МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ВІННИЦЬКИЙ ДЕРЖАВНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

**В.О. Хорошко, О.Д. Азаров, М.Є. Шелест, В.А. Мухачьов,
В.І. Андрєєв, В.П. Щербина, Ю.Є. Яремчук**

КОМП'ЮТЕРНА КРИПТОГРАФІЯ

Київ 2003

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ВІННИЦЬКИЙ ДЕРЖАВНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

**В.О. Хорошко, О.Д. Азаров, М.Є. Шелест, В.А. Мухачьов,
В.І. Андрєєв, В.П. Щербина, Ю.Є. Яремчук**

КОМП'ЮТЕРНА КРИПТОГРАФІЯ

Лабораторний практикум для студентів, що навчаються з напрямків підготовки "Інформаційна безпека" і "Комп'ютерні системи та мережі".

Київ 2003

УДК 681.322:621.391
Х 87

Рецензенти:
В.П. Тарасенко, доктор технічних наук, професор
Л.М. Осинський, доктор технічних наук, професор

Затверджено Ученою радою Інституту інформаційно-діагностичних систем Національного авіаційного університету Міністерства освіти і науки України (протокол № 11 від 17.06.2003 р.)

Хорошко В.О., Азаров О.Д., Шелест М.Є., Андреєв В.І., Мухачьов В.А., Щербина В.П., Яремчук Ю.Є.

Х87 Комп'ютерна криптографія. Лабораторний практикум. – Київ: НАУ, 2003. – 94 с.

У матеріалах лабораторного практикуму розглянуті сучасні методи та засоби криптографічного захисту інформації в лабораторних роботах та характерних практичних задачах.

Рекомендується для студентів і аспірантів, що навчаються з напрямку підготовки "Інформаційна безпека" і "Комп'ютерні системи та мережі" а також для фахівців, що працюють в галузі захисту інформації.

УДК 681.322:621.391

ЗМІСТ

ПЕРЕДМОВА

Розвиток і широке впровадження сучасних інформаційних технологій значно підвищили вразливість інформації, що циркулює в інформаційно-телекомунікаційних системах. Однією з причин цього є масове використання для обробки інформації засобів обчислювальної техніки з програмним забезпеченням, що дозволяє порівняно легко спотворювати, копіювати або знищувати оброблювану інформацію, а також змінювати штатні алгоритми накопичення, оброблення та передавання інформації каналами зв'язку. Розвиток нових інформаційних технологій створив наукові, технологічні та економічні передумови для появи таких понять, як «інформаційна війна» та «інформаційна зброя». Об'єктивні процеси масового впровадження подібних технологій в Україні висувають на перший план інформаційну складову такого загального поняття, як «національна безпека».

Відомим недоліком систем, створених на основі «відкритої архітектури», є принципова можливість доступу до інформації з боку осіб, для яких вона не призначена. Забезпечення адекватної і випереджальної протидії загрозам безпеки інформації є, у кінцевому рахунку, однією з найважливіших умов забезпечення політичної, економічної, оборонної та інших складових національної безпеки держави.

В даний час методи захисту інформації відіграють величезну роль у кредитно-фінансовій сфері, бізнесі, під час зберігання конфіденційної інформації та її передавання незахищеними каналами зв'язку, застосовуються в системах електронного документообігу та електронної комерції, використовуються під час забезпечення безпеки польотів і навіть можуть сприяти для з'ясування істини в ході судового розгляду. Протидія загрозам безпеки інформації повинна здійснюватися комплексно. Захист інформації необхідно забезпечувати на всіх етапах її накопичення, оброблення, зберігання та передавання. Надійно захистити інформацію в зовнішніх каналах зв'язку можна лише за допомогою криптографічних методів, до яких, зокрема, відноситься перетворення інформації з використанням секретних параметрів (шифрування). Криптографічні методи дозволяють, крім забезпечення конфіденційності, забезпечити цілісність та справжність інформації, організувати процедуру автентифікації абонентів, які обмінюються інформацією, реалізувати (без

