end{matrix}$ B||Text1); f – криптографічна функція хешування із ключем.

Назва наступного протоколу аналога стратегії шифрування з відкритим ключем – ***однопрохідний односторонній протокол автентифікації з відкритим ключем ISO*** (ISO Public Key One-pass Unilateral Authentication Protocol) [10]:

1. A → B:$Cert_A$||Token AB;

де Token AB = $\begin{matrix} T_A \\ N_A \end{matrix}$ || B ||Text2||Sig_A ($\begin{matrix} T_A \\ N_A \end{matrix}$ B||Text1).

5.4.7. Взаємна автентифікація

Під час взаємної автентифікації користувачі автентифікують один одного. Відповідно природно очікувати збільшення кількості проходів під час процесу взаємної автентифікації порівняно з односторонньою.

Сьогодні стандартизовано велику кількість протоколів взаємної автентифікації. ***Трипрохідний протокол взаємної автентифікації з відкритим ключем ISO*** (ISO Public Key Three-pass Mutual Authentication Protocol) описує стратегію, основану на застосуванні цифрового підпису. В історії цього протоколу ми бачимо основні помилки, пов'язані зі взаємною автентифікацією.

Протокол 1. Трипрохідний протокол взаємної автентифікації з відкритим ключем ISO

ПОЧАТКОВІ УМОВИ:

Користувач А є власником сертифіката відкритого ключа $Cert_A$.

Користувач В є власником сертифіката відкритого ключа $Cert_B$.

МЕТА: Взаємна автентифікація.

1. B → A: R_B.
2. A → B: $Cert_A$,Token AB.
3. B → A: $Cert_B$, Token BA.

де Token AB = $R_A||R_B||B||Sig_A(R_A||R_B||B||)$,
Token BA = $R_B||R_A||A||Sig_B(R_B||R_A||A||)$.

Може видатися, що взаємна автентифікація є двократною односторонньою автентифікацією, тобто для взаємної автентифікації достатньо двічі використати протокол односторонньої автентифікації в протилежних напрямах. На перших етапах стандартизації експерти не розрізняли взаємної й односторонньої автентифікації. У декількох попередніх варіантах цього протоколу повідомлення Token BA мало дещо інший вигляд.

Token BA= $R`_B||R_A||A||Sig_B(R`_B||R_A||A||)$.

За першим варіантом протоколу користувач В не використовує повторно випадкове число R_B, щоб уникнути підписання частково визначеного рядка, який заздалегідь відомий користувачеві А. Повідомлення Token BA було дзеркальним відображенням повідомлення Token AB. Протокол ISO/ IEC 9798-3 кілька разів переглядався, поки ***Вінер*** (Wiener) з канадського відділу інституту ISO не винайшов знамениту "канадську атаку" [1], яку стали називати атакою Вінера.

5.4.8. Автентифікація із залученням довіреного посередника

В основних конструкціях протоколів автентифікації передбачено, що між учасниками протоколу встановлено захищений канал (створений за допомогою методів симетричної криптографії), або вони знають відкритий ключ партнера (якщо використано методи асиметричної криптографії). Отже, можна стверджувати, що ці протоколи застосовують користувачі, які вже знайомі один з одним. Навіщо ж вони запускають протокол автентифікації? Можливі декілька

відповідей. По-перше, вони можуть "освіжити" захищений канал зв'язку між ними, переконавшись у реальному існуванні партнера.

По-друге, що важливіше, ці конструкції є будівельними конструкціями загальніших протоколів автентифікації, що функціонують у відкритих системах.

У стандартному режимі функціонування відкритих систем користувачі взаємодіють, а потім забувають один одного. Відкриті системи дуже великі, щоб користувач міг зберігати інформацію про свої контакти з іншими користувачами. Якщо два користувачі, не знайомі один з одним, захочуть установити між собою секретний зв'язок, вони можуть організувати захищений канал зв'язку. У сучасній криптографії такий канал зв'язку захищають криптографічним ключем. Отже, два користувачі для встановлення захищеного каналу зв'язку між собою повинні запустити протокол автентифікації – так званий ***протокол генерації автентифікованого ключа***. Завершивши сеанс секретного зв'язку, вони руйнують канал. Термін "руйнування каналу" означає, що користувачі забувають ключ, що захищав канал, і ніколи більше не використовують його. От чому канал, використовуваний для генерації автентіфікованого ключа, часто називають ***сеансовим каналом***, а ключ, що захищав його, – ***сеансовим ключем***.

Як правило, для автентифікації й генерації ключів у відкритих системах використовують централізовану службу автентифікації, відому за назвою ***довірений посередник*** (trusted third party – TTP). Така служба може бути ***інтерактивною*** (online) або ***автономною*** (offline). Якщо автентифікацію здійснює інтерактивний довірений посередник, вважають, що між ним і великою кількістю користувачів у системі існують довготривалі відносини. Протоколи автентифікації й генерації автентифікаційних ключів в інтерактивних системах із використанням довіреного посередника розробляють на базі основних конструкцій, в яких одним зі "знайомих" користувачів є сам довірений посередник. Криптографічні операції, що виконує довірений посередник, можуть залежати від криптографічних операцій, які виконує користувачі. За допомогою довіреного посередника можна встановити захищений канал навіть між користувачами, не знайомими один з одним.

Стандартні протоколи автентифікації, гармонізовані в Україні як ДСТУ ISO/IEC 9798, містять дві стандартні конструкції, в яких бере участь інтерактивний довірений посередник. Одну із таких стратегій називають ***чотирипрохідним протоколом автентифікації ISO*** (ISO Four-pass Authentication Protocol), а іншу – ***п'ятипрохідним протоколом автентифікації ISO*** (ISO Five-pass Authentication Protocol). Ці протоколи забезпечують взаємну автентифікацію користувачів і генерацію автентичних сеансових ключів. Вони

вже є повноцінними протоколами автентифікації, і їх не потрібно використовувати як будівельні конструкції для створення протоколів автентифікації вищого рівня.

Як протокол, який виконують за участі третьої довірчої сторони, розглянемо ***протокол автентифікації Нідхема – Шредера***. На прикладі цього протоколу проявилась уся складність процесу перевіряння безпеки протоколів автентифікації. Цей протокол розробили в 1978 році ***Нідхем*** (Needham) і ***Шредер*** (Schroeder). Існують варіанти здійснення цього протоколу із використанням як симетричних, так і асиметричних ключів. Вразливість цього протоколу для варіанта з використанням симетричних ключів виявили ***Дороті Деннінг*** (Dorothy E. Denning) і ***Джованні Сакко*** (Giovanni Maria Sacco) в 1981 році [15]. І майже через 20 років використання для його асиметричного варіанта вразливість виявив ***Гевін Лоу*** (Gavin Lowe) у 1995 році [16].

Розглянемо версію протоколу з використанням асиметричної криптографії. Третя довірча сторона, назвемо її Т, має сертифікати відкритих ключів усіх клієнтів, яких вона обслуговує, зокрема користувачів А і В.

Тобто А – (K_A, $Cert_A$),
В – (K_B, $Cert_B$),
Т – (K_T, $Cert_A$, $Cert_B$, $Cert_T$),

де K_A – секретний ключ користувача А; $Cert_A$ – сертифікат відкритого ключа користувача А; K_B – секретний ключ користувача В; $K^{-1}{}_B$ – сертифікат відкритого ключа користувача В; K_T – секретний ключ користувача В; $K^{-1}{}_T$ – сертифікат відкритого ключа користувача Т.

На першому кроці користувач А, ініціатор протоколу, запитує в Т сертифікат користувача В.

1. А → Т: {А, В}.

На 2-му кроці Т повертає А значення сертифіката користувача В та його ім'я. Усе підписано закритим ключем Т. користувачі А і В мають сертифікат третьої довірчої сторони Т і так можуть перевірити його підпис.

2. Т → А: { $Cert_B$, В}$_{CertT}$.

На 3-му кроці користувач А генерує випадкове число R_A і відправляє його користувачу В разом зі своїм іменем, повідомлення шифрує відкритим ключем користувача В.

3. А → В: { R_A, А}$_{CertB}$.

Отримавши повідомлення, користувач В його розшифровує своїм закритим ключем, і зрозумівши що користувач А хоче почати з ним сеанс зв'язку, виконує із третьою довірчою стороною Т кроки аналогічні тим, що були поведені на початку між користувачем А і Т.

4. B → T: {B, A}.
5. T → B: { $Cert_A$, A}$_{CertT}$.

У результаті користувачі A і B мають сертифікати відкритих ключів один одного й проводять взаємну автентифікацію з допомогою генерації випадкових чисел.

6. B → A: { R_A, R_B}$_{CertA}$.
7. A → B: { R_B}$_{CertB}$.

Використання випадкових чисел R_A і R_B не гарантує захисту від атаки типу "повтор". Для виправлення вразливості Д. Деннінг і Д. Сакко запропонували використовувати мітки часу [10].

5.4.9. Типові атаки на протоколи автентифікації

З погляду мережевого рівня для зламу протоколу зловмисникові зовсім не обов'язково застосовувати дуже хитромудрі методи. Поки вважатимемо, що зловмисник дійсно всемогутній і може організовувати різноманітні атаки на дефектні протоколи.

Незважаючи на те, що перерахувати всі прийоми, які зловмисник може застосовувати для організації своїх атак, загалом неможливо, знання основних способів атаки дає можливість розробляти стійкіші протоколи. Розглянемо декілька добре відомих методів атаки. Слід зазначити, що зловмисник може комбінувати різні прийоми, конструюючи нові, більш хитромудрі атаки.

Успішна атака на протокол автентифікації або протокол генерації автентифікованого ключа зазвичай не пов'язана зі зламом криптографічного алгоритму. Навпаки, як правило, атака напрямлена на несанкціоноване й непомітне оволодіння криптографічними мандатами або на знищення криптографічного сервісу без зламу криптографічного алгоритму. Зрозуміло, атака стає можливою унаслідок помилок, допущених під час розроблення протоколу, а не криптографічного алгоритму.

Отже, розглянемо ***атаку Лоу*** на ***протокол Нідхема–Шредера***. Отже, Лоу поділив протокол на дві логічно не пов'язані частини: перша – це кроки отримання відкритого ключа (1, 2, 4, 5), що передбачає взаємодію користувачів лише із третьою довірчою стороною, і друга – автентифікація користувачів A і B (3, 6, 7), яка передбачає взаємодію користувачів A і B між собою. У цьому випадку вважаємо, що атакуючий є законним користувачем системи, тобто він може здійснювати стандартні сеанси зв'язку з усіма користувачами системи.

Отже, користувач A здійснює коректний сеанс зв'язку зі зловмисником (згідно з [17] – intruder), який є легальним користувачем системи.

При цьому він передає випадкове число R_A, підписане сертифікатом зловмисника, і той його розшифровує.

3.1 $A \rightarrow I$: $\{ R_A, A\}_{CertI}$,

де *CertI* – сертифікат зловмисника.

Маючи R_A і отриманий завчасно сертифікат B, зловмисник формує повідомлення користувачу B на його відкритому ключі, вклавши в нього випадкове число R_A та ім'я користувача A.

3.2 $I \rightarrow B$: $\{ R_A, A\}_{CertB}$.

Користувач B, отримавши від зловмисника число R_A, генерує своє випадкове число R_B і, зашифрувавши обидва випадкові числа відкритим ключем користувача A, передає його зловмиснику. Зловмисник зразу ж пересилає його користувачу A.

6.1 $B \rightarrow I$: $\{ R_A, R_B\}_{CertA}$,

6.2 $I \rightarrow A$: $\{ R_A, R_B\}_{CertA}$.

Користувач A розшифровує повідомлення й повертає зловмиснику випадкове число користувача B, зашифрувавши повідомлення сертифікатом зловмисника.

7.1 $A \rightarrow I$: $\{ R_B\}_{CertI}$.

Зловмисник, отримавши від користувача A випадкове число R_B користувача B, тепер може повернути його користувачу B, зашифрувавши його відкритим ключем користувача B.

Таку атаку достатньо ефективно виявити за допомогою програмного засобу ***SPIN*** (simple Promela interpretator) [17], який є засобом комплексного перевіряння систем. SPIN дає можливість будувати моделі паралельних програм (і серед них протоколів), моделювати темпоральні властивості їх поведінки й автоматично перевіряти виконання темпоральних властивостей паралельних систем на їх моделях за формальним підходом. У цьому продукті використано ***мову Promela*** (Protocol Meta Language), спеціально розроблену для моделювання процесів на основі темпоральної логіки. Вступ до цієї мови й програмний засіб SPIN можна знайти в [17]. Результат перевіряння протоколу Нідхема–Шредера в [12] наведено як приклад, який ілюструє ефективність цього методу перевіряння протоколів автентифікації.

5.4.10. Протокол автентифікації Kerberos

Сьогодні ***протокол автентифікації Kerberos***, який розробили в Массачусетському технологічному інституті в 1983 році (проект ATHENA), і Open source, версія якого вийшла в 1987 році та стала IETF- стандартом у 1993 році, є фактичним стандартом системи централізованої автентифікації й розподілу ключів симетричного шифрування (RFC 4120).

Цей протокол підтримують операційні системи сімейства Unix, Windows (починаючи з Windows'2000), існують його версії для Mac OS.

Автентифікацію за протоколом Kerberos v.5 (RFC 1510) у мережах Windows (починаючи з Windows'2000 Serv.) здійснено на рівні доменів. Поряд з Kerberos, з метою забезпечення сумісності з попередніми версіями, також підтримують протокол NTLM.

Протокол Kerberos забезпечує розподіл ключів симетричного шифрування та перевіряння автентичності користувачів, що працюють у незахищеній мережі. Kerberos – це програмна система, побудована за архітектурою "клієнт-сервер". Клієнтську частину встановлюють на всі комп'ютери мережі, що захищають, крім тих, на які встановлюють компоненти сервера Kerberos. Як клієнти Kerberos можна використовувати й мережеві сервери (файлові сервери, сервери друку тощо). Серверну частину Kerberos називають ***центром розподілу ключів*** (Key Distribution Center – KDC), який складається із двох компонентів:

– сервер автентифікації (Authentication Server – AS);

– сервер видачі дозволів (Ticket Granting Server – TGS).

Кожному суб'єкту мережі сервер Kerberos призначає розподілений з ним ключ симетричного шифрування і підтримує базу даних суб'єктів та їхніх секретних ключів. Схему функціонування протоколу Kerberos наведено на рис. 5.15.

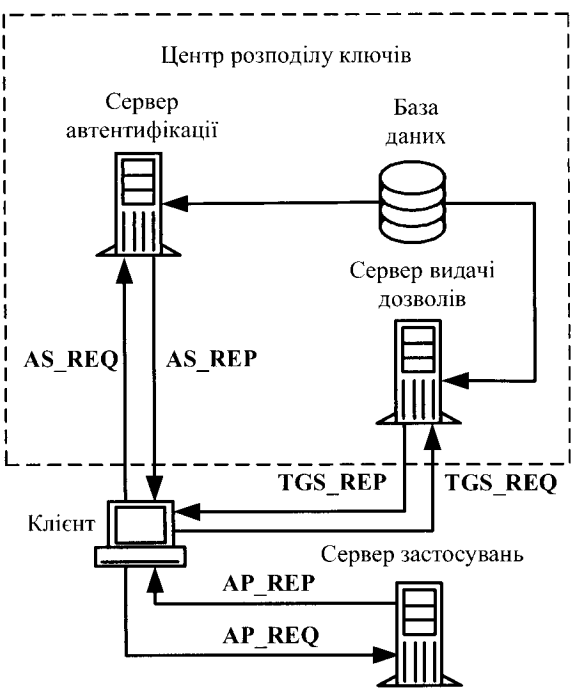


Рис. 5.15. Протокол Kerberos

Нехай клієнт С збирається почати взаємодію із сервером AS (Application Server – сервер, що надає мережевий доступ до додатка). Розглянемо основні кроки, передбачені протоколом [14]:

C -> AS: {*c*}.

Клієнт С надсилає серверу автентифікації AS свій ідентифікатор *c* (ідентифікатор передають відкритим текстом).

AS -> C: {{TGT} KAS_TGS, KC_TGS} KC,

де KC – основний ключ клієнта С;KC_TGS – ключ, що видають клієнту С для доступу до сервера видачі дозволів TGS; {TGT} – Ticket Granting Ticket – квиток на доступ до сервера видавання дозволів;

{TGT} = {*c*, tgs, t1, p1, KC_TGS},

де tgs – ідентифікатор сервера видавання дозволів, t1 – відмітка часу, p1 – період дії квитка.

Запис \ {\ cdot \} K_ {X} тут і далі означає, що вміст фігурних дужок зашифровано на ключі KX. На цьому кроці сервер автентифікації AS, перевіривши, що клієнт С є в його базі, повертає йому квиток для доступу до сервера видавання дозволів і ключ для взаємодії із сервером видавання дозволів. Уся посилка зашифрована на ключі клієнта С. Отже, навіть якщо на першому кроці взаємодії ідентифікатор *c* надіслав не клієнт С, а порушник X, то отриману від AS посилку X розшифрувати не зможе.

Отримати доступ до вмісту квитка TGT не може не лише порушник, але й клієнт С, тому квиток зашифрований на ключі, який розподілили між собою сервер автентифікації й сервер видачі дозволів.

C-> TGS: {TGT} KAS_TGS, {Aut1} KC_TGS, {ID},

де {Aut1} – автентифікаційний блок – Aut1 = {*c*, t2}, t2 – мітка часу; ID – ідентифікатор запитуваного сервісу (зокрема, це може бути ідентифікатор сервера SS).

Клієнт С цього разу звертається до сервера видавання дозволів TGS. Він пересилає отриманий від AS квиток, зашифрований на ключі KAS_TGS, і автентифікаційний блок, що містить ідентифікатор *c* і мітку часу, яка показує, коли було сформовано посилку. Сервер видачі дозволів розшифровує квиток TGT і отримує з нього інформацію про те, кому видано квиток, коли й на який термін, ключ шифрування, згенерований сервером AS для взаємодії між клієнтом С і сервером TGS. За допомогою цього ключа розшифровують автентифікаційний блок. Якщо мітка в блоці збігається з міткою в квитку, це доводить, що посилку згенерував насправді клієнт С (адже тільки він знав ключ KC_TGS і міг правильно зашифрувати свій ідентифікатор). Потім перевіряють час дії квитка та час справляння посилки 3). Якщо результат перевіряння позитивний, і діюча в системі політика дає змогу клієнту С звертатися до клієнта SS, то виконують крок 4).

TGS-> C: {{TGS} KTGS_SS, KC_SS} KC_TGS,

де KC_SS – ключ для взаємодії клієнта C і клієнта SS, {TGS} – Ticket Granting Service – квиток для доступу до клієнта SS (зверніть увагу, що таким саме скороченням в описі протоколу позначають і сервер видачі дозволів); {TGS} = {*c*, ss, t3, p2, KC_SS}.

Після цього сервер видачі дозволів TGS надсилає клієнту C ключ шифрування й квиток, необхідні для доступу до сервера SS. Структура квитка така ж, як на кроці 2): ідентифікатор того, кому видали квиток; ідентифікатор того, для кого видали квиток; мітка часу; період дії; ключ шифрування.

C-> SS: {TGS} KTGS_SS, {Aut2} KC_SS,

де Aut2 = {*c*, t4}.

Клієнт C надсилає квиток, отриманий від сервера видачі дозволів, до свого автентифікаційного блока сервера, з яким хоче встановити сеанс захищеної взаємодії. Передбачено, що SS вже зареєструвався в системі й розподілив із сервером TGS ключ шифрування KTGS_SS. Маючи цей ключ, він може розшифрувати квиток, отримати ключ шифрування KC_SS і перевірити справжність відправника повідомлення.

SS-> C: {t4 +1} KC_SS

Сенс останнього кроку полягає в тому, що тепер уже SS має довести клієнту C свою автентичність. Він може зробити це, показавши, що правильно розшифрував попереднє повідомлення. Саме тому SS бере мітку часу з автентифікаційного блока клієнта C, змінює її заздалегідь певним чином (збільшує на 1), шифрує на ключі KC_SS і повертає клієнту C.

Якщо всі кроки виконано правильно й усі перевірки пройшли успішно, то сторони взаємодії клієнт C і SS, по-перше, упевнилися в справжності один одного, а по-друге, отримали ключ шифрування для захисту сеансу зв'язку – ключ KC_SS.

У процесі сеансу роботи клієнт проходить кроки (1) і (2) тільки один раз. Коли потрібно отримати квиток на доступ до іншого сервера (назвемо його SS1), клієнт C звертається до сервера видачі дозволів TGS з уже наявними в нього квитком, тобто протокол виконують, починаючи із кроку 3).

У разі використання протоколу Kerberos комп'ютерну мережу логічно поділяють на області дії серверів Kerberos. Kerberos-область – це ділянка мережі, клієнти й сервери якої зареєстровано в базі даних одного сервера Kerberos (або в одній базі, що розділена декількома серверами). Одна область може охоплювати сегмент локальної мережі, усю локальну мережу або об'єднувати кілька пов'язаних локальних мереж. Схему взаємодії між Kerberos-областями зображено на рис. 5.16.

Для взаємодії між областями має бути здійснена взаємна реєстрація серверів Kerberos, у процесі якої сервер видачі дозволів однієї області реєструється як клієнт в іншій області (тобто заноситься в базу сервера автентифікації й спільно використовує з ним ключ).

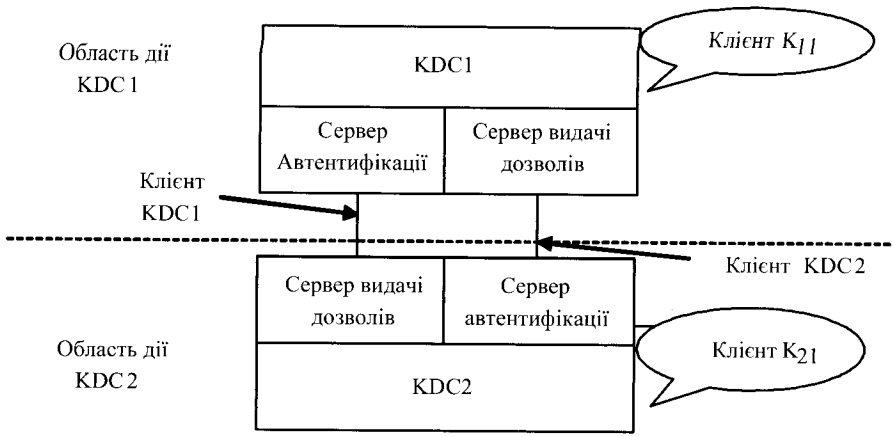


Рис. 5.16. Взаємодія між Kerberos-областями

Після встановлення взаємних угод клієнт з області 1 (нехай це буде K11) може встановити сеанс взаємодії з клієнтом з області 2 (наприклад, K21). Для цього K11 повинен отримати у свого сервера видачі дозволів квиток на доступ до Kerberos-сервера, з користувачем якого він хоче встановити взаємодію (тобто сервера Kerberos KDC2). Отриманий квиток містить відмітку про те, в якій області зареєстрований власник квитка. Квиток шифрують із використанням ключа, розділеного між серверами KDC1 і KDC2. За успішного розшифрування квитка віддалений Kerberos-сервер може бути впевнений, що квиток виданий клієнту Kerberos-області, з якою встановлено довірчі стосунки. Потім протокол працює в звичайному режимі.

Крім розглянутих можливостей, Kerberos надає ще ряд додаткових. Наприклад, зазначений у структурі квитка параметр *p* (період часу) задають парою значень "час початку дії" – "час закінчення дії", що дає змогу отримувати квитки відкладеної дії.

Є тип квитка "із правом передавання", що дає змогу, наприклад, серверу виконувати дії від імені клієнта, який звернувся до нього.

Що стосується здійснення протоколу Kerberos в Windows, то треба зазначити таке.

Ключ клієнта генерують на основі його пароля. Отже, з використанням слабких паролів ефект від надійного захисту процесу автентифікації буде зведено до нуля.

Ролі Kerberos-серверів виконують контролери домену, на кожному з яких повинна працювати служба ***Kerberos Key Distribution Center*** (KDC). Роль сховища інформації про користувачів і паролі бере на себе служба каталогу ***Active Directory***. Ключ, який поділяють між собою сервер автентифікації й сервер видачі дозволів, формують на основі пароля службового облікового запису krbtgt – цей запис автоматично створюють у разі організації домену, і він завжди заблокований.

Microsoft у своїх ОС використовує розширення Kerberos для застосування криптографії з відкритим ключем. Це дає змогу здійснювати реєстрацію в домені й за допомогою смарт-карт, що зберігають ключову інформацію та цифровий сертифікат клієнта.

Використання Kerberos потребує синхронізації внутрішнього годинника комп'ютерів, що входять у домен Windows.

Контрольні питання до розділу 5

1. Ідентифікація.
2. Автентифікації.
3. Авторизація.
4. Аудит.
5. Задачі автентифікації
6. Захист цілісності даних.
7. Загрози автентифікації.
8. Класифікація систем автентифікації за ступенем стійкості.
9. Поняття безумовно безпечних кодів автентифікації.
10. Загрози, які можуть здійснити абоненти й криптоаналітик за схемою автентифікації повної недовіри.
11. Оцінювання ступеня загроз у моделі автентифікації Сімонса.
12. Визначення криптографічної хеш-функції.
13. Властивості криптографічних хеш-функцій.
14. MD5.
15. SHA.
16. SHA3.
17. Застосування криптографічних хеш-функцій у криптографії.
18. Хеш-функції, що використовують симетричні блокові шифри.
19. Хеш-функція ГОСТ.
20. Загальні характеристики функції хешування "Купина" – національного стандарту України ДСТУ 7564:2014.
21. Функція стиснення хеш-функції "Купина".
22. Перетворення KupynaShiftRows.
23. Коди автентифікації повідомлень, що використовують функції хешування із ключем.
24. CBC-MAC.
25. Загальна характеристика протоколів автентифікації.
26. Атаки на протоколи автентифікації.
27. Взаємна автентифікація й одностороння автентифікація.
28. Автентифікація із залученням довіреного посередника.
29. Механізми виявлення свіжості повідомлення.
30. Стратегія "виклик–відгук".
31. Мітки часу.
32. Типові атаки на протоколи автентифікації.
33. Протокол автентифікації Kerberos.

Список літератури до розділу 5

1. "Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу" НД ТЗІ 1.1-003-99, ДСТСЗІ СБ України. – К., 1999.
2. Венбо Мао. Современная криптография. Теория и практика : пер. с англ. – М. : Издательский дом "Вильямс", 2005. – 768 с.

3. Горбенко І. Д., Горбенко Ю. І. Прикладна криптологія:. Теорія. Практика. Застосування: монографія. – 2-ге вид., переробл. і допов. – Харків : Форт, 2012. – 880 с.
4. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М. : Триумф, 2003. – 806 с. Режим доступу : http://www.dut.edu.ua/uploads/l_1134_27449793.pdf/
5. Зубов А. Ю. Математика кодов аутентификации. – М. : Гелиос. АРВ 2007. – 480 с.
6. G. J. Simmons, Authentication theory/coding theory, in Advances in Cryptology { CRYPTO 84, G. R. Blakley and D. Chaum (Eds.), Lecture Notes in Computer Science, No. 196, Berlin:Springer Verlag, 1985, pp. 411–431.
7. Morris J. Dworkin SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. – DOI:10.6028/nist.fips.202
8. Функція Хешування „Купина" – Новий національний стандарт України / Р. В. Олійников, І. Д. Горбенко та ін. // Радиотехника. – 2015. – Вып. 181. – С. 23–30.
9. Аутентификация. Теория и практика обеспечения безопасного доступа к информационным ресурсам: учеб. пособ. для вузов / А. А. Афанасьев, Л. Т. Веденьев, А. А. Воронцов и др.; под ред. А. А. Шелупанова, С. Л. Груздева, Ю. С. Нахаева. – М. : Горячая линия – Телеком, 2009. – 552 с.
10. ISO/IEC. Information Technology – Security Techniques – Entity Authentication – Part 2: Mechanisms using symmetric encripment algoritms. International Organization for Standardization and International Electro-technical Commissions, December 1998. ISO/IEC JTC 1/SC 27 N2145 FDIS 9798-2.
11. ISO/IEC. Information Technology – Security Techniques – Entity Authentication – Part 3: Mechanisms using digital signature techniques. International Organization for Standardization and International Electro-technical Commissions, October 1998. BS ISO/IEC 9798-3.
12. ISO/IEC. Information Technology – Security Techniques – Entity Authentication – Part 4: Mechanisms using cryptographic check-function. International Organization for Standardization and International Electro-technical Commissions, April 1999. ISO/IEC JTC 1/SC 27 N2289 FDIS 9798-4.
13. RFC 6649 Accordingly, this document updates RFC 1964, RFC 4120, RFC 4121, and RFC 4757 to deprecate the use of DES, RC4-HMAC-EXP, and other weak cryptographic algorithms in Kerberos. Because RFC 1510 (obsoleted by RFC 4120) supports only DES, this document recommends the reclassification of RFC 1510 as Historic.
14. RFC 4120 – The Kerberos Network Authentication Service (V5).
15. Dorothy E. Denning, Giovanni Maria Sacco Timestamps in key distribution protocols (англ.) // Commun. ACM. – New York, NY, USA: ACM, 1981. – Vol. 24, iss. Aug, 1981, no. 8. – P. 533–536.
16. Lowe G. "Breaking and Fixing the Needham-Schroeder Public-Key Protocol using FDR" Lecture Notes In Computer Science; v. 1055, Proceedings of the Second International Workshop on Tools and Algorithms for Construction and Analysis of Systems table of contents, p. 147–166, 1996.
17. Введение в язык Promela и систему комплексной верификации / И. В. Шошмина, Ю. Г. Карпов. – СПб. : Санкт-Петербургский государственный политехнический университет, 2009. – С. 66.
18. Електронний ресурс. – Режим доступу : http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf/
19. International Organization for Standardization (1987). International Standard 8731-2. Approved Algorithms for Message Authentication – Par 2: Message Authenticator Algorithm (MAA) (Report). Geneva.
20. International Organization for Standardization (1992). International Standard 8731-2. Approved Algorithms for Message Authentication – Part 2: Message Authenticator Algorithm (MAA) (Report). Geneva.
21. International Organization for Standardization (1990). International Standard 8730. Requirements for Message Authentication (Wholesale) (Report). Geneva.

Розділ 6
БЕЗПЕКА ІНФОРМАЦІЙНИХ СИСТЕМ

У цьому розділі розглянуто питання інформаційної безпеки інформаційних систем.

Інформаційна система – сукупність телекомунікаційних мереж і засобів для накопичування, оброблення, зберігання та розповсюджування інформації (даних) [1–4].

Телекомунікаційна мережа – комплекс технічних засобів телекомунікацій та споруд (вузли комутації, станційні та лінійні споруди, що містять лінії, канали чи системи зв'язку), призначених для маршрутизації, комутації, передавання знаків, сигналів, письмового тексту, зображень та звуків або будь-яких повідомлень по радіо, проводовими, оптичними чи іншими електромагнітними системами між кінцевим обладнанням джерела інформації та отримувача інформації. Така мережа є сукупністю вузлів комутації й кінцевого обладнання, об'єднаних лініями, каналами електрозв'язку чи телекомунікаційними системами.

Телекомунікаційна система (система зв'язку) – сукупність каналу електрозв'язку та технічних засобів (зокрема кінцевого обладнання), що забезпечують передавання інформації між джерелом інформації й отримувачем інформації.

Телекомунікаційна мережа та телекомунікаційна система мають подібне призначення, проте характерною особливістю мережі є те, що в ній, на відміну від системи, інформацію можна передавати різними маршрутами.

Телекомунікації (електрозв'язок) – передавання, випромінювання та/або приймання знаків, сигналів, письмового тексту, зображень та звуків або будь-яких повідомлень по радіо-, проводовими, оптичними або іншими електромагнітними системами.

Інформаційна безпека телекомунікаційних мереж – здатність телекомунікаційних мереж захищати від знищення, перекручення, блокування інформації, її несанкціонованого витоку або від порушення встановленого порядку її маршрутизації.

6.1. Захист інформації в каналах електрозв'язку

6.1.1. Види та принципи побудови каналів електрозв'язку, телекомунікаційних систем та мереж

Канал електрозв'язку – сукупність технічних засобів і середовища передавання, які забезпечують передавання сигналів електрозв'язку між двома вузлами телекомунікаційної мережі. Канал електрозв'язку характеризується смугою робочих частот та / або швидкістю передавання [1–4].

Середовищем передавання є лінія зв'язку. Одна й та сама лінія зв'язку може бути спільною для декількох каналів. Інколи до складу одного каналу електрозв'язку може входити декілька ліній.

Класифікацію каналів електрозв'язку найчастіше за такими ознаками:

– призначення;

– спосіб експлуатації;

– смуга частот, виділених для одного каналу;

– тип лінії зв'язку, яку використано у каналі;

– тип сигналів, які передають каналом.

За призначенням розрізняють телефонні, телеграфні, факсимільні, телевізійні, відеофонні канали та канали звукового мовлення, телемеханіки та передавання даних.

За способом експлуатації розрізняють односторонні (симплексні) та двосторонні (дуплексні та напівдуплексні) канали. Односторонніми каналами електрозв'язку передають інформацію лише в один бік, двосторонніми дуплексними – одночасно в обидва боки, а двосторонніми напівдуплексними – почергово в обидва боки.

Канали бувають вузькосмугові та широкосмугові. Для широкосмугового каналу відношення ширини смуги пропускання до середньої робочої частоти є більшим за одиницю, а для вузькосмугового – меншим.

Для створення каналу електрозв'язку використовують лінії зв'язку на основі таких середовищ передавання:

– симетричні кабелі зв'язку;

– коаксіальні кабелі;

– волоконно-оптичні кабелі;

– повітряні лінії;

– радіолінії.

Залежно від типу сигналів, які передають каналом електрозв'язку, розрізняють неперервні, дискретні, неперервно-дискретні та дискретно-неперервні канали.

У загальному випадку односторонній канал електрозв'язку містить шифратор, кодер, модулятор, передавальний пристрій, лінію зв'язку, приймальний пристрій, демодулятор, декодер та дешифратор. Переважно застосовують двосторонні канали електрозв'язку або два односторонні канали, що містять компоненти, які дають змогу передавати інформацію в двох напрямках одночасно. У такому випадку для під'єднання передавача та приймача до однієї лінії зв'язку використовують розподільчий фільтр. У простіших каналах відсутні деякі компоненти, наприклад, кодер та декодер.

Телекомунікаційна система забезпечує передавання інформації між двома віддаленими вузлами (кінцевим обладаннням). Її переважно будують за лінійною топологією. Один із варіантів структурної схеми телекомунікаційної системи зображено на рис. 6.1. До складу такої системи входить кодер джерела інформації 1, кодер джерела інформації 2, декодер отримувача інформації 1, декодер отримувача інформації 2 (кодери та декодери джерела інформації в цьому випадку є кінцевим обладнанням) та канал електрозв'язку, що використовує одну лінію зв'язку й забезпечує передавання інформації від джерела інформації 1 до отримувача інформації 2, а від джерела інформації 2 – до отримувача інформації 1. У багатоканальних телекомунікаційних системах додатково присутнє обладнання ущільнення та розділення каналів.

Телекомунікаційна мережа забезпечує передавання інформації між багатьма віддаленими вузлами (кінцевим обладаннням). Її, здебільшого, будують за розгалуженою топологією (сітка, зірка, шина, ієрархічна зірка, деревоподібна, кільцева тощо). За функційним призначенням телекомунікаційну мережу поділяють на ***транспортну мережу*** (Transport Network) і ***мережу доступу*** (Access Network). Транспортна телекомунікаційна мережа призначена для передавання високошвидкісних потоків інформації між вузлами комутації, а телекомунікаційна мережа доступу – для передавання інформації між кінцевим обладнанням, під'єднаним до пункту закінчення телекомунікаційної мережі, і найближчим вузлом комутації.

До телекомунікаційних систем належать:

– проводові системи передавання;
– волоконно-оптичні системи передавання;
– системи радіозв'язку;
– радіорелейні системи передавання;
– супутникові системи зв'язку.

До телекомунікаційних мереж належать:

– оптичні транспортні мережі;
– мережі фіксованого телефонного зв'язку;
– мережі коміркового зв'язку;
– комп'ютерні мережі.

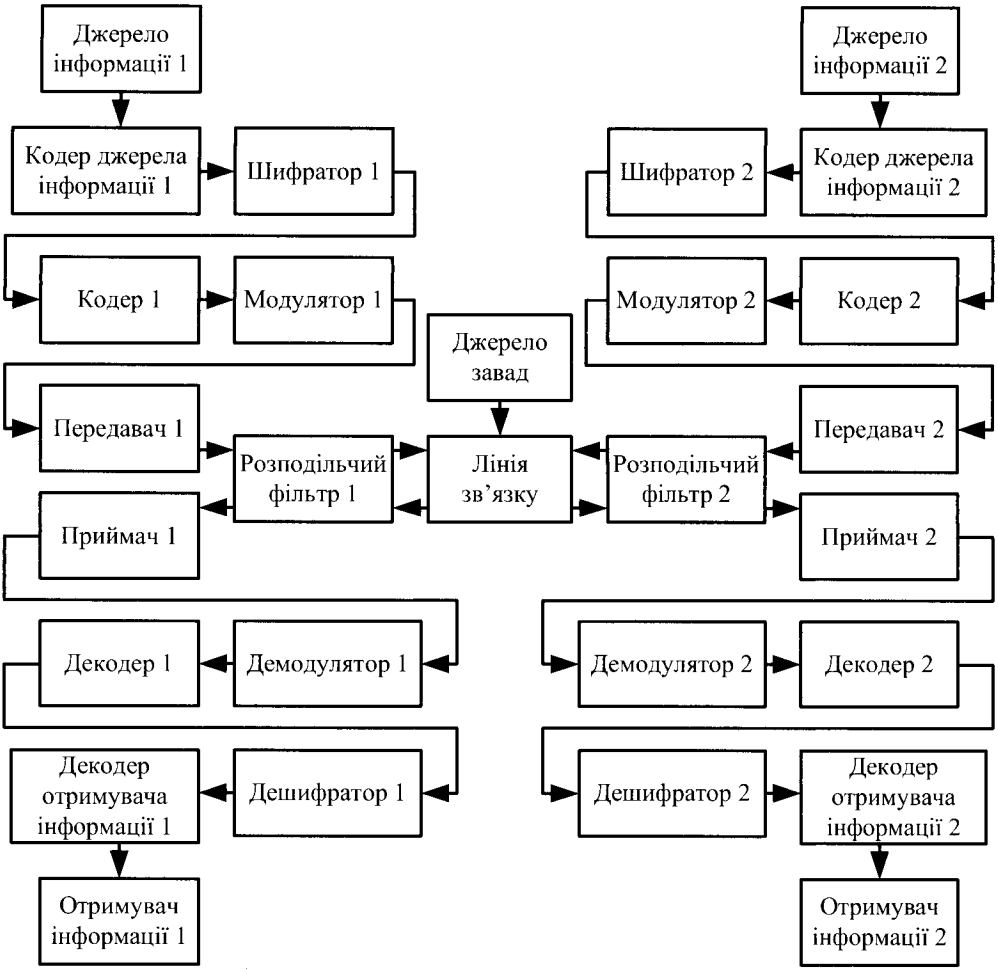


Рис. 6.1. Структурна схема телекомунікаційної системи

6.1.2. Засоби несанкціонованого доступу до інформації в абонентських телефонних лініях

Телефонну мережу будують за розгалуженою топологією. Основою сучасних телефонних мереж є автоматичні телефонні станції (АТС), що забезпечують доступ множині кінцевого обладнання абонентів (телефонних апаратів, факсимільних апаратів, модемів тощо) до з'єднувальних ліній (магістральних каналів електрозв'язку) мережі [5–6].

Абонентська телефонна лінія (АТЛ) (рис. 6.2, *а*) призначена для з'єднання телефонного апарата (ТА) абонента з автоматичною телефонною

станцією. До її складу входять: ділянка з використанням двопроводового телефонного проводу (ТРП) між телефонним апаратом та розподільчою коробкою (РК) у будівлі; ділянка із застосуванням телефонного розподільчого кабелю (ТРК) між розподільчою коробкою та розподільчою шафою (РШ), яка знаходиться ззовні будівлі; ділянка із застосуванням телефонного магістрального кабелю (ТМК) між розподільчою шафою та автоматичною телефонною станцією, що прокладений у ґрунті, телефонній каналізації, по стінах будівель чи підвішений на опорах повітряних ліній зв'язку.

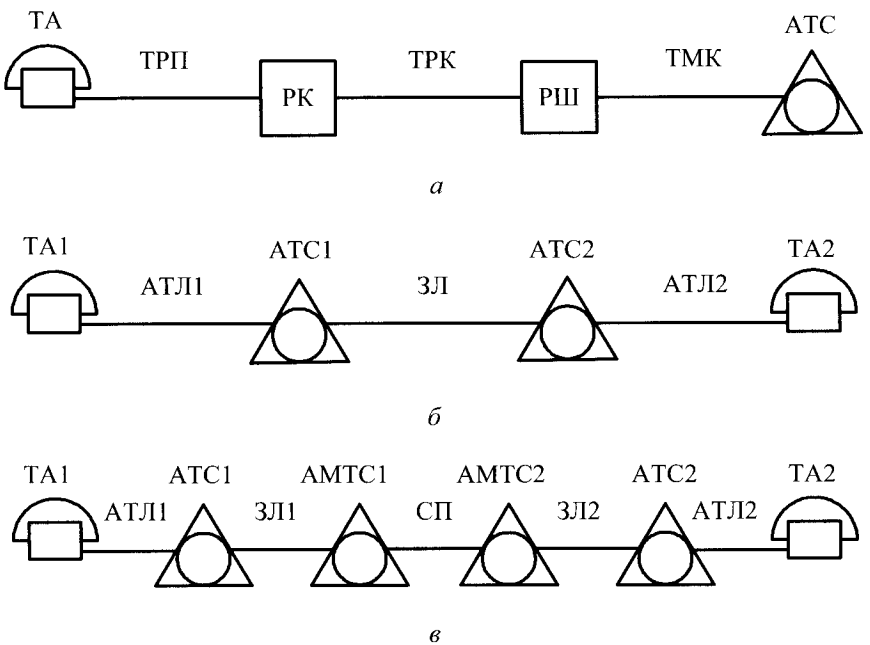


Рис. 6.2. Абонентська телефонна лінія (а), телефонний тракт міської (б) та міжміської (в) телефонних мереж

Зв'язок між телефонними апаратами двох абонентів здійснюють із використанням ***телефонного тракту міської телефонної мережі*** (рис. 6.2, *б*), до складу якого входять абонентські телефонні лінії, автоматичні телефонні станції та з'єднувальні лінії (ЗЛ) між автоматичними телефонними станціями. Якщо два абоненти знаходяться в різних населених пунктах, для забезпечення телефонного зв'язку використовують ***телефонний тракт міжміської телефонної мережі*** (рис. 6.2, *в*), який складається з абонентських телефонних ліній, автоматичних телефонних станцій, з'єднувальних ліній, автоматичних міжміських телефонних станцій (АМТС) та систем передавання (СП) між автоматичними міжміськими телефонними станціями.

Оскільки доступ до обладнання автоматичних телефонних станцій є обмеженим, а перехоплення голосових повідомлень у з'єднувальних лініях і магістральних каналах електрозв'язку систем передавання є складним через потребу демультиплексування групових сигналів, абонентські телефонні лінії та телефонні апарати є найуразливішими елементами телефонного тракту з погляду інформаційної безпеки.

Абонентські телефонні лінії та телефонні апарати, що входять до складу мереж фіксованого телефонного зв'язку, можуть бути використані для несанкціонованого доступу до інформації так:

– можливе перехоплення (підслуховування) телефонних розмов внаслідок контактного або безконтактного підключення до телефонної лінії телефонних закладок, диктофонів та інших засобів несанкціонованого доступу до інформації;

– телефонні лінії, що проходять через приміщення, можуть бути використані як джерела живлення акустичних закладок, установлених у цих приміщеннях, а також для передавання перехопленої інформації;

– телефонні апарати (навіть за покладеної трубки) можуть бути використані для перехоплення акустичної мовної інформації із приміщень, у яких їх встановлено, тобто для підслуховування розмов у цих приміщеннях.

Серед засобів несанкціонованого доступу до інформації в абонентських телефонних лініях набули широкого поширення ***телефонні закладки***. Залежно від способу підключення до абонентських телефонних ліній розрізняють безконтактні та контактні телефонні закладки. ***Безконтактна телефонна закладка*** може мати вигляд індуктивного знімача, який у найпростішому випадку є навитою на розрізане феритове кільце котушкою. При охопленні кільцем одного із проводів абонентської телефонної лінії відбувається наведення на таке кільце електромагнітних коливань, які випромінює лінія зв'язку під час проходження через неї мовного сигналу. Вони після підсилення надходять на пристрій відтворення чи запису. Безконтактні телефонні закладки не вдається виявити вимірюванням електричних параметрів телефонної лінії, але якість перехопленого сигналу низька через обмежену чутливість індуктивного знімача та дестабілізуючий вплив різних електромагнітних перешкод. Крім того, безконтактні пристрої є доволі громіздкими, що ускладнює їх камуфлювання.

Контактні телефонні закладки мають гальванічний контакт із телефонною лінією, тому вони забезпечують вищу якість перехопленого інформаційного сигналу. Вони бувають послідовного й паралельного типу. ***Паралельну телефонну закладку*** підключають до двох проводів телефонної лінії. Вона характеризується високим вхідним опором (понад 10 кОм) і малою

вхідною ємністю, що ускладнює її виявлення. ***Послідовну телефонну закладку*** включають у розрив одного із проводів телефонної лінії. Вона має вхідний опір 100–500 Ом і значну вхідну ємність, що полегшує її виявлення. Живлять телефонні закладки безпосередньо від абонентської телефонної лінії або від автономного джерела. У першому випадку блок живлення виконують у вигляді спеціального узгоджувального пристрою, що забезпечує практично необмежений термін дії, хоча може бути виявлений за ознакою додаткового навантаження на телефонну лінію. У другому випадку закладка має протилежні властивості. Для заощадження ресурсу автономних джерел живлення та маскування до складу телефонних закладок вводять спеціальні пристрої-активатори. Їхня робота може ґрунтуватися на аналізі стану телефонної лінії (активується телефонна закладка після підняття трубки) або на детектуванні розмовного сигналу в абонентській телефонній лінії (наявна система акустозапуску (VOX – voice-operator-switch)).

Методи виявлення контактних телефонних закладок такі:

– контроль знеструмленої абонентської телефонної лінії;

– контроль абонентської телефонної лінії в робочому стані.

Методи виявлення несанкціонованих підключень у знеструмлених абонентських телефонних лініях ґрунтуються на відхиленні параметрів "чистої лінії" та лінії з підключеною телефонною закладкою. Залежно від того, які параметри лінії контролюють, розрізняють методи контролю:

– опору шлейфа;

– асиметрії проводів;

– первинних параметрів (R, C, L);

– вольт-амперної характеристики;

– фігур Лісажу;

– перехідної чи імпульсної характеристик;

– амплітудно-частотної та фазочастотної характеристик;

– нелінійності (нелінійна локація);

– неоднорідності (імпульсна рефлектометрія).

Методи виявлення несанкціонованих підключень в абонентських телефонних лініях у робочому стані ґрунтуються на контролі таких параметрів лінії:

– напруги живлення;

– струму короткого замикання;

– навантажувальної характеристики;

– контролі сигналів.

Пристрої, що працюють за переліченими методами, характеризуються різним ступенем складності й рівнем достовірності результатів контролю.

6.1.3. Методи виявлення та боротьби із засобами несанкціонованого доступу до інформації в абонентських телефонних лініях у робочому стані

Контроль напруги живлення абонентської телефонної лінії є одним із найпростіших та найінформативніших методів [5–6]. Пристрої, в яких використано цей метод, визначають факт підключення до абонентської телефонної лінії за зміною напруги живлення порівняно з напругою на "чистій" лінії (без підключених засобів несанкціонованого доступу до інформації). У сторожовому режимі за допомогою пристрою захисту неперервно вимірюють напругу живлення на затискачах телефонного апарата. Якщо зміна цієї напруги, що зумовлена несанкціонованим підключенням, перевищує встановлений пороговий рівень, то пристрій захисту видає сигнал тривоги (звуковий або світловий).

Напруга живлення на затискачах телефонного апарата без підключеної закладки (на "чистій" лінії) у режимі "Очікування" (покладена трубка телефонного апарата) дорівнює напрузі $U_{АТС}$, що видає в лінію автоматична телефонна станція (АТС):

$$U_{ТА}^{o} = U_{АТС}, \text{ (60 В для аналогової АТС та 48 В для цифрової АТС)}, \quad (6.1)$$

а в режимі "Розмова"

$$U_{ТА}^{p} = U_{АТС}\frac{R_{ТА}}{R_{ТА} + R_{АТЛ}} \text{ (може бути в межах 8–12 В)}, \quad (6.2)$$

де $R_{ТА}$ – вхідний опір телефонного апарата; $R_{АТЛ}$ – опір шлейфа абонентської телефонної лінії.

За наявності паралельного підключення засобу несанкціонованого доступу до інформації (***паралельної телефонної закладки***) із вхідним опором $R_{пар}$ до абонентської телефонної лінії напруга живлення на затискачах телефонного апарата в режимі "Очікування" становитиме

$$U_{ТА\,пар}^{o} = U_{АТС}\frac{R_{пар}}{R_{АТЛ} + R_{пар}}, \quad (6.3)$$

а в режимі "Розмова"

$$U_{ТА\,пар}^{p} = U_{АТС}\frac{1}{1 + R_{АТЛ}\left(1 + R_{ТА}/R_{пар}\right)/R_{ТА}}. \quad (6.4)$$

Підключення паралельної телефонної закладки зумовлює зміну напруги живлення на затискачах телефонного апарата в режимі "Очікування" на величину

$$\Delta U^{o}_{TA\ пар} = U^{o}_{TA\ пар} - U^{o}_{TA}, \tag{6.5}$$

а в режимі "Розмова"

$$\Delta U^{p}_{TA\ пар} = U^{p}_{TA\ пар} - U^{p}_{TA}. \tag{6.6}$$

За наявності послідовного підключення засобу несанкціонованого доступу до інформації (***послідовної телефонної закладки***) із вхідним опором $R_{пос}$ у розрив одного із проводів абонентської телефонної лінії напруга живлення на затискачах телефонного апарата в режимі "Очікування" становитиме

$$U^{o}_{TA\ пос} = U_{АТС}, \tag{6.7}$$

а в режимі "Розмова"

$$U^{o}_{TA\ пос} = U_{АТС}\frac{R_{TA}}{R_{АТЛ} + R_{пос} + R_{TA}}. \tag{6.8}$$

Підключення послідовної телефонної закладки зумовлює зміну напруги живлення на затискачах телефонного апарата в режимі "Очікування" на величину

$$\Delta U^{o}_{TA\ пос} = U^{o}_{TA\ пос} - U^{o}_{TA} = 0, \tag{6.9}$$

а в режимі "Розмова"

$$\Delta U^{p}_{TA\ пар} = U^{p}_{TA\ пар} - U^{p}_{TA}. \tag{6.10}$$

Отже, метод контролю напруги живлення абонентської телефонної лінії в режимі "Очікування" дає можливість виявляти лише паралельні телефонні закладки, а в режимі "Розмова" – паралельні й послідовні.

В абонентській телефонній лінії можна використати ***метод вимірювання струму короткого замикання*** $I_{КЗ}$ (струм шлейфа) за допомогою амперметра, який підключають до лінії замість телефонного апарата. Струм короткого замикання в "чистій" лінії дорівнює

$$I_{КЗ} = U_{АТС} / R_{АТЛ}, \tag{6.11}$$

а в лінії з послідовно включеною закладкою

$$I_{КЗ\ пос} = U_{АТС} / (R_{АТЛ} + R_{пос}). \tag{6.12}$$

Зміна струму, зумовлена таким підключенням, дорівнює

$$\Delta I_{КЗ\ пос} = I_{КЗ\ пос} - I_{КЗ}. \tag{6.13}$$

Отже, метод контролю струму короткого замикання придатний для виявлення лише послідовних телефонних закладок.

Деякі телефонні закладки, призначені для перехоплення телефонних повідомлень, підключають до абонентської телефонної лінії лише в режимі

"Розмова", коли струм шлейфа перевищує певне порогове значення. ***Метод контролю навантажувальної характеристики*** використовують для виявлення саме таких закладок. ***Навантажувальна характеристика*** – залежність струму в абонентській телефонній лінії від опору навантаження, під'єднаного замість телефонного апарата. Ця залежність є плавно змінною за відсутності закладки й стрибкоподібно змінюється в момент підключення до лінії додаткового опору закладки.

Метод контролю сигналів передбачає підключення до абонентської телефонної лінії пристроїв контролю сигналів в абонентській телефонній лінії з метою виявлення й аналізу сигналів мовного та позамовного діапазону частот. Цей метод дає змогу: прослуховувати низькочастотний сигнал у лінії, виявляючи його зв'язок з акустичним сигналом у контрольованому приміщенні; виявляти наявність сигналів високочастотного зондування; виявляти наявність модуляції зондувального високочастотного сигналу, що пов'язана з акустичним сигналом у контрольованому приміщенні.

Основними ***методами боротьби із засобами несанкціонованого доступу до інформації*** в абонентських телефонних лініях є пошук закладок із використанням розглянутих вище методів їх виявлення з подальшим припиненням їх роботи (відключення від абонентської телефонної лінії), а також випалювання закладок високою напругою, коли від абонентської телефонної лінії відключають усі абонентські кінцеві пристрої, після чого подають у лінію короткочасні імпульси високої напруги. Також можливе застосування генераторів шуму в діапазоні звукових частот, проте їхня робота в приміщенні погіршує умови мовного спілкування людей.

6.1.4. Засоби несанкціонованого доступу до інформації, що використовують радіолінії

До засобів несанкціонованого доступу до інформації, що використовують радіолінії, належать радіозакладки, серед яких розрізняють радіомікрофони, закладні відеокамери, телефонні закладки тощо [5, 6, 17–22]. Радіомікрофони є найпоширенішими технічними засобами отримання акустичної інформації. Їхня популярність пояснюється простотою використання, відносною дешевизною, малими розмірами й можливістю камуфльованого виконання. Закладні відеокамери використовують для отримання відеоінформації про об'єкт спостереження. Телефонні закладки дають змогу несанкціоновано знімати інформацію з ліній зв'язку, наприклад, дистанційно прослуховувати телефонні розмови.

Радіозакладка – передавач, що дає змогу перенести в діапазон радіочастот акустичний (звуковий) сигнал, отриманий від вбудованого мікрофона

(розташованого в контрольованому приміщенні), підсилювача нижніх частот (підключеного до телефонної лінії зв'язку), або відеосигнал (отриманий від вбудованої мініатюрної відеокамери). Розрізняють радіозакладки з параметричною чи кварцовою стабілізацією частоти. Як джерело електроживлення радіозакладок використовують малогабаритні акумулятори. Термін роботи таких закладок залежить від часу роботи акумулятора (1–2 доби безперервної роботи). Радіозакладки можуть бути дуже складними (використовувати системи накопичення та передавання інформації, пристрої дистанційного накопичення). Найпростіші радіозакладки містять три основні вузли, які визначають їхні технічні характеристики. Ними є: мікрофон, що визначає зону акустичної чутливості радіозакладки (відеокамера, що визначає зону видимості радіозакладки в просторі); радіопередавач, що визначає дальність її дії й прихованість роботи; джерело електроживлення, що визначає час безперервної роботи.

Прихованість роботи радіозакладок забезпечують невеликою потужністю передавача, вибором частоти випромінювання, обмеженням часу безперервної роботи (ввімкнення за допомогою дистанційного керування лише тоді, коли це необхідно, або короткочасне передавання попередньо накопиченої інформації), а також застосуванням спеціальних заходів закриття інформації. Часто робочу частоту вибирають поблизу носійної частоти потужної радіостанції, яка власними сигналами маскує працюючу закладку. Для закриття (приховування) радіоканалу передавання інформації застосовують різні підходи: шифрування (скремблювання) переданого сигналу, наприклад, методом аналогового маскування сигналу інверсією низькочастотного спектра або адаптивної дельта-модуляції інформаційного сигналу з додаванням цифрового псевдовипадкового потоку. Радіозакладки із закритим каналом важче виявити навіть із застосуванням високовартісних пошукових технічних засобів, але й ціни на такі пристрої значно вищі. Мікрофони чи відеокамери, що використовують у радіозакладках, можуть бути вбудованими або виносними. Фізична прихованість радіозакладок залежить від ретельності їх маскування в контрольованому приміщенні. Найчастіше радіозакладки виготовляють у камуфльованому вигляді (запальнички, картини, авторучки, предмети інтер'єру тощо).

Дальність дії радіозакладок переважно залежить від потужності передавача, частоти носійного коливання, виду модуляції та характеристик приймального пристрою. Час безперервної роботи залежить від способу забезпечення живлення радіозакладки. Якщо радіозакладка живиться від мережі 220 В (а такі закладки найчастіше виконують у вигляді трійників, розеток, подовжувачів, перехідників), час їх роботи необмежений. Якщо живлення здійснюють від батарейок або акумуляторів, то вихід із такої ситуації знаходять

у застосуванні режиму акустозапуску (керування голосом), використанні дистанційного керування включенням або збільшенні ємності батарей. Дальність дії, габарити й час безперервної роботи взаємопов'язані.

6.1.5. Виявлення засобів несанкціонованого доступу до інформації, що використовують радіолінії

Радіозакладки виявляють за такими ознаками [5, 6, 17–22]:

1. Відносно високий рівень випромінювання сигналу радіозакладкою, зумовлений необхідністю передавання сигналу за межі контрольованого приміщення. Якщо радіозакладку встановлено в певному приміщенні, то рівень сигналу в ньому завжди вищий, аніж за його межами.

2. Наявність гармонік випроміненого радіозакладкою сигналу. Ослаблення випромінювань на гармоніках, як правило, не перевищує 40–50 дБ відносно рівня сигналу. Виявити гармоніки можна за допомогою спеціальних сканерів на відстані до 10 м; ця відстань обмежена лише частотним діапазоном сканера.

3. Поява у зазвичай вільному діапазоні частот нового джерела випромінювання. Переважно радіозакладка використовує діапазон частот, що не зайнятий у певній місцевості радіомовними, телевізійними станціями, системами мобільного та транкінгового зв'язку.

4. Істотна нерівномірність рівня випромінювання сигналу радіозакладкою в межах контрольованого приміщення, що спричинена використанням у ній напрямлених антен і, як наслідок, наявністю сильної локалізації випромінювання.

5. Особливості поляризації випроміненого радіозакладкою сигналу. За зміни просторового розташування або орієнтації зондувальної антени спостерігають зміну рівня всіх джерел радіовипромінювання, що пов'язано із просторовим розподілом та поляризацією випромінювання. При цьому однотипні віддалені джерела в одному діапазоні частот (якщо шукати за допомогою аналізатора спектра) поводяться приблизно однаково, на відміну від сигналу, випроміненого радіозакладкою.

6. Зміна (розширення) спектра випромінювання сигналу радіозакладкою у разі виникнення будь-яких акустичних сигналів чи шумів у контрольованому приміщенні, що проявляється лише тоді, коли вона працює без кодування передаваної інформації. При цьому незалежно від того, чи використано маскування, чи ні – спектр випромінювання завжди розширюється відповідно до збільшення рівня звуку. Це добре видно під час аналізу спектра випроміненого радіозакладкою сигналу, якщо видати різкі звуки або вдарити в долоні в приміщенні, де встановлено радіозакладку.

7. Розпізнавання людиною акустичних сигналів, отриманих у результаті демодуляції прийнятого сигналу під час озвучення контрольованого приміщення певними акустичними сигналами. Якщо радіозакладка працює без маскування, то під час озвучення приміщення та демодуляції прийнятого сигналу людина може почути шум приміщення, тестовий акустичний сигнал або мовний сигнал. У разі застосування маскування демодульований сигнал нагадує нерозбірливу мову, якщо як тестовий сигнал використовують музику. У разі застосування кодування можна почути білий шум, при цьому не буде жодного взаємозв'язку з тестовим сигналом.

8. Безперервний, безперервний протягом деякого часу або переривчастий режим роботи радіозакладки. Найпростіша радіозакладка працює безперервно протягом часу, який забезпечує джерело живлення. Закладка із системою акустозапуску (VOX – voice-operator-switch) працюватиме переривчасто вдень і практично не працюватиме вночі, коли в контрольованому приміщенні немає акустичних сигналів чи шумів. Якщо радіозакладку встановлено в телефонний апарат, випромінювання сигналу виникає одночасно з підняттям трубки й зникає, коли трубку покладено. Радіозакладка з дистанційним керуванням працюватиме протягом коротких сеансів зв'язку вдень, переважно в моменти проведення в контрольованому приміщенні важливих переговорів.

У більшості випадків для несанкціонованого знімання інформації із приміщення зловмисник застосовує відповідні закладки. Процедура пошуку закладок складається з декількох етапів:

– візуальний огляд і фізичний пошук закладок;

– виявлення закладок як електронних засобів;

– перевіряння наявності каналів витоку інформації.

Візуальний огляд здійснюють, обстежуючи всі предмети у зоні контролю, розміри яких достатньо великі для того, щоб можна було розмістити в них технічні закладки (настільні прилади, рами картин, телефони, квіткові горщики, книги; пристрої, що живляться від електромережі: комп'ютери, ксерокси, радіоприймачі тощо). Фізичний пошук закладок здійснюють із застосуванням спеціальних засобів відеоспостереження та металодетекторів.

Закладки, яких не було виявлено під час візуального огляду й фізичного пошуку, шукають за переліченими вище демаскуючими ознаками із застосуванням ***спеціальних радіотехнічних засобів виявлення*** (радіозакладка – джерело сигналу в діапазоні радіочастот). Також необхідно перевірити в зоні контролю наявність каналів витоку інформації.

До основних пристроїв, які використовують для виявлення радіозакладок, належать:

– індикатори поля;

– спеціальні радіоприймачі;
– програмно-апаратні комплекси радіоконтролю;
– нелінійні радіолокатори.

Індикатори поля (детектори поля) – прилади, що дають змогу визначити наявність закладок за їх радіовипромінюванням. Вони є найпростішими засобами виявлення радіозакладок. Це приймачі з низькою чутливістю, тому вони фіксують випромінювання закладок на гранично малих відстанях (10–40 см) на фоні сигналів інших радіотехнічних пристроїв та систем. Важлива перевага індикаторів поля – здатність виявляти наявність передавальних пристроїв незалежно від уживаної в них модуляції сигналу. Основний принцип пошуку полягає у виявленні абсолютного максимуму рівня випромінювання сигналу в приміщенні.

Іноді індикатори поля використовують у сторожовому режимі. У цьому випадку після повного перевіряння приміщення на відсутність радіозакладок фіксують рівень електромагнітного поля в деякій точці простору (зазвичай це стіл керівника або місце ведення переговорів), і прилад переводять у черговий режим. У разі включення закладки (приблизно на відстані до двох метрів від індикатора поля) індикатор видає сигнал про підвищення рівня електромагнітного поля. Проте необхідно враховувати той факт, що якщо використовуватиметься радіозакладка з дуже низьким рівнем випромінювання сигналу, то детектор швидше за все не зафіксує її включення.

Спеціальні радіоприймачі як пристрої виявлення радіозакладок повинні задовольняти три основні умови:

– давати можливість пошуку та налаштування на сигнал радіозакладки в заданому діапазоні радіочастот протягом невеликого проміжку часу;
– володіти властивістю селективності сигналу радіозакладки за його характерними ознаками на фоні завад;
– володіти здатністю до демодуляції різних видів сигналів.

Програмно-апаратні комплекси радіоконтролю володіють ширшими можливостями завдяки суміщенню функцій спеціальних приймачів та персональних комп'ютерів, що істотно підвищує надійність і оперативність пошуку закладок, робить процедуру їх виявлення більш зручною та технологічною. Такі комплекси дають змогу:

– зберігання апріорної інформації про радіоелектронні засоби, що працюють у контрольованій області простору й вибраних діапазонах частот;
– отримання програмними методами часових і частотних характеристик прийнятих сигналів;
– тестування прийнятих сигналів за сукупністю ознак на приналежність до випромінювання закладок.

Програмно-апаратні комплекси радіоконтролю забезпечують:

– виявлення випромінювань закладок;

– пеленгацію закладок у реальному масштабі часу;

– визначення дальності до джерел випромінювання;

– аналого-цифрове оброблення сигналів із метою визначення їх приналежності до випромінювання закладок;

– контроль силових, телефонних, радіотрансляційних та інших мереж;

– роботу в багатоканальному режимі, що дає змогу контролювати декілька об'єктів одночасно;

– установлення прицільних перешкод на частотах випромінювання закладок тощо.

Програмно-апаратні комплекси радіоконтролю складаються з таких елементів:

– широкодіапазонного переналаштовуваного за частотою приймача (сканера);

– блока розпізнавання закладок, що ідентифікує їхні випромінювання, порівнюючи прийняті демодульовані сигнали із природним акустичним фоном приміщення (пасивний спосіб) або тестовим акустичним сигналом (активний спосіб);

– блока акустичної локації, що дає можливість за затримкою перевипроміненого зондуючого звукового імпульсу визначати відстань до активних радіомікрофонів;

– електронно-обчислювальної машини (процесора), що обробляє отримані дані і керує приймачем.

За принципом побудови програмно-апаратні комплекси радіоконтролю поділяють на дві основні групи:

– спеціально розроблені комплекси, конструктивно виконані у вигляді єдиного пристрою;

– комплекси, сформовані на базі серійного сканера, персонального комп'ютера (зазвичай ноутбук) і спеціалізованого програмного забезпечення.

Нелінійні радіолокатори застосовують для пошуку встановлених закладок, що не використовують радіоканал для передавання інформації, або таких, що знаходяться в пасивному (невипромінюючому) режимі.

Нелінійні радіолокатори випромінюють електромагнітну хвилю на певній частоті f і приймають перевипромінений сигнал на цій частоті f. Якщо такий сигнал буде прийнято, то в зоні дії радіолокатора є напівпровідникові елементи, які необхідно перевірити на можливу приналежність до закладок. Нелінійний радіолокатор знаходить лише радіоелектронну апаратуру й, на відміну від класичного лінійного радіолокатора, не реагує на відбиті сигнали від навко-

лишніх предметів, тобто володіє високою вибірковістю. Джерелами перешкод для його роботи можуть слугувати ненадійні контакти проводів, для яких є характерною наявність проміжного оксидного шару.

Для успішного пошуку радіозакладок необхідно забезпечити умови для їх роботи. Для цього необхідно: "озвучити" приміщення, у якому проводять пошук, тобто створити розумний подібний до природного шум (звук), за можливості ввімкнути в мережу побутову радіоелектронну апаратуру й оргтехніку; уникати шумів, які характерні для демаскуючого процесу пошуку (різні тематичні розмови, передавання зондуючих звукових сигналів). Інакше зловмисник, що встановив закладку (якщо вона має дистанційне керування) може просто відключити її.

Зазвичай шукають радіозакладки із використанням приладів оперативного контролю. Оператор стає посередині контрольованого приміщення (у місці, де, найімовірніше, відсутні радіозакладки), вмикає прилад, фіксує рівень поля в цій точці приміщення, потім повільно переміщується приміщенням разом із приладом поблизу меблів, електронної техніки, елементів конструкції стін, стелі тощо, фіксуючи зміни рівня поля на дисплеї приладом. При цьому він намагається постійно змінювати орієнтацію антени приладу, щоб не пропустити зміни рівня поля з урахуванням поляризації сигналу. Якщо знаходяться місця, у яких рівень поля високий, то досліджує їх детальніше, змінюючи чутливість приладу, розміри антени тощо.

Із наближенням антени приладу до джерела радіовипромінювання прилад фіксує підвищення рівня прийнятого сигналу. Якщо прилад обладнано частотоміром, показ частоти прийнятого сигналу із наближенням до джерела радіовипромінювання стає менш хаотичним і за достатньої потужності джерела зберігає стабільність протягом декількох вимірювань, – відбувається так зване "захоплення" частоти. Перелічені вище ознаки дають змогу виявити місце розташування радіозакладки. Після виявлення закладки вживають заходів із припинення її роботи (відключають її від мережі електроживлення; відключають її внутрішній акумулятор; виносять із контрольованого приміщення предмет, у який вбудовано закладку тощо). Необхідно враховувати, що у разі приймання імпульсних сигналів (наприклад, сигнал від мобільного телефону, що працює в цифровій мережі коміркового зв'язку другого покоління) результати вимірювання частоти або рівня сигналу змінюватимуться в часі. Під час досліджень слід ураховувати, що рівень виміряного сигналу залежить від частотних характеристик приймача та вимірювальної антени.

Цікаві можливості з'являються за використання ***диференційних індикаторів поля***, які дають змогу аналізувати рівні електромагнітного поля, наведені на дві антени, рознесені в просторі. Ці прилади контролюють як абсолютні

значення сигналів, наведених зовнішнім електромагнітним полем на кожну антену, так і різницю сигналів, наведених у цих антенах. Це дає додаткову інформацію про структуру електромагнітного поля в просторі й полегшує роботу в умовах високого техногенного фону (особливо у великих містах). При цьому використовують особливості структури ближнього електромагнітного поля (тобто поля в безпосередній близькості від випромінювальної антени). Для неї характерне поступове зменшення сигналу, що наводиться в антені приймального пристрою, із віддаленням від джерела сигналу. Наявність двох рознесених антен дає змогу прийняти рішення про знаходження поряд малопотужного джерела випромінювання без пересування приладу.

Засоби пошуку закладних відеокамер виявляють будь-які приховані відеокамери незалежно від їхнього режиму роботи. Дальність виявлення становить від 2 до 30 метрів. Як правило, такий прилад містить декілька десятків світлодіодів, розташованих навколо об'єктива, крізь який ведуть спостереження. Коли оператор крізь об'єктив приладу оглядає контрольоване приміщення, прихована відеокамера, що з'являється в полі зору, яскраво відбиватиме світло від світлодіодів. У місце, звідки відбивається світло, встановлено відеокамеру. Також такий прилад може бути оснащений інфрачервоним світлофільтром для ослаблення природних відблисків. Також передбачено можливість регулювання потужності підсвітки.

Крім пошуку закладок із подальшим припиненням їх роботи, для боротьби з ними також застосовують ***генератори шуму*** (радіозавад). Переважно використовують генератори, що випромінюють випадкові шуми в широкому діапазоні частот, у якому переважно працюють радіоканали радіозакладок. Такі випромінювання погіршують відношення потужності сигналу до потужності завади в місці, де встановлено приймач зловмисника, чим зменшують дальність дії радіозакладок або повністю унеможливлюють їхню роботу.

6.1.6. Методи несанкціонованого доступу до інформації та методи боротьби з ним у волоконно-оптичних лініях зв'язку

Основною перевагою передавання інформації волоконно-оптичною лінією зв'язку є те, що одним оптичним волокном можливо передавати інформацію в широкому діапазоні частот. Крім цього, використання методу частотного ущільнення каналів дало змогу істотно збільшити загальну швидкість передавання інформації існуючими волоконно-оптичними лініями зв'язку. У результаті за допомогою одного оптичного волокна можна замінити до шестисот пар проводів симетричного кабелю зв'язку [5, 6, 17–22].

Інші переваги використання волоконно-оптичної лінії зв'язку такі:

– висока завадостійкість порівняно із симетричним кабелем зв'язку;

– практично повна відсутність взаємного впливу окремих оптичних волокон кабелю;

– велика пропускна здатність;

– висока захищеність лінії зв'язку, оскільки дуже важко підключитися до неї без її істотного ушкодження;

– суттєве збільшення довжини регенераційної ділянки (довжини кабельної лінії, на яку можна передавати інформацію лінією без використання регенераторів сигналу);

– загальна безпечність (насамперед, пожежна) оптичних кабелів у разі їх замикання чи розірвання.

Проте волоконно-оптичні лінії зв'язку мають й такі недоліки:

– підвищена ламкість оптичних волокон за надлишкового вигину;

– складність ремонту ліній;

– залежність затухання сигналу від кількості з'єднань між оптичними волокнами.

У волоконно-оптичних лініях зв'язку, як і в будь-яких інших лініях зв'язку, можливий несанкціонований доступ до інформації. Підключення до оптичного волокна може бути ***інтрузивним*** або ***неінтрузивним***. Перший метод передбачає перерізання волокна й під'єднання в його розріз проміжного пристрою для знімання інформації. За неінтрузивного методу підключення виконують без порушення цілісності оптичного волокна й розірвання зв'язку. Основні методи несанкціонованого доступу до волоконно-оптичних ліній зв'язку такі:

– ***оптичне розщеплювання*** – оптичне волокно розрізають та вставляють у розгалужувач (сплітер), який відгалужує частину оптичної хвилі. Цей метод є інтрузивним, тому його застосування може спричинити розірвання зв'язку й появу аварійної ситуації в телекомунікаційній системі. Проте невиявлене підключення такого типу можна використовувати протягом багатьох років;

– ***згинання волокна***, яке можливе після знімання всіх захисних оболонок на невеликій ділянці кабелю та отримання фізичного доступу безпосередньо до оптичного волокна. Цей метод є неінтрузивним. Із застосуванням цього методу згинають оптичне волокно настільки сильно, щоб кут падіння оптичної хвилі, що є носієм інформації та поширюється оптичним волокном, між напрямком руху оптичної хвилі та границею між серцевиною волокна та його оболонкою був більшим за критичний кут повного внутрішнього відбиття, і оптична хвиля почала проникати крізь оболонку за межі оптичного волокна. У місці згинання

волокна встановлюють спеціальний оптичний приймач для приймання такої оптичної хвилі. При цьому розрізняють мікрозгинання та макрозгинання оптичного волокна.

Мікрозгинання волокна, що спричиняє мікроскопічне, але різке викривлення поверхні волокна. Через цей дефект частина оптичної хвилі проникає за межі оптичного волокна й може бути прийнята спеціальним приймачем.

Макрозгинання волокна з радіусом, меншим за мінімально допустимий. Якщо волокно згинають із малим радіусом, то можливе проникнення частини оптичної хвилі за межі оптичного волокна, достатньої для її приймання спеціальним оптичним приймачем. Для кожного типу оптичного волокна існує мінімально допустимий радіус згинання. Зазвичай мінімальний радіус згинання одномодового волокна становить 6,5–7,5 см, за винятком волокна спеціального типу;

– ***тимчасове з'єднання*** (Evanescent Coupling). Цей метод використовують для перехоплення сигналу від волокна-джерела у волокно-приймач за допомогою акуратного спилювання їхніх оболонок з одного боку аж до поверхні серцевини й подальшого їх з'єднання. Метод дає змогу забезпечити проникнення частини оптичної хвилі з одного волокна в інше. Його важко здійснити в польових умовах;

– ***V-подібний виріз*** (V Groove Cut). Це спеціальна близька до серцевини виїмка в оболонці волокна, зроблена так, що кут між напрямком поширення оптичної хвилі в оптичному волокні й проекцією V-подібного вирізу є більшим від критичного. Це спричиняє повне внутрішнє відбиття, за якого частина світла виходитиме за межі волокна через V-подібний виріз;

– ***розсіювання***, яке отримують створенням решітки Брега на серцевині оптичного волокна, за допомогою якої частина оптичної хвилі виходить за межі волокна.

Несанкціоновано отриманий за допомогою перелічених вище методів сигнал обробляють або відразу на місці, або передають його в спеціалізований центр.

Є такі основні методи захисту волоконно-оптичних ліній зв'язку від несанкціонованого доступу до інформації:

– ***фізична охорона кабельних ліній*** з їх періодичним візуальним оглядом для виявлення місць несанкціонованого підключення;

– ***спостереження за параметрами кабельних ліній*** для виявлення неоднорідностей хвильового опору кабелю; за допомогою оптичного рефлектометра можна визначити відстань до місця неоднорідності, що може бути місцем несанкціонованого доступу до інформації;

– ***застосування надгнучкого оптичного волокна,*** що є стійкішим до втрат сигналу внаслідок згинання, проколювання, перекручування та інших механічних впливів на волокно;

– ***використання методів шифрування інформації***, яку передають волоконно-оптичною лінією зв'язку; використання таких методів не запобігає перехопленню інформації, проте забезпечує достатній рівень таємності.

6.2. Інформаційна безпека в мережах коміркового зв'язку

Мережі коміркового зв'язку забезпечують захист інформації в чотирьох аспектах [8]:

- захист від несанкціонованого доступу до послуг;
- захист від підслуховування в радіоканалі;
- нерозголошення місцезнаходження абонента;
- захист від використання незареєстрованого обладнання.

Інформаційна безпека повністю знаходиться в компетенції оператора, а її рівень залежить від надійності оператора.

Перші два аспекти здійснено за принципом електронного підпису. Згідно з ним, оператор і абонент зберігають таємний ключ (число) і алгоритм утворення відгуку Y на запит X. У процесі ідентифікації оператор відсилає абонентові запит X_i (інший у кожному сеансі), обчислює правильну відповідь Y_i та порівнює з відповіддю, отриманою від абонента. Будова алгоритму забезпечує неможливість його відтворення навіть за великою кількістю пар (X_i, Y_i). Тобто, навіть зібравши достатню кількість правильних відповідей Y_i, неуповноважений користувач не зможе дати правильну відповідь Y_{i+1} на запит X_{i+1} за спроби отримати доступ до послуг.

Особливості забезпечення інформаційної безпеки в мережах безпроводового зв'язку безпосередньо пов'язані зі способом їх фунціональної та прострової організації.