розголошення) синхронне формування однакових псевдовипадкових даних на обох кінцях лінії зв'язку і т.д. Криптографічні методи базуються на тонких і не до кінця досліджених властивостях математичних об'єктів. У їхній основі лежить ідея використання математичних перетворень, побудова обернених до яких, без додаткових даних, обчислювально нереалізовна. Подібні перетворення називаються криптографічними. Відповідні додаткові дані (якщо вони є секретними параметрами) називаються ключами. Побудова криптографічних перетворень як і реалізація методів, що базуються на їх основі, не тривіальні. Тому знайомство з основами криптографії не тільки необхідно для повноцінної технічної освіти, але й корисно для практичних застосувань, у тому числі, в комерційній і фінансовій сферах.

ВСТУП

Проблема захисту інформації шляхом перетворення, що дозволяє уникнути її сприйняття сторонніми, хвилювала людський розум з давніх часів. Цій проблемі зобов'язана своїм народженням криптологія (kryptos – таємний, logos – наука) – наука, яка вивчає проблеми теорії і практики секретного зв'язку. Вона розділяється на два напрямки – криптографію і криптоаналіз – дисципліни, що у своєму розвитку переслідують прямо протилежні цілі. Криптографія займається пошуком і дослідженням математичних методів перетворення інформації, тобто криптографи намагаються забезпечити безпеку листування, винаходячи все нові і нові системи шифрування повідомлень. Сфера інтересів криптоаналізу – дослідження можливості розшифрування інформації без знання ключів, тобто криптоаналітики вирішують обернену задачу, розкриваючи шифри або підробляючи шифровані повідомлення, заміняючи істинний відкритий текст помилковими даними. Історія криптографії – однолітка історії людської мови. Більш того, спочатку писемність сама по собі була криптографічною системою, тому що в давніх суспільствах нею володіли тільки обрані. Священні книги Давнього Єгипту, Давньої Індії тому приклади. Із широким поширенням писемності криптографія стала формуватися як самостійна наука. Перші криптосистеми зустрічаються вже на початку нашої ери. Так, Цезар у своєму листуванні використовував шифр, який отримав його ім'я. Бурхливий розвиток криптографічні системи отримали в роки першої і другої світових війн. Починаючи з післявоєнного часу і по цей день поява обчислювальних засобів прискорила розробку й удосконалювання криптографічних методів. Чому проблема використання криптографічних методів в інформаційних системах стала в даний момент особливо актуальною? З одного боку, розширилося використання комп'ютерних мереж, зокрема, глобальної мережі Інтернет, якою передаються великі об'єми конфіденційної інформації і яка не допускає можливість несанкціонованого доступу. З іншого боку, поява нових потужних комп'ютерів, технологій мережних і нейронних обчислень, уможливило дискредитацію криптографічних систем, які ще недавно вважалися такими, які практично неможливо розкрити.

У наведених нижче лабораторних роботах основна увага приділена криптографічним методам захисту інформації. Сучасна криптографія містить у собі чотири загальних розділи:

1. Симетричні криптосистеми.
2. Криптографічні системи з відкритим ключем.
3. Криптографічні протоколи.
4. Керування ключами.

Широковідомим є використання криптографічних методів для передавання конфіденційної інформації каналами зв'язку (наприклад, в електронній пошті), під час встановлення справжності повідомлень, що передаються, а також для зберігання інформації (документів, баз даних) у зашифрованому вигляді на зовнішніх носіях. Криптографія дає можливість перетворити інформацію таким чином, що її прочитання (відновлення) можливо лише при знанні ключа.