6.2.1. Функціональна та просторова структура мереж коміркового зв'язку

Функціональна та просторова структури мереж коміркового зв'язку зумовлені основним принципом їхньої побудови – наявністю базових станцій, які покривають відносно невеликі зони (комірки), і роботу яких (БС) необхідно координувати за допомогою відповідної апаратури та стаціонарних ліній зв'язку (кабельних, волоконно-оптичних, радіорелейних).

З огляду на це архітектура мереж коміркового зв'язку є практично однаковою для аналогових мереж (AMPS – Advanced Mobile Phone System – розвинута мобільна мовна система), цифрових мереж із часовим і частотним розділенням каналів (D-AMPS, GSM – Global System for Mobile Communication – глобальна система мобільного зв'язку) та мереж із кодовим розділенням каналів (IS-95; CDMA 2000 – Code Division Multiple Access 2000, множинний доступ із кодовим розділенням каналів; W-CDMA – Wideband CDMA, широкосмуговий CDMA). Тому склад мереж коміркового зв'язку розглянемо на прикладі мережі стандарту GSM, додатково зазначивши на особливості термінології та функціонального призначення вузлів в інших мережах.

За виконуваними функціями пристрої мережі GSM поділяють на чотири групи (рис. 6.3):

1. Ансамбль базових станцій (Base Station Subsystem – BSS);
2. Комутаційно-мережеве обладнання (Network & Switching Subsystem – NSS);
3. Абонентські (мобільні) станції (Mobile Station – MS);
4. Підсистема адміністрування, або експлуатації й обслуговування (Operation & Maintenance Subsystem – OMS).

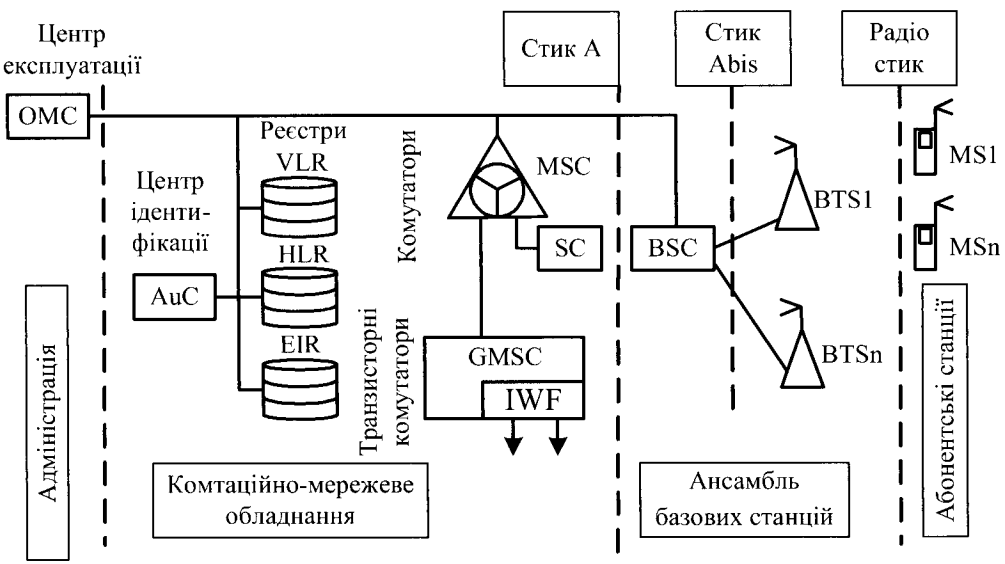


Рис. 6.3. Функціональні групи та стики мережі GSM

Ансамбль базових станцій (у мережі IS-95 названий "мережею радіодоступу") складається з базових станцій (Base Transceiver Station – BTS) та контролерів базових станцій (Base Station Controler – BSC). Крім того, у мережах CDMA ансамбль базових станцій містить пристрій вибору кадру (Selector Unit – SU).

Кожна базова станція складається із приймально-передавального пристрою, антени, пристрою оброблення радіосигналів. Базова станція пов'язана через радіоканал з абонентськими станціями, а також стаціонарними кабелями з контролером базових станцій. Інтерфейс BTS-MS названо радіостиком, а інтерфейс BTS-BSC – стиком Abis.

Контролер базових станцій керує декількома (до сотні) базовими станціями. Він вирішує питання перемикання радіоканалів (handover) та керування потужністю абонентських станцій (power control). Контролер базових станцій забезпечує через стик А зв'язок із комутатором.

Стандартизація рознімів і протоколів стиків дає змогу в одній мережі GSM використовувати апаратуру різних виробників.

Комутаційно-мережеве обладнання (у мережі IS-95 – "базова мережа") виконує основні функції комутації інформаційних каналів, збереження актуальної інформації про розташування рухомих абонентів (mobility management) та зв'язку з іншими телекомунікаційними мережами.

Комутатор (Mobile Switches Centre – MSC) – основний модуль комутаційно-мережевого обладнання. Він виконує функції забезпечення зв'язку між двома абонентами, а також функції, котрих вимагає мобільність абонентів: реєстрація положення, виклик, передавання криптографічних та ідентифікаційних кодів. Комутатор зв'язаний з одного боку з контролерами базових станцій (через стик А), а із іншого боку – з іншими комутаторами, зокрема із транзитними.

Транзитний комутатор (Gateway Mobile Switching Centre – GMSC) забезпечує стик мережі GSM з іншими телекомунікаційними мережами (мережа фіксованого телефонного зв'язку, мережа передавання даних, інша мережа коміркового зв'язку). GMSC, на відміну від MSC, додатково обладнаний ***модулем функцій спряження*** (Inter Working Functions – IWF). До транзитного комутатора можна під'єднувати контролери базових станцій, але, як правило, його з'єднують із декількома звичайними комутаторами.

Необхідність додаткових функцій, пов'язаних із рухомістю абонента, ілюструє приклад виклику абонента, місцезнаходження якого невідомо. Спроба виклику з усіх базових станцій мережі призведе до неефективного використання ресурсів і навіть блокування інформаційних каналів. Тому необхідно зберігати детальнішу інформацію про актуальне місцезнаходження абонента для більш цілеспрямованого виклику. Таку інформацію зберігають у двох базах даних (VLR та HLR), які обслуговують один або декілька комутаторів.

Реєстр власних станцій (Home Location Register – HLR) зберігає інформацію про всіх абонентів, які під'єдналися до мережі (зареєстрували стартовий пакет) на одному з комутаторів зони обслуговування HLR. Крім

ідентифікаційної інформації, в HLR зберігають адресу комутатора, з базової станції якого відбувався останній зв'язок з абонентом.

Реєстр відвідувачів (Visitors Location Register – VLR) зберігає інформацію про всіх абонентів, які в поточний момент часу знаходяться в зоні обслуговування одного з комутаторів, віднесених до цього VLR.

Із появою нової абонентської станції в зоні обслуговування VLR він скерує до HLR цієї станції свою адресу й отримає повну інформацію про абонента. Після цього в VLR зберігають детальнішу інформацію про абонента, ніж в HLR, і для створення інформаційного каналу звертатися в HLR вже не потрібно.

Центр ідентифікації (Authentication Centre – AuC) пов'язаний з HLR і призначений для захисту від несанкціонованого доступу (як підслуховування, так і використання чужих рахунків). Містить шифрувальні послідовності, алгоритми шифрування й генератор випадкових чисел.

Реєстр ідентифікації апаратури (Equipment Identity Register – EIR) є базою даних, в якій зберігають інформацію про апаратуру абонентських станцій. На відміну від аналогових мереж коміркового зв'язку, у мережі GSM окремо ідентифікують абонента та його абонентську станцію. Комутатор, пов'язаний з EIR, може заборонити використання абонентської станції, якщо останню, наприклад, оголошено як викрадену.

Абонентські станції (рухомі станції, мобільні станції, термінали) призначені для зв'язку абонента з мережею GSM і є єдиною частиною мережі, доступною для пересічного абонента.

Підсистема експлуатації та обслуговування призначена для диспетчеризації та адміністрування мережі GSM – введення й оновлення даних про абонентів, локалізації та усунення пошкоджень, ведення статистики, оцінювання інтенсивності зв'язку, нарахування оплати. Вона з'єднана з комутаторами (а через них – з базовою станцією та абонентськими станціями), з одного боку, й через комп'ютерний стик – з обслуговуючим персоналом – із іншого боку. Ця підсистема кладається з розподіленої структури центрів експлуатації та обслуговування (Operation and Maintenance Centre – OMC).

Просторова структура мереж коміркового зв'язку, зумовлена необхідністю роботи з рухомими абонентами, ґрунтується на відповідності зон покриття різним функціональним групам апаратури й утворює ієрархічну структуру з таких п'яти вкладених рівнів (від нижчого до вищого): комірка (cell); зона виклику (Location Area – LA); зона комутатора (MSC service area); система (Operator service area); мережа (Network Area).

Таку просторову структуру на прикладі мережі GSM схематично зображена на рис. 6.4. Різні мережі (у наведеному випадку мережі GSM) відповідно позначено GSM1...GSM3.

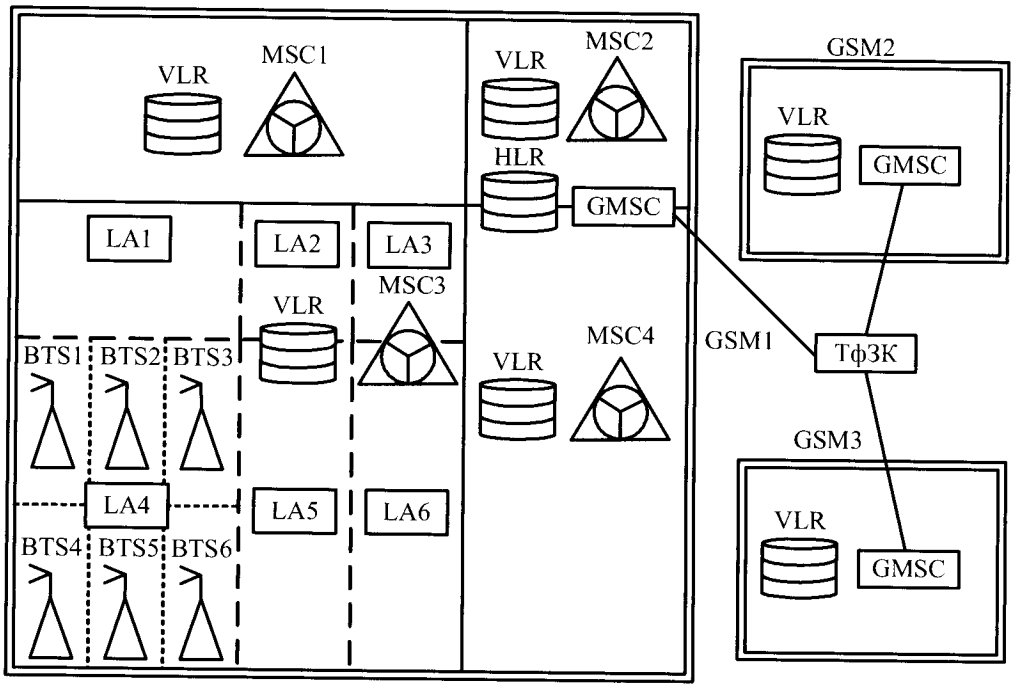


Рис. 6.4. Просторова структура мережі GSM

Комірка є найменшим фрагментом мережі GSM. Кожній комірці відповідає одна базова станція (BTS).

Зона виклику – це група сусідніх комірок, пересуваючись якою немає потреби оновлювати дані про місцезнаходження абонента. Сигнал виклику абонента передають усі базові станції зони виклику одночасно. У зону виклику об'єднують базові станції одного або декількох контролерів, але обов'язково з'єднаних з одним комутатором.

При переході абонента в іншу зону виклику того самого комутатора оновлюють адресу абонента в VLR, а при переході в зону іншого комутатора оновлюють адресу в HLR. Із зменшенням зони виклику збільшують частоту оновлення записів адрес у реєстрах (і обсяг переданої службової інформації) і зменшують кількість одночасно переданих сигналів виклику одного абонента (і обсяг сигнальної інформації). Мінімальний розмір LA збігається з коміркою, максимальний – із зоною комутатора.

Зона комутатора охоплює всі комірки, з'єднані з одним комутатором (MSC). Адресу зони комутатора, в якій знаходиться абонент, зберігають у HLR, а детальнішу адресу в середині зони – у VLR того комутатора, в зоні якого знаходиться абонент.

Система охоплює зону покриття, на території якої апаратуру GSM адмініструє один оператор. Якщо на території країни діють декілька операторів, то територіально системи операторів перекриваються, але між собою оператори спілкуються тільки через телефонну мережу загального користування (ТфЗК).

Мережа GSM – це вся поверхня земної кулі, охоплена послугами GSM. Географічно відповідає території всіх країн, в яких працюють оператори GSM. Мережа GSM пов'язана з іншими телекомунікаційними мережами через транзитні комутатори.

6.2.2. Захист від несанкціонованого доступу

Процедура ідентифікації в стандарті GSM використовує алгоритм генерації A3 і таємний ключ Ki (identification key), які записані в ***ідентифікаційному модулі абонента*** (Subscriber Identification Module – SIM) (рис. 6.5).

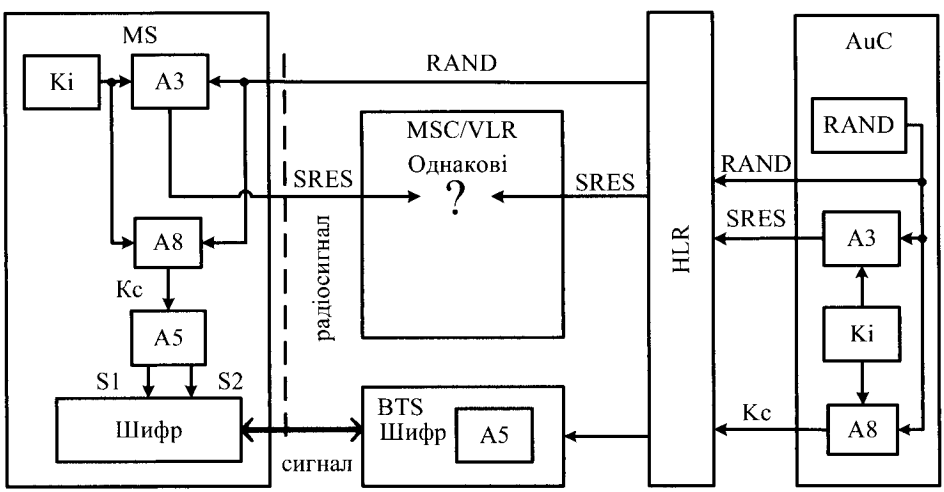


Рис. 6.5. Елементи мережі GSM, що беруть участь у криптографічних процедурах

Доступ до модуля SIM абонент отримує тільки після набору на клавіатурі 4-цифрового ***персонального ідентифікаційного коду*** (Personal Identity Number – PIN). Триразове введення неправильного PIN блокує карту SIM, і розблокування можливе введенням 12-цифрового ***персонального розблокувального коду*** (Personal Unblocking Key – PUK). Після неправильного введення PUK розблокувати SIM-карту може тільки оператор мережі.

Після ввімкнення абонентська станція отримує ***псевдовипадкове число*** (RANDom number – RAND), згенероване в центрі ідентифікації (AuC), формує ***електронний підпис*** (Signed RESponse – SRES) і скеровує його комутаторові (MSC).

Комутатор порівнює числа SRES, отримані від абонентської станції й центру ідентифікації, і за умови їх збігу надає абонентові доступ до мережі. Стандарт GSM визначає тільки розрядність чисел RAND (128 бітів) і SRES (32 біти), що відповідає 1036 вхідних і 1010 вихідних комбінацій. Алгоритм A3 нестандартизований, і кожен оператор відповідає за його вибір або розроблення, а також надійність.

6.2.3. Захист від підслуховування

Захист від підслуховування в радіоканалі також ґрунтується на принципі електронного підпису, який використовують для отримання шифрувального ключа Кс та для генерування шифрувальних послідовностей S1 і S2 (рис. 6.6).

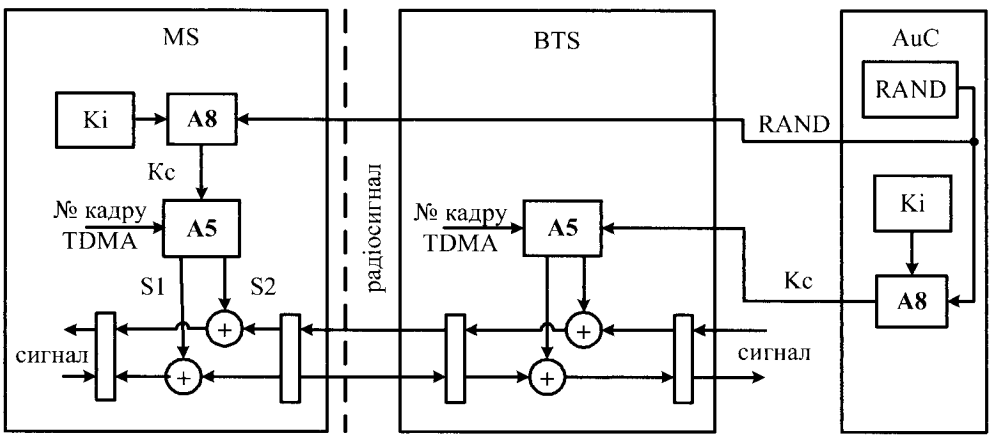


Рис. 6.6. Шифрування інформаційних сигналів

Алгоритм A8 на підставі числа RAND і ключа Ki генерує ключ Кс (завдовжки 64 біти). Алгоритм A5 на підставі ключа Кс і 22-бітового номера кадру в мережі із часовим розділенням каналів (Time division multiple access – TDMA) генерує дві шифрувальні послідовності для передавання "догори" (S1) і "донизу" (S2), кожна завдовжки 114 бітів. Номер кадру повторюють із періодом гіперкадру (приблизно 3,5 годин). Отже, шифрувальні послідовності змінюють у кожному кадрі.

Процес шифрування й дешифрування радіоповідомлення полягає в додаванні за модулем 2 побітово 114 інформаційних бітів основного пакета в інформаційному каналі та 114 бітів шифрувальної послідовності. Цю логічну операцію називають "виключне АБО" (XOR – eXclusive OR). Алгоритм A8 генерування ключа Кс нестандартизований (як і A3), а алгоритм A5 стандартизований, зберігається на всіх SIM-картах і базових станціях, а відповідальність за його нерозголошення несуть оператори мереж.

Інформація про алгоритми є конфіденційною, але деякі загальні характеристики алгоритму А5 відомі. Алгоритм А5 побудований на трьох регістрах зсуву зі зворотним зв'язком, розрядність регістрів становить 19, 22 і 23 біти. Сумарна кількість розрядів становить 64, і як початкову установку до цих регістрів заносять 64-розрядний ключ Кс, згенерований алгоритмом А8. Для оцінювання якості шифрування розмов у табл. 6.1 наведено час, необхідний для зламу ключів різної довжини повним перебором (брутальна атака).

Таблиця 6.1

Час, необхідний для зламу ключів різної довжини

Довжина ключа	32 біти	40 бітів	56 бітів	64 біти	128 бітів
Час перебору	1,2 години	12,7 днів	2,3 року	584,5 року	$10{,}8 \cdot 10^{24}$ років

З табл. 6.1 видно, що час підбору ключа є значним не тільки з погляду підслуховування в реальному часі, а й у випадку спроб розшифровування раніше записаного з радіоканалу сеансу зв'язку. Навіть за припущення, що використана довжина ключа становить 40 бітів замість 64, на момент закінчення дешифрування інформація вже втратить свою важливість, яка, за різними оцінками, зберігається декілька тижнів. Отже, стандарт GSM забезпечує надійний короткотерміновий захист інформації.

Порядок обміну й збереження криптографічної інформації простежимо за рис. 6.3. Для кожного абонента материнський центр AuC генерує ідентифікаційний триплет (ciphering triplet) – набір чисел (RAND, Кс, SRES). Із цього набору через радіоканал передають тільки 128-бітовий випадковий запит RAND. Триплет зберігають у реєстрі HLR і на вимогу надсилають у VLR комутатора, у зоні якого знаходиться абонент. Під час кожного виклику генерують новий триплет, а якщо з'єднання не відбулося, триплет зберігають у HLR і VLR. Для кожного абонента реєстри зберігають декілька (не менше одного) триплетів, які використовують у наступних з'єднаннях.

6.2.4. Конфіденційність локалізації абонента

Проблема конфіденційності інформації про місцезнаходження абонента пов'язана з тим, що велику кількість інформації, зокрема й номер абонента, передають на частоті радіомаяка без шифрування. Шифрування починають лише після закінчення ідентифікації за спеціальною командою з базової станції. Тому теоретично радіопідслуховуванням можна визначити номери абонентів, які вмикаються або активізуються в певній комірці, якщо потрапити власне на момент початкової ідентифікації.

Проблему вирішено наданням абонентові ***тимчасового номеру*** (Temporary Mobile Subscriber Identity – TMSI), який діє тільки в межах зони виклику. Надання й звільнення номерів TMSI здійснює комутатор (MSC), котрий у разі необхідності зв'язується з іншими комутаторами стаціонарними лініями. Інформацію про наданий номер передають абонентові в захищеному режимі, наприклад, при оновленні інформації про розташування. З моменту надання TMSI в радіоканалі передають лише цей номер, його підслуховування не дає інформації про абонента. У виняткових випадках комутатор може потребувати від абонентської станції повідомлення Міжнародного номера рухомого абонента (IMSI – International Mobile Subscriber Identity) – коли в VLR відсутній поданий абонентською станцією номер TMSI.

Побічним ефектом приховування локалізації абонента є необхідність завжди при виклику зв'язуватися з його материнським транзитним комутатором (і оплачувати відповідні міжнародні дзвінки). Наприклад, якщо в Україні зустрілися абоненти українського (У) і німецького (Н) операторів, то дзвінок від У до Н буде оплачений за міжнародним тарифом, а від Н до У – за внутрішнім українським. Якщо один абонент українського оператора (У1) викликає іншого абонента українського оператора (У2), який на момент виклику знаходиться за кордоном, то абонент У2 має оплатити міжнародний вхідний дзвінок, інакше абонент У1 дізнається про виїзд абонента У2 за сумою рахунку. Якщо ж у Німеччині зустрілись два абоненти українського оператора У1 і У2, то зв'язок між ними потребуватиме міжнародних з'єднань, незважаючи на фактичну близькість у просторі.

Розвиток послуг роумінгу за міжнародними домовленостями операторів урізноманітив варіанти зв'язку й тарифікації. Зокрема абонент має можливість не приховувати свого переміщення в зону іншого оператора – вхідні дзвінки для нього будуть безкоштовними, а оплату міжнародного з'єднання здійснюватиме викликаючий абонент, якого інформують відповідним повідомленням. Інший напрямок розвитку роумінгу – значне здешевлення міжнародних з'єднань із використанням протоколу IP-телефонії (Internet protocol).

6.2.5. Особливості забезпечення конфіденційності в мережах CDMA

Якщо описані в стандарті GSM процедури генерування псевдовипадкових послідовностей та кодування ними інформаційного повідомлення є додатковими й запроваджені виключно для вирішення питань ідентифікації та захисту інформації в радіоканалі, то для мереж стандарту CDMA ці процедури є основним принципом функціонування. Для цих мереж проблему захисту вирішено в той самий спосіб, що й розділення каналів.

Нагадаємо, що кожний інформаційний біт передають за допомогою 128 чипів у мережі IS-95 та від 4 до 256 чипів у CDMA 2000 та W-CDMA. У IS-95 та CDMA 2000 використовують псевдовипадкові послідовності трьох типів, а саме:

1. Функції Уолша завдовжки 64 (змінної довжини в CDMA 2000 та W-CDMA) – для утворення каналів.

2. Короткий код на основі послідовності Голда, утворений із двох m-послідовностей завдовжки $2^{15}-1 = 32\,767$ – для ідентифікації базової станції (фактично використовують одну з 512 унікальних адрес, утворених різними фазовими зсувами короткого коду).

3. Довгий код, тобто m-послідовність завдовжки $2^{42}-1 = 4\,398\,046\,511\,103$ – для ідентифікації абонента в кожному сеансі зв'язку.

Функції Уолша відомі, як відомими є частотні та часові характеристики фізичних каналів стандарту GSM, і тому вони не можуть забезпечувати захист передавання.

Породжувальні поліноми короткого коду також відомі, а його порівняно невеликий період (25,43 мс) дає можливість виконати повний перебір та виявити фазовий зсув базової станції за 14 хвилин. Крім того, кожна базова станція постійно передає свій короткий код через відкритий пілотний канал.

Породжувальний поліном довгого коду також відомий, але його період за чипової швидкості 1,2288 Мчип/с становить 39,5 діб, що значно перевищує розумну тривалість сеансу й унеможливлює підслуховування в реальному часі. Спроба кореляційного аналізу перебором зайняла би близько 10^{10} років, що нереально. Для генерування відрізку довгого коду, який використовують для шифрування, перед кожними сеансом до 42-розрядного регістра зсуву заносять початкове значення – 42-бітову маску коду. За своїм призначенням ця маска аналогічна до ключа Кс у стандарті GSM. Маску ніколи не передають через радіоканал – її формують перед кожним сеансом окремо в абонентській станції на підставі ключа Кі та запиту RAND і окремо в центрі ідентифікації AuC на підставі тих самих даних. Використання породженого відрізка довгого коду аналогічне до використання послідовностей S1 і S2, зображеного на рис. 6.6, і забезпечує конфіденційність у радіоканалі.

6.2.6. Ідентифікація апаратури абонента

Необхідність захисту від використання незареєстрованої апаратури спричинена, з одного боку, турботою про якість зв'язку, якій може зашкодити несправна або саморобна абонентська станція, а з іншого – бажаністю виявлення спроб використання викраденої апаратури.

Міжнародний ідентифікатор мобільного обладнання (International Mobile Equipment Identity – IMEI) записує виробник у постійну пам'ять мобільного телефону. Виробникам номери IMEI надають після процедури сертифікації в ***міжнародній організації операторів стандарту GSM*** (GSM MoU Association). Про можливість появи нових номерів апаратури секретаріат GSM MoU повідомляє операторів, і вони зберігають ці номери в реєстрі ідентифікації апаратури (EIR).

Ідентифікацію апаратури здійснюють у чотири кроки:

1. Комутатор (MSC) надсилає до абонентської станції команду з вимогою повідомити IMEI.
2. Абонентська станція надсилає свій номер IMEI комутаторові.
3. Комутатор надсилає номер IMEI в реєстр EIR, де номер перевіряють на входження до білого списку (всі зареєстровані на Землі номери), сірого списку (розглядають можливість блокування) чи чорного списку (заблоковані).
4. Реєстр EIR надсилає комутаторові результати ідентифікації, що зумовлює допущення чи недопущення терміналу до роботи в мережі.

Незалежно від результатів ідентифікації апаратури, навіть з апаратів із відсутнім чи заблокованим модулем SIM можна здійснювати аварійні виклики (emergensy call): пожежної служби, міліції, швидкої допомоги тощо.

6.3. Інформаційна безпека комп'ютерних мереж

6.3.1. Види комп'ютерних мереж та основи їх функціонування

Комп'ютерна мережа – телекомунікаційна мережа, що об'єднує два чи більше комп'ютерів. Вона складається з ліній зв'язку та вузлів зв'язку, у яких розміщено мережеве обладнання [1–26].

Сучасні комп'ютерні мережі є мультисервісними, тобто забезпечують передавання відеозображень, графічних зображень, мовних сигналів, даних тощо. Такі мережі у поєднанні із засобами для накопичування, оброблення, зберігання та розповсюджування даних (серверів баз даних, файлових серверів, серверів додатків тощо) утворюють ***інформаційні системи***. Якщо такі системи ґрунтуються на поєднанні інформаційних та телекомунікаційних технологій, їх ще називають ***інформаційно-комунікаційними системами*** (ІКС), або ***інформаційно-комунікаційними мережами*** (ІКМ).

За територіальною ознакою мережі поділяють на:

– глобальні мережі (Global Area Network – GAN);

– великі територіальні мережі (Wide Area Network – WAN);

– міські мережі (Metropolitan Area Network – MAN);

– локальні мережі (Local Area Network – LAN).

Найчастіше використовують поняття локальних та глобальних мереж. Локальні мережі забезпечують користувачам (клієнтам) доступ до розподілених ресурсів, які знаходяться на інших комп'ютерах (серверах) і становлять основу для побудови приватних (корпоративних) мереж. Глобальні мережі забезпечують своїх користувачів засобами зв'язку для передавання різних видів трафіку на великі відстані. Користувачами глобальної мережі можуть бути як окремі комп'ютери й різноманітне термінальне обладнання із вбудованими процесорами (касові апарати, банкомати, вимірювальні системи тощо), так і локальні мережі. Слід зазначити, що в загальному випадку ***клієнтом*** називають комп'ютер, який формує запит на доступ до мережевих послуг (наприклад, потребує доступу до мережевих ресурсів), а ***сервером*** – комп'ютер, який надає мережеві послуги (наприклад, надає доступ до необхідних мережевих ресурсів).

І глобальні, і локальні мережі будують за певними стандартизованими технологіями, в основу яких покладено розроблену під егідою ***Міжнародної організації зі стандартизації*** (Internationl Organization for Standartization – ISO) ієрархічну семирівневу ***Еталонну модель взаємодії відкритих систем*** (Open System Interconnection – OSI) (табл. 6.2).

Таблиця 6.2

Рівні моделі OSI/ISO

Номер рівня	Назва рівня
7	Прикладний рівень (Application layer)
6	Рівень представлення (Presentation layer)
5	Сеансовий рівень(Session layer)
4	Транспортний рівень (Transport layer)
3	Мережевий рівень (Network layer)
2	Канальний рівень (Data Link layer)
1	Фізичний рівень (Physical layer)

У моделі OSI процедуру взаємодії двох відкритих систем (комп'ютерів, вузлів мережі) описують у вигляді набору правил (протоколів) взаємодії кожної пари модулів відповідних рівнів цих систем. Принципи побудови, технології та протоколи мереж визначено в міжнародних стандартах та детально описано в літературі [4]

З появою стеків (сімейств) комунікаційних протоколів з'явилася можливість об'єднувати в одну велику мережу мережі, які відрізняються своєю структурою, форматом повідомлень, швидкістю передавання даних, інтер-

фейсом фізичного рівня тощо. До складу такої об'єднаної мережі можуть входити побудовані за різними технологіями як локальні, так і глобальні мережі, зв'язок між якими забезпечують спеціальні комунікаційні пристрої – маршрутизатори, які підтримують протоколи мережевого рівня стеку комунікаційних протоколів. Об'єднану мережу ще називають великою, складеною, або ***інтернет-мережею*** (internet). Прикладом об'єднаної мережі є загальновідома інформаційна система – всесвітня мережа ***Інтернет*** (Internet), яка використовує стек комунікаційних протоколів ***TCP / IP*** (Transmission Control Protocol / Internet Protocol) і охоплює всю земну кулю.

Передавання даних між користувачами різних мереж об'єднаної мережі, яка використовує стек TCP / IP, забезпечують протоколи транспортного рівня й рівня міжмережевої взаємодії (транспортного й мережевого в моделі OSI). Для адресації користувачів об'єднаної мережі стек TCP / IP використовує ***числову складену (ієрархічну) IP-адресацію***, при якій кожній окремій (LAN чи WAN) мережі присвоюють свій унікальний номер (***ідентифікатор мережі***), а кожному вузлу цієї мережі – свій номер (***ідентифікатор вузла***). Поділ IP-адреси завдовжки чотири байти, яку використовує протокол IPv4, між ідентифікатором мережі й ідентифікатором вузла здійснюють за її класом або маскою, яка містить "1" у тих розрядах, які належать ідентифікатору мережі.

На прикладному рівні, мережеві служби якого забезпечують виконання задач користувача, стек протоколів TCP / IP дає змогу використовувати текстову ієрархічну доменну систему імен, яка має деревоподібну структуру й допускає використання довільної кількості складових (до 256 символів). На рівні мережевих інтерфейсів стек TCP / IP дає можливість у кожній окремій мережі використовувати протоколи фізичного й канального рівнів тих технологій, за якими вони побудовані, а їхнім вузлам – використовувати унікальні апаратні адреси. Так, вузли локальних мереж Ethernet використовують унікальні MAC-адреси своїх мережевих адаптерів.

Обмін даними у вигляді кадрів (фреймів, пакетів) конкретної технології між користувачами однієї мережі здійснюють за їхніми апаратними адресами. Обмін даними між користувачами різних мереж у складі об'єднаної мережі здійснюють у вигляді IP-пакетів за допомогою маршрутизаторів, робота яких ґрунтується на протоколах рівня мережевого інтерфейсу та міжмережевого рівня стеку TCP / IP. При цьому портам мережевого маршрутизатора присвоюють апаратні та IP-адреси тих мереж, до яких їх під'єднано, і кожний порт використовує протоколи своєї мережі. Маршрутизацію IP-пакетів здійснюють за допомогою маршрутизаторів на основі аналізу таблиць маршрутизації, записи в яких створюють та поновлюють протоколи маршрутизації міжмережевого рівня.

Відповідність між доменними іменами та IP-адресами можна встановлювати як засобами локального комп'ютера, так і за допомогою централізованої ***служби присвоєння адрес*** (DNS – Domain Name System, служба імен доменів) типу "клієнт-сервер", яка функціонує на прикладному рівні і є характерною для об'єднаних мереж. При цьому виділені DNS-сервери підтримують розподілену базу відображень, а DNS-клієнти звертаються до серверів із запитами на відображення доменного імені в IP-адресу. Службу DNS об'єднаної мережі побудовано за ієрархічним принципом. Кожний сервер DNS зберігає таблиці відображення імен свого домена й посилання на DNS-сервери своїх піддоменів.

Служба встановлення адрес, що працює на основі ***протоколу розрізнення адрес*** (ARP – Adress Resolution Protocol), установлює відповідності між IP-адресами й апаратними адресами вузлів мережі на основі аналізу arp-таблиць. Arp-таблиці містять дані про відповідність IP-адрес вузлів мережі їх локальним адресам із позначенням типу запису – динамічний чи статичний запис. Статичні записи виконує адміністратор мережі з використанням утиліти ***arp***. Динамічні записи будують ARP-служби, посилаючи в мережу широкомовні запити. Динамічні записи, на відміну від статичних, мають обмежений термін існування.

Більшість приватних (корпоративних) мереж будують за однією або декількома технологіями LAN, використовуючи поширену мережеву операційну систему Windows NT або Linux та стек комунікаційних протоколів TCP / IP. Приватні мережі можуть належати до класу ***інтернет-*** або ***інтранет-мереж*** (intranet), тобто з правом прямого виходу або без права прямого виходу в об'єднану мережу Інтернет. Для виходу в мережу Інтернет приватна мережа повинна отримати унікальну IP-адресу від організації, яка має право на надання відповідних послуг і є ***точкою входу*** (точкою присутності, Point of Presence – POP) у мережу Інтернет. Це може бути регіональний ***Інтернет-провайдер*** (постачальник послуг мережі Інтернет, інтернет-сервіс-провайдер, Internet Service Provider – ISP, у загальному випадку – провайдер телекомунікацій), який, своєю чергою, повинен отримати діапазон адрес для надання їх користувачам в ***Адміністрації призначених інтернет-номерів*** (Internet Assigned Numbers Authority – IANA), або ж в організації ***Інформаційний центр мережі Інтернет*** (Internet Network Information *Center – InterNIC*).

Виділення вузлам індивідуальних адрес у межах виділеної Інтернет-провайдером IP-адреси мережі забезпечує служба на базі ***протоколу конфігурації динамічного хоста*** (Dynamic Host Configuration Protocol – DHCP), яка може функціонувати як на базі виділеного комп'ютера, так і на базі маршрутизатора. Разом з IP-адресою й маскою DHCP-сервер повідомляє вузлу адресу найближчого маршрутизатора (шлюзу в зовнішню мережу), адресу DNS-сервера й час оренди адреси.

Зв'язок з інформаційною системою користувачів, які знаходяться на значній відстані від мережі, забезпечують за допомогою технологій віддаленого доступу. Одним із найперших видів віддаленого доступу був комутований доступ через канали ***телефонних мереж загального користування*** (ТфЗК, Public Switched Telephone Network – PSTN). Застосування цифрових абонентських ліній технології xDSL забезпечує доступ віддалених абонентів (користувачів) до комп'ютерних мереж телефонними лініями зв'язку й дає змогу передавати різні типи даних та мультимедійного трафіку. ***Кабельні системи телевізійних мереж*** (Cable television, Community Antenna Television, *CATV* – телебачення із загальною антеною) забезпечують віддалений доступ до мережі Інтернет для своїх абонентів за допомогою кабельних модемів. Останнім часом усе ширшого застосування в системах віддаленого доступу знаходить безпроводовий зв'язок, який для передавання даних використовує електромагнітні хвилі різних діапазонів частот. Класифікація й особливості застосування різних видів безпроводового зв'язку буде розглянуто нижче.

Сьогодні ***всесвітня павутина*** (World Wide Web – WWW, web, веб) містить розміщену на веб-серверах мережі Інтернет величезну за обсягом інформацію з різних видів людської діяльності, науки, техніки, природи, особистих даних тощо. Важливою особливістю цієї інформації є простота доступу до неї та її активація за запитом комп'ютера користувача. Спеціальні програмні засоби пошуку та гіперпосилань дають можливість користувачам Інтернет оперативно отримувати від комп'ютерів-серверів потрібну їм інформацію.

Нагромадження в мережах великих обсягів різноманітної (зокрема й конфіденційної) інформації та під'єднання до них об'єктів критичної інфраструктури породило новий вид кримінальної діяльності – ***кібертероризм***. Якщо в двадцятому столітті питання безпеки в мережі Інтернет зводилося переважно до захисту банківської та особистої інформації, то з появою кібертероризму на передній план вийшли проблеми захисту як приватних, так і публічних інформаційних систем.

6.3.2. Інциденти інформаційної безпеки

Інцидент (incident, здійснена загроза) – це дія зловмисника, яка завдає шкоди програмно-апаратним засобам комп'ютерної мережі та конфіденційності, цілісності й доступності комп'ютерних даних. Згідно із запропонованою Конвенцією Ради Європи класифікацією кібернетичних втручань і загроз інциденти поділяють на такі види:

– несанкціонований доступ до інформаційного середовища (протиправний навмисний доступ до комп'ютерної мережі або її частини, здійснений в обхід систем безпеки);

– втручання в роботу мережі (протиправне порушення або створення перешкод функціонуванню комп'ютерної мережі завдяки розробленню та поширенню вірусного програмного забезпечення, застосування апаратних закладок, радіоелектронного та інших видів впливу на технічні засоби й телекомунікаційні системи);

– перехоплення (протиправне навмисне аудіовізуальне або електромагнітне перехоплення не призначених для загального доступу комп'ютерних даних в обхід заходів безпеки);

– незаконне використання комп'ютерного й телекомунікаційного обладнання або його повне вилучення.

Розрізняють внутрішні й зовнішні інциденти. Внутрішні інциденти виникають внаслідок шкідливих дій осіб, безпосередньо пов'язаних із постраждалою комп'ютерною мережею. До найпоширеніших внутрішніх інцидентів належать неправомірний доступ до конфіденційної інформації та її витік, вилучення інформації та її використання в протизаконних операціях. Під зовнішнім інцидентом розуміють подію, джерелом якого є порушник, безпосередньо не пов'язаний із постраждалою комп'ютерною мережею. До подій такого типу належать вірусні атаки на програмне та технічне забезпечення мережі, атаки типу "відмова в обслуговуванні", перехоплення трафіку, неправомірний доступ до конфіденційної інформації та її розміщення в мережі Інтернет, сканування порталу корпоративної мережі, шахрайство в системах електронного документообігу тощо.

Мережу Інтернет зловмисники все частіше використовують для розсилання фальшивих електронних листів із метою шантажу та шахрайства, організації бот-мереж для здійснення атак на приватні сайти, розповсюдження нових видів шпигунських програм. Програми-шпигуни призначені для виявлення та викрадення конфіденційної інформації. Серед них найпоширеніші сканери даних із портів комп'ютерів, клавіатури та екрана. У комп'ютерних мережах кібершпигуни аналізують трафік, електронну пошту, сайти, розмови через звукову карту тощо.

Для організації атак на інформаційну систему зловмисники часто використовують ***вразливості*** програмного забезпечення мережі, які дають змогу отримати несанкціонований доступ до даних. Для виявлення вразливостей системи зловмисники останнім часом використовують набір спеціальних програм – сканерів вразливостей відомої платформи ***Burp Suite***. Платформа містить ряд розширень, які забезпечують моніторинг системи, тестування програмного забезпечення, складання карти веб-додатків, пошуку файлів і папок, модифікації запитів, підбору паролів тощо. Отримані результати використовують для створення експлойтів. ***Експлойт*** (exploit – експлуатува-

ти) – це розширення шкідливої програми, яке використовує виявлені вразливості в роботі програмного забезпечення мережі та призначене для проведення атаки на інформаційну систему. Метою атаки може бути і захоплення контролю над мережею з усіма можливими наслідками, і порушення її функціонування (наприклад, організуванням DoS-атаки). Атаки можуть бути напрямлені на системне та прикладне програмне забезпечення, інформаційні дані, браузери, інтернет-сайти тощо. Залежно від способу отримання доступу до вразливого програмного забезпечення та їх запуску експлойти можуть бути як зовнішніми (віддаленими), так і внутрішніми (локальними).

В останні роки зловмисники почали активно застосовувати методи ***моніторингу відкритих і відносно відкритих джерел*** (МВВД) та ***соціальної інженерії*** (СІ). Моніторинг відкритих і відносно відкритих джерел полягає в збиранні різної інформації про об'єкт розвідки з її подальшим опрацюванням та підготовленням оперативних дій. Соціальна інженерія використовує різноманітні способи, спрямовані на отримання зловмисником доступу до конфіденційної інформації завдяки зовнішньому впливу на працівників об'єкта атаки та наступним провокуванням внутрішнього інциденту.

Для підготовки атаки агенти соціальної інженерії користуються переважно такими засобами, як фішинг, претекстинг і бейтинг. ***Фішинг*** (Phishing – риболовля) – використання фальшивих веб-сайтів легальних організацій, соціальних мереж та електронних повідомлень користувачам із метою їх дезорієнтації й спонукання до розкриття своїх персональних даних. ***Бейтинг*** (To bait – підгодовування, використання наживки) – знайомство через соціальні мережі, спеціально створені веб-сайти та форуми з фахівцями тих галузей, які цікавлять агентів соціальної інженерії, та отримання від них конфіденційної інформації. ***Претекстинг*** (Pretexting – привід) – дії, за яких зловмисник, видаючи себе за іншу людину та використовуючи телефон, електронну пошту або програму ***Skype*** мережі Інтернет, згадуючи імена реальних людей, відомі події та дати, входить у довіру до жертви й отримує від неї інформацію, яка його цікавить.

Разом із тим, методи соціальної інженерії передбачають використання різних закладок (пристроїв підслуховування, відео- та аудіоспостереження тощо), замаскованих під предмети побуту, аксесуари індивідуального користування, іграшки тощо.

Для доступу до інформаційних ресурсів комп'ютерних мереж зловмисники можуть використовувати технічні канали витоку інформації. Технічні канали витоку інформації поділяють на канали первинних електромагнітних випромінювань, канали вторинних наведень у навколишніх конструкціях і системах комунікацій, акустичні й оптичні канали. Ці канали зловмисники

можуть використовувати для несанкціонованого доступу до конфіденційної інформації в процесі її передавання, опрацювання та зберігання.

Останніми роками зловмисники активно використовують методи соціальної інженерії для проведення шахрайських дій із картками банківських клієнтів, тобто виманюють реквізити банківських карток або заохочують власників платіжних карток перерахувати гроші на свої рахунки. Шахраї публікують на сайтах оголошення про продаж за передоплатою товарів за привабливою ціною або спеціально створюють інтернет-магазини із продажу неіснуючих товарів. Серед банкоматного шахрайства зловмисники широко використовують кеш-трепінг і скімінг. ***Кеш-трепінг*** (Cash Trapping – захоплення готівки) – це встановлення на отвір для видавання готівки банкомата шахрайських пристроїв, які унеможливлюють отримання готівки жертвою й дають змогу зловмиснику отримати готівку після відходу жертви від банкомата. ***Скімінг*** (Skimming – проглядати) – установлення на картрідер (зчитувач карток) банкомату спеціальних пристроїв, які дають можливість зловмисникам копіювати магнітну смугу платіжної картки. При цьому ***персональний ідентифікаційний номер*** (Personal Identification Number – PIN-код) платіжної картки шахраї отримують за допомогою встановленої на банкоматі мініатюрної відеокамери розміром менше голівки сірника.

У процесі проникнення інформаційних технологій до усіх сфер діяльності людини залежність кожної людини від інформаційних систем і мереж, а також її вразливість щодо стороннього кібернетичного впливу постійно зростають. Сучасний кібертероризм вийшов за межі викрадення секретної інформації та отримання фінансової вигоди від проведення атаки. Розміщення неправдивої (фейкової) інформації на спеціально створених сайтах (consumer opinion sites), а також поширення небезпечного контенту (вмісту) через соціальні мережі, форуми, блоги, електронну пошту може не лише спровокувати соціальні заворушення та паніку серед населення, а й запустити незворотні процеси в державних масштабах (зруйнувати банківську та фінансову системи держави, вплинути на роботу об'єктів критичної інфраструктури тощо).

Діяльність злочинців у сфері інформаційних технологій характеризується відсутністю національних кордонів, а їхні дії можуть бути спрямовані як на приватні, так і на урядові об'єкти. У міжнародних стосунках на заміну військовим діям із застосуванням армійських підрозділів усе частіше приходять інформаційні та гібридні війни, коли активно використовують кібершпигунство та кібератаки на об'єкти критичної інфраструктури. Зловмисники спрямовують свої атаки на урядові сервери, сервери політичних партій, громадських організацій, виборчих комісій тощо.

Яскравим прикладом міжнародного кібертероризму є проведення атак за допомогою відомих шкідливих програм-вірусів ***WannaCry*** та ***Nyetya***, які використали вразливості системного й прикладного програмного забезпечення інформаційних систем. Перший вірус у 2016 році масово шифрував дані заражених комп'ютерів із метою отримання викупу, а другий улітку 2017 року атакував об'єкти критичної інфраструктури багатьох країн у різних частинах світу. В обох випадках заражені комп'ютери були атаковані через електронну пошту. Міжнародного розголосу набули викрадення зловмисниками конфіденційної інформації з сайтів державних установ, використання соціальних мереж для впливу на результати виборів до державних органів влади у ряді європейських країн та США, результати проведення референдумів тощо.

Успішна боротьба з міжнародним кібертероризмом можлива лише за умови тісної співпраці національних і міжнародних правоохоронних органів із провідними компаніями в галузі інформаційних технологій та кібербезпеки.

6.3.3. Принципи організації безпеки комп'ютерних мереж

Безпека комп'ютерної мережі передбачає організацію такого її стану, за якого її інформаційні й програмно-технічні ресурси, а також допоміжна інфраструктура будуть захищеними від шкідливого впливу і звичайних користувачів, і зловмисників. Захист мережі від несанкціонованих дій із боку як внутрішнього, так і зовнішнього середовища забезпечують політикою інформаційної безпеки. Організація безпеки комп'ютерної мережі ґрунтується на дотриманні ряду принципів, найважливішими з яких є:

– надання працівникам підприємства чи організації мінімальних прав, достатніх для виконання ними своїх обов'язків;

– забезпечення надійного захисту всіх рівнів багаторівневої інформаційної системи;

– використання єдиного міжмережевого екрана для захисту внутрішньої корпоративної (приватної) мережі від зовнішніх загроз;

– блокування входу в корпоративну мережу при виході з ладу міжмережевого екрана;

– використання захищених каналів для зв'язку з користувачами, які знаходяться в зовнішніх мережах;

– шифрування трафіку безпроводових мереж;

– захист технічних каналів витоку інформації;

– регулярне резервне копіювання найважливіших даних.

Для здійснення цих принципів необхідно забезпечити автентифікацію взаємодіючих сторін, конфіденційність передавання інформації, підтвердження

достовірності й цілісності переданої інформації. Перераховані функції багато в чому пов'язані між собою, а їх здійснення основане на криптографічному захисті даних, які передають мережею.

Автентифікацію взаємодіючих сторін у сучасних комп'ютерних мережах здійснюють із використанням спеціальних протоколів (правил). Існує багато різних протоколів автентифікації, які забезпечують надійну ідентифікацію користувачів інформаційної мережі.

Найпоширенішими є протокол ***автентифікації за паролем*** (Password Authentication Protocol – PAP) і протокол ***автентифікації за запитом*** (Challenge Handshake Authentication Protocol – CHAP). Протокол автентифікації за паролем PAP вимагає від користувача вказати своє ім'я й пароль, які сервер порівнює із записаними в його пам'яті зашифрованими образами. Протокол автентифікації за запитом CHAP, який входить до сімейства протоколів "точка-точка" (Point-to-Point Protokol – PPP) використовують для автентифікації під'єднаних до Інтернету віддалених користувачів. За цим протоколом паролі користувачів зберігають у їх агентів автентифікації на сервері Інтернет-провайдера. При встановленні зв'язку користувач і агент автентифікації обмінюються послідовністю зашифрованих повідомлень. При отриманні позитивних результатів їх аналізу користувач отримує дозвіл на доступ до ресурсів мережі.

У мережах із великою кількістю користувачів замість автентифікації за паролем використовують автентифікацію за сертифікатами, які містять інформацію, що підтверджує особистість користувачів. Сертифікати видають користувачам спеціальні уповноважені організації – ***центри сертифікації*** (Certifikat Authority – CA). Користувачі у своїх запитах до серверів мережі вказують сертифікати із закритими ключами, за якими їх автентифікують.

Автентифікацію можуть застосовувати для підтвердження не лише особистості учасників сеансу зв'язку, але й достовірності документів, програмних та технічних засобів тощо. Так, деякі розробники програмних засобів із метою захисту авторського права вносять у свої програми спеціальні коди, які дають можливість автентифікувати їхні програми.

Одночасно з автентифікацією авторизують користувача. ***Авторизація*** забезпечує контроль доступу легальних користувачів до ресурсів мережі й надання їм прав, визначених адміністратором мережі. У деяких мережах використовують ідентифікацію фізичної особи. Найпоширенішим і надійним способом ідентифікації фізичної особи є її ідентифікація за біометричними параметрами: сітківкою ока, відбитком пальця, рисами обличчя, особливостями голосу тощо.

Конфіденційність передбачає, що лише відправник і отримувач пакета здатні зрозуміти зміст повідомлень, які передають мережею. Одним із найужи-

ваніших способів забезпечення конфіденційності повідомлень є їх шифрування й дешифрування з використанням одного або декількох ключів. Розрядність ключа визначає кількість можливих комбінацій, які можна використати для шифрування одного символу текстової інформації.

У сучасних мережах можна використовувати асиметричні й симетричні, відкриті та особисті ключі, моноалфавітні та поліалфавітні шифри, одинарне, подвійне та потрійне шифрування. З середини 90-их років для шифрування й дешифрування несекретних повідомлень почали використовувати стандарт ***DES*** (Data Encryption Standartd – стандарт шифрування даних). Шифр DES шифрує 64-розрядні блоки повідомлення за допомогою 64-розрядного ключа. Шифр ***3DES*** (потрійний DES) шифрує блоки повідомлення тричі з використанням трьох різних ключів. Від 2001 року в комп'ютерних мережах почали використовувати покращений стандарт шифрування ***AES*** (Advanced Encryption Standartd – удосконалений стандарт шифрування), який опрацьовує дані 128-розрядними блоками із ключами завдовжки 128, 192 і 256 бітів.

Цілісність повідомлень передбачає забезпечення незмінності повідомлення під час його пересилання мережею. Одним із найбільш надійних і розповсюджених засобів забезпечення цілісності повідомлень є використання цифрового підпису. Цифровий підпис підтверджує, що повідомлення зашифрував й надіслав насправді відправник, що ніхто інший надіслати його не міг, що відмовитися від свого підпису відправник не може. З використанням цифрового підпису трудомістке шифрування цілого повідомлення не є обов'язковим. Формують цифровий підпис часто хешуванням характерного блоку обмеженої довжини (дайджесту) повідомлення з його подальшим шифруванням за допомогою особистого ключа користувача. ***Хешуванням*** (гешування, hashing) називають перетворення з допомогою ***хеш-функції*** (геш-функція, hash function) повідомлення довільної довжини у вихідний бітовий рядок фіксованої довжини. Ця бітова послідовність носить назву ***хеш-код*** (геш-код, хеш-сума, хеш, hash) або ***дайджест повідомлення*** (message digest). Зміна вхідного тексту навіть на один символ повністю змінює результат обчислення хеш-коду. Для хешування повідомлень використовують велику кількість хеш-функцій, які відрізняються як розрядністю хешів, так і обчислювальними алгоритмами та криптостійкістю. Серед простих і надійних хеш-алгоритмів широкого застосування знайшли алгоритми, побудовані на використанні математичних операцій ділення із залишком, множення із застосуванням "золотого перерізу", коефіцієнтів поліномів тощо. Випадок, коли хеш-функція перетворює різні повідомлення на одинакові хеші, називають ***колізією*** (collision). На практиці хеш-функції часто вибирають із заданої множини за допомогою випадкових чисел, що дає можливість зменшити ймовірність виникнення колізій.

Цифровий підпис можна також використовувати для автентифікації користувача чи отриманих повідомлень.

Загалом захист комп'ютерних мереж від загроз із боку зовнішнього середовища передбачає якісне вирішення двох базових завдань:

1. Захист підключених до публічних каналів зв'язку локальних мереж і окремих комп'ютерів від несанкціонованих дій із боку зовнішнього середовища.

2. Захист інформації в процесі її передавання відкритими каналами зв'язку.

Захищають приватні мережі від загроз з боку публічної мережі за допомогою міжмережевих екранів (брандмауерів) та програмних засобів виявлення й знешкодження загроз. Міжмережеві екрани, функції яких залежать від їх типу, моделі та конкретної конфігурації, можуть застосовувати різні алгоритми опрацювання вхідних та вихідних даних і забезпечувати різні ступені захисту мережі від зовнішніх загроз.

Постійним аудитом мережі підвищують ступінь її захисту. Під час аудиту аналізують поведінку підозрілих об'єктів мережі, вносять відповідні записи в журнал реєстрації та ідентифікують порушника. Функції аудиту можуть виконувати різні засоби забезпечення безпеки мережі: мережевими моніторами, системами виявлення вторгнень, міжмережевими екранами, антивірусними програмами тощо.

Захищають дані під час їх передавання через публічну мережу відкритими лініями зв'язку створенням окремих ***захищених каналів*** або ***віртуальних приватних мереж*** (Virtual Private Network – VPN), які забезпечують виконання взаємної автентифікації користувачів під час встановлення логічного з'єднання, шифрування даних та підтвердження цілісності отриманих повідомлень.

6.3.4. Методи та засоби забезпечення вимог політики безпеки комп'ютерної мережі

Згідно з вимогами стандартів ISO 7498-4 та X.700 заходи, скеровані на підвищення безпеки мережі, впроваджує й супроводжує системний адміністратор, а у великих мережах – адміністратор безпеки мережі. Він здійснює статичні записи в таблицях адресації комутаторів, маршрутних таблицях вузлів (хост, гост) і маршрутизаторів, виділяє статичні IP-адреси вузлам мережі, визначає IP-адресу шлюзу за замовчуванням тощо. До його функцій входять реєстрація користувачів, надання їм пріоритетів, паролів, забезпечення доступу до мережевих ресурсів тощо.

Вимоги політики безпеки щодо захисту приватної мережі від внутрішніх та зовнішніх загроз можна забезпечити на різних рівнях стеку комунікаційних протоколів. На фізичному рівні захист приватних мереж від витоку інформації забезпечують різноманітними способами, зорієнтованими на тип каналу витоку. Політика інформаційної безпеки мережі передбачає використання таких заходів запобігання витоку інформації:

– розміщення комп'ютерного та допоміжного обладнання, а також розташування приміщень для офісних працівників у спеціалізованих екранованих будівлях;

– використання для екранування приміщень металевих сіток, металізованих вікон та побудова спеціальних екранованих приміщень ("капсул");

– прокладання проводів і кабелів мережі в екранованих оболонках, заміна неекранованих кабелів на волоконно-оптичні кабелі;

– установлення на лініях зв'язку та мережах електропостачання високочастотних фільтрів;

– використання екранованого як основного, так і допоміжного обладнання;

– установлення активних систем зашумлення навколишнього середовища;

– використання систем сканування для виявлення радіо- та відеозакладок.

Для створення електромагнітного шумового бар'єру використовують різноманітні широкосмугові генератори шуму. Ефективним є запаковування кабелів у наповнену інертним газом герметичну металеву трубу, у разі пошкодження якої зловмисником тиск газу понижується, що активізує відповідні давачі й викликає сигнал тривоги. Практично повністю позбутися електромагнітного випромінювання лініями зв'язку можливо з використанням волоконно-оптичних кабелів.

На канальному рівні мережі Ethernet використовують фільтрування кадрів комутаторами за MAC-адресами та технологію створення ***віртуальних локальних мереж*** (ВЛМ, Virtual LAN – VLAN). Стандарт IEEE 802.1Q описує побудову віртуальних мереж на основі одного або декількох ***комутаторів*** (switch). При побудові VLAN адміністратор за допомогою спеціальної програми закріплює порти комутаторів за номерами та іменами робочих груп. Обмін кадрами допускається між вузлами тільки в границях тієї групи, до якої вони належать.

На мережевому рівні з метою зменшення чутливості до широкомовного трафіку, полегшення адміністрування, зменшення завантаженості та покращення захищеності мережу з великою кількістю вузлів розбивають на декілька підмереж, тобто структуризують. Згідно з вимогами стандарту RFC 950 структуризацію IP-мережі здійснюють її розбиттям на окремі підмережі з

використанням масок як постійної, так і змінної довжини. При цьому під ідентифікатори підмереж виділяють старші біти адресного поля вузлів виділеної IP-адреси певного класу, а підмережі з'єднують між собою за допомогою маршрутизаторів або комутаторів 3-го рівня.

Захист даних від загрозливих дій з боку агресивного середовища виконують за допомогою апаратно-програмних ***міжмережевих екранів*** (мережевий екран, захисний екран, brandmauer – брандмауер, firewall – фаєрвол). Брандмауери можуть застосовувати різні алгоритми опрацювання мережевого трафіку, програмні засоби виявлення й знешкодження загроз, забезпечувати різні ступені захисту інформаційної системи на мережевому, сеансовому та прикладному рівнях моделі OSI.

Важливими різновидами міжмережевих екранів є ***проксі-сервери*** (proxy-server, сервер-посередник, уповноважений сервер) – програмні додатки, які виконують функції посередника між клієнтськими й серверними розподіленими додатками. Проксі-сервер може бути встановлений не тільки на платформі міжмережевого екрана, але й на будь-якому вузлі мережі. При цьому програмне забезпечення комп'ютерів користувачів повинно бути сконфігуроване так, щоб їх запити до ресурсного сервера не могли обійти проксі-сервера. Проксі-служби такого програмного сервера-посередника зорієнтовані на конкретні протоколи різних рівнів, що дає змогу аналізувати отримане повідомлення й приймати рішення про підозрілий характер поточного сеансу.

Сучасні ***маршрутизатори*** (роутер – router) мають багато додаткових функцій, які дають можливість будувати на їх базі доволі ефективні брандмауери для захисту інформаційної системи на канальному та мережевому рівнях.

Для забезпечення інформаційної безпеки мережі системні адміністратори найчастіше використовують такі підходи:

– виконання вимог стандарту IEEE 802.1x;

– застосування сервера автентифікації RADIUS (Remote Authentication Dial-In User Service – віддалена автентифікація користувацьких сервісів);

– використання технологій захищених каналів та віртуальних приватних мереж (Virtual Private Network).

Стандарт IEEE 802.1x описує організацію контролю доступу до LAN на рівні портів комунікаційного обладнання. Його використовують у мережах, побудованих на основі технологій Token Ring, FDDI, сімейства Ethernet та стека TCP / IP. При цьому комутатори, до портів яких підключають кінцеве обладнання користувачів з унікальними ***MAC-адресами*** (Media Access Control – управління доступом до середовища), конфігурують відповідно до поставлених вимог. Це забезпечує можливість ідентифікації користувача за його

MAC-адресою вже в точці доступу. Фільтрують кадри за MAC-адресою комутаторами мережі на основі записів, указаних адміністратором. Це запобігає несанкціонованому доступу до вузлів мережі та створює основу для детального обліку (білінг – billing) послуг, що надають користувачу в мережі.

Сервер автентифікації, авторизації та обліку різноманітних послуг на основі протоколу ***RADIUS*** застосовують для ідентифікації, обліку та контролю прав користувачів. Взаємодії між комутаторами мережі й сервером RADIUS досягають обміном керуючими повідомленнями із забезпеченням взаємної автентифікації та шифрування переданих даних. Для контролю за правами користувачів комутатор надсилає запит серверу RADIUS і на підставі наявної інформації про користувача дозволяє йому доступ до певних мережевих ресурсів або відмовляє в цьому. Рішення про віднесення обладнання з певною MAC-адресою до якої-небудь VLAN або переадресацію запитів приймає центральна база даних RADIUS. Невідоме обладнання може бути підключене до спеціальної VLAN із подальшим доступом до некритичних базових послуг.

Для діагностики й моніторингу комп'ютерних мереж, комп'ютерів та програмного забезпечення з метою виявлення можливих проблем у системі безпеки, оцінювання й усунення вразливості використовують спеціальні програмні або апаратні засоби – ***сканери вразливостей***. Сканери вразливостей забезпечують перевірку програмних додатків на наявність ознак, якими можуть скористатися зловмисники для порушення роботи мережі, погіршення якості надання послуг та несанкціонованого доступу до інформаційних ресурсів мережі. Сканери дають можливість виявляти активні програмні додатки, IP-адреси, відкриті порти та параметри мережевої операційної системи. Вони визначають рівень можливого втручання в операційну систему або програмні додатки та формують звіт про рівень безпеки мережі.

Системний адміністратор систематично виконує моніторинг мережі: аналізує її стан, мережевий трафік, контролює й керує параметрами апаратно-програмних компонентів. При інтегрованому моніторингу складної комп'ютерної мережі використовують здебільшого дві системи централізованого керування: ***систему керування мережею*** (Network Management System – NMS) та ***систему керування системою*** (System Management System – SMS). NMS зорієнтована на керування комунікаційним обладнанням та контроль трафіку мережі, а SMS збирає інформацію про встановлені в мережі комп'ютери та програмно-апаратні засоби, а також записує отримані параметри до спеціальної бази даних.

Слід зауважити, що надійна безпека інформаційних систем може бути забезпечена лише у випадку застосування методів комплексного захисту з використанням як апаратно-програмних засобів, так і організаційних та техно-

логічних заходів. Політика безпеки визначає необхідний і достатній рівень захисту інформаційної системи за критерієм доцільності, який ґрунтується на співвідношенні коштів, затрачених організацію захисту мережі, до втрат, які може понести підприємство чи організація у випадку проникнення зловмисника в мережу.

6.3.5. Підвищення рівня інформаційної безпеки за допомогою маршрутизаторів

Сучасні апаратно-програмні ***маршрутизатори*** (роутер – router) володіють багатьма додатковими функціями, які дають можливість на їх базі доволі ефективно захищати ресурси локальної мережі.

Так, функції ***контролю доступу*** (Access Control) та ***фільтрації портів*** TCP (Port filtering) дають змогу адміністратору локальної мережі обмежити доступ користувачів мережі до ресурсів чи послуг Інтернету. При цьому деяким користувачам забезпечують доступ тільки до електронної пошти, а іншим надають доступ до веб-сторінок, дають можливість користуватися широким спектром послуг. Маршрутизатори дають змогу створювати групи локальних користувачів, для яких можна встановлювати різний рівень доступу за часовим графіком. При блокуванні недозволеного трафіку маршрутизатори можуть не надсилати користувачу про це повідомлення, і в користувача складається враження, що послуги недоступні. При надсиланні користувачу повідомлення про блокування трафіку маршрутизатори можуть реєструвати спроби доступу в системному журналі.

Функція ***фільтрування MAC-адрес*** (MAC filtering) дає змогу обмежувати або надавати доступ до зовнішніх мереж лише для комп'ютерів локальної мережі із заданими MAC-адресами. Ця функція буває особливо ефективною при використанні служби динамічного присвоєння вузлам мережі IP-адрес.

Застосування в маршрутизаторах технології ***віртуальних приватних мереж*** (Virtual Private Networking – VPN) дає змогу використовувати загальнодоступні мережі для захищеного передавання даних, використовуючи для цього можливості шифрування й електронного цифрового підпису. При такому під'єднанні користувач може працювати з ресурсами віддаленої мережі так само, як і з ресурсами локальної мережі. Багато виробників маршрутизаторів стали випускати моделі з підтримкою VPN, починаючи від простого формування тунелів, до повноцінних вбудованих серверів VPN. Для створення VPN використовують такі протоколи, як ***IPSec*** (Internet Protocol Security – протокол безпеки Інтернет), ***PPTP*** (Point-to-Point Tunneling Protocol – протокол

тунелювання точка-точка), ***L2TP*** (Layer 2 Tunneling Protocol – протокол тунелювання 2 рівня), ***SSL*** (Secure Sockets Layer – шар захищених сокетів).

Підтримка ***VPN-тунелів*** (VPN Endpoint) дає змогу створення віртуального тунелю між маршрутизаторами мережі. При цьому переважно використовують протокол IPSec, що дає можливість шифрувати й дешифрувати передавані дані, а також перевіряти їхню незмінність і обмінюватися ключами. Саме таку функцію найактивніше використовують для об'єднання кількох віддалених одна від однієї мереж.

Функція ***пропускання VPN*** (VPN pass through) дає змогу зашифрованому трафіку проходити через маршрутизатор. При цьому VPN-клієнт дає можливість ініціювати з'єднання з VPN-сервером, а VPN-сервер дозволяє під'єднання ініційованим клієнтам. Цю функцію часто використовують у центральних офісах підприємств чи організацій для під'єднання філіалів і окремих комп'ютерів співробітників.

Функції ***перенаправлення портів*** (Port Forwarding) і ***створення віртуальних серверів*** (Virtual Server) дають змогу перенаправляти звернення до вказаних портів зовнішнього інтерфейсу маршрутизатора на пристрої, під'єднані до внутрішнього інтерфейсу. Необхідність перенаправлення може виникнути при розміщенні всередині мережі різних серверів (наприклад, веб-сервер, файловий сервер). Існує декілька способів перенаправлення портів:

– статичне перенаправлення окремих портів (Static), за якого задають відповідності між протоколами й портами зовнішнього інтерфейсу, протоколами й портами внутрішнього інтерфейсу, а також пристроями внутрішньої мережі. Використання такого перенаправлення дає змогу зробити сервер, розташований у внутрішній мережі, доступним із зовнішньої мережі;

– статичне перенаправлення груп портів, що відрізняється від статичного перенаправлення окремих портів лише тим, що для перенаправлення можна вказувати не окремі порти, а їхні групи. Це дає можливість забезпечити роботу таких додатків, як ігри, аудіо- / відеоконференції;

– динамічне перенаправлення портів (Dynamic, Triggered Mapping, Special Application), основною відмінністю якого від статичного перенаправлення портів є те, що один номер порту можна перенаправити на декілька внутрішніх IP-адрес. Використання динамічного перенаправлення актуальне для додатків, що використовують короткочасне передавання даних, за якого порт не займають надовго. Подія, що ініціює динамічне перенаправлення, повинна відбуватися у внутрішньому сегменті мережі, що накладає істотні обмеження на використання цього типу перенаправлення при хостингу служб.

Слід зауважити, що деякі маршрутизатори автоматично створюють відповідні перенаправлення портів за правилами міжмережевого екрана, проте

в більшості випадків задавати правила перенаправлення трафіку доводиться мережевому адміністратору.

Функція створення ***демілітаризованої зони*** (Demilitarized Zone – DMZ) дає змогу створити ще один рівень захисту комп'ютерної мережі за допомогою проміжної зони (мережі периметра) між внутрішньою й зовнішньою мережами. В DMZ можуть бути розміщені публічні сервери, доступні клієнтам із зовнішньої мережі.

Функція ***зовнішнього сервера*** (Exposed Server) дає змогу підключити комп'ютер внутрішньої мережі до зовнішньої мережі. У цьому випадку маршрутизатор здійснює перенаправлення пакетів, що адресовані всім TCP портам, на одну внутрішню IP-адресу. При цьому для DMZ можна використати окремий фізичний порт на маршрутизаторі або вказати IP-адресу комп'ютера, підключеного до одного зі звичайних портів.

Функція ***віддаленого керування*** (Remote Administration / Remote Management) дає змогу підключатись до інтерфейсу налаштувань маршрутизатора із зовнішнього сегмента мережі. Як правило, цю функцію використовують адміністратори мережі для налаштування територіально віддалених мережевих пристроїв. Проте, використовуючи функцію віддаленого керування, слід особливо уважно підійти до питання безпеки, оскільки цю функцію зловмисники можуть використати для отримання доступу до ресурсів мережі. Для захисту такого підключення використовують захищений протокол, а для здійснення функції віддаленого керування використовують комп'ютер із фіксованими IP-адресою та номером TCP-порта.

Функція ***ведення журналу подій*** (Logging) забезпечує ведення статистики звернень користувачів і реєстрацію всіх змін у конфігурації маршрутизатора.

6.4. Атаки на інформаційні та програмно-технічні ресурси комп'ютерної мережі

6.4.1. Види атак

Зловмисники загрожують інформаційним та програмно-технічним ресурсам мережі за допомогою цілеспрямованих дій, які називають ***атаками*** [9–26]. Атаки на ресурси комп'ютерної мережі із використанням спеціально створених шкідливих (злоякісних) програм зловмисники можуть використовувати з метою:

– передавання в мережу та поширення вірусів та інших шкідливих програм;

– перехоплення та відтворення конфіденційної інформації з метою особистого збагачення або нанесення шкоди приватним чи державним установам;

– збирання адрес електронної пошти з метою розсилання спаму;

– створення перешкод у роботі антивірусних програм, систем моніторингу подій і мережевих екранів;

– стирання або переписування даних із дисків та пошкодження файлів;

– виведення з ладу комп'ютерного та комунікаційного обладнання мережі.

До найпоширеніших належать такі види атак на ресурси комп'ютерної мережі.

Відмова в обслуговуванні (*Denial of Service – DoS*) – атака з поодинокого джерела з метою зробити ресурси комп'ютерної мережі недоступними для авторизованих користувачів внаслідок перевищення допустимого рівня функціонування операційної системи, додатків або технічних засобів мережі. У переважній більшості DoS-атаки спрямовують на інформаційні сервери: вебсервери, поштові й файлові сервери, DNS-сервери відображення імен тощо. До найпоширеніших DoS-атак належать: Flood, ICMP flood, TCP SYN flood, Identification flood, UDP flood, TCP flood і Ping of Death.

Flood (затоплення) та ***ICMP flood*** (***flood ping*** – потік коротких запитів, що вимагають відповіді) – це атаки, під час яких система отримує велику кількість ICMP- або UDP-пакетів, які не несуть корисної інформації, але перевантажують мережу непродуктивними діями щодо їх аналізу.

TCP SYN flood – атака, що полягає в надсиланні великої кількості запитів на ініціалізацію TCP-з'єднань без їх закриття, що переповнює інформаційний канал SYN-пакетами, блокує роботу сервера й не дає йому змоги відповідати на запити інших клієнтів.

Identification flood (запит ідентифікації системи) – атака, під час якої мережа постійно отримує запити на ідентифікацію системи. Це знижує продуктивність системи, оскільки аналіз цих запитів і генерування на них відповідей перевантажують процесор.

UDP flood, ***TCP flood*** і ***Ping of Death*** – атаки посиланням до мережі великої кількості, відповідно, пакетів UDP, TCP та фрагментованого ICMP-пакета великого обсягу. Це призводить до зв'язування мережевих ресурсів і зависання операційної системи комп'ютера чи сервера (останнім часом використовують рідко).

Загрозу DoS атак можна послабити конфігуруванням на маршрутизаторах і міжмережевих екранах функцій ***антиспуфінгу*** (Spoof – використання фальшивої адреси) – фільтрування пакетів за підозрілими IP-адресами. Ефективним

є обмеження обсягу ***некритичного трафіку*** (non-critical traffic), який проходить мережею (наприклад, обмеження обсягів пакетів ICMP).

Розподілена DDoS-атака (Distributed Denial of Service) – скоординована DoS-атака відразу з багатьох комп'ютерів. Для її організації комп'ютери, що беруть у ній участь, попередньо заражають спеціальними вірусами-хробаками. Заражені комп'ютери за командою зловмисника починають одночасно надсилати запити до атакованого сервера. У результаті сервер не справляється з навантаженням, і доступ до атакованого ресурсу ускладнюється або взагалі стає неможливим. Фахівці вважають DDoS-атаки одними з найскладніших видів мережевих загроз, що можуть паралізувати роботу комп'ютерної мережі.

До найпоширеніших DDoS-атак належать TCP SYN flood (SYN flooding), TCP flood, UDP flood, ICMP flood та Smurf Attack.

Атаку ***Smurf Attack*** здійснюють передаванням до мережі з великою кількістю комп'ютерів широкомовних ICMP-повідомлень, підмінивши в них адресу відправника адресою сервера-жертви. Множина комп'ютерів, які прийняли такі запити, надсилають серверу-жертві пакети у відповідь. У результаті сервер не справляється з навантаженням, тому доступ до атакованого ресурсу стає неможливим.

Найнебезпечнішими є злоякісні програми, що використовують одночасно кілька видів атак. Так, програма ***Stacheldracht*** дає можливість генерувати лавини ***широкомовних коротких запитів*** (пінг-запитів) і здійснювати атаки на програмно-апаратні засоби мережі. Змінюючи значення параметрів ***бази керуючої інформації*** (Management Information Base) на керованих адміністратором вузлах мережі, злоякісна програма дестабілізує їхню роботу.

Протидія DDoS атакам передбачає встановлення на міжмережевих екранах спеціалізованих ***antiflood-фільтрів*** (antiflood-filters), що здатні в реальному часі блокувати за IP-адресами доступ до веб-серверів тих клієнтів, які генерують інтенсивний потік запитів, сформованих за протоколом ***HTTP*** HyperText Transfer Protocol – протокол передавання гіпертексту). Використовують також резервування розподілених систем, які не припиняють обслуговувати користувачів навіть тоді, коли деякі їхні ресурси стають недоступними.

Сніфер пакетів (Sniffer – перехоплювач) – програма, яка використовує карту мережевого інтерфейсу, що працює в нерозбірливому режимі (promiscuous mode) приймання й дає змогу перехоплювати всі пакети мережі. Зібрані так дані передають для опрацювання на мережевий монітор, призначений для аналізу трафіку мережі. Зловмисник при цьому може видати себе за санкціонованого користувача, перебуваючи в самій організації або за її межами. Застосування програми дає можливість перехопити будь-який

незашифрований, а іноді й зашифрований трафік із метою отримання корисної для зловмисника інформації (паролі, IP-адреси електронної пошти, інформація про структуру мережі тощо).

IP-спуфінг (spoof – обман, підміна) – атака, що передбачає використання зловмисником, який видає себе за санкціонованого користувача, чужої IP-адреси. Її часто застосовують як складову комплексної атаки (наприклад, DDoS-атаки, для здійснення якої зловмисник розміщує відповідну програму за чужою IP-адресою).

Знизити загрозу IP-спуфінгу можна, жорстко контролюючи права доступу користувачів із забороною будь-якого трафіку, вихідна адреса якого не є однією з IP-адрес внутрішньої мережі, запроваджуючи додаткові заходи автентифікації та системи криптографічного захисту.

Вірус (Virus) – шкідлива програма, здатна до впровадження в інші файли комп'ютера, зокрема у файли системних і прикладних програм. В інші комп'ютери вірус може потрапити тільки в тілі переданого користувачем файла. Шкода, яку може нанести системі вірус, залежить від мети, яку поставив зловмисник, розробляючи вірус: від періодичної появи на екрані різноманітних зображень, що заважають користувачу працювати, аж до порушення інформаційної безпеки, втрати даних, руйнування системних програм і втрати працездатності мережі.

Антивірусний захист передбачає виявлення й діагностування вірусного зараження за допомогою програм антивірусного захисту з подальшим відновленням працездатності інформаційної системи. Для цього аналізують поведінку як системних, так і прикладних програм, а також перевіряють вміст підозрілих файлів.

Логічні бомби (Logic bombs) – спеціально сконструйований програмний код, який спричиняє деструктивну роботу програми, що виконують на комп'ютері, зокрема, її повне припинення.

Керовані атаки (War driving) – отримання несанкціонованого доступу до серверів комп'ютерних мереж, що використовують безпроводові технології. Для проникнення в мережу застосовують антени та безпроводові мережеві адаптери.

Спам (Spam) – це вид атаки за допомогою електронної пошти. Велика кількість листів, яку надсилає зловмисник на певні адреси електронної пошти, унеможливлює роботу поштових скриньок і поштових серверів. Разом із непотрібною інформацією зловмисники засобами електронної пошти можуть надсилати шкідливі програми й організовувати різні види атак. При цьому зловмисники можуть використовувати бот-мережі (будуть описані нижче), генерувати адресу відправника й теми його листа з використанням випадкових

чисел. Із цим видом атаки складно боротися, тому що інтернет-провайдер може обмежити кількість листів тільки від одного відправника.