Інформація, яка підлягає шифруванню і розшифруванню, представляється різними способами, найчастіше, у вигляді текстів, записаних у деякому алфавіті. Ці терміни розуміються так. Алфавіт – скінченна множина використовуваних для кодування інформації знаків. Текст – упорядкований набір з елементів алфавіту. Як приклади алфавітів, використовуваних у сучасних інформаційних системах можна, привести такі:

– алфавіт Z33 – 32 букви російського алфавіту і пропуск;
– алфавіт Z256 – символи, які входять у стандартний код ASCII, KOI-8;
– бінарний алфавіт – $Z2 = \{0,1\}$;
– вісімковий алфавіт або шістнадцятковий алфавіт.

Шифром називається множина обернених перетворень текстів повідомлень, що виробляються з метою приховування від зловмисника (противника) інформації, яка міститься в них. Перетворення тексту (повідомлення) за допомогою конкретно вибраного перетворення називається його шифруванням. Цей процес полягає в тому, що вихідний текст, який носить також назву відкритого тексту, замінюється шифрованим текстом. Процес застосування оберненого перетворення до отриманого шифрованого повідомлення називається розшифруванням. З використанням ключа шифрований текст перетвориться у вихідний. Ключем шифру називається сукупність даних, які визначають вибір

конкретного перетворення з усієї множини перетворень, що реалізуються шифром. По суті, ключ – це секретна інформація, необхідна для безперешкодного шифрування і розшифрування текстів. Найчастіше ключ являє собою послідовність, складену з букв алфавіту. Відновлення відкритих текстів повідомлень за умови, що ключі застосованих перетворень невідомі, називають дешифруванням. Криптографічні системи розділяються на симетричні і асиметричні (з відкритим ключем). У симетричних криптосистемах для шифрування і для розшифрування використовується той самий ключ. У системах з відкритим ключем використовуються два ключі – відкритий і закритий (секретний, особистий), які математично зв'язані один з одним, але не обчислюються один через одного за оглядовий час. Інформація для одержувача шифрується за допомогою відкритого ключа, який доступний усім бажаючим, а розшифровується за допомогою особистого ключа, відомого тільки одержувачу повідомлення. Терміни «розподіл ключів» і «керування ключами» відносяться до процесів системи обробки інформації, змістом яких є складання і розподіл ключів між користувачами. (Очевидно, ключі не повинні бути доступні стороннім). Криптографічні протоколи призначені для забезпечення взаємодії віддалених користувачів, у результаті чого формуються умови для коректного виконання процедур оброблення інформації. При цьому, коректність процедур оброблення інформації не можуть порушити ні сторонні, ні самі користувачі. Прикладом криптографічного протоколу може служити процедура перевірки авторства (справжності) повідомлення на основі електронного підпису. Електронним (цифровим) підписом документа називається результат особливого криптографічного перетворення, зробленого над документом його власником. При отриманні документа з підписом одержувач перевіряє деяке математичне співвідношення, істинність якого може забезпечити тільки власник документа. Роль криптографії полягає в неможливості формування підпису сторонніми, наприклад при внесенні в документ перекручувань.

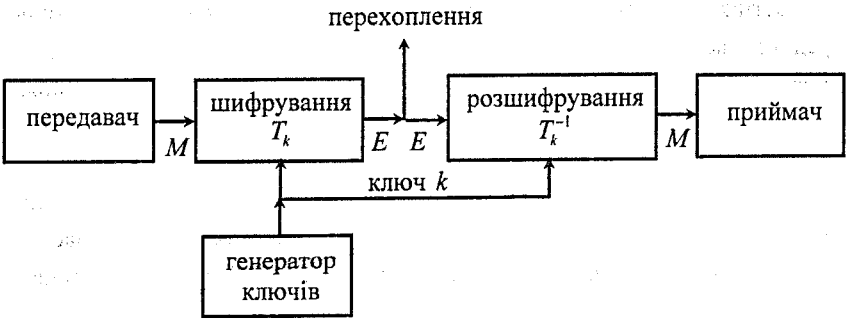
Теоретично, існують шифри, які не піддаються дешифруванню. Поняття стійкості шифру розглядалося К. Шенноном у його роботі «Теорія зв'язку в секретних системах», опублікованій в 1949 році. Прийнято вважати, що ця робота сповістила початок ери наукової криптології. Шеннон назвав «нерозкривні шифри» ідеальними,