Ін'єкції (Exploit tools) – вид атаки впровадженням шкідливих команд або даних у працюючу систему з метою впливу на її роботу так, щоб отримати доступ до комп'ютера-жертви та даних або дестабілізувати роботу всієї системи. Найчастіше цей вид атаки зловмисники використовують для формування шпіонських запитів до баз даних, зламу веб-сайтів та внесення неправдивої (фейкової) інформації на веб-сторінки, які користуються популярністю певного кола користувачів. Прикладом такого виду атак може бути SQL-ін'єкція (Structured query language), яку використовують зловмисники для зміни параметрів запитів до бази даних.

Програми-шпигуни (програми мережевої розвідки) – це вид шкідливих програм, які зловмисник таємно встановлює на комп'ютери мережі з метою збирання секретної інформації про будову та принципи функціонування мережі, виявлення проксі-серверів, аналізу їх роботи, перехоплення й аналізу пакетів та інших даних. Зібрану шпигунськими програмами інформацію використовують, здебільшого, для організації наступних атак. Розрізняють такі види шпигунських програм:

– сканери портів, які записують інформацію, що передають через порти комп'ютера, під'єднані до мережевого адаптера, модема, принтера тощо (програма ***NeoTrace***);

– клавіатурні та екранні шпигуни, які копіюють інформацію, що вводять у комп'ютер із клавіатури (програма ***Hook Dump***), або виводять на екран комп'ютера (програма ***Ghost spy***);

– модемні та мережеві кіберрозвідники, які автоматично записують телефонні розмови в режимі диктофона, копіюють записи через телефонну лінію або звукову карту (програми ***Modem spy***, ***Flexispy***, ***Mobile Spy*** й ***Mobistealth***), здійснюють моніторинг мережевого трафіку, електронної пошти, веб-сайтів тощо;

– програми для перехоплення й перескерування трафіку з метою направити трафік атакованого комп'ютера за фальшивою адресою. У переважній більшості цього досягають, формуючи фальшиву ARP- або DNS-відповідь при надсиланні атакованим комп'ютером широкомовних запитів на відображення імен. Перескерувати трафік зловмисник може також, сформувавши від імені маршрутизатора фальшиве ICMP-повідомлення про зміну маршруту.

Бекдор (back door – чорний хід, люк) – злоякісна програма, яка використовує вразливість програмного забезпечення системи й забезпечує зловмиснику доступ до конфіденційної інформації, розміщеної на віддаленому комп'ютері, а також дає змогу виконувати на ньому несанкціоновані дії.

Завдяки бекдору зловмисник може витягнути з інфікованої мережі назву та реквізити підприємства, пароль проксі-сервера, ім'я поштового сервера, паролі, адреси електронної пошти клієнтів тощо.

Зловмисники доволі часто для організації атак на ресурси комп'ютерної мережі використовують ***бот-мережі*** (botnet, зомбі-мережі), до складу яких може входити велика кількість інфікованих комп'ютерів зі встановленими зловмисником шкідливими програмами. До отримання команди від зловмисника бот-мережа знаходиться в сплячому режимі стані. Отримавши команду, вона починає виконувати закладені в неї функції, наносячи шкоду ресурсам мережі (розкриття IP-адрес та пароля доступу до інформаційних ресурсів, розповсюдження спаму, здійснення атак на сервери з метою спровокувати відмову в обслуговуванні користувачів тощо). Керувати ботом зловмисник може як надсилаючи відповідний код на адресу одного з комп'ютерів, так і через веб-сайти з використанням наперед сформованої ***URL-адреси*** (Uniform Resource Locator – інфікований вказівник на ресурс) та з використанням ***p2p-***мереж (Peer-to-peer network – мережа рівноправних вузлів). При застосуванні технології одноранговогo розподілення ресурсів p2p забезпечують безпосередній обмін повідомленнями між кінцевими системами, якими можуть бути персональні комп'ютери, під'єднані до публічної мережі. При цьому для пошуку необхідних ресурсів p2p-технологія передбачає використання як децентралізованого, так і централізованого каталогів, або системи запитів на потрібний ресурс. Найчастіше бот-мережі використовують для організації DDoS-атак.

Для підготовлення атак на інформаційні та програмно-апаратні ресурси мережі зловмисники часто використовують методи й засоби моніторингу відкритих та відносно відкритих джерел і соціальної інженерії, які розглянуто вище.

6.4.2. Виявлення атак на ресурси комп'ютерної мережі

Обов'язки з виявлення мережевих атак, спроб несанкціонованого доступу та використання ресурсів мережі покладено на адміністратора безпеки, який відповідає за безпеку всієї мережі. Ці функції може також виконувати системний адміністратор. Одним із найпоширеніших адміністративних методів боротьби з атакуючими програмами є заборона тих програм, які не мають дозволу на виконання в конкретній інформаційній системі. Список дозволених програм формує адміністратор мережі.

Мережеві монітори (sniffers), типовими представниками яких є ***Wireshark*** і ***Microsoft Network Monitor***, надають адміністратору засоби для

слідкування за мережевим трафіком, відфільтровування пакетів та перегляду їх змісту, збирання статистичної інформації про роботу пристроїв мережі. Отримані дані заносять до журналу реєстрації. Висновки про можливі загрози мережевий адміністратор робить після аналізу характеристик мережевого трафіку та записів у журналах реєстрації. Це тривалий і трудомісткий процес, який є ефективним для невеликих локальних мереж, але не завжди приносить очікуваний ефект у разі моніторингу мережі великого підприємства.

Для централізованого моніторингу подій у режимі реального часу в комп'ютерних мережах використовують системи ***управління інформацією та повідомленнями безпеки*** (Security Information and Event Management – SIEM). SIEM-системи забезпечують уповноважений персонал та користувачів мережі інформацією про її стан та дають змогу, відповідно до заданих правил та налаштувань, оперативно реагувати на виникнення загрозливих ситуацій.

Серед сучасних програмно-апаратних засобів моніторингу атак на інформаційну систему широкого застосування знайшли ***системи виявлення атак*** (Intrusion detection system – IDS) та ***системи запобігання втручанням*** (Intrusion prevention system – IPS). Основним призначенням цих систем є виявлення фактів несанкціонованого доступу до корпоративної мережі та прийняття відповідних заходів протидії: інформування адміністратора з мережевої безпеки про факт вторгнення, припинення з'єднання та переустановлення міжмережевого екрана для блокування подальших дій зловмисника. Ці системи містять спеціальні програми та процедури, призначені для аналізу в режимі реального часу великого обсягу мережевого трафіку з метою виявлення атакуючих дій зловмисників та запобігання ним. Їх використання дає змогу вирішити низку завдань, які забезпечують інформаційну безпеку мережі.

Існуючі системи виявлення атак відрізняються між собою і за типом подій, які вони здатні виявляти, і за методами виявлення атак. До основних функцій цих систем належать:

– аналіз системної конфігурації та виявлення вразливостей мережі;

– моніторинг користувацької, мережевої та системної активності в мережі;

– контроль цілісності файлів та інших ресурсів інформаційної системи;

– розпізнавання ознак атакуючих дій, що можуть загрожувати програмно-апаратним ресурсам мережі;

– контроль за роботою та станом компонентів інформаційної системи й реєстрація інформації про порушення безпеки;

– аналіз характерних ознак, що можуть свідчити про проведення атакуючих дій, а також їх статистичне опрацювання;

– надання послуг адміністратору мережі в питаннях інформаційної безпеки.

Крім функцій моніторингу та аналізу атакуючих дій та виявлення інцидентів, IDS виконують додаткові функції, зокрема:

– передавання інформації про виявлені атаки в систему централізованого моніторингу подій;

– повідомлення адміністратору даних про виявлені інциденти інформаційної безпеки декількома каналами: електронною поштою, SNMP-повідомленнями, системним журналом, консоллю управління системою IDS;

– генерування звітів за вказаними подіями.

Системи IPS доповнюють системи IDS такими важливими функціями:

– блокування атаки (розірвання з'єднання з користувачем, який порушує політику безпеки, блокування доступу до ресурсів, вузлів (хостів), додатків);

– змінювання конфігурації мережевих пристроїв для запобігання подальшим атакам;

– видалення інфікованих файлів і відправлення отримувачу лише очищених файлів.

Виникнення загроз для програмно-апаратних компонентів мережі може бути визначене системою виявлення атак двома методами:

– виявлення загроз (misuse detection);

– виявленням аномалій (anomaly detection).

Метод виявлення загроз передбачає наявність формалізованого опису характерних ознак різних видів атак та їх модифікацій і ґрунтується на порівнянні сигнатур, другий ґрунтується на знанні опису очікуваної поведінки контрольованих об'єктів системи. ***Сигнатура*** – це послідовність байтів, яка є характерною для певного виду загрози. Якщо програма виявлення атак при скануванні файлів інформаційної системи виявляє сигнатуру відомого виду загрози, це кваліфікується як її наявність у мережі. Після виявлення загрози система захисту може запропонувати видалити пошкоджений файл або спробувати його відновити видаленням загрозливої послідовності байтів. Отже, метод порівняння сигнатур дає можливість ефективно виявляти шкідливі програми з відомими ознаками, але не дає змоги виявити нових, ще не описаних програм.

Метод аномалій ґрунтується на припущенні, що відхилення поведінки контрольованого об'єкта від описаної норми свідчить про наявність ворожих дій. Прикладами аномалій у роботі мережі є різке збільшення інтенсивності мережевого трафіку, перевищення кількістю з'єднань між вузлами мережі заданої норми, високе завантаження центрального процесора та серверів, використання пристроїв, які зазвичай не використовують. Метод аномалій дає можливість виявляти нові атаки, але потребує детального багатопараметричного опису поведінки об'єктів інформаційної системи. Більшість сучасних

систем виявлення атак поєднує обидва методи й дає змогу боротися з певним спектром шкідливих програм: вірусами, мережевими хробаками, троянами тощо.

Слід зазначити, що системи IDS/IPS не роблять інформаційну систему цілком безпечною, проте вони є корисним доповненням до міжмережевого екрана, який, згідно з політикою мережевої безпеки, створює бар'єр для забороненого типу трафіку. Вони можуть виявити атаки, які обійшли брандмауер, і запобігти нанесенню шкоди інформаційній системі. Вибір системи виявлення атак залежить від особливостей побудови та функціонування інформаційної системи й вимог політики мережевої безпеки.

Важливим є використання методів запобігання можливим атакуючим діям зловмисників на ресурси комп'ютерних мереж. Методи соціальної інженерії для боротьби з інцидентами передбачають використання механізму оповіщення користувачів мережі Інтернет про злоякісні веб-сайти, які спеціально створюють зловмисники для збирання конфіденційних даних із метою підготовки атак на інформаційні ресурси мережі. Так, антифішинг здійснюють введенням в популярні браузери (***Microsoft Internet Explorer***, ***Firefox*** тощо) спеціальних програмних додатків – ***плагінів*** (Plug-in – під'єднувати), що попереджають користувачів про їх звертання до підроблених або підозрілих веб-сайтів.

6.5. Захист приватної мережі від зовнішнього втручання

6.5.1. Забезпечення доступу користувачів приватної мережі до ресурсів мережі Інтернет

Локальна мережа підприємства чи організації (приватна мережа, корпоративна мережа), яка використовує стек комунікаційних протоколів TCP / IP, але не має виходу в мережу Інтернет, має назву інтранет-мережі (intranet) [9–26]. Для користувачів інтранет-мережі, яка працює за стандартами стеку протоколів TCP / IP, передбачено в класах A, B і C окремі групи IP-адрес, що не обробляються маршрутизаторами й не потребують реєстрації в інтернет-провайдера. Міжнародна організація ***IANA*** (Internet Assigned Numbers Authority), відповідальна за використання числових адрес у мережі Інтернет, установила для використання в інтранет-мережах три діапазони IP-адрес:

– клас A: 10.0.0.1 – 10.255.255.254;
– клас B: 172.16.0.1 – 172.31.255.254;
– клас C: 192.168.0.1 – 192.168.255.254.

Для виходу в мережу Інтернет корпоративні мережі використовують міжмережевий екран із програмним проксі-сервером, установленим на виділений комп'ютер або маршрутизатор. Проксі-сервер між закритою локальною мережею й зовнішніми мережами здійснює від свого імені за допомогою протоколу ***трансляції мережевих адрес*** (Network Address Translation – NAT) згенеровані запити користувачів корпоративної мережі адресатам у мережі Інтернет. Тобто проксі-сервер контролює та обмежує вихід у зовнішню мережу внутрішніх користувачів і подає їхні запити назовні з мережевими адресами із зареєстрованого в інтернет-провайдера ***пулу адрес*** (adress pool, неперервна множина). Пул виділених інтернет-провайдером адрес, містить, як правило, обмежену кількість адрес, а може обмежитися й однією зареєстрованою адресою.

Сучасні маршрутизатори (наприклад, маршрутизатори Cisco ASA) під час виконання функцій проксі-сервера використовують три види трансляції мережевих адрес протоколом NAT:

а) ***статична трансляція адрес*** (Static NAT);

б) ***динамічна трансляція адрес*** (Dynamic NAT);

в) ***маскарадна трансляція адрес*** (NAT with overload або Port Address Translation – PAT).

При конфігуруванні (налаштуванні) маршрутизатора системний адміністратор зазначає параметри трансляції адрес протоколом NAT, межі пула адрес мережі Інтернет та пула виділених інтернет-провайдером зовнішніх IP-адрес, а також виконує необхідні записи в таблиці маршрутизації.

Static NAT задає однозначну відповідність внутрішньої адреси зовнішній із виділеного інтернет-провайдером пула адрес. Запис про таку трансляцію зберігають у таблиці налаштувань маршрутизатора до її зміни адміністратором мережі. Її застосовують у тому випадку, коли локальний вузол і зовнішній вузол повинні підтримувати зв'язок лише один з одним або для забезпечення доступу зовнішніх користувачів до сервера приватної мережі.

Dynamic NAT під час проходження пакета із внутрішньої мережі в зовнішню вибирає для нього вільну адресу із зазначеного пула зареєстрованих адрес. Запис про створену трансляцію зберігають протягом невеликого проміжку часу, щоб пакети у відповідь могли потрапити адресату. Якщо протягом цього часу трафік за створеною трансляцією відсутній, то запис видаляють, а адресу повертають у пул. Якщо потрібно створити нову трансляцію, а вільних адрес в пулі немає, то пакет відкидають.

NAT with overload відображає множину приватних адрес в одну виділену інтернет-провайдером публічну IP-адресу, використовуючи різні

порти протоколів транспортного рівня. Цей протокол дає змогу замаскувати топологію внутрішньої мережі, його використовують частіше ніж два інші.

На рис. 6.7 показано схему підключення комп'ютерів користувачів приватної мережі класу С 192.168.1.0/24 до зовнішньої мережі за допомогою міжмережевого екрана на базі маршрутизатора 1. Першому порту (шлюзу за замовчуванням) маршрутизатора 1 приватної мережі присвоєно адресу 192.168.1.1, а другому – виділено інтернет-провайдером адресу 202.30.1.2 виродженої мережі 202.30.1.0/30, яка забезпечує доступ до зовнішніх мереж. Користувачам внутрішньої мережі, під'єднаним до комутатора, виділено пул адрес 192.168.2 – 192.168.1.254. Прямий і зворотний переходи пакетів із приватної мережі в Інтернет забезпечує протокол NAT with overload проксі-сервера, який працює на мережевому рівні маршрутизатора R1.

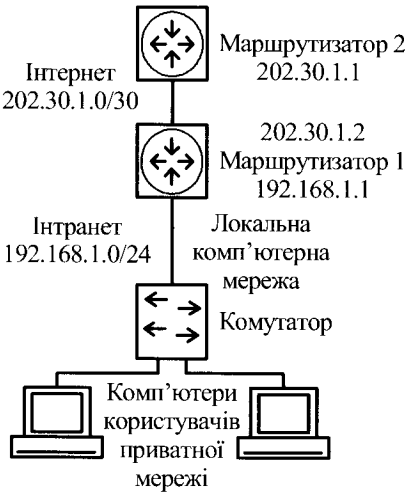


Рис. 6.7. Підключення комп'ютерів користувачів приватної мережі до Інтернету

Вузол внутрішньої мережі 192.168.1.0/24 формує запит адресату зовнішньої мережі із зазначенням в ньому своєї адреси з номером порту, IP-адреси зовнішнього адресата і надсилає його NAT-маршрутизатору за адресою 192.168.1.1. Останній формує обліковий номер відправника, замінює його зареєстровану IP-адресу 202.30.1.2, робить відповідні записи в таблиці відстеження з'єднань (NAT-таблиці) та переправляє запит на порт маршрутизатора 2 Інтернет-провайдера за адресою 202.30.1.1. Отримавши відповідь від зовнішнього адресата, маршрутизатор на основі аналізу NAT-таблиці переправляє його за призначенням. Поле номера вузла в NAT-таблиці проксі-сервера містить 16 двійкових розрядів, що дає змогу протоколу обслуговувати понад 60000 користувачів.

Використання проксі-сервера із протоколом PAT на маршрутизаторі 1 дає змогу підприємству не лише зекономити кошти на реєстрації в інтернет-провайдера обмеженої кількості IP-адрес, але й частково захистити внутрішню мережу від загроз, що можуть надійти з боку зовнішнього середовища. Проксі-сервер дає змогу запобігти доступу із зовнішньої мережі до внутрішніх вузлів, залишаючи можливість звернення із внутрішньої мережі до зовнішньої. Пакети із зовнішньої мережі у внутрішню можуть надійти тільки за наявності відповідного запису в таблиці відстеження з'єднань (NAT-таблиці).

Застосування протоколу трансляції адрес для забезпечення інформаційного обміну між вузлами приватної мережі й вузлами зовнішньої мережі, при якому з'єднання ініціює вузол зовнішньої мережі (наприклад, обмін повідомленнями електронної пошти), розглянемо на прикладі об'єднаної мережі. Структуру об'єднаної мережі, до складу якої входять три мережі, об'єднані маршрутизатором, на якому функціонує протокол Static NAT, зображено на рис. 6.8.

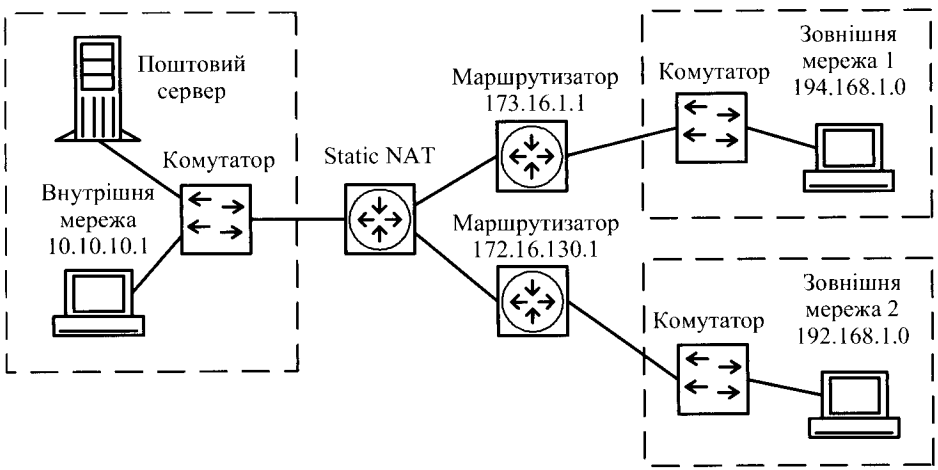


Рис. 6.8. Об'єднана мережа зі статичною трансляцією адрес

Вигляд NAT-таблиці наведено на рис. 6.9.

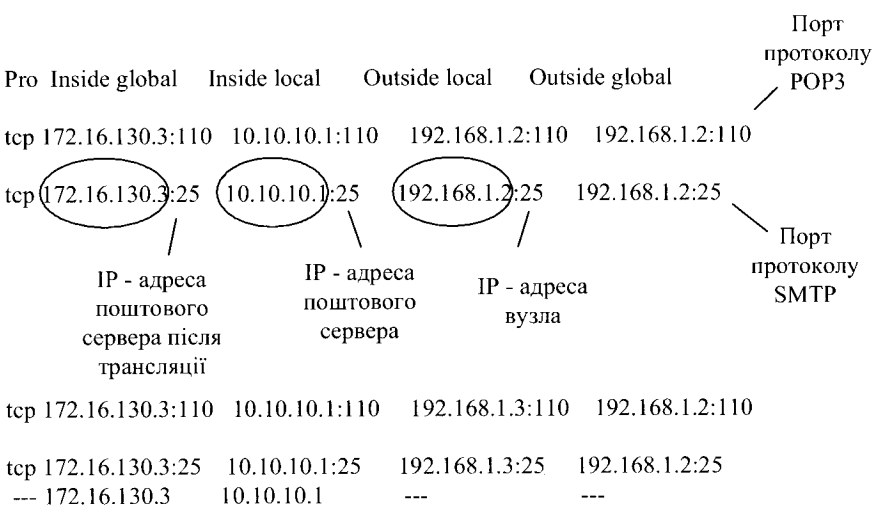


Рис. 6.9. Таблиця трансляції адрес

Внутрішня мережа – 10.10.10.0/24 <http://10.10.10.0/24; перший порт маршрутизатора R1 (шлюз за замовчуванням з внутрішньою мережею) – 10.10.10.1; поштовий сервер – 10.10.10.2; зовнішня мережа 1 – 194.168.1.0/24 <http://194.168.1.0/24>; перший порт маршрутизатора R2 (шлюз за замовчуванням з зовнішньою мережею 1) – 194.168.1.1; зовнішня мережа 2 – 192.168.1.0/24 <http://192.168.1.0/24>; перший порт маршрутизатора R3 (шлюз за замовчуванням з зовнішньою мережею 2) – 192.168.1.1; вироджена мережа 1 173.16.1.0/24 <http://173.16.1.0/24>; другий порт маршрутизатора R1 – 173.16.1.2; другий порт маршрутизатора R2 – 173.16.1.1; вироджена мережа 2 – 172.16.130.0/24 <http://172.16.130.0/24>; третій порт маршрутизатора R1 – 173.16.130.2; другий порт маршрутизатора R3 – 173.16.130.1.

Трансляцію мережевих адрес налаштовано тільки для мережі 192.168.1.0/24.

За спроби доступу до поштового сервера з іншої мережі трансляція адрес не відбувається.

6.5.2. Захист інформаційних ресурсів за допомогою міжмережевих екранів та створення DMZ

Для захисту даних від загрозливих дій із боку зловмисників використовують апаратно-програмні ***міжмережеві екрани*** (захисний екран, мережевий екран, brandmauer – брандмауер, firewall – фаєрвол), які встановлюють між інформаційною системою та ймовірним агресивним середовищем, з боку якого можуть надійти загрози. Міжмережеві екрани можуть застосовувати різні алгоритми опрацювання мережевого трафіку, програмні засоби виявлення й знешкодження загроз, забезпечувати різні ступені захисту інформаційної системи. Функції міжмережевого екрана залежать від його типу, моделі та конкретної конфігурації. Розрізняють такі типи міжмережевих екранів:

– лінійні шлюзи;
– фільтри пакетів;
– шлюзи додатків;
– комбіновані міжмережеві екрани.

Найпоширенішим прикладом здійснення ***лінійного шлюзу*** є його побудова на базі маршрутизатора з використанням протоколу NAT, який перехоплює згенеровані протоколами ***TCP*** (Transmission Control Protocol) або ***UDP*** (User Datagram Protocol) запити від вузлів приватної мережі й надсилає їх у зовнішню мережу Інтернет від свого імені. При цьому в зовнішнє середовище не потрапляють адреси вузлів захищеної мережі. ***Фільтрацію вхідних і***

вихідних пакетів на основі аналізу заголовка IP-пакета здійснюють, здебільшого, маршрутизатори з використанням протоколів мережевого рівня. ***Шлюзи додатків*** (персональні міжмережеві екрани) призначені для захисту від несанкціонованого доступу окремих комп'ютерів. Це спеціальні програми, які реєструють усю вхідну та вихідну інформацію комп'ютера та аналізують її за певними критеріями (ознаками). Найвідомішими шлюзами додатків є антивірусні програми.

Комбіновані міжмережеві екрани поєднують різні функції захисту, наприклад, фільтри пакетів та лінійні шлюзи або лінійні шлюзи та шлюзи додатків. Вони забезпечують вищий рівень захисту IP-мереж порівняно з міжмережевими екранами одного типу.

Класифікують міжмережеві екрани за рівнями моделі OSI, протоколи яких вони використовують для захисту внутрішньої мережі:

– міжмережеві екрани мережевого рівня (лінійні фільтри, packet filtering firewall);

– міжмережеві екрани сеансового рівня (circuit level gateway);

– міжмережеві екрани прикладного рівня (application level gateway).

Міжмережеві екрани мережевого рівня, які ще називають мережевими екранами з фільтрацією пакетів, будують, переважно, на базі апаратного або програмного маршрутизатора. Вони аналізують зміст IP-заголовків пакетів, порівнюють отриману інформацію із записаними у своїй таблиці правилами, ухвалюють рішення про проходження пакета чи його відкидання. Найчастіше інформацією, на підставі якої ухвалюють рішення про проходження пакета, є його адресна інформація, інформація про протокол, який сформував запит, номери портів отримувача та відправника. Конкретна конфігурація правил залежить від політики безпеки підприємства чи організації. Міжмережевий екран мережевого рівня також може виконувати трансляцію мережевих адрес із використанням NAT-протоколу.

Міжмережеві екрани сеансового рівня виконують розпізнавання учасників сеансу. Після підтвердження автентичності користувача та сервера такі екрани перевіряють послідовність обміну між ними повідомленнями на відповідність установленим процедурам роботи відповідних протоколів. Пакети, послідовність обміну якими не відповідає логіці роботи відповідних протоколів, підлягають ***фільтруванню з урахуванням контексту*** (stateful packet inspection). Така перевірка дає змогу захистити мережу від різних видів атак, зокрема DoS-атак.

Міжмережеві екрани прикладного рівня можуть аналізувати й контролювати зміст повідомлень, якими обмінюються відповідні додатки, викорис-

товуючи для цього різні алгоритми опрацювання вхідних та вихідних даних. Для фільтрування потоку даних вони використовують спеціальні шаблони, порівняння за зразками, евристичні правила та інші методи з арсеналу експертних систем. Міжмережеві екрани прикладного рівня можуть містити програмний проксі-сервер, що виконує функції посередника між клієнтськими й серверними розподіленими додатками. У цьому випадку програмне забезпечення комп'ютерів користувачів (клієнтів) мережі необхідно сконфігурувати так, щоб їх запити до ресурсного сервера не могли обійти проксі-сервер. Такі екрани забезпечують найвищий рівень захисту мережі підприємства чи організації.

Переважно що вищий рівень роботи міжмережевого екрана, то вищий рівень захисту він забезпечує. На практиці використовують екрани, у яких служби захисту організовано на декількох рівнях.

У комп'ютерних мережах, робота яких ґрунтується на використанні стеку комунікаційних протоколів TCP / IP, найпоширеніші два типи міжмережевих екранів:

– міжмережеві екрани прикладного рівня, які містять модулі доступу до портів протоколів прикладного рівня (наприклад, Hypertext Transfer Protocol – HTTP, протокол пересилання гіпертексту; Simple Mail Transfer Protocol – SMTP, простий протокол пересилання електронної пошти; File Transfer Protocol – FTP, протокол пересилання файлів; Telnet – протокол емуляції терміналу тощо);

– міжмережеві екрани рівня з'єднань, які контролюють, переважно, TCP-з'єднання й виконують фільтрацію тих видів пакетів, які не передбачено або заборонено правилами.

Міжмережевий екран, установлений між внутрішньою й зовнішньою мережами, можна влаштувати на базі одного або декількох комп'ютерів чи маршрутизаторів.

На рис. 6.10 наведено структурну схему мережі підприємства, до складу якої входить дві мережі:

а) внутрішня мережа (інтранет), яка містить конфіденційну (закриту) інформацію;

б) мережа периметра (інтернет), в якій розміщено публічні сервери з відкритою інформацією, доступ до якої дозволений користувачам зовнішніх (публічних) мереж.

Мережа периметра (Demilitarized Zone – DMZ, демілітаризована зона) розміщена між міжмережевим екраном на базі виділеного комп'ютера і маршрутизатором провайдера. Підприємство в цьому випадку має орендувати в інтернет-провайдера зовнішні IP-адреси як для внутрішньої мережі, так і для вузлів демілітаризованої зони. Кількість орендованих зовнішніх IP-адрес

залежить і від специфіки роботи підприємства, і від типу протоколу NAT, який функціонує на виділеному комп'ютері. На наведеній схемі провайдер за заявкою підприємства виділив для його мережі пул адрес 202.30.1.1 – 202.30.1.30 з маскою 255.255.255.224. Чотири адреси 202.30.1.1 – 202.30.1.4 цього пула системний адміністратор присвоїв вузлам мережі DMZ. Залишок адрес протокол Dynamic NAT може використовувати для трансляції адрес користувачів приватної мережі.

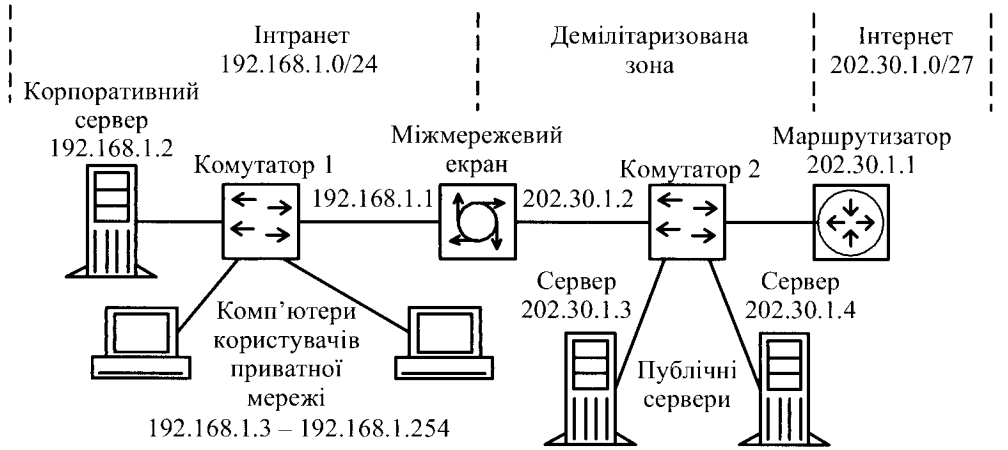


Рис. 6.10. Мережа підприємства, до складу якої входить внутрішня мережа та мережа периметра

Зовнішній маршрутизатор захищає внутрішню мережу й публічні сервери, розміщені в зоні DMZ, від загроз із боку зовнішньої мережі фільтрацією вхідного трафіку. Він, як правило, належить інтернет-провайдеру, який виконує його налаштовування. Використання додаткових функцій маршрутизатора дає змогу організувати захист демілітаризованої зони й внутрішньої мережі від зовнішніх загроз. Для забезпечення доступності й цілісності розміщеної на публічних серверах інформації на них установлюють персональні міжмережеві екрани, такі, наприклад, як антивірусні програми й фільтри спаму.

Міжмережевий екран на базі ***шлюзу безпеки*** (Security Geteway – SG) захищає внутрішню мережу від загроз із боку зовнішньої мережі й демілітаризованої зони, виконує трансляцію мережевих адрес та шифрування даних. Спеціальні програми реєструють усю вхідну й вихідну інформацію та аналізують її за певними критеріями (ознаками). Вони фіксують підозрілі запити, направлені на сканування портів та збирання інформації про мережу, відстежують стан TCP-з'єднань, а також виконують фільтрування пакетів.

Програмні додатки проксі-сервера, що виконує функції посередника між клієнтськими й серверними розподіленими додатками мережі, можуть бути

розміщеними як на міжмережевому екрані, так і на корпоративному сервері внутрішньої мережі. У такому випадку ***проксі-служби*** (proxy service, програмні додатки проксі-сервера) зможуть обслуговувати основні служби вищих рівнів стеку комунікаційних протоколів. Для цього системний адміністратор повинен сконфігурувати програмне забезпечення так, щоб їх запити до ресурсного сервера не могли обійти проксі-сервер.

Розміщення проксі-сервера у внутрішній мережі наведено на рис. 6.11. За такої схеми всі запити користувачів і внутрішньої, і зовнішньої мережі надходять на проксі-сервер за адресою 192.168.1.2. Отримавши запит, проксі-сервер аналізує його на основі заданих адміністратором мережі правил та обмежень. Алгоритми опрацювання, аналізу та фільтрації запитів можуть бути доволі складними й багатопараметричними, ураховувати типи файлів, їх сигнатуру та пріоритети користувачів.

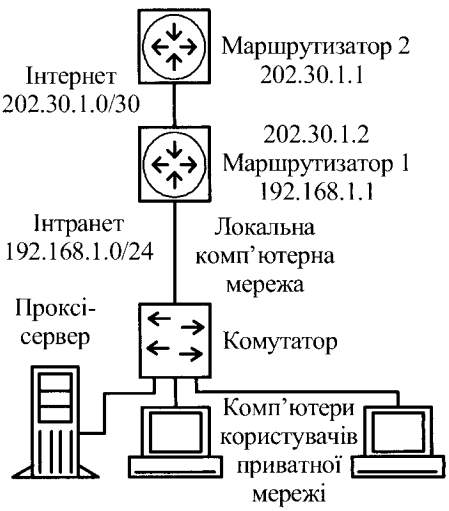


Рис. 6.11. Розміщення проксі-сервера у внутрішній мережі

У результаті проведеного аналізу проксі-сервер визначає рівень допуску користувача й приймає рішення щодо його статусу. Запити внутрішніх користувачів до зовнішніх адресатів проксі-сервер скеровує за адресою 192.168.1.1 на маршрутизатор 1, на якому функціонує NAT-служба. Пакети, які надійшли від зовнішніх користувачів, маршрутизатор 1 скеровує на проксі-сервер для визначення їх статусу й прийняття відповідного рішення.

Окрім основних, проксі-сервер може виконувати також додаткові функції, до найпоширеніших з яких належать зберігання копій (кешування) найбільш популярних веб-сторінок та файлів, збирання статистичних даних про вихід внутрішніх користувачів у мережу Інтернет, які саме сайти вони відвідували та термін перебування там тощо. Функцію ***фільтрації змісту*** (Content filtering) використовують для обмеження доступу користувачів локальної мережі до ресурсів мережі Інтернет із сумнівним вмістом. Залежно від версії, ця функція дає можливість створити чорний або білий список URL або IP-адрес. Слід зазначити, що фільтрацію вмісту можна застосовувати для всіх комп'ютерів локальної мережі або тільки для деяких; також можливо задати періоди часу, коли здійснюють цю фільтрацію. В останні роки функції проксі-служб почали вбудовувати в мережеві операційні системи.

Однак, незважаючи на постійне розширення та вдосконалення методів інформаційної безпеки, використання міжмережевих екранів не забезпечує повного захисту комп'ютерних мереж від несанкціонованого доступу з боку зовнішніх мереж.

6.6. Особливості забезпечення безпеки корпоративних мереж

6.6.1. Особливості будови та захисту корпоративних сховищ даних

Сховище даних (Data Warehouse – DW) – це особливий тип бази даних, яка містить отриману з різних джерел структуровану тематичну інформацію, призначену для використання в системах аналізу та прийняття рішень. На відміну від бази даних сховище накопичує всю необхідну інформацію для виконання задач довгострокового стратегічного керування. Інформація, яка знаходиться в сховищах даних, дає змогу, до прикладу, на основі аналізу перебігу процесу протягом декількох попередніх років розробити перспективний план розвитку підприємств чи організацій, окремих галузей економіки держави, банківської системи, фонду соціального страхування, надходження в бюджет податкових коштів тощо [9–26].

Сховища даних функціонують, здебільшого, у складі ***центру опрацювання даних.*** Їх будують на базі клієнт-серверної архітектури, додатків прийняття рішень та систем управління базами даних. Вони можуть бути побудовані і за відкритою структурою на базі виділених серверів із великим обсягом дискової пам'яті, і за закритою структурою із вбудованою операційною системою. У сховищах можуть використовувати різні види ***систем зберігання даних***, які відрізняють між собою архітектурою й способом з'єднання із клієнтською частиною мережі. Найпоширеніші для зберігання даних у сховищах три види систем зберігання даних: DAS, NAS і SAN.

У ***системах зберігання даних із прямим під'єднанням*** (Direct Attached Storage – DAS) кошик із жорсткими дисками під'єднують безпосередньо до сервера мережі. DAS-система може використовувати декілька файлових серверів з індивідуально під'єднаними пристроями зберігання даних. Проте у разі виходу з ладу сервера, до якого підключений пристрій зберігання даних, дані стають недоступними. Системи знайшли широке використання в корпоративних мережах завдяки простоті адміністрування та низькій вартості.

У ***системах зберігання даних із мережевим під'єднанням*** (Network Attached Storage – NAS) до інтерфейсу локальної мережі під'єднують спеціалізований файловий сервер із набором жорстких дисків, вбудованою операційною системою та набором функцій швидкого доступу до файлів. Системи NAS використовують, здебільшого, у мережах сімейства Ethernet, що забезпечує доступ великої кількості як серверів, так і користувачів локальної мережі до файлів, які зберігають на дисках NAS. Проте файлові сервери системи NAS не дають змоги забезпечити швидкісний і гнучкий доступ до даних на рівні файлових блоків.

У ***мережах зберігання даних*** (Storage Area Network – SAN) різні типи пристроїв зберігання даних (дискові масиви, бібліотеки на магнітних стрічках, оптичні диски тощо) під'єднують до файлових серверів комп'ютерної мережі через спеціальну мережу на базі комутаторів. SAN забезпечує доступ будь-якого сервера локальної мережі до будь-якого пристрою зберігання даних. Перенесення інтенсивного трафіку записування / читання в окрему SAN дає можливість розвантажити локальну мережу та підвищити безпеку зберігання даних.

Інформація, яка міститься в сховищах даних, має, здебільшого, конфіденційний характер, тому її розкриття може призвести до серйозних наслідків. Тому політика безпеки мережі повинна забезпечити створення навколо сховища даних такого периметра інформаційного захисту, який відповідав би вимогам довготривалого, надійного й захищеного зберігання класу C2, а для особливо важливих та конфіденційних даних – класу B1 за класифікацією Помаранчевої книги Міністерства оборони США. Це стосується насамперед конфіденційної інформації, яка може бути пов'язаною з персональними даними, бути державною таємницею тощо.

Слід зазначити, що надійний захист сховища даних, який відповідав би вимогам вказаних класів, може бути забезпечений лише у випадку створення комплексного захисту з використанням різноманітних програмно-апаратних засобів та організаційних заходів, описи та рекомендації з використання яких подано в багатьох відомих документах, зокрема ISO/SES 27001:2005.

Політика інформаційної безпеки центру обробки даних, у складі якого функціонує сховище даних, повинна передбачити якісне вирішення таких базових завдань:

1. Захист локальної мережі та серверів сховища від зовнішнього втручання за допомогою єдиного багаторівневого міжмережевого екрана.

2. Використання VLAN у внутрішній локальній мережі або шифрування як даних, так і службових пакетів.

3. Захист інформації в процесі її передавання відкритими каналами зв'язку з використанням захищених каналів та VPN.

4. Використання багаторівневої системи захисту від вірусів і спаму, яка охоплює всі комп'ютери внутрішньої локальної мережі.

5. Аналіз мережевого трафіку з використанням програмно-апаратних засобів моніторингу мережі, зокрема мережевих моніторів, систем запобігання вторгненням, журналів реєстрації тощо.

6. Виконання згідно з вимогами політики мережевої безпеки резервного копіювання та архівування зашифрованих даних сховища.

У комп'ютерних мережах зі сховищами даних доволі ефективним є використання разом із мережевим екраном зовнішнього периметра виділеного проксі-сервера між серверною і користувацькою частинами мережі.

На рис. 6.12 наведено структуру локальної мережі зі сховищем даних закритої структури на базі системи NAS.

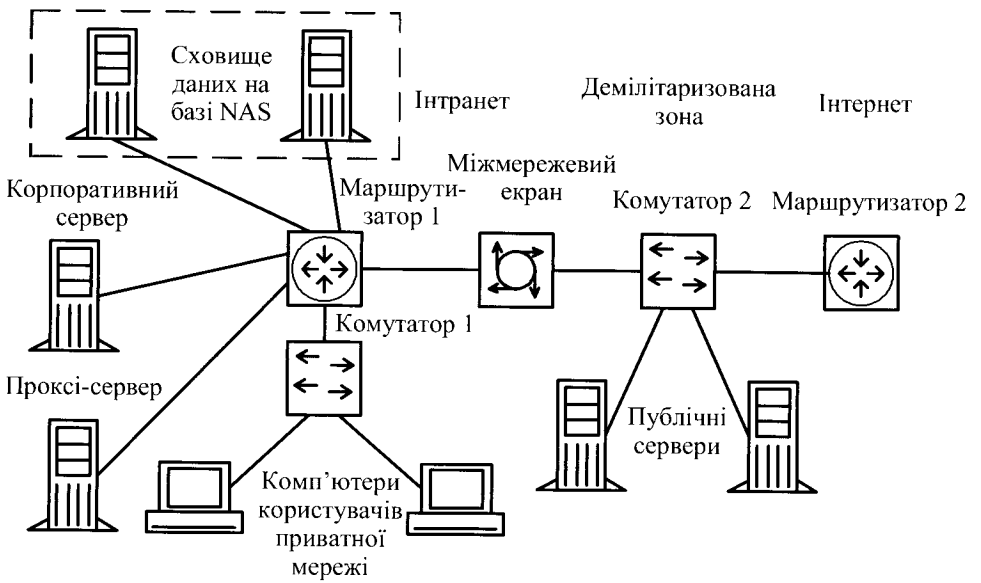


Рис. 6.12. Локальна мережа зі сховищем даних закритої структури на базі системи NAS та внутрішнім проксі-сервером PS

Вбудована операційна система NAS керує роботою серверів сховища, містить набір функцій швидкого доступу до файлів та забезпечує можливість повного дублювання даних сховища. Міжмережевий екран виконує функції шлюзу в публічну мережу та захищає внутрішню мережу від спроб несанкціонованого проникнення пакетів із зовнішніх мереж. Проксі-сервер на базі виділеного комп'ютера містить програмний додаток, який виконує функції

проксі-сервера між розподіленими додатками клієнтів на комп'ютерах користувачів і серверів сховища даних. Клієнтське програмне забезпечення на комп'ютерах користувачів локальної мережі налаштовано так, що всі їхні запити надходять на проксі-сервер, який на основі заданих адміністратором правил аналізує отримані пакети та захищає серверну частину сховища від можливих атак. Правила враховують права доступу користувачів до системи збереження даних, критерії фільтрації пакетів тощо. У таблиці дозволів доступу до системи NAS максимально обмежено діапазони дозволених IP-адрес.

Якщо запит стосується серверів зовнішньої мережі, і користувачу надано на це відповідне право, то проксі-сервер скеровує його пакети на міжмережевий екран. Після опрацювання пакета відповідними проксі-службами й виконання процедури трансляції адрес міжмережевий екран скеровує запит у зовнішню мережу від свого імені, використовуючи тунельний режим протоколу IPSec. У випадку, коли локальний вузол і зовнішній вузол повинні підтримувати зв'язок лише один з одним або для забезпечення доступу зовнішніх користувачів до серверів приватної мережі, адміністратор мережі задає однозначну відповідність внутрішніх адрес зовнішнім із виділеного пула адрес, використовуючи протокол Static NAT.

Спеціальні програми антивірусного захисту сховища перевіряють файли сховища у разі звертання до них користувачів мережі. Проксі-сервер дає змогу читати або змінювати файли, якщо антивірусна програма визнала їх безпечними. За замовчуванням спеціальна програма лікує заражені файли, а якщо лікування неможливе, то видаляє їх. Умовно заражені файли поміщають на карантин. Перед лікуванням або видаленням файла його копію надсилають до резервного сховища. Перевірку сховища адміністратор може запланувати заздалегідь, призначивши її на період низької активності сервера. Протоколи перевірки сховищ даних заносять до журналу звітності.

Структуризацію внутрішньої локальної мережі, її розбиття на VLAN та шифрування внутрішніх пакетів (протоколи TLS (Transport Layer Security – безпека на транспортному рівні), IPSec, SSL тощо) виконують тоді, коли це передбачено політикою інформаційної безпеки. При цьому, через великі обсяги інформації, шифрування даних на рівні ядра, зазвичай, не використовують у зв'язку з виникненням проблем із продуктивністю мережі. Для побудови багаторівневого захисту локальної комп'ютерної мережі від спаму й вірусів часто використовують систему ***Microsoft Fore Front***, яка зарекомендувала себе як надійна та гнучка система захисту.

Контроль доступу до серверів системи зберігання даних висуває вимоги до авторизації та багатофакторної автентифікації користувачів як на рівні операційних і прикладних систем, так і на рівні доступу до операцій із базами

даних. Рівень доступу до програмно-апаратних засобів та даних необхідно визначати лише тим способом і за допомогою тих засобів, які дозволені політикою безпеки. Ця вимога стосується й механізмів контролю запуску авторизованих програм авторизованими користувачами. Використання відповідних прикладних програмних засобів і організаційних заходів дає змогу розподілити права не лише для користувачів, але й для системного адміністратора, адміністратора баз даних та фахівців з інформаційної безпеки. Мандатне керування доступом до баз даних, коли всіх користувачів ділять на групи відповідно до рівня їх уповноважень та приналежності до тієї чи іншої групи суб'єктів, забезпечують при цьому засобами систем керування базами даних. Це дає можливість зберігати в одному сховищі даних інформацію з різним ступенем конфіденційності, обмежуючи при цьому доступ користувачів до даних відповідно до категорій їх допуску.

Для централізованого моніторингу подій у режимі реального часу доцільно використовувати ***системи керування інформацією та повідомленнями безпеки*** (Security Information and Event Management – SIEM). SIEM-системи забезпечують уповноважений персонал та користувачів мережі інформацією про її стан та дають змогу, відповідно до заданих правил та налаштувань, оперативно реагувати на виникнення загрозливих ситуацій. Для аудиту подій у прикладних і операційних системах та комунікаціях часто використовують системи моніторингу корпорацій Cisco Systems, CA Security Command Center, IBM Qradar тощо. Система MS SCOM (Microsoft System Center Operation Manager) забезпечує повний контроль внутрішнього стану серверів, мережі, процесів тощо.

Менеджмент на рівні баз даних виконують засобами системи керування базами даних. Так, система ***SQL Server***, яку використовують у сховищах технологій класу ***OLAP*** (On-Line Analytical Processing – аналітична обробка даних у реальному часі), виконує керування мандатним доступом до даних з урахуванням повноважень груп користувачів, забезпечує двовузлову відмовостійку кластеризацію, автоматизує збирання даних для журналу виконаних акцій, формування звітів тощо.

Незважаючи на те, що в сховищі даних необхідно забезпечувати наскрізну безпеку, вирішальне значення має здатність забезпечити в ньому гнучку багатошарову модель безпеки. Отже, вихід із ладу одного механізму безпеки не призводить до компрометації критично важливої інформації. Oracle Database 10g (Oracle whitepaper) пропонує такі рішення:

– керування доступом на основі ролей (Role-based Access Control – RBAC);

– безпека на рівні рядків (Row-Level Security – RLS);

– віртуальна приватна база даних (Virtual Private Database – VPD);

– безпека Oracle за мітками (Oracle Label Security – OLS).

Описані рішення обмежують доступ користувачів до бази даних, використовуючи різні параметри. Керування доступом на основі ролей визначає чіткі й зрозумілі для користувачів бази даних правила доступу, що змінюються динамічно в процесі функціонування. Безпеку на рівні рядків використовують для контролю доступу до рядків (записів) у таблиці бази даних, обмежуючи рівень доступу за різними критеріями. Віртуальна приватна база даних забезпечує обмежений доступ до даних на рівні рядків і прив'язує політику безпеки до самого об'єкта бази даних. Це дає можливість декільком користувачам мати безпечний прямий доступ до критично важливих даних у межах одного сервера із забезпеченням повного розмежування даних. Опція безпеки Oracle за мітками дає змогу забезпечити віртуальну приватну базу даних контролем доступу за мітками. Віртуальна приватна база даних дає можливість зберігати в одній базі даних інформацію з різним рівнем конфіденційності, обмежуючи доступ до даних за категоріями користувачів.

За статистикою більшість зламів сховищ даних здійснюють безпосереднім копіюванням інформації з бази даних, використанням незахищених резервних чи архівних копій. Для шифрування даних, які зберігають у сховищі, використовують здебільшого симетричний алгоритм блокового шифрування AES, що опрацьовує дані 128-розрядними блоками із ключами завдовжки 128, 192 і 256 бітів. Програму шифрування встановлюють на сервер, до якого безпосередньо під'єднано жорсткі диски. При цьому ключ знаходиться в оперативній пам'яті сервера, усі дані при їх записуванні на диск автоматично зашифровують, а при читанні – розшифровують.

Регламент резервного копіювання та архівування даних установлюють згідно з вимогами інформаційної безпеки та відповідно до затвердженого центром обробки даних планом. Для виконання резервного копіювання й відновлення даних використовують спеціальні програмно-апаратні засоби, які забезпечують високу надійність зберігання резервних копій та архівних даних як на дисках, так і на магнітних стрічках. При плануванні копіювання компонентів сховища даних необхідно оцінювати баланс між рівнем захищеності даних і витраченими на це засобами.

Резервне копіювання всіх даних сховища й серверів доцільно здійснювати на постійній основі. Завдяки цьому дані сховища й будь-який сервер центру обробки інформації можна відновити за допустимий регламентом проміжок часу. Зазвичай для цього використовують сервер резервного копіювання, який координує процеси копіювання та зберігання створених копій даних. Якщо шифрування даних на жорстких дисках серверів сховища є не завжди

виправданим, то резервні копії та архіви зберігають у зашифрованому вигляді. Зберігати магнітні носії із зашифрованими резервними копіями даних та їх архівами необхідно в спеціально виділеному приміщенні, яке перебуває під охороною.

6.6.2. Забезпечення безпеки в мережах Wi-Fi

Архітектура, компоненти й стандарти комп'ютерної мережі Wi-Fi. На сучасному етапі розвитку мережевих технологій технологія безпроводових мереж ***Wi-Fi*** (wireless fidelity – безпроводова точність) є найзручнішою для забезпечення мобільності, простоти встановлення й використання [2–4, 9–26]. Переважно технологію Wi-Fi використовують для організації безпроводових локальних комп'ютерних мереж, а також створення "гарячих точок" високошвидкісного доступу в Інтернет. Базові принципи цієї технології описано в стандарті Radio Enthernet IEEE 802.11 – стандарті організації безпроводових комунікацій на обмеженій території в режимі локальної мережі, тобто коли декілька користувачів мають рівноправний доступ до загального середовища передавання. У стандарті визначено один варіант MAC (Medium Access Control) рівня й три типи фізичних каналів. IEEE 802.11 визначає протокол використання єдиного середовища передавання, що отримав назву ***carrier sense multiple access collision avoidance*** (CSMA/CA). Імовірність колізій безпроводових вузлів мінімізують, попередньо надсилаючи коротке повідомлення – ***ready to send*** (RTS), яке інформує інші вузли про тривалість майбутнього передавання даних і адресата. Це дає змогу іншим вузлам затримати передавання на час, що дорівнює оголошеній тривалості повідомлення. Приймальна станція повинна відповісти на RTS посиланням повідомлення ***clear to send*** (CTS). Це дає можливість передавальному вузлу дізнатись, чи вільне середовище, чи готовий приймальний вузол до приймання. Після отримання пакета даних приймальний вузол повинен передати повідомлення ***підтвердження*** (acknowledge – ACK) факту безпомилкового приймання. Якщо ACK не отримано, спробу передавання пакета даних буде повторено. У стандарті передбачено забезпечення безпеки даних автентифікацією для перевіряння того, чи вузол у мережі є авторизованим у ній, а також шифрування для захисту від підслуховування. На фізичному рівні стандарт передбачає два типи радіоканалів і один інфрачервоного діапазону. В основу стандарту IEEE 802.11 покладено стільникову архітектуру. Мережа може складатися з одного або декількох стільників. Кожним стільником керує базова станція – точка доступу (Access Point – AP). Точка доступу й робочі станції в межах радіуса її дії утворюють ***базову зону***

обслуговування (Basic Service Set – BSS). Точки доступу багатостільникової мережі взаємодіють між собою через ***розподільну систему*** (Distribution System – DS). Уся інфраструктура, що містить точки доступу й розподільну систему, утворює розширену зону ***обслуговування*** (Extended Service Set).

Кожна мережа Wi-Fi має ***унікальне ім'я*** (Service Set Identifier – SSID), що відрізняє одну мережу від іншої. У налаштуваннях усіх пристроїв, які повинні працювати в одній безпроводовій мережі, повинен бути зазначений однаковий SSID, що може містити до 32 символів. Значення SSID на клієнтському пристрої, що дорівнює "ANY", означає можливість підключення до будь-якої доступної мережі.

Стандарт IEEE 802.11 визначає роботу мережі на двох нижніх рівнях моделі ISO/OSI: фізичному і канальному. Інакше кажучи, використовувати обладнання Wi-Fi так само просто, як і Ethernet: протокол TCP / IP накладають поверх протоколу, що описує процес передавання інформації каналом зв'язку. У безпроводовій локальній мережі є два типи обладнання: клієнт (зазвичай це комп'ютер, укомплектований безпроводовою мережевою картою, але може бути й інший пристрій) і точка доступу, яка виконує роль моста між безпроводовою та проводовою мережами. Точка доступу містить приймач-передавач, інтерфейс проводової мережі, а також вбудований мікрокомп'ютер і програмне забезпечення для оброблення даних.

У Wi-Fi мережі можливі такі різновиди з'єднання:

1. З'єднання Ad-Hoc (точка – точка). Усі комп'ютери з'єднуються один з одним.
2. Інфраструктурне з'єднання. Усі комп'ютери підключаються до точки доступу.
3. Точка доступу з маршрутизатором і модемом. Точку доступу через маршрутизатор і модем підключають до інших мереж.
4. Мостове з'єднання. Окремі проводові мережі об'єднують за допомогою точок доступу, які з'єднуються між собою радіоканалом. Цей режим призначений для об'єднання двох і більше мереж.
5. Повторювач. Точка доступу розширює радіус дії іншої точки доступу, яка працює в інфраструктурному режимі.

Шифрування в мережах Wi-Fi за стандартом IEEE 802.11. Пристрої стандарту IEEE 802.11 з'єднуються один з одним, використовуючи для перенесення даних сигнали, які передають у діапазоні радіочастот. Недоліком такого механізму є те, що будь-який інший пристрій, що використовує цей діапазон, також може прийняти ці дані. Якщо не використовувати який-небудь механізм захисту, будь-яка станція стандарту IEEE 802.11 зможе обробити дані, передані безпроводовою локальною мережею, якщо тільки її приймач працює в

тому самому радіодіапазоні. Для забезпечення хоча б мінімального рівня безпеки необхідні такі компоненти: автентифікація користувача в мережі; використання алгоритмів шифрування даних.

У специфікації стандарту IEEE 802.11 визначено застосування механізму автентифікації пристроїв з відкритим та спільно використовуваним ключем і технології ***еквівалентної проводовій конфіденційності*** (Wired Equivalent Privacy – WEP). Алгоритми автентифікації з відкритим та спільно використовуваним ключем основано на WEP-шифруванні й застосуванні WEP-ключів для контролю доступу. Використовують два види шифрів: потоковий (груповий) та блоковий.

Шифри обох типів працюють, генеруючи ***ключовий потік*** (key stream), одержуваний на основі значення секретного ключа. Ключовий потік змішують з даними або відкритим текстом, внаслідок чого отримують шифрограму.

Потоковий шифр генерує безперервний ключовий потік, ґрунтуючись на значенні ключа. Наприклад, потоковий шифр може генерувати 15-розрядний ключовий потік для шифрування одного пакета й 200-розрядний ключовий потік для шифрування іншого. Потокові шифри – це невеликі й ефективні алгоритми шифрування, завдяки яким навантаження на центральний процесор виявляється невеликим. Найпоширенішим є потоковий шифр ***RC4*** (Rivest's Cipher 4), який і покладено в основі алгоритму WEP.

Блоковий шифр, навпаки, генерує єдиний ключовий потік шифрування фіксованого розміру. Відкритий текст ділять на блоки, і кожний блок змішують із ключовим потоком незалежно. Якщо блок відкритого тексту менший, ніж блок ключового потоку, перший доповнюють із метою отримання блоку потрібного розміру. Процес фрагментації, а також інші особливості шифрування з використанням блокового шифру спричиняють підвищене, порівняно з потоковим шифруванням, навантаження на центральний процесор, що понижує продуктивність таких пристроїв.

Описаний вище процес шифрування для потокових і блокових шифрів називають режимом шифрування за допомогою ***книги електронних кодів*** (Electronic Code Book – ECB). Цей режим характеризується тим, що один і той самий відкритий текст після шифрування перетворять на один і той самий зашифрований текст. Цей чинник є потенційною загрозою для безпеки, оскільки зловмисники можуть отримувати зразки зашифрованого тексту й висувати якісь припущення про початковий текст. Цю проблему можливо вирішити використанням для шифрування ***векторів ініціалізації*** (initialization vectors – IV) та ***режими зі зворотним зв'язком*** (feedback modes).

Вектор ініціалізації – це послідовність символів, яку додають до ключа, результатом чого є зміна інформації ключового потоку. Вектор ініціалізації

зв'язують із ключем до того, як почнеться генерація ключового потоку. Вектор ініціалізації увесь час змінюють; те саме відбувається із ключовим потоком. Стандарт IEEE 802.11 рекомендує змінювати вектор ініціалізації для кожного пакета даних (on a per-frame basis). Це означає, що якщо один і той самий пакет буде переданий двічі, імовірно, що зашифрований текст буде різним.

Режими зі зворотним зв'язком – модифікації процесу шифрування, які використовують для уникнення ситуації, за якої один і той самий відкритий текст перетворюють у процесі шифрування на однаковий зашифрований текст.

Специфікація стандарту IEEE 802.11 передбачає забезпечення захисту даних з використанням алгоритму WEP, основаного на застосуванні симетричного потокового шифру RC4. Симетричність RC4 означає, що узгоджені WEP-ключі розміром 40 або 104 біт статично конфігурують на кінцевому обладнанні користувача й у точці доступу. Алгоритм WEP вибрано переважно тому, що він не потребує об'ємних обчислень. WEP – простий у застосуванні алгоритм, для запису якого в деяких випадках достатньо 30 рядків коду. Малі непродуктивні витрати, що виникають у разі застосування цього алгоритму, роблять його ідеальним алгоритмом шифрування для спеціалізованих пристроїв. Щоб уникнути шифрування в режимі ECB, WEP використовує 24-розрядний вектор ініціалізації, який додають до ключа перед обробленням згідно з алгоритмом RC4.

Вектор ініціалізації необхідно змінювати для кожного пакета даних, щоб уникнути його колізій. Колізії такого роду виникають, коли використовують той самий вектор ініціалізації та той самий WEP-ключ, внаслідок чого для шифрування пакета даних використовують той самий ключовий потік. Така колізія надає зловмисникам великі можливості для розшифрування шифрограми порівнянням подібних елементів. За використання вектора ініціалізації важливо запобігти подібному сценарію, тому вектор ініціалізації часто змінюють.

Специфікація стандарту IEEE 802.11 передбачає, що однакові WEP-ключі сконфігуровано як на кінцевому обладнанні користувачів, так і на точках доступу. Можна визначати до чотирьох ключів на один пристрій, але одночасно для шифрування пакетів, що передають, використовують тільки один із них. WEP-шифрування використовують лише до пакетів даних і під час процедури автентифікації зі спільно використовуваним ключем. При цьому шифрують такі поля пакета даних стандарту IEEE 802.11: дані або ***корисне навантаження*** (payload), ***контрольна ознака цілісності*** (integrity check value – ICV).

Інші поля пакета даних передають без шифрування. Вектор ініціалізації надсилають незашифрованим усередині пакета, щоб приймальна станція могла отримати його й використати для коректного розшифрування корисного навантаження й ICV.

Додатково до шифрування даних специфікація стандарту IEEE 802.11 пропонує використовувати 32-розрядне значення, функція якого – контролювати цілісність. Ця контрольна ознака цілісності говорить приймачу про те, що пакет отримано без пошкодження в процесі передавання. Контрольну ознаку цілісності обчислюють за всіма полями пакета з використанням 32-розрядної поліноміальної функції контролю за допомогою ***циклічного надлишкового коду*** (cyclic redundancy code – CRC-32). Станція-відправник обчислює це значення й розміщує результат у полі ICV. Значення поля ICV вміщують до складу частини пакета, зашифрованої за алгоритмом WEP так, що його не можуть просто так прочитати зловмисники. Отримувач пакета дешифрує його, обчислює значення ICV і порівнює результат зі значенням поля ICV отриманого пакета. Якщо ці значення збіглися, пакет вважають непідробленим. Якщо вони не збігаються, такий пакет відкидають.

Найсерйозніші й непереборні проблеми захисту мереж стандарту IEEE 802.11 з використанням WEP-шифрування виявили криптоаналітики Флурер (Fluhrer), Мантін (Mantin) і Шамір (Shamir). Вони показали, що WEP-ключ можна отримати, пасивно накопичуючи окремі пакети, які передають у безпроводовій мережі.

Уразливість зумовлена тим, як механізм WEP застосовує ***алгоритм складання ключа*** (key scheduling algorithm – KSA) на основі потокового шифру RC4. Частина векторів ініціалізації (їх називають слабкі IV – weak IV) можуть розкрити біти ключа в результаті проведення статистичного аналізу. Дослідники компанії AT&T і університету Rice скористалися цією уразливістю й з'ясували, що можна дістати WEP-ключі завдовжки 40 або 104 біти після оброблення 4 мільйонів пакетів. Для перших безпроводових мереж стандарту IEEE 802.11b це означає, що вони повинні передавати пакети приблизно одну годину, після чого можна виявити 104-розрядний WEP-ключ. Подібна уразливість робить WEP неефективним механізмом забезпечення захисту інформації.

Автентифікація в мережах Wi-Fi за стандартом IEEE 802.11. Специфікація стандарту IEEE 802.11 визначає два механізми, які можна застосувати для автентифікації користувачів мережі:

– відкрита автентифікація (open authentication);

– автентифікація зі спільно використовуваним ключем (shared key authentication).

Відкрита автентифікація по суті є алгоритмом із нульовою автентифікацією (null authentication algorithm). Точка доступу приймає будь-який запит на автентифікацію. Спрощені вимоги до автентифікації дають змогу пристроям швидко отримати доступ до мережі.

Контроль доступу за відкритої автентифікації здійснюють, використовуючи наперед сконфігурований WEP-ключ у точці доступу та на обладнанні користувача. Вони повинні мати однакові ключі, щоб могти зв'язуватися між собою. Якщо точка доступу та обладнання користувача не підтримують алгоритм WEP, у базовій зоні обслуговування неможливо забезпечити захист. Будь-який пристрій може підключитися до такої зони, і всі пакети даних передаватимуться незашифрованими.

Після виконання відкритої автентифікації та завершення процесу з'єднання користувач може почати передавання та приймання даних. Якщо ключ користувача відрізняється від ключа точки доступу, пакети не передаватимуться.

На відміну від відкритої автентифікації, при автентифікації зі спільно використовуваним ключем потрібно, щоб кінцеве обладнання користувача й точка доступу були здатні підтримувати WEP і мали однакові WEP-ключі. Процес автентифікації зі спільно використовуваним ключем є таким.

1. Клієнт надсилає точці доступу запит на автентифікацію зі спільно використовуваним ключем.
2. Точка доступу відповідає пакетом виклику (challenge frame), що містить відкритий текст.
3. Клієнт шифрує виклик і надсилає його точці доступу.
4. Якщо точка доступу може правильно розшифрувати цей пакет і отримати свій початковий виклик, вона надсилає клієнту повідомлення про успішну автентифікацію.
5. Клієнт отримує доступ до мережі.

Передумови, на яких основано автентифікацію зі спільно використовуваним ключем, точно такі самі, як і ті, які передбачено за відкритої автентифікації, що використовує WEP-ключі як засіб контролю доступу. Різниця між цими двома методами полягає в тому, що клієнт не може з'єднатись з точкою доступу у разі використання механізму автентифікації зі спільно використовуваним ключем, якщо його ключ належно не сконфігуровано.

Вразливість полягає в тому, що зловмисник у процесі прослуховування отримує як відкритий, так і зашифрований текст. Ці дані потім використовують для впровадження фальшивих пакетів-запитів автентифікації, примушуючи точку доступу генерувати пакети, що містять вектор ініціалізації. Для того, щоб розшифрувати ключ, буває достатньо від 20 до 100 тис. пакетів із вектором ініціалізації.

Автентифікацію з використанням MAC-адрес не специфіковано стандартом IEEE 802.11, але забезпечено багатьма виробниками. Під час автентифікації з використанням MAC-адрес перевіряють наявність MAC-адреси кінцевого

обладнання користувача в списку дозволених адрес або списку, що зберігається на зовнішньому автентифікаційному сервері. Автентифікація з використанням MAC-адрес підсилює дію відкритої автентифікації й автентифікації зі спільно використовуваним ключем, які забезпечено стандартом IEEE 802.11, потенційно знижуючи імовірність того, що неавторизовані пристрої отримають доступ до мережі. Проте такий захист не є надійним. Для підключення до точки доступу, яка фільтрує користувачів за MAC-адресою їх адаптера, зловмиснику необхідно знати лише MAC-адресу вже підключеного адаптера клієнта. Потім йому треба змінити MAC-адресу свого адаптера на клієнтську, дочекатися, поки легітимний клієнт відключиться, і тоді підключитися або використати атаку деавтентіфікації легітимного клієнта.

У разі використання механізму відкритої автентифікації точка доступу не має можливості перевірити правомочність клієнта. Відсутність такої можливості є недоліком системи захисту, якщо в беспроводовій локальній мережі не використовують WEP-шифрування. Навіть за використання й клієнтом, і точкою доступу статичного WEP механізм відкритої автентифікації не надає засобів для визначення того, хто використовує пристрій у мережі. Авторизований пристрій у руках неавторизованого користувача – це загроза безпеки, рівносильна повній відсутності будь-якого захисту мережі.

Стандарт IEEE 802.1X. Як показав час, WEP виявилася недостатньо надійною технологією захисту. Після 2001 року для проводових і безпроводових мереж запроваджено новий стандарт IEEE 802.1X, який використовує варіант динамічних (періодично змінних у часі) 128-розрядних ключів шифрування. Тобто користувачі мережі використовують окремі сеанси зв'язку, після закінчення яких їм надсилають новий ключ. Наприклад, Windows XP підтримує такий стандарт, і за замовчуванням час одного сеансу дорівнює 30 хвилин. IEEE 802.1X – це новий стандарт, який виявився ключовим для розвитку індустрії безпроводових мереж загалом. За основу взято виправлення недоліків технологій безпеки, застосованих в IEEE 802.11, зокрема можливість зламу WEP, залежність від технологій виробника тощо. З іншого боку, IEEE 802.1X і IEEE 802.11 є сумісними стандартами. В IEEE 802.1X застосовують той самий алгоритм, що і в WEP, а саме – RC4, але з деякими відмінностями. IEEE 802.1X ґрунтується на ***протоколі розширюваної автентифікації*** (Extensible Authentication Protocol – EAP).

Протокол EAP (RFC 2284) і стандарт IEEE 802.1X не регламентують використання особливого алгоритму автентифікації. Адміністратор мережі може застосовувати відповідний протоколу EAP різновид автентифікації – або IEEE 802.1X, або EAP. Єдина вимога – щоб як клієнт стандарту IEEE 802.11 (тут він називається претендент (supplicant)), так і сервер автентифікації

підтримували алгоритм EAP-автентіфікації. Така відкрита й розширювана архітектура дає можливість використовувати базову автентифікацію в різних умовах, і в кожній ситуації можна застосовувати відповідний різновид автентифікації.

Автентифікація за стандартом IEEE 802.1X потребує наявності трьох груп пристроїв:

– претендент (Supplicant) – кінцеве обладнання користувача;

– перевіряльник (Authenticator) – точка доступу;

– сервер автентифікації (Authentication Server) – сервер RADIUS.

Процес автентифікації складається з таких стадій:

– претендент надсилає запит на автентифікацію (EAP-start message) до перевіряльника;

– перевіряльник у відповідь надсилає претенденту запит на ідентифікацію претендента (EAP-request/identity message); перевіряльник може надіслати EAP-request самостійно, якщо побачить, що будь-який із його портів перейшов в активний стан;

– претендент надсилає пакет відповіді (EAP-response packet) з потрібними даними, який перевіряльник перескеровує до сервера автентифікації;

– сервер автентифікації надсилає перевіряльнику запит інформації про автентичність претендента (challenge-пакет); перевіряльник пересилає його претенденту;

– відбувається процес взаємної ідентифікації сервера автентифікації та претендента;

– на наступній стадії сервер автентифікації, отримавши від претендента необхідну інформацію, дозволяє (accept) або забороняє (reject) йому доступ із пересиланням такого повідомлення перевіряльнику; перевіряльник відкриває порт для претендента, якщо з боку сервера автентифікації прийшла позитивна відповідь (accept);

– порт на точці доступу відкривається, перевіряльник пересилає претенденту повідомлення про успішне завершення процесу, і претендент отримує доступ до мережі;

– після відключення кінцевого обладнання користувача порт на точці доступу знову переходить у стан "зачинений".

Стандарт WPA. У кінці 2003 року було впроваджено стандарт ***Wi-Fi Protected Access*** (захищений доступ до безпроводових мереж – WPA), який суміщає переваги динамічного оновлення ключів IEEE 802.1X із кодуванням ***протоколу інтеграції тимчасового ключа*** (Temporal Key Integrity Protocol – TKIP), протоколом розширеної автентифікації (EAP) і ***технології перевіряння цілісності повідомлень*** (Message Integrity Check – MIC). WPA – це тимчасовий

стандарт, про який домовилися виробники обладнання, поки не впроваджено стандарт IEEE 802.11i. WPA є сумою протоколу розширеної автентифікації (Extensible Authentication Protocol – EAP), протоколу інтеграції тимчасового ключа (Temporal Key Integrity Protocol – TKIP) та технології перевірки цілісності повідомлень (MIC).

Від зовнішнього проникнення й зміни інформації також захищає технологія MIC. Достатньо складний математичний алгоритм дає змогу звіряти відправлені в одній точці й отримані в інший дані. Якщо помічено зміни й результат порівняння не збігається, такі дані вважають помилковими й викидають. Завдяки MIC можна ліквідувати слабкі місця захисту, що сприяють проведенню атак із використанням підроблених пакетів і жонглювання бітами. IEEE запропонував спеціальний алгоритм, що отримав назву Michael (Майкл), щоб підсилити роль ICV в шифруванні пакетів даних стандарту IEEE 802.11.

Алгоритм MIC полягає у виконанні приймачем таких дій:

– приймач видаляє існуючий ключ на з'єднання;

– приймач реєструє проблему як таку, що стосується безпеки мережі;

– під'єднаного користувача, від якого отримано помилковий пакет, не можна під'єднати і автентифікувати протягом 60 секунд, щоб уповільнити атаку;

– якщо користувач отримав помилковий пакет, то він відкидає всі пакети, які не відповідають стандарту IEEE 802.1X; такий користувач також просить новий ключ.

Протокол TKIP використовує автоматично підібрані 128-бітові ключі, які формують непередбачуваним способом, і загальна кількість варіацій яких досягає 500 мільярдів. Складна ієрархічна система алгоритму підбору ключів і їх динамічна заміна через кожні 10 кБ (10000 переданих пакетів) роблять систему максимально захищеною.

WPA працює в двох режимах автентифікації: персональному (Personal) і корпоративному (Enterprise).

У режимі WPA-Personal із введеної відкритим текстом парольної фрази генерують 256-розрядний ключ, який називають ***попередньо розподіленим ключем*** (PreShared Key – PSK). PSK, а також унікальне ім'я мережі (SSID) і використовують для формування ***головного парного ключа*** (Pairwise Master Key – PMK), який використовують для започаткування (ініціалізації) чотиристоронньої "рукостискання" (handshake) (Pairwise Transient Key – PTK) для взаємодії безпроводового пристрою користувача з точкою доступу.

Повідомлення чотиристоронньої "рукостискання" (рис. 6.13) містять інформаційні поля такого вмісту:

– MAC-адресу точки доступу;

– MAC-адресу пристрою клієнта;

– випадкове 32-байтове число, що генерує точка доступу при встановленні з'єднання (ANounce) – повідомлення 1;

– випадкове 32-байтове число, що генерує пристрій клієнта (SNounce) – повідомлення 2;

– розмір поточного повідомлення автентифікації (без канального заголовка) – повідомлення 2, 3, 4;

– ключ цілісності повідомлення (MIC);

– версія протоколу захисту даних.

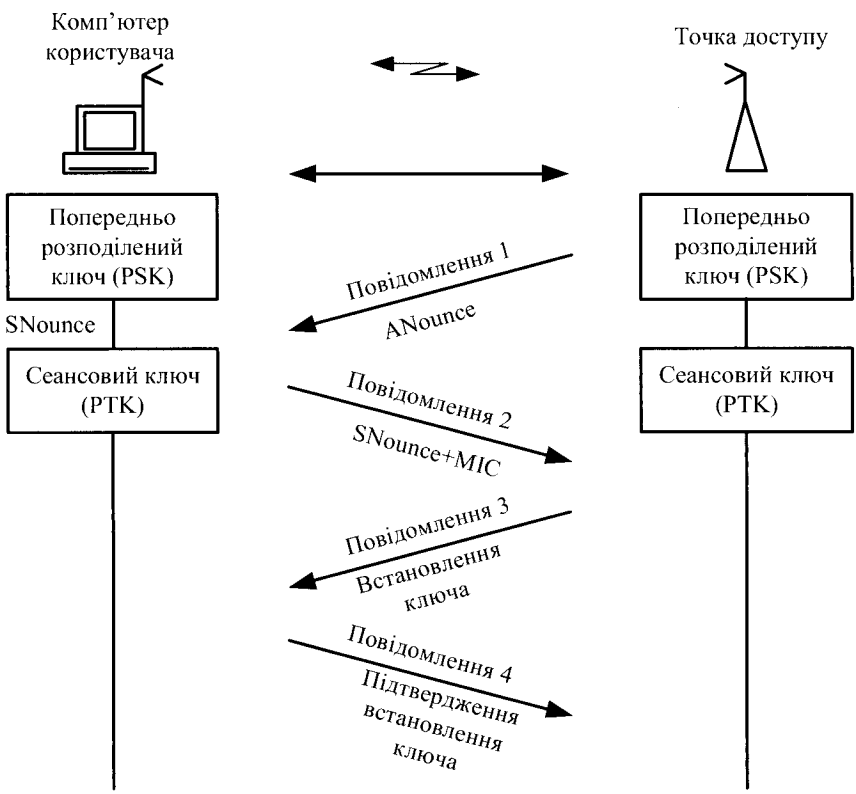


Рис. 6.13. Процедура чотиристороннього "рукостискання"

Єдиним способом атаки на WPA-мережі є зламування захоплених рукостискань. Проводять її методом брутальної атаки (перебору паролів). Оскільки використання брутальної атаки не гарантує позитивного результату, зловмисники розробили кілька технік, які дають змогу істотно підвищити шанси на успіх, зокрема такі:

– використання відеокарт для підбирання паролів, що значно збільшує швидкість перебирання;

– використання таблиць із попередньо розрахованими хешами, що дає можливість використати їх повторно для однієї й тієї самої точки доступу та перепробувати значну кількість рукостискань від однієї точки доступу за відносно короткий проміжок часу;

– використання словників значних обсягів.

Стандарт WPA2. Стандарт IEEE 802.11i (WPA2) використовує концепцію ***мережі підвищеної безпеки*** (Robust Security Network – RSN), яка передбачає, що безпроводові пристрої повинні забезпечувати додаткові можливості. Це вимагає змін в апаратній частині та програмному забезпеченні, тобто мережа, що повністю відповідає RSN, стає несумісною з існуючим обладнанням WEP. Сьогодні підтримують обладнання як RSN, так і WEP, але надалі пристрої WEP не використовуватимуть.

IEEE 802.11i є придатним для різних варіантів побудови мереж і може задіювати TKIP, але за замовчуванням RSN використовує AES і ***CCMP*** (Counter Mode CBC MAC Protocol), тобто, є потужнішим розширюваним рішенням.

У концепції RSN для шифрування застосовують AES, подібно до того, як алгоритм RC4 задіяний у WPA. Однак механізм шифрування є набагато складнішим і не має недоліків, що були в WEP. AES – блоковий шифр, який оперує блоками даних по 128 бітів. CCMP, своєю чергою, – протокол безпеки, що використовує AES. Він є еквівалентом TKIP в WPA. CCMP обчислює MIC, вдаючись до добре відомого й перевіреного методу ***CBC-MAC*** (Cipher Block Chaining Message Authentication Code). Зміна навіть одного біта в повідомленні призводить до зовсім іншого результату.

RSN визначає ієрархію ключів з обмеженим терміном дії, схожу з TKIP. У AES/CCMP, щоб умістити всі ключі, потрібно 512 біти – менше ніж у TKIP. В обох випадках майстер-ключі використовують не безпосередньо, а для отримання інших ключів. Адміністратор повинен забезпечити єдиний майстер-ключ. Повідомлення складаються з 128-бітового блоку даних, зашифрованого секретним ключем такої самої довжини (128 бітів). Кінцевим результатом є шифр, який набагато складніший, ніж WPA.