зауваживши, однак, що під час їхнього створення виникають нездоланні перешкоди. З цього він зробив висновок, що оцінка стійкості шифрів повинна спиратися на практичну складність їхнього розкриття. Криптостійкістю називається характеристика шифру, яка визначає міру складності його дешифрування. Ефективність шифрування з метою захисту інформації залежить, насамперед, від секретності ключа. У той же час, секретність алгоритму суттєвою не визнається. Є кілька показників криптостійкості, такі, як загальна кількість усіх можливих ключів; середній час, затрачуваний для визначення ключа; необхідний об'єм пам'яті. Процес криптографічного перетворення даних може здійснюватися як програмно, так і апаратно. Апаратна реалізація відрізняється істотно більшою вартістю, однак їй притаманні й переваги: висока продуктивність, простота, захищеність і т.д. Програмна реалізація більш практична і допускає відому гнучкість у використанні. Для сучасних криптографічних систем захисту інформації є загальні вимоги, частина з яких наводиться нижче:

- зашифроване повідомлення повинно піддаватися читанню тільки при наявності ключа;
- кількість операцій, необхідних для визначення ключа шифрування за фрагментом шифрованого повідомлення і відповідного йому відкритого тексту, повинна бути не менша загальної кількості можливих ключів;
- кількість операцій, необхідних для розшифровування інформації шляхом перебору всіх можливих ключів, повинна мати строгу нижню оцінку і виходити за межі можливостей сучасних комп'ютерів (з урахуванням можливості використання мережних обчислень та прогнозу зростання потужності обчислювальних засобів);
- знання алгоритму шифрування не повинно впливати на надійність криптографічного захисту;
- незначна зміна ключа повинна приводити до істотної зміни вигляду зашифрованого повідомлення;
- структурні елементи алгоритму шифрування повинні бути незмінними;
- довжина шифрованого тексту повинна бути близькою довжині вихідного тексту;

– не повинно бути простих і легко встановлюваних залежностей між ключами, послідовно використовуваними в процесі шифрування;
– алгоритм повинен допускати як програмну, так і апаратну реалізацію, при цьому зміна довжини ключа не повинна вести до якісного погіршення алгоритму шифрування.

Нижче показано проходження інформації в системі секретного зв'язку, що відповідає класичній симетричній криптосистемі.



Відкритий текст M передається від передавача до приймача в зашифрованому вигляді. Шифрування проводиться за допомогою оберненого перетворення T_k, що вибирається з скінченної множини відображень за індексом k, який є ключем. Кожному ключу відповідає апріорна ймовірність його вибору. Таким чином, генератор ключів є пристроєм, який вибирає одне з відображень $T_1, T_2, ..., T_m$ з ймовірностями $p_1, ..., p_m$ відповідно. Шифртекст $E = T_k M$ передається незахищеним каналом зв'язку, де він може бути перехоплений. На прийомному кінці шифртекст розшифровується за допомогою оберненого перетворення T_k^{-1}, вибираного за допомогою усе того ж ключа k. Таким чином, криптографічна система є сімейством обернених відображень $\{T_i\}$ множини можливих повідомлень у множину криптограм. При цьому ймовірність зашифрування чергового повідомлення за допомогою відображення T_i дорівнює p_i.

Криптографічну систему часто називають секретною системою, загальною системою або просто системою. Зазвичай вважається, що множина можливих відкритих повідомлень $M_1, ..., M_n$ скінченна і ці повідомлення мають апріорні ймовірності $q_1, ..., q_n$. Дві системи збігаються,

якщо в них однакові множини відкритих повідомлень, криптограм і ключів, причому розподіли ймовірностей ключів рівні. Вибір ключа відбувається випадково, відповідно до розподілу ймовірностей ключів $p_1,\ldots,p_m$.