WPA2 так само, як і WPA, використовує чотиристоронє "рукостискання" як механізм автентифікації. У 2017 році дослідник ***Мегі Ванхов*** (Mathy Vanhoef) опублікував докладний опис вразливості ***KRACK*** (Key Reinstallation Attacks – атака переустановлення ключів), що дає змогу зловмисникам, які знаходяться в зоні дії Wi-Fi жертви, виконати вимушене переустановлення унікальних ключів шифрування, які захищають трафік WPA2. На цей час

виконують роботу над необхідними патчами безпеки (патч – програма, що автоматично виправляє вразливість). Виправлення стосуватимуться механізму "рукостискань" і зможуть гарантувати, що кожен ключ шифрування буде використано лише один раз.

6.7. Основи технології віртуальних приватних мереж

6.7.1. Основні поняття й функції мережі VPN

Під час підключення приватної (корпоративної) локальної мережі (мережі приватної особи, корпорації, підприємства чи організації) до публічної (відкритої) мережі виникають загрози безпеки двох основних типів [9–22]:

– несанкціонований доступ до внутрішніх ресурсів приватної локальної мережі, отримуваний зловмисником у результаті несанкціонованого входу до цієї мережі;

– несанкціонований доступ до приватних даних у процесі їх передавання відкритою мережею.

Забезпечення безпеки інформаційної взаємодії локальних мереж і окремих комп'ютерів через відкриті мережі, зокрема через мережу Інтернет, можливе завдяки ефективному вирішенню таких завдань:

– захист підключених через відкриті лінії зв'язку локальних мереж і окремих комп'ютерів від несанкціонованих дій із боку зовнішнього середовища;

– захист інформації в процесі її передавання відкритими лініями зв'язку.

Як уже зазначали, для захисту локальних мереж і окремих комп'ютерів від несанкціонованих дій із боку зовнішнього середовища зазвичай використовують міжмережеві екрани, що підтримують безпеку інформаційної взаємодії фільтрацією двостороннього потоку повідомлень, а також виконання функцій посередництва під час обміну інформацією. Міжмережеві екрани розташовують на стику між локальною й відкритою мережею. Для захисту окремого віддаленого комп'ютера, підключеного до відкритої мережі, на цьому комп'ютері встановлюють міжмережевий екран, який називають персональним.

Захищають дані під час передавання відкритими лініями зв'язку, створюючи окремі ***захищені канали*** або ***віртуальні приватні мережі***.

Канал вважають захищеним, якщо забезпечено виконання трьох основних умов:

– під час встановлення з'єднання виконано взаємну автентифікацію обох користувачів;

– виконано захист даних від несанкціонованого доступу під час їх передавання каналом;

– підтверджено цілісність отриманого повідомлення.

Ці умови можна виконати, обмінюючись паролями, шифруючи інформацію та передаючи її разом із повідомленням цифрового підпису.

Розрізняють дві схеми створення ***захищеного каналу***:

– захищений канал між користувачами різних приватних мереж із використанням відкритих ліній зв'язку;

– захищений канал між двома приватними мережами через публічну проміжну мережу, зокрема мережу Інтернет.

У першому випадку захищений канал створюють програмними засобами окремих комп'ютерів двох мереж (двох філій мережі підприємства), трафік між якими проходить через відкриті лінії зв'язку. Перевагою такої схеми є захист трафіку вздовж усього шляху проходження пакетів між двома комп'ютерами, а також вільний вибір користувачами погодженого протоколу захисту.

У другому випадку захищені канали створюють на прикордонному обладнанні інтернет-провайдерів, і захищають трафік лише під час проходження пакетів через публічну мережу. Такий централізований захист контролюють адміністратори мереж, а для клієнтів кінцевих мереж він є прозорим. Недоліком такої схеми є відсутність захисту каналів доступу до публічної мережі, а також залежність користувачів мережі підприємства від постачальника послуг.

Віртуальна приватна (захищена) мережа (Virtual Private Network – VPN) – це об'єднання локальних мереж і окремих комп'ютерів через публічну мережу в єдину віртуальну мережу, що забезпечує безпеку передавання даних. Віртуальні приватні мережі створюють з використанням сукупності віртуальних захищених каналів (криптозахищених тунелів), створюваних на базі відкритих каналів зв'язку публічної мережі. Ці віртуальні захищені канали називають ***тунелями VPN***. Канали VPN призначені для комплексного захисту інформації в процесі її передавання відкритими лініями зв'язку. Їх основним призначенням є забезпечення захищеним зв'язком центрального офісу підприємства чи організації з віддаленими філіями. Кожен такий тунель є логічним з'єднанням, здійсненим через відкриту мережу, по якому передають пакети повідомлень віртуальної мережі. Створення криптозахищених тунелів між брандмауерами двох приватних мереж, в яких знаходяться кінцеві користувачі, забезпечує безпечне передавання даних через публічну мережу. Мережа VPN дає змогу за допомогою тунелів VPN з'єднати центральний офіс підприємства, офіси філій, офіси бізнес-партнерів і віддалених користувачів, а також безпечно передавати інформацію через Інтернет.

Тунель VPN є з'єднанням, здійсненим через відкриту мережу, по якому передають криптографічно захищені пакети повідомлень віртуальної мережі. Захист інформації в процесі її передавання по тунелю VPN ґрунтується на автентифікації взаємодіючих сторін, шифруванні переданих даних, перевірці автентичності та цілісності доставленої інформації.

Для створення захищених каналів використовують криптографічні методи захисту інформації. Ефективність такого захисту забезпечують завдяки спільному використанню симетричних і асиметричних криптографічних методів. Тунель VPN, сформований пристроями VPN, має властивості захищеної виділеної лінії, яку розгортають у межах публічної мережі, наприклад, Інтернет. Пристрої VPN можуть виконувати у віртуальних приватних мережах роль VPN-клієнта, VPN-сервера або шлюзу безпеки VPN.

VPN-клієнт є програмним або програмно-апаратним комплексом, який виконують зазвичай на базі персонального комп'ютера. Його мережеве програмне забезпечення модифікують для шифрування й автентифікації трафіку, яким цей пристрій обмінюється з іншими VPN-клієнтами, VPN-серверами або шлюзами безпеки VPN. Зазвичай VPN-клієнт виконують у вигляді програми, яка доповнює стандартну операційну систему – представника сімейств Windows або Unix.

VPN-сервер є програмним або програмно-апаратним комплексом, який установлюють на комп'ютері, що виконує функції сервера. VPN-сервер забезпечує захист серверів від несанкціонованого доступу із зовнішніх мереж, а також організацію ***захищених з'єднань*** (безпечних асоціацій) з окремими комп'ютерами та з комп'ютерами із сегментів локальних мереж, захищених відповідними VPN-продуктами. VPN-сервер є функціональним аналогом продукту VPN-клієнта для серверних платформ. Він відрізняється насамперед розширеними ресурсами для підтримки множинних з'єднань з VPN-клієнтами. VPN-сервер може підтримувати захищені з'єднання з мобільними користувачами.

Шлюз безпеки VPN (security gateway) – це мережевий пристрій, що підключають між двома мережами: приватною й публічною. Він виконує функції шифрування й автентифікації для численних вузлів мережі, розташованих за ним. Розміщений шлюз безпеки VPN так, щоб через нього проходив увесь трафік, призначений для приватної мережі. Мережеве з'єднання шлюзу VPN прозоре для користувачів позаду шлюзу й виглядає для них немов виділена лінія, хоча насправді здійснено через відкриту мережу з комутацією пакетів. Адресу шлюзу безпеки VPN зазначають як зовнішню адресу вхідного пакета, а внутрішня адреса пакета є адресою конкретного вузла позаду шлюзу. Шлюз безпеки VPN можна здійснити у вигляді окремого програмного рішення,

окремого апаратного пристрою, а також у вигляді маршрутизатора або міжмережевого екрана, доповнених функціями VPN.

Відкрите зовнішнє середовище передавання інформації може містити і канали швидкісного передавання даних, у якості якого використовують мережу Інтернет, і повільніші загальнодоступні канали електрозв'язку, в якості яких зазвичай застосовують канали телефонної мережі. Ефективність віртуальної приватної мережі VPN залежить від ступеня захищеності інформації, що передають відкритими каналами електрозв'язку. Для безпечного передавання даних через відкриті мережі широко використовують ***інкапсуляцію*** й ***тунелювання***. За допомогою методики тунелювання пакети даних передають через загальнодоступну мережу, як при встановленні звичайного двоточкового з'єднання. Між кожною парою "відправник – отримувач даних" встановлюють своєрідний тунель – логічне з'єднання, що дає змогу інкапсулювати (вмістити) дані одного протоколу в пакети іншого.

Суть тунелювання полягає в тому, щоб інкапсулювати частину даних разом зі службовими полями в новий пакет. При цьому пакет протоколу нижчого рівня розміщують у поле даних пакета протоколу вищого або такого самого рівня. Слід зазначити, що тунелювання саме по собі не захищає дані від несанкціонованого доступу або спотворення, але завдяки тунелюванню з'являється можливість повного криптографічного захисту інкапсульованих вихідних пакетів. Щоб забезпечити конфіденційність переданих даних, відправник шифрує вихідні пакети, упаковує їх у зовнішній пакет із новим IP-заголовком і відправляє відкритою мережею. Наприклад, вихідний IP-пакет шифрують з використанням протоколів AH та ESP і після цього вміщують у поле даних у новому IP-пакеті (рис. 6.14).

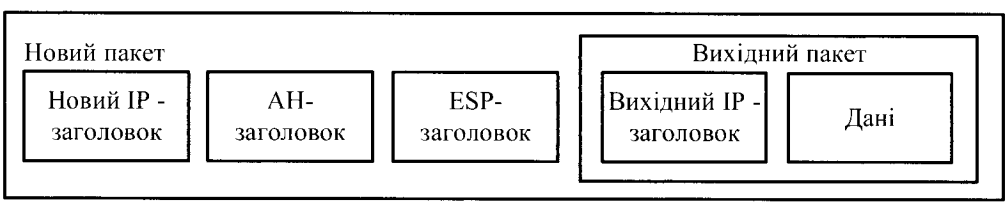


Рис. 6.14. Структура пакета при тунелюванні

Тунелювання можна використати для захисту не лише конфіденційності вмісту пакета, але і його цілісності та автентичності; при цьому електронний цифровий підпис можна поширити на всі поля пакета.

На додачу до приховування мережевої структури між двома точками, тунелювання може також запобігти можливому конфлікту адрес між двома локальними мережами. При створенні локальної мережі, не пов'язаної з

Інтернетом, компанія може використовувати будь-які IP-адреси для своїх мережевих пристроїв та комп'ютерів. У разі об'єднання раніше ізольованих мереж ці адреси можуть почати конфліктувати одна з однією та з адресами, які вже використовують в Інтернеті. Інкапсуляція пакетів вирішує цю проблему, оскільки дає можливість приховати початкові адреси та додати нові, унікальні в просторі IP-адрес Інтернету, які потім використовують для передавання даних окремими розділеними мережами. Сюди ж входить завдання налаштування IP-адреси та інших параметрів для мобільних користувачів, обладнання яких підключають до локальної мережі.

Механізм тунелювання широко застосовують у різних протоколах формування захищеного каналу. Зазвичай тунель створюють тільки на ділянці відкритої мережі, де існує загроза порушення конфіденційності й цілісності даних, наприклад, між точкою входу у відкритий Інтернет і точкою входу до приватної мережі. При цьому для зовнішніх пакетів використовують адреси прикордонних маршрутизаторів, установлених у цих двох точках, а внутрішні адреси кінцевих вузлів містяться у внутрішніх вихідних пакетах у захищеному вигляді. Слід зазначити, що сам механізм тунелювання не залежить від того, з якою метою застосовують тунелювання. Його можливо застосовувати не тільки для забезпечення конфіденційності та цілісності всієї переданої частини даних, але й для організування переходу між мережами з різними протоколами (наприклад, IPv4 і IPv6). Тунелювання дає змогу організувати передавання пакетів одного протоколу в логічному середовищі, що використовує інший протокол. У результаті з'являється можливість вирішити проблеми взаємодії декількох різнотипних мереж, починаючи з необхідності забезпечення цілісності й конфіденційності даних і закінчуючи подоланням невідповідностей зовнішніх протоколів або схем адресації.

Здійснення механізму тунелювання можна представити як результат роботи протоколів трьох типів: протоколу-"пасажира", носійного протоколу й протоколу тунелювання. Наприклад, як протокол-"пасажир" можна використати транспортний протокол ***IPX*** (Internetwork Packet Exchange) або протокол ***IP*** (Internet Protocol), що переносить дані в локальних мережах філій одного підприємства. Як протоколи тунелювання можна використати протоколи канального чи мережевого рівня, що будуть розглянуті нижче. Найпоширенішим варіантом носійного протоколу є протокол ***IP***. Завдяки тунелюванню стає можливим приховування інфраструктури Інтернету від VPN-додатків.

Тунелі VPN можливо створювати для різних типів кінцевих користувачів: локальної мережі зі шлюзом безпеки або окремих комп'ютерів віддалених і мобільних користувачів. Для створення віртуальної приватної мережі великого підприємства потрібні VPN-шлюзи, VPN-сервери й VPN-клієнти. VPN-шлюзи

доцільно використовувати для захисту локальних мереж підприємства, VPN-сервери й VPN-клієнти використовують для організації захищених з'єднань віддалених і мобільних користувачів із приватною мережею через Інтернет.

6.7.2. Варіанти побудови віртуальних захищених каналів

Безпеку інформаційного обміну необхідно забезпечувати як у випадку об'єднання локальних мереж, так і у випадку доступу до локальних мереж віддалених або мобільних користувачів. Під час проектування VPN зазвичай розглядають дві основні схеми:

1) віртуальний захищений канал між локальними мережами;

2) віртуальний захищений канал між користувачем і локальною мережею.

Схема 1 з'єднання дає змогу замінити дорогі виділені лінії між окремими підрозділами підприємства й створити постійні захищені канали між ними. У цьому випадку шлюз безпеки слугує інтерфейсом між тунелем і локальною мережею, при цьому користувачі локальних мереж використовують тунель для спілкування один з одним. Багато компаній використовують цей вид VPN як заміну або доповнення до наявних з'єднань глобальної мережі.

Схема 2 захищеного каналу VPN призначена для встановлення з'єднань із віддаленими або мобільними користувачами. Створення тунелю ініціює віддалений користувач. Для зв'язку зі шлюзом, що захищає віддалену мережу, він запускає на своєму комп'ютері спеціальне клієнтське програмне забезпечення. Цей вид VPN замінює собою комутовані з'єднання й може бути використаний поряд із традиційними методами віддаленого доступу.

Існують варіанти схем віртуальних захищених каналів. У принципі будь-який із двох вузлів віртуальної приватної мережі, між якими формують віртуальний захищений канал, може належати кінцевій або проміжній точці потоку повідомлень.

З погляду забезпечення інформаційної безпеки, кращим є варіант, за якого кінцеві точки криптозахищеного тунелю збігаються з кінцевими точками потоку повідомлень. У цьому випадку забезпечують захищеність каналу вздовж усього шляху проходження пакетів повідомлень. Однак такий варіант спричиняє децентралізацію управління й надмірність витрат ресурсів. У цьому випадку необхідно встановлювати засоби створення VPN на кожному комп'ютері користувачів локальної мережі. Це ускладнює централізоване управління доступом до комп'ютерних ресурсів і не завжди виправдано економічно. Окреме адміністрування кожного комп'ютера користувачів з метою конфігурування в ньому засобів захисту є доволі трудомісткою процедурою у великій мережі.

Якщо всередині локальної мережі, що входить до віртуальної мережі, не потрібно захищати трафік, тоді як кінцеву точку криптозахищеного тунелю можна вибрати міжмережевий екран або граничний маршрутизатор цієї локальної мережі. Якщо ж потік повідомлень усередині локальної мережі повинен бути захищений, тоді як кінцеву точку тунелю в цій мережі призначають комп'ютер, який бере участь у захищеній взаємодії. У разі доступу до локальної мережі віддаленого користувача комп'ютер цього користувача повинен бути кінцевою точкою віртуального захищеного каналу.

Доволі поширений варіант, коли криптозахищений тунель прокладають тільки всередині відкритої мережі з комутацією пакетів, наприклад, усередині мережі Інтернет. Цей варіант відрізняється зручністю застосування, але має порівняно низьку безпеку. Як кінцеві точки такого тунелю зазвичай використовують обладнання інтернет-провайдерів або граничні маршрутизатори (міжмережеві екрани) локальних мереж.

У разі об'єднання локальних мереж тунель формують лише між обладнанням граничних інтернет-провайдерів або маршрутизаторами (міжмережевим екранами) локальних мереж. За віддаленого доступу до локальної мережі тунель створюють між сервером віддаленого доступу інтернет-провайдера та обладнанням граничного інтернет-провайдера або маршрутизатором (міжмережевим екраном) локальної мережі. Побудовані згідно з таким варіантом віртуальні приватні мережі мають хорошу масштабованість і керованість. Сформовані криптозахищені тунелі повністю прозорі для комп'ютерів користувачів і серверів локальної мережі, що входить до такої віртуальної мережі. Програмне забезпечення цих вузлів залишається без змін. Однак цей варіант характеризується порівняно низькою безпекою інформаційної взаємодії, оскільки частково трафік проходить відкритими каналами зв'язку в незахищеному вигляді. Якщо створює й експлуатує таку VPN інтернет-провайдер, то всю віртуальну приватну мережу можна побудувати на його шлюзах прозоро для локальних мереж і віддалених користувачів підприємства. Але в цьому випадку виникають проблеми довіри до інтернет-провайдера й постійної оплати його послуг.

Криптозахищений тунель створюють із використанням компонентів віртуальної мережі, що функціонують на вузлах, між якими формують тунель. Ці компоненти називають ***ініціатором тунелю*** (tunnel initiator) і ***терминатором тунелю*** (tunnel terminator).

Ініціатор тунелю інкапсулює вихідний пакет у новий пакет, що містить новий заголовок з інформацією про відправника та отримувача. Інкапсульовані пакети можуть належати до протоколу будь-якого типу, зокрема пакети немаршрутизованих протоколів. Усі передані тунелем пакети є пакетами IP.

Маршрут між ініціатором і термінатором тунелю визначає звичайна мережа IP, яка може бути мережею, відмінною від Інтернету.

Ініціювати й розривати тунель можуть різні мережеві пристрої та програмне забезпечення. Наприклад, тунель може бути ініційований ноутбуком мобільного користувача, обладнаним модемом і відповідним програмним забезпеченням для встановлення з'єднань віддаленого доступу. Ініціатором може бути також маршрутизатор локальної мережі, наділений відповідними функціональними можливостями. Тунель зазвичай завершується комутатором мережі або шлюзом інтернет-провайдера.

Термінатор тунелю виконує процес, зворотний інкапсуляції. Термінатор видаляє нові заголовки і скеровує кожен вихідний пакет користувачеві в локальній мережі.

Конфіденційність інкапсульованих пакетів забезпечують їхнім шифруванням, а цілісність і справжність – формуванням електронного цифрового підпису. Існує багато методів і алгоритмів криптографічного захисту даних, тому необхідно, щоб ініціатор і термінатор тунелю своєчасно погодилися використовувати одні й ті самі методи й алгоритми захисту. Для забезпечення можливості розшифрування даних і перевіряння цифрового підпису при прийманні ініціатор і термінатор тунелю повинні також підтримувати функції безпечного обміну ключами. Крім того, кінцеві сторони інформаційної взаємодії повинні пройти автентифікацію, щоб гарантувати створення тунелів VPN тільки між уповноваженими користувачами.

Мережеву інфраструктуру підприємства можна підготувати до використання VPN за допомогою як програмного, так і апаратного забезпечення.

6.7.3. Засоби забезпечення безпеки VPN

Для побудови захищеної віртуальної мережі VPN першочергового значення набуває забезпечення інформаційної безпеки. Згідно із загальноприйнятим визначенням, під безпекою даних розуміють їх конфіденційність, цілісність і доступність. Стосовно до завдань VPN критерії безпеки даних можна визначити так:

– конфіденційність – гарантія того, що в процесі передавання даних захищеними каналами VPN ці дані будуть відомі тільки легальним відправнику й отримувачу;

– цілісність – гарантія збереження переданих даних під час проходження захищеним каналом VPN; будь-які спроби зміни, модифікації, руйнування або створення нових даних будуть виявлені й стануть відомі легальним користувачам;

– доступність – гарантія того, що засоби, які виконують функції VPN, постійно доступні легальним користувачам; Доступність засобів VPN є комплексним показником, який залежить від надійності здійснення, якості обслуговування та ступеня захищеності вмісту від зовнішніх атак.

Конфіденційність забезпечують за допомогою різних методів та алгоритмів симетричного й асиметричного шифрування. Цілісності переданих даних зазвичай досягають за допомогою різних варіантів технології електронного підпису, основаних на асиметричних методах шифрування й односторонніх функціях.

Автентифікацію здійснюють на основі багаторазових та одноразових паролів, цифрових сертифікатів, смарт-карт, протоколів строгої автентифікації, що забезпечує встановлення VPN-з'єднання тільки між легальними користувачами й запобігає доступу до засобів VPN небажаних осіб.

Авторизація передбачає надання користувачам, які довели свою легальність (автентичність), різних видів обслуговування, зокрема різних способів шифрування їхнього трафіку. Авторизацію та управління доступом часто здійснюють одними й тими самими засобами.

Для забезпечення безпеки переданих даних у віртуальних захищених мережах необхідно вирішити такі основні завдання мережевої безпеки:

– взаємна автентифікація користувачів під час встановлення з'єднання;

– забезпечення конфіденційності, цілісності та автентичності переданої інформації;

– авторизація та управління доступом;

– безпека периметра мережі та виявлення вторгнень;

– управління безпекою мережі.

Процедура автентифікації (встановлення автентичності) дає змогу отримувати доступ до мережі легальним користувачам і запобігає доступу до мережі небажаних осіб.

Завдання забезпечення конфіденційності інформації полягає в захисті переданих даних від несанкціонованого читання й копіювання. Основним засобом забезпечення конфіденційності інформації є шифрування.

Ключовим компонентом безпеки VPN є гарантія того, що доступ до комп'ютерних ресурсів отримують авторизовані користувачі, тоді як для неавторизованих користувачів мережа повністю закрита.

Для побудови програмних засобів авторизації застосовують:

– централізовану схему авторизації;

– децентралізовану схему авторизації.

Основне призначення централізованої системи авторизації – забезпечити дотримання принципу єдиного входу. Управляє процесом надання ресурсів

користувачеві сервер. Централізований підхід до процесу авторизації вжито у системах Kerberos, RADIUS і TACACS.

Сьогодні активно розвивають рольове управління доступом. Воно не так вирішує проблеми безпеки, як поліпшує керованість систем. Суть рольового керування доступом полягає в тому, що між користувачами та їхніми привілеями поміщають проміжні сутності – ролі. Для кожного користувача одночасно можуть бути активними кілька ролей, кожна з яких дає йому цілком певні права.

Оскільки ролей набагато менше ніж користувачів і привілеїв, використання ролей сприяє пониженню складності, й, отже, поліпшенню керованості системи. Крім того, на підставі рольової моделі управління доступом можна забезпечити дотримання такого важливого принципу, як поділ обов'язків (наприклад, неможливість поодинці зупинити критично важливий процес).

Жорсткий контроль доступу до додатків, послуг (сервісів) та ресурсів мережі, яку захищають, є важливою функцією правильно побудованої мережі. Використання таких засобів безпеки, як міжмережеві екрани, системи виявлення вторгнень, системи аудиту безпеки, антивірусні комплекси забезпечує системний захист переданих мережею даних. Важливою частиною загального рішення безпеки мережі є міжмережеві екрани, які контролюють трафік, що перетинає периметр захищеної мережі, і накладають обмеження на пропуск трафіку відповідно з політикою безпеки підприємства чи організації.

Додатковим елементом гарантії безпеки периметра мережі є ***система виявлення вторгнень*** (Intrusion Detection System – IDS), що працює в реальному часі й призначена для виявлення, фіксації та припинення неавторизованої мережевої активності як від зовнішніх, так і від внутрішніх джерел.

Системи аналізу захищеності сканують корпоративну мережу з метою виявлення потенційних вразливостей безпеки, даючи можливість менеджерам мережі краще захистити мережу від атак.

Мережі VPN об'єднують і мережеві пристрої, і численні послуги управління безпекою та пропускною здатністю. Компаніям необхідно цілісне управління цими пристроями та послугами через інфраструктуру VPN, включно з користувачами віддаленого доступу й засобами ***міжкорпоративних мереж*** (extranet). У зв'язку із цим управління засобами VPN стає одним із найважливіших завдань забезпечення ефективного функціонування VPN. Система управління приватною мережею повинна містити необхідний набір засобів для управління політиками безпеки, пристроями та службами VPN будь-якого масштабу.

6.7.4. Класифікація мереж VPN

Завдяки технології VPN багато компаній починають будувати свою стратегію з урахуванням використання Інтернету як головного засобу передавання інформації, причому навіть тієї, яка є вразливою або життєво важливою.

Існують різні ознаки класифікації VPN. Найчастіше використовують:

- рівень моделі OSI, на якому працює VPN;
- архітектура технічного рішення VPN;
- спосіб технічного здійснення VPN.

Для технологій безпечного передавання даних відкритою (загальнодоступною, незахищеною) мережею застосовують узагальнену назву – ***захищений канал*** (secure channel). Термін "канал" підкреслює той факт, що захист даних забезпечують між двома вузлами мережі (вузлами або шлюзами) уздовж деякого віртуального шляху, прокладеного в мережі з комутацією пакетів.

Захищений канал можна побудувати за допомогою системних засобів, здійснених на різних рівнях моделі OSI.

Класифікація VPN за рівнем моделі OSI, на якому працює VPN. Стандарти описують побудову VPN на різних рівнях моделі OSI. Що нижчий рівень, на якому здійснюють захист, то VPN прозоріша для користувачів. Проте на нижчих рівнях зменшується набір послуг безпеки і стає складнішим організовування управління. На вищих рівнях є ширший набір послуг безпеки, надійніший контроль доступу й простіше конфігурування системи захисту. Формування криптозахищених тунелів одночасно на декількох рівнях збільшує криптостійкість віртуальної мережі, але внаслідок зниження загальної швидкості криптографічних перетворень зменшується її пропускна здатність. Для незалежності від прикладних протоколів і програмних додатків віртуальні мережі часто будують на одному з нижніх рівнів.

За рівнем моделі OSI, на якому працює VPN, розрізняють:

– VPN канального рівня;
– VPN мережевого рівня;
– VPN сеансового рівня.

Засоби VPN, що використовують на канальному рівні моделі OSI, забезпечують захист кадрів (фреймів, пакетів), сформованих згідно з певною мережевою технологією, що застосована для побудови конкретної телекомунікаційної мережі. Такі засоби дають змогу забезпечити інкапсуляцію різних видів трафіку третього рівня (і вище) і побудову віртуальних криптозахищених тунелів типу "точка-точка" (від маршрутизатора до маршрутизатора або від персонального комп'ютера до шлюзу локальної мережі). Для формування криптозахищених тунелів на канальному рівні компанія Microsoft за підтримки

компаній Ascend Communications, 3Com/Primary Access, ECI-Telematics і US Robotics розробила протокол тунелювання ***PPTP*** (Point-to-Point Tunneling Protocol), який є розширенням протоколу ***PPP*** (Point-to-Point Protocol). Канальному рівню відповідає також протокол тунелювання ***L2F*** (Layer-2 Forwarding), розроблений компанією Cisco Systems за підтримки компаній Shiva і Northern Telecom.

Протоколи PPTP і L2F у 1996 році об'єднано в один протокол, який увібрав від них усе краще й був названий ***L2TP*** (Layer-2 Tunneling Protocol). Цей протокол підтримують компанії Cisco, Microsoft, 3Com, Ascend і багато інших виробників. Крім інформаційних кадрів, протокол L2TP для налагодження з'єднання використовує керівні кадри. Як і попередні протоколи канального рівня, специфікація L2TP не регламентує використання методів автентифікації й шифрування.

Найбільш оптимальними для захисту трафіку є протоколи, які працюють на мережевому рівні й не зорієнтовані на специфіку додатків прикладного рівня чи технологію мережі. Програмні засоби VPN мережевого рівня виконують інкапсуляцію IP пакетів в IP пакети. Одним із поширених на цьому рівні є стек протоколів ***IPSec*** (IP Security), призначений для автентифікації, тунелювання й шифрування IP-пакетів. Ядром IPSec є три протоколи:

– AH (Authentication Header – заголовок автентифікації) – забезпечує автентифікацію користувачів і цілісність даних;

– ESP (Encapsulating Security Payload – інкапсуляція зашифрованих даних) – шифрує дані й забезпечує їх конфіденційність;

– IKE (Internet Key Exchange – обмін ключами Інтернет) – надає протоколам AH і ESP секретні ключі, необхідні для автентифікації й шифрування даних.

IPSec використовує стандартні методи автентифікації користувачів при ініціалізації тунелю, стандартні способи шифрування даних та формування й перевірки цифрового підпису вузлами. Для управління криптографічними ключами IPSec використовують, переважно, протоколи ***SKIP*** (Simple Key management for Internet Protocols) і ***ISAKMP*** (Internet Security Association and Key Management Protocol). SKIP простіший при здійсненні, але не підтримує переговорів із вибору алгоритмів шифрування.

Тунель IPSec між двома локальними мережами може підтримувати багато індивідуальних каналів передавання даних, унаслідок чого має переваги порівняно з технологіями другого рівня. Для розширення функцій системи IPSec і зміни параметрів захисту ядро можна доповнювати іншими додатковими протоколами. Систему IPSec можна використовувати спільно із протоколом L2TP, що забезпечує вищий рівень гнучкості при захисті віртуальних каналів.

Деякі VPN використовують ***посередники каналів*** (circuit proxy). Цей метод функціонує над транспортним рівнем і ретранслює трафік із захищеної мережі в загальнодоступну мережу Інтернет для кожного сокета окремо (сокет IP ідентифікують комбінацією TCP-з'єднання й конкретного порту або заданим портом ***UDP*** (User Datagram Protocol). Стек TCP / IP не має п'ятого – сеансового – рівня, однак орієнтовані на сокети операції часто називають операціями сеансового рівня).

Серед посередників каналів часто використовують протоколи компанії Netscape Communications ***SSL*** (Secure Sockets Layer – шар захищених сокетів) і ***TLS*** (Transport Layer Security – безпека на транспортному рівні). Ці протоколи створюють криптозахищені тунелі між кінцевими точками віртуальної мережі, забезпечуючи взаємну автентифікацію користувачів, конфіденційність, достовірність і цілісність даних. Ядром названих протоколів є технологія комплексного використання асиметричних і симетричних криптосистем компанії RSA Data Security. Для автентифікації взаємодіючих сторін і криптозахисту ключів симетричного шифрування використовують цифрові сертифікати відкритих ключів клієнта й сервера, які завірені цифровими підписами спеціальних сертифікаційних центрів. Підтримують цифрові сертифікати відповідно до загальноприйнятого стандарту X.509. Вибір алгоритму шифрування залежить від багатьох чинників, але найчастіше використовують 3DES та AES. Протокол SSL підтримують усі популярні браузери.

Для стандартизації автентифікованого проходу через міжмережевий екран консорціум ***IETF*** (Internet Engineering Task Force) визначив протокол за назвою ***SOCKS*** (Socket Secure – безпека сокета), і сьогодні протокол SOCKS v.5 застосовують для стандартизованого формування посередників каналів.

Протоколи захищених каналів, які працюють на прикладному рівні, не залежать від технології транспортування даних, а зорієнтовані на захист додатків, створених конкретними протоколами прикладного рівня. Оскільки такі протоколи повністю залежать від додатків користувачів, їх рідко використовують для побудови захищених каналів віртуальних мереж. Проте формування захищених віртуальних каналів на нижніх рівнях є прозорим для протоколів прикладного рівня, а їх сумісне використання із протоколами захисту прикладного рівня підвищує рівень безпеки мережі.

Класифікація VPN за архітектурою технічного рішення. За архітектурою технічного рішення розрізняють три основні види віртуальних приватних мереж:

– внутрішньокорпоративні VPN (Intranet VPN);
– VPN із віддаленим доступом (Remote Access VPN);
– міжкорпоративні VPN (Extranet VPN).

Внутрішньокорпоративні мережі VPN призначені для забезпечення захищеної взаємодії між підрозділами всередині підприємства або між групою підприємств, об'єднаних корпоративними телекомунікаційними мережами, зокрема виділеними лініями.

VPN із віддаленим доступом призначені для забезпечення захищеного віддаленого доступу до корпоративних інформаційних ресурсів для мобільних і / або віддалених (home-office) співробітників корпорації (підприємства).

Міжкорпоративні мережі VPN призначені для забезпечення захищеного обміну інформацією зі стратегічними партнерами з бізнесу, постачальниками, великими замовниками, користувачами, клієнтами тощо. Такі мережі забезпечують прямий доступ з мережі одного підприємства до мережі іншого підприємства, сприяючи підвищенню надійності зв'язку під час ділового співробітництва.

Слід зазначити, що сьогодні спостерігають тенденцію до об'єднання різних конфігурацій VPN.

Класифікація VPN за способом технічного здійснення. Конфігурація та характеристики віртуальної приватної мережі багато в чому залежать від типу застосовуваних VPN-пристроїв. За способом технічного здійснення розрізняють VPN на основі:

- маршрутизаторів;
- міжмережевих екранів;
- програмних рішень;
- спеціалізованих апаратних засобів із вбудованими шифропроцесорами.

VPN на основі маршрутизаторів будують із використанням маршрутизаторів із функцією шифрування. Оскільки вся інформація, що виходить із локальної мережі, проходить через маршрутизатор, то цілком природно покласти на нього й завдання шифрування. Приклад обладнання для VPN – маршрутизатор компанії Cisco Systems (рис. 6.15).

VPN на основі міжмережевих екранів із функцією тунелювання й шифрування, що підтримують, наприклад, програмне забезпечення Fire Wall-1 компанії Check Point Software Technologies, використовують переважно для невеликих мереж із невеликим обсягом переданої інформації, якщо як міжмережеві екрани застосовують персональні комп'ютери. Недоліками цього методу є висока вартість рішення в перерахунку на одне робоче місце й залежність продуктивності від апаратного забезпечення, на якому працює міжмережевий екран.

VPN на основі програмного забезпечення з погляду продуктивності поступаються спеціалізованим пристроям, однак володіють достатньою потужністю для здійснення VPN-мереж. Слід зазначити, що у випадку

віддаленого доступу вимоги до необхідної смуги пропускання невеликі. Тому суто програмні продукти легко забезпечують продуктивність, достатню для віддаленого доступу. Безсумнівною перевагою програмних продуктів є гнучкість і зручність у застосуванні, а також відносно невисока вартість.

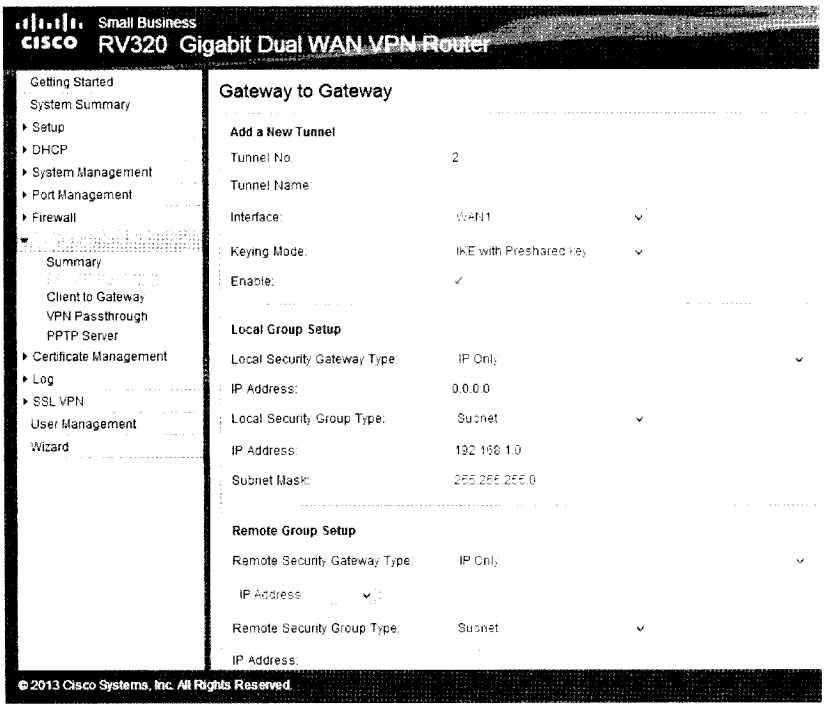


Рис. 6.15. Інтерфейс маршрутизатора Cisco RV320 Gigabit Dual WAN VPN Router

VPN на основі спеціалізованих апаратних засобів мають перевагу – високу продуктивність. Це зумовлено тим, що шифрування в них здійснюють за допомогою спеціалізованих мікросхем. Спеціалізовані VPN-пристрої забезпечують високий рівень безпеки, однак вони дорогі.

6.8. Протоколи захисту інформації на канальному рівні моделі OSI

Протоколи PPTP, L2F і L2TP – це протоколи тунелювання канального рівня моделі OSI. Загальною властивістю цих протоколів є те, що їх використовують для організації захищеного багатопротокольного віддаленого доступу до ресурсів приватної мережі через відкриту мережу, наприклад, через Інтернет [9–22].

Усі три протоколи – PPTP, L2F і L2TP – зазвичай зараховують до протоколів формування захищеного каналу, однак цьому визначенню точно відповідає лише протокол PPTP, який забезпечує тунелювання й шифрування переданих даних. Протоколи L2F і L2TP підтримують тільки функції тунелювання. Для захисту даних у цих протоколах необхідно використовувати деякий додатковий протокол, зокрема IPSec.

У клієнтському програмному забезпеченні зазвичай використовують для віддаленого доступу стандартний протокол канального рівня PPP. Протоколи PPTP, L2F і L2TP ґрунтуються на протоколі PPP і є його розширеннями. Спочатку протокол PPP, розташований на канальному рівні, було розроблено для інкапсуляції даних та їх доставлення при встановленні з'єднання типу "точка-точка". Цей протокол слугує також для організації асинхронних (наприклад, комутованих) з'єднань.

У набір PPP входять ***протокол управління з'єднанням*** LCP (Link Control Protocol), відповідальний за конфігурацію, установлення, роботу й завершення з'єднання "точка-точка", а також ***протокол управління мережею*** (Network Control Protocol – NCP), здатний інкапсулювати в PPP протоколи мережевого рівня для транспортування при встановленні з'єднання "точка-точка". Це дає змогу одночасно передавати пакети Novell IPX і Microsoft IP у разі використання одного встановленого з'єднання PPP.

Для доставлення конфіденційних даних з однієї точки в іншу через мережі загального користування спочатку здійснюють інкапсуляцію даних за допомогою протоколу PPP, потім протоколи PPTP і L2TP виконують шифрування даних і власну інкапсуляцію. Після того, як тунельний протокол доставляє пакети з початкової точки тунелю в кінцеву, виконують деінкапсуляцію.

На фізичному й канальному рівнях протоколи PPTP і L2TP однакові, але на цьому їх схожість закінчується, і починаються відмінності.

6.8.1. Принцип роботи протоколу PPTP

Протокол PPTP розроблений компанією Microsoft за підтримки інших компаній і призначений для створення захищених віртуальних каналів при доступі віддалених користувачів до локальних мереж через Інтернет. Він передбачає створення криптозахищеного тунелю на канальному рівні моделі OSI як для випадку прямого з'єднання віддаленого комп'ютера з відкритою мережею, так і для випадку під'єднання його до відкритої мережі телефонною лінією через обладнання провайдера телекомунікацій.

Протокол PPTP отримав практичне розповсюдження завдяки компанії Microsoft, що впровадила його у свої операційні системи Windows. Деякі вироб-

ники міжмережевих екранів і шлюзів VPN також підтримують цей протокол. Протокол PPTP дає можливість створювати захищені канали для обміну даними з використанням протоколів IP, IPX або NetBEUI. Дані цих протоколів запаковують у кадри PPP і потім інкапсулюють за допомогою протоколу PPTP у пакети протоколу IP, за допомогою якого переносять у зашифрованому вигляді через мережу TCP / IP.

Пакети, передані в межах PPTP з'єднання, мають таку структуру:

– заголовок канального рівня, що використовують у мережі Інтернет, наприклад, заголовок кадру Ethernet;

– заголовок IP, що містить адреси відправника й отримувача пакета;

– заголовок ***протоколу загальної інкапсуляції маршрутів*** (Generic Routing Encapsulation – GRE);

– вихідний пакет PPP, що містить пакет IP, IPX або NetBEUI.

Приймальний вузол мережі виділяє з пакетів IP кадри PPP, а потім виділяє з кадру PPP вихідний пакет IP, IPX або NetBEUI і відправляє його локальною мережею конкретного адресата. Багатопротокольність інкапсулюючих протоколів канального рівня, до яких належить протокол PPTP, є їхньою важливою перевагою перед протоколами захищеного каналу більш високих рівнів. Наприклад, якщо в корпоративній мережі використовують IPX або NetBEUI, застосування протоколів IPSec або SSL просто неможливе, оскільки вони орієнтовані тільки на один протокол мережевого рівня IP.

Такий спосіб інкапсуляції забезпечує незалежність від протоколів мережевого рівня моделі OSI й дає змогу здійснювати захищений віддалений доступ через відкриті IP-мережі до будь-яких локальних мереж (IP, IPX або NetBEUI). Згідно із протоколом PPTP при створенні захищеного віртуального каналу виконують автентифікацію віддаленого користувача й шифрування переданих даних (рис. 6.16).

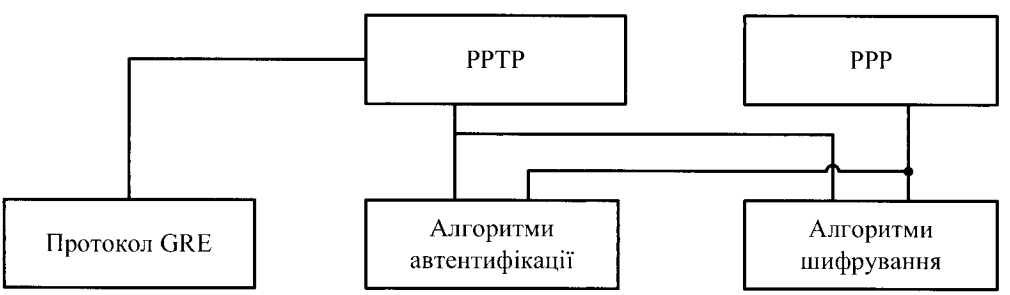


Рис. 6.16. Компоненти протоколу PPTP

Для автентифікації віддаленого користувача можна використовувати різні протоколи, що застосовують для PPP. У виконанні протоколу PPTP, що входить

до складу операційної системи Microsoft Windows, підтримують такі протоколи автентифікації: ***протокол розпізнавання за паролем*** (Password Authentication Protocol – PAP), ***протокол розпізнавання при рукостисканні*** (Microsoft Challenge-Handshaking Authentication Protocol – MSCHAP) і ***розширюваний протокол розпізнавання*** (Extensible Authentication Protocol – Transport Layer Security – EAP-TLS). З використанням протоколу PAP ідентифікатори й паролі передають лінією зв'язку в незашифрованому вигляді, при цьому лише сервер здійснює автентифікацію клієнта (комп'ютера користувача). При використанні протоколів MSCHAP і EAP-TLS забезпечують захист від повторного використання зловмисником перехоплених пакетів із зашифрованим паролем, а також взаємну автентифікацію клієнта й VPN-сервера.

Шифрування за допомогою PPTP гарантує, що ніхто не зможе отримати доступ до даних під час їх передавання через Інтернет. ***Протокол шифрування точка-точка*** (Microsoft Point-to-Point Encryption – MPPE) сумісний тільки з MSCHAP (версії 1 і 2) і EAP-TLS, а також забезпечує автоматичний вибір довжини ключа шифрування при узгодженні параметрів між клієнтом і сервером. Протокол MPPE підтримує роботу із ключами завдовжки 40, 56 або 128 бітів. Згідно із протоколом PPTP змінюють значення ключа шифрування після кожного прийнятого пакета.

Для протоколу PPTP визначено дві основні схеми застосування:

1) схема тунелювання за прямого з'єднання комп'ютера віддаленого користувача з Інтернетом;

2) схема тунелювання у разі підключення комп'ютера віддаленого користувача до Інтернету телефонною лінією через обладнання провайдера телекомунікацій.

Розглянемо першу схему тунелювання. Віддалений користувач установлює віддалене з'єднання з локальною мережею за допомогою клієнтської частини ***служби віддаленого доступу*** (Remote Access Service – RAS), що входить до складу Windows. Потім користувач звертається до сервера віддаленого доступу локальної мережі, зазначаючи його IP-адресу, і встановлює з ним зв'язок за протоколом PPTP (рис. 6.17).

Функції сервера віддаленого доступу може виконувати граничний маршрутизатор локальної мережі. На комп'ютері віддаленого користувача повинні бути встановлені клієнтська частина служби RAS і драйвер PPTP, які входять до складу Windows, а на сервері віддаленого доступу локальної мережі – сервер RAS і драйвер PPTP, що входять до складу Windows Server. Протокол PPTP визначає формат кількох службових повідомлень, якими обмінюються взаємодіючі сторони. Службові повідомлення передають за протоколом TCP.

Після успішної автентифікації починають процес захищеного інформаційного обміну. Внутрішні сервери локальної мережі можуть не підтримувати протокол PPTP, оскільки граничний маршрутизатор виділяє кадри PPP із пакетів IP і передає їх локальною мережею в необхідному форматі – IP, IPX або NetBIOS.

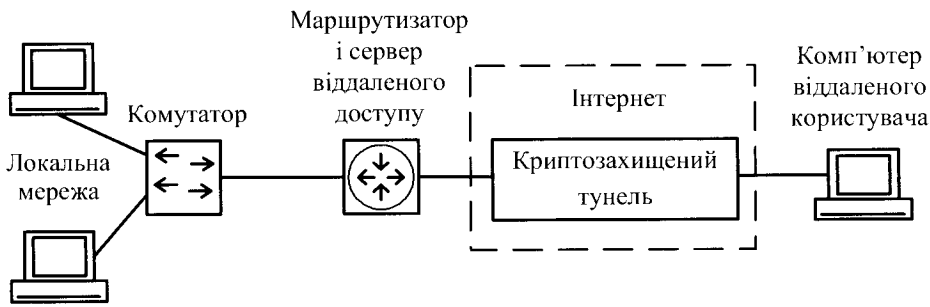


Рис. 6.17. Схема тунелювання за прямого з'єднання комп'ютера віддаленого користувача з Інтернетом

Друга схема тунелювання не отримала широкого розповсюдження.

6.8.2. Протоколи L2F і L2TP

Протокол L2F розробила компанія Cisco Systems для захисту віртуальних мереж на канальному рівні моделі OSI як альтернатива протоколу PPTP.

Проте сьогодні його фактично поглинув протокол L2TP, тому нижче розглядатимемо основні можливості та властивості протоколу L2TP.

Протокол L2TP розроблено в організації IETF за підтримки компаній Microsoft і Cisco Systems. Протокол L2TP розробляли як протокол захищеного тунелювання PPP-трафіку через мережі загального призначення з довільним середовищем. Роботу над цим протоколом вели на основі протоколів PPTP і L2F, і в результаті він увібрав у себе кращі якості вихідних протоколів.

На відміну від PPTP, протокол L2TP не прив'язаний до протоколу IP, тому його можна використовувати у мережах із комутацією пакетів, наприклад, у мережах ***ATM*** (Asynchronous Transfer Mode – асинхронний режим передавання даних) або в мережах із ***ретрансляцією кадрів*** (frame relay). Крім того, до протоколу L2TP додано важливу функцію управління потоками даних, а також відсутні у специфікації протоколу PPTP функції захисту, зокрема передбачено можливість роботи із протоколом AH і протоколом ESP стека протоколів IPSec.

По суті, гібридний протокол L2TP є розширенням протоколу PPP функціями автентифікації віддалених користувачів, створення захищеного віртуального з'єднання й управління потоками даних.

Протокол L2TP застосовує як транспорт протокол UDP і використовує однаковий формат повідомлень і для управління тунелем, і для пересилання даних.

Хоча протокол PPTP забезпечує достатній ступінь безпеки, але все ж протокол L2TP (поверх IPSec) надійніший. Протокол L2TP (поверх IPSec) забезпечує автентифікацію на рівнях "користувач" і "комп'ютер", а також виконує автентифікацію й шифрування даних.

Після того, як L2TP (поверх IPSec) завершить процес автентифікації комп'ютера, виконують автентифікацію на рівні користувача.

На відміну від своїх попередників – протоколів PPTP і L2F, – протокол L2TP надає можливість відкривати між кінцевими користувачами відразу декілька тунелів, кожен з яких може бути виділений для окремого додатка. Ці особливості забезпечують гнучкість і безпеку тунелювання.

Згідно зі специфікацією протоколу L2TP як сервер віддаленого доступу інтернет-провайдера використовують ***концентратор доступу L2TP*** (L2TP Access Concentrator – LAC), який забезпечує віддаленому користувачеві мережевий доступ до його локальної мережі через Інтернет. Як сервер віддаленого доступу локальної мережі використовують ***мережевий сервер L2TP*** (L2TP Network Server), що функціонує на сумісних із протоколом PPP платформах (рис. 6.18).

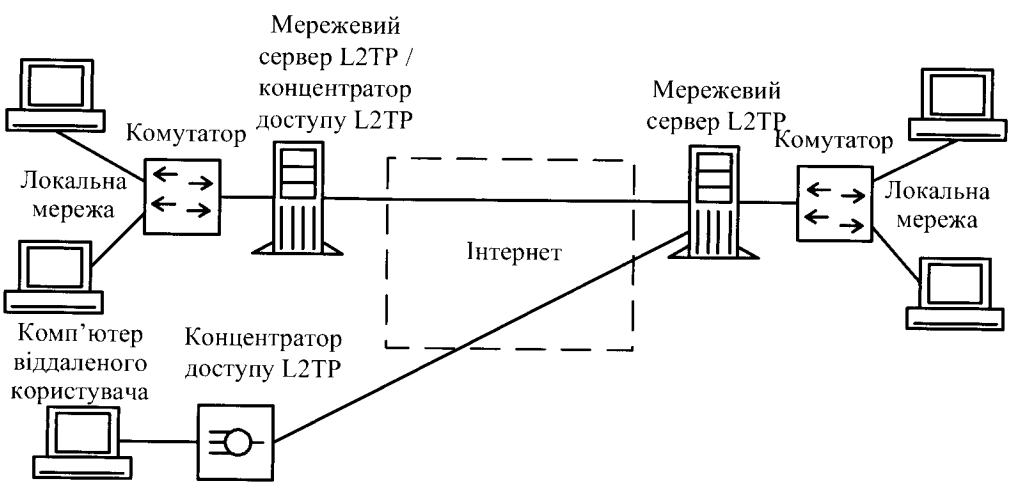


Рис. 6.18. Схема із захищеним віртуальним каналом у протоколі L2TP

Формують захищений віртуальний канал за протоколом L2TP у три етапи:

• установлення з'єднання із сервером віддаленого доступу локальної мережі;

• автентифікація користувача;

• конфігурування криптозахищеного тунелю.

Протокол L2TP не визначає конкретних методів криптозахисту й передбачає можливість застосування різних стандартів шифрування. Якщо криптозахищений тунель планують сформувати в IP-мережах, то для криптозахисту використовують протокол IPSec. Протокол L2TP поверх IPSec забезпечує вищий ступінь захисту даних, ніж PPTP, оскільки використовує алгоритм шифрування 3DES або AES. Якщо такого високого рівня захисту не потрібно, можна використовувати алгоритм DES з одним 56-розрядним ключем. Крім того, за допомогою алгоритму ***HMAC*** (Hash Message Authentication Code – хеш-код автентифікації повідомлень) протокол L2TP забезпечує автентифікацію даних, для чого цей алгоритм створює хеш (бітовий рядок) завдовжки 128 розрядів.

Отже, функційні можливості протоколів PPTP і L2TP різні. Протокол PPTP можна застосовувати лише в IP-мережах. Протокол L2TP можна використовувати не тільки в IP-мережах. Протокол L2TP поверх IPSec пропонує більше рівнів безпеки, ніж PPTP, і може гарантувати майже стовідсоткову безпеку важливих даних.

Однак за всіх своїх переваг протокол L2TP не зміг подолати низку недоліків тунельного передавання даних на канальному рівні:

• для виконання протоколу L2TP необхідна підтримка інтернет-провайдера;

• протокол L2TP обмежує трафік в обраному тунелі й позбавляє користувачів доступу до інших частин Інтернету;

• специфікація L2TP забезпечує стандартне шифрування тільки в IP-мережах за допомогою протоколу IPSec.

6.9. Протоколи захисту даних на мережевому рівні моделі OSI

Радикальне усунення вразливостей комп'ютерних мереж можливе у разі створення системи захисту не для окремих класів додатків, а для всієї мережі. Стосовно IP-мереж це означає, що системи захисту повинні діяти на мережевому рівні моделі OSI [9–22]. Перевагою такого вибору є те, що в IP-мережах саме мережевий рівень відрізняється найбільшою незалежністю від верхніх протоколів, фізичного середовища передавання й технології канального рівня. Транспортування даних мережею не можна здійснити в обхід мережевого протоколу IP. Тому захист мережі на третьому рівні автоматично гарантує як мінімум такий самий ступінь захисту всіх мережевих додатків, причому без будь-якої модифікації останніх.

У разі формування захищених віртуальних каналів на мережевому рівні моделі OSI досягають оптимального співвідношення між прозорістю та якістю захисту. Розміщення засобів захисту на мережевому рівні робить їх прозорими для додатків, оскільки між мережевим рівнем і додатком працює протокол транспортного рівня. Для користувачів процедури захисту виявляються настільки ж прозорими, як і сам протокол IP. На мережевому рівні існує можливість доволі повного здійснення функцій захисту трафіку та управління ключами, оскільки саме на мережевому рівні виконують маршрутизацію пакетів повідомлень.

Стек протоколів IPSec використовують для автентифікації учасників обміну, тунелювання трафіку й шифрування IP-пакетів. Основне призначення протоколу ***IPSec*** (Internet Protocol Security) – безпечне передавання даних мережами IP. Оскільки архітектура IPSec сумісна із протоколом IPv4, її підтримку достатньо забезпечити на обох кінцях з'єднання; проміжні мережеві вузли не потребують додаткових налаштувань IPSec. Протокол IPSec може захищати трафік як поточної версії протоколу IPv4, уживаної сьогодні в Інтернеті, так і трафік нової версії IPv6, яку поступово впроваджують в Інтернет.

6.9.1. Компоненти IPSec

Основне призначення компонентів IPSec – захищати інформацію під час її передавання мережами IP. Застосування IPSec гарантує:

– цілісність переданих даних (тобто дані при передаванні не спотворені, не втрачені й не продубльовані);

– автентичність відправника (тобто дані передані саме тим відправником, який довів, що він той, за кого себе видає);

– конфіденційність переданих даних (тобто дані передають у формі, що запобігає їх несанкціонованому перегляду).

Зазвичай до поняття безпеки даних входить ще одна вимога – доступність даних, що в розглянутому контексті можна інтерпретувати як гарантію їх доставлення. Протоколи стека IPSec не вирішують цього завдання, залишаючи її протоколу транспортного рівня TCP. Стек протоколів IPSec забезпечує захист інформації на мережевому рівні, що робить цей захист невидимим для працюючих додатків.

Фундаментальною одиницею комунікації в IP-мережах є IP-пакет. IP-пакет містить адресу відправника й адресу отримувача повідомлення, транспортний заголовок, інформацію про тип даних, які переносять у цьому пакеті, і самі дані.

Користувач сприймає мережу як надійно захищене середовище тільки в тому випадку, якщо він упевнений, що його партнер за обміном – саме той, за кого він себе видає (автентифікація сторін), що передані пакети не проглядають сторонні особи (конфіденційність зв'язку) і що отримувані дані не змінено в процесі передавання (цілісність даних).

Для того, щоб забезпечити автентифікацію, конфіденційність і цілісність переданих даних, стек протоколів IPSec побудований за стандартизованими криптографічними технологіями:

– обміну ключами згідно з алгоритмом Діффі – Хеллмана для розподілу секретних ключів між користувачами у відкритій мережі;

– криптографії відкритих ключів для підписування обмінів Діффі – Хеллмана, щоб гарантувати справжність двох сторін і уникнути атак типу "людина посередині ("man-in-the-middle");

– цифрових сертифікатів для підтвердження автентичності відкритих ключів;

– блокових симетричних алгоритмів шифрування даних;

– алгоритмів автентифікації повідомлень на основі функцій хешування.

Протокол IPSec задає стандартні способи захисту інформаційного обміну на мережевому рівні моделі OSI для IP-мережі, що є основним видом мереж. Цей протокол входить до складу нової версії протоколу IP (IPv6) і до його широко вживаної версії (IPv4). Для протоколу IPv4 підтримка IPSec є бажаною, а для IPv6 – обов'язковою. Протокол IPSec є системою відкритих стандартів, яка має чітко окреслене ядро, і водночас дає змогу доповнювати її новими протоколами, алгоритмами та функціями. Стандартизованими функціями IPSec-захисту можуть користуватися протоколи високих рівнів, зокрема, керуючі протоколи, протоколи конфігурування, а також протоколи маршрутизації.

Основними завданнями встановлення й підтримки захищеного каналу є такі:

– автентифікація користувачів або комп'ютерів при запиті на встановлення захищеного каналу;

– шифрування й автентифікація переданих даних між кінцевими точками захищеного каналу;

– забезпечення кінцевих точок каналу секретними ключами, необхідними для роботи протоколів автентифікації й шифрування даних. Для вирішення перерахованих завдань система IPSec використовує комплекс засобів безпеки інформаційного обміну.

Більшість виконань протоколу IPSec мають такі компоненти.

Основний протокол IPSec. Цей компонент виконує протоколи ESP і AH. Він обробляє заголовки, взаємодіє з базами даних SPD і SAD для визначення політики безпеки, застосовуваної до пакета.

Протокол обміну ключами Інтернету (Internet Key Exchange – IKE). Він надає протоколам AH і ESP секретні ключі, необхідні для автентифікації й шифрування даних.

База даних політик безпеки (Security Policy Database – SPD). Це один із найважливіших компонентів, оскільки він визначає політику безпеки, застосовувану до пакета даних. SPD використовує основний протокол IPSec для оброблення вхідних і вихідних пакетів.

Безпечна асоціація (Security Association) ***SA*** – це симплексне (одностороннє) логічне з'єднання, яке підтримує й забезпечує безпечне передавання даних між мережевими пристроями. Створення SA описано в RFC 2408.

База даних безпечних асоціацій (Security Association Database – SAD). База даних SAD зберігає список безпечних асоціацій для оброблення вхідної та вихідної інформації. Вихідні SA використовують для захисту вихідних пакетів, а вхідні SA використовують для оброблення пакетів із заголовками IPSec. SA заповнює SAD самостійно або за допомогою протоколу IKE.

Керування політикою безпеки й безпечними асоціаціями SA. Це – додатки, які керують політикою безпеки й SA.

Основний протокол IPSec (виконує ESP і AH) тісно взаємодіє із транспортним і мережевим рівнями стека протоколів TCP / IP. Фактично протокол IPSec є частиною мережевого рівня. Основний модуль протоколу IPSec забезпечує два інтерфейси: вхідний і вихідний. Вхідний інтерфейс використовують вхідні пакети, а вихідний – вихідні. Здійснення IPSec не повинне залежати від інтерфейсу між транспортним і мережевим рівнем стека протоколів TCP / IP.

SPD і SAD суттєво впливають на ефективність роботи IPSec. Вибір структури даних для зберігання SPD і SAD є критичним моментом, від якого залежить продуктивність IPSec. Особливості виконання SPD і SAD залежать від вимог продуктивності та сумісності системи.

Усі протоколи, що входять до IPSec, можна поділити на дві групи:

1) протоколи, що безпосередньо обробляють передані дані (для забезпечення їх захисту);

2) протоколи, що дають змогу автоматично узгодити параметри захищених з'єднань, необхідні для протоколів першої групи.

Архітектуру засобів безпеки IPSec зображено на рис. 6.19.

На верхньому рівні розташовано 3 протоколи, що складають ядро IPSec:

– протокол IKE, що визначає спосіб установлення захищеного каналу, зокрема узгодження використовуваних алгоритмів криптозахисту, а також процедури обміну та керування секретними ключами в межах захищеного з'єднання;

– протокол AH, що забезпечує автентифікацію джерела даних, перевірку їх цілісності та автентичності після приймання, а також захист від нав'язування повторних повідомлень;

– протокол ESP, що забезпечує криптографічне закриття, автентифікацію та цілісність переданих даних, а також захист від нав'язування повторних повідомлень.

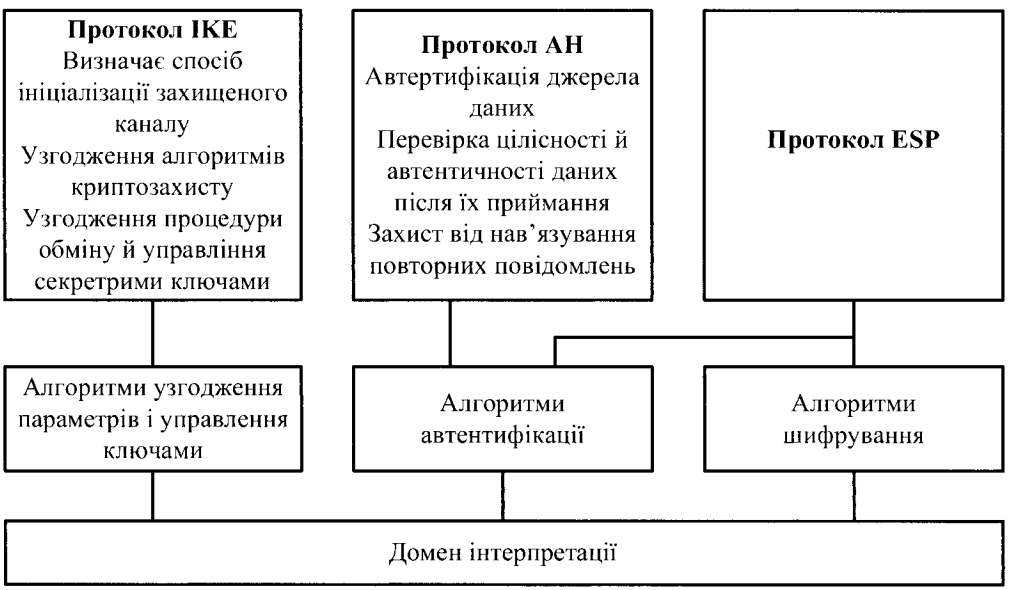


Рис. 6.19. Архітектура засобів безпеки IPSec

Поділ функцій захисту між двома протоколами AH і ESP зумовлений застосовуваною в багатьох країнах практикою обмеження експорту та / або імпорту засобів, що забезпечують конфіденційність даних шифруванням. Кожен із протоколів AH і ESP можна використовувати як самостійно, так і спільно з іншим. З короткого переліку функцій протоколів AH і ESP видно, що можливості цих протоколів частково перекриваються.

Протокол AH відповідає тільки за забезпечення цілісності й автентифікації даних, тоді як протокол ESP є потужнішим, оскільки може шифрувати дані, а також виконувати функції протоколу AH (хоча, як побачимо пізніше, він забезпечує автентифікацію й цілісність у дещо обмеженому вигляді).

Протокол ESP може підтримувати функції шифрування й автентифікації / цілісності в будь-яких комбінаціях, тобто або обидві групи функцій, або тільки автентифікацію / цілісність, або тільки шифрування.

На середньому рівні архітектури IPSec знаходяться алгоритми узгодження параметрів і управління ключами, застосовувані в протоколі IKE, а також алгоритми автентифікації й шифрування, які використовують у протоколах AH та ESP.

Протоколи захисту віртуального каналу верхнього рівня архітектури IPSec (AH і ESP) не залежать від конкретних криптографічних алгоритмів. За можливості використання великої кількості різноманітних алгоритмів автентифікації й шифрування IPSec забезпечує високий ступінь гнучкості організації захисту мережі. Гнучкість IPSec полягає в тому, що для кожного завдання він пропонує кілька способів його вирішення. Обрані методи для одного завдання зазвичай не залежать від методів здійснення інших завдань. Наприклад, вибір для шифрування алгоритму AES не впливає на вибір функції обчислення хешу, використовуваного для автентифікації даних.

На нижньому рівні архітектури IPSec знаходиться ***домен інтерпретації*** (Domain of Interpretation – DOI). Необхідність застосування DOI обумовлена наведеними нижче причинами. Протоколи AH і ESP мають модульну структуру, допускаючи застосування користувачами за їх узгодженим вибором різних криптографічних алгоритмів шифрування й автентифікації. Тому необхідний модуль, який міг би забезпечити спільну роботу всіх протоколів і алгоритмів. Саме такі функції покладено на домен інтерпретації. Домен інтерпретації як база даних зберігає відомості про використовувані в IPSec протоколи й алгоритми, їх параметри, протокольні ідентифікатори тощо. По суті, він виконує роль фундаменту в архітектурі IPSec. Для того, щоб використовувати алгоритми, які відповідають національним стандартам як алгоритми автентифікації й шифрування в протоколах AH і ESP, необхідно зареєструвати ці алгоритми в DOI.

6.9.2. Режими та функції протоколів AH і ESP

Протокол AH і протокол ESP можуть працювати в тунельному або транспортному режимах. Для виконання своїх завдань щодо безпечного передавання даних протоколи AH і ESP вводять в оброблювані ними пакети додаткову службову інформацію, оформляючи її у вигляді заголовків.

Протокол AH забезпечує перевіряння автентичності та цілісності IP-пакетів, а також захист від повторення раніше відправлених IP-пакетів.

Протокол AH дає можливість приймальній стороні переконатися в тому, що:

– пакет був відправлений саме тією стороною, з якою встановлено з'єднання;

– вміст пакета не було спотворено в процесі передавання його мережею;

– пакет не є дублікатом деякого пакета, отриманого раніше.

Протокол АН повністю захищає від підроблення та спотворення вміст IP-пакетів, зокрема дані протоколів вищих рівнів. Повнота захисту полів IP-заголовків залежить від використовуваного режиму роботи – тунельного або транспортного. Однак протокол АН не забезпечує конфіденційність переданих даних, тобто не призначений для їх шифрування. Дані можуть бути прочитані проміжними вузлами, але не можуть бути змінені. Цілісність і автентичність даних забезпечують додаванням заголовка (АН) перед заголовком IP і заголовком транспортного рівня (TCP / UDP). Формат заголовка АН зображено на рис. 6.20.

0 31

Наступний заголовок Однобайтове поле, яке містить код протоколу наступного заголовку, що вкладений в IPSec-пакет	**Довжина заголовку АН** (в 32-бітних словах)	**Зарезервовано**
Індекс параметрів захисту SPI 32-розрядна мітка безпечної асоціації, що містить всі параметри тунелю IPSec, включно з типами криптографічних алгоритмів і ключами шифрування. На основі індексу SPI пакет буде правильно віднесено до однієї з безпечних асоціацій на шлюзі чи вузлі		
Порядковий номер SN Беззнакове ціле число. Збільшується на 1 після передавання кожного захищеного IP-пакету		
Автентифікаційні дані (змінна довжина) Інформація, яку використовують для автентифікації пакету (Message Authentication Code) Integrity Check Value HMAC-MD5 HMAC-SHA1		

Рис. 6.20. Формат заголовка АН

Протокол АН захищає весь IP-пакет за винятком деяких полів в IP-заголовку, таких як ***час життя*** (Time-to-Live – TTL) і ***тип служби*** (Type of Service), які можуть змінюватися в процесі передавання пакета в мережі. Зауважимо, що протокол АН забезпечує захист від змін IP-адреси в заголовку пакета. Протокол АН створює своєрідний конверт, що забезпечує автентифікацію джерела даних, їх цілісність і захист від нав'язування повторних повідомлень.

Розташування заголовка АН у пакеті залежить від того, в якому режимі – транспортному або тунельному – сконфігурований захищений канал. На рис. 6.21 показане розташування АН-заголовка щодо IP-заголовка в обох режимах.

Заголовок вихідного IP-пакету	Заголовок AH	Заголовок TCP (UDP)	Дані
Автентифіковано			

а

Заголовок зовнішнього IP-пакету	Заголовок вихідного IP-пакету	Заголовок AH	Заголовок TCP (UDP)	Дані
Автентифіковано				

б

Рис. 6.21. Розташування заголовка AH у транспортному (а) або тунельному (б) режимах

У транспортному режимі заголовок вихідного IP-пакета стає зовнішнім заголовком, за ним іде заголовок AH, а потім усі дані захищеного пакета (тобто пакет протоколу верхнього рівня). Протокол AH захищає весь отриманий так пакет, зокрема заголовок IP і власне сам заголовок AH. Отже, будь-яку зміну даних у пакеті або заголовках буде виявлено. Слід також зауважити, що в цьому режимі дані пакета відсилають відкритими, тобто дані пакета захищені від змін, але не захищені від перегляду. Зокрема, не вдається приховати IP-адреси джерела й отримувача від можливого перегляду сторонніми особами, оскільки ці поля завжди присутні в незашифрованому вигляді й відповідають дійсним адресам вузлів мережі.

У тунельному режимі як заголовок зовнішнього IP-пакета створюють новий заголовок IP. IP-адреси відправника й отримувача можуть відрізнятися від адрес у заголовку вихідного IP-пакета. У захищеному IP-пакеті внутрішній (первинний) IP-заголовок містить цільову адресу пакета, а зовнішній IP-заголовок містить адресу кінця тунелю. За новим заголовком зовнішнього IP-пакета йде заголовок AH, а потім увесь вихідний пакет (заголовок IP і самі дані). Як і у випадку транспортного режиму, протокол AH захищає весь створений пакет (два заголовки IP, заголовок AH і дані), що також дає змогу виявити будь-які зміни в пакеті. Як і в транспортному режимі, сам пакет не захищений від перегляду.

Незалежно від режиму роботи протокол AH уживає заходів захисту від атак, спрямованих на порушення цілісності та автентичності пакетів повідомлень. За допомогою цього протоколу автентифікують кожен пакет, що робить неефективними програми, які намагаються перехопити управління

сеансом зв'язку. Протокол AH забезпечує автентифікацію не тільки вмісту, але й заголовків IP-пакетів. Проте слід пам'ятати, що автентифікація за протоколом AH не допускає маніпулювання основними полями IP-заголовка під час проходження пакета. Тому цей протокол не можна застосовувати в середовищі, де використовують механізм трансляції мережевих адрес (Network Address Translation – NAT), оскільки для його роботи необхідно мати можливість змінювати IP-заголовки.

Протокол AH можна застосовувати як окремо, так і в комбінації із протоколом ESP або навіть із пакетом, який уже містить AH-заголовок (вкладене застосування).

Протокол ESP. Протокол ESP забезпечує конфіденційність, автентичність, цілісність і захист від повторів для пакетів даних. Слід зазначити, що конфіденційність даних протокол ESP забезпечує завжди, а цілісність і автентичність є для нього вибірковими вимогами. Конфіденційність даних забезпечують шифруванням вмісту окремих пакетів. Цілісність і автентичність даних забезпечують на основі обчислення хешу.

З наведеного переліку функцій захисту інформаційного обміну видно, що функціональність протоколу ESP ширша, ніж у протоколу AH. Протокол ESP підтримує всі функції протоколу AH із захисту зашифрованих потоків даних від підроблення, відтворення й випадкового спотворення, а також забезпечує конфіденційність даних.

У протоколі ESP функції автентифікації й криптографічного закриття можна задіяти або разом, або окремо. Під час шифрування без автентифікації з'являється можливість використання механізму трансляції мережевих адрес, оскільки в цьому випадку адреси в заголовках IP-пакетів можна змінювати.

Для вирішення своїх завдань протокол ESP використовує заголовок формату, наведеного на рис. 6.22.

<table>
<tr><td colspan="3">Індекс параметрів захисту SPD</td></tr>
<tr><td colspan="3">Порядковий номер SN</td></tr>
<tr><td colspan="3">Дані</td></tr>
<tr><td>Дані</td><td colspan="2">Заповнювач (від 0 до 255 байт)</td></tr>
<tr><td>PAD</td><td>Довжина заповнювача</td><td>Наступний заголовок</td></tr>
<tr><td colspan="3">Автентифікаційні дані</td></tr>
</table>

Рис. 6.22. Структура заголовка ESP

Заголовок ESP містить такі поля:

– індекс параметрів захисту (Security Parameters Index – SPI) – використовують спільно з адресою отримувача й протоколом захисту (AH або ESP); він указує на відповідну угоду SA; отримувач використовує це значення для визначення угоди про захист, з яким ідентифікують цей пакет;

– порядковий номер SN (Sequence Number) – забезпечує захист від повторів для SA; він є 32-бітовим числом, яке спочатку дорівнює 1 і збільшується із кроком 1; воно не повторюється циклічно й указує номер пакета, відправленого за цією угодою; отримувач перевіряє це поле з метою впевнитися, що пакетів з таким номером прийнято ще не було; якщо ж такий пакет уже був, його не приймають;

– дані (Payload Data);

– заповнювач (Padding) – дописують від 0 до 255 байтів для 32-бітового вирівнювання з розміром блоку шифру;

– довжина заповнювача (Padding Length) – указує довжину поля заповнювача в байтах;

– наступний заголовок (Next Header) – указує природу переданих даних (наприклад, TCP або UDP);

– автентифікаційні дані (Authentication Data) – містять код перевірки цілісності (Integrity Check Value – ICV) і код автентичності повідомлення, використовувані для перевірки автентичності відправника й цілісності повідомлення. Значення ICV обчислюють для заголовка ESP, переданих даних і кінцевої мітки ESP. Поле Authentication Data розміщують у заголовок ESP тільки при включеній автентифікації.

Неважко помітити, що деякі поля заголовка ESP аналогічні полям заголовка AH: Next Header, SPI, SN, Authentication Data. Але є й два додаткові поля: заповнювач (Padding) і довжина заповнювача (Pad Length). Заповнювач може знадобитися в трьох випадках. По-перше, для нормальної роботи деяких алгоритмів шифрування необхідно, щоб текст містив кратну кількість блоків певного розміру. По-друге, формат заголовка ESP передбачає, що поле даних закінчується на межі чотирьох байтів. По-третє, заповнювач можна використовувати для приховування дійсного розміру пакета з метою забезпечення часткової конфіденційності трафіку, хоча протокол ESP обмежує можливості маскування 255 байтами заповнювача. Це зроблено для того, щоб корисна пропускна здатність каналу зв'язку не занадто знижувалася через великий обсяг надлишкових даних.

Програмне забезпечення для підтримки перерахованих протоколів (утиліти шифрування, цифрового підпису тощо) може функціонувати на серверах або комп'ютерах кінцевих користувачів. Однак частіше його

встановлюють на маршрутизаторах або спеціальних пристроях, які в архітектурі IPSec іменують ***шлюзами безпеки*** (security gateway).

Протокол ESP також використовують у двох режимах: транспортному й тунельному.

У транспортному режимі зашифровані дані транспортують безпосередньо між вузлами. У транспортному режимі протоколу ESP заголовок вихідного IP-пакета залишається зовнішнім. Заголовок ESP розміщують у переданий пакет між заголовками протоколів третього (IP) і четвертого (наприклад, TCP) рівнів. Слід зауважити, що поля протоколу ESP йдуть після стандартного IP-заголовка, а це означає, що такий пакет можливо маршрутизувати в мережі за допомогою звичайного обладнання, що підтримує IP.

Шифруванню піддаються тільки дані вихідного IP-пакета (пакет верхнього рівня) і заключна частина ESP заголовка (ESP trailer). У цьому режимі ESP не шифрує заголовок IP-пакета, інакше маршрутизатор не зможе прочитати поля заголовка й коректно здійснити передавання пакета між мережами. Не шифрують також поля SPI і SN, які необхідно передавати у відкритому вигляді, – для того, щоб доставлений пакет можна було зарахувати до певної SA і захиститися від помилкового відтворення пакета.

На відміну від протоколу AH, контроль цілісності та автентичності даних у протоколі ESP не поширюється на заголовок вихідного пакета, тому із цієї причини має сенс застосовувати обидва протоколи спільно: ESP для шифрування, а AH – для контролю цілісності.

Отже, адресну інформацію (IP-адреси обох сторін) видно під час пересилання пакета мережею, і несанкціонована зміна цих IP-адрес може залишитись непоміченою.

У тунельному режимі головна роль належить шлюзам безпеки, оскільки передбачають, що комп'ютери користувачів (або сервери) можуть не підтримувати IPSec і відправляють у мережу звичайний IP-трафік. Перед тим як досягти каналів глобальної мережі, кожен вихідний IP-пакет спочатку потрапляє в шлюз, який розміщує цей пакет цілком в "оболонку" IPSec, зашифрувавши його вміст разом із вихідним IP-заголовком. Щоб забезпечити можливість маршрутизації отриманого пакета, шлюз надає йому нового IP-заголовка і тільки після цього відправляє в мережу. Шлюз, що знаходиться на протилежному кінці з'єднання, розшифровує цей пакет і передає його на кінцевий пристрій у первісному вигляді. Описану процедуру називають ***тунелюванням***.

У тунельному режимі як зовнішній заголовок створюють новий заголовок IP. Весь вихідний IP-пакет (і дані, і заголовок IP) і заключну частину заголовка ESP (трейлер ESP) шифрують. Тому адресна інформація вихідного IP-пакета не доступна для перегляду. Заголовок зовнішнього IP-пакета протокол ESP не захищає.