Під час створення криптографічної системи криптографи зазвичай враховують в яких умовах вона буде використовуватися, на який потік повідомлень розрахований канал зв'язку, за спливанням якого часу інформація втрачає свою цінність, тобто може бути розсекречена та інше. З цього випливає, що застосування дорогих систем шифрування, які мають високу стійкість, в багатьох випадках не виправдано, тобто на криптосистеми зі зниженою стійкістю також є попит.

Цикл лабораторних робіт дозволить познайомитися з деякими системами шифрування на конкретних прикладах, починаючи з таких класичних шифрів, як шифри заміни та перестановки і закінчуючи алгоритмами блокового шифрування та відкритого розподілу ключів, а також особливостями еліптичної криптографії, яка останнім часом бурхливо розвивається. Крім того будуть розглянуті генератори випадкових чисел та сучасні методи і засоби комп'ютерної стеганографії.

Лабораторна робота № 1
ШИФРИ ЗАМІНИ

Мета роботи – вивчити теоретичні основи побудови шифрів заміни, на практиці здійснити створення ключа шифру заміни, провести зашифрування відкритого і розшифрування шифрованого повідомлення. Усвідомити сильні і слабкі сторони шифрів заміни.

Короткі теоретичні відомості

Заміна (підстановка) – це метод шифрування, при якому кожен знак вихідного тексту взаємнооднозначно заміняється шифропозначенням – одним, або декількома знаками деякого набору символів (алфавіту).

Шифр однобуквеної простої заміни – один з найдавніших шифрів. Шифропозначення для нього застосовувались різні – від букв алфавіту до фігурок «танцюючих чоловічків». Давно відомі і його очевидні слабкості – у шифрованому тексті зберігаються всі частотні характеристики відкритого тексту, усі сполучення і повторення. У зв'язку з цим, навіть у художній і науково-популярній літературі наводяться приклади його дешифрування. У найпростішому вигляді даний шифр полягає в тому, що буква переходить у букву, а вхідний і вихідний алфавіти збігаються як множини, тобто з точністю до перестановки. Для зашифрування чергової букви відкритого тексту визначається її номер у вхідному алфавіті і на відповідне місце формовного шифртексту поміщається буква з тим же номером, але вже з вихідного алфавіту. Нехай, наприклад, вхідний, вихідний алфавіти і відкритий текст мають, відповідно, вигляд

ABCDEFGHIJKLMNOPQRSTUVWXYZ
JKLMNOPQRSTUVWXYZABCDEFGHI
TOBEORNOTTOBETHATISTHEQUESTION.

Буква T – двадцята у вхідному алфавіті. У вихідному алфавіті на двадцятому місці знаходиться буква C. Таким чином, першим знаком шифртексту є буква C. Аналогічно, друга буква, яка має у вхідному алфавіті номер 15, заміняється на п'ятнадцяту букву вихідного алфавіту, тобто X. Отже, другою буквою шифрованого тексту є буква X. У підсумку, виходить таке зашифроване повідомлення:

CXKNXAWXCCXKNCQJCRBCQNZDNBCRXE

Наступний приклад являє собою так званий шифр двозначної заміни, у якому кожна буква вхідного алфавіту замінюється взаємно однозначно парою цифр. Для зручності користувача, наприклад, застосовуються таблиці наведеного нижче вигляду.

	0	1	2	3	4
0	V	P	B	A	C
1	Q	N	Z	T	R
2	D	U	X	L	E
3	O	J	S	I	F
4	K	G	Y	H	M
5	W				

Буква, яка підлягає зашифруванню, відшукується в таблиці. На відповідне місце в шифртексті розміщується пара цифр, яка складається з номера рядка і номера стовпця, на перетині яких знаходиться згадана буква. Наприклад, буква *T* знаходиться на перетині першого рядка і третього стовпця. Отже, вона заміняється на пару 13. Аналогічно, буква *O* виглядає як пара цифр 30. Після зашифрування відкритого тексту з попереднього прикладу отримуємо таке шифроване повідомлення:

13300224301411301313300224134303133332134324102124321333301 1.