Тунелювання дає змогу поширити дію засобів захисту на мережевий рівень моделі OSI і, зокрема, приховати справжні адреси джерела й отримувача. При цьому зменшується ризик атак, заснованих на детальному аналізі трафіку.

Порівнюючи протоколи ESP і AH, можна помітити, що вони дублюють функціональність один одного в галузі забезпечення автентифікації даних. Головною відмінністю протоколу AH від ESP у цьому питанні є те, що протокол AH забезпечує автентифікацію всього пакету (і IP заголовка, і самих даних), тоді як протокол ESP автентифікує тільки дані з пакета. Для шифрування в протоколі ESP використовують симетричний секретний ключ, тобто передані дані зашифровують та розшифровують за допомогою одного й того самого ключа. Для протоколу ESP також визначено перелік обов'язкових алгоритмів шифрування – DES, ***MD5*** (Message Digest 5) і ***SHA-1*** (Secure Hash Algorithm).

Під час автентифікації даних протокол ESP використовує ті самі алгоритми HMAC, що й протокол AH (що використовують MD5 або SHA-1 як функції хешування). Однак способи застосування відрізняються.

У транспортному режимі:

– протокол ESP автентифікує тільки дані з пакета, не зачіпаючи IP-заголовка;

– протокол AH захищає й дані, і обидва заголовки.

У тунельному режимі:

– автентифікацію в ESP протоколі застосовують до даних пакета й вихідного IP-заголовка, але не зачіпають новий IP-заголовок;

– протокол AH автентифікує дані, AH-заголовок і обидва IP-заголовки.

Протокол ESP можна застосовувати окремо або разом із протоколом AH. У разі спільного використання протоколи AH і ESP можна комбінувати різними способами. Якщо використовують транспортний режим, то аналогічно тому, як при застосуванні ESP автентифікацію здійснюють після шифрування, протокол AH необхідно застосовувати після протоколу ESP. У тунельному режимі протоколи AH і ESP застосовують до різних вкладених пакетів і, крім того, допускають багаторазову вкладеність тунелів із різними початковими й/або кінцевими точками.

6.9.3. Алгоритми автентифікації та шифрування в IPSec

Стек протоколів IPSec є узгодженим набором відкритих стандартів, що має цілком визначене ядро, водночас він може бути доволі просто доповнений новими протоколами, алгоритмами та функціями. Завдяки модульній структурі протоколи AH і ESP допускають застосування користувачами за їх узгодженим вибором різних криптографічних алгоритмів автентифікації й шифрування. Для

шифрування даних в IPSec (протокол ESP) можна застосовувати практично будь-який симетричний алгоритм шифрування, який використовує секретні ключі.

Для забезпечення цілісності та автентифікації даних (протоколи AH і ESP) використовують один із прийомів шифрування – шифрування за допомогою односторонньої функції (one-way function), званої також ***хеш-функцією*** (hash function) або ***дайджест-функцією*** (digest function). Ця функція, застосована до даних, дає в результаті значення-хеш, що складається з фіксованої невеликої кількості байтів. Хеш передають в IP-пакеті разом із вихідним повідомленням. Отримувач, знаючи, яку односторонню функцію шифрування застосували для складання хешу, заново обчислює його, використовуючи вихідне повідомлення. Якщо значення отриманого й обчисленого хешів збігаються, це означає, що вміст пакета під час передавання не було піддано жодним змінам. Знання хешу не дає можливості відновити вихідне повідомлення й тому не може бути використано для захисту конфіденційності, але воно дає змогу перевірити цілісність даних.

Хеш є свого роду контрольною сумою для вихідного повідомлення. На відміну від традиційної контрольної суми, для обчислення хешу використовують секретний ключ. Якщо для отримання хешу застосовували односторонню функцію з параметром (як такий використовують секретний ключ), відомим тільки відправнику й отримувачу, будь-яку модифікацію вихідного повідомлення буде негайно виявлено.

З метою забезпечення сумісності продуктів різних виробників робоча група IETF визначила базовий набір підтримуваних функцій і алгоритмів, який повинен бути однотипно здійснений у всіх продуктах, що підтримують IPSec. Сьогодні визначено 2 алгоритми автентифікації й 7 алгоритмів шифрування.

На цей час для протоколів AH і ESP зареєстровано 2 алгоритми автентифікації – HMAC-MD5 і HMAC-SHA1. Алгоритм HMAC визначений стандартом RFC 2104. Функції MD5 (стандарт RFC 1321) і SHA1 (стандарт FIPS 180-1) є функціями хешування. Алгоритми HMAC-MD5 і HMAC-SHA1 є алгоритмами автентифікації зі спільним секретним ключем. Секретний ключ має довжину 128 бітів у випадку MD5 і 160 бітів у випадку SHA1.

Якщо секретний ключ відомий тільки передавальній і приймальній сторонам, це забезпечить автентифікацію джерела даних, а також цілісність пакетів, що пересилають між двома сторонами. Ключі для HMAC генерують за допомогою процедури ISAKMP/Oakley. Для забезпечення сумісності обладнання й програмного забезпечення на початковій стадії виконання протоколу IPSec один із зареєстрованих алгоритмів автентифікації прийнято використовувати за замовчуванням. Як такий алгоритм визначено алгоритм HMAC-MD5.

Структуру алгоритму HMAC показано на рис. 6.23. Принцип дії алгоритму HMAC полягає в дворазовому обробленні пакета функцією хешування, керованою ключем автентифікації (наприклад, функцією хешування MD5). Як видно з рис. 6.23, обидва рази в оброблювані дані вміщують секретний ключ, який забезпечує автентифікацію переданої інформації. Отриману контрольну суму розміщують у заголовку AH протоколу. Перевіряння автентифікації на іншій стороні здійснюють шляхом повторного обчислення контрольної суми для пакета, що прийшов із використанням такого самого ключа, й порівняння отриманого результату з надісланим.

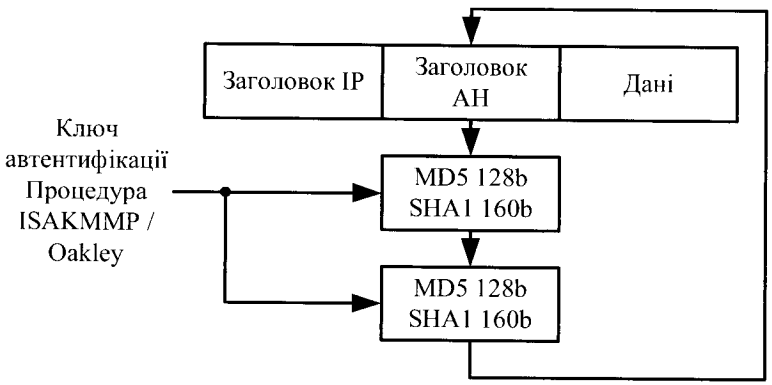


Рис. 6.23. Структура алгоритму HMAC

Алгоритм HMAC здійснює симетричну схему автентифікації, використовуючи ***код перевірки цілісності пакета*** (Integrity Check Value – ICV). По суті, він є цифровим підписом, що розміщують у поле автентифікації. Він дає змогу відправникові підписати результат попереднього хешування змістовної частини пакета ESP.

Аналіз умісту цього поля дає можливість отримувачу ідентифікувати джерело даних і переконатися в тому, що їх не було змінено в процесі передавання. Якщо для протоколу ESP функції автентифікації є необов'язковими, то для протоколу AH процес автентифікації обов'язковий.

Для протоколу ESP зареєстровано декілька алгоритмів шифрування. Найчастіше як алгоритми шифрування для ESP застосовують DES, 3DES і AES. Для забезпечення IPSec-сумісності за замовчуванням як алгоритм шифрування стандартом передбачено симетричний метод ***DES-CBC*** (Cipher BlockChaining – режим ланцюгування шифроблоків) з явно заданим вектором ініціалізації й з 56-розрядним ключем. Алгоритм AES всюди вбудовують у стандарт IPSec як альтернативу DES і 3DES.

Вибір алгоритму шифрування цілком залежить від розробника. Можливість вибору алгоритму шифрування надає користувачеві додаткову перевагу: зловмисник повинен не тільки розкрити шифр, а й визначити, який саме шифр йому треба розкривати, а разом із необхідністю підбору ключів це зменшує його шанси своєчасно розшифрувати дані користувача.

IPSec може працювати спільно із протоколами L2TP або L2F, які виконують тільки тунелювання, але не забезпечують шифрування й автентифікацію даних. Ці протоколи створюють тунель через Інтернет для пакетів будь-яких протоколів, упаковуючи їх у пакети IP. Коли трафік за допомогою L2F або L2TP виявляється впакованим у пакети IP, то далі для його захисту можна використовувати IPSec. У результаті комбінування IPSec із протоколами тунелювання типу L2F/L2TP дає змогу вирішити завдання захисту даних для протоколів, відмінних від IP.

Алгоритмічна незалежність протоколів AH і ESP потребує попереднього узгодження взаємодіючими сторонами набору застосовуваних алгоритмів та їхніх параметрів.

6.9.4. Протокол управління криптоключами IKE

Протоколи ESP і AH дають змогу забезпечити конфіденційність зв'язку, автентифікацію сторін і цілісність даних. Проте їхні функції втрачають будь-яку цінність за відсутності могутньої підтримувальної інфраструктури, що забезпечувала б розподіл ключів і узгодження протоколів між учасниками обміну.

Роль такої інфраструктури в IPSec виконує група протоколів **IKE** (Internet Key Exchange – протокол обміну ключами Інтернет). Ця назва прийшла в 1998 році на зміну більш ранньому – ISAKMP / Oakley, що безпосередньо вказувало на походження засобів керування ключами в складі IPSec.

Протокол ISAKMP, описаний у документі RFC 2408, дає можливість погоджувати алгоритми та математичні структури для процедури обміну ключами Діффі – Хеллмана, а також процесів автентифікації. Протокол Oakley, описаний в RFC 2412, оснований на алгоритмі Діффі – Хеллмана й слугує для безпосереднього обміну ключами.

Протоколи IKE вирішують три завдання:

– здійснюють автентифікацію взаємодіючих сторін, погоджують алгоритми шифрування й характеристики ключів, які буде використано в захищеному сеансі обміну інформацією;

– забезпечують створення, управління, безпосередній обмін ключами (зокрема можливість їхньої частої зміни);

– управляють параметрами з'єднання й захистом від деяких типів атак, контролюють виконання всіх досягнутих угод.

Розробники IPSec почали свою діяльність із вирішення останнього з перерахованих завдань. У результаті з'явилася концепція захищених віртуальних з'єднань, або безпечних асоціацій (SA).

Основою функціонування IPSec є захищені віртуальні з'єднання, або SA. Для того, щоб протоколи AH і ESP могли виконувати свою роботу із захисту даних, які передають, між двома кінцевими точками необхідно сформувати SA – угоду про захист обміну даними між двома взаємодіючими партнерами.

Установлення SA має починатися зі взаємної автентифікації сторін, тому що заходи безпеки втрачають будь-який сенс, якщо дані передають або приймають неавторизовані користувачі. Процедури встановлення SA виправдані лише в тому випадку, якщо в кожної зі сторін є повна впевненість у тому, що її партнер саме той, за кого він себе видає.

Для виконання автентифікації сторін у IKE застосовують два основні способи.

Перший спосіб оснований на використанні спільного секрету. Перед активуванням IPSec-пристроїв, що утворюють безпечні асоціації, в їхніх базах даних розміщують попередньо розподілений спільний секрет. Цифровий підпис на основі односторонньої функції, наприклад, MD5, що використовує як аргумент цей спільний секрет, доводить автентичність протилежної сторони.

Другий спосіб оснований на використанні технології цифрового підпису та цифрових сертифікатів стандарту X.509. Кожна зі сторін підписує свій цифровий сертифікат своїм закритим ключем і передає ці дані протилежній стороні. Якщо підписаний сертифікат розшифровують відкритим ключем відправника, то це засвідчує той факт, що відправник, який надав дані, насправді володіє відповідною частиною цього відкритого ключа – відповідним закритим ключем.

Однак слід зазначити, що для засвідчення автентичності сторони потрібно ще переконатися в автентичності самого сертифіката, і для цього сертифікат має бути підписаний не лише його власником, а й деякою третьою стороною, що видала сертифікат і викликає довіру. В архітектурі IPSec цю третю сторону називають ***органом сертифікації*** (Certification Authority – CA). Цей орган покликаний засвідчити справжність обох сторін і користуватися повною довірою сторін, а його відкритий ключ – відомий усім вузлам, що використовують його сертифікати для посвідчення особистостей один одного.

Після взаємної автентифікації взаємодіючі сторони можуть безпосередньо перейти до узгодження параметрів захищеного каналу. Обрані параметри SA визначають: протокол, використовуваний для забезпечення безпеки переда-

вання даних; алгоритм автентифікації протоколу AH і його ключі; алгоритм шифрування, використовуваний протоколом ESP, та його ключі; наявність або відсутність криптографічної синхронізації; способи захисту сеансу обміну; частоту зміни ключів та інші параметри. Важливим параметром SA є криптографічний матеріал, тобто секретні ключі, що використовують у роботі протоколів AH і ESP. Служби (сервіси) безпеки, пропоновані IPSec, використовують колективні секрети для формування криптографічних ключів.

Параметри SA повинні влаштовувати обидві кінцеві точки захищеного каналу. Тому у разі використання автоматичної процедури встановлення SA протоколи IKE, що працюють з різних боків каналу, вибирають параметри під час переговорного процесу. Для кожного завдання, виконуваного протоколами AH і ESP, пропонують кілька схем автентифікації й шифрування – це робить IPSec дуже гнучким засобом. У межах однієї SA може працювати тільки один із протоколів захисту даних – або AH, або ESP, але не обидва разом.

Для ідентифікації кожної SA призначено ***індекс параметрів безпеки*** (Security Parameters Index – SPI). Цей індекс вводять до заголовків захищених IPSec-пакетів, щоб приймальна сторона змогла правильно їх розшифрувати й автентифікувати, скориставшись зазначеною SA.

Система IPSec припускає застосування ручного та автоматичного способу встановлення SA. За ручного способу адміністратор конфігурує кожен кінцевий вузол так, щоб вони підтримували узгоджені параметри SA, включно з таємними ключами.

Для автоматичного встановлення SA необхідний відповідний протокол, яким у стандартах IPSec визначено протокол IKE. Він є комбінацією протоколів ISAKMP, Oakley і SKEME. Цей протокол забезпечує узгодження параметрів віртуального каналу та керування ключами й описує базову технологію автентифікації, обміну ключами та узгодження решти параметрів IPSec-тунеля при створенні SA, проте самі протоколи автентифікації сторін та обміну ключами в ньому детально не визначено. Тому при розробленні протоколу IKE загальні правила й процедури протоколу ISAKMP доповнено процедурами автентифікації й обміну ключами, узятими із протоколів Oakley і SKEME. Оскільки протокол IKE використовує для управління SA алгоритми та формати протоколу ISAKMP, назви цих протоколів іноді використовують як синоніми.

На підставі протоколу ISAKMP узгоджувати параметри захищеної взаємодії необхідно під час формування як IPSec-тунелю, так і в його межах кожного захищеного односпрямованого з'єднання. Параметри IPSec-тунелю узгоджують за протоколом ISAKMP / Oakley. Параметри кожного захищеного односпрямованого з'єднання узгоджують у межах сформованого IPSec-тунелю й утворюють SA.

Криптографічні ключі для кожного захищеного односпрямованого з'єднання генерують на основі ключів, вироблених у межах IPSec-тунелю. При цьому враховують алгоритми автентифікації й шифрування, які використовують у протоколах AH і ESP.

Стандарти IPSec дають змогу шлюзам використовувати як одну асоціацію SA для передавання трафіку всіх взаємодіючих через Інтернет вузлів, так і створювати для цієї мети довільну кількість SA, наприклад, по одній на кожне з'єднання TCP.

Бази даних SAD і SPD. IPSec пропонує різні методи захисту трафіку. У кожному вузлі, що підтримує IPSec, використовують бази даних двох типів:

- база даних безпечних асоціацій (Security Associations Database – SAD);
- база даних політики безпеки (Security Policy Database – SPD).

При встановленні SA дві сторони приймають ряд угод, що регламентують процес передавання потоку даних між ними. Угоди подають у вигляді набору параметрів. Для SA такими параметрами є, зокрема, тип і режим роботи протоколу захисту (AH або ESP), методи шифрування, секретні ключі, значення поточного номера пакета в SA та інша інформація.

Об'єднання службової інформації в межах SA надає користувачеві можливість сформувати різні класи захисту, призначені, наприклад, для електронного спілкування з різними користувачами. Інакше кажучи, застосування структур SA надає можливість побудови множини віртуальних приватних мереж, що відрізняються своїми параметрами.

Набори поточних параметрів, що визначають усі активні SA, зберігають на обох кінцевих вузлах захищеного каналу у вигляді SAD. Кожен вузол IPSec підтримує дві бази SAD – одну для вихідних SA, іншу – для вхідних.

SPD задає відповідність між IP-пакетами й установленими для них правилами оброблення. При обробленні пакетів бази даних SPD використовують спільно з базами даних SAD. SPD є впорядкованим набором правил, кожне з яких містить сукупність вибіркових ознак і допустимих політик безпеки. Вибіркові ознаки слугують для відбору пакетів, а політики безпеки задають необхідне оброблення. Таку базу даних формують і підтримують на кожному вузлі, де встановлено програмне забезпечення IPSec.

6.9.5. Режими та схеми застосування IPSec

Застосування тунельного або транспортного режиму залежить від вимог, що пред'являють до захисту даних, а також від ролі вузла, в якому працює IPSec. Вузлом (host – хост), що завершує захищений канал, може бути вузол (кінцевий вузол) або шлюз (проміжний вузол). Відповідно розрізняють три основні схеми застосування IPSec:

1) вузол–вузол;

2) шлюз безпеки–шлюз безпеки;

3) вузол–шлюз безпеки.

У схемі 1 захищений канал, або, що в даному контексті одне й те саме, SA, установлюють між двома кінцевими вузлами мережі – вузлом 1 і вузлом 2 (рис. 6.24).

Рис. 6.24. Захищений канал між вузлами

Протокол IPSec у цьому випадку працює на кінцевому вузлі й захищає дані, що надходять на нього. Для вузлів, що підтримують IPSec, можна використовувати як транспортний режим, так і тунельний.

Відповідно до схеми 2 захищений канал установлюють між двома проміжними вузлами, названими шлюзом безпеки 1 (security gateway) і шлюзом безпеки 2, на кожному з яких працює протокол IPSec (рис. 6.25).

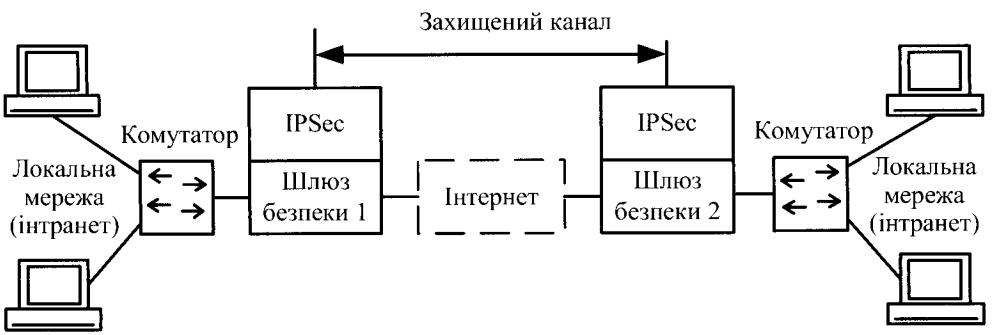


Рис. 6.25. Захищений канал між шлюзами безпеки

Захищений обмін даними може відбуватися між будь-якими двома кінцевими вузлами, підключеними до мереж, які розташовані позаду шлюзів безпеки. Від кінцевих вузлів підтримка протоколу IPSec не потрібна, вони передають свій трафік у незахищеному вигляді через довірену приватну мережу підприємства. Трафік, що скеровують до публічної мережі, проходить через шлюз безпеки, який і забезпечує його захист за допомогою IPSec, діючи від свого імені. Шлюзам безпеки дозволено використовувати тільки тунельний

режим роботи, хоча вони могли б підтримувати й транспортний режим, але він у цьому випадку малоефективний.

У разі захищеного віддаленого доступу часто застосовують схему вузол–шлюз безпеки (рис. 6.26). Тут захищений канал організують між кінцевим вузлом 1, на якому працює IPSec, і шлюзом безпеки, що захищає трафік для всіх вузлів, що входять до приватної мережі підприємства. Кінцевий вузол 1 може використовувати для відправлення пакетів шлюзу як транспортний, так і тунельний режим, а шлюз безпеки відправляє пакети вузла лише в тунельному режимі.

Цю схему можна модифікувати, створивши паралельно ще один захищений канал – між віддаленим вузлом 1 і яким-небудь вузлом 2, що належить внутрішній мережі, яку захищають шлюзом. Таке комбіноване використання двох захищених каналів дає можливість надійно захистити трафік і у внутрішній мережі.

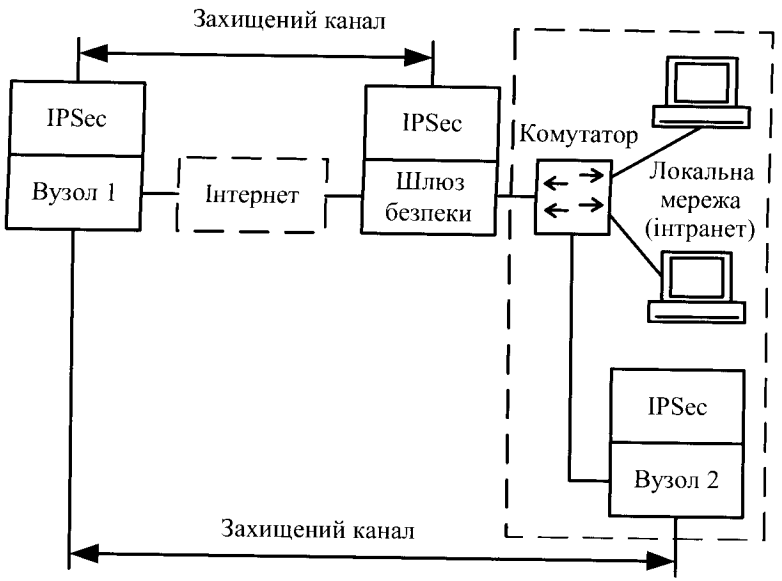


Рис 6.26. Захищений віддалений доступ

Розглянуті схеми побудови захищених каналів на базі IPSec широко застосовують для створення різноманітних віртуальних приватних мереж (VPN) – від мереж інтернет-провайдерів, що дають змогу керувати обслуговуванням користувачів безпосередньо на їхніх площах, до приватних мереж, які розгортають і якими керують самі компанії. На базі IPSec успішно здійснюють віртуальні приватні мережі будь-якої архітектури, зокрема ***VPN із віддаленим доступом*** (Remote Access VPN), ***внутрішньокорпоративні VPN*** (Intranet VPN) і ***міжкорпоративні VPN*** (Extranet VPN).

6.9.6. Переваги застосування засобів безпеки IPSec

Система стандартів IPSec увібрала в себе прогресивні методики й досягнення в галузі мережевої безпеки, завоювала визнання фахівців як надійна й легко інтегрована система безпеки для IP-мереж. Система IPSec міцно займає сьогодні лідируючі позиції в наборі стандартів для створення VPN. Цьому сприяє її відкрита побудова, здатна використовувати всі нові досягнення в галузі криптографії. IPsec дає змогу захистити мережу від більшості мережевих атак, відкидаючи чужі пакети ще до того, як вони досягнуть рівня IP на приймальному комп'ютері. У захищений комп'ютер або мережу можуть прийти тільки пакети від зареєстрованих партнерів із взаємодії. IPsec забезпечує:

– автентифікацію – доказ відправлення пакетів вашим партнером із взаємодії, тобто власником спільного секрету;

– цілісність – неможливість зміни даних у пакеті;

– конфіденційність – неможливість розкриття переданих даних;

– надійне управління ключами – протокол IKE обчислює спільний секрет, відомий лише отримувачу й відправнику пакета;

– тунелювання – повне маскування топології локальної мережі підприємства.

Робота з використанням стандартів IPSec забезпечує повний захист інформаційного потоку даних від відправника до отримувача, припиняючи трафік для спостерігачів на проміжних вузлах мережі. VPN-рішення на основі стека протоколів IPSec забезпечують побудову віртуальних приватних мереж, їх безпечну експлуатацію та інтеграцію з відкритими комунікаційними системами.

6.10. Протоколи захисту даних на сеансовому рівні моделі OSI

Найвищим рівнем моделі OSI, на якому можливе формування захищених віртуальних каналів, є п'ятий – сеансовий – рівень [9–22]. Під час побудови віртуальних приватних мереж на сеансовому рівні з'являється можливість криптографічного захисту інформаційного обміну, включаючи автентифікацію, а також здійснення ряду функцій посередництва між взаємодіючими сторонами.

Дійсно, сеансовий рівень моделі OSI відповідає за встановлення логічних з'єднань і управління цими з'єднанями. Тому існує можливість застосування на цьому рівні програм-посередників, які перевіряють допустимість запитаних з'єднань і забезпечують виконання інших функцій захисту міжмережевої взаємодії.

Однак на сеансовому рівні починається безпосередня залежність від додатків, що здійснюють високорівневі протоколи. Тому здійснення протоколів захисту інформаційного обміну, що відповідають цьому рівню, у більшості випадків вимагає внесення змін до високорівневих мережевих додатків.

Для захисту інформаційного обміну на сеансовому рівні поширений ***протокол захищених з'єднань*** (Secure Sockets Layer – SSL). Для виконання на сеансовому рівні функцій посередництва між взаємодіючими сторонами організація IETF як стандарт прийняла протокол SOCKS (Socket Secure – безпека сокета).

6.10.1. Протоколи SSL / TLS

Протокол SSL застосовують як протокол захищеного каналу, який працює на сеансовому рівні моделі OSI. Цей протокол використовує криптографічні методи захисту інформації для забезпечення безпеки інформаційного обміну. Протокол SSL виконує всі функції зі створення захищеного каналу між двома користувачами мережі, зокрема їх взаємну автентифікацію, забезпечення конфіденційності, цілісності та автентичності переданих даних. Ядром протоколу SSL є технологія комплексного використання асиметричних і симетричних криптосистем.

Взаємну автентифікацію обох сторін в SSL виконують, обмінюючись цифровими сертифікатами відкритих ключів користувачів (клієнта й сервера), завіреними цифровим підписом спеціальних сертифікаційних центрів. Протокол SSL підтримує сертифікати, які відповідають загальноприйнятому стандарту X.509, а також стандарти ***інфраструктури відкритих ключів*** (Public Key Infrastructure – PKI), за допомогою якої організовують видавання та перевіряння достовірності сертифікатів.

Конфіденційність забезпечують шифруванням переданих повідомлень із використанням симетричних сесійних ключів, якими сторони обмінюються під час встановлення з'єднання. Сесійні ключі передають також у зашифрованому вигляді, при цьому їх шифрують за допомогою відкритих ключів, отриманих із сертифікатів користувачів. Використання для захисту повідомлень симетричних ключів пов'язано з тим, що швидкість процесів шифрування й розшифрування на основі симетричного ключа істотно вища, ніж із використанням несиметричних ключів. Автентичність та цілісність переданої інформації забезпечують завдяки формуванню та перевірянню електронного цифрового підпису.

Як алгоритми асиметричного шифрування використовують алгоритм RSA, а також алгоритм Діффі – Хеллмана. Допустимими алгоритмами симетричного шифрування є ***RC2*** (Ron's Code 2 або Rivest's Cipher 2), ***RC4***

Rivest's Cipher 4), DES, 3DES і AES. Для обчислення хеш-функцій можна застосовувати стандарти MD5 і SHA-1. У протоколі SSL версії 3.0 набір криптографічних алгоритмів є розширюваним.

Протокол SSL передбачає такі етапи взаємодії клієнта й сервера під час формування та підтримки захищеного з'єднання:

– установлення SSL-з'єднання;

– захищена взаємодія.

У процесі встановлення SSL-з'єднання вирішують такі завдання:

– автентифікація сторін;

– узгодження криптографічних алгоритмів і алгоритмів стиснення, які використовуватимуть під час захищеного інформаційного обміну;

– формування загального секретного майстер-ключа;

– генерування на основі сформованого майстер-ключа загальних секретних сеансових ключів для криптозахисту інформаційного обміну.

Процедуру встановлення SSL-з'єднання, звану також процедурою рукостискання, відпрацьовують перед безпосереднім захистом інформаційного обміну й виконують за ***протоколом початкового привітання*** (Handshake Protocol), що входить до складу протоколу SSL.

При встановленні повторних з'єднань між клієнтом і сервером сторони можуть, за взаємною згодою, формувати нові сеансові ключі на основі старого спільного ключа (цю процедуру називають продовженням SSL-з'єднання).

Протокол SSL 3.0 підтримує три режими автентифікації:

– взаємну автентифікацію сторін;

– односторонню автентифікацію сервера без автентифікації клієнта;

– повну анонімність.

За останнім варіантом забезпечують захист інформаційного обміну без будь-яких гарантій щодо справжності сторін. У цьому випадку взаємодіючі сторони не захищені від атак, пов'язаних із підміною учасників взаємодії.

За протоколом SSL для автентифікації взаємодіючих сторін і формування спільних секретних ключів зазвичай використовують алгоритм RSA. Відповідність між відкритими ключами та їх власниками встановлюють за допомогою цифрових сертифікатів, що видають спеціальні центри сертифікації.

Протокол SSL пройшов перевірку часом, працюючи в популярних браузерах ***Netscape Navigator*** та ***Internet Explorer***, а також веб-серверах провідних виробників. У січні 1999 року на зміну версії SSL 3.0 прийшов протокол ***TLS*** (Transport Layer Security – протокол захисту транспортного рівня), який ґрунтується на протоколі SSL і водночас є стандартом Інтернету. Відмінності між протоколами SSL 3.0 і TLS 1.0 не надто істотні. Протокол SSL став промисловим протоколом.

Протокол SSL підтримує програмне забезпечення серверів і клієнтів, розроблене провідними західними компаніями. Істотним недоліком протоколу SSL є те, що практично всі продукти, що підтримують SSL, через експортні обмеження доступні за межами США лише в скороченому варіанті (з довжиною сеансового ключа 40 бітів для алгоритмів симетричного шифрування й 512 біти для алгоритму RSA, використовуваного на етапі встановлення SSL-з'єднання).

Недоліками протоколів SSL і TLS є й те, що для транспортування своїх повідомлень вони використовують тільки один протокол мережевого рівня – IP, – і, отже, можуть працювати тільки в IP-мережах.

Крім того, в SSL для автентифікації й шифрування використовують однакові ключі, що за певних умов може призвести до потенційної уразливості. Подібне рішення дає можливість зібрати більше статистичного матеріалу, ніж у разі автентифікації й шифрування різними ключами.

Згідно із протоколом SSL криптозахищені тунелі створюють між кінцевими точками віртуальної мережі. Ініціаторами кожного криптозахищеного тунелю є клієнт і сервер, що функціонують на комп'ютерах у кінцевих точках тунелю.

6.10.2. Протокол SOCKS

Протокол SOCKS організовує процедуру взаємодії клієнт-серверних додатків на сеансовому рівні моделі OSI через проксі-сервер.

У загальному випадку програми-посередники, які традиційно використовують у міжмережевих екранах, можуть виконувати такі функції:

– ідентифікацію та автентифікацію користувачів;

– криптозахист переданих даних;

– розмежування доступу до ресурсів внутрішньої мережі;

– розмежування доступу до ресурсів зовнішньої мережі;

– фільтрацію й перетворення потоку повідомлень, наприклад, пошук вірусів і прозоре шифрування інформації;

– трансляцію внутрішніх мережевих адрес для вихідних потоків повідомлень.

Спочатку протокол SOCKS розробляли лише для перескерування запитів до серверів із боку клієнтських додатків, а також повернення цим додаткам отриманих відповідей. Перескерування запитів і відповідей між клієнт-серверними додатками вже дає змогу здійснити функцію трансляції мережевих IP-адрес. Заміна у вихідних пакетів внутрішніх IP-адрес відправників однією IP-адресою шлюзу дає можливість приховати топологію внутрішньої мережі від зовнішніх користувачів і тим самим ускладнити завдання несанкціонованого доступу.

На основі протоколу SOCKS можна здійснити й інші функції посередництва із захисту мережевої взаємодії. Наприклад, протокол SOCKS можна застосовувати для контролю над напрямами інформаційних потоків і розмежування доступу залежно від атрибутів користувачів та інформації. Ефективність використання протоколу SOCKS для виконання функцій посередництва забезпечують його орієнтацією на сеансовий рівень моделі OSI. Порівняно з посередниками прикладного рівня на сеансовому рівні досягають вищої швидкодії й незалежності від високорівневих протоколів (HTTP, FTP, POP3, SMTP тощо). Крім того, протокол SOCKS не прив'язаний до протоколу IP і не залежить від операційної системи. Наприклад, для обміну інформацією між клієнтськими додатками й посередником можна використовувати протокол IPX.

Завдяки протоколу SOCKS міжмережеві екрани й віртуальні приватні мережі можуть організувати безпечну взаємодію та обмін інформацією між різними мережами. Протокол SOCKS дає можливість здійснити безпечне управління цими системами на основі уніфікованої стратегії. Слід зазначити, що на основі протоколу SOCKS можна створювати криптозахищені тунелі для кожної програми й сеансу окремо.

Згідно із специфікацією протоколу SOCKS розрізняють SOCKS-сервер, який доцільно встановлювати на міжмережевий екран мережі, і SOCKS-клієнт, який установлюють на кожен користувацький комп'ютер. SOCKS-сервер забезпечує взаємодію з будь-яким прикладним сервером від імені відповідного цьому серверу прикладного клієнта. SOCKS-клієнт призначений для перехоплення всіх запитів до прикладного сервера з боку клієнта й передавання їх SOCKS-серверу. Слід зазначити, що SOCKS-клієнти, які перехоплюють запити клієнтських додатків і взаємодіють з SOCKS-сервером, можуть бути вбудовані в універсальні клієнтські програми. SOCKS-серверу відомо про трафік на рівні сеансу (сокета), тому він може насправді контролювати і, зокрема, блокувати роботу конкретних програм користувачів, якщо вони не мають необхідних повноважень на інформаційний обмін.

Протокол SOCKS v5 схвалено організацією IETF як стандарт Інтернет і включений в RFC 1928.

Загальну схему встановлення з'єднання за протоколом SOCKS v5 можна описати так:

– запит прикладного клієнта, який бажає встановити з'єднання з яким-небудь прикладним сервером у мережі, перехоплює встановлений на цьому самому комп'ютері SOCKS-клієнт;

– з'єднавшись із SOCKS-сервером, SOCKS-клієнт повідомляє йому ідентифікатори всіх методів автентифікації, які він підтримує;

– SOCKS-сервер вирішує, яким методом автентифікації скористатися (якщо SOCKS-сервер не підтримує жодного із методів автентифікації, запропонованих SOCKS-клієнтом, то з'єднання розриває);

– за підтримки будь-яких запропонованих методів автентифікації SOCKS-сервер відповідно до обраного методу автентифікує користувача, від імені якого виступає SOCKS-клієнт; у разі неуспішної автентифікації SOCKS-сервер розриває з'єднання;

– після успішної автентифікації SOCKS-клієнт передає SOCKS-серверу DNS-ім'я або IP-адресу запитуваного прикладного сервера в мережі, потім SOCKS-сервер на основі наявних правил розмежування доступу приймає рішення про встановлення з'єднання із цим прикладним сервером;

– у разі встановлення з'єднання прикладний клієнт і прикладний сервер взаємодіють один з одним по ланцюжку з'єднань, в якому SOCKS-сервер ретранслює дані, а також може виконувати функції посередництва із захисту мережевої взаємодії; наприклад, якщо під час автентифікації SOCKS-клієнт і SOCKS-сервер обмінялися сеансовим ключем, то весь трафік між ними може бути зашифрований.

Автентифікація користувача, виконувана SOCKS-сервером, може ґрунтуватися на цифрових сертифікатах у форматі X.509 або паролях. Для шифрування трафіку між SOCKS-клієнтом і SOCKS-сервером можна використовувати протоколи, орієнтовані на сеансовий або нижчі рівні моделі OSI. Крім автентифікації користувачів, трансляції IP-адрес і криптозахисту трафіку, SOCKS-сервер може виконувати також такі функції:

– розмежування доступу до ресурсів внутрішньої мережі;

– розмежування доступу до ресурсів зовнішньої мережі;

– фільтрація потоку повідомлень, наприклад, динамічний пошук вірусів;

– реєстрування подій та реагування на події, що відбулись;

– кешування даних, які запитують із зовнішньої мережі.

Протокол SOCKS здійснює вбудовану підтримку ***Internet Explorer*** компанії Microsoft.

Спеціальні програми, звані соксифікаторами, доповнюють клієнтські програми підтримкою протоколу SOCKS. До таких програм належить, наприклад, NEC SocksCap тощо. При встановленні соксифікатор впроваджують між додатками й стеком комунікаційних протоколів. Потім у процесі роботи він перехоплює комунікаційні виклики, що формують додатки, і перескеровує їх у разі потреби на SOCKS-сервер. За відсутності порушень установлених правил безпеки робота SOCKS-клієнта абсолютно прозора для клієнтських додатків і користувачів.

Отже, для формування захищених віртуальних мереж за протоколом SOCKS в точці сполучення кожної локальної мережі з Інтернетом на комп'ютері-шлюзі встановлюють SOCKS-сервер, а на робочих станціях у локальних мережах і на комп'ютерах віддалених користувачів установлюють SOCKS-клієнти (рис. 6.27). По суті, SOCKS-сервер можна розглядати як міжмережевий екран, що підтримує протокол SOCKS.

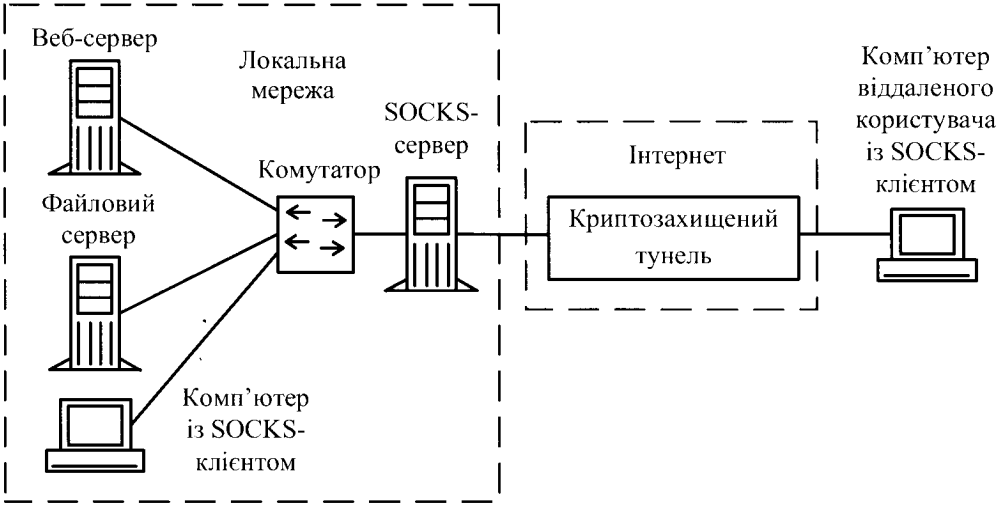


Рис. 6.27. Захищена віртуальна мережа за протоколом SOCKS

Віддалені користувачі можуть підключатися до Інтернету будь-яким способом – через комутовану або виділену лінію. При спробі користувача захищеної віртуальної приватної мережі встановити з'єднання з яким-небудь прикладним сервером SOCKS-клієнт починає взаємодіяти із SOCKS-сервером. У разі завершення першого етапу взаємодії користувач буде автентифікований, а перевірка правил доступу покаже, чи має він право з'єднатися з конкретним серверним додатком, функціонуючим на комп'ютері з указаною адресою. Подальша взаємодія може відбуватися по криптографічно захищеному каналу.

Крім захисту локальної мережі від несанкціонованого доступу, на SOCKS-сервер може покладатися контроль доступу користувачів цієї локальної мережі до відкритих ресурсів Інтернету (Telnet, WWW, SMTP, POP3 тощо) Доступ є повністю вповноваженим, оскільки ідентифікують і автентифікують конкретних користувачів, а не комп'ютери, з яких вони входять у мережу. Правила доступу можуть забороняти або дозволяти з'єднання з конкретними ресурсами мережі Інтернет залежно від повноважень конкретного співро-

бітника. Дія правил доступу може залежати й від інших параметрів, наприклад, від методу автентифікації або часу доби.

На додаток до функцій розмежування доступу SOCKS-сервер може реєструвати події та реагувати на них. Для досягнення вищого ступеня безпеки мережевої взаємодії сервери локальної мережі, до яких дозволено доступ із боку Інтернету, повинні бути виділені в окремий приєднаний до SOCKS-сервера сегмент, який утворює захищену відкриту підмережу.

6.11. Протоколи захисту даних на прикладному рівні моделі OSI

Розвиток інформаційних технологій (Information Technologies – IT) дає можливість підвищити ефективність діяльності підприємств та організацій, а також відкриває нові можливості для взаємодії з потенційними клієнтами на базі публічних мереж, зокрема Інтернету [9–22]. Створення веб-сайту – своєрідного представництва підприємства чи організації в Інтернет – є лише першим кроком на цьому шляху. Активне ведення комерційних операцій в Інтернет передбачає масовий доступ споживачів електронних послуг (або веб-клієнтів) до інтернет-додатків і проведення електронних платежів мільйонами користувачів Інтернету. Розміщення інтернет-додатків усередині приватної мережі може завдати шкоди безпеці IT-інфраструктури, оскільки відкриття доступу через міжмережевий екран неминуче створює потенційну можливість для несанкціонованого проникнення зловмисників у мережу підприємства чи організації.

Забезпечення інформаційної безпеки повинно передбачати вирішення таких завдань, як безпечний доступ до веб-серверів і веб-додатків, автентифікація та авторизація користувачів, забезпечення цілісності та конфіденційності даних, здійснення електронного цифрового підпису тощо.

Підприємства та організації потребують надійних, гнучких, безпечних методів та засобів для отримання й використання відкритої чи конфіденційної інформації численними групами людей – своїх співробітників, партнерів, клієнтів та постачальників. Проблема полягає в забезпеченні доступу до такої інформації тільки авторизованим користувачам. Доцільно використовувати інтегровану систему управління доступом користувачів до інформації з багатьох точок доступу й додатків. Така система вирішує багато проблем контролю доступу, з якими стикаються підприємства та організації, забезпечуючи при цьому зручний доступ і високу безпеку.

6.11.1. Керування ідентифікацією та доступом

Для забезпечення зростаючих потреб електронного бізнесу необхідно побудувати надійне з погляду безпеки середовище для здійснення електронного бізнесу в режимі реального часу. Технології, які дають можливість здійснювати електронний бізнес, виконують чотири основні функції:

– автентифікації або перевіряння автентичності користувача;

– керування доступом, що дає змогу авторизованим користувачам отримувати доступ до необхідних ресурсів;

– шифрування, яке гарантуватиме, що зв'язок між користувачем і базовою інфраструктурою захищений;

– неспростовності, що означає, що користувачі не можуть пізніше відмовитися від виконаної дії (зазвичай здійснюють за допомогою цифрового підпису та інфраструктури відкритих ключів).

Тільки рішення, яке виконує всі ці чотири функції, може створити довірене середовище, здатне по-справжньому забезпечити потреби електронного бізнесу.

Керування доступом є критичним компонентом загальної системи безпеки. Система керування доступом забезпечує авторизованим користувачам доступ до належних ресурсів. Проектування цієї інфраструктури потребує тонкого балансу між наданням доступу до критичних ресурсів тільки авторизованим користувачам і забезпеченням необхідної безпеки цих ресурсів, відомих великій кількості користувачів.

Особливості керування доступом. У розподіленій мережі підприємства чи організації зазвичай застосовують два методи керування доступом:

– керування мережевим доступом (регулює доступ до ресурсів внутрішньої мережі підприємства);

• керування веб-доступом (регулює доступ до веб-серверів та їхнього вмісту).

Усі запити на доступ до ресурсів проходять через один або більше ***списків контролю доступу*** (Access Control List – ACL). ACL є набором правил доступу, які задають для набору ресурсів, що захищають. Ресурси з низьким ризиком підпорядковуються менш суворим правилам доступу, тоді як висококритичні ресурси повинні мати суворіші правила доступу. ACL, по суті, визначають політику безпеки. Доступ до мережевих ресурсів організації можна регулювати створенням списків контролю доступу, які дають можливість точно визначити конкретні дозволи та умови для отримання доступу до ресурсів внутрішньої мережі.

Засоби контролю й керування веб-доступом дають змогу створювати та виконувати політику веб-доступу. Створюючи конкретні списки контролю веб-доступу, адміністратори безпеки визначають, які користувачі можуть отримати доступ до веб-серверів підприємства та їхнього вмісту й за яких заздалегідь установлених умов.

Керування доступом спрощується із застосуванням єдиної централізованої інфраструктури контролю та керування доступом, яка може дозволити користувачам самообслуговування, доручаючи їм такі завдання керування, як реєстрація, редагування профілю, відновлення пароля й керування підпискою. Вона може також забезпечити делегування адміністрування, передання функцій керування користувачами людям, найбільш обізнаним про конкретну групу користувачів як усередині – у бізнес-підрозділах підприємства, так і поза нею – у клієнтів і в підрозділах бізнес-партнерів. Щоб полегшити підтримування системи безпеки масштабу підприємства, засоби керування доступом можуть отримувати дані користувачів і політик, уже збережених у таких існуючих сховищах даних, як каталоги ***LDAP*** (Lightweight Directory Access Protocol – спрощений протокол доступу до мережевих каталогів) і реляційні бази даних.

6.11.2. Функціонування системи керування доступом

Централізовані системи керування доступом випускає ряд компаній, зокрема Secure Computing, RSA Security Inc., Baltimore тощо.

Розглянемо функціонування системи керування доступом на прикладі системи PremierAccess компанії Secure Computing. Ця система здійснює керування веб-доступом і мережевим доступом усіх користувачів, зокрема внутрішніх користувачів, віддалених співробітників, клієнтів, постачальників і бізнес-партнерів. Вона ґрунтується на політиці безпеки, яка дає змогу персоналізувати права доступу користувачів. Користувачі отримують доступ тільки до тих ресурсів, на які було дано дозвіл відповідно до їхніх прав доступу, через веб-доступ, VPN-доступ або віддалений доступ із використанням серверів RADIUS. Система використовує засновані на застосуванні каталогів процеси автентифікації, авторизації й адміністрування дій користувачів. Система підтримує різні типи автентифікації – від багаторазових паролів до біометричних засобів автентифікації. Перевагу надають методам і засобам суворої автентифікації.

Засоби керування користувачами дають змогу керувати великою кількістю користувачів. Сервер реєстрації дає можливість самим користувачам реєструватися в мережі, використовуючи стандартні веб-браузери. У процесі

реєстрації користувачам призначають ролі. Ролі є ярликами, що ідентифікують групи користувачів, які володіють однаковими правами доступу. Інакше кажучи, ролі визначають набори правил доступу, застосовувані до конкретних груп користувачів. Категоріювання користувачів за ролями можна виконати на основі їхніх функційних обов'язків.

У системі керування доступом використовують агенти (рис. 6.28). Агент системи – це програмний модуль, установлений на відповідний сервер у межах мережі підприємства.

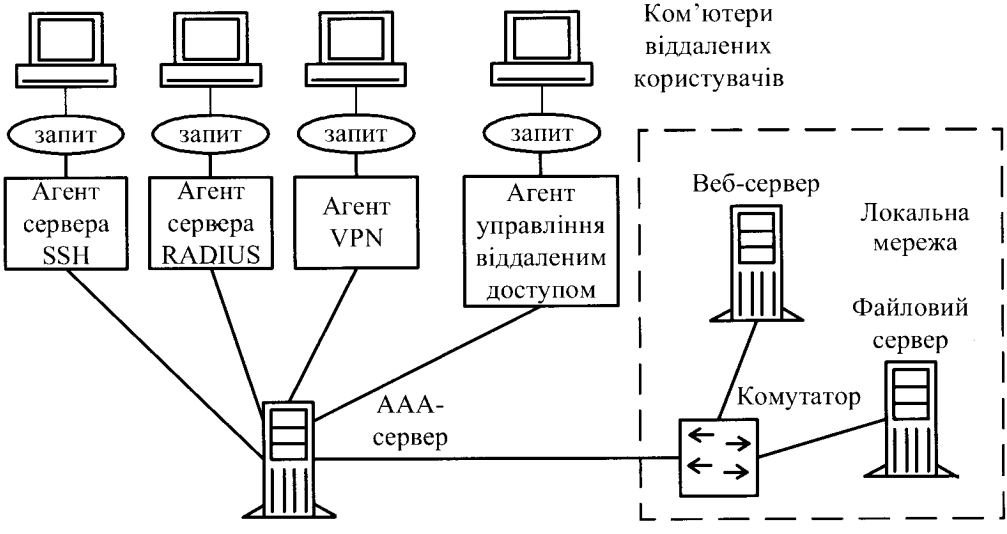


Рис. 6.28. Агенти системи керування віддаленим доступом

Такими агентами є агенти віддаленого доступу, агенти VPN-доступу, агенти серверів RADIUS, Novel, RAS, Citrix тощо. За спроби користувача підключитися до внутрішньої мережі агенти системи перехоплюють запит користувача на вхід у мережу.

Агенти діють як ***точки автентифікації користувачів*** (User Authentication Points – UAPs) на лініях з'єднання із сервером PremierAccess. У відповідь на запит користувача агент запитує в користувача його довірчі дані – код користувача та автентифікатор. Відповідаючи на запит агента, користувач вводить свої дані. Ці довірчі дані передають ***AAA-серверу*** (AAA – Authentication, Authorization, Accounting – автентифікація, авторизація, облік).

AAA-сервер порівнює ідентифікатор користувача або сертифікат із даними, збереженими в каталозі LDAP, з метою перевіряння їхньої тотожності.

Якщо ідентифікатор користувача збігається зі збереженим, запис користувача в базі даних перевіряють за роллю (або ролями) і ресурсами, до яких він авторизується. Для автентифікації можуть застосовувати фіксований пароль, апаратний або програмний автентифікатор. Якщо користувач успішно проходить усі кроки підтвердження своєї достовірності, він отримує доступ до ресурсу мережі.

Система PremierAccess використовує ***універсальний веб-агент*** (Universal Web Agent – UWA). У розглянутому прикладі користувачем є бізнес-партнер, який запитує доступ до захищеного веб-ресурсу підприємства.

Керують веб-доступом за допомогою процесу у два етапи (рис. 6.29).

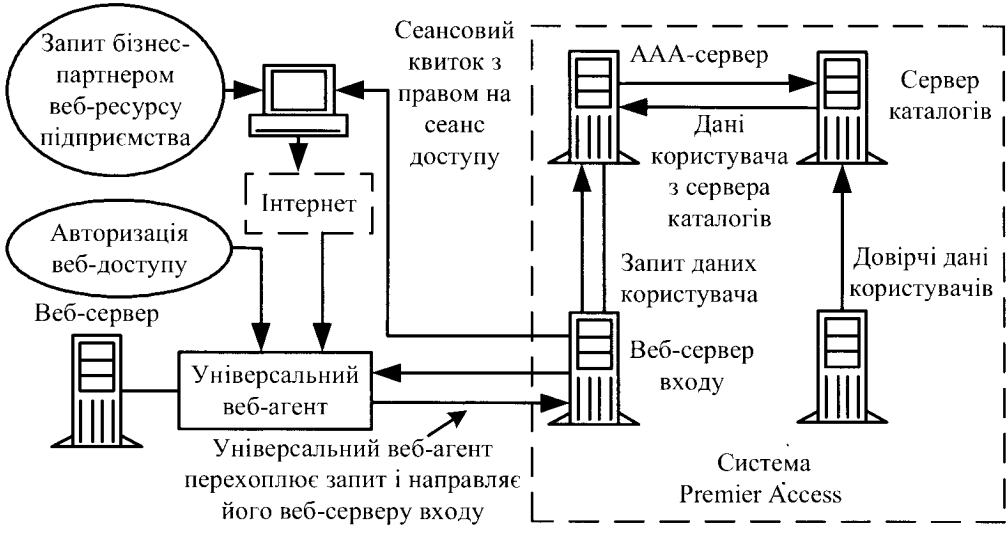


Рис. 6.29. Схема керування доступом за допомогою системи PremierAccess

1. Користувач намагається ввійти в систему, використовуючи ***веб-сервер входу*** (Web Login Server – WLS). Запит користувача на доступ до захищеного веб-ресурсу компанії перехоплює агент UWA, який для оброблення цього запиту звертається до сервера WLS. Сервер WLS запитує результат автентифікації в AAA-сервера. У разі успішної автентифікації сервер WLS генерує сеансовий квиток (cookie), який містить сеансовий ідентифікатор користувача.

2. Користувач намагається отримати доступ до веб-ресурсу. Сервер WLS використовує сеансовий ідентифікатор у квитку для запиту в AAA-сервера даних сеансу користувача. Щоб виконати запит на доступ, сервер WLS передає користувачеві сеансовий квиток із правами на сеанс. Агент UWA отримує

сеансовий ідентифікатор, потім отримує від AAA-сервера дані сеансу. Ґрунтуючись на ролях користувача та політиці доступу, він приймає рішення, давати чи заборонити користувачеві доступ до веб-ресурсу.

Під час побудови систем керування доступом важливе значення мають:

– засоби та протоколи автентифікації віддалених користувачів;

– засоби керування доступом за схемою ***одноразового входу з авторизацією*** (Single Sign-On);

– інфраструктура відкритих ключів PKI.

Контрольні питання до розділу 6

1. Структура каналу електрозв'язку.
2. Види каналів зв'язку.
3. Засоби несанкціонованого доступу до інформації в абонентських телефонних лініях.
4. Методи виявлення та боротьби із засобами несанкціонованого доступу до інформації на абонентських телефонних лініях у робочому стані.
5. Засоби несанкціонованого доступу до інформації, що використовують радіолінії.
6. Виявлення засобів несанкціонованого доступу до інформації, що використовують радіолінії.
7. Безпека телекомунікаційних систем та мереж.
8. Підходи до забезпечення збереження та захисту інформації в телекомунікаційних системах та мережах.
9. Як здійснюють у стандарті GSM процедуру захисту від несанкціонованого доступу?
10. Як здійснюють у стандарті GSM процедуру захисту від підслуховування в радіоканалі?
11. Як вирішують у стандарті GSM проблему конфіденційності інформації про місцезнаходження абонента?
12. У чому полягають особливості забезпечення конфіденційності інформації про місцезнаходження абонента в системах CDMA?
13. Як здійснюють у стандарті GSM процедуру захисту від використання незареєстрованої апаратури?
14. На які групи поділяють пристрої системи GSM за виконуваними функціями? Назвіть функції кожної групи.
15. Назвіть основні рівні ієрархічної просторової структури GSM та їх функціональне призначення.
16. Опишіть види інформаційно-комунікаційних мереж.
17. Опишіть характерні особливості інтранет-мереж та інтернет-мереж.
18. Назвіть види інцидентів, які згідно із запропонованою Конвенцією Ради Європи класифікацією можуть завдати шкоди комп'ютерним мережам.
19. Назвіть найважливіші принципи, на яких ґрунтується організація служби безпеки комп'ютерних мереж.
20. Опишіть суть та способи здійснення процедури автентифікації в комп'ютерних мережах.
21. Як забезпечують конфіденційність повідомлень у комп'ютерних мережах?
22. Як забезпечують цілісність повідомлень у комп'ютерних мережах?
23. Вимоги політики безпеки щодо захисту приватної мережі від внутрішніх та зовнішніх загроз.
24. Назвіть найпоширеніші види атак на ресурси комп'ютерної мережі.

25. Опишіть характерні особливості DoS- атаки на ресурси комп'ютерної мережі.
26. Охарактеризуйте методи моніторингу відкритих і відносно відкритих джерел та соціальної інженерії.
27. Опишіть використання бот-мережі для організації атак на ресурси комп'ютерної мережі.
28. Назвіть основні функції систем виявлення атак.
29. Опишіть методи, які використовують у системах виявлення атак.
30. Назвіть три діапазони IP-адрес, які не обробляють маршрутизатори.
31. Як називають сервер-посередник між приватною й зовнішніми мережами? Опишіть основні функції, які він виконує.
32. Який протокол забезпечує прямий і зворотний переходи пакетів з інтранет-мережі в інтернет-мережу?
33. Опишіть три види трансляції мережевих адрес протоколом NAT.
34. Які типи брандмауерів є найпоширенішими?
35. Опишіть функції лінійного шлюзу мережевого рівня.
36. Опишіть найпоширеніші функції брандмауерів прикладного рівня.
37. Опишіть структуру мережевого екрана на базі виділеного сервера й зовнішнього маршрутизатора.
38. Опишіть побудову та функції захищеного каналу між користувачами різних мереж.
39. Опишіть побудову та функції захищеного каналу між двома приватними мережами через відкриту проміжну мережу.
40. Опишіть основні функції протоколів захищених каналів, які функціонують на прикладному рівні.
41. Опишіть особливості функціонування протоколів, які використовують для побудови захищених каналів на мережевому рівні.
42. Опишіть призначення та принцип побудови віртуальної приватної мережі.
43. Опишіть особливості формування криптозахищених тунелів на канальному рівні.
44. Назвіть протоколи та опишіть особливості формування криптозахищених тунелів на мережевому рівні.
45. Назвіть особливості будови та функціонування сховища даних.
46. Охарактеризуйте системи зберігання даних, які найчастіше використовують у сховищах даних.
47. Яким вимогам політики безпеки повинен відповідати периметр інформаційного захисту сховища даних?
48. Назвіть базові задачі, які необхідно розв'язати для надійного захисту сховища даних.
49. Опишіть функції виділеного проксі-сервера, який виконує роль посередника між серверами системи зберігання даних і комп'ютерами користувачів.
50. Які вимоги до авторизації та автентифікації користувачів висуває політика безпеки сховища даних?
51. Як здійснюють моніторинг подій та менеджмент на рівні баз даних сховища?
52. Як забезпечують гнучку багатошарову модель безпеки сховища даних?
53. Опишіть вимоги, які висуває політика безпеки до резервного копіювання та архівування даних у сховищі даних.
54. Що є основними компонентами IPSec?
55. Які алгоритми шифрування використовує IPSec?
56. Як функціонують системи керування доступом?
57. Як працює система керування відкритими ключами?

Список літератури до розділу 6

1. Про телекомунікації [Електронний ресурс] : закон України № 1280-IV : [прийнятий Верховною Радою України 18 листопада 2003 р. : редакція від 10 серпня 2012 р.]. – Режим доступу : http://zakon.rada.gov.ua/go/1280-15.
2. Горбатий І. В. Технічна експлуатація сучасних комплексів зв'язку : навч. посіб. / І. В. Горбатий, О. В. Тимченко. – Львів : Сполом, 2006. – 244 с.
3. Горбатий І. В. Математичні моделі та методи дослідження телекомунікаційних каналів : монографія / І. В. Горбатий. – Львів : СПОЛОМ, 2006. – 156 с.
4. Горбатий І. В. Телекомунікаційні системи та мережі. Принципи функціонування, технології та протоколи : навч. посіб. / І. В. Горбатий, А. П. Бондарєв. – Львів : Видавництво Львівської політехніки, 2016. – 336 с.
5. Дудикевич В. Б. Захист засобів і каналів телефонного зв'язку : навч. посіб. / В. Б. Дудикевич, В. В, Хома, Л. Т. Пархуць. – Львів : Видавництво Львівської політехніки, 2012. – 212 с.
6. Гарасимчук О. І. Комплексні системи санкціонованого доступу : навч. посіб. / О. І. Гарасимчук, В. Б. Дудикевич, В. А. Ромака. – Львів : Видавництво Львівської політехніки, 2010. – 212 с.
7. Телекомунікаційні системи та мережі. Структура й основні функції. Том 1 [Електронний ресурс] / В. В. Поповський, О. В. Лемешко, В. К. Ковальчук та ін. – Режим доступу : http://www.znanius.com/3533.html/
8. Бондарєв А. П. Пристрої цифрових систем стільникового зв'язку : навч. посіб. / А. П. Бондарєв, Б. А. Мандзій, С. В. Давіденко. – Львів : Видавництво Львівської політехніки, 2011. – 222 с.
9. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов / В. Г. Олифер, Н. А. Олифер. – 4-е изд. – СПб. : Питер, 2014. – 944 с.
10. Таненбаум Э. Компьютерные сети / Э. Таненбаум, Д. Уэзеролл. – СПб. : Питер, 2012. – 960 с.
11. Куроуз Дж., Росс К. Компьютерные сети: многоуровневая архитектура Интернета / Дж. Куроуз, К. Росс. – 2-е изд. – СПб. : Питер, 2004. – 765 с.
12. Ирвин Дж. Передача даних в сетях: инженерний подход : [пер. с англ.] / Дж. Ирвин, Д. Харль. – СПб. : БХВ-Петербург, 2003. – 448 с.
13. Компьютерные сети и сетевые технологии : [пер. с англ.] / М. Спортак, Ф. Паппас и др. – К. : ООО "ТИД "ДС", 2002. – 736 с.
14. Буров Є. Комп'ютерні мережі / Є. Буров. – Львів: СП "Бак", 2006. – 416 с.
15. Березюк Б. М. Системи і мережі передавання даних : навч. посіб. / Б. М. Березюк. – Серія "Дистанційне навчання". № 34. – Львів : Вид-во Національного університету "Львівська політехніка", 2005. – 200 с.
16. Бурячок В. Л. Інформаційна та кібербезпека: соціотехнічний аспект: підручник / В. Л. Бурячок, В. Б. Толубко, В. О. Хорошко, С. В. Толюпа; за заг. ред. д-ра техн. наук, професора В. Б. Толубка. – К. : ДУТ, 2015. – 288 с.
17. Грищук Р. В. Основи кібернетичної безпеки : монографія / Р. В. Грищук, Ю. Г. Даник; за заг. ред. проф. Ю. Г. Даника. – Житомир : ЖНАЕ, 2016. – 636 с. : іл.
18. Грайворонський М. В. Безпека інформаційно-комунікаційних систем / М. В. Грайворонський, О. М. Новіков. – К. : Видавнича група BHV, 2009. – 608 с.
19. Юдін О. К. Захист інформації в мережах передачі даних : підручник МОН України / О. К. Юдін, Г. Ф. Конахович, О. Г. Корченко. – К. : Видавництво DIRECTLINE, 2009. – 714 с.

20. Радченко М. М. Аналіз систем виявлення вторгнень та комп'ютерних атак / М. М. Радченко, О. І. Іванов, С. І. Прохорський, К. К. Мужеський / Междисциплинарные исследования в науке и образовании. – 2013. – 379 с.
21. Шаньгин В. Ф. Защита компьютерной информации. Эффективные методы и средства / В. Ф. Шаньгин. – М. : ДМК Пресс, 2010. – 544 с.
22. Ленков С. В. Методы и средства защиты информации : в 2 т. / С. В. Ленков, Д. А. Перегудов, В. А. Хорошко. – К. : Арий, 2008. – Т. 2: Информационная безопасность, 2008. – 344 с.
23. Яковина В. С. Основи безпеки комп'ютерних мереж: навч. посіб. / В. С. Яковина, Д. В. Федасюк; за ред. Д. В. Федасюка. – Львів : НВФ "Українські технології", 2008. – 396 с.
24. Пасічник В. В. Сховища даних : навч. посіб. / В. В. Пасічник, Н. Б. Шаховська. – Львів : "Магнолія 2006", 2008. – 492 с.
25. Ramachandran V. BackTrack 5 Wireless Penetration Testing / V. Ramachandran. – Packt Publishing, 2011. – 207 p.
26. Microsoft TCP / IP : учебний курс : [пер. с англ.]. – М. : Издательский отдел "Русская Редакция" ТОО "Channel Trading Ltd.", 1998. – 392 с.

Розділ 7

ЗАХИСТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

7.1. Актуальність

Захист ***програмного забезпечення (ПЗ)*** загалом можна визначити як комплекс заходів, спрямованих на захист ПЗ від несанкціонованого придбання, використання, поширення, модифікування, вивчення й відтворення аналогів.

На сучасному етапі розвитку інформаційних технологій процеси збирання, накопичення, оброблення, передавання та зберігання інформації здійснюють переважно з використанням ***інформаційних систем (ІС)***. Деструктивні дії на інформацію в процесі функціонування таких систем мають на меті порушення її конфіденційності, цілісності та доступності. Оскільки інформація є експлуатованим ресурсом для ПЗ ІС, ***безпека ПЗ*** ІС є важливою складовою безпеки інформації загалом. Питанням безпеки ПЗ в ІС присвячено, зокрема, низку робіт [6–10], матеріали яких використані під час написання цього розділу.

7.2. Безпека програмного забезпечення

Під час вирішення завдання забезпечення безпеки інформаційних ресурсів ІС виходять із припущення, що найімовірнішим інформаційним об'єктом деструктивних дій в ІС буде ПЗ. Особливу увагу при цьому необхідно звернути на безпеку ПЗ ***критичних ІС***. Критичними вважають ІС, що забезпечують функціонування об'єктів державного управління й контролю, оборони, фінансового обігу, комунікації, енергетики, житлово-комунального господарства, промисловості й транспорту. Блокування або порушення функціонування таких систем можуть призвести до порушення чи втрати державного управління й контролю, обороноздатності, руйнування фінансового обігу та енергетичного й комунікаційно-транспортного забезпечення, глобальних екологічних і техногенних катастроф.

Під безпекою ПЗ розуміють властивість певного ПЗ функціонувати без прояву негативних наслідків для конкретної ІС. При цьому ***рівень безпеки ПЗ*** у

процесі його експлуатації визначають як імовірність забезпечення ***функціональної придатності*** (suitability) ІС. Функціональну придатність визначено, своєю чергою, стандартом ISO 9126:2001 як здатність розв'язувати потрібний набір задач.

До загальних причин, що призводять до зниження рівня безпеки ПЗ, зараховують:

– збої ІС;

– помилки програмістів і операторів;

– дефекти ПЗ.

Дефекти ПЗ умовно поділяють на ненавмисні та навмисні. Ненавмисні дефекти ПЗ є, як правило, результатом помилкових дій людини, навмисні дефекти ПЗ – результатом зловмисних дій. Для усунення невизначеності стосовно дефектів ПЗ припускають, що вони завжди є результатом навмисних дій.

У загальному випадку, дослідження проблем безпеки ПЗ, пов'язаних із потенційною можливістю наявності в ньому навмисних дефектів, передбачає вирішення таких завдань:

– визначення кола суб'єктів, які потенційно можуть здійснити практичне впровадження дефектів у ПЗ;

– визначення можливих мотивів дій суб'єктів, що створюють та впроваджують такі дефекти;

– визначення способів виявлення та ідентифікації дефектів ПЗ;

– визначення найімовірніших наслідків активізації дефектів ПЗ.

7.3. Життєвий цикл програмного забезпечення

Прагнення розробників до підвищення якості ПЗ зумовило необхідність визначення його життєвого циклу як сукупності окремих етапів робіт, які виконують у заданому порядку протягом періоду часу, що починається з вирішення питання про розроблення ПЗ і закінчується припиненням його використання. Такий підхід надав можливість організовувати оптимальне управління розробленням та контролювати якість на кожному етапі.

У загальному випадку ***життєвий цикл ПЗ*** представляють такими базовими етапами:

– системний аналіз і обґрунтування вимог до ПЗ;

– попереднє (ескізне) і детальне (технічне) проектування ПЗ;

– розроблення програмних компонентів, їх об'єднання та відлагодження ПЗ у цілому;

– випробовування, дослідна експлуатація та тиражування ПЗ;

– регулярна експлуатація ПЗ, підтримування експлуатації та аналіз її результатів;

– супровід ПЗ, його модифікація й удосконалення, створення нових версій.

У ході еволюційного розвитку теорії проектування ПЗ виникло декілька основних моделей життєвого циклу:

– каскадна;

– ітераційна;

– спіральна.

Каскадна модель. Першою за часом появи й найпоширенішою стала ***каскадна модель життєвого циклу ПЗ*** (рис. 7.1), яка характеризується такими основними особливостями:

– послідовним виконанням етапів;

– завершенням кожного попереднього етапу до початку наступного;

– відсутністю (або певним обмеженням) повернення до попередніх етапів;

– наявністю результату тільки в кінці розроблення.

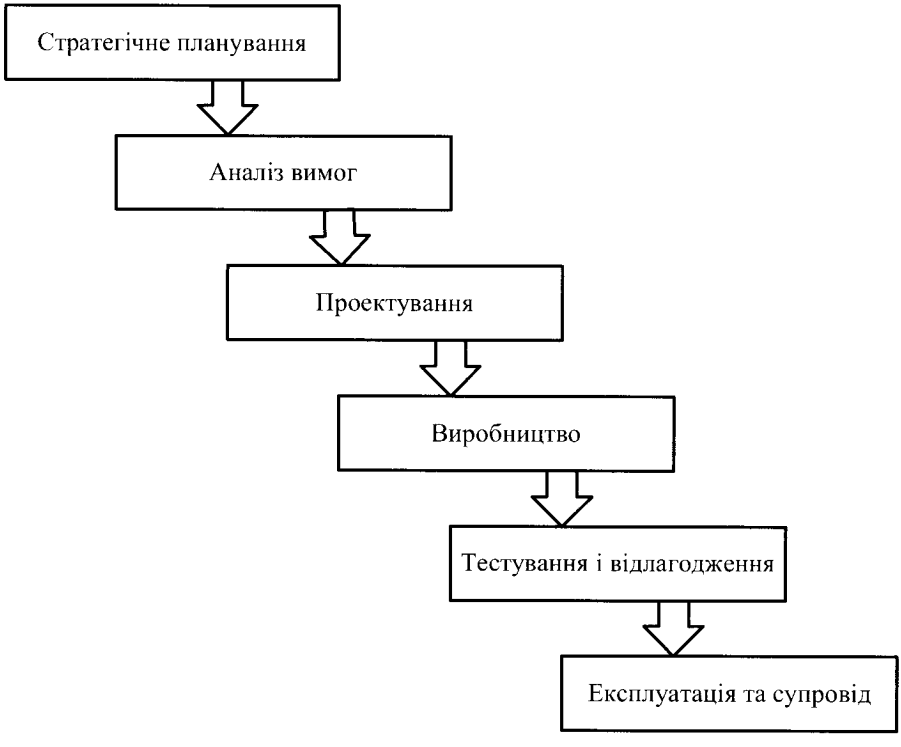


Рис. 7.1. Каскадна модель життєвого циклу ПЗ

Виявляють та усувають помилки згідно з каскадною моделлю лише на етапі тестування й відлагодження, який може розтягнутися в часі або взагалі ніколи не завершитися.

Ітераційна модель. Наступним кроком розвитку теорії проектування програмного забезпечення стала ***ітераційна модель життєвого циклу ПЗ***, зображена на рис. 7.2.

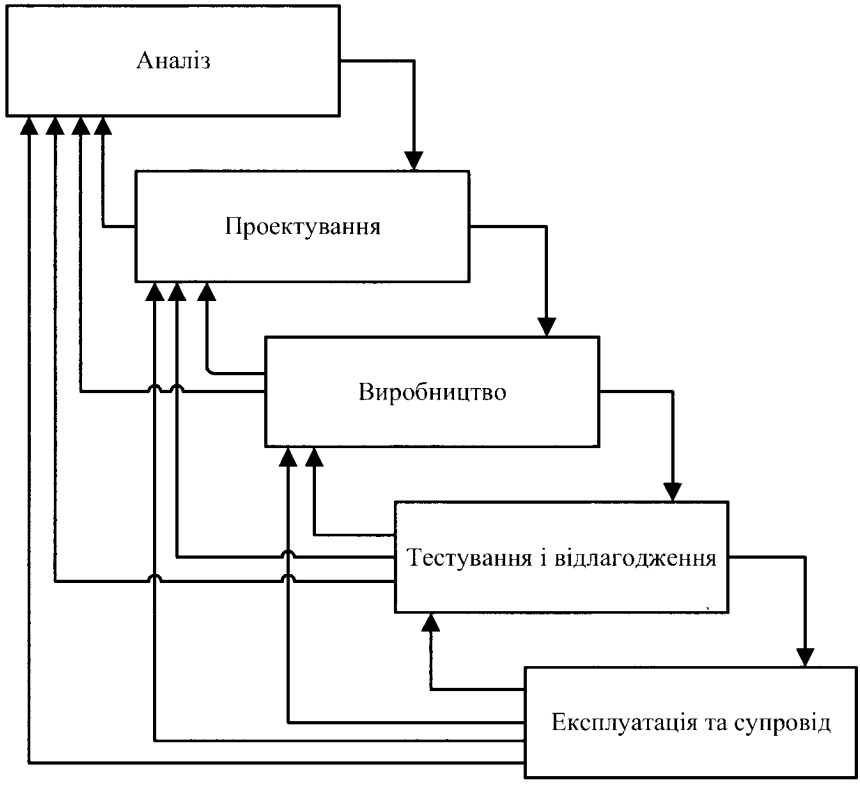


Рис. 7.2. Ітераційна модель життєвого циклу ПЗ

Основною особливістю ітераційної моделі, відомої також як "поетапна модель із проміжним контролем", є наявність зворотних зв'язків між етапами, внаслідок чого з'являється можливість здійснення перевірок і коригувань проектованого ПЗ на кожній стадії розроблення. У результаті трудомісткість відлагодження порівняно з каскадною моделлю істотно знижується. Однак за зміни початкових вимог до ПЗ під час його розроблення ітераційна модель може виявитися неефективною.

Спіральна модель. ***Спіральна модель життєвого циклу ПЗ*** (рис. 7.3) підтримує ітераційний підхід, властивий ітераційній моделі, проте особливу

увагу приділяють початковим етапам проектування: аналізу вимог, проектування специфікацій, попереднього проектування та детального проектування. Кожен виток спіралі відповідає ітераційній моделі створення фрагмента або версії ПЗ, при цьому уточнюють цілі та вимоги до ПЗ, оцінюють якість розробленого фрагмента або версії й планують роботи для наступної стадії розроблення (витка). Так поглиблюють і конкретизують усі деталі проектованого ПЗ, у результаті чого отримують продукт, який задовольняє всі початкові вимоги замовника.

На практиці часто застосовують комбіновану модель життєвого циклу. За основу беруть "спіральну" модель, в якій із секторами спіралі, що є етапами процесу, зіставляють відповідні етапи каскадної моделі, якщо такі є.

На цей час розроблено низку інших моделей життєвого циклу ПЗ, які стали поширеними. Серед них можна назвати, наприклад, ***модель раціонального уніфікованого процесу*** (Rational Unified Process – RUP) фірми Rational Software, що входить до складу корпорації IBM.

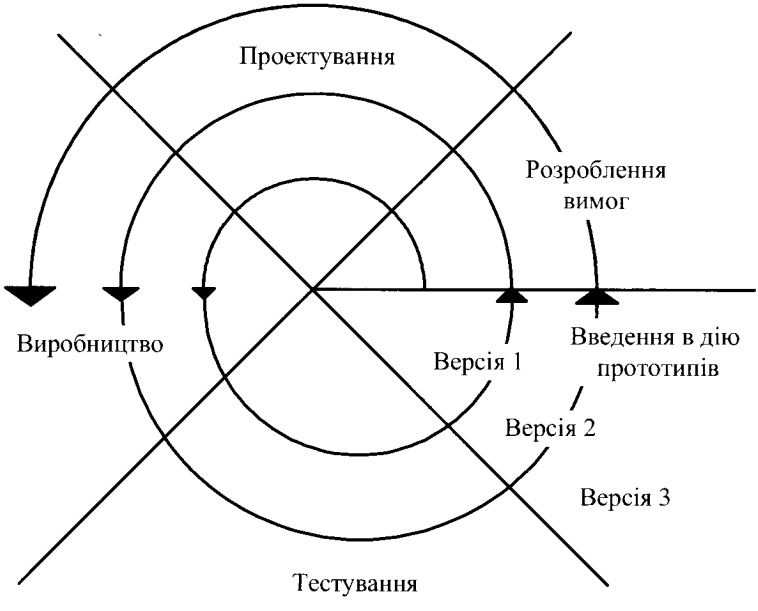


Рис. 7.3. Спіральна модель життєвого циклу ПЗ

Сукупність етапів розглянутих моделей життєвого циклу ПЗ умовно поділяють на дві частини, що істотно відрізняються особливостями процесів, техніко-економічними характеристиками й чинниками, що впливають на них. У першій частині, яка отримала назву "технологічна", здійснюють системний аналіз, обґрунтування вимог, проектування, розроблення, тестування, випробо-

вування, дослідну експлуатацію та тиражування ПЗ. Друга частина – "експлуатаційна" – стосується підтримання експлуатації й супроводу ПЗ. Відповідно до цього поділу вирізняють:

– ***технологічну безпеку ПЗ*** – властивість ПЗ не бути навмисно зміненим і (або) обладнаним шкідливими компонентами на технологічній частині життєвого циклу;

– ***експлуатаційну безпеку ПЗ*** – властивість ПЗ не бути навмисно зміненим і (або) обладнаним шкідливими компонентами на експлуатаційній частині життєвого циклу.