Неважко уявити собі ускладнення шифру заміни, коли заміні підлягають не поодинокі букви, а пари букв, причому в результаті зашифрування пара знаків відкритого тексту переходить у пару знаків шифртексту відповідно до деякої таблиці.

Розглянуті вище найпростіші приклади шифрів природним чином наводять на думку про ту велику роль, яку відіграє для користувача «технологічність» процесу зашифрування.

Якщо шифр призначений для роботи вручну, то система, яка легко запам'ятовується та не потребує додаткових записів і складних дій, легше освоюється і частіше використовується, а значить, за інших рівних умов, є більш кращою. В прикладі простої заміни, розглянутому вище, легко помітити, що вхідний і вихідний алфавіти відрізняються циклічним

зсувом. Тому алгоритм зашифрування можна записати у вигляді формули: $x_i \equiv a_i + 9 (\mathrm{mod}\, 26)$, де x_i, a_i – букви, відповідно, шифрованого і відкритого текстів, що знаходяться на місцях з тим самим номером i.

Даний варіант шифру простої заміни називається шифром Цезаря, оскільки є відомості, що аналогічним шифром користувався сам Юлій Цезар близько 2000 років тому.

У загальному випадку перетворення відкритого тексту в шифрах заміни задається за допомогою різних таблиць, які носять назву таблиць заміни. Для вказаних шифрів криптографічна система фактично полягає в алгоритмі типу «взяти чергову букву відкритого тексту, знайти її в лівій частині таблиці, вибрати відповідне значення з правої частини і записати його на відповідне місце в шифртекст». Варто підкреслити, що для даної процедури не важливо, в якому алфавіті представлений відкритий текст і яка «начинка» таблиці. В принципі, вона може виявитися порожньою. В останньому випадку найбільш очевидно, що собою являє система шифрування як така. Для того, щоб її «оживити», необхідно щоразу перед зашифруванням криптограми вирішувати конкретне питання: якими даними заповнити таблицю? Відповідь полягає в тому, що ці дані є ключем для шифру заміни і, отже, виникають з генератора ключів.

Залежно від структури таблиці дані можна представити різним чином. Очевидно, що чим компактніше представлення даних, тим краще. Для шифру простої заміни, наприклад, ключем служить будь-яка перестановка букв алфавіту, тому що таблиця заміни за нею будується легко. Для шифру Цезаря ключами можуть служити числа, що задають величину циклічного зсуву алфавіту. В першому випадку кількість ключів дорівнює $2 \cdot 3 \ldots 26 = 26!$. У другому – ключів усього 26. В обох випадках не всі ключі рівноцінні. Наприклад, при вихідному алфавіті, який відрізняється від вхідного перестановкою двох букв, шифрований текст можна прочитати так само легко, як і відкритий. З цієї точки зору прийнято говорити про якість ключів шифрсистеми.

Задамо тепер таку шифрувальну підстановку:

А Б В Г Д Е Ж З И К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ы Ь Э Ю Я
Ю Я В Т О Р Н И К А Б Г Д Е Ж З Л М П С У Ф Х Ц Ч Ш Щ Ы Ь Э

Неважко помітити закономірність у побудові її нижнього рядка. Можна сформулювати і правило її побудови – вона будується як функція від двох ключових параметрів: (В, ВТОРНИК) – букви і слова, яке складається з різних букв, при цьому перший параметр – «буква» визначає місце виписки другого ключового параметра в нижньому рядку підстановки, а відсутні в цьому слові-ключі букви підставляються за абеткою. Такий прийом побудови ключа шифру розповсюджений і носить назву «гаслового» способу. Відповідний ключовий параметр – слово (або група слів) у криптографії іменується «гаслом».

Природно, що гаслові ключі простої однобуквеної заміни з буквеними шифрпозначеннями складають лише малу частку від 30! (для розглянутого випадку) довільних ключових підстановок, але вони зручніші для застосування, оскільки шифрувальник може легко запам'ятати кілька таких ключів. Нижче наведені характерні прийоми побудови інших гаслових ключів простої заміни.