7.4. Загрози безпеці програмного забезпечення

7.4.1. Загальна характеристика

До загроз безпеці ПЗ належать зокрема:

– незаконне розповсюдження та збут ПЗ (піратство);

– незаконне використання алгоритмів (порушення авторського права на інтелектуальну власність);

– несанкціонована модифікація ПЗ (упровадження навмисних дефектів);

– несанкціоноване використання ПЗ (копіювання).

Ураховуючи особливості використання ІС, при забезпеченні їхньої безпеки основну увагу приділяють вирішенню питань запобігання несанкціонованій модифікації та несанкціонованому використанню відповідного ПЗ.

Сьогодні основну загрозу безпеці ПЗ ІС несуть такі програмні засоби деструктивної дії на ІС, які за своєю природою мають, як правило, руйнівний характер – ***руйнівні програмні засоби (РПЗ)***. Це зокрема:

– комп'ютерні віруси;

– закладки;

– способи й засоби, що дають змогу впроваджувати комп'ютерні віруси й закладки в ІС і керувати ними на відстані, зокрема, за допомогою атак на ІС.

Слід зазначити, що необхідною умовою віднесення програмного засобу деструктивної дії до класу РПЗ є наявність у ньому процедури нападу, яку можна визначити як процедуру порушення цілісності ***обчислювального середовища***, оскільки об'єктом нападу РПЗ завжди є елемент цього середовища. Під обчислювальним середовищем розуміють сукупність установлених для такої ІС алгоритмів використання системних ресурсів, програмного та інформаційного забезпечення, яка потенційно може бути надана користувачеві для розв'язання прикладних задач. Частину обчислювального середовища, надану користувачеві для вирішення конкретної задачі, називають ***операційним середовищем***.

Можливість програмної дії на обчислювальне середовище зумовлена його відкритістю – користувач може формувати елементну базу обчислювального середовища під свої завдання й використовувати в повному обсязі системні ресурси.

Для опису основних класів РПЗ застосовують узагальнену концептуальну модель, яка передбачає наявність у їх складі такого набору функцій:

– захоплення управління;
– самовідтворення;
– самомодифікація;
– маскування.

7.4.2. Комп'ютерні віруси

Однією із найсуттєвіших і найпоширеніших загроз безпеці ПЗ ІС на цей час є комп'ютерні віруси. Різноманітність їхніх видів і модифікацій та швидкий темп еволюції ускладнюють формулювання вичерпного визначення поняття "комп'ютерний вірус". Переважно під ***комп'ютерним вірусом*** розуміють РПЗ, що функціонує автономно та має здатність до самостійного впровадження в тіла інших програм із подальшим самовідтворенням і саморозповсюдженням в ІС та окремих комп'ютерах. При цьому вважають, що достатньою умовою для віднесення РПЗ до класу комп'ютерних вірусів є наявність у його складі процедури самовідтворення. Ця особливість відрізняє комп'ютерні віруси від їх попередників – троянських програм. ***Троянські програми*** не здатні до самовідтворення, маскуються під широко відомі програми масового застосування й містять приховані фрагменти, що виконують шкідливі дії.

Вирізняють такі стадії життєвого циклу комп'ютерного вірусу:

– латентна стадія, на якій вірус жодних дій не здійснює;
– інкубаційна стадія, на якій основна задача вірусу – створити якомога більше своїх копій і впровадити їх у середовище перебування;
– активна стадія, на якій вірус, продовжуючи розмноження, проявляється й виконує свої деструктивні дії.

При цьому початок активної стадії може бути зумовлений такими подіями:

– настанням певної дати;
– запуском програми;
– відкриванням документа тощо.

Структурно комп'ютерний вірус складається з голови й, можливо, хвоста. Головою вірусу називають його частину, що отримує управління, хвостом – частина, розташована в тексті інфікованої програми окремо від голови. Віруси,

що складаються з однієї голови, називають несегментованими, а віруси, що містять голову й хвіст – сегментованими.

Під час інфікування вірус вбудовується в програму так, щоб під час її запуску отримати управління першим. Отримавши управління, вірус виконує властиві йому цільові функції, наприклад, інфікує інший файл, можливо, виконує які-небудь інші дії, після чого віддає управління інфікованій програмі. Інфіковану вірусом програму називають ***вірусоносієм***.

Комп'ютерні віруси класифікують за їхніми найістотнішими ознаками.

За режимом функціонування розрізняють:

– ***резидентні віруси*** – віруси, які після активізації постійно знаходяться в оперативній пам'яті до моменту вимкнення або перезавантаження комп'ютера й контролюють доступ до його ресурсів;

– ***нерезидентні віруси*** – віруси, які не інфікують пам'ять комп'ютера і є активними протягом обмеженого часу.

За об'єктом впровадження бувають:

– ***файлові віруси*** – віруси, що інфікують файли із програмами;

– ***завантажувальні віруси*** – віруси, що інфікують програми, які зберігаються в системних областях дисків, наприклад, у завантажувальному секторі системного диска;

– ***файлово-завантажувальні віруси*** інфікують як файли, так і завантажувальні сектори дисків;

– ***мережеві віруси*** – поширюються в інформаційних мережах, використовуючи для свого розповсюдження команди, протоколи й програмне забезпечення таких мереж (наприклад, електронної пошти).

Своєю чергою, файлові віруси поділяють на віруси, що інфікують:

– виконувані файли;

– командні файли й файли конфігурації;

– файли, створені макромовами програмування, або файли, що можуть містити макрокоманди (наприклад, файли Microsoft Office);

– файли із драйверами пристроїв;

– файли з бібліотеками вихідних, об'єктних, завантажувальних і оверлейних модулів, бібліотеками динамічного компонування тощо.

Завантажувальні віруси поділяють на віруси, що інфікують:

– системний завантажувач, розташований у завантажувальному секторі дискет і логічних дисків;

– позасистемний завантажувач, розташований у завантажувальному секторі жорстких дисків.

Серед мережевих вірусів окремо виділяють клас вірусів-самореплікаторів – ***мережевих хробаків (мережевих черв'яків)***. Мережеві хробаки не інфікують ПЗ безпосередньо, а поширюють свої копії інформаційними системами із

метою проникнення на комп'ютер-жертву, запуску своєї копії на комп'ютері й подальшого розповсюдження, завдаючи шкоди завдяки споживанню пропускної здатності або, можливо, видаленню файлів чи надсиланню документів електронною поштою. Часто мережеві хробаки можна використати для транспортування інших шкідливих РПЗ до вузлів мережі. До таких РПЗ належать зокрема ***логічні бомби*** – програми, які запускаються за певних часових чи інформаційних умов для здійснення шкідливих дій, спрямованих, як правило, на порушення цілісності, конфіденційності та доступності інформації.

За ступенем і способом маскування комп'ютерні віруси поділяють на:

– віруси, що не використовують засобів маскування;

– ***стелс-віруси*** (віруси-невидимки) – віруси, що намагаються бути невидимими на основі контролю доступу до інфікованих елементів, приховуючи себе за спроби виявлення;

– ***віруси-мутанти*** – віруси, що містять алгоритми шифрування, які забезпечують відмінність зашифрованих копій вірусу, чим унеможливлюють пошук вірусів за сигнатурами – характерними послідовностями байтів у фрагментах коду вірусів. Структурно вірус-мутант складається із зашифрованого тіла та шифрувальної частини.

Серед вірусів-мутантів вирізняють:

– ***звичайні віруси-мутанти***, копії яких відрізняються лише зашифрованими тілами;

– ***поліморфні віруси***, копії яких відрізняються як зашифрованими тілами, так і шифрувальними частинами.

Здійснення вірусами цільових функцій спричиняє ефекти, які поділяють на такі групи:

– порушення цілісності файлової системи або окремих файлів;

– ініціація помилок у системному або прикладному ПЗ;

– імітація збоїв апаратних засобів;

– створення візуальних і звукових ефектів.

Тому основними ознаками прояву функціонування вірусів є:

– непередбачувана втрата працездатності комп'ютера або його компонентів;

– неможливість завантаження операційної системи;

– часті зависання та збої комп'ютера;

– непередбачуване сповільнення роботи комп'ютера;

– суттєве зменшення обсягу доступної вільної оперативної пам'яті;

– порушення цілісності даних у CMOS (Complementary Metal-Oxide-Semiconductor) пам'яті комп'ютера;

– непередбачуване форматування логічних та фізичних дисків;

– порушення цілісності файлів, каталогів чи файлової системи загалом;
– непередбачуване збільшення кількості файлів;
– непередбачувана зміна розмірів файлів;
– непередбачувана зміна атрибутів фалів, дати й часу їх модифікації;
– порушення працездатності прикладних програм;
– непередбачуване зниження пропускної здатності каналів зв'язку в ІС;
– поява непередбачуваних повідомлень, зображень та звукових сигналів, що можуть ввести користувача в оману або утруднити його роботу.

7.4.3. Алгоритмічні та програмні закладки

Окрім вірусів, до основних засобів деструктивної дії на ІС належать алгоритмічні та програмні закладки.

Під ***алгоритмічною закладкою*** розуміють навмисну, приховану зміну частини алгоритму програми або побудову його так, що в результаті програмного здійснення цього алгоритму за певних умов проходження обчислювального процесу можлива поява нових або змінювання вже передбачених специфікацією ПЗ функцій.

Під ***програмною закладкою*** розуміють навмисно, приховано привнесені в програмне забезпечення функційні об'єкти, які при певних умовах протікання обчислювального процесу ініціюють виконання непередбачених специфікацією ПЗ функцій. Класичним прикладом програмних закладок є троянські програми.

Дії алгоритмічних та програмних закладок на інформацію, що обробляється в ІС, умовно поділяють на три класи, які можуть перетинатися між собою. Це, зокрема:
– зміна функціонування ІС;
– несанкціоноване читання інформації;
– несанкціоноване змінювання інформації (як даних, так і ПЗ), аж до її знищення.

Зміну функціонування ІС супроводжують такі прояви:
– зменшення швидкості роботи ІС;
– часткове або повне блокування роботи ІС;
– імітація апаратних збоїв;
– обхід криптографічних програмно-апаратних засобів захисту;
– переадресація повідомлень;
– забезпечення доступу в систему з несанкціонованих периферійних пристроїв.

Другий клас дій полягає у:
– перехопленні паролів та їх ототожненні з користувачами;

– підміні паролів;

– отриманні секретної інформації;

– ідентифікації інформації користувачів;

– контролі активності користувачів інформаційних систем для отримання непрямих даних про їх взаємодію й характер інформації, якою вони обмінюються.

Несанкціоноване змінювання інформації є найнебезпечнішим різновидом дії алгоритмічних та програмних закладок, оскільки, загалом, може призвести до найбільш негативних наслідків. У цьому класі дій, зокрема, можна виокремити:

– внесення прихованих змін до інформаційних масивів;

– руйнування даних і кодів ПЗ;

– упровадження програмних закладок в інше ПЗ;

– змінювання пакетів повідомлень.

Програмні закладки можуть бути впроваджені в ПЗ на різних етапах його життєвого циклу. Враховуючи це, їх умовно поділяють на дві категорії:

– апріорні (природжені);

– апостеріорні (набуті).

Апріорні закладки привносять на етапі розроблення ПЗ, а тому вони стосуються питань забезпечення технологічної безпеки, ***апостеріорні закладки*** – на етапах випробування, експлуатації або модернізації ПЗ, тому вони більше стосуються проблеми забезпечення експлуатаційної безпеки ПЗ. Однак слід зазначити, що методи визначення наявності програмних закладок і методи оцінювання рівня безпеки ПЗ для обидвох категорій значною мірою є спільними або доповнюють один одного. Окрім того, дія програмної закладки, привнесеної до ПЗ на технологічній частині його життєвого циклу, практично не відрізнятиметься від дії програмної закладки, впровадженої до ПЗ на експлуатаційній частині.

7.5. Захист програмного забезпечення від загроз

7.5.1. Експлуатаційна безпека програмного забезпечення

Забезпечення безпеки ПЗ ІС насамперед залежить від вирішення питань експлуатаційної безпеки, оскільки на експлуатаційній частині життєвого циклу таке ПЗ є найбільш вразливим до привнесення дефектів, а спричинене цими дефектами ймовірне припинення чи непередбачуване порушення його штатного функціонування можуть призвести до значних негативних наслідків.

Забезпечення експлуатаційної безпеки ПЗ ІС тісно пов'язане з такими поняттями як ***"модель порушника"*** та ***"модель загроз"***.

Модель порушника. Модель порушника, з погляду концепції захисту інформації, є абстрактним (формалізованим чи неформалізованим) описом порушника, що визначає, зокрема, категорії порушників, які можуть діяти на об'єкт.

При вирішенні завдань, пов'язаних із забезпеченням експлуатаційної безпеки ПЗ ІС, зокрема, оцінюванні ризиків, аналізі вразливості, ефективності існуючих і планових заходів захисту, необхідно керуватися моделлю порушника, наведеною в [5]. Цей нормативний документ розглядає порушника як особу, яка може отримати доступ до роботи із засобами, що входять до складу ІС. Порушників класифікують за рівнем можливостей, що надаються їм штатними засобами ІС. Вирізняють чотири рівні таких можливостей. Класифікація є ієрархічною, тобто кожний наступний рівень містить функційні можливості попереднього. Припускають, що на своєму рівні порушник – це фахівець вищої кваліфікації, який має повну інформацію про ІС і комплекс її засобів захисту.

Модель загроз. Модель загроз експлуатаційній безпеці ПЗ є фізичним, математичним чи описовим представленням властивостей чи характеристик загроз експлуатаційній безпеці ПЗ у вигляді офіційно прийнятого нормативного документа. Після побудови такої моделі розробляють практичні рекомендації й методики з її використання для конкретного об'єкта захисту, а також механізми оцінювання адекватності моделі й ефективності її застосування. Отже, розроблення моделі загроз експлуатаційній безпеці ПЗ є важливою складовою вирішення проблеми забезпечення безпеки ІС загалом.

Модель загроз експлуатаційній безпеці ПЗ будують на основі аналізу загроз, притаманних цій частині життєвого циклу ПЗ. Такі загрози поділяють на:

– прямі та непрямі;

– випадкові та навмисні;

– активні та пасивні.

Прямі загрози характеризують безпосереднім деструктивним впливом на ПЗ, а непрямі – опосередкованим.

Випадкові загрози мають імовірнісний характер і пов'язані з відмовами та збоями апаратури, помилками операторів тощо. Навмисні загрози, натомість, спричиняє, як правило, зловмисне бажання суб'єкта несанкціоновано порушити цілісність, конфіденційність та доступність інформації.

Активні загрози спрямовані на змінювання зумовлених специфікацією ПЗ алгоритмів і функційних перетворень або інформації, над якою ці перетворення здійснюють. Пасивні загрози орієнтовані на порушення безпеки ПЗ без здійснення таких змін.

Можливий варіант загальної структури набору потенційних загроз експлуатаційній безпеці ПЗ ІС наведено в табл. 7.1.

Таблиця 7.1

Можливий варіант загальної структури набору потенційних загроз експлуатаційній безпеці ПЗ ІС

Загрози	Несанкціоновані дії		
	Випадкові	Навмисні	
		Пасивні	Активні
Прямі	Невиявлені помилки ПЗ ІС; відмови та збої технічних засобів ІС; помилки операторів; несправність засобів шифрування; стрибки електроживлення на технічних засобах; старіння носіїв інформації; руйнування інформації під впливом фізичних чинників (аварії тощо)	Маскування несанкціонованих запитів під запити операційної системи; обхід програм розмежування доступу; читання конфіденційних даних із джерел інформації; підключення до каналів зв'язку з метою отримання інформації під час аналізу трафіку; використання терміналів та комп'ютерів інших операторів	Включення в ПЗ РПЗ; Впровадження в обчислювальне середовище нових програм, що виконують функції порушення безпеки ПЗ; незаконне застосування ключів розмежування доступу; обхід засобів розмежування доступу; виведення з ладу підсистеми реєстрації й обліку; знищення ключів шифрування й паролів; підключення до каналів зв'язку з метою змінювання, знищення, затримування й перевпорядкування даних; виведення з ладу елементів фізичних засобів захисту інформації ІС; навмисний виклик випадкових чинників
Непрямі	Порушення пропускного режиму й режиму секретності; природні потенційні поля; завади тощо	Перехоплення електромагнітного випромінювання від технічних засобів; викрадення виробничих відходів (роздруківок тощо); візуальний канал; підслуховуючі пристрої; дистанційне фотографування тощо	Завади; відключення електроживлення; навмисний виклик випадкових чинників

Основні принципи забезпечення експлуатаційної безпеки програмного забезпечення. До основних принципів забезпечення експлуатаційної безпеки ПЗ належать, зокрема:

– створення стратегії наскрізного тотального контролю ПЗ на експлуатаційній частині його життєвого циклу;

– постійний, комплексний та ефективний контроль за користувачами ПЗ;

– забезпечення конфіденційності комплексу заходів щодо забезпечення експлуатаційної безпеки ПЗ;

– організування збереження еталонів ПЗ, обмеження доступу до них і недопущення їх змінювання;

– профілактичне вибіркове та повне тестування й сканування ПЗ на наявність навмисних дефектів;

– ідентифікація ПЗ на момент введення його в експлуатацію відповідно до передбачуваних загроз безпеці ПЗ та його контроль;

– забезпечення механізму модульного змінювання ПЗ без зміни його загальної структури й зв'язків з іншим ПЗ;

– суворий облік і каталогізація всього ПЗ;

– статистичний аналіз інформації про всі процеси, робочі операції, відхилення від режимів штатного функціонування ПЗ;

– гнучке та оперативне застосування додаткових засобів захисту ПЗ у разі виявлення нових, непрогнозованих загроз.

7.5.2. Адаптивна безпека інформаційних систем

Однією з основних загроз безпеці ІС є атака на неї, а саме, будь-яка дія, яку виконує порушник для здійснення загрози внаслідок використання вразливостей ІС. Під ***вразливістю ІС*** розуміють будь-яку характеристику або елемент ІС, використання яких порушником може призвести до здійснення загрози.

Як приклад розглянемо технологію атаки на ІС підприємства (організації). Архітектура такої системи містить чотири рівні:

– рівень ПЗ, що відповідає за взаємодію з користувачем;

– рівень системи управління базами даних (СУБД), що відповідає за зберігання й оброблення даних в ІС; до елементів ІС, що працюють на цьому рівні, можна зарахувати СУБД Oracle, MS SQL Server, Sybase і MS Access;

– рівень операційної системи, що відповідає за обслуговування СУБД і прикладного ПЗ;

– рівень мережі, що відповідає за взаємодію вузлів ІС; елементами ІС, що працюють на цьому рівні, можна вважати стеки протоколів TCP / IP, IPX / SPX і SMB / NetBIOS.

Зловмисник має широкий спектр можливостей для порушення безпеки ІС підприємства (організації). Ці можливості можна використати на всіх чотирьох

перерахованих вище рівнях. Наприклад, для отримання несанкціонованого доступу до інформації в СУБД MS SQL Server зловмисник може використати одну з таких можливостей:

– перехопити передані мережею дані (рівень мережі);

– прочитати файли бази даних, звертаючись безпосередньо до файлової системи (рівень операційної системи);

– прочитати потрібні дані засобами самої СУБД (рівень СУБД);

– прочитати записи бази даних за допомогою SQL-запитів через програму MS Query, яка дає змогу отримувати доступ до записів СУБД (рівень прикладного ПЗ).

Розглянемо етапи здійснення атаки на ІС підприємства (організації).

Перший, підготовчий, етап полягає в пошуку зловмисником передумов для здійснення тієї чи іншої атаки. На цьому етапі зловмисник шукає вразливості системи.

Початково, за допомогою відповідного спеціалізованого ПЗ, зловмисник збирає інформацію про підприємство (організацію) – власника ІС: її веб-сторінку, виділені діапазони IP-адрес, структуру мережі, операційні системи серверів тощо (рис. 7.4–7.6). Додатково, за можливості, копіює веб-сторінку підприємства (організації) з метою аналізу її вмісту й відслідковування повідомлень електронної пошти.

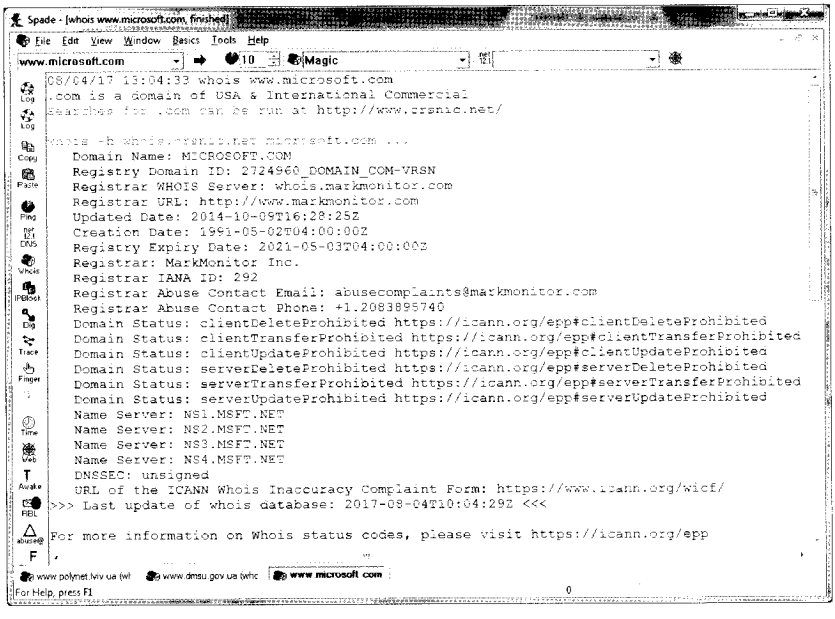


Рис. 7.4. Збирання інформації за допомогою програми Sam Spade

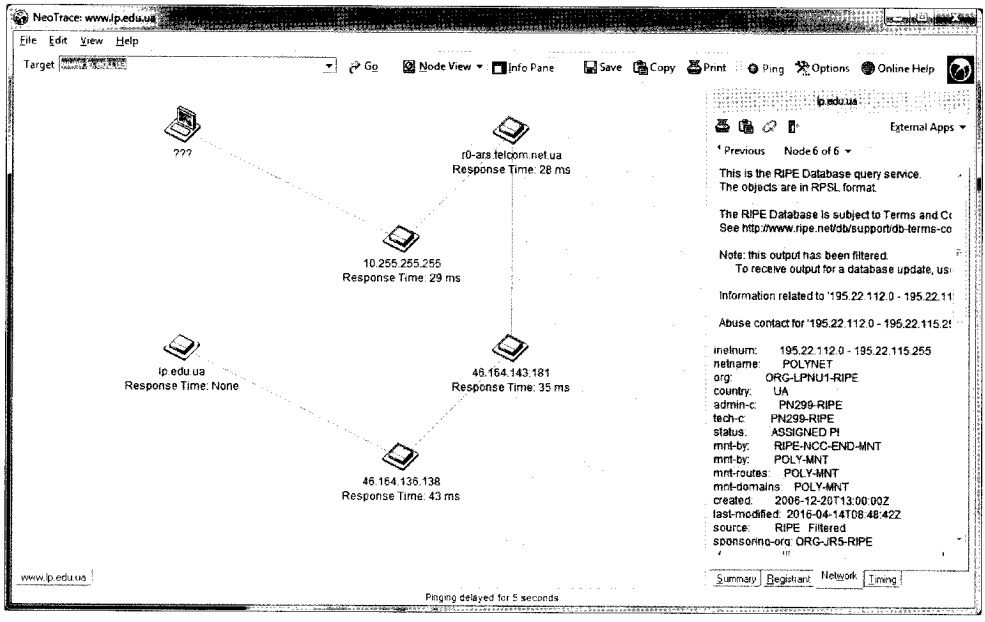


Рис. 7.5. Збирання інформації про структуру мережі за допомогою програми NeoTrace

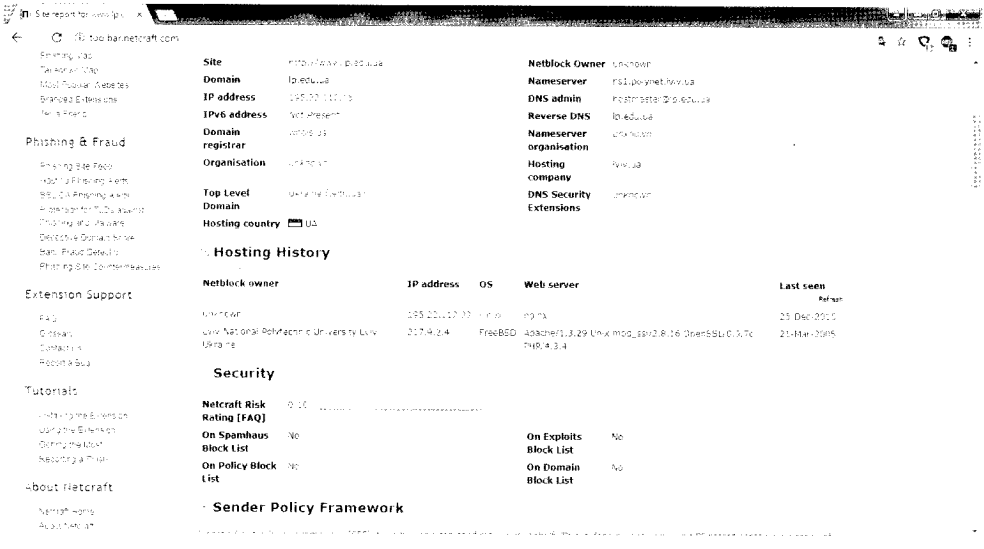


Рис. 7.6. Визначення операційної системи сервера за допомогою програми Netcraft

Далі зловмисник вибирає вузли-цілі атак і сканує їхні порти з метою виявлення відкритих, а також аналізує трафік (рис. 7.7).

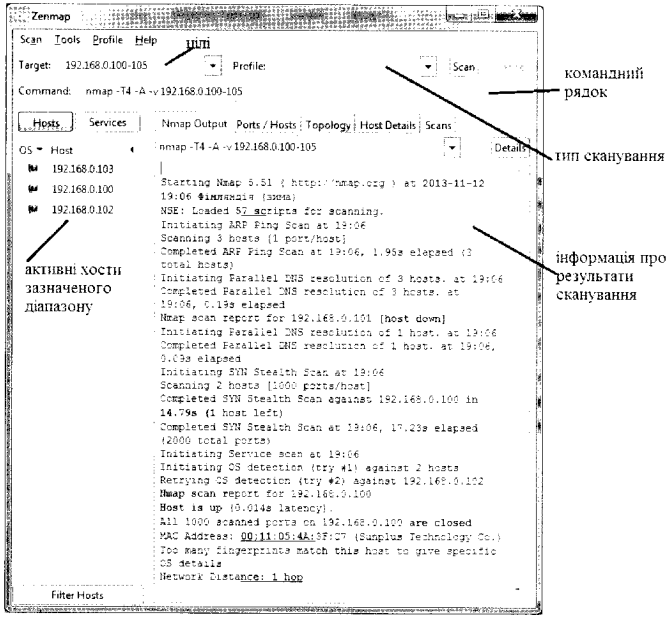


Рис. 7.7. Сканування портів за допомогою утиліти Zenmap

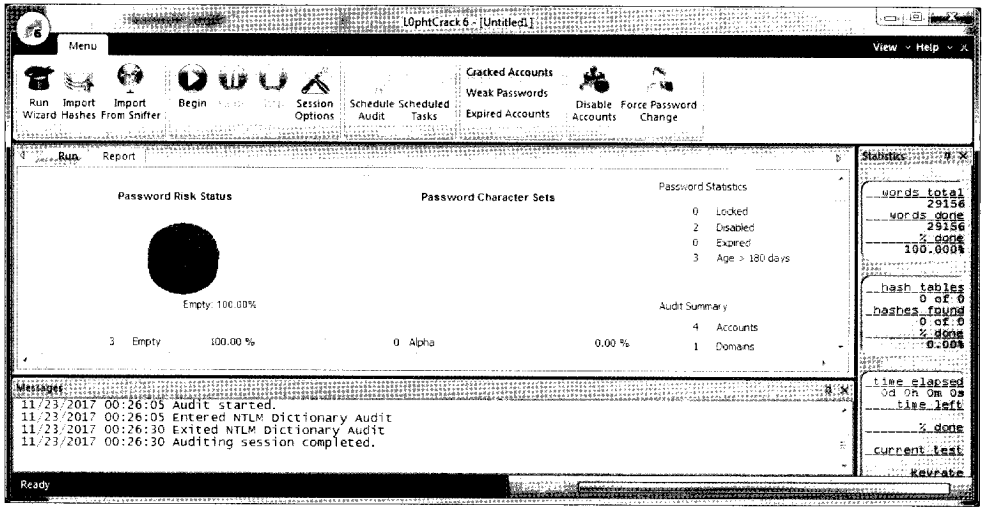


Рис. 7.8. Зламування паролів за допомогою пакета L0phtCrack Password Auditor

На другому, основному, етапі (етапі здійснення атаки) зловмисник використовує знайдені вразливості. Це може бути "затоплення" пакетами (наприклад, за допомогою утиліти macof), ***ARP-спуфінг*** (мережева атака, коли зловмисник надсилає підроблені повідомлення протоколу ARP (Address Resolution Protocol) у локальну мережу), зламування паролів (див. рис. 7.8) тощо.

На третьому, завершальному, етапі зловмисник завершує атаку й прагне приховати сліди вторгнення.

Слід зазначити, що перший і третій етапи самі по собі можуть бути атаками. Наприклад, пошук зловмисником вразливостей за допомогою сканерів безпеки вважають атакою. Також треба враховувати, що існуючі механізми захисту, здійснені в міжмережевих екранах, серверах автентифікації, системах розмежування доступу, працюють тільки на етапі здійснення атаки. По суті, ці механізми захищають від атак, які перебувають уже в процесі здійснення.

На рис. 7.9 наведено приклад виявлення атак та запобігання деяким їх видам.

Рис. 7.9. Оснастка netcraft для запобігання фішингу

Ефективнішою технологією захисту було б випередження атак на ІС, тобто запобігання самим передумовам здійснення вторгнення. Система інформаційної безпеки повинна ефективно працювати на всіх трьох етапах здійснення атаки.

На практиці часто не беруть до уваги той факт, що адміністратори й користувачі постійно змінюють конфігурацію ІС. У результаті цих змін можуть з'являтися нові вразливості, пов'язані з операційною системою й ПЗ. Окрім того, відбуваються дуже швидкі зміни інформаційних та мережевих технологій, постійно з'являється нове ПЗ. Безперервний розвиток мережевих технологій за відсутності постійного аналізу їхньої безпеки й нестача ресурсів для забезпечення захисту призводять до того, що із часом захищеність ІС зменшується, оскільки з'являються нові невраховані загрози та вразливості системи.

У більшості випадків для вирішення проблем із захистом використовують часткові підходи, що зумовлено насамперед поточним рівнем доступних ресурсів. Окрім того, адміністратори безпеки мають тенденцію реагувати лише на зрозумілі їм ризики безпеки. Насправді таких ризиків може бути істотно більше. Отже, тільки суворий поточний контроль захищеності ІС та комплексний підхід, що забезпечує єдину політику безпеки, дають змогу істотно знизити ризики безпеки.

Розглянемо адаптивний підхід до безпеки, який дає можливість контролювати, виявляти ризики й реагувати на них у реальному режимі часу, використовуючи правильно спроектовані й добре керовані процеси та засоби.

Адаптивна безпека мережі складається із трьох основних елементів:

– ***технології управління ризиками*** (risk management);

– ***технології аналізу захищеності*** (security assessment);

– ***технології виявлення атак*** (intrusion detection).

Оцінювання ризику полягає у виявленні та ранжуванні вразливостей (за ступенем серйозності шкоди потенційних впливів), підсистем мережі (за ступенем критичності), загроз (виходячи з імовірності їх здійснення). Оскільки конфігурацію мережі постійно змінюють, то й оцінювати ризик потрібно постійно. З оцінювання ризиків повинна починатися побудова системи захисту ІС.

Аналіз захищеності – це пошук уразливих місць у мережі. Мережа складається з ліній зв'язку, вузлів (хост, host), робочих станцій, ПЗ тощо. Усі вони потребують як оцінювання ефективності їх захисту, так і пошуку невідомих вразливостей у них. За допомогою технологій аналізу захищеності досліджують мережу й шукають "слабкі" місця в ній, узагальнюють ці відомості й формують відповідний звіт. Якщо система, що використовує цю технологію, містить і адаптивний компонент, то усувають знайдену вразливість не вручну, а автоматично. Технологія аналізу захищеності є дієвим методом, що дає змогу упровадити політику мережевої безпеки, перш ніж буде здійсненоа спробу її порушити зовні або зсередини підприємства.

До проблем, які потенційно можуть бути виявлені за допомогою технології аналізу захищеності, належать:

– "люки" в системах (back door) і програми типу "троянський кінь";

– слабкі паролі;

– сприйнятливість до проникнення з незахищених систем і до атак типу "відмова в обслуговуванні" (denial-of-service-DoS);

– відсутність необхідних оновлень операційної системи;

– неправильне налаштування мережевих екранів, веб-серверів тощо.

Виявлення атак є процесом оцінювання підозрілих дій, які відбуваються в мережі. Виявляють атаки за допомогою аналізу журналів реєстрації операційної

системи й програмного забезпечення або аналізу мережевого трафіку в реальному часі. Компоненти виявлення атак, розміщені на вузлах або в сегментах мережі, оцінюють різні події та дії, зокрема дії, що використовують відомі вразливості. До програмних продуктів, призначених для виявлення таких атак, належать IBM Security Network, OSSEC, Peek&Spy, Cisco Intrusion Prevention Systems, INTOUCH INSA-Network Security Agent, Advanced Intrusion Detection Enviroment, KFSensor, SilverSky, SNARE і багато інших. На рис. 7.10 зображено головне вікно програми KFSensor, призначеної для виявлення атак на вузол.

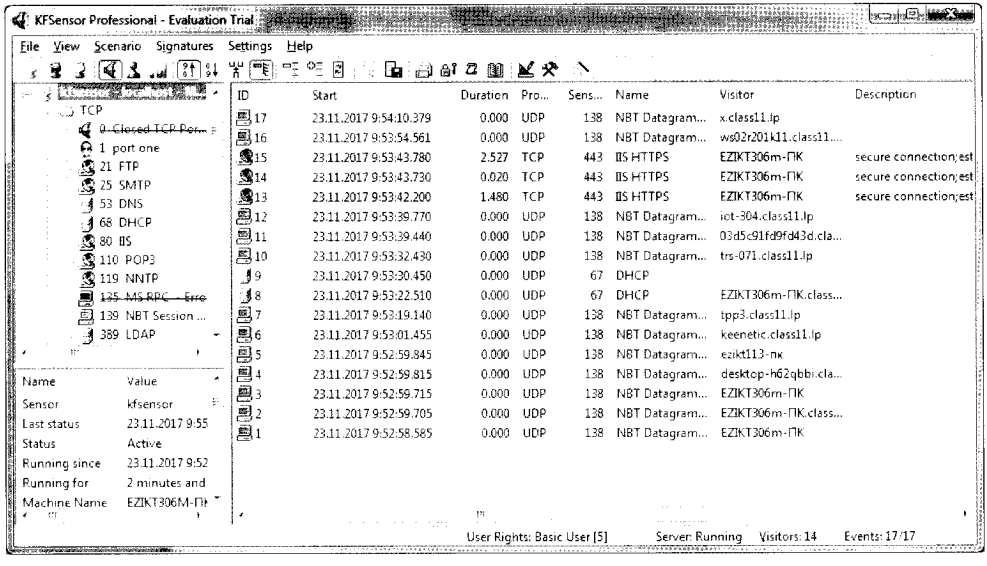


Рис. 7.10. Головне вікно програми KFSensor, призначеної для виявлення атак на вузол

На основі адаптивного підходу до безпеки компанією ISS (англ. *Internet Security Systems*) було створено ***модель адаптивного управління безпекою*** ANS (Adaptive Network Security). Адаптивний компонент такої моделі відповідає за вдосконалення процесу аналізу захищеності, надаючи йому найсвіжішу інформацію про нові вразливості. Він також модифікує компонент виявлення атак, доповнюючи його аналогічною інформацією про нові види атак. Прикладом адаптивного компонента може бути механізм оновлення баз даних антивірусних програм для виявлення нових вірусів.

Керівний компонент формує звіти та аналізує тенденції, пов'язані із налаштуванням системи захисту ІС.

Адаптація даних може полягати в різних формах реагування, зокрема:

– відправлення повідомлень системам мережевого управління за протоколом SNMP (Simple Network Management Protocol – простий прото-

кол мережевого керування) електронною поштою або SMS-повідомленнями адміністратору;

– автоматичне завершення з'єднання з атакуючим вузлом або користувачем, переналаштування міжмережевого екрана чи інших мережевих пристроїв (наприклад, маршрутизаторів);

– вироблення рекомендацій адміністратору, що дають змогу своєчасно усунути виявлені вразливості в мережах, ПЗ чи інших компонентах ІС.

Використання моделі адаптивної безпеки мережі (рис. 7.11) дає можливість контролювати практично всі загрози і своєчасно ефективно реагувати на них, що дає змогу не лише усунути вразливості, які можуть призвести до здійснення загрози, а й проаналізувати умови, що призводять до появи вразливостей. Така модель дає можливість зменшити зловживання в мережі, підвищити обізнаність користувачів, адміністраторів і керівництва підприємства (організації) про події безпеки в мережі.

Рис. 7.11. Модель адаптивної безпеки мережі

Модель адаптивної безпеки мережі не відкидає вже використовувані механізми захисту (розмежування доступу, автентифікація тощо), а розширює їх функційність завдяки новим технологіям. Для приведення своїх систем забезпечення інформаційної безпеки у відповідність до сучасних вимог підприємствам (організаціям) необхідно доповнити наявні рішення компонентами, що відповідають за аналіз захищеності, виявлення атак і управління ризиками.

7.5.3. Юридичний та технічний захист програмного забезпечення

Юридичний захист програмного забезпечення. ***Юридичний захист ПЗ*** в ІС ґрунтується на відповідній правовій базі, зокрема, законах [3, 4] і спрямований переважно на охорону авторських прав власників інтелектуальної власності, зокрема ПЗ. Існують такі форми юридичного захисту:

– цивільно-правовий;

– адміністративний;

– кримінальний.

Цивільно-правовий захист юридично забезпечує власника авторських прав на ПЗ при виявленні факту їх порушення правом:

– вимагати визнання та поновлення своїх прав і заборони дій, що порушують авторське право чи створюють загрозу його порушення;

– звертатися до суду з позовом щодо поновлення порушених прав та припинення дій, які порушують авторське право чи створюють загрозу його порушення;

– подавати позови про відшкодування моральної та матеріальної шкоди, завданих порушенням авторського права;

– вимагати припинення підготовчих дій до порушення авторського права;

– вимагати, зокрема в судовому порядку, публікації в засобах масової інформації даних про допущені порушення авторського права та судові рішення щодо цих порушень;

– вимагати від осіб, які порушують авторське право на ПЗ, надання інформації про третіх осіб, задіяних у цьому порушенні, а також про засоби подолання технічних заходів захисту певного ПЗ;

– вимагати прийняття інших передбачених законодавством заходів, пов'язаних із захистом авторського права.

Адміністративний захист ПЗ забезпечує стаття 51^2 Кодексу України про адміністративні правопорушення [1], відповідно до якої за незаконне використання ПЗ передбачено відповідальність у вигляді штрафу.

Кримінальний захист ПЗ в інформаційних системах регулюють статті 176, 361, 361-1, 362-2, 362, 363, 363-1 Кримінального кодексу України [2], які передбачають покарання штрафом, або виправними роботами, або позбавленням волі.

Технічний захист ПЗ. ***Технічний захист ПЗ*** полягає в забезпеченні безпеки ПЗ технічними засобами. Його поділяють на:

– програмний захист;

– апаратний захист;

– програмно-апаратний захист.

Програмний захист ПЗ – це захист, здійснений програмно. Перевагами програмного захисту можна вважати універсальність, гнучкість, простоту встановлення, здатність до модифікації й розвитку, відносну дешевизну, а до недоліків – високу чутливість до випадкових або навмисних змін компонентів, використання частини ресурсів обчислювального середовища та можливу залежність від його складу. Слід зазначити, що будь-який програмний захист прийнято вважати таким, що може бути подоланим за обмежений проміжок часу, оскільки процесорний код системи захисту в момент свого виконання присутній в програмний код комп'ютера у відкритому вигляді. Тому першочерговим завданням програмного захисту є максимально можливе утруднення виявлення, дослідження та (або) модифікації механізму захисту.

Апаратний захист ПЗ використовує спеціалізоване апаратне обладнання і забезпечує високу надійність захисту, проте порівняно із програмним дорожчий при здійсненні.

Програмно-апаратний захист ПЗ є комбінацією програмного та апаратного захистів.

7.5.4. Захист програмного забезпечення від комп'ютерних вірусів

Методи захисту програмного забезпечення від комп'ютерних вірусів. Технологія захисту ПЗ ІС від комп'ютерних вірусів ґрунтується на створенні комплексної, багаторівневої, розподіленої системи захисту, що передбачає такі заходи організаційно-технічного характеру:

– регламентування виконання робіт;

– застосування спеціальних програмних засобів;

– використання спеціальних апаратних засобів.

Кількість необхідних рівнів захисту залежить від цінності оброблюваної в ІС інформації. На перших рівнях доцільно використовувати засоби захисту від деструктивних дій РПЗ, а на наступних – засоби виявлення комп'ютерних вірусів і, нарешті, засоби їх нейтралізації.

До ***методів захисту ПЗ від комп'ютерних вірусів*** зараховують:

– архівування – регулярне створення резервних копій ПЗ та зберігання архівів змінених файлів;

– вхідний контроль – перевіряння новоотриманого ПЗ спеціальними програмними та апаратними засобами;

– профілактика – розмежування доступу до ПЗ та мінімізація періодів його доступності;

– ревізія – періодичне планове перевіряння ПЗ спеціальними програмними та апаратними засобами;

– карантин – перевіряння протягом певного проміжку часу новоотриманого ПЗ на відомі типи вірусів на ізольованій ІС.

– сегментування – розташування ПЗ у захищених від спроб запису зонах окремо від оброблюваних даних;

– фільтрування – використання спеціальних програмних засобів, що виявляють спроби виконання несанкціонованих дій;

– вакцинування – оброблення файлів та їхніх носіїв із метою імітації ознак, які використовують деякі віруси, для уникнення повторного інфікування;

– автоконтроль цілісності – використання алгоритмів, здатних у момент запуску ПЗ виявити порушення його цілісності;

– терапія – деактивування конкретного вірусу в інфікованому ПЗ спеціальними програмними засобами.

Необхідно зазначити, що архівування вважають основним і найнадійнішим методом захисту ПЗ від комп'ютерних вірусів. Інші методи є допоміжними й дають можливість підвищити рівень захисту.

Засоби захисту програмного забезпечення від комп'ютерних вірусів. Залежно від функційних особливостей ***програмні засоби захисту ПЗ від комп'ютерних вірусів***, відомі за загальною назвою ***"антивірусне програмне забезпечення"***, поділяють на такі групи:

– детектори (сканери);

– фаги (лікарі);

– ревізори;

– фільтри (монітори);

– вакцини (імунізатори);

– евристичні аналізатори.

Детектори виявляють віруси пошуком в оперативній пам'яті та файлах сигнатур – стійких послідовностей байтів, що є характерними для відомих на цей час вірусів. Антивірусну програму, що здатна шукати різні сигнатури, називають полідетектором.

Фаги є детекторами з розширеною функційністю. Окрім пошуку вірусу, фаги здатні їх деактивувати або видаляти з тіла вірусоносія. Фаги, орієнтовані на нейтралізацію різних вірусів, називають поліфагами.

Сигнатурний метод виявлення та неперервна поява нових типів і модифікацій вірусів зумовлюють необхідність регулярного оновлення детекторів та фагів.

Ревізори вважають найнадійнішими засобами захисту від вірусів. Ревізори запам'ятовують початковий стан програм, каталогів і системних

областей диска тоді, коли комп'ютер не інфікований вірусом, а потім періодично або за бажанням користувача порівнюють поточний стан із вихідним. Виявлені зміни одразу ж фіксують. Під час порівняння перевіряють довжину файла, контрольну суму файла, дату й час змінювання, інші параметри. Сучасні ревізори мають достатньо розвинені алгоритми, зокрема виявляють стелс-віруси й навіть здатні відрізнити зміну версії програми від змін, внесених вірусом.

Існує різновид ревізорів, які впроваджуються в програму, яку захищають, та зберігають ряд її кількісних і структурних характеристик. Під час повторних запусків захищеної програми такий ревізор виконує перевіряння відповідності характеристик, які були збережені, аналогічним характеристикам, отриманим на поточний момент. Якщо набори характеристик не збігаються, роблять висновок про наявність змін у програмному коді захищеної програми.

Фільтри є резидентними програмами, які контролюють ті виклики операційної системи, які використовують віруси для розмноження й нанесення шкоди, і повідомляють про них користувача. Користувач може дозволити або заборонити виконання відповідної операції. У разі заборони або відсутності підтвердження фільтр блокує виконання програми користувача.

Перевагою фільтрів є можливість виявлення та блокування вірусів на ранній стадії активності до виконання ними деструктивних дій.

Вакцини використовують певну особливість функціонування вірусів: для уникнення багатократного повторного інфікування, що призводить до істотного збільшення розміру інфікованого файла, яке легко виявити, віруси, як правило, певним чином позначають інфіковані ними файли. Вакцини модифікують об'єкт захисту, імітуючи такі позначки. Віруси після активізації й перевіряння наявності вказаних позначок вважають такий об'єкт вже інфікованим і не інфікують його повторно.

Сьогодні, коли кількість вірусів вимірюють мільйонами, вакцини мають обмежене застосування. Їх найчастіше застосовують для захисту від вірусів, які не можуть бути нейтралізовані існуючими фагами.

Евристичні аналізатори перевіряють програмні об'єкти, намагаючись виявити в них програмний код, характерний для вірусів. Евристичний аналізатор може виявити, наприклад, що програма, яку перевіряють, намагається встановити резидентний модуль в оперативній пам'яті або записує дані у виконуваний файл іншої програми.

Перевагою евристичних аналізаторів є можливість виявлення нових, раніше не відомих вірусів, причому для цього не треба попередньо отримувати інформацію про стан файлової системи, як цього вимагають, наприклад,

ревізори. Слід зазначити, що практично всі сучасні антивірусні програми використовують власні методи евристичного аналізу.

Загальні рекомендації із захисту програмного забезпечення інформаційних систем від комп'ютерних вірусів. Для підвищення рівня захисту ПЗ від комп'ютерних вірусів необхідно:

– установлювати антивірусне ПЗ у режимі відключення від локальної мережі та мережі Інтернет (режим "offline ").

– оперативно оновлювати антивірусне ПЗ та його бази даних;

– увімкнути в антивірусному ПЗ режим фільтра;

– регулярно виконувати антивірусний контроль внутрішніх носіїв інформації та встановленого на них ПЗ;

– при використанні зовнішніх носіїв інформації та встановленого на них ПЗ обов'язково здійснювати їх антивірусний контроль;

– без необхідності не вимикати в зовнішніх носіях режим захисту від запису;

– не залишати зовнішні носії підключеними до комп'ютера під час його вмикання чи вимикання;

– регулярно оновлювати операційну систему з метою зменшення кількості притаманних їй вразливостей;

– не використовувати неліцензійне або неперевірене ПЗ;

– не використовувати ПЗ, призначення якого невідоме або незрозуміле;

– не відвідувати сайти із сумнівною чи скомпрометованою репутацією;

– не завантажувати файли з невідомих чи сумнівних джерел;

– не відкривати без необхідності та без додаткового підтвердження з боку відправника вкладені файли електронних листів;

– регулярно створювати резервні копії ПЗ та здійснювати їх антивірусний контроль.

7.5.5. Захист програмного забезпечення від упровадження програмних закладок

На експлуатаційній частині життєвого циклу ПЗ використовують три основні способи впровадження програмних закладок. Перший спосіб полягає в дисасемблюванні ПЗ та змінюванні отриманого асемблерного коду внесенням до його складу програмних закладок із подальшим компілюванням зміненого коду й замінюванням ним оригіналу. Другий спосіб полягає в отриманні вихідних текстів ПЗ і внесенні до них програмних закладок із подальшим компілюванням модифікованого коду й замінюванням ним оригіналу. Третій

спосіб полягає в підміні виконуваних програм їх повними функційними аналогами, які розроблені порушниками й уже містять програмні закладки.

Найпоширеніший перший спосіб впровадження програмних закладок. Практичне здійснення другого способу утруднене необхідністю мати доступ до вихідних текстів ПЗ, а третього – потребою в повній та точній інформації про цільове призначення й функційність імітованого ПЗ.

Досліджують ПЗ для виявлення програмних закладок зазвичай відтворенням використаних у цьому ПЗ алгоритмів (реверс-інжиніринг ПЗ). Із цією метою застосовують відповідні засоби дослідження, які поділяють на два класи:

– статичні;

– динамічні.

Статичні засоби дослідження ПЗ оперують кодом програми як даними і будують її алгоритм без виконання. ***Динамічні засоби дослідження ПЗ*** аналізують програму, інтерпретуючи її в реальному або віртуальному обчислювальному середовищах. Алгоритм програми будують на підставі аналізу конкретної послідовності команд (траси), виконаних під час такої інтерпретації для певного набору вхідних даних, а задача отримання повного алгоритму програми зводиться до формування вичерпного набору тестів, що, на практиці, або складно, або неможливо. Тому динамічні засоби дослідження дають змогу, загалом, побудувати лише деяку частину алгоритму. Статичні засоби дослідження, натомість, дають можливість повністю реконструювати алгоритм програми, зокрема й тих її частин, які ніколи не виконуються.

Сьогодні широке практичне застосування отримали такі засоби дослідження ПЗ, як ***дисасемблери***, ***декомпілятори*** та ***відлагоджувачі***. Дисасемблери належать до класу статичних. Вони безпосередньо здійснюють представлення досліджуваного алгоритму у вигляді тексту мовою низького рівня – асемблером. Декомпілятори відрізняються від дисасемблерів лише тим, що представляють досліджуваний алгоритм у вигляді тексту мовою високого рівня. Відлагоджувачі, що є динамічними засобами, виконують програму покроково й забезпечують при цьому можливість контролю за значеннями її змінних.

Слід зазначити, що багато сучасних засобів дослідження ПЗ поєднують у собі функційність дисасемблерів, декомпіляторів та відлагоджувачів. Характерним представником такого класу можна вважати інтерактивний дисасемблер-відлагоджувач IDA, який підтримує формати виконуваних файлів багатьох операційних систем, зокрема, DOS, Windows, Linux та Mac OS X.

Дисасемблери, декомпілятори та відлагоджувачі використовують у межах ***методів дослідження ПЗ***, які утворюють, відповідно, дві групи:

– синтаксичні методи;

– статистичні методи.

До першої групи належать методи, що базуються виключно на результатах лексичного, синтаксичного й семантичного аналізу ПЗ. До другої – методи, що використовують інформацію, накопичену в результаті запуску ПЗ для кожного з достатньо великої множини набору вхідних даних.

Розглянуті методи й засоби широко використовують порушники для дослідження ПЗ із метою впровадження до нього програмних закладок та інших видів РПЗ. Забезпечити захист ПЗ від такого дослідження можливо, застосовуючи методи захисту від статичних і динамічних засобів дослідження виконуваного коду програми, розташованого як у файлах, так і в оперативній пам'яті комп'ютера.

У першому випадку захист може базуватися, наприклад, на шифруванні конфіденційної частини (частини, яку захищають від дослідження) програми, а в другому – на блокуванні доступу відлагоджувачів до виконуваного коду програми в оперативній пам'яті. Для запобігання можливості несанкціонованого копіювання виконуваного коду програми з оперативної пам'яті його необхідно знищувати після завершення програми.

Загалом, програма, яку захищають від дослідження, повинна містити такі компоненти:

– ініціалізуючу частину;

– конфіденційну частину;

– деініціалізуючу частину.

Ініціалізуюча частина повинна забезпечити:

– збереження поточних параметрів операційного середовища;

– заборону системних викликів, контроль за якими у програмі, яку захищають, не передбачено;

– завантаження в оперативну пам'ять із подальшим дешифруванням (якщо захист здійснюють за допомогою шифрування) коду конфіденційної частини програми;

– передавання управління конфіденційній частині.

Конфіденційна частина програми призначена для виконання основних цільових функцій і повинна бути захищена від дослідження, наприклад, шифруванням.

Деініціалізуюча частина програми активізується після завершення виконання конфіденційної частини й повинна забезпечити:

– знищення конфіденційної частини виконуваного коду програми в оперативній пам'яті;

– відновлення параметрів операційного середовища;

– виконання операцій, які неможливо було виконати у разі заборони системних викликів;

– звільнення всіх задіяних ресурсів комп'ютера й завершення роботи програми.

Рівень захисту ПЗ від дослідження можна збільшити застосовуванням динамічного шифрування ініціалізуючої та (або) конфіденційної частин, за якого чергові ділянки програми дешифрують перед безпосереднім виконанням, а після виконання дешифрований код програми відразу знищують.

Для підвищення ефективності захисту ПЗ від дослідження динамічними засобами, наприклад, відлагоджувачами, у програмі, яку захищають, передбачають такі додаткові функції:

– контроль цілісності виконуваного коду програми, яку захищають, в оперативній пам'яті;

– контроль обсягу оперативної пам'яті, зайнятої програмою, яку захищають;

– контроль часу виконання окремих частин програми, яку захищають;

– блокування клавіатури на час виконання особливо секретних алгоритмів.

Для перешкоджання процесу реверс-інжинірингу ПЗ за допомогою таких статичних засобів, як дисасемблери та декомпілятори, окрім шифрування, широко застосовують методи ***обфускації***. Процес обфускації полягає в заплутуванні програмного коду й усуненні більшості логічних зв'язків у ньому, тобто в такому його перетворенні, щоб вивчення й змінювання цього коду сторонніми особами були максимально утруднені.

Обфускацію вважають успішною, якщо, по-перше, час, витрачений порушником на розуміння коду обфускованого ПЗ, перевищує час, протягом якого актуальність алгоритму залишається значущою, і, по-друге, вартість ***деобфускації*** (зворотного до обфускації процесу) перевищує вартість самого ПЗ.

Залежно від типу вихідного коду ПЗ, яке обфускують, розрізняють два рівні процесу обфускації:

– низький рівень, коли обфускують асемблерний або бінарний (двійковий) код;

– високий рівень, коли обфускують код, що представлений мовою високого рівня.

Розрізняють такі види обфускації:

– лексична (символьна) обфускація;

– обфускація даних;

– обфускація графа потоку керування.

Лексична обфускація полягає у форматуванні коду програми та зміні його структури так, щоб він став нечитабельним, менш інформативним і важчим для дослідження дисасемблерами й декомпіляторами.

Обфускація даних пов'язана з перетворенням структур даних; вона є сучасним і часто використовуваним методом, хоча й складнішим порівняно з лексичною обфускацією.

Найскладнішою з погляду здійснення, але найстійкішою до спроб зламу є ***обфускація графа потоку управління***, де під графом потоку управління розуміють множину всіх можливих шляхів виконання програми, представлену у вигляді графа.

7.5.6. Захист програмного забезпечення від несанкціонованого копіювання

Існуючі методи програмного захисту ПЗ від несанкціонованого копіювання основані на протидії спробам його відтворення шляхом копіювання та спробам запуску й (або) виконання його незаконних копій.

Системи захисту, що здійснюють такі методи, повинні забезпечувати:

– неможливість копіювання ПЗ із дистрибутивних носіїв стандартними засобами (потребує від порушника ретельного дослідження структури носія за допомогою спеціалізованих програмних або програмно-апаратних засобів);

– неможливість застосування стандартних статичних і динамічних засобів дослідження ПЗ (передбачає високу кваліфікацію порушника та використання або розроблення спеціалізованих засобів дослідження);

– ускладнення процесу вивчення алгоритму розпізнавання індивідуальних параметрів комп'ютера, на якому встановлено ПЗ, і його користувача або аналізу застосовуваних засобів захисту (ускладнює емулювання легального операційного середовища виконання програми, яку захищають).

Для захисту від несанкціонованого копіювання застосовують підходи, які полягають у прив'язуванні ПЗ до параметрів штатного операційного середовища виконання або до деякого унікального ідентифікатора.

У першому випадку встановлене ПЗ під час кожного запуску має виконувати такі дії:

– аналіз операційного середовища та визначення його поточних характеристик;

– перевіряння автентичності середовища виконання порівнянням його поточних характеристик з еталонними;

– блокування подальшої роботи ПЗ за розбіжності поточних характеристик з еталонними.

У другому випадку при встановленні ПЗ формують і зберігають на деякому носієві (наприклад, на жорсткому диску) унікальний ідентифікатор, наявність якого перевіряють під час кожного запуску встановленого ПЗ. За

відсутності або неавтентичності цього ідентифікатора ПЗ блокує можливість свого подальшого виконання. Такий підхід використовують, якщо характеристики апаратно-програмного середовища відсутні в явному вигляді або їх визначення значно вповільнює запуск програм чи знижує зручність їх використання. Основною вимогою до такого унікального ідентифікатора є забезпечення неможливості його копіювання стандартними засобами.

Слід зазначити, що з погляду надійності захисту етап перевіряння автентичності середовища виконання або автентичності унікального ідентифікатора є одним із найуразливіших.

Іншим напрямком захисту ПЗ від копіювання є використання одного з різновидів моделі хмарних обчислень, відомого як "ПЗ на вимогу", тобто надання користувачеві функційності ПЗ (усієї або її частини) у вигляді віддаленого сервісу. При цьому код ПЗ, доступ до якого здійснюють за принципом тонкого клієнта, розташовують та виконують на доступному в глобальній мережі сервері, звідки цей код не можна скопіювати.

Стійкість такого захисту залежить від ступеня захищеності серверів, рівня забезпечення конфіденційності запитів та автентифікації користувачів, цілісності ресурсу тощо.

Методи прив'язування ПЗ до параметрів штатного операційного середовища виконання або до деякого унікального ідентифікатора переважно спрямовані на протидію статичним способам зняття захисту від копіювання.

Для протидії динамічним способам зняття захисту ПЗ від копіювання, серед інших, застосовують такі методи:

– контроль за цілісністю області оперативної пам'яті, зайнятої ПЗ у процесі виконання – дає змогу помітити зміни, внесені до цієї області динамічними засобами дослідження;

– контроль за кількістю вільної оперативної пам'яті – дає можливість виявити наявність сторонніх резидентних модулів;

– контроль за часом виконання окремих частин програми – дає змогу виявити моменти призупинення штатного процесу виконання ПЗ динамічними засобами дослідження.

7.5.7. Захист програмного забезпечення від несанкціонованого доступу

Захист від несанкціонованого доступу є системою заходів, скерованих на протидію несанкціонованому використанню ПЗ. Такий захист загалом можливий із використанням організаційних, юридичних, програмних та програмно-апаратних засобів.

Захист ПЗ від несанкціонованого доступу полягає в автентифікації користувача ПЗ або ІС з отриманням додаткової інформації й можливий за допомогою:

– парольного захисту ПЗ;

– прив'язування ПЗ до конкретної ІС;

– електронних ключів;

– ключових дисків.

У першому випадку додаткову інформацію вводить користувач, у другому – вона міститься в унікальних параметрах ІС користувача, у третьому – зберігається в мікросхемі електронного ключа, а в четвертому – на диску.

Парольний захист ПЗ сьогодні є найпоширенішим завдяки простоті програмного здійснення. Слабким місцем парольного захисту є наявність у захищеному ПЗ фрагмента перевіряння правильності введеного пароля, який може бути піддано відповідному аналізу. Це полегшує пошук правильного пароля, а, за певних умов, дає можливість примусово змінити логіку перевіряння для отримання несанкціонованого доступу до ПЗ.

Переваги парольного захисту ПЗ:

– програмне здійснення;

– достатній рівень захисту від некваліфікованого зловмисника;

– простота використання;

– можливість передавання пароля у вигляді повідомлення;

– відсутність конфліктів із програмними та апаратними засобами;

– простота здійснення;

– низька собівартість.

Недоліки парольного захисту ПЗ:

– недостатня стійкість до зламу;

– можливість перехоплення пароля під час його введення;

– необхідність зберігання пароля та спричинена цим можливість його крадіжки.

Прив'язування ПЗ до конкретної ІС полягає в пошуку її унікальних ознак або штучному встановленні таких ознак. Під час подальших запусків ПЗ ідентифікує ці ознаки та визначає санкціонованість доступу.

Основною вразливістю цього типу захисту є можливість примусового збереження ПЗ після відпрацювання системи захисту з подальшим дослідженням цього захисту та виявленням даних, які використовують для автентифікації користувача.

Переваги захисту з прив'язуванням ПЗ до конкретної ІС:

– програмне здійснення;

– простота використання;

– непомітність для користувача;

Недоліки захисту з прив'язуванням ПЗ до конкретної ІС:

– критичність до змін параметрів ІС;

– недостатня стійкість до зламу при доступі зловмисника до ІС;

– можливість конфліктів із системним ПЗ.

Електронні ключі належать до апаратної частини відповідних програмно-апаратних систем захисту. Їх поділяють за архітектурою на ключі без мікропроцесора та ключі з мікропроцесором.

Ключі першого типу зберігають критичну інформацію, зокрема, ключ дешифрування, у пам'яті і є менш стійкими до зламу, оскільки процедури шифрування / дешифрування виконує програмна частина.

Більшою стійкістю відзначаються ключі другого типу, що містять не лише ключ дешифрування, але й блоки шифрування / дешифрування. У таких системах перехоплення ключа дешифрування ускладнене, оскільки всі процедури виконує апаратна частина. Проте залишається можливість примусового збереження захищеної програми після відпрацьовування системи захисту з подальшим його дослідженням. Також можна використати методи криптоаналізу.

Переваги захисту електронними ключами:

– значне ускладнення несанкціонованого доступу до ПЗ;

– високий рівень автоматизації процесу захисту ПЗ;

– великий вибір таких систем на ринку.

Недоліки захисту електронними ключами:

– необхідність придбання системи захисту та навчання персоналу;

– можливість виникнення проблем сумісності із програмно-апаратними засобами ІС;

– можливість зниження відмовостійкості захищеного ПЗ;

– можливість погіршення масштабованості ІС;

– несумісність електронних ключів різних виробників;

– необхідність зберігання електронного ключа та спричинена цим можливість його крадіжки.

Захист ПЗ від несанкціонованого використання за допомогою ***ключових дисків*** сьогодні непопулярний через моральне старіння. Системи захисту цього типу переважно аналогічні системам з електронними ключами. Основними загрозами для них є перехоплення критичної інформації під час її зчитування та незаконне копіювання ключового диска.

7.6. Оцінювання рівня безпеки програмного забезпечення

Методи, які на цей час використовують для аналізу та оцінювання рівня безпеки ПЗ, можна умовно поділити на дві групи:

– контрольно-випробувальні;

– логіко-аналітичні.

Такий поділ грунтується, насамперед, на принциповій відмінності підходів до оцінювання безпеки ПЗ, яке досліджують. У межах контрольно-випробувальних методів аналізу та оцінювання РПЗ розглядають в аспекті фіксації факту порушення безпечного стану системи. Натомість, логіко-аналітичні методи розглядають РПЗ із погляду доведення факту існування відношення еквівалентності між моделлю досліджуваного ПЗ і моделлю РПЗ.

Перевагою такої класифікації порівняно, наприклад, з поділом на статичні та динамічні методи, можна вважати незалежність від типу засобів аналізу.

Контрольно-випробувальні методи. Критерієм безпеки ПЗ у разі застосування ***контрольно-випробувальних методів*** є факт фіксації під час тестування порушення тих вимог безпеки, які стосуються застосування певного ПЗ відповідно до його специфікації. Тестування здійснюють за допомогою тестових запусків, виконання у віртуальному програмному середовищі, за допомогою символічного виконання тощо.

З погляду об'єкта контролю серед контрольно-випробувальних методів розрізняють:

– методи, за якими контролюють процес виконання ПЗ;

– методи, за якими контролюють зміни в операційному середовищі, спричинені функціонуванням ПЗ.

Такі методи не потребують формального аналізу, дають змогу використовувати наявні технічні та програмні засоби й швидко створювати готові методики.

Типова схема аналізу та оцінювання рівня безпеки ПЗ контрольно-випробувальними методами може мати такий вигляд:

– визначення набору контрольованих параметрів операційного середовища або ПЗ;

– складання програми випробувань;

– виконання програми випробувань;

– перевіряння вимог до безпеки певного ПЗ у відповідному його специфікації середовищі експлуатації у разі запротокольованих дій та змін в операційному середовищі;

– формування висновку про рівень безпеки ПЗ на підставі отриманих результатів.

Перелік контрольованих параметрів на першому етапі залежить від апаратного та програмного забезпечення й досліджуваного ПЗ. На останньому етапі можуть бути використані методи екстраполяції результатів і стохастичні методи.

За використання контрольно-випробувальних методів аналізу та оцінювання рівня безпеки ПЗ найскладніше встановити набір критичних із погляду безпеки параметрів цього ПЗ та відповідного операційного середовища, які визначають методом експертних оцінок. Отриманий у результаті аналізу висновок щодо безпеки ПЗ в умовах обмежених обсягів випробувань, як правило, має імовірнісний характер.

Логіко-аналітичні методи. Критерієм безпеки ПЗ у разі застосування ***логіко-аналітичних методів*** є факт формального доведення еквівалентності моделей досліджуваного ПЗ і РПЗ. Наприклад, якщо моделлю ПЗ є його бінарне представлення, а моделями РПЗ – їхні бінарні сигнатури, то доказ еквівалентності полягає в пошуку сигнатур РПЗ у ПЗ. У загальному випадку використовують формальні моделі, які основані на сукупності ознак, властивих тій чи іншій групі РПЗ.

Типова схема аналізу та оцінювання рівня безпеки ПЗ логіко-аналітичними методами має такий вигляд:

– вибирання способів представлення та отримання моделей ПЗ та РПЗ;

– побудова моделі досліджуваного ПЗ;

– доведення факту існування відношення еквівалентності між моделлю досліджуваного ПЗ і моделями РПЗ;

– формування висновку про рівень безпеки ПЗ на підставі отриманих результатів.

Різні способи представлення моделей на першому етапі (наприклад, представлення у вигляді алгоритму, або послідовністю команд процесора, або послідовністю байтів) зумовлюють утворення їх ієрархії. У межах застосування логіко-аналітичних методів можна використати моделі ПЗ та РПЗ будь-якого рівня ієрархії та способу представлення. Необхідно лише, щоб вони були представлені одним і тим самим способом, з використанням понять одного рівня. Механізм установлення відношення еквівалентності між програмою й РПЗ визначають способом представлення моделі.

Слід зазначити, що процес створення формальних моделей ПЗ та РПЗ є загалом нетривіальною задачею.

Комплексна система оцінювання рівня безпеки ПЗ повинна містити як контрольно-випробувальні, так і логіко-аналітичні методи аналізу та використо-

вувати переваги кожного з них. Важливою перевагою логіко-аналітичних методів є можливість оцінювання надійності отриманих результатів та відслідковування послідовності їх отримання. Проте ці методи є більш трудомісткими та на сьогодні менш розробленими.

Контрольні питання до розділу 7

1. Що таке захист програмного забезпечення?
2. Які ІС вважають критичними?
3. Поняття безпеки ПЗ.
4. Поняття рівня безпеки ПЗ.
5. Дефекти ПЗ.
6. Життєвий цикл ПЗ та його моделі.
7. Загрози безпеці ПЗ.
8. Класифікація руйнівних програмних засобів.
9. Поняття про обчислювальне та операційне середовища.
10. Який фактор визначає можливість програмної дії на обчислювальне середовище?
11. Класифікація руйнівних програмних засобів.
12. Узагальнена концептуальна модель руйнівних програмних засобів.
13. Комп'ютерні віруси.
14. Які ефекти спричиняють віруси у разі здійснення цільових функцій?
15. Основні ознаки прояву функціонування вірусів.
16. Алгоритмічні та програмні закладки.
17. Класи дій алгоритмічних та програмних закладок.
18. Апріорні та апостеріорні закладки.
19. Експлуатаційна безпека ПЗ: модель порушника та модель загроз.
20. Основні принципи забезпечення експлуатаційної безпеки ПЗ.
21. Адаптивна безпека мережі та її основні елементи.
22. Модель адаптивної безпеки мережі.
23. Юридичний захист ПЗ.
24. Технічний захист ПЗ: програмний, апаратний та програмно-апаратний.
25. Методи захисту ПЗ від комп'ютерних вірусів.
26. Засоби захисту ПЗ від комп'ютерних вірусів.
27. Загальні рекомендації із захисту ПЗ ІС від комп'ютерних вірусів.
28. Основні способи впровадження програмних закладок у ПЗ.
29. Статичні та динамічні засоби дослідження ПЗ.
30. Синтаксичні та статистичні методи дослідження ПЗ.
31. Компоненти ПЗ, яке захищється від дослідження.
32. Шляхи підвищення рівня захисту ПЗ від дослідження.
33. Обфускація ПЗ.
34. Вимоги до систем захисту ПЗ від несанкціонованого копіювання.
35. Методи протидії статичним та динамічним способам зняття захисту ПЗ від копіювання.
36. Методи захисту ПЗ від несанкціонованого доступу.
37. Парольний захист ПЗ від несанкціонованого доступу.
38. Захист ПЗ від несанкціонованого доступу прив'язуванням до операційного середовища.
39. Захист ПЗ від несанкціонованого доступу за допомогою електронних ключів.
40. Захист ПЗ від несанкціонованого доступу за допомогою ключових дискет.

41. Які методи використовують для аналізу та оцінювання рівня безпеки ПЗ?
42. Контрольно-випробувальні методи аналізу та оцінювання рівня безпеки ПЗ та їхні різновиди.
43. Типова схема аналізу та оцінювання рівня безпеки ПЗ контрольно-випробувальними методами.
44. Логіко-аналітичні методи аналізу та оцінювання рівня безпеки ПЗ.
45. Типова схема аналізу та оцінювання рівня безпеки ПЗ логіко-аналітичними методами.

Список літератури до розділу 7

1. Кодекс України про адміністративні правопорушення [Електронний ресурс] : закон України № 8073-X : [прийнятий Верховною Радою України 7 грудня 1984 р. : редакція від 14 червня 2018 р.]. – Режим доступу : http://zakon2.rada.gov.ua/laws/show/80731-10/
2. Кримінальний кодекс України [Електронний ресурс] : закон України № 2341-III : [прийнятий Верховною Радою України 5 квітня 2001 р. : редакція від 14 червня 2018 р.]. – Режим доступу : http://zakon5.rada.gov.ua/laws/show/2341-14/
3. Про телекомунікації [Електронний ресурс] : закон України № 1280-IV : [прийнятий Верховною Радою України 18 листопада 2003 р. : редакція від 10 серпня 2012 р.]. – Режим доступу : http://zakon.rada.gov.ua/go/1280-15/
4. Про захист інформації в інформаційно-телекомунікаційних системах [Електронний ресурс] : закон України №80/94-ВР : [прийнятий Верховною Радою України 05 липня 1994 р. : редакція від 19 квітня 2014 р.]. – Режим доступу : http://zakon1.rada.gov.ua/laws/show/80/94-вр
5. Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу : НД ТЗІ 1.1-002-99 : [Електронний ресурс] : [затверджений наказом Департаменту спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України від 28 квітня1999 р. № 22 із змінами згідно наказу Адміністрації Держспецзв'язку від 28.12.2012 № 806]. – Київ : ДСТСЗІ СБ України. – 1999. – 21 с. Режим доступу : http://dstszi.kmu.gov.ua/dstszi/doccatalog/document?id=106340/
6. Горбатий І. В. Технічна експлуатація сучасних комплексів зв'язку : навч. посіб. / І. В. Горбатий, О. В. Тимченко. – Львів : Сполом, 2006. – 244 с.
7. Казарин О. В. Безопасность программного обеспечения компьютерных систем : монография / О. В. Казарин. – М. : МГУЛ, 2003. – 212 с.
8. Казарин О. В. Программно-аппаратные средства защиты информации. Защита программного обеспечения : учебник и практикум для вузов / О. В. Казарин, А. С. Забабурин. – М. : Издательство Юрайт, 2017. – 312 с. – Серия : Специалист.
9. Дудатьєв А. В. Захист програмного забезпечення. Ч. 1 : навч. посіб. / А. В. Дудатьєв, В. А. Каплун, В. П. Семеренко. – Вінниця : ВНТУ, 2005. – 140 с.
10. Каплун В. А. Захист програмного забезпечення. Ч. 2 : навч. посіб. / В. А. Каплун, О. В. Дмитришин, Ю. В. Баришев. – Вінниця : ВНТУ, 2014. – 105 с.

Розділ 8

ІНФОРМАЦІЙНА БЕЗПЕКА ПІДПРИЄМСТВ ТА ОРГАНІЗАЦІЙ. СИСТЕМИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

8.1. Інформаційна безпека підприємств та організацій

8.1.1. Модель багаторівневого захисту інформаційних систем підприємств та організацій

Поняття ***багаторівневого захисту***, або ***ешелонованої оборони*** (Defence in depth) надійшло в інформаційні технології з військових директив. З погляду інформаційної безпеки ***модель багаторівневого захисту*** визначає набір рівнів захисту інформаційної системи підприємства, організації, компанії чи корпорації [1–9]. Коректне організування захисту на кожному з виділених рівнів (рубежів) дає змогу вберегти систему інформаційної безпеки від інцидентів інформаційної безпеки (здійснення загроз інформаційній безпеці). Така модель повинна мати декілька рівнів. Щоб дістатися до закритої інформації, зловмисникові необхідно подолати всі рівні захисту. Один із можливих варіантів моделі багаторівневого захисту інформаційної системи підприємства чи організації наведено на рис. 8.1.

Для окремого об'єкта підприємства чи організації можна виділити 3 ***рівні захисту***:

– організаційно-правовий;
– фізичний;
– технічний,

які, своєю чергою, можна поділити на 10 підрівнів захисту:

– політика інформаційної безпеки;
– захист периметра території об'єкта;
– захист периметра будівлі;
– захист периметра приміщення;
– захист периметра інформаційної системи;
– захист локальної мережі;

– захист вузлів локальної мережі;
– захист апаратних засобів;
– захист програмного забезпечення;
– захист інформації.

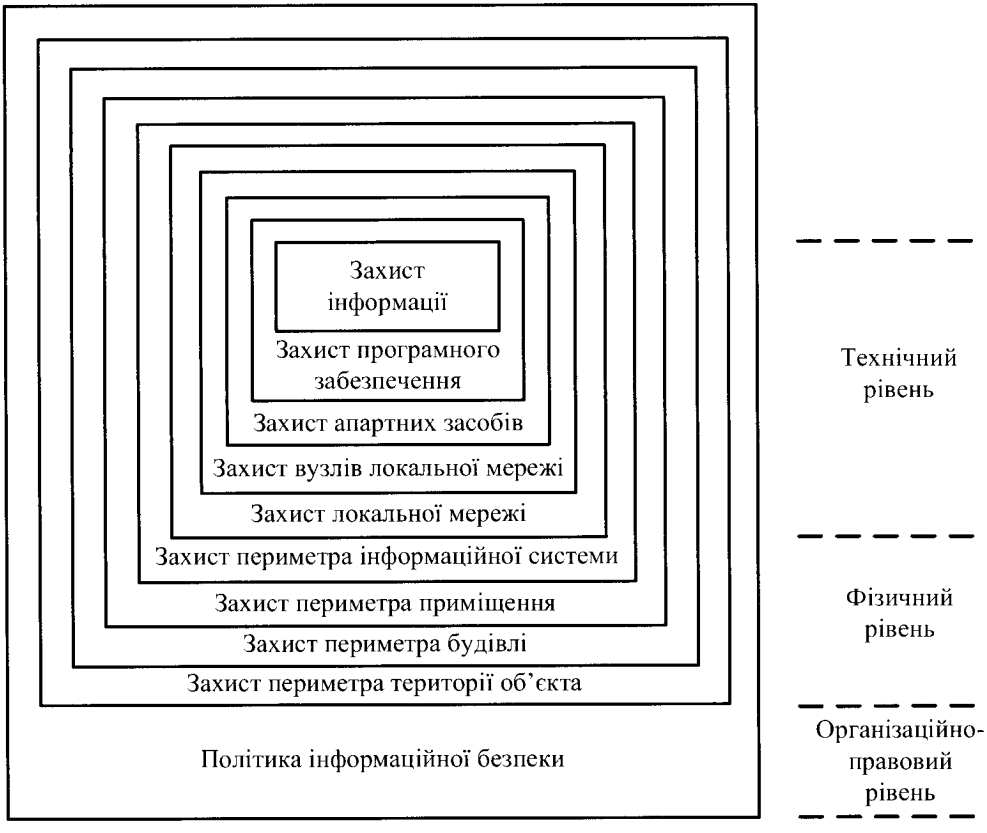


Рис. 8.1. Модель багаторівневого захисту інформаційної системи підприємства чи організації

Згідно з поданою моделлю, підрівень політики інформаційної безпеки розглядають як базовий. Політика інформаційної безпеки описує всі аспекти роботи системи з погляду забезпечення інформаційної безпеки. Цей підрівень також передбачає наявність документованих правових та організаційних заходів захисту (процедур) і порядку інформування про нештатні події, навчання користувачів у галузі інформаційної безпеки та інші аналогічні заходи, які рекомендованої стандартом ISO/IEC 27002.

Фізичний захист підприємства чи організації здійснюють на таких підрівнях: захист периметра території об'єкта, захист периметра будівлі, захист периметра

приміщення та захист периметра ІС. Рівень фізичного захисту містить заходи щодо обмеження фізичного доступу до ресурсів системи: захист території, будівлі, приміщень, контроль доступу, відеоспостереження тощо. До них зараховують і засоби захисту мобільних пристроїв, які використовують співробітники зі службовою метою.

Підрівень захисту периметра інформаційної системи визначає заходи безпеки в "точках входу" в мережу, яку захищають від зовнішніх, потенційно небезпечних мереж. Класичним засобом захисту периметра інформаційної системи є міжмережевий екран, який на підставі заданих правил приймає рішення щодо перепускання мережевого пакета із зовнішньої мережі в захищену мережу. Інші приклади засобів захисту периметра інформаційної системи – системи виявлення вторгнень, засоби антивірусного захисту для шлюзів безпеки тощо.

Технічний захист здійснюють на таких підрівнях: захист внутрішньої локальної мережі, захист вузлів локальної мережі, захист апаратних засобів, захист програмного забезпечення, захист інформації.

На підрівні захисту локальної мережі як складової інформаційної системи забезпечують безпеку мережевого трафіку, який передають усередині мережі, та мережевої інфраструктури. Приклади засобів і механізмів захисту на цьому рівні – застосування технології захисту локальних сегментів мережі від несанкціонованого доступу (наприклад, Port security) за допомогою керованих комутаторів, організування віртуальних приватних мереж для захищеного передавання трафіку.

Наступний на схемі – підрівень захисту вузлів локальної мережі. Тут розглядають атаки на окремий вузол мережі й, відповідно, заходи захисту від них. Можна врахувати функціональність вузла й окремо розглянути захист серверів та робочих станцій. Насамперед необхідно приділяти увагу захисту на рівні операційної системи вузла мережі: виконання налаштувань, які підвищують безпеку конфігурації (зокрема відключення служб, які не використовують, або потенційно небезпечних служб), установлення оновлень та забезпечення надійної автентифікації користувачів. Винятково важлива роль тут належить антивірусному захисту.

Підрівень захисту апаратних засобів відповідає за збереження цілісності та працездатності мережевого активного обладнання та кінцевих засобів – комп'ютерів, телефонних апаратів, IP-телефонів, факсимільних апаратів тощо.

Підрівень захисту програмного забезпечення відповідає за захист від атак, спрямованих на конкретні програми – поштові сервери, web-сервери, сервери баз даних. Як приклад можна назвати SQL-ін'єкції – атаки на сервер баз даних, суть яких у тому, що у вхідний текстовий рядок уміщують оператори мови SQL,

які можуть порушити логіку оброблення даних і призвести до отримання порушником конфіденційної інформації. До них можна зарахувати й модифікацію додатків комп'ютерними вірусами. Для захисту від таких атак використовують налаштування безпеки самих додатків, установлення оновлень операційних систем та засобів антивірусного захисту.

Підрівень захисту інформації визначає порядок її захисту в інформаційній системі від несанкціонованого доступу до неї під час її опрацювання, зберігання тощо. Як приклади захисту інформації можна назвати розмежування доступу до інформації засобами операційної системи, шифрування даних при зберіганні та передаванні.

У процесі виявлення ризиків визначають мету порушника й те, на якому рівні або рівнях захисту можна йому протистояти. Відповідно вибирають і засоби забезпечення інформаційної безпеки. Захист від загрози на декількох рівнях знижує ймовірність її здійснення, а отже, і рівень ризику.

8.1.2. Засоби забезпечення інформаційної безпеки підприємств та організацій

До ***засобів забезпечення інформаційної безпеки*** підприємств та організацій належать [1–9]:

– правові засоби, що передбачають заходи контролю за виконанням нормативних актів загальнодержавного значення, механізми розроблення й удосконалення нормативної бази, яка регулює питання захисту інформації;

– організаційні засоби, що полягають у раціональній організації роботи й адмініструванні інформаційної системи підприємства, зокрема конфігурування й адміністрування операційних систем, регламентування повноважень адміністратора, розроблення набору обов'язкових інструкцій, що визначають порядок доступу й роботи в системі;

– технологічні засоби, що ґрунтуються на технології виконання мережевого адміністрування, моніторингу й аудиту безпеки інформаційних ресурсів, ведення електронних журналів реєстрування користувачів, фільтрування й антивірусного оброблення прийнятої інформації;

– фізичні засоби збереження інформації, що забезпечують фізичну охорону носіїв інформації від викрадення, доступу, змінювання чи знищення інформації на них, а також доступу до інформаційної системи з метою доступу, змінювання чи знищення інформації, що передають, накопичують, обробляють, зберігають чи розповсюджують у такій системі;

– апаратні засоби, що здійснюють фізичний захист інформаційної системи підприємства від несанкціонованого доступу, апаратні функції ідентифікації периферійних терміналів чи користувачів, апаратні шифратори тощо;

– програмні засоби, що здійснюють захист за допомогою програм ідентифікації користувачів, парольного захисту й перевіряння повноважень, міжмережевих екранів, криптопротоколів тощо).

Найбільшу увагу розробники й користувачі сьогодні приділяють таким ***напрямкам захисту інформації*** й відповідним їм програмно-технічним засобам захисту:

– захист від несанкціонованого доступу до інформаційних ресурсів автономно працюючих і підключених до мережі комп'ютерів, зокрема серверів і комп'ютерів користувачів мережі Інтернет, що здійснюють програмними, програмно-апаратними й апаратними засобами;

– захист секретної, конфіденційної та особистої інформації від читання сторонніми особами та цілеспрямованого її спотворення, що забезпечують засобами захисту від несанкціонованого доступу й криптографічними засобами;

– захист інформаційних систем від комп'ютерних вірусів, здатних не тільки знищити інформацію, але деколи й пошкодити технічні компоненти систем, який здійснюють за допомогою антивірусного програмного забезпечення.

Забезпечення загальної безпеки інформаційної системи. Це можливо здійснювати за допомогою трьох способів ідентифікації користувачів, які мають права доступу до інформації в телекомунікаційній системі чи мережі:

– спосіб, що забезпечує найвищу безпеку й грунтується на ідентифікації особи користувача на основі унікальних фізичних параметрів (наприклад, з використанням пристроїв для зняття відбитка пальців або долоні, сканування сітківки ока, розпізнавання голосу, ходи тощо);

– спосіб, що оснований на використанні ключа або магнітної картки, із застосуванням якого користувач отримує доступ, якщо в нього є ключ (або магнітна картка), інформацію з якого можна прочитати за допомогою спеціального пристрою; недоліком цього способу є те, що ключ чи картка можуть бути викрадені або скопійовані;

– спосіб, що оснований на використанні пароля, з використанням якого для отримання доступу на клавіатурі спеціального пристрою (наприклад, комп'ютера) необхідно ввести пароль, що є послідовністю букв, цифр або інших символів; недоліком цього способу є необхідність запам'ятовування пароля користувачем, а також можливість потрапляння пароля до іншої людини, якщо вона підгледіла пароль під час введення його користувачем, або до неї потрапив документ із записаним паролем; важливою вимогою є формування унікального, важкого для розгадування, але легкого для запам'ятовування пароля.

8.1.3. Правові, організаційні та технологічні засоби захисту інформації

Правові засоби захисту інформації передбачають заходи контролю за виконанням нормативних актів загальнодержавного значення, механізми розроблення й удосконалення нормативної бази, яка регулює питання захисту інформації [1–9]. Порядок зберігання носіїв інформації повинен бути чітко визначеним у відповідному правовому акті й передбачати збережуваність носіїв інформації, контроль за роботою з інформацією, відповідальність за несанкціонований доступ до носіїв інформації з метою зняття з них копій, зміни чи пошкодження.

Організаційні засоби захисту інформації полягають у раціональній організації роботи й адмініструванні системи (мережі), зокрема конфігуруванні й адмініструванні операційних систем, регламентуванні повноважень адміністратора, розробленні набору обов'язкових інструкцій, що визначають порядок доступу й роботи в інформаційній системі.

До організаційних засобів захисту інформації належать організаційно-технічні (підготовлення приміщень для встановлення в них комп'ютерів, прокладання кабельної системи з урахуванням вимог обмеження доступу до неї тощо) і організаційно-правові засоби (національні законодавства й правила роботи, установлені керівництвом конкретного підприємства або організації). Переваги організаційних засобів – можливість вирішення багатьох різнорідних проблем, простота здійснення, можливість швидкого реагування на небажані дії в інформаційній системі, необмежені можливості модернізації й розвитку. Недоліки – велика залежність від суб'єктивних факторів, зокрема від загальної організації роботи в певному підрозділі.

Технологічні засоби захисту інформації ґрунтуються на технології виконання мережевого адміністрування, моніторингу й аудиту безпеки інформаційних ресурсів, ведення електронних журналів реєстрування користувачів, фільтрування й антивірусного оброблення прийнятої інформації.

Захист інформації від комп'ютерних вірусів сьогодні набуває особливого значення, особливо під час роботи в мережі Інтернет або використанні електронної пошти. ***Комп'ютерним вірусом*** називають рукотворну програму, що здатна самостійно створювати свої копії й впроваджуватися в інші програми, у системні ділянки дискової пам'яті комп'ютера, розповсюджуватися каналами електрозв'язку з метою переривання й порушення роботи програм, псування файлів, операційних систем і компонентів комп'ютера, порушення нормальної роботи користувачів. Сьогодні існують сотні тисяч різних вірусів, і їх можливо класифікувати за низкою ознак.

Для захисту від вірусів необхідно здійснювати антивірусну профілактику. Джерелами ненавмисного вірусного зараження можуть бути тільки переносні носії інформації й телекомунікаційні системи (мережі). Переносні носії інформації – це дискети, переносні жорсткі диски, контрафактні компакт-диски, флеш-пам'ять. Віруси із заражених переносних носіїв можуть потрапити на накопичувач на жорстких магнітних дисках комп'ютера, навіть коли інформацію із цього носія на накопичувач не переносили, а лише було здійснено спробу завантажити операційну систему із зараженого носія. Переносний носій може бути заражений сам, або зараженим може бути який-небудь файл на цьому носії. В окремих випадках можливе зараження вірусами після встановлення на комп'ютер програмного забезпечення, установлювач (інсталятор) якого був попередньо заражений вірусами.

Телекомунікаційні системи чи мережі можуть слугувати постачальниками вірусів при їх підключенні до комп'ютера через модеми й мережеві адаптери. Тому доцільно здійснювати автоматичний антивірусний вхідний контроль усіх файлів, які надходять із мережі, а також використовувати програмні міжмережеві екрани, які дадуть змогу, хоч би частково, усувати можливість зараження вірусами з мережі. Особливо обережно необхідно користуватись засобами електронної пошти. У разі отримання листа із прикріпленим файлом, якщо в тексті листа немає посилання на додаток, рекомендують для забезпечення безпеки взагалі цей додаток не розкривати.

У наш час розроблено значну кількість антивірусних програм для пошуку й видалення комп'ютерних вірусів. Основні заходи, спрямовані на захист комп'ютерів від вірусів:

– установлення на комп'ютер одного або декількох пакетів антивірусних програм і регулярне їх оновлення;

– невикористання неліцензійного або неперевіреного програмного забезпечення;

– періодичне перевіряння операційної системи та файлів на накопичувачі на жорстких магнітних дисках на наявність комп'ютерних вірусів;

– перевіряння зовнішніх носіїв інформації на наявність комп'ютерних вірусів.

Засоби захисту від комп'ютерних вірусів детально розглянуто в розділі 7 цієї книги.

8.1.4. Фізичний захист об'єктів підприємств та організацій

Підприємства чи організації, які здійснюють діяльність, пов'язану з відомостями обмеженого доступу, широко використовують різні засоби фізичного захисту при організації та забезпеченні охорони території та об'єктів [1–9].

Під ***фізичним захистом*** розуміють сукупність організаційних заходів, інженерно-технічних засобів і дій підрозділів охорони для запобігання диверсіям, розкраданням носіїв конфіденційної інформації та інших матеріальних засобів на об'єктах охорони підприємства чи організації.

Застосування засобів фізичного захисту значно підвищує ефективність функціонування системи охорони підприємства чи організації загалом, а з урахуванням особливостей розташування деяких об'єктів підприємства й виконуваних ними завдань практично гарантує досягнення головної мети та вирішення основних завдань щодо охорони підприємства чи організації.

Завдання фізичного захисту такі:

– запобігання випадкам несанкціонованого доступу до об'єктів підприємства;

– своєчасне виявлення несанкціонованих дій на території підприємства чи організації;

– затримування (уповільнення) проникнення порушника, створення перешкод його діям;

– припинення несанкціонованих дій на території підприємства чи організації;

– затримання осіб, причетних до підготовлення або вчинення диверсії, розкрадання носіїв конфіденційної інформації або інших матеріальних цінностей підприємства чи організації.

Основу функціонування системи фізичного захисту становлять організаційні заходи. Вони містять комплекс заходів, які вживає керівництво підприємства чи організації для фізичного захисту його об'єктів, а також нормативно-методичні документи, які регламентують їх здійснення.

Безпосереднє планування комплексу організаційних заходів та контроль за їх виконанням здійснює служба безпеки підприємства спільно з іншими його структурними підрозділами, які беруть участь у процесі захисту конфіденційної інформації. Діяльність цих структурних підрозділів координує заступник керівника підприємства, який відповідає за вирішення завдань щодо захисту конфіденційної інформації.

З метою фізичного захисту об'єктів служба безпеки підприємства забезпечує:

– розроблення, створення й функціонування системи фізичного захисту;

– проведення аналізу уразливості об'єкта для визначення внутрішніх та зовнішніх загроз і можливих способів їх здійснення;

– розроблення із залученням структурних підрозділів підприємства та затвердження в установленому порядку документів, які регламентують питання

організації та здійснення перепускного режиму на об'єктах підприємства, охорони об'єктів підприємства.

З метою ефективного вирішення завдань фізичного захисту об'єктів використовують інженерно-технічні засоби захисту інформації, до основних завдань яких належать:

– запобігання проникненню сторонніх осіб (зловмисників) до носіїв конфіденційної інформації;

– захист носіїв конфіденційної інформації від знищення й нанесення іншої шкоди в результаті впливу стихійних лих та інших надзвичайних ситуацій;

– закриття можливих технічних каналів витоку конфіденційної інформації.

Формулюючи завдання інженерно-технічного захисту інформації на підприємстві, визначають:

– перелік можливих загроз захисту підприємства чи організації;

– об'єкти підприємства чи організації, що підлягають захисту;

– методи захисту інформації;

– порядок контролю ефективності вирішення завдань фізичного захисту об'єктів.

Основними принципами створення системи інженерно-технічного захисту інформації, яка володітиме високою ефективністю й надійністю, є:

– утаємниченість – збереження в таємниці факту створення й особливостей побудови системи інженерно-технічного захисту інформації;

– гнучкість – можливість оперативного реагування на зміни ступеня захищеності конфіденційної інформації залежно від вибору критеріїв, способів і методів її захисту;

– багатозональність – розміщення джерел інформації в різних зонах захисту з контрольованим рівнем безпеки, тобто поділ території підприємства чи організації на зони з різними рівнями доступу для персоналу підприємства відповідних категорій (керівництво підприємства, керівники підрозділів, окремі посадові особи тощо) і розміщення об'єктів у різних зонах залежно від ступеня важливості інформації, яку зберігають або опрацьовують на цих об'єктах;

– багаторубіжність – створення відповідних рубежів захисту об'єктів підприємства чи організації на межі обраних зон захисту інформації.

На межах зон створюють, як правило, один або кілька рубежів захисту. Багаторубіжність захисту об'єктів дає можливість спорудити одну або кілька перешкод на можливому напрямку дії або шляху просування зловмисника. Ці рубежі можуть бути оснащені технічними засобами або містити спеціальні пости охорони (контрольно-перепускні пункти).

Основні методи інженерно-технічного захисту інформації на підприємстві:

– створення фізичних, електронних та інших перешкод зловмиснику на його шляху до носіїв конфіденційної інформації та їх джерел;

– введення зловмисника в оману за допомогою технічних засобів шляхом підготовлення й розповсюдження (нав'язування) неправдивої інформації;

– застосування різних засобів контролю несанкціонованого доступу для виявлення спроб здійснення зловмисником загроз безпеці інформації та інформування про виявлені спроби посадових осіб, які беруть участь у виробленні заходів захисту інформації на об'єктах підприємства.

Інженерно-технічні засоби фізичного захисту складаються з технічних засобів і фізичних бар'єрів.

До технічних засобів фізичного захисту належать:

– засоби охоронної сигналізації, розміщені по периметру зон охорони, будівель, споруд та інших об'єктів підприємства, а також службових приміщень на цих об'єктах;

– засоби контролю проходу (доступу), установлені на контрольно-перепускних пунктах у будівлях, спорудах (об'єктах) підприємства й у службових приміщеннях, що підлягають охороні;

– засоби спостереження за периметрами зон охорони, контрольно-перепускними пунктами в будівлях, спорудами підприємства, а також службовими приміщеннями, що підлягають охороні;

– засоби спеціального зв'язку (зокрема екстрені);

– засоби виявлення заборонених предметів (зокрема носіїв конфіденційної інформації);

– засоби систем життєзабезпечення (електроживлення, освітлення тощо).

Перерахованими засобами обладнують периметри зон охорони, будівлі, споруди й приміщення, а також контрольно-перепускні пункти.

У сучасних системах фізичного захисту підприємств чинне місце займають засоби охоронної сигналізації та відеоспостереження.

Засоби охоронної сигналізації призначені для своєчасного інформування відповідних осіб (співробітників охорони підприємства, операторів пультів охорони) про порушення вихідного стану або встановленого режиму роботи кінцевих пристроїв цих засобів, спричинених несанкціонованим проникненням осіб у зони охорони (території), а також спробами несанкціонованого перетину об'єктів (службових приміщень) підприємства, що підлягають охороні.

Засоби охоронної сигналізації, як правило, функціонують у неробочий час, коли службові приміщення (об'єкти) підприємства в установленому порядку здано під охорону.

Засоби відеоспостереження за об'єктами охорони забезпечують отримання та документування відеоінформації про обстановку на об'єкті охорони з метою прийняття своєчасного рішення про застосування сил і засобів охорони для припинення (виключення) спроб несанкціонованого проникнення на об'єкт сторонніх осіб.

Системи відеоспостереження за об'єктами охорони призначені для:

– визначення причин спрацювання засобів охоронної сигналізації, установленої на об'єкті;

– отримання оперативної інформації про проникнення на об'єкт охорони сторонньої особи;

– документування інформації про події та злочини, скоєні на території об'єкта охорони.

Усі технічні засоби, які входять до складу системи фізичного захисту підприємства чи організації, у випадку відключення основного джерела електроживлення повинні зберігати працездатність, що забезпечують їх автоматичним перемиканням на резервні джерела електроживлення.

До фізичних бар'єрів належать:

– будівельні конструкції, установлені в будівлях і спорудах (стіни, перекриття, ворота, двері);

– спеціально розроблені конструкції (загородження, протитаранні пристрої, решітки, посилені двері);

– інші фізичні перешкоди, які утруднюють доступ на об'єкти підприємства чи організації.

Важливе місце в системі фізичного захисту об'єктів підприємства займають підрозділи охорони, які є структурними підрозділами підприємства. Організаційну структуру й чисельність підрозділів охорони визначають залежно від особливостей об'єктів охорони й ступеня обладнання їх інженерно-технічними засобами захисту, специфіки несення чергування з охорони об'єктів, а також від інших умов, пов'язаних із забезпеченням надійного захисту цих об'єктів.

8.1.5. Апаратні засоби захисту та збереження інформації

Апаратні засоби захисту інформації. За допомогою апаратних засобів здійснюють фізичний захист систем (мереж) від несанкціонованого доступу, апаратні функції ідентифікації периферійних терміналів чи користувачів тощо [1–3]. Останнім часом активно розвивають засоби захисту від витоку інформації через мережу електроживлення (встановлюють спеціальні мережеві фільтри), канали електрозв'язку (використовують спеціальні пристрої для шифрування

інформації перед передаванням у канал зв'язку та дешифрування прийнятої інформації) або завдяки електромагнітному випромінюванню комп'ютера (застосовують екрановані приміщення, генератори шумових випромінювань, здійснюють спеціальний підбір моніторів та інших складових комп'ютерів, які характеризуються найменшим випромінюванням), а також засоби захисту від електронних пристроїв несанкціонованого збирання інформації, установлених безпосередньо в комп'ютери тощо.

Апаратні засоби збереження інформації в умовах апаратних збоїв та поламок. Найчастіше трапляються такі апаратні збої:

– раптове вимкнення електроживлення робочих станцій, серверів, комутаторів та інших мережевих пристроїв;

– вихід із ладу накопичувача на жорстких магнітних дисках сервера або його контролера;

– поява зіпсутих кластерів у накопичувачі на жорстких магнітних дисках, у які неможливо записувати або з яких неможливо читати інформацію.

Для захисту від апаратних збоїв при раптовому вимкненні електроживлення робочих станцій, серверів, комутаторів та інших мережевих пристроїв, що можуть спричинити втрату інформації через відсутність напруги живлення, їх живлять через блоки безперебійного живлення. У звичайному режимі роботи блок безперебійного живлення передає зі свого входу на вихід напругу електроживлення на підключені до його виходу пристрої. При зникненні напруги в мережі електроживлення цей блок в автоматичному режимі формує напругу електроживлення за допомогою спеціальної схеми та акумулятора, що вбудовані в нього, і подає її на ці пристрої.

Для надійного електроживлення розподільчих пунктів телекомунікаційних мереж, серверних приміщень та навіть великих об'єктів інфраструктури інформаційних систем використовують системи безперебійного електроживлення, що забезпечують електроживлення за 1 класом. До складу системи безперебійного електроживлення входять: основна та резервна електролінія від різних електричних підстанцій міської електромережі, пристрій автоматичного включення резерву (АВР), генератор напруги 380 В частотою 50 Гц від акумуляторних батарей, блок акумуляторних батарей, дизельний або бензиновий генератор напруги 380 В частотою 50 Гц. При застосуванні такої системи у випадку зникнення напруги на основній електролінії пристрій АВР від'єднує навантаження (користувачів електромережі об'єкта) від основної електролінії та під'єднує їх до резервної електролінії. У випадку короткочасної відсутності напруги на основній електролінії та в моменти переключення навантаження з однієї електролінії на іншу в автоматичному режимі запускають генератор напруги від акумуляторних батарей. У випадку тривалої відсутності напруги на

основній та резервній електролініях в автоматичному режимі запускають дизельний або бензиновий генератор напруги. За появи напруги на основній чи резервній електролінії пристрій АВР від'єднує генератор напруги від акумуляторних батарей або дизельний (бензиновий) генератор. Після цього такі генератори вимикають.

Для захисту від апаратних збоїв при виході з ладу накопичувача на жорстких магнітних дисках сервера або його контролера, появи зіпсутих кластерів у такому накопичувачі застосовують такі засоби:

– дзеркальний накопичувач на жорстких магнітних дисках або дублювання накопичувача;

– використання масивів накопичувачів на жорстких магнітних дисках;

– захист службових таблиць;

– захищений режим записування на накопичувач на жорстких магнітних дисках та використання області Hot Fix.

Використання дзеркального накопичувача на жорстких магнітних дисках або дублювання накопичувача. Одним із найменш надійних компонентів комп'ютера є накопичувач на жорстких магнітних дисках. З метою збереження інформації у випадку виходу з ладу такого накопичувача використовують дзеркальний накопичувач, тобто два однакові накопичувачі приєднують паралельно до одного контролера. Записують інформацію одночасно на два накопичувачі. Якщо один вийде з ладу, система попередить оператора, але при цьому інформацію буде збережено. ***Дублювання накопичувача*** полягає в приєднанні двох однакових накопичувачів до двох контролерів. Такий варіант надійніший від використання дзеркального накопичувача.

Використання масивів накопичувачів на жорстких магнітних дисках. У системах збереження даних застосовують ***масиви накопичувачів на жорстких магнітних дисках*** (redundant array of independent/inexpensive disks – RAID), які суттєво зменшують ризик простою інформаційної системи через відмови таких накопичувачів, які є одним із найменш надійних компонентів сучасних комп'ютерів. У таких масивах застосовують, наприклад, чотири однакові накопичувачі на жорстких магнітних дисках. Інформацію записують частинами на окремі диски з дублюванням, щоби при виході з ладу одного з дисків можливим було відновлення інформації за допомогою читання її частин, що записані на трьох інших накопичувачах.

Захист службових таблиць. Важливими системними таблицями, які зберігають інформацію про розміщення каталогів та файлів на накопичувачі на жорстких магнітних дисках, є ***DET*** (Directory Entry Table) та ***FAT*** (File Allocation Table) таблиці. Якщо вийшов із ладу один із кластерів накопичувача, де записані ці таблиці, інформація може стати недоступною. Щоб запобігти

цьому, у системі зберігають дві окремі копії цих таблиць. Якщо зіпсутий один із кластерів із записаною таблицею, то операційна система комп'ютера звертається до його копії. Номер зіпсутого кластера записують у таблицю зіпсутих кластерів комп'ютера, і дані з нього зберігають на диску в спеціально призначеній ділянці жорсткого магнітного диска. Ідентичність DET- та FAT-таблиць перевіряють під час кожного вмикання комп'ютера.

Захищений режим записування на накопичувач на жорстких магнітних дисках та використання області Hot Fix. Унаслідок інтенсивних процесів записування та читання деякі кластери накопичувача на жорстких магнітних дисках можуть втратити можливість зберігати інформацію. Мережева операційна система Netware захищає дані від записування в такі кластери, використовуючи два механізми, які доповнюють один одного: читання після записування, використання області Hot Fix.

Відразу після записування даних у кластер їх порівнюють із тими даними, які ще є в пам'яті. Якщо збіг повний, то пам'ять очищують, і система записує наступний блок. Якщо ж виявлено розходження, то блок вважають дефектним, і діє механізм використання Hot Fix.

Деяку частину жорсткого магнітного диска резервують як область Hot Fix. За замовчуванням область Hot Fix займає 4 % від ємкості накопичувача на жорстких магнітних дисках. Її можна збільшити або зменшити під час налаштування операційної системи. Кластери цієї області заміщують дефектні кластери жорсткого магнітного диска. При виявленні дефектного кластера його позначають як зіпсутий, а інформацію з нього записують у кластер, який належить області Hot Fix. Згодом у випадку звернення до зіпсутого кластера фактично відбуватиметься звернення до кластера в області Hot Fix.

8.1.6. Програмні засоби захисту інформації

Найбільшу увагу розробники й користувачі сьогодні приділяють програмним засобам захисту від несанкціонованого доступу до інформаційних ресурсів і особливо до мережі Інтернет [1–3]. Організаційні, технологічні й апаратні методи захисту, як правило, не можна використати без програмної компоненти. При цьому слід мати на увазі, що вартість багатьох програмних системних рішень із захисту інформації суттєво перевищує за затратами апаратні, технологічні й організаційні рішення.

Найпоширеніші такі ***програмні засоби захисту інформації***:

– програми ідентифікації користувачів;

– парольний захист і перевіряння повноважень;

– міжмережеві екрани;

– криптографічні протоколи.

Ідентифікація користувачів, парольний захист і перевіряння повноважень. Захищають інформацію від несанкціонованого доступу до ресурсів інформаційної системи підприємства чи організації за допомогою:

– надавання користувачеві, а також терміналам, програмам, файлам і каналам електрозв'язку унікальних імен і кодів (ідентифікаторів);

– виконання процедур установлення справжності у разі доступу до інформаційної системи чи необхідної інформації, тобто перевіряння того, що особа або пристрій, які повідомили ідентифікатор, насправді йому відповідають (ідентифікують програми, термінали й користувачів у разі доступу до системи часто перевірянням паролів, рідко – зверненням у спеціальну службу, яка завідує сертифікацією користувачів); почали використовувати й апаратно-програмні засоби біометричної ідентифікації користувачів (миші й клавіатури з функцією дактилоскопічної ідентифікації, системи розпізнавання користувача за голосом, за відеозображенням, зокрема за сітківкою й райдужною оболонкою очей, відбитками пальців, ходою тощо);

– перевіряння повноважень, тобто перевіряння права користувача на доступ до інформаційної системи підприємства чи організації та окремо права на доступ до інформації (даних);

– автоматичного реєстрування в спеціальному журналі аудиту всіх запитів до інформаційних ресурсів із зазначенням ідентифікатора користувача, термінала, часу й суті запиту, за допомогою чого стає можливим визначення місцезнаходження комп'ютера, з якого діяв зловмисник.

Міжмережеві екрани. Міжмережевий екран знаходиться між локальною (внутрішньою) мережею (яку захищають) і зовнішньою мережею та контролює всі інформаційні потоки, що надходять до локальної мережі й виходять із неї. Під час контролю трафіку його фільтрують (вибіркове перепускання через екран) з виконанням спеціальних перетворень і формуванням сповіщень для відправника, якщо його даним у перепустці було відмовлено. Фільтрування здійснюють на основі набору правил згідно з концепцією інформаційної безпеки підприємства чи організації, що здійснені в програмному забезпеченні міжмережевого екрана. Міжмережевий екран може бути виконаний у вигляді спеціального пристрою із вбудованим програмним забезпеченням або стандартного мережевого комутаційного пристрою чи сервера доступу (сервер-шлюз, проксі-сервер, вузловий комп'ютер тощо) із записаним спеціальним програмним забезпеченням, що здійснює функцію міжмережевого екрана й працює під управлінням операційної системи такого пристрою.

Робота міжмережевого екрана полягає в аналізі структури та вмісту інформаційних пакетів, що надходять із зовнішньої мережі, і (залежно від результатів аналізу) пропусканні або непропусканні пакетів до локальної мережі.

Міжмережевий екран переважно виконує такі функції:

– фізичне відділення робочих станцій і серверів локальної мережі від зовнішніх мереж;

– ідентифікацію запитів, які надходять до мережі;

– перевіряння повноважень і прав доступу користувачів до внутрішніх ресурсів мережі;

– реєстрування всіх запитів ззовні до компонентів локальної мережі;

– контроль цілісності програмного забезпечення й даних;

– економію адресного простору зовнішньої мережі;

– приховання IP-адреси (Internet Protocol) локальних серверів із метою захисту від зловмисників.

Криптографічні протоколи. Зараз активно розвивають та впроваджують в інформаційних системах підприємств та організацій криптографічні протоколи (протоколи шифрування інформації), які мають забезпечувати конфіденційність таких складних мережевих додатків, як електронна пошта, електронний банк, електронна торгівля, електронний бізнес. Криптографічний захист інформації полягає в перетворенні інформації за спеціальним алгоритмом із використанням методів шифрування й ключів, у результаті чого за зовнішнім виглядом даних неможливо без знання ключа визначити їх зміст.

За допомогою криптографічних протоколів можливо забезпечити безпечне передавання інформації через мережу, зокрема реєстраційних імен та паролів, що необхідні для ідентифікації програм і користувачів. На практиці використовують два типи шифрування: симетричне й асиметричне.

При ***симетричному шифруванні*** для шифрування й дешифрування даних застосовують один секретний ключ. Цей ключ повинен бути переданий безпечним способом учасникам взаємодії до початку передавання зашифрованих даних. При симетричному шифруванні застосовують блокові або потокові методи шифрування (шифри). У першому випадку початкове повідомлення ділять на блоки постійної довжини, кожен з яких перетворюють за певними правилами на блок зашифрованого тексту. За потоковимиі методами шифрування оперують з окремими бітами й байтами початкового повідомлення й ключа. Такі методи шифрування мають вищу криптостійкість, але необхідно використовувати довгі ключі, що переважно дорівнюють довжині переданого повідомлення.

Асиметричне шифрування основане на використанні двох різних ключів – відкритого й закритого, що пов'язані між собою, але знання одного ключа не дає змоги визначити другий. Відкритий ключ вільно розповсюджують у мережі, закритий ключ відомий лише його власнику. Якщо шифрування виконують відкритим ключем, то повідомлення може розшифрувати тільки власник закритого ключа – такий метод шифрування використовують для передавання конфіденційної інформації. Якщо повідомлення шифрують закритим ключем, то його може розшифрувати будь-який користувач, який знає відкритий ключ, але непомітно змінити або підмінити зашифроване повідомлення власник відкритого ключа не може.

Прикладом асиметричного шифрування є ***електронний цифровий підпис***. Такий підпис – це послідовність символів, отримана в результаті криптографічного перетворення вихідної інформації з використанням закритого ключа, що дає змогу підтвердити цілісність, незмінність та авторство цієї інформації із застосуванням відкритого ключа.

У разі використання електронного цифрового підпису файл за допомогою спеціальної програми перетворюють на набір символів. Генерують два ключі: відкритий і закритий. Набір символів шифрують за допомогою закритого ключа. Таке зашифроване повідомлення і є електронним цифровим підписом. Незашифрований файл у початковому вигляді передають через інформаційну систему разом з електронним цифровим підписом. На приймальній стороні, отримавши файл і підпис, за допомогою відкритого ключа розшифровують набір символів із підпису. Потім порівнюють два набори символів. Якщо вони повністю збігаються, то насправді прийняли файл, що створений і підписаний на передавальній стороні.

Методи криптографічного захисту детально розглянуто в розділі 3, а криптографічні протоколи – у розділах 5 та 6 цієї книги.

8.2. Безпека інформації на об'єктах підприємств та організацій

8.2.1. Канали витоку інформації в інформаційних системах підприємств та організацій

Під час оброблення інформації в інформаційній системі підприємства чи організації завжди існує обмін інформацією між компонентами інформаційної системи, отже, можна говорити про наявність каналів обміну та каналів витоку інформації [4–15].

Канал витоку інформації (channel of information leakage) – сукупність джерела інформації, матеріального носія або середовища розповсюдження сигналу, що несе вказану інформацію, і засобу виділення інформації із сигналу або носія. Канал витоку інформації створює загрозу розкриття інформації. Найбільш ***загальною класифікацією каналів витоку*** може бути така:

– несанкціонований доступ – канал спеціального впливу порушника, який, використовуючи штатні засоби доступу до інформаційних ресурсів, порушує встановлені правила розмежування доступу з метою здійснення будь-яких з основних видів загроз інформації;

– канал спеціального недопустимого регламентом впливу на параметри середовища функціонування, здійснюваного з метою порушення доступності ресурсів інформаційної системи;

– канал спеціального впливу нештатними програмними й / або програмно-технічними засобами на елементи обладнання, програми, дані та процеси в інформаційній системі, що встановлюють у процесі її експлуатації, з метою здійснення будь-яких з основних видів загроз інформації;

– канал спеціального впливу на компоненти інформаційної системи за допомогою закладних пристроїв і / або програмних закладок, впроваджених у середовище функціонування інформаційної системи на передексплуатаційних стадіях її життєвого циклу, що використовують із метою здійснення будь-яких з основних видів загроз інформації;

– канал витоку інформації, утворений за допомогою використання інформаційних параметрів побічного електромагнітного випромінювання та наведень із метою порушення конфіденційності;

– канал витоку інформації, утворений за допомогою використання інформаційних параметрів випромінювання в оптичному діапазоні частот із метою порушення конфіденційності;

– канал витоку інформації, утворений за допомогою використання інформаційних параметрів сигналів, що виникли внаслідок побічних акустоелектричних перетворень інформаційних сигналів у кінцевому обладнанні, з метою порушення конфіденційності;

– канал здійснення будь-яких з основних видів загроз інформації, утворений внаслідок використання випадкових збоїв та відмов у роботі обладнання;

– канал спеціального впливу на компоненти інформаційної системи за допомогою впровадження комп'ютерних вірусів.

Представлену класифікацію можна деталізувати на основі такого показника, як ***ступінь взаємодії зловмисника з елементами об'єкта оброблення інформації та самою інформацією***.

До першого класу зараховують ***канали від джерела інформації у разі несанкціонованого доступі до нього***:

– розкрадання носіїв інформації;

– копіювання інформації з носіїв (матеріально-речовинних, магнітних тощо);

– фотографування або відеознімання носіїв інформації всередині приміщення;

– підслуховування розмов (зокрема підслуховування аудіозаписів та перегляд відеозаписів);

– установлення закладних пристроїв у приміщення та знімання інформації з їхньою допомогою;

– вивідування інформації в обслуговуючого персоналу на об'єкті.

До другого класу зараховують ***канали від засобів оброблення інформації у разі несанкціонованого доступу до них***:

– зняття інформації із пристроїв електронної пам'яті;

– установлення закладних пристроїв у системи оброблення інформації;

– установлення програмного забезпечення, що дає змогу отримувати інформацію;

– копіювання інформації з технічних пристроїв відображення (фотографування з моніторів тощо).

До третього класу належать ***канали від джерела інформації***:

– отримання інформації з акустичних каналів (у системах вентиляції, теплопостачання, а також за допомогою напрямлених мікрофонів);

– отримання інформації з віброакустичних каналів (з використанням акустичних давачів, лазерних пристроїв);

– використання технічних засобів оптичної розвідки (біноклів, підзорних труб тощо);

– використання технічних засобів оптико-електронної розвідки (зовнішніх телекамер, приладів нічного бачення тощо);

– огляд відходів і сміття;

– вивідування інформації в обслуговуючого персоналу за межами об'єкта;

– вивчення вихідної за межі об'єкта відкритої інформації (публікацій, рекламних проспектів тощо.

До четвертого класу належать ***канали від засобів оброблення інформації без несанкціонованого доступу до них***:

– електромагнітні випромінювання системи оброблення інформації (паразитні електромагнітні випромінювання, паразитна генерація підсилювальних каскадів, паразитна модуляція високочастотних генераторів низькочастотним сигналом, що містить конфіденційну інформацію);

– електромагнітні випромінювання ліній зв'язку;

– підключення до ліній зв'язку;

– зняття наведень електричних сигналів із ліній зв'язку;

– зняття наведень із системи електроживлення;

– зняття наведень із системи заземлення;

– зняття наведень із системи теплопостачання;

– використання високочастотного нав'язування;

– зняття з ліній, що виходять за межі об'єкта, сигналів, утворених на технічних засобах завдяки акустоелектричним перетворенням;

– зняття випромінювань волоконно-оптичних ліній зв'язку;

– підключення до баз даних і персональних комп'ютерів через комп'ютерні мережі.

З погляду ***способу здійснення*** можна виділити фізичні та інформаційні канали витоку інформації.

Широко вживані ***фізичні канали витоку інформації***:

1. Акустичний (віброакустичний) канал, що пов'язаний із розповсюдженням звукових хвиль у повітрі або пружних коливань в інших середовищах, які виникають під час роботи засобів відображення інформації інформаційної системи.

2. Електромагнітний канал, причиною виникнення якого є електромагнітне поле, що виникає в результаті проходження електричного струму в технічних засобах інформаційної системи.

3. Оптичний канал, що пов'язаний із можливістю отримання інформації за допомогою оптичних засобів в інфрачервоному, видимому або ультрафіолетовому діапазонах частот.

У ряді випадків застосовують канали витоку інформації, що пов'язані з розповсюдженням рентгенівського або гамма-випромінювання.

Серед ***інформаційних каналів витоку інформації*** найбільш розповсюдженим є ***прихований канал з обміном через пам'ять*** (storage covert channel), тобто канал, що виникає завдяки використанню доступу до спільних об'єктів системи (як правило, спільна пам'ять). У цьому випадку порушник активізує деякий процес, за допомогою якого може отримати дозвіл на читання спільного з користувачем ресурсу, при цьому користувач може читати (має право читання) або змінювати цей ресурс (має право запису), а порушник може лише читати цей ресурс. Наприклад, у каталог внесено імена файлів. Доступ до самих файлів для порушника неможливий, а доступ до каталогу можливий. Якщо користувач створив закриті файли, то інформація про файлову структуру стала доступною для порушника. Отже, виник витік частини інформації, що належить користувачеві, зокрема, існування чи неіснування конкретного файлу. Захист від витоку інформації через такий канал здійснюють шляхом контролю доступу.

Іншим каналом витоку інформації є ***прихований канал з обміном у часі*** (timing covert channel). Порушник отримує інформацію не про увесь процес, що ініційований користувачем, а лише про його змінювання за дії цінної закритої інформації. Відмінність цього каналу витоку інформації від прихованого каналу з обміном через пам'ять полягає в тому, що порушник отримує не саму конфіденційну інформацію, а дані про операції з нею. Наприклад, користувач ініціює процес роздруку на принтері результатів чергового етапу оброблення цінної інформації. Якщо принтер є спільним для користувача й порушника, то порушник отримує інформацію про періодичність оброблення користувачем цінної інформації, а це вже є витоком інформації. Як правило, такий канал використовують із метою подальшого отримання інформації за допомогою прихованого каналу з обміном через пам'ять.

Додамо, що будь-які канали витоку можуть бути ***контактними*** (коли здійснюють безпосередній доступ до елементів інформаційної системи) або ***безконтактними*** (коли здійснюють візуальне перехоплення інформації або перехоплення за рахунок випромінювань).

Також використовують поняття ***технічного каналу витоку інформації***, що є сукупністю джерела небезпечного сигналу, середовища поширення небезпечного сигналу та засобу технічної розвідки. Технічні канали витоку інформації поділяють на канали первинних електромагнітних випромінювань, канали вторинних наведень у навколишніх конструкціях і системах комунікацій, акустичні й оптичні канали.

Серед каналів первинних електромагнітних випромінювань найпоширенішими є канали, що використовують випромінювання таких об'єктів:

– процесорів робочих станцій та серверів;

– зовнішніх запам'ятовувальних пристроїв та сховищ даних;

– друкувальних пристроїв;

– пристроїв підготовлення, введення / виведення та копіювання даних;

– комунікаційного обладнання для передавання даних, а також ліній зв'язку.

Серед каналів вторинного наведення розрізняють канали, що використовують наведення електромагнітних полів у металевих деталях таких об'єктів:

– ліній телефонного зв'язку та мереж проводового мовлення;

– мереж електропостачання, систем живлення технічних засобів та шин заземлення;

– систем водо- газо- і теплопостачання, вентиляції, каналізації тощо.

Серед акустичних та оптичних каналів витоку інформації розрізняють канали, що використовують:

– акустичну інформацію, отриману з використанням спрямованих мікрофонів та диктофонів;

– лазерне сканування акустичної інформації з вікон та тонких перегородок;

– дистанційне знімання оптичних зображень із дисплеїв, пристроїв наочного відображення інформації та здійснення відеоспостереження;

– витік інформації за рахунок прослуховування звуку, що поширюється в стінах і перекриттях.

8.2.2. Забезпечення безпеки інформації на об'єктах підприємств та організацій

При забезпеченні безпеки інформації на об'єктах підприємств та організацій прагнуть запобігти несанкціонованому доступу, зміні або знищенню інформації, забезпечити безпеку приміщень, засобів оброблення критичної або службової інформації за допомогою використання різних засобів контролю на проникнення [1–9].

Можливі об'єкти впливу в інформаційних системах:

– апаратне забезпечення;

– програмне забезпечення;

– комунікації (забезпечення передавання та оброблення даних через канали зв'язку й комутаційне обладнання);

– персонал.

На рис. 8.2 розглянуто основні заходи, пов'язані саме із забезпеченням безпеки інформаційних ресурсів підприємства.

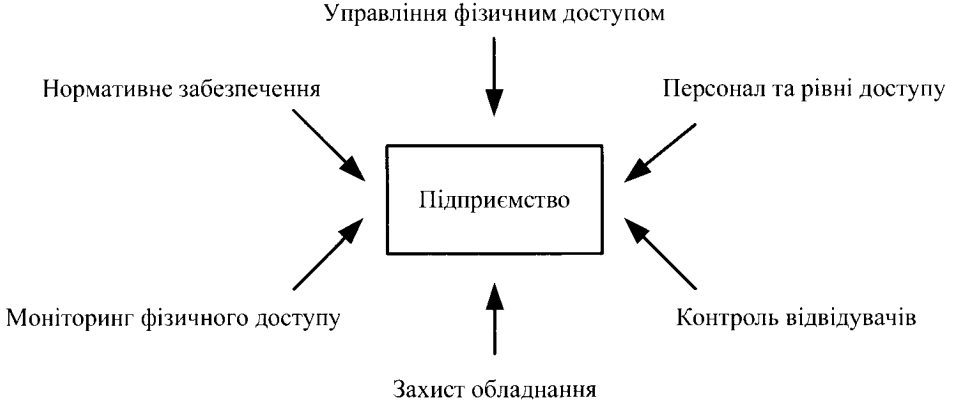


Рис. 8.2. Заходи щодо забезпечення фізичної безпеки інформаційних ресурсів

До ***заходів нормативного забезпечення*** належать:

– розроблення, документування й періодичне оновлення політики фізичного захисту й захисту середовища інформаційної системи;

– розроблення процедур і заходів, пов'язаних зі здійсненням політики фізичного захисту й захисту засобів інформаційної системи.

Заходи з розмежування персоналу за рівнями доступу передбачають розроблення списків персоналу для організації контролю доступу їх до зони, яку охороняють, відповідно до політики безпеки підприємства чи організації, а також механізму ідентифікації (бейджики, інформаційні карти тощо). До ***заходів з управління фізичним доступом*** належать:

– наявність системи управління доступом у всіх точках доступу до інформаційних ресурсів і активів (апаратно-програмні ресурси, бази даних, системна документація тощо);

– чітке визначення периметра безпеки для захисту приміщень та зон розташування засобів оброблення інформації;

– надання доступу в приміщення будівлі лише авторизованому персоналу;

– надання фізичного доступу до інформаційних ресурсів підприємства чи організації після виконання процедур перевіряння повноважень на доступ;

– регулярний аналіз і перегляд права доступу співробітників у зоні безпеки.

До ***заходів із моніторингу фізичного доступу*** належать:

– використання в процесі моніторингу пристроїв спостереження й сигналізації, а також автоматизованих засобів, які забезпечують розпізнання порушень та ініціюють відповідні дії;

– використання методів ідентифікації / автентифікації для контролю фізичного доступу до приміщень.

До ***заходів із захисту обладнання*** належать:

– розташовування обладнання з урахуванням вимог мінімізації доступу в робоче приміщення осіб, не пов'язаних із його обслуговуванням;

– захист обладнання від перебоїв електроживлення (застосування блоків безперебійного живлення, резервних джерел живлення тощо);

– забезпечення протипожежного захисту, а також захисту від впливу навколишнього середовища та від інших екологічних і техногенних катастроф;

– забезпечення регулярного огляду й контролю обладнання з метою виявлення ознак, які можуть спричинити його відмову.

При виведенні з експлуатації носіїв інформації, які містять цінну (критичну / службову) інформацію, необхідно забезпечити гарантоване стирання з них залишкової інформації або їх фізично знищити.

Також необхідно вжити заходів із забезпечення захисту телекомунікаційних кабельних мереж від перехоплення інформації або їх пошкодження. Цього можна досягти, прокладаючи мережеві кабелі зовні загальнодоступних зон, закриваючи їх захисними коробами, контролюючи кросові приміщення і телекомунікаційні шафи.

До ***заходів із контролю відвідувачів*** належить організація зони їх реєстрації та ведення журналів обліку доступу.

Отже, безпека інформаційних ресурсів – це комплекс інженерно-технічних заходів, дотримання яких дасть змогу ефективно захистити бізнес від можливих атак конкуруючих організацій, атак із середини підприємства чи організації, а також можливих стихійних лих.

Питання безпеки інформаційних об'єктів розглянуто, зокрема, у стандарті ISO/IEC 27002 "Інформаційні технології – Технології безпеки – Практичні правила управління інформаційною безпекою". Стандарт містить рекомендації з комплексного захисту інформації та надає найкращі практичні поради щодо управління інформаційною безпекою для тих, хто відповідає за створення, побудову або обслуговування систем управління інформаційною безпекою. Інформаційну безпеку визначають у стандарті як збереження конфіденційності (упевненість у тому, що інформація доступна тільки тим, хто уповноважений мати такий доступ), цілісності (гарантії точності й повноти інформації) та доступності (гарантії того, що уповноважені користувачі матимуть доступ до інформації й до пов'язаних із нею ресурсів).

Об'єктами впливу з метою порушення конфіденційності, цілісності або доступності інформації можуть бути не лише елементи інформаційної системи, але й підтримувальна інфраструктура (рис. 8.3), зокрема: мережі інженерних комунікацій (системи електро-, теплопостачання, вентиляції, кондиціонування тощо). Крім того, слід звертати увагу на територіальне розташування технічних засобів, які слід розміщувати на території, що підлягає охороні. Безпроводове обладнання рекомендують установлювати так, щоб зона дії безпроводової мережі не виходила за межі контрольованої зони.

На рис. 8.3 показано можливі канали витоку інформації за допомогою відповідних засобів розвідки через систему опалення, стіни приміщення, мережу електроживлення, телефонний кабель, Wi-Fi мережу.

На рис. 8.4 наведено узагальнену схему можливих каналів витоку інформації, з якою працюють співробітники підприємства в типовому одноповерховому офісі. Тут передбачено можливість використання зловмисниками різноманітних пристроїв розвідки (закладок) у середині та ззовні офісу.

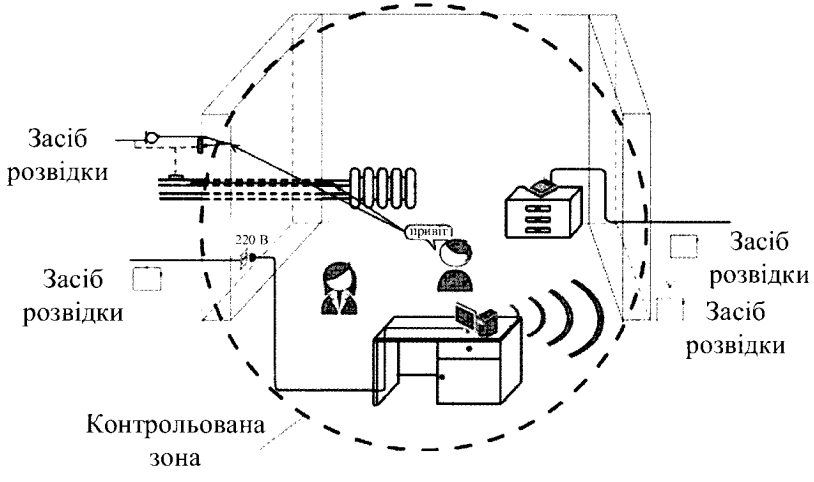


Рис. 8.3. Можливі канали витоку інформації за межі контрольованої зони

Загалом порушник (зловмисник) може отримати доступ до конфіденційної інформації підприємства, використовуючи:

1. Витік інформації через стіни й перекриття, зокрема за допомогою стетоскопа, прокольного мікрофона.
2. Зчитування інформації зі стрічки принтера, роздруківок тощо.
3. Зчитування інформації з використанням відеозакладок.
4. Програмно-апаратні закладки в ПЕОМ.
5. Радіозакладки в стінах і меблях.
6. Зчитування інформації через систему вентиляції.
7. Лазерне зчитування акустичної інформації з вікон.
8. Виробничі та технологічні відходи.
9. Комп'ютерні віруси, логічні "бомби", програмні закладки, витоки мережею Internet тощо.
10. Зчитування інформації наведенням і високочастотним "нав'язуванням".
11. Дистанційне зчитування відеоінформації (оптичними засобами).
12. Зчитування акустичної інформації з використанням диктофонів.
13. Розкрадання носіїв інформації.
14. Високочастотний канал витоку в побутовій техніці.
15. Зчитування інформації напрямленим мікрофоном.
16. Внутрішні канали витоку інформації (через обслуговуючий персонал).
17. Несанкціоноване копіювання.
18. Витік внаслідок побічного випромінювання терміналу.

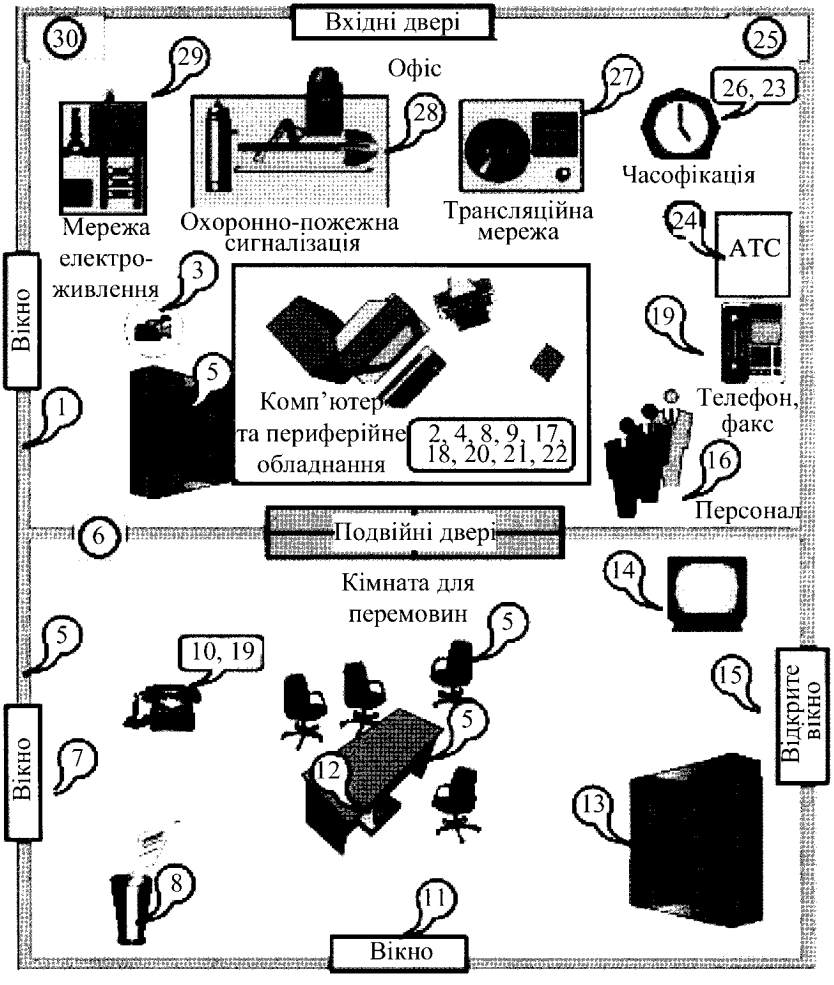


Рис. 8.4. Схема каналів витоку інформації в типовому одноповерховому офісі

19. Зчитування інформації з використанням "телефонного вуха" ("довгого вуха").
20. Зчитування із клавіатури й принтера через акустичний канал.
21. Зчитування з дисплея через електромагнітний канал.
22. Візуальне зчитування з дисплея й принтера.
23. Наведення на лінії комунікацій і сторонні проводи.
24. Витік через лінії зв'язку.
25. Витік колами заземлення.
26. Витік через систему часофікації.

27. Витік через радіотрансляційну мережу й мережу гучномовного зв'язку.

28. Витік через охоронно-пожежну сигналізацію.

29. Витік через мережу електроживлення.

30. Витік через мережі опалення, газо- і водопостачання.

Серед каналів витоку помітну роль відіграють допоміжні засоби, що виходять за межі контрольованої зони, а також сторонні проводи, кабелі, металеві труби систем опалення, водопостачання та інші струмопровідні металоконструкції, що проходять через приміщення, де встановлено основні й допоміжні технічні засоби.

Загалом для захисту інформації в інформаційних системах підприємств та організацій використовують сукупність правових, організаційних, технологічних, фізичних, апаратних та програмних методів і засобів. Розглянемо деякі приклади.

Для ***захисту інформації в інформаційній системі***, до якої входять телекомунікаційна мережа й засоби для накопичування, оброблення, зберігання та розповсюджування інформації (сервери), застосовують такі методи й засоби:

– захист інфраструктури телекомунікаційної мережі (ліній зв'язку, розподільчих пунктів);

– обмеження фізичного доступу до мережевого обладнання;

– обмеження фізичного доступу до серверів;

– забезпечення нормальних кліматичних умов роботи мережевого обладнання та серверів, а також зберігання накопичувачів інформації (дискети, компакт-диски, магнітні диски тощо);

– застосування апаратних засобів ідентифікації у разі доступу до розподільчих пунктів та серверних приміщень інформаційної системи;

– застосування програмних засобів ідентифікації у разі доступу до адміністрування й керування мережевим обладнанням та серверами;

– шифрування таємної чи конфіденційної інформації, яку зберігають на серверах та накопичувачах інформації.

Для ***захисту та збереження інформації на файлових серверах*** інформаційної системи застосовують:

– дзеркальний накопичувач на жорстких магнітних дисках;

– дублювання накопичувача на жорстких магнітних дисках;

– використання масивів накопичувачів на жорстких магнітних дисках;

– захист службових таблиць;

– захищений режим записування на накопичувач на жорстких магнітних дисках;

– використання області Hot Fix у накопичувачі на жорстких магнітних дисках.

– застосування блока безперебійного живлення;

– застосування системи безперебійного електроживлення;

– забезпечення нормальних кліматичних умов роботи серверів;

– обмеження доступу користувачів до даних на серверах.

Захист інформації на персональному комп'ютері можливий за допомогою таких методів і засобів:

– обмеження фізичного доступу до комп'ютера;

– установлення паролю на BIOS комп'ютера;

– установлення механічного замка на системний блок комп'ютера, що перешкоджає його непомітному для власника комп'ютера відкриттю;

– установлення на комп'ютер програми Boot manager, що не дає змоги запустити операційну систему комп'ютера без введення пароля;

– заборона використання дисковода, приводу компакт-дисків і флеш-пам'яті як стартового диска операційної системи;

– установлення пароля на операційну систему комп'ютера;

– установлення пароля на використання ресурсів виділеного сервера в локальній мережі;

– розмежування прав доступу різних користувачів операційної системи;

– обмеження прав запису на жорсткий диск або флеш-пам'ять;

– застосування можливостей розмежування прав доступу мережевої операційної системи;

– архівування файлів архіваторами, установлення пароля на архіви;

– шифрування вмісту дисків накопичувача на жорстких магнітних дисках.

Ураховуючи широкий спектр впливу загроз, для надійного захисту інформації необхідний комплексний підхід, що передбачає створення системи інформаційної безпеки підприємства чи організації.

8.3. Системи інформаційної безпеки

Система інформаційної безпеки – сукупність правових, організаційних, технологічних, фізичних, апаратних та програмних засобів, призначених для забезпечення інформаційної безпеки підприємства, організації, компанії чи корпорації [4–15]. У загальному випадку до складу системи інформаційної безпеки входять системи:

– охоронної сигналізації;

– пожежної сигналізації;

– автоматичного пожежогасіння;
– контролю й управління доступом;
– відеоспостереження (відеонагляду);
– протидії економічному шпигунству;
– безпеки інформаційної системи;
– захисту інформації;
– збирання й опрацювання інформації.

8.3.1. Система охоронної сигналізації

Система охоронної сигналізації призначена для своєчасного сповіщення служби охорони підприємства чи організації, служби позавідомчої охорони (у випадку охорони об'єкта співробітниками позавідомчої охорони або сигналізації на пульт централізованого спостереження позавідомчої охорони) про проникнення (спробу проникнення) порушників у будівлю або на територію, які охороняють.

Система охоронної сигналізації повинна забезпечувати:

– фіксування факту й часу порушення рубежу охоронної сигналізації у разі його подолання порушником (під подоланням рубежу охоронної сигналізації мають на увазі проникнення порушника на територію об'єкта, що охороняють, відкриттям більш ніж на 100 мм або пролому дверей, відкриттям або розбиттям вікон у разі проникнення крізь вікна, руйнування інших будівельних конструкцій, що підлягають облаштуванню засобами охоронної сигналізації, переміщення порушника в зоні дії приладів об'ємного виявлення) з одночасним відображенням інформації на пультах управління й на поверхових планах на моніторі персонального комп'ютера диспетчера із зазначенням ділянки рубежу сигналізації, що спрацьовує;

– установлення й зняття зон з охорони:

1) особистих паролів користувачів із пультів управління, установлених в окремих підрозділах підприємства чи організації;

2) особистих паролів служби охорони з пультів, установлених у приміщенні охорони;

3) особистого пароля оператора системи охоронної сигналізації з автоматизованого робочого місця в приміщенні охорони;

– контроль стану шлейфів, давачів, приладів із відображенням несправностей на моніторі комп'ютера;

– довготривале зберігання інформації про факти порушення рубежу охоронної сигналізації для подальшого відображення чи роздруку на принтері цієї інформації;

– відображення вхідних сигналів: "зламування"; "пожежа"; "напад", "відновлення"; "тест"; "закриття"; "відкриття";

– відображення несправностей системи: відсутність мережі електроживлення, несправність батареї, несправність телефонної лінії, несправність принтера;

– контроль наявності на робочому місці оператора автоматизованого робочого місця з періодичним введенням особистого пароля.

Системи охоронної сигналізації бувають активні та пасивні. Активна система охоронної сигналізації призначена для запобігання несанкціонованому проникненню на об'єкт охорони. Пасивна система охоронної сигналізації – це комплекс засобів, які призначені для привертання уваги охоронної служби або власника об'єкта охорони, до факту несанкціонованого проникнення на цей об'єкт.

За способом підключення окремих пристроїв у системі розрізняють проводову (зв'язок між усіма пристроями здійснюють за допомогою кабельних ліній зв'язку) та безпроводову (зв'язок між усіма пристроями здійснюють за допомогою радіолінії) системи охоронної сигналізації.

Безпроводові системи охоронної сигналізації зручніші в процесі монтажу й експлуатації порівняно із проводовими й можуть додатково містити пристрої дистанційного керування й контролю, проте характеризуються меншою завадозахищеністю.

За типом об'єктів, які охороняють, розрізняють системи охоронної сигналізації, що забезпечують охорону будівлі (приміщення) чи території (периметра).

Переважно ***система охоронної сигналізації складається*** з:

– центрального комп'ютера (сервера);

– приймально-контрольних приладів;

– засобів виявлення (давачів та сповіщувачів);

– засобів повідомлення (сирен, світлових приладів, інших виконавчих пристроїв);

– лінійної частини.

Центральний комп'ютер (сервер) призначений для загального керування системою охоронної сигналізації, автентифікації користувачів системи та визначення їх повноважень, зберігання інформації про факти й час порушення рубежу охоронної сигналізації, фіксування вхідних сигналів та несправностей системи, що надходять від приймально-контрольних приладів.

Приймально-контрольний прилад призначений для приймання й аналізу інформації, що надійшла від давачів чи сповіщувачів, а у випадку виявлення порушення рубежу охоронної сигналізації – подання сигналів на засоби

повідомлення, а також формування сповіщення на пульт служби охорони чи пульт централізованого спостереження служби позавідомчої охорони (через лінії зв'язку). Пульт приймально-контрольного приладу відображує стан охоронної сигналізації, дає змогу здійснити встановлення (зняття) з охорони, налаштування алгоритмів роботи охоронної сигналізації. Ємкість і кількість приймально-контрольних приладів вибирають відповідно до кількості охоронних зон залежно від умов доступу й функційного призначення приміщень.

Засоби виявлення (давачі чи сповіщувачі), які застосовують у системах охоронної сигналізації для охорони будівель (приміщень), розрізняють за призначенням:

– для виявлення розбиття засклених поверхонь – акустичні сповіщувачі, інфрачервоні сповіщувачі;

– для виявлення відкриття вікон і дверей – магнітоконтактні сповіщувачі;

– для виявлення пролому дверей – інфрачервоні сповіщувачі, акустичні сповіщувачі (аналізатори удару), радіохвильові сповіщувачі;

– для охорони об'ємів – інфрачервоні, ультразвукові, комбіновані сповіщувачі;

– для блокування вентиляційних коробів – дріт НВ-0,2 мм, що пропускають у фальшрешітках, виготовлених зі сталевих труб діаметром 8 мм із кроком 100х100 мм;

– для виявлення пролому стін – вібраційні (сейсмічні) сповіщувачі;

– спеціальні давачі для контролю витікання газу чи води.

Для охорони територій (периметра) найчастіше використовують такі засоби виявлення:

– ємнісні сповіщувачі;

– інфрачервоні сповіщувачі;

– ультразвукові сповіщувачі;

– радіохвильові сповіщувачі;

– вібраційні сповіщувачі;

– сповіщувачі тиску;

– сенсорний кабель.

Той або інший тип сигналізації для охорони периметра (або поєднання декількох типів) вибирають залежно від типу загорожі, рослинності, рельєфу місцевості, кліматичних умов, інших чинників.

Засоби повідомлення призначені для повідомлення про порушення рубежу охоронної сигналізації за допомогою звукових (при використанні сирени) або світлових (при застосуванні світлових приладів, зокрема ламп і світильників) сигналів, а також керування іншими системами (наприклад, із використанням виконавчих пристроїв, що у випадку порушення рубежу охоронної сигналізації закривають вхідні двері в приміщення, у якому спрацювала сигналізація). До них також належить таке обладнання:

– блок голосового телефонного додзвонювача, який передає голосове повідомлення на задані телефонні номери;

– модуль для відправлення повідомлень на пейджери із заданими номерами абонентів;

– GSM-модуль для відправлення SMS-повідомлень на задані телефонні номери.

Лінійна частина містить лінії зв'язку, що об'єднують усі компоненти системи охоронної сигналізації.

Засоби виявлення, які використовують у системах охоронної сигналізації. Засоби виявлення (давачі та сповіщувачі) охоронної сигналізації повинні забезпечувати достовірність контролю й високу надійність, якої досягають застосуванням інновацій в області обробки сигналу, поєднанням більш ніж одного принципу виявлення в одному пристрої.

Залежно від принципу виявлення розрізняють такі основні види сповіщувачів (давачів):

– інфрачервоні об'ємні;
– інфрачервоні зі спрямованою діаграмою виявлення;
– інфрачервоні променеві;
– ультразвукові;
– акустичні;
– ємнісні;
– радіохвильові;
– вібраційні;
– магнітоконтактні.

Об'ємний інфрачервоний сповіщувач (пасивний інфрачервоний детектор) уловлює теплове випромінювання. У ньому використовують піроелектричний елемент. Деколи в таких сповіщувачах використовують декілька піроелектричних елементів, щоб забезпечити нечутливість до рівномірного фонового опромінення. Межі чутливого сектора огляду забезпечують завдяки застосуванню лінзи Френеля. Вона складається з безлічі окремих фокусуючих ділянок. Ці ділянки формують свій чутливий промінь, що приходить із певного напряму. У разі переміщення об'єкта теплового випромінювання із сектора в сектор спрацьовує сповіщувач, який передає сигнал тривоги на приймально-контрольний прилад системи охоронної сигналізації. Інфрачервоні сповіщувачі доволі надійні. Їх використовують для охорони житлових будинків, котеджів, квартир. Їх рекомендують застосовувати спільно з іншими типами охоронних сповіщувачів.

Об'ємний інфрачервоний сповіщувач зображено на рис. 8.5.

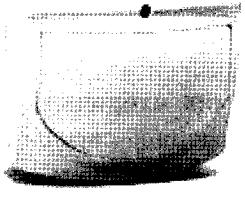


Рис. 8.5. Об'ємний інфрачервоний сповіщувач "Swan Quad"

Інфрачервоний сповіщувач зі спрямованою діаграмою виявлення містить лінзу певного типу. Принцип дії оснований на контролі за тепловим випромінюванням. Існує декілька типів таких сповіщувачів: "завіса", "штора", "коридор". Сповіщувачі зі спрямованою діаграмою виявлення прості в монтажі та характеризуються високим ступенем захисту від помилкового спрацьовування.

Променевий інфрачервоний сповіщувач використовують для охорони великої території. Він складається із приймача й передавача, які розташовані один навпроти одного в зоні прямої видимості. Сповіщувач спрацьовує в разі переривання променя, що потрапляє на приймач. Він дає змогу сформувати вузьку зону виявлення. Для зниження помилкових тривог і підвищення стійкості системи охоронної сигналізації їх роблять дво- або чотирипроменевими. Променеві інфрачервоні сповіщувачі застосовують для охорони периметра, їх установлюють уздовж огорожі.

Ультразвуковий сповіщувач працює за принципом локатора – передає та приймає відбитий переданий сигнал у діапазоні ультразвукових хвиль. Він складається з передавача й приймача. Будь-який рух порушника в зоні дії ультразвукового сповіщувача призведе, згідно із законом Допплера, до зміни довжини хвилі, що слугує сигналом для спрацьовування. Використання ультразвукового сповіщувача забезпечує дуже високу чутливість за високої економічності в системах охоронної сигналізації. Для підвищення стійкості системи до помилкових спрацьовувань передавач і приймач розташовують на одній стіні. Так само передавач не можна розташовувати під прямим кутом до вібруючих поверхонь (двері, вікна) або направляти на місце з найбільшою циркуляцією повітря (сходи, батареї опалювання). Ультразвуковий сповіщувач рекомендують установлювати там, де необхідно контролювати всі входи і виходи приміщення для запобігання несанкціонованому проникненню.

Акустичний сповіщувач (давач розбиття скла, аналізатор удару) спрацьовує при розбитті скла, а також від гучного звуку (наприклад, при ударі). Як чутливий елемент у ньому використано мікрофон. У сучасних акустичних сповіщувачах використовують мікропроцесор, який аналізує різні звуки, а в пам'яті знаходиться база звуків розбиття різних типів стекол: звичайне, армоване, триплекс. Такі акустичні сповіщувачі точно визначають звук розбиття скла, що знижує імовірність помилкового спрацьовування охоронної сигналізації.

На рис. 8.6 зображено малогабаритний, безпроводовий сповіщувач розбиття скла LifeSOS TX-3GS. Сповіщувач призначений для виявлення сильних ударів по склу та розбиття скла на об'єкті охорони. Сповіщувач передає сигнал тривоги радіолінією зв'язку з використанням захищеного протоколу передавання. Максимальна відстань між сповіщувачем і центральним пультом охоронної сигналізації становить 300 м. Відстань виявлення досягає 9 м. Для забезпечення надійності сповіщувач постійно надсилає сигнал тестування на центральний пульт охоронної сигналізації, що унеможливлює непомітне виведення його з ладу.

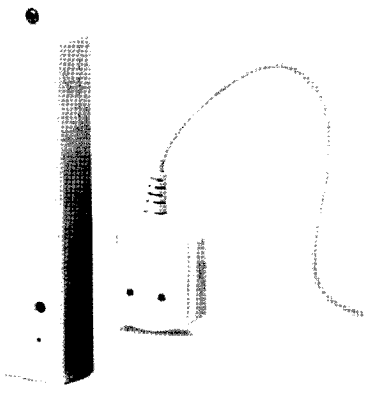


Рис. 8.6. Сповіщувач розбиття скла TX-3GS

Ємнісний сповіщувач складається з одного або декількох металевих електродів. Електроди ізолюють, кріплять уздовж огорожі. Сповіщувач часто виконують у вигляді декоративних ґрат. Секції ґрат ізольовані від основної огорожі й сполучені в загальний контур, до якого підключено електронний блок, що вимірює ємність системи. У разі наближення порушника до електродів ємність системи змінюється. Електронний блок реєструє зміни й видає сигнал тривоги на контрольно-приймальний пристрій системи охоронної сигналізації. Ємнісні сповіщувачі є універсальними й нечутливими до нерівностей рельєфу або лінії огорожі. Їх також використовують для охорони цінного майна, сейфів. Ємнісні сповіщувачі дуже складні в налаштуванні й мають значні габарити.

Об'ємний радіохвильовий сповіщувач працює за принципом виявлення людини за допомогою реєстрування допплерівського зсуву частоти відбитого надвисокочастотного сигналу, який виникає під час руху в електромагнітному полі, що створює надвисокочастотний передавач цього давача. Такі сповіщувачі використовують для виявлення проникнення на об'єкт, що охороняють. Для електромагнітних хвиль діапазону надвисоких частот деякі будівельні матеріали й конструкції не є перешкодою, тому вони з деяким невеликим ослабленням проникають крізь них. Тому зона виявлення радіохвильового сповіщувача може виходити, у деяких випадках, за межі приміщення, що охороняють, що може спричиняти помилкові спрацьовування. До таких матеріалів і конструкцій належать, наприклад, тонкі гіпсокартонні перегородки, вікна, дерев'яні та пластикові двері тощо. Радіохвильові сповіщувачі бувають одно- і двопозиційні. Однопозиційні сповіщувачі застосовують для захисту

об'ємів закритих приміщень і відкритих майданчиків, двопозиційні – для захисту периметрів територій.

Вібраційний сповіщувач має чутливий елемент – сенсорний кабель, який перетворює механічні вібрації на електричний сигнал. Такі сповіщувачі реагують на механічні вібрації й призначені для захисту стін, огорож від руйнування, від відкривання сейфів, розбиття вікон. Вібраційні сповіщувачі доволі складні в налаштуванні й чутливі до зовнішніх джерел вібрації (рух поїздів, робота великих механізмів).

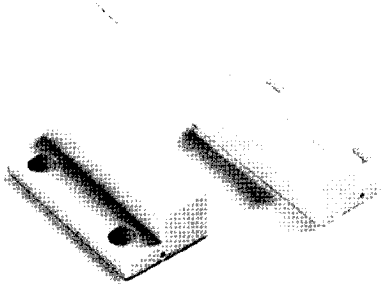


Рис. 8.7. Дротовий магніто-контактний сповіщувач

Магнітоконтактний сповіщувач (з використанням геркона) складається із двох частин – геркона й магніту, які встановлюють один навпроти одного: один на нерухомій частині (наприклад, дверна рама), інший на рухомій (наприклад, дверне полотно) (рис. 8.7). При закритих дверях магніт наближений до геркона, тому контакти геркона замкнуті. Під час відкривання дверей магніт віддаляється від геркона, тому контакти геркона розмикаються, що використовують для формування сигналу тривоги. Цей сигнал від сповіщувача передають на контрольно-приймальний прилад системи охоронної сигналізації. Магнітоконтактні сповіщувачі використовують на вікнах, дверях для контролю їх розкривання. Вони дуже надійні. Магнітоконтактні сповіщувачі рекомендують поєднувати з іншими типами сповіщувачів охоронної сигналізації.

8.3.2. Система пожежної сигналізації

Система пожежної сигналізації призначена для своєчасного сповіщення служби охорони підприємства чи організації (служби пожежної охорони), служби позавідомчої охорони (у випадку охорони об'єкта співробітниками позавідомчої охорони або сигналізації на пульт централізованого спостереження позавідомчої охорони) про виникнення пожежі в будівлі, яку охороняють.

Облаштовують пожежною сигналізацією із забезпеченням цілодобової роботи системи усі пожежонебезпечні приміщення будівлі, за винятком приміщень із "мокрими" процесами або інженерним обладнанням, у яких відсутні матеріали (вентиляційні камери, насосні, бойлерні тощо). Система автоматичної пожежної сигналізації призначена для:

– виявлення осередку пожежі;

– повідомлення про конкретне місце виникнення пожежі черговому персоналу, що цілодобово чергує в приміщенні охорони;

– повідомлення про пожежу в будівлі (загальний сигнал "пожежа" – звуковий і світловий) у приміщення охорони й чергової служби;

– формування сигналу управління системами будівлі під час пожежі.

Система пожежної сигналізації переважно містить:

– центральний комп'ютер (сервер);

– робоче місце оператора пожежної сигналізації з комп'ютером, монітором і принтером для відображення інформації у вигляді поверхових планів на моніторі комп'ютера й друку текстової інформації на принтері;

– станцію пожежної сигналізації або приймально-контрольні прилади;

– засоби виявлення – пожежні сповіщувачі;

– засоби повідомлення (сирени, світлові прилади);

– лінійну частину.

Тип сповіщувачів визначають з урахуванням первинних ознак виникнення пожежі, категорії приміщень і обладнання, що знаходиться в них, меблів, матеріалів за вимогами будівельних норм і правил.

Центральний комп'ютер (сервер) призначений для загального керування системою пожежної сигналізації, зберігання інформації про факти й час виникнення сигналів тривоги пожежної сигналізації, фіксування вхідних сигналів та несправностей системи, що надходять від приймально-контрольних приладів.

Станція пожежної сигналізації повинна забезпечити формування сигналу при пожежі за вимогами будівельних норм і правил для:

– відключення систем кондиціонування й вентиляції з механічним управлінням;

– спрацьовування протипожежних клапанів;

– автоматичного пуску пожежних насосів;

– ввімкнення систем нагнітання повітря в сходові клітки, ліфтові шахти й тамбури-шлюзи;

– опускання ліфтів будівлі на 1-й поверх;

– ввімкнення системи димовідводу, автоматичного відкриття клапанів;

– ввімкнення табло "Вихід";

– ввімкнення сповіщення про пожежу;

– розблокування дверей, обладнаних системою контролю доступу.

Перелік протипожежних систем, кількість необхідних для керування релейних модулів та їх розміщення уточнюють під час проектування інженерних систем.

Приймально-контрольний прилад призначений для приймання й аналізу інформації, що надійшла від давачів чи сповіщувачів, а у випадку виявлення пожежі – подання сигналів на засоби повідомлення, а також формування сповіщення на пульт служби охорони чи пульт централізованого спостереження служби позавідомчої охорони (через лінії зв'язку). Пульт приймально-контрольного приладу відображує стан пожежної сигналізації, дає змогу здійснити встановлення (зняття) з пожежної охорони, налаштування алгоритмів роботи пожежної сигналізації. Ємкість і кількість приймально-контрольних приладів вибирають відповідно до кількості охоронних зон залежно від умов доступу й функційного призначення приміщень.

Засоби виявлення (давачі чи сповіщувачі), які застосовують у системах пожежної сигналізації, бувають такі:

– димові;
– теплові;
– комбіновані;
– ручні;
– аспіраційні;
– сповіщувачі полум'я.

Засоби повідомлення призначені для повідомлення про пожежу за допомогою звукових (із використанням сирени) або світлових (із застосуванням світлових приладів, зокрема ламп і світильників) сигналів, а також керування іншими системами (наприклад, із використанням виконавчих пристроїв, що у випадку пожежі відкривають вхідні двері до приміщення, у якому спрацювала сигналізація). До них також належить таке обладнання:

– блок голосового телефонного додзвонювача, який передає голосове повідомлення на задані телефонні номери;

– модуль для відправлення повідомлень на пейджери із заданими номерами абонентів;

– GSM-модуль для відправлення SMS-повідомлень на задані телефонні номери.

Лінійна частина містить лінії зв'язку, що об'єднують усі компоненти системи пожежної сигналізації.

Засоби виявлення, які використовують у системах пожежної сигналізації. Основним пристроєм, що відповідає за наявність пожежі, є пожежний сповіщувач. Існують сповіщувачі пожежної сигналізації, що діагностують різні властивості довкілля з метою визначення небезпеки або факту займання (пожежі). Нижче розглянуто найпоширеніші пожежні сповіщувачі.

Димовий пожежний сповіщувач призначений для виявлення часток диму, що утворюються в процесі горіння. Цей сповіщувач застосовують для

виявлення тліючої пожежі на ранніх стадіях займання. У димових сповіщувачах використовують спеціальну камеру з оптико-електронним сенсором, який працює за принципом відбиття променя інфрачервоного діапазону хвиль від часток диму (рис. 8.8). Також існують димові сповіщувачі, що реагують на аерозольні продукти горіння.

Димовий оптико-електронний лінійний сповіщувач, оптичний промінь якого проходить поза межами сповіщувача крізь контрольоване середовище, зазвичай називають лінійним сповіщувачем. Він призначений для виявлення часток диму на довгих ділянках; контрольована зона може досягати до 100 метрів. Переважно його використовують для контролю протяжних приміщень із висотою до 12 метрів і більше.

Димовий радіоізотопний пожежний сповіщувач спрацьовує в результаті впливу продуктів горіння на струм іонізації робочої камери сповіщувача.

Тепловий пожежний сповіщувач забезпечує виявлення пожежі в разі швидкого підвищення температури (диференційний принцип виявлення) і/або в разі повільного підвищення температури до максимального значення (максимальний принцип виявлення) (рис. 8.9).

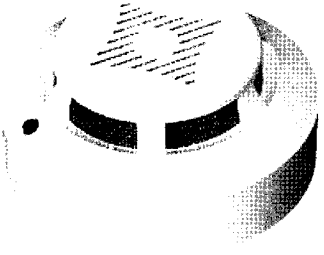


Рис. 8.8. Димовий пожежний сповіщувач СПД-3.2

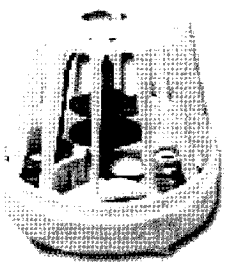


Рис. 8.9. Тепловий пожежний сповіщувач СП-105

Комбінований пожежний сповіщувач – сповіщувач широкого спектра застосування з використанням оптико-електронної сенсорної системи для виявлення часток диму й диференційно-максимального принципу виявлення підвищення температури. Цей сповіщувач забезпечує найвищу надійність виявлення за різних чинників займання.

Ручний пожежний сповіщувач призначений для механічного подавання сигналу тривоги розбиттям скла або відкриттям прозорих дверцят й натисненням тривожної кнопки. Цей тип сповіщувача встановлюють у коридорах, на сходових клітках, біля виходів із будівлі, тобто на всіх шляхах евакуації.

Аспіраційний сповіщувач здійснює хімічний аналіз повітря в приміщеннях. Він складається із системи пластикових трубок з отворами, крізь які примусово здійснюють забори повітря, а потім спеціальний пристрій виконує його хімічний аналіз. Завдяки таким системам можна виявити пожежу на дуже ранній стадії. Часто сповіщувачі цього типу застосовують у випадках, коли порушення інтер'єру неприпустиме. Застосовувати аспіраційні пожежні сповіщувачі зручно у великих будівлях, монтуючи сповіщувачі у вентиляційні короби.

Сповіщувач полум'я призначений для виявлення пожеж, за яких процес горіння не супроводжується виділенням диму: відкрите полум'я горючих рідин або газів, матеріалів, які містять вуглець, таких як деревина, пластмаса, гази, нафтопродукти тощо. Реагує на оптичне випромінювання відкритого полум'я. Також існують модифікації сповіщувача полум'я, що реагує на електромагнітне випромінювання вогню.

8.3.3. Система автоматичного пожежогасіння

Система автоматичного пожежогасіння призначена для автоматичного виявлення пожежі за допомогою давачів пожежної сигналізації, видавання звукових і світлових сигналів оповіщення, повідомлення служби охорони підприємства чи організації (служби пожежної охорони) про пожежу, відключення виробничого обладнання й управління подаванням пожежогасильної речовини в приміщення, які захищають від пожежі, в автоматичному, дистанційному чи ручному режимах керування.

До складу системи автоматичного пожежогасіння входять:

– пристрої пожежогасіння;
– сигнально-пускові пристрої;
– пристрої управління;
– оповіщувачі світлові;
– оповіщувачі звукові;
– джерела живлення;
– пристрої ручного пуску.

Системи автоматичного пожежогасіння можна класифікувати за такими основними ознаками, що визначають варіанти їх технічного здійснення:

– за типом пожежогасильної речовини;
– за структурою побудови;
– за способом подавання пожежогасильної речовини до зони пожежі;
– за можливістю гасіння пожежі в одній зоні або одночасно в декількох зонах.

Залежно від типу пожежогасильної речовини система може бути:

– водяна;

– пінна;

– аерозольна;

– газова.

Кожен із цих варіантів, класифікованих за типом пожежогасильної речовини, можна здійснити різними способами:

– із централізованим зберіганням пожежогасильної речовини, подаванням її до зони пожежі за допомогою системи трубопроводів;.

– із децентралізованим зберіганням пожежогасильної речовини в потенційно пожежонебезпечних приміщеннях, обладнаних автоматичною системою пожежогасіння.

8.3.4. Система контролю й управління доступом

Систему контролю й управління доступом використовує служба охорони підприємства чи організації для посилення охорони об'єкта й контролю допуску співробітників до службових та технічних приміщень об'єкта, а також управління евакуаційними дверима в разі аварійних ситуацій.

Вхід і вихід співробітників у дозволені зони доступності здійснюють за персоніфікованими електронними картками-перепустками в автоматичному режимі в дозволений час. Постійні (особисті) картки-перепустки виготовляють для співробітників і видають їм в особисте користування. Код, записаний на картку-перепустку, є незмінним особистим кодом співробітника, з використанням якого він має можливість проходити в дозволені зони доступу й виділені приміщення, і на підставі якого автоматично реєструють проходи.

У разі втрати картки-перепустки забороняють її використовувати.

Усі приміщення, залежно від призначення й характеру здійснюваних у них операцій, поділяють на зони за доступністю.

Система контролю доступу об'єкта повинна забезпечувати:

– доступ у приміщення – за електронною карткою-перепусткою;

– доступ у приміщення – за електронною карткою-перепусткою й кодом, що набирають на клавіатурі зчитувача;

– вихід із приміщення з використанням картки-перепустки або кнопки виходу;

– видавання сигналу тривоги в приміщення охорони у випадку несанкціонованого проникнення в зони доступу (зламування, незакриття дверей; спроба підбирання коду);

– примусове розблокування (з обов'язковим розбиванням захисного скла або автоматичне з пульта оператора) у випадку пожежі або іншої екстреної ситуації дверей евакуаційних виходів, якщо їх оснащено засобами контролю доступу з реєстрацією цих фактів на сервері системи контролю й управління доступом;

– облік, реєстрування й документування фактів проходження співробітників у місцях установлення пристроїв системи контролю й управління доступом із зазначенням дати й часу проходження;

– створення й ведення бази даних на всіх співробітників, із введенням у неї паспортних та інших даних, кольорових фотографій, а також її оперативне коригування;

– доступ до бази даних та журналу подій, а також видавання довідок із них із виведенням на принтер і екран монітора оператора системи на вимогу користувача залежно від рівня доступу; замовник визначає рівні доступу до бази даних і журналу подій системи, а також може змінювати їх у процесі експлуатації системи;

– облік, реєстрування й документування дій оператора;

– резервування журналу подій і бази даних співробітників.

До складу системи контролю й управління доступом входить таке обладнання:

– робоче місце оператора системи контролю й управління доступом, обладнане комп'ютером із монітором і принтером;

– локальні контролери управління й збирання інформації;

– дистанційні або інші зчитувачі;

– кнопки ручного розблокування дверей при виході із приміщень;

– електромагнітні, електромеханічні замки;

– блоки живлення контролерів і замків;

– обладнання й програмне забезпечення для виготовлення карток-перепусток і ведення інформаційної бази даних;

– електронні ключі (картки).

8.3.5. Система відеоспостереження

Система відеоспостереження (відеонагляду) призначена забезпечити службі охорони підприємства чи організації можливість візуального контролю обстановки по периметру об'єкта й у його внутрішніх приміщеннях засобами телевізійної техніки.

Система містить:

– внутрішні й зовнішні (поворотні чи стаціонарні) відеокамери для отримання відеозображення;

– пристрої опрацювання й перетворення відеозображення;

– апаратуру відеозапису й відтворення;

– апаратуру управління й комутації відеосигналів.

Система повинна забезпечувати запис візуальної й службової інформації від відеокамер на відеомагнітофони або накопичувачі на жорстких магнітних дисках, переглядання цієї інформації на телевізійних моніторах або моніторах комп'ютерів.

Типова модель функціонування системи відеоспостереження. Усе обладнання системи повинно працювати цілодобово. Телевізійні сигнали від відеокамер зовнішнього й внутрішнього встановлення через пристрої опрацювання й перетворення відеозображення, зокрема мультиплексори (пристрої, що дають змогу переглядати на екрані монітора одночасно зображення з 16 відеокамер), квадратори (перегляд зображення з 4 відеокамер) надходять на вхід матричного комутатора.

Переглядають зображення з відеокамер на моніторах у режимах: повноекранне зображення, послідовне перемикання відеокамер, перемикання відеокамер за програмою, квадрозображення, поліекранне зображення. Перемикають відеокамери із системної клавіатури контролера.

Поточний відеозапис зображення з усіх відеокамер здійснюють в 24-годинному мультиплексному режимі за допомогою відеомагнітофонів або накопичувачів на жорстких магнітних дисках. При цьому зображення із частини відеокамер записують зі зниженою частотою вибірки кадру, але при активізації діяльності в зоні їх спостереження частоту запису збільшують. Зображення з відеокамер, що вмикаються від джерела тривоги, записують протягом певного часу, припиняючи процес вручну або автоматично після певного проміжку часу.

Оператори основного поста охорони можуть контролювати зображення із будь-якої відеокамери. З інших постів охорони контролюють зображення із відеокамер, закріплених за цим постом.

На рис. 8.10 зображено IP-відеокамеру D-Link DCS-6112, яка здатна передавати відеопотік високої роздільної здатності та здійснювати його стиснення у форматі H.264, що дає змогу записувати зображення високої чіткості. При підключенні до комп'ютерної мережі (наприклад, Інтернету) камера може транслювати високоякісне відео через таку мережу. Підключення до мережі здійснюють, наприклад, через безпроводову точку доступу Cisco WAP4410N-G5 (рис. 8.11).

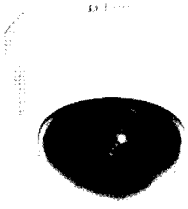


Рис. 8.10. IP-відеокамера D-LinkDCS-6112

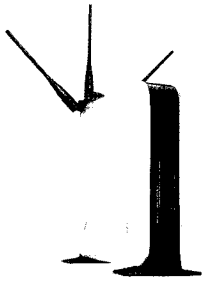


Рис. 8.11. Cisco WAP4410N-G5

8.3.6. Система протидії економічному шпигунству

Система протидії економічному шпигунству повинна забезпечувати службі охорони підприємства чи організації можливість виявлення й локалізації закладних пристроїв, вогнепальної чи холодної зброї, вибухових речовин, радіоактивних речовин і пристроїв, що містять речовини, які приховано переносить людина на собі або в ручній поклажі.

До складу системи входять пункти огляду й контролю на входах у будівлю (в'їздах на територію), які обладнують:

– стаціонарними металошукачами;

– стаціонарними пристроями детектування системи виявлення радіоактивних забруднень;

– ручними металошукачами;

– ручними дозиметрами;

– переносними рентген-телевізійними інтроскопами;

– пристроями локалізації вибухових речовин;

– комплектами для візуального огляду автомобілів.

Система повинна забезпечувати світлову й звукову сигналізацію виявлення небезпечних предметів. Пункти огляду й контролю додатково обладнують засобами зв'язку із черговою частиною й засобами охоронної сигналізації.

8.3.7. Система безпеки інформаційної системи

Система безпеки інформаційної системи призначена для захисту інформаційних потоків усередині інформаційної системи та захисту засобів управління компонентами інформаційної системи від несанкціонованого доступу. Основними завданнями цієї системи є:

– захист периметра інформаційної системи;

– захист локальної мережі;

– захист вузлів локальної мережі.

Поставлені завдання виконують переважно за допомогою організаційних технологічних, апаратних та програмних засобів, зокрема таких: конфігурування активного мережевого обладнання й адміністрування операційних систем серверів; регламентування повноважень системних (мережевих) адміністраторів; розроблення набору обов'язкових інструкцій, що визначають порядок доступу й роботи в системі; застосування певних технологій виконання мережевого адміністрування, моніторингу й аудиту безпеки інформаційних ресурсів; ведення електронних журналів реєстрування користувачів; фільтрування й антивірусне оброблення прийнятої інформації тощо.

Для виконання поставлених завдань на підприємстві чи в організації, що має власну інформаційну систему, створюють окремий підрозділ – ***службу безпеки інформаційної системи***. Створюючи цю службу, враховують розміри інформаційної системи, наявність реальних чи потенційних загроз безпеці інформаційної системи, бажані результати від створення цієї служби. Види діяльності служби безпеки інформаційної системи повинні збігатися з напрямками здійснення політики безпеки підприємства, яка, своєю чергою, має відображати інтереси підприємства.

Служба безпеки інформаційної системи забезпечує відображення основних положень прийнятої політики безпеки підприємства чи організації у відповідних інструкціях і розпорядженнях. У них насамперед визначають:

– посадові обов'язки системних (мережевих) адміністраторів;

– посадові обов'язки груп користувачів;

– правила доступу (розмежування доступу) при конфігуруванні активного мережевого обладнання й адмініструванні операційних систем серверів;

– правила доступу (розмежування доступу) до інформації;

– заходи щодо забезпечення контролю та функціонування системи безпеки інформаційної системи;

– заходи реагування на порушення режиму безпеки;

– планування та організовування відновлювальних робіт.

Для успішної роботи системи безпеки інформаційної системи необхідно визначити права й обов'язки служби безпеки інформаційної системи, а також правила її взаємодії з іншими підрозділами з питань захисту інформації на об'єкті підприємства чи організації.

Організаційно-правовий статус служби безпеки інформаційної системи визначають такими положеннями:

– чисельність служби повинна бути достатньою для виконання всіх перерахованих вище функцій;

– служба повинна бути підпорядкованою тій особі, яка на цьому підприємстві несе персональну відповідальність за дотримання правил захисту інформації;

– штатний склад служби не повинен мати інших обов'язків, пов'язаних із функціонуванням інформаційної системи;

– співробітники служби повинні мати право доступу до всіх приміщень, де встановлено обладнання інформаційної системи й право припиняти автоматизоване оброблення інформації за наявності безпосередньої загрози для інформації, яку захищають;

– керівнику служби має бути надано право забороняти включення до складу інформаційної системи компонентів, якщо вони не відповідають вимогам захисту інформації;

– службі безпеки інформаційної системи повинні бути забезпечені всі умови, необхідні для виконання своїх функцій.

Основною посадовою особою в службі безпеки інформаційної системи є ***адміністратор безпеки***, який повинен володіти відповідними навичками, мати необхідні повноваження та засоби для виконання своїх посадових обов'язків.

Слід зауважити, що все ж інколи керівники підприємств чи організацій вважають за потрібне створювати підрозділ, який, крім функцій захисту інформаційних потоків усередині інформаційної системи та захисту засобів управління компонентами інформаційної системи від несанкціонованого доступу, виконує функції адміністрування й управління інформаційною системою, адміністрування користувачів інформаційної системи, оскільки ці функції є взаємопов'язаними.

8.3.8. Система захисту інформації

Система захисту інформації є сукупністю організаційних і технологічних методів та засобів, які перешкоджають несанкціонованому доступу до таємної або конфіденційної інформації.

Служба захисту інформації (або режимно-секретний відділ) – підрозділ підприємства чи організації, який забезпечує захист інформації керуванням системою захисту інформації. Метою створення такої служби є організаційне забезпечення завдань керування системою захисту інформації в інформаційній системі та контроль за її функціонуванням. На службу захисту інформації покладають виконання робіт із визначення вимог із захисту інформації в інформаційній системі, проектування, розроблення й модернізації системи захисту інформації, а також з експлуатації, обслуговування, підтримування працездатності системи захисту інформації, контролю за станом захищеності інформації в інформаційній системі.

На підприємствах чи в організаціях, де штатним розкладом не передбачено створення служби захисту інформації, заходи щодо забезпечення

захисту інформації в інформаційній системі здійснюють призначені наказом керівника підприємства чи організації працівники. У цьому випадку посадові обов'язки цих працівників повинні містити положення, які б передбачали виконання ними вимог щодо діяльності служби захисту інформації.

Службу захисту інформації обов'язково створюють, коли підприємство чи організація обробляє інформацію, вимогу щодо захисту якої встановлено законодавством. Роботу цієї служби регламентовано в НД ТЗІ 1.4-001-2000 "Типове положення про службу захисту інформації в автоматизованій системі".

Власник інформації особисто визначає не лише перелік інформації, яка належить захисту, але й відповідні методи та засоби захисту. Одночасно він розробляє заходи матеріального й морального стимулювання співробітників, які дотримуються порядку захисту цінної інформації, а також регламентує ступінь відповідальності персоналу за розголошення певних таємниць підприємства.

Система захисту інформації повинна бути багаторівневою з ієрархічним доступом до інформації, гранично конкретизованою й прив'язаною до специфіки підприємства за структурою методів та засобів захисту, що використовують, можливою для регулярного оновлення, надійною як у звичайних, так і в надзвичайних ситуаціях. Вона не повинна створювати співробітникам підприємства серйозні незручності в роботі. Система захисту інформації ґрунтується на використанні правових, організаційних, технологічних, фізичних, апаратних та програмних методів та засобів захисту інформації.

Правовий захист інформації передбачає наявність у засновницьких та організаційних документах підприємства чи організації, контрактах, що укладають зі співробітниками, і в посадових інструкціях положень та зобов'язань із захисту відомостей, що складають таємницю підприємства та його партнерів, формулювання й доведення до відома всіх співробітників підприємства механізму правової відповідальності за розголошення таємної або конфіденційної інформації. Також може бути передбачено страхування інформації від різних ризиків.

Організаційний, технологічний та фізичний захист інформації полягає у виробленні й застосуванні заходів управлінського та обмежувального характеру, які спонукають персонал дотримуватися правил захисту таємної чи конфіденційної інформації, зокрема: формування й регламентування діяльності служби безпеки інформації підприємства; забезпечення цієї служби нормативно-методичними документами з організації й технології захисту інформації; регламентування та регулярне оновлення переліку таємної чи конфіденційної інформації, що підлягає захисту; складання й ведення переліку конфіденційних документів підприємства; регламентування обмеження доступу персоналу до

конфіденційної інформації; регламентування технології захисту й оброблення конфіденційних документів підприємства; побудова захищеного паперового або електронного документообігу; побудова технології документування таємної чи конфіденційної інформації, складання, оформлення, виготовлення і видання конфіденційних документів; побудова технологічної системи оброблення та збереження конфіденційних документів; організовування архівного зберігання таємних та конфіденційних документів; регламентування захисту таємної та конфіденційної інформації підприємства від несанкціонованих дій персоналу; порядок і правила роботи персоналу з таємними та конфіденційними документами й інформацією, контроль за виконанням усіма співробітниками цього порядку і правил; відбір персоналу для роботи з конфіденційною інформацією, навчання та інструктування співробітників; порядок захисту інформації під час переговорів, нарад з конфіденційних питань, приймання відвідувачів, здійснення рекламної, виставкової та іншої діяльності; регламентування аналітичної роботи з виявлення загроз таємній та конфіденційній інформації підприємства і каналів витоку інформації; обладнання й атестування приміщень і робочих зон, виділених для здійснення конфіденційної діяльності, ліцензування технічних систем і засобів захисту інформації та охорони; регламентування роботи з управління системою захисту інформації підприємства чи організації.

Головною проблемою при розробленні методів організаційного захисту інформації є формування дозвільної (обмежувальної) системи і доступу персоналу до конфіденційних відомостей, документів і баз даних. Дозвільна система доступу вирішує такі завдання: забезпечення співробітників усіма необхідними для роботи документами й інформацією; обмеження переліку осіб, які допущені до таємних чи конфіденційних документів. Що вища цінність конфіденційних відомостей, то менша чисельність співробітників повинна їх знати. Відповідно до цієї вимоги визначають необхідний ступінь посилення заходів захисту, перелік рівнів захисту інформації.

Доступ співробітника до таємних чи конфіденційних відомостей, який відповідає дозвільній системі, називають ***санкціонованим***. Дозвіл (санкція) на доступ до цих відомостей завжди є персоніфікованим, і його видає керівник у письмовому вигляді: наказом, що затверджує схему посадового чи іменного доступу до інформації, резолюцією на документі, списком-дозволом у карточці видавання справи або на обкладинці справи ознайомлення з документом. Організаційні заходи захисту відображають у нормативно-методичних документах служби безпеки підприємства.

Апаратний захист інформації передбачає використання: засобів захисту від каналів витоку інформації, що виникають під час роботи комп'ютерів,

засобів зв'язку, копіювальних апаратів, принтерів, факсимільних апаратів та інших приладів і обладнання; засобів захисту приміщень від візуальних та акустичних способів технічної розвідки; засобів охорони будівель і приміщень від проникнення сторонніх осіб (засоби спостереження, сповіщення, сигналізації, інформування й ідентифікації, інженерні споруди); засобів протипожежної охорони; засобів виявлення приладів і пристроїв технічної розвідки (підслуховуючих та передавальних пристроїв, звукозаписувальної та телевізійної апаратури тощо).

Програмний захист інформації ґрунтується на: регламентації доступу до електронних документів персональними паролями; регламентації спеціальних засобів і продуктів програмного захисту; регламентації криптографічних методів і засобів захисту інформації в персональних комп'ютерах та інформаційних мережах, криптографування (шифрування) текстів під час їх передавання каналами зв'язку, під час пересилання поштою.

Система захисту інформації, яку використовує підприємство чи організація, є індивідуалізованою сукупністю необхідних елементів захисту, кожний з яких окремо вирішує свої специфічні для певного підприємства завдання й володіє конкретизованим відносно цих завдань змістом. У комплексі ці елементи забезпечують багатогранний захист таємниць підприємства чи організації та дають відносну гарантію безпеки його діяльності.

8.3.9. Система збирання й опрацювання інформації

Систему збирання й опрацювання інформації переважно виконують у вигляді інтегрованої системи, призначеної для управління системами безпеки будівлі чи території підприємства чи організації й отримання інформації про стан усіх її підсистем. Цю систему використовує служба збирання й опрацювання інформації підприємства чи організації.

Система збирання й опрацювання інформації повинна забезпечувати:

– об'єднання всіх підсистем до єдиної системи;

– автоматичне управління роботою підсистем;

– виконання команд оператора в безпечній для всіх технічних засобів і навколишніх осіб послідовності;

– приймання, реєстрування й відповідне оброблення тривожних сповіщень і сигналів, що надходять від підсистем;

– управління з робочих місць операторів відповідно до їхніх функцій;

– ієрархічний доступ операторів до управління підсистемами, ресурсів та інформації в системі;

– внесення змін, модернізацію, заміну версій програмного забезпечення.

Ця система складається з автоматизованих робочих місць (комп'ютерів) і серверів (основного й резервного), що об'єднані окремою локальною комп'ютерною мережею.

Програмне забезпечення системи збирання й опрацювання інформації дає змогу виводити інформацію про стан кожної підсистеми в графічному вигляді, із зазначенням місць розташування елементів системи та їхнього поточного стану в реальному масштабі часу.

8.4. Створення системи інформаційної безпеки

8.4.1. Концепція створення захищених інформаційних систем

Під час розроблення й побудови ***системи інформаційної безпеки***, що входить до складу інформаційної системи, необхідно дотримуватися певних методологічних принципів проведення досліджень, проектування, створення (виготовлення), експлуатації й модернізації такої системи [4–17]. Системи інформаційної безпеки належать до класу складних систем. Для їх побудови використовують основні принципи побудови складних систем з урахуванням специфіки вирішуваних завдань:

– паралельне розроблення інформаційної системи та системи інформаційної безпеки;

– системний підхід до побудови захищеної інформаційної системи;

– багаторівнева структура системи інформаційної безпеки;

– ієрархічна система управління системою інформаційної безпеки;

– блокова архітектура захищеної інформаційної системи;

– можливість модернізації системи інформаційної безпеки;

– дружній інтерфейс захищеної інформаційної системи з користувачами й обслуговуючим персоналом.

Перший із наведених принципів побудови системи інформаційної безпеки передбачає одночасне паралельне розроблення інформаційної системи й механізмів захисту. Тільки в цьому випадку можна ефективно забезпечити дотримання решти принципів. У процесі розроблення захищеної інформаційної системи необхідно досягати розумного компромісу між створенням вбудованих нероздільних механізмів захисту та блокових уніфікованих засобів і процедур захисту. Тільки на етапі розроблення інформаційної системи можна повністю врахувати взаємний вплив блоків і пристроїв власне інформаційної системи та механізмів захисту, досягати системності захисту оптимальним чином.

Принцип системності є одним з основних концептуальних і методологічних принципів побудови захищених інформаційних систем. Він передбачає:

– аналіз усіх можливих загроз безпеці інформації;

– забезпечення захисту на всіх життєвих циклах інформаційної системи;

– захист інформації у всіх компонентах інформаційної системи;

– комплексне використання механізмів захисту.

Потенційні загрози виявляють у процесі розроблення й дослідження моделі загроз. У результаті досліджень мають бути отримані дані про можливі загрози безпеці інформації, про ступінь їх небезпеки та ймовірності здійснення. Під час побудови системи інформаційної безпеки враховують потенційні загрози, виникнення яких може призвести до істотного збитку та ймовірність яких не є дуже близькою до нуля.

Захищати ресурси інформаційної системи необхідно на етапах розроблення, створення, експлуатації й модернізації, а також під час введення, передавання, оброблення, зберігання й розповсюдження інформації. Дотримання цих принципів дає змогу створити систему інформаційної безпеки, в якій відсутні слабкі ланки як на різних життєвих циклах інформаційної системи, так і в будь-яких елементах і режимах роботи інформаційної системи.

Механізми захисту, які використовують під час побудови захищених систем, мають бути взаємопов'язані за місцем, часом і характером дії. Комплексність передбачає також використання в оптимальному поєднанні різних методів і засобів захисту інформації: правових, організаційних, технологічних, фізичних, апаратних і програмних.

Система інформаційної безпеки повинна мати декілька рівнів, що перекривають один одного, тобто такі системи доцільно будувати за принципом "матрьошки". Щоб дістатися до закритої інформації, зловмисникові необхідно подолати всі рівні захисту.

Системи інформаційної безпеки повинні мати централізоване управління. У розподілених інформаційних системах управління захистом можуть здійснювати за ієрархічним принципом. Централізація управління захистом інформації пояснюється необхідністю проведення єдиної політики в галузі безпеки інформаційних ресурсів у межах підприємства, організації, корпорації, міністерства. Для здійснення централізованого управління в системі інформаційної безпеки мають бути передбачені спеціальні засоби дистанційного контролю, розподілу ключів, розмежування доступу, виготовлення атрибутів ідентифікації тощо.

Одним із важливих принципів побудови захищених інформаційних систем є використання блокової архітектури. Застосування такого принципу дає можливість отримати такі переваги:

– спростити розроблення, налагодження, контроль і перевіряння пристроїв, програм, алгоритмів;

– забезпечити паралельність розроблення блоків;

– використовувати уніфіковані стандартні блоки;

– спростити модернізацію систем;

– забезпечити зручність і простоту експлуатації.

Ґрунтуючись на принципі блокової архітектури захищеної інформаційної системи, можна зобразити структуру ідеальної захищеної системи. У такій системі є мінімальне ядро захисту, що відповідає нижній межі захищеності систем певного класу (наприклад, персональний комп'ютер). Якщо в системі необхідно забезпечити вищий рівень захисту, то цього досягають встановленням спеціальних додаткових апаратних блоків або додаткових програмних засобів. У разі потреби можна використати досконаліші блоки інформаційної системи, щоб не допустити зниження ефективності застосування системи за прямим призначенням. Це пояснюється використанням частини ресурсів інформаційної системи блоками захисту, які додатково встановлюють. Стандартні вхідні й вихідні інтерфейси блоків дають змогу спростити процес модернізації системи інформаційної безпеки, альтернативно використовувати апаратні або програмні блоки.

Розробляючи складну інформаційну систему, необхідно передбачати можливість її модернізації в двох напрямах: збільшення кількості користувачів і нарощування можливостей системи в процесі вдосконалення інформаційних технологій.

Із цією метою під час розроблення інформаційної системи передбачають певний запас ресурсів порівняно з потребами на момент розроблення. Найбільший запас продуктивності необхідно передбачити для найбільш консервативної частини складних систем – каналів зв'язку. Частина резерву ресурсів інформаційної системи може бути затребувана для модернізації системи інформаційної безпеки. Причому механізми захисту, постійно вдосконалюючись, спричиняють необхідність нарощування ресурсів інформаційної системи. Нові можливості, режими інформаційної системи, а також поява нових загроз, своєю чергою, стимулюють розроблення нових механізмів захисту.

Система інформаційної безпеки має бути дружньою стосовно користувачів і обслуговуючого персоналу. Вона має бути максимально автоматизованою й не повинна потребувати виконання користувачем значного обсягу дій, пов'язаних із захистом інформації. Система інформаційної безпеки не повинна створювати обмежень під час виконання користувачем своїх функційних обов'язків. У системі інформаційної безпеки необхідно передбачити заходи зняття захисту із пристроїв, що відмовили, для відновлення їхньої працездатності.

8.4.2. Етапи створення системи інформаційної безпеки

Систему інформаційної безпеки необхідно створювати спільно зі створюваною інформаційною системою. Для побудови системи інформаційної безпеки можна використати існуючі засоби захисту або ж розробити їх спеціально для конкретної інформаційної системи. Залежно від особливостей інформаційної системи, умов її експлуатації й вимог до захисту інформації процес створення системи інформаційної безпеки може не містити окремих етапів, або зміст їх може дещо відрізнятися від загальноприйнятих норм під час створення складних апаратно-програмних систем [1–17]. Але зазвичай створення системи інформаційної безпеки передбачає такі етапи:

– формування загальних вимог до системи інформаційної безпеки в інформаційній системі;

– розроблення політики безпеки інформації в інформаційній системі;

– розроблення технічного завдання на створення системи інформаційної безпеки;

– розроблення проекту системи інформаційної безпеки;

– введення системи інформаційної безпеки в експлуатацію та оцінювання захищеності інформації в інформаційній системі;

– супроводження системи інформаційної безпеки.

Одним із важливих етапів створення системи інформаційної безпеки є етап розроблення технічного завдання, який є основою для розроблення проекту такої системи. Частину процесу створення системи, що закінчується розробленням технічного завдання, називають науково-дослідною розробкою, а частину роботи з розроблення проекту системи називають дослідно-конструкторською розробкою. Дослідно-конструкторську розробку апаратно-програмних засобів виконують із застосуванням систем автоматизованого проектування; алгоритми проектування добре вивчено й відпрацьовано. Тому особливий інтерес становить розгляд виконання науково-дослідної розробки.

8.4.3. Науково-дослідне розроблення системи інформаційної безпеки

Метою цього етапу є розроблення технічного завдання на проектування системи інформаційної безпеки [9–11]. Технічне завдання містить основні технічні вимоги до системи інформаційної безпеки, що розробляють, а також узгоджені взаємні зобов'язання замовника та виконавця розробки. Технічні вимоги визначають значення основних технічних характеристик, виконувані функції, режими роботи, взаємодію із зовнішніми системами тощо.

Етап науково-дослідної розробки виконує замовник, як правило, із залученням науково-дослідних організацій, що спеціалізуються в галузі інформаційної безпеки.

Апаратні засоби оцінюють за такими характеристиками: швидкодія, продуктивність, ємність запам'ятовувальних пристроїв, розрядність, вартість, характеристики надійності тощо. Програмні засоби характеризують необхідним об'ємом оперативної й зовнішньої пам'яті, сумісністю з операційними системами й іншими програмними засобами, часом виконання, вартістю тощо.

На етапі науково-дослідної розробки системи інформаційної безпеки визначають:

– значення основних характеристик системи;

– перелік виконуваних функцій і режимів роботи засобів захисту;

– порядок використання функцій і режимів роботи засобів захисту та взаємодії із зовнішніми системами.

Науково-дослідну розробку починають з аналізу загроз безпеці інформації, аналізу інформаційної системи, яку захищають, і аналізу конфіденційності й важливості інформації в інформаційній системі.

Насамперед аналізують конфіденційність й важливість інформації, яку необхідно обробляти, зберігати й передавати в інформаційній системі. На основі аналізу роблять висновок про доцільність створення системи інформаційної безпеки. Якщо інформація не є конфіденційною й легко може бути відновлена, то створювати систему інформаційної безпеки немає необхідності. Нема сенсу також створювати систему інформаційної безпеки в інформаційній системі, якщо втрата цілісності й конфіденційності інформації пов'язана з незначними втратами. У цих випадках достатньо використовувати штатні засоби інформаційної системи й, можливо, страхування від втрати інформації.

Аналізуючи інформацію, визначають потоки конфіденційної інформації, елементи інформаційної системи, в яких її обробляють і зберігають. На цьому етапі розглядають також питання розмежування доступу до інформації окремих користувачів і цілих сегментів інформаційної системи. На основі аналізу інформації визначають вимоги до її захищеності. Вимоги задають, надаючи певний гриф конфіденційності та установлюючи правила розмежування доступу.

Дуже важливу початкову інформацію для створення системи інформаційної безпеки отримують у результаті аналізу характеристик інформаційної системи, яку захищають. Оскільки система інформаційної безпеки є підсистемою інформаційної системи, то взаємодію цих систем можна вважати внутрішньою, а взаємодію із зовнішнім середовищем – зовнішньою.

Архітектура інформаційної системи визначає внутрішню взаємодію. Під час побудови системи інформаційної безпеки враховують:

– географічне положення інформаційної системи;

– тип інформаційної системи (розподілена або зосереджена);

– структуру інформаційної системи (технічна, програмна, інформаційна тощо);

– продуктивність і надійність елементів інформаційної системи;

– типи використовуваних апаратних і програмних засобів, а також режими їх роботи;

– загрози безпеці інформації, які породжуються всередині інформаційної системи (відмови апаратних і програмних засобів, алгоритмічні помилки тощо).

Враховують такі зовнішні умови, що впливають на зовнішню взаємодію:

– взаємодія із зовнішніми системами;

– випадкові й навмисні зовнішні загрози.

Аналіз загроз безпеці є однією з обов'язкових умов побудови системи інформаційної безпеки. За результатами аналізу розробляють модель загроз безпеці інформації в інформаційній системі. Така модель містить систематизовані дані про випадкові й навмисні загрози безпеці інформації в конкретній інформаційній системі. Припускають наявність відомостей про всі можливі загрози, їх небезпеку, час дії, імовірність здійснення. Часто модель загроз розглядають як композицію моделі порушника (зловмисника) і моделі випадкових загроз. Модель представляють у вигляді таблиць, графів або на вербальному рівні. Для побудови моделі порушника використовують два підходи:

1) модель орієнтується тільки на висококваліфікованого порушника-професіонала, оснащеного всім необхідним, який має право легального проходу всіх рівнів захисту;

2) модель ураховує кваліфікацію порушника, його оснащеність, можливості й офіційний статус в інформаційній системі.

Перший підхід простіший. Він дає можливість визначити верхню межу навмисних загроз безпеці інформації. Другий підхід відрізняється гнучкістю й дає змогу враховувати особливості інформаційної системи повною мірою. Класифікація порушників може бути різною. Наприклад, можна виділити три класи порушників:

– висококваліфікований порушник-професіонал;

– кваліфікований порушник-непрофесіонал;

– некваліфікований порушник-непрофесіонал.

Клас порушника, його оснащеність та офіційний статус на об'єкті інформаційної системи визначають його можливості для здійснення несанкціонованого доступу до ресурсів інформаційної системи.

Загрози, пов'язані з ненавмисними діями, добре вивчені, і велика частина їх може бути формалізована. До них можна зарахувати загрози безпеки, які

пов'язані з обмеженою надійністю технічних систем. Загрози, що породжуються стихією або людиною, формалізувати складніше. Але, з іншого боку, стосовно них накопичено великий об'єм статистичних даних. На підставі цих даних можна прогнозувати прояв загроз цього класу.

Модель порушника та модель випадкових загроз дають змогу отримати повний спектр загроз та їхніх характеристик. У сукупності з початковими даними, отриманими в результаті аналізу інформації та особливостями архітектури проектованої інформаційної системи розроблені моделі загроз безпеці інформації дають можливість отримати початкові дані для розроблення моделі системи інформаційної безпеки.

8.4.4. Моделювання системи інформаційної безпеки

Оцінювання ефективності функціонування системи інформаційної безпеки є складним науково-технічним завданням [9–11]. Систему інформаційної безпеки оцінюють у процесі розроблення інформаційної системи, у період експлуатації й під час створення (модернізації) системи інформаційної безпеки для вже існуючих інформаційних систем. Для розроблення складних систем поширеним методом проектування є синтез із подальшим аналізом. Систему синтезують узгодженим об'єднанням блоків, пристроїв, підсистем і аналізують (оцінюють) ефективність отриманого рішення. З безлічі синтезованих систем вибирають кращу за результатами аналізу, який здійснюють за допомогою моделювання.

Для дослідження ефективності системи інформаційної безпеки використовують моделювання. Моделювання такої системи полягає в розробленні моделі системи, яка з певною точністю відтворює процеси, що відбуваються в реальній системі. Розроблення й подальше використання моделі дає змогу отримувати й досліджувати характеристики реальної системи.

Для оцінювання систем використовують аналітичні й імітаційні моделі. В аналітичних моделях функціонування досліджуваної системи описують у вигляді математичних або логічних співвідношень. Для цієї мети використовують могутній математичний апарат: алгебра, функціональний аналіз, різницеві рівняння, теорія ймовірності, математична статистика, теорія множин, теорія масового обслуговування тощо. Для імітаційного моделювання використовують алгоритми зміни основних характеристик реальної системи відповідно до еквівалентних реальних процесів.

Моделі поділяють також на детерміновані й стохастичні. Моделі, які оперують із випадковими величинами, називають стохастичними. Оскільки на

процеси захисту інформації пепеважно впливають випадкові чинники, то моделі систем захисту є стохастичними.

Моделювання системи інформаційної безпеки є складним завданням, тому що ці системи належать до класу складних організаційно-технічних систем, яким властиві такі особливості:

– складність формального представлення процесів функціонування таких систем, переважно через складність формалізації дій людини;

– різноманіття архітектури складної системи, яке зумовлене різноманіттям структур її підсистем і множинністю шляхів об'єднання підсистем у єдину систему;

– велика кількість взаємопов'язаних елементів і підсистем;

– складність функцій, що виконує система;

– функціонування систем в умовах неповної визначеності й випадковості процесів, що діють на систему;

– наявність безлічі критеріїв оцінювання ефективності функціонування складної системи;

– існування інтегрованих ознак, властивих системі загалом, але не властивих кожному елементу окремо (наприклад, система з резервуванням є надійною за ненадійних елементів);

– наявність управління, що часто має складну ієрархічну структуру;

– розгалуженість і висока інтенсивність інформаційних потоків.

Для подолання цих складнощів застосовують:

– спеціальні методи неформального моделювання;

– декомпозицію загального завдання на низку окремих завдань;

– макромоделювання.

Спеціальні методи неформального моделювання. Спеціальні методи неформального моделювання основані на застосуванні неформальної теорії систем. Основними складовими неформальної теорії систем є:

– структуризація архітектури та процесів функціонування складних систем;

– неформальні методи оцінювання;

– неформальні методи пошуку оптимальних рішень.

Структуризація є розвитком формального опису систем, поширеного на організаційно-технічні системи. Прикладом структурованого процесу є конвеєрне виробництво. В основу такого виробництва покладено два принципи:

– строге регламентування технологічного процесу виробництва;

– спеціалізація виконавців та обладнання.

Передбачають, що конструкція виробленої продукції відповідає таким вимогам:

– виріб складається з конструктивних ієрархічних елементів (блоків, вузлів, схем, деталей тощо);

– максимальна простота, уніфікованість і стандартність конструктивних рішень і технологічних операцій.

Сьогодні процес виробництва технічних засобів інформаційної системи достатньо повно структурований. Структурне програмування також вписується в межі структурованих процесів. На основі узагальнення принципів і методів структурного програмування можна сформулювати умови структурованого опису систем, які досліджують, і процесів їхнього функціонування:

– повнота відображення основних елементів;

– адекватність;

– простота внутрішньої організації елементів опису та взаємозв'язків елементів між собою;

– стандартність та уніфікованість внутрішньої структури елементів і структури взаємозв'язків між ними;

– модульність;

– гнучкість, під якою розуміють можливість розширення й змінювання структури одних компонентів моделі без істотного змінювання інших компонентів;

– доступність для вивчення й використання моделі будь-якому фахівцеві середньої кваліфікації відповідного профілю.

У процесі проектування систем необхідно отримати їхні характеристики. Деякі характеристики можна отримати вимірюванням. Інші отримують із використанням аналітичних співвідношень, а також у процесі оброблення статистичних даних. Проте існують характеристики складних систем, які не можна отримати за такими методами. До таких характеристик системи інформаційної безпеки відносять імовірність здійснення деяких загроз, окремі характеристики ефективності системи тощо.

Такі характеристики можна отримати лише за допомогою ***неформальних методів оцінювання***. Суть методів полягає в залученні для отримання деяких характеристик системи фахівців-експертів у відповідних галузях знань.

Найпоширеніші серед неформальних методів оцінювання методи експертних оцінок. Методом експертних оцінок є алгоритм вибирання фахівців-експертів, задання правил отримання незалежних оцінок кожним експертом і подальшого статистичного оброблення отриманих результатів. Методи експертних оцінок використовують давно, вони добре відпрацьовані. У деяких випадках вони є єдино можливими методами оцінювання характеристик систем.

Неформальні методи пошуку оптимальних рішень можна поділити на дві групи:

– методи неформального зведення складного завдання до формального опису й вирішення завдання формальними методами;

– неформальний пошук оптимального рішення.

Для моделювання систем інформаційної безпеки доцільно використовувати такі теорії й методи, що дають змогу звести вирішення завдання до формальних методів:

– теорія нечітких множин;

– теорія конфліктів;

– теорія графів;

– формально-евристичні методи;

– еволюційне моделювання.

Методи теорії нечітких множин дають змогу отримувати аналітичні співвідношення для кількісних оцінок нечітких умов приналежності елементів до тієї або іншої множини. Теорія нечітких множин добре узгоджується з умовами моделювання систем інформаційної безпеки, оскільки багато початкових даних моделювання (наприклад, характеристики загроз і окремих механізмів захисту) не є строго визначеними.

Теорія конфліктів є відносно новим напрямом дослідження складних людино-машинних систем. Конфлікт між порушником і системою інформаційної безпеки, що розгортається на тлі випадкових загроз, є класичним для застосування теорії конфлікту. Дві протиборчі сторони мають протилежні цілі. Конфлікт розвивається в умовах неоднозначності й слабкої передбаченості процесів, здатності сторін оперативно змінювати цілі. Теорія конфліктів є розвитком теорії ігор. Теорія ігор дає можливість:

– структурувати завдання, подати його в осяжному вигляді, знайти області кількісних оцінок, переваг, виявити переважальні стратегії, якщо вони існують;

– до кінця вирішити завдання, які описують стохастичні моделі.

Теорія ігор дає змогу знайти рішення, що є оптимальним або раціональним у середньому. Вона ґрунтується на принципі мінімізації середнього ризику. Такий підхід не цілком адекватно відображає поведінку сторін у справжніх конфліктах, кожен з яких є унікальним. У теорії конфліктів зроблено спробу подолання цих недоліків теорії ігор. Теорія конфліктів дає можливість вирішувати практичні завдання дослідження складних систем, проте вона ще не набула поширення й потребує подальшого вдосконалення.

З теорії графів для дослідження систем захисту інформації доцільно застосовувати апарат мереж Петрі. Управління умовами у вузлах мережі Петрі дає змогу моделювати процеси подолання захисту порушником. Апарат мереж Петрі дає можливість формалізувати процес дослідження ефективності системи інформаційної безпеки.

До формально-евристичних методів належать методи пошуку оптимальних рішень не на основі строгих математичних, логічних співвідношень, а ґрунтуючись на досвіді людини, наявних знаннях та інтуїції. Отримувані рішення можуть бути далекі від оптимальних, але вони завжди будуть кращі за рішення, отримані не за евристичними методами. Найпоширеніші з евристичних методів лабіринтові й концептуальні методи.

Відповідно до лабіринтової моделі завдання представляють людині у вигляді лабіринту можливих шляхів рішення. Передбачають, що людина володіє здатністю швидкого відсікання безперспективних шляхів руху по лабіринту. У результаті серед шляхів, що залишилися, людина з великою ймовірністю знаходить шлях, що веде до досягнення мети.

Концептуальний метод припускає виконання дій із концептами. Під концептами розуміють узагальнені елементи та зв'язки між ними. Концепти формує людина, можливо й неусвідомлено, у процесі побудови структурованої моделі. Відповідно до концептуального методу набір концептів є універсальним, і йому відповідають механізми обчислення, трансформації й формування стосунків, що є в людини. Людина виконує уявний експеримент зі структурованою моделлю й породжує обмежену ділянку лабіринту, в якій уже нескладно знайти рішення.

Еволюційне моделювання є різновидом імітаційного моделювання. Особливість його полягає в тому, що в процесі моделювання вдосконалюють алгоритм моделювання.

Суть неформальних методів безпосереднього пошуку оптимальних рішень полягає в тому, що людина бере участь не лише в побудові моделі, але й у процесі її використання.

Декомпозиція загального завдання на окремі завдання. Складність виконуваних функцій, значна частка нечітко визначених початкових даних, велика кількість механізмів захисту, складність їхніх взаємних зв'язків і багато інших чинників роблять практично нерозв'язною проблему оцінювання ефективності системи загалом за допомогою одного якого-небудь методу моделювання. Для вирішення цієї проблеми застосовують метод декомпозиції завдання оцінювання ефективності функціонування системи, що передбачає декомпозицію (розділення) загального завдання оцінювання ефективності на низку окремих завдань. Головна ідея оцінювання системи полягає в обліку взаємозв'язку й взаємного впливу окремих завдань оцінювання й оптимізації. Цей вплив ураховують як при вирішенні завдання декомпозиції, так і в процесі отримання інтегральних оцінок. Наприклад при вирішенні завдання захисту інформації від електромагнітних випромінювань використовують екранування металевими екранами, а для підвищення надійності функціонування системи

необхідне резервування блоків, зокрема й блоків, що забезпечують безперебійне живлення. Вирішення цих двох окремих завдань взаємозв'язане, наприклад, при створенні системи інформаційної безпеки на літальних апаратах, де існують строгі обмеження щодо ваги. При декомпозиції завдання оптимізації такої системи доводиться враховувати загальне обмеження щодо ваги обладнання.

Макромоделювання. Для оцінювання складних систем використовують також макромоделювання. Таке моделювання здійснюють для загального оцінювання системи. Завдання при цьому спрощують завдяки використанню для побудови моделі лише основних характеристик. До макромоделювання вдаються переважно для отримання попередніх оцінок системи.

На макрорівні можна, наприклад, досліджувати необхідну кількість рівнів захисту, їхню ефективність відносно передбачуваної моделі порушника з урахуванням особливостей інформаційної системи й фінансових можливостей проектування й побудови системи інформаційної безпеки. Для системи інформаційної безпеки макропоказниками можуть бути: імовірність правильного виявлення проникнення порушника в контрольовану зону, імовірність хибної тривоги.

8.4.5. Вибір показників ефективності та критеріїв оптимальності системи інформаційної безпеки

Ефективність систем оцінюють за допомогою показників ефективності. Стосовно складних людино-машинних систем переважно використовують ***показник ефективності функціонування***, який характеризує ступінь відповідності оцінюваної системи своєму призначенню [9–11].

Прикладом показника ефективності є криптостійкість шифру, яку виражають часом або вартістю зламу шифру. Цей показник, наприклад, для шифру DES залежить від однієї характеристики – довжини ключа.

Для того, щоб оцінити ефективність системи інформаційної безпеки або порівняти декілька систем за їх ефективністю, необхідно задати деяке правило переваги. Таке правило або співвідношення, основане на використанні показників ефективності, називають ***критерієм ефективності***. Для отримання критерію ефективності при використанні деякої множини показників використовують ряд підходів.

Методи, основані на ранжируванні показників за важливістю. Порівнюючи системи, однойменні показники ефективності зіставляють у порядку зменшення їх важливості за визначеними алгоритмами. Прикладами таких методів можуть бути лексикографічний метод і метод послідовних поступок.

Лексикографічний метод доцільний, якщо ступінь відмінності показників за важливістю значний. Дві системи порівнюють спочатку за найважливішим показником. Оптимальною вважають таку систему, у якої цей показник є більшим. За рівності найважливіших показників порівнюють показники, які займають другу позицію за рангом. За рівності й цих показників порівняння триває до отримання переваги за i-м показником.

Оцінювати ефективність системи інформаційної безпеки можуть також із використанням ***методу Парето***. Суть методу полягає в такому. При використанні n показників ефективності системі відповідає точка в n-вимірному просторі. У n-вимірному просторі будують область парето-оптимальних рішень. У цій області розташовують рішення, для яких поліпшення якого-небудь показника неможливе без погіршення інших показників ефективності. Вибір якнайкращого рішення з ряду парето-оптимальних можна здійснювати за різними правилами.

8.4.6. Підходи до оцінювання ефективності системи інформаційної безпеки

Ефективність системи інформаційної безпеки оцінюють як на етапі розроблення, так і в процесі експлуатації. Під час оцінювання ефективності системи інформаційної безпеки, залежно від використовуваних показників і способів їх отримання, можна виділити три підходи [9–11]:

– класичний;
– офіційний;
– експериментальний.

Класичний підхід. Під класичним підходом до оцінювання ефективності розуміють використання критеріїв ефективності, отриманих за допомогою показників ефективності. Значення показників ефективності отримують моделюванням або обчислюють за характеристиками реальної інформаційної системи. Такий підхід використовують для розроблення й модернізації системи інформаційної безпеки. Проте можливості класичних методів комплексного оцінювання ефективності стосовно системи інформаційної безпеки обмежені через низку обставин. Високий ступінь невизначеності початкових даних, складність формалізації процесів функціонування, відсутність загальновизнаних методик розрахунку показників ефективності й вибору критеріїв оптимальності створюють значні труднощі для застосування класичних методів оцінювання ефективності.

Офіційний підхід. Велику практичну значущість має підхід до визначення ефективності системи інформаційної безпеки, який умовно можна

назвати офіційним. Він ґрунтується на використанні для оцінювання системи інформаційної безпеки нормативних актів. Держава здійснює політику безпеки інформаційних технологій, що регламентована в державних нормативних актах. У цих документах визначено вимоги до захищеності інформації різних категорій конфіденційності та важливості.

Вимоги до ефективності системи можна задати переліком механізмів захисту інформації, які необхідно мати в інформаційній системі, щоб вона відповідала певному класу захисту. Такі вимоги описують у відповідних державних нормативних документах. Використовуючи такі документи, можна оцінити ефективність системи інформаційної безпеки. У цьому випадку критерієм ефективності системи інформаційної безпеки є її клас захищеності.

Безперечною перевагою таких класифікаторів (стандартів) є простота використання. Основним недоліком офіційного підходу до визначення ефективності систем захисту є те, що не визначають ефективність конкретного механізму захисту, а констатують лише факт його наявності або відсутності. Цей недолік певною мірою компенсується заданням у деяких документах достатньо докладних вимог до цих механізмів захисту.

У всіх розвинених країнах розроблено свої стандарти захищеності комп'ютерних систем критичного застосування. Наприклад, у міністерстві оборони США використовують стандарт TCSEC (Department of Defence Trusted Computer System Evaluation Criteria), який відомий як Оранжева книга.

Захист інформації в обчислювальних мережах міністерства оборони США організовують відповідно до вимог керівництва "The Trusted Network Interpretation of Department of Defense Trusted Computer System Evaluation Guidelines". Цей документ отримав назву Червона книга (як і попередній – за кольором обкладинки).

Подібні стандарти захищеності інформаційних систем прийнято також в інших розвинених країнах. Так, у 1991 році Франція, Німеччина, Нідерланди й Великобританія прийняли узгоджені "Європейські критерії", в яких розглянуто 7 класів безпеки від Е0 до Е6.

Деякі основні нормативні документи України в галузі захисту інформації розглянуто в розділі 1 цієї книги.

Експериментальний підхід. Експериментальний підхід передбачає експериментальне перевіряння рівня безпеки системи. Для експериментального оцінювання ефективності системи інформаційної безпеки необхідно здійснити експеримент із залученням кваліфікованих фахівців та спеціального обладнання. Перевагою такого підходу є можливість оцінювання дійсного стану системи. Недолік полягає в необхідності значних фінансових затрат на експериментальні дослідження, оскільки вартість залучення кваліфікованих фахівців та

задіювання спеціального обладнання може бути чималою. Іншим недоліком є частковість окремих отриманих результатів, оскільки для отримання узагальнених результатів оцінювання ефективності складних систем інформаційної безпеки необхідно повторити експеримент значну кількість разів.

8.4.7. Проектування системи інформаційної безпеки

Порядок проведення робіт із проектування системи інформаційної безпеки (комплексної системи захисту інформації), що входить до складу інформаційної системи, розглянуто у НД ТЗІ 3.7-003 – 2005 "Порядок проведення робіт із створення комплексної системи захисту інформації в інформаційно-телекомунікаційній системі" [17]. Згідно із цим документом, проект системи інформаційної безпеки розробляють на підставі та відповідно до технічного завдання. Під час розроблення цього проекту обґрунтовують і приймають проектні рішення, які дають змогу виконати вимоги технічного завдання, забезпечити сумісність і взаємодію різних компонентів системи інформаційної безпеки, а також різних методів і засобів захисту інформації [1–17].

Процес проектування системи інформаційної безпеки складається з таких ***етапів***:

– ескізний проект;

– технічний проект;

– робочий проект.

У деяких випадках можливо вилучати етап ескізного проекту, а також поєднувати етапи технічного проекту та робочого проекту в один етап.

Для всіх стадій розроблення проекту системи інформаційної безпеки склад документації визначає технічне завдання на систему, види та зміст – ГОСТ 34.201, НД ТЗІ 2.5-004, документацію на програмні засоби виконують згідно з комплексом стандартів Єдиної системи програмної документації, на технічні засоби – згідно з комплексом стандартів Єдиної системи конструкторської документації.

На ***етапі ескізного проекту*** розробляють попередні проектні рішення системи інформаційної безпеки та, у разі необхідності, її окремих складових частин, а також розробляють, оформляють, узгоджують та затверджують документацію на систему інформаційної безпеки. Зміст та стиль документації повинні бути достатніми для повного опису проектних рішень рівня ескізного проекту.

На цьому етапі визначають: функції системи інформаційної безпеки загалом та функції її окремих складових; склад комплексів технічного захисту інформації від витоку технічними каналами та від спеціальних впливів; склад заходів протидії технічним розвідкам, організаційних, правових та інших

заходів захисту; склад комплексу засобів захисту від несанкціонованого доступу; узагальнена структура системи інформаційної безпеки та схема взаємодії складових частин. Також пропонують попередні технічні рішення, за допомогою яких передбачають здійснення завдань і функцій системи інформаційної безпеки.

На ***етапі технічного проекту*** розробляють проектні рішення системи інформаційної безпеки та документацію на цю систему.

На першому етапі розробляють: загальні проектні рішення, необхідні для виконання вимог технічного завдання на систему інформаційної безпеки; рішення щодо структури системи інформаційної безпеки (організаційної структури, структури технічних і програмних засобів), алгоритми функціонування та умов використання засобів захисту; рішення щодо архітектури комплексу засобів захисту від несанкціонованого доступу та механізмів здійснення, визначених функціональним профілем послуг безпеки інформації. Також здійснюють організаційно-технічні заходи щодо забезпечення послідовності розроблення комплексу засобів захисту від несанкціонованого доступу, архітектури, середовища розроблення, випробувань, середовища функціонування та експлуатаційної документації комплексу засобів захисту від несанкціонованого доступу відповідно до заданих рівнем гарантій забезпечення послуг безпеки згідно з відповідними нормативними документами із захисту інформації.

На другому етапі розробляють, оформляють, узгоджують та затверджують документацію в обсязі, передбаченому технічним завданням на систему інформаційної безпеки. Зміст та стиль документації повинні бути достатніми для повного опису проектних рішень рівня технічного проекту. Також готують та оформляють документацію на постачання засобів захисту або продукції, що містить їх у своєму складі, для комплектації системи інформаційної безпеки. Якщо необхідної продукції немає на ринку засобів захисту, то визначають технічні вимоги (складають технічні завдання) на розроблення відповідних засобів.

Крім цього, на етапі технічного проекту розробляють, оформляють і затверджують завдання на проектування із суміжних питань, які пов'язані зі створенням системи інформаційної безпеки або впливають на умови її функціонування (будівельні, електротехнічні, санітарно-технічні та інші підготовчі роботи).

На ***етапі робочого проекту*** розробляють, оформляють та затверджують робочу та експлуатаційну документацію системи інформаційної безпеки та, у разі необхідності, її окремі складові. Робоча документація містить детальні рішення щодо здійснення технічного проекту системи інформаційної безпеки, щодо забезпечення управління системою інформаційної безпеки та взаємодії її

компонентів, а також документацію, необхідну для тестування, пусконалагоджувальних робіт, випробувань системи інформаційної безпеки.

Також на цьому етапі розробляють засоби захисту інформації, передбачені на етапі технічного проекту, або адаптують готову продукцію до умов функціонування системи інформаційної безпеки.

До складу робочої документації на комплекси технічного захисту інформації від витоку технічними каналами повинні входити схеми розміщення основних технічних засобів інформаційної системи, кабельної інфраструктури, мереж живлення та систем заземлення, які виконують згідно з вимогами відповідних нормативних документів із захисту інформації. При цьому враховують умови їх розміщення й мінімально допустимі відстані між цими засобами та допоміжними технічними засобами (засоби зв'язку, системи та засоби кондиціювання, сигналізації, електроосвітлення, радіомовлення, часофікації тощо), що знаходяться в приміщенні, де розташовано обладнання інформаційної системи, та в суміжних приміщеннях. Зазначені умови розміщення та мінімально допустимі відстані беруть з експлуатаційної документації, яка супроводжує сертифіковані основні технічні засоби. У разі відсутності для основних технічних засобів, що використовують у складі системи інформаційної безпеки, сертифікатів відповідності вимогам із технічного захисту інформації, мінімально допустимі відстані та інші умови розміщення цих засобів визначають за результатами їх спеціальних досліджень.

До складу робочої документації на комплекс засобів захисту від несанкціонованого доступу повинні входити описи процедур установлення та ініціалізації комплексу, налагодження всіх механізмів розмежування доступу користувачів до інформації та апаратних ресурсів інформаційної системи, контролю за діями користувачів, формування та оновлення баз даних захисту, а також контролю цілісності програмного забезпечення та баз даних захисту. Документація робочого проекту повинна містити вихідні дані для внесення їх до баз даних захисту.

Експлуатаційна документація містить опис порядку функціонування системи інформаційної безпеки та інструкції щодо забезпечення цього порядку обслуговуючим персоналом і користувачами, порядку супроводження системи інформаційної безпеки впродовж життєвого циклу інформаційної системи.

Контрольні питання до розділу 8

1. Яка мета використання моделі багаторівневого захисту? Охарактеризуйте кожен її рівень.
2. Охарактеризуйте завдання фізичного захисту на підприємстві.
3. Наведіть основні принципи та методи інженерно-технічного захисту інформації на підприємстві.

4. Що входить до складу інженерно-технічних засобів фізичного захисту?
5. Для чого призначена система відеоспостереження за об'єктами охорони? А засоби охоронної сигналізації?
6. Які Ви знаєте технічні засоби системи фізичного захисту підприємства? Наведіть приклади їх застосування.
7. Якими основними властивостями повинна володіти безпечна інформаційна система згідно із стандартом ISO/IEC 27002?
8. Охарактеризуйте можливості несанкціонованого зчитування інформації відповідними засобами розвідки за межами контрольованої зони приміщення підприємства.
9. Що таке система інформаційної безпеки?
10. Складові системи інформаційної безпеки.
11. Опишіть концепцію створення захищених інформаційних систем.
12. Етапи створення системи інформаційної безпеки.
13. У чому полягає науково-дослідне розроблення системи інформаційної безпеки?
14. Що таке моделювання системи інформаційної безпеки?
15. Як вибирають показники ефективності й критерії оптимальності системи інформаційної безпеки?
16. Опишіть підходи до оцінювання ефективності системи інформаційної безпеки.
17. Проектування системи інформаційної безпеки.

Список літератури до розділу 8

1. Горбатий І. В. Телекомунікаційні системи та мережі. Принципи функціонування, технології та протоколи : навч. посіб. / І. В. Горбатий, А. П. Бондарєв. – Львів : Видавництво Львівської політехніки, 2016. – 336 с.
2. Горбатий І. В. Технічна експлуатація сучасних комплексів зв'язку : навч. посіб. / І. В. Горбатий, О. В. Тимченко. – Львів : Сполом, 2006. – 244 с.
3. Бройдо В. Л. Вычислительные системы, сети и телекоммуникации : учебник для вузов / В. Л. Бройдо. – 2-е изд. – СПб. : Питер, 2004. – 703 с. : ил.
4. Головань С. М. Нормативне забезпечення інформаційної безпеки : підручник / С. М. Головань, О. С. Петров, В. О. Хорошко, Д. В. Чирков, Л. М. Щербак. – К. : Держ. ун-т інформ.-комунікац. технологій, 2008. – 533 с.
5. Лужецький В. А. Основи інформаційної безпеки : навч. посіб. / В. А. Лужецький, А. Д. Кожухівський, О. П. Войтович. – Вінниця : ВНТУ, 2013. – 221 с.
6. ISO/IEC 27002:2013 Information technology — Security techniques — Code of practice for information security controls (second edition) [Електронний ресурс]. – Режим доступу: http://www.iso27001security.com/html/27002.html/
5. Архипов О. Є. Захист інформації в телекомунікаційних мережах та системах зв'язку: навч. посібник / О. Є Архипов, В. М. Луценко, В. О. Худяков. – К.: Політехніка, 2003. – 38 с.
6. Дворський М. Н., Палатченко С. Н. Технічна безпека об'єктів підприємництва : II том / М. Н. Дворський, С. Н. Палатченко. – 2006 [Електронний ресурс]. – Режим доступу: http://biblio.royalwebhosting.net/obschaya-shema-tehnicheskih-kanalov-utechki-40509.html/
7. Jason A. The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice / A. Jason. – Waltham : Syngress, 2014. – 240 p.
8. Інформаційний ресурс. – Режим доступу : http://helpiks.org/6-26903.html/
9. Інформаційний ресурс. – Режим доступу : http://irttri.com/
10. Гарасимчук О. І. Комплексні системи санкціонованого доступу : навч. посіб. / О. І. Гарасимчук, В. Б. Дудикевич, В. А. Ромака. – Львів : Видавництво Львівської політехніки, 2010. – 212 с.

11. Дудикевич В. Б. Захист засобів і каналів телефонного зв'язку : навч. посіб. / В. Б. Дудикевич, В. В, Хома, Л. Т. Пархуць. – Львів : Видавництво Львівської політехніки, 2012. – 212 с.
12. Шаньгин В. Ф. Информационная безопасность компьютерных систем и сетей : учеб. пособ. / В. Ф. Шаньгин. – М. : ИД «ФОРУМ» : ИНФРА-М, 2008. – 416 с. : ил. – (Профессиональное образование).
13. Халяпин Д. Б. Защита информации. Вас подслушивают? Защищайтесь! / Д. Б. Халяпин. – М. : НОУ ШО «Баярд», 2004. – 432 с.
14. Завгородний В. И. Комплексная защита информации в компьютерных системах : учеб. пособ. / В. И. Завгородний. – М. : Логос; ПБОЮЛ Н. А. Егоров, 2001. – 264 с. :ил.
15. Романец Ю. В. Защита информации в компьютерных системах и сетях / Ю. В. Романец, П. А. Тимофеев, В. Ф Шаньгин; под ред. В. Ф. Шаньгина. – 2-е изд., перераб. и доп. – М. : Радио и связь, 2001. – 376 с : ил.
16. Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу : НД ТЗІ 1.1-002-99 : [затверджений наказом Департаменту спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України від 28 квітня 1999 р. № 22 із змінами згідно наказу Адміністрації Держспецзв'язку від 28.12.2012 № 806]. – Київ : ДСТСЗІ СБ України, 1999. – 21 с.
17. Порядок проведення робіт із створення комплексної системи захисту інформації в інформаційно-телекомунікаційній системі : НД ТЗІ 3.7-003 -2005 : [затверджений наказом Департаменту спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України від 8 листопада 2005 р. № 125 із змінами згідно наказу Адміністрації Держспецзв'язку від 28.12.2012 № 806]. – Київ : ДСТСЗІ СБ України, 2005. – 22 с.

ПРЕДМЕТНИЙ ПОКАЖЧИК

Г

Д

Е

Ж

З

І

К

Л

М

Н

О

П

Р

С

Т

У

Ф

Х

Ц

Ч

Ш

Ю

Я

Бобало Юрій Ярославович

Доктор технічних наук, професор, професор кафедри теоретичної радіотехніки та радіовимірювання Інституту телекомунікацій, радіоелектроніки та електронної техніки Національного університету "Львівська політехніка", Заслужений працівник освіти України, ректор Національного університету "Львівська політехніка".

Стаж педагогічної роботи – понад 40 років. Автор понад 260 наукових та науково-методичних праць, серед яких 10 монографій, 12 навчальних посібників, 5 підручників.

Напрям наукових досліджень – теорія електронних кіл та методи забезпечення їхньої відмовостійкості.

Горбатий Іван Володимирович

Доктор технічних наук, доцент, професор кафедри теоретичної радіотехніки та радіовимірювання Інституту телекомунікацій, радіоелектроніки та електронної техніки Національного університету "Львівська політехніка".

Стаж педагогічної роботи – 14 років. Автор понад 200 наукових та науково-методичних праць, зокрема 2 монографій, 3 навчальних посібників, 8 патентів України.

Напрям наукових досліджень – безпека інформаційних систем; підвищення ефективності телекомунікаційних та радіотехнічних систем, зокрема розроблення методів та засобів формування й оброблення сигналів для підвищення ефективності таких систем.

Кіселичник Мирослав Дмитрович

Кандидат технічних наук, доцент, завідувач кафедри теоретичної радіотехніки та радіовимірювання Інституту телекомунікацій, радіоелектроніки та електронної техніки Національного університету "Львівська політехніка".

Стаж педагогічної роботи – понад 50 років. Автор понад 200 наукових та науково-методичних праць, зокрема 3 монографій, 11 навчальних посібників, 3 авторських свідоцтв.

Напрям наукових досліджень – багатокритеріальна оптимізація і багатофакторне математичне моделювання процесів проектування, виробництва та експлуатації радіоелектронних пристроїв, теорія і практика забезпечення їх безвідмовності та якості.

Бондарєв Андрій Петрович

Доктор технічних наук, професор, професор кафедри теоретичної радіотехніки та радіовимірювання Інституту телекомунікацій, радіоелектроніки та електронної техніки Національного університету "Львівська політехніка".

Стаж педагогічної роботи – понад 25 років. Автор понад 150 наукових та науково-методичних праць, зокрема 7 навчальних посібників та патенту України.

Напрям наукових досліджень – завадостійкість пристроїв синхронізації, випадкові процеси, методи забезпечення надійності великих систем.

Войтусік Степан Степанович

Кандидат фізико-математичних наук, доцент, доцент кафедри безпеки інформаційних технологій Інституту комп'ютерних технологій, автоматики та метрології Національного університету "Львівська політехніка".

Стаж педагогічної роботи – 8 років. Автор понад 30 наукових праць, зокрема одного навчального посібника.

Напрям наукових досліджень – безпека кіберфізичних систем, системи банківської безпеки, моделювання фізичних процесів методом Монте-Карло.

Горпенюк Андрій Ярославович

Кандидат технічних наук, доцент, доцент кафедри захисту інформації Інституту комп'ютерних технологій, автоматики та метрології Національного університету "Львівська політехніка".

Стаж педагогічної роботи – 20 років. Автор понад 50 наукових та науково-методичних праць, зокрема двох навчальних посібників.

Напрям наукових досліджень – швидкі методи обчислень криптографічних функцій, розроблення та дослідження генераторів псевдовипадкових чисел, спеціалізовані число-імпульсні функціональні перетворювачі.

Нємкова Олена Анатоліївна

Кандидат фізико-математичних наук, доцент, доцент кафедри безпеки інформаційних технологій Інституту комп'ютерних технологій, автоматики та метрології Національного Університету "Львівська політехніка".

Стаж педагогічної роботи – 14 років. Автор 70 наукових та науково-методичних праць, зокрема двох монографій та одного навчального посібника.

Напрям наукових досліджень – автентифікація електронних пристроїв за внутрішніми електричними шумами, ідентифікація стохастичних процесів і об'єктів.

Журавель Ігор Михайлович

Кандидат технічних наук, старший науковий співробітник, доцент кафедри безпеки інформаційних технологій Інституту комп'ютерних технологій, автоматики та метрології Національного університету "Львівська політехніка".

Стаж педагогічної роботи – 15 років. Автор 81 наукової та науково-методичної праці, зокрема одного навчального посібника.

Напрям наукових досліджень – методи обробки, аналізу та розпізнавання цифрових зображень.

Березюк Богдан Михайлович

Кандидат технічних наук, доцент, доцент кафедри захисту інформації Інституту комп'ютерних технологій, автоматики та метрології Національного університетету "Львівська політехніка".

Стаж науково-педагогічної роботи – 35 років. Автор 88 наукових та науково-методичних праць, зокрема одного навчального посібника, 25 авторських свідоцтв та одного патенту на винаходи. Нагороджений нагрудним знаком "Винахідник СРСР".

Напрям наукових досліджень – захист інформації в комп'ютерних мережах.

Яковенко Євгенія Ігорівна

Кандидат технічних наук, доцент, доцент кафедри електронних засобів інформаційно-комп'ютерних технологій Інституту телекомунікацій, радіоелектроніки та електронної техніки Національного університету "Львівська політехніка".

Стаж педагогічної роботи у вищій школі – 14 років. Автор понад 70 наукових та науково-методичних праць, зокрема одного навчального посібника.

Напрям наукових досліджень – моделювання електромагнітних полів у складних структурах, адміністрування комп'ютерних мереж.

Отенко Віктор Іванович

Кандидат технічних наук, доцент, доцент кафедри захисту інформації Інституту комп'ютерних технологій, автоматики та метрології Національного університету "Львівська політехніка".

Стаж педагогічної роботи – 30 років. Автор 61 наукової та науково-методичної праці, зокрема одного навчального посібника.

Напрям наукових досліджень – технології програмування, захист програмного забезпечення, число-імпульсне функціональне перетворення інформації.

Тишик Іван Ярославович

Кандидат технічних наук, доцент кафедри захисту інформації Інституту комп'ютерних технологій, автоматики та метрології Національного університету "Львівська політехніка".

Стаж педагогічної роботи 17 років. Автор 63 наукових та науково-методичних праць, зокрема трьох навчальних посібників.

Напрям наукових досліджень – вейвлет-перетворення сигналів, системи охоронної сигналізації.

Книги для навчання і роботи!

Горбатий І. В., Бондарєв А. П.

ТЕЛЕКОМУНІКАЦІЙНІ СИСТЕМИ ТА МЕРЕЖІ. ПРИНЦИПИ ФУНКЦІОНУВАННЯ, ТЕХНОЛОГІЇ ТА ПРОТОКОЛИ

Навчальний посібник. – 2016. – 336 с.
ISBN 978-617-607-919-4

Автори прагнули ознайомити читача з новими технічними рішеннями в галузі телекомунікацій, сучасними технологіями передавання даних та мовних сигналів. Розглянуто принципи функціонування, технології та протоколи сучасних телекомунікаційних систем та мереж. Описано принципи функціонування й вимоги до телекомунікаційних систем та мереж наступних поколінь. Детально розглянуто методи формування й оброблення сигналів у телекомунікаційних системах та мережах. Описано методи обчислення частотних та енергетичних характеристик модульованих і кодованих сигналів.

Також у посібнику наведено матеріали, присвячені найважливішим показникам телекомунікаційних систем та мереж – якості та технічній ефективності. Особливу увагу присвячено розгляду питань надійності, достовірності та безпеки телекомунікаційних систем та мереж.

Для студентів вищих навчальних закладів спеціальності 7.05090302 "Телекомунікаційні системи та мережі", спеціальності 172 "Телекомунікації та радіотехніка" та споріднених спеціальностей, а також для тих, хто цікавиться телекомунікаційними й радіотехнічними технологіями.

Климаш М. М., Колодій Р. С.

ТЕЛЕКОМУНІКАЦІЙНІ СИСТЕМИ ПЕРЕДАВАННЯ ІНФОРМАЦІЇ

Навчальний посібник. Львів. – 2018. – 632 с.
ISBN 978-966-941-180-8

Висвітлено основні поняття теорії функціонування та методи побудови багатоканальних систем передавання інформації, наведено основні відомості щодо технологій частотного та цифрового мультиплексування синхронної і асинхронної ієрархій з подальшою передачею лінійними трактами телекомунікаційних мереж, а також розглянуто перспективи розвитку систем передавання у ракурсі новітніх технологій транспортування даних.

Для спеціалістів у галузі розробки та експлуатації телекомунікаційного обладнання.

Дронюк І. М.

ТЕХНОЛОГІЇ ЗАХИСТУ ІНФОРМАЦІЇ НА МАТЕРІАЛЬНИХ НОСІЯХ

Монографія. – 2017. – 200 с.
ISBN 978-966-941-022-1

Монографія присвячена розвитку цифрових методів та засобів захисту інформації на матеріальних носіях. Розвинуто теоретичні основи інформаційних технологій захисту. Застосовано методи теорії диференціальних рівнянь для побудови математичних моделей. Розроблено Ateb-перетворення як узагальнення поняття тригонометричних перетворень. Реалізовано комп'ютерні застосування для задач захисту інформації на матеріальних носіях.

Для наукових працівників, інженерів, аспірантів, студентів та дослідників у сфері розроблення інформаційних технологій захисту.

Дудикевич В. Б. та ін.

ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ ДЕРЖАВИ

Навчальний посібник. – 2017. – 204 с.
ISBN 978-966-941-091-7

Наведено сучасні погляди на стан та забезпечення інформаційної безпеки особистості, суспільства та держави. Інформаційна безпека особистості – це насамперед захищеність психіки і свідомості людини від небезпечних інформаційних впливів: маніпулювання свідомістю, дезінформування, спонукання до самогубства, образ тощо. Інформаційна безпека держави (суспільства) характеризується мірою захищеності держави (суспільства) та стійкості основних сфер життєдіяльності (політики, економіки, науки, техносфери, сфери управління, військової сфери і т. ін.) відносно небезпечних інформаційних впливів, причому як з упровадження, так і добування інформації. Інформаційна безпека держави – це здатність нейтралізувати такі впливи.

Видавництво Львівської політехніки
вул. Ф. Колесси, 4, корп. 23А, м. Львів, 79013
тел. +380 32 2582146, факс +380 32 2582136, http://vlp.com.ua, vmr@vlp.com.ua

НАВЧАЛЬНЕ ВИДАННЯ

Бобало Юрій Ярославович
Горбатий Іван Володимирович
Кіселичник Мирослав Дмитрович
Бондарєв Андрій Петрович
Войтусік Степан Степанович
Горпенюк Андрій Ярославович
Нємкова Олена Анатоліївна
Журавель Ігор Михайлович
Березюк Богдан Михайлович
Яковенко Євгенія Ігорівна
Отенко Віктор Іванович
Тишик Іван Ярославович

ІНФОРМАЦІЙНА БЕЗПЕКА

Навчальний посібник

Редактор *Ольга Дорошенко*
Коректор *Анна Весній*
Технічне редагування *Лілія Саламін*
Комп'ютерне верстання *Наталії Максимюк*
Художник-дизайнер *Марія Іванець*

Здано у видавництво 21.06.2018. Підписано до друку 20.05.2019.
Формат 70×100 1/16. Папір офсетний. Друк офсетний.
Умовн. друк. арк. 39,4. Обл.-вид. арк. 35,9.
Наклад 300 прим. Зам. 180987.

Видавець і виготівник: Видавництво Львівської політехніки.
Свідоцтво суб'єкта видавничої справи ДК № 4459 від 27.12.2012 р.

вул. Ф. Колесси, 4, Львів, 79013
тел. + 380 32 2582146, факс +380 32 2582136
vlp.com.ua, ел. пошта: vmr@vlp.com.ua