	7	1	9	4	8	2	0	5	6	3
5	Ш	И	Ф	Р	Л	О	З	У	Н	Г
4	А	Б	Г	Д	Е	Ж	З	Л	М	П
3	Т	Х	Ц	Ч	Щ	Ы	Ь	Э	Ю	Я

	6	8	1	9	3	7	0	4	5	2
	В	Т	О	Р	Н	И	К			
4	А	Б	Г	Д	Е	Ж	К	М	П	С
5	С	У	Ф	Х	Ц	Ч	Ш	Щ	Ы	Ь
2	Э	Ю	Я							

Проста заміна тут побудована за допомогою гасел ШИФРЛОЗУНГ і ВТОРНИК шляхом послідовної нумерації «старшинства» букв за розташуванням в алфавіті. У першому з цих прикладів шифрпозначення – двозначні цифрові Ш ⇔ 57, Я ⇔ 33, у другому – різнозначні В ⇔ 6, Т ⇔ 8, ..., К ⇔ 0, але А ⇔ 46, Б ⇔ 48 і т.д. Під час розшифрування шифртексту отриманого застосуванням різнозначного шифру, непорозумінь не виникає, оскільки двозначні шифрпозначення починаються з цифр, які не є окремими шифрпозначеннями. Наприклад,

9	46	40	3	1	40	3	46	57	3	55	7	50	7	51	9
Р	А	З	Н	О	З	Н	А	Ч	Н	Ы	Й	Ш	И	Ф	Р.

Основною слабкістю шифру простої однобуквеної заміни є відображення в частоті шифрпозначення ймовірнісних властивостей букв відкритого тексту. У зв'язку з цим напрошується удосконалення такого

шифру. Можна надати кожній шифрвеличині по кілька шифрпозначень, причому тим більше, чим більша ймовірність появи букви у відкритому тексті. Такий шифр називається шифром пропорційної заміни. Нижче наведений приклад такого шифру і для наочності поруч із шифрвеличинами (ліворуч у шифранті) приведені значення їхніх ймовірностей (у %).

Шифрант

7,9	А	02,19,23,42,48,61,76,89	0,5	Р	08,20,44,79,94
1,7	Б	35,62	5,4	С	01,11,49,65,80
5,0	В	00,12,59,82,97	6,0	Т	04,13,38,41,56,60
1,6	Г	05	2,6	У	36,98
3,1	Д	39,55,93	0,1	Ф	85
8,3	Е	07,09,21,37,53,68,73,87	1,2	Х	92
0,8	Ж	25	0,6	Ц	50
1,6	З	52	1,3	Ч	18
8,9	И	14,17,28,34,46,64,78,83,95	0,8	Ш	47
3,3	К	31,70,91	0,3	Щ	74
4,0	Л	24,40,72,84	2,2	Ы	16,77
3,1	М	30,63,67	1,6	Ь	81
6,9	Н	03,32,45,69,71,88,96	0,2	Э	66
11,0	О	06,10,15,27,33,43,57,75,86,99	0,7	Ю	22
2,8	П	26,58,90	2,0	Я	29,54

Дешифрант

	1	2	3	4	5	6	7	8	9	0
1	С	В	Т	И	О	Ы	И	Ч	А	О
2	Е	Ю	А	Л	Ж	П	О	И	Я	Р
3	К	Н	О	И	Б	У	Е	Т	Д	М
4	Т	А	О	Р	Н	И	Ш	А	С	Л
5	О	З	Е	Я	Д	Т	О	П	В	Ц
6	А	Б	М	И	С	Э	М	Е	Н	Т
7	Н	Л	Е	Щ	О	А	Ы	И	Р	К
8	Ь	В	И	Л	Ф	О	Е	Н	А	С
9	К	Х	Д	Р	И	Н	В	У	О	П
0	С	А	Н	Т	Г	О	Е	Р	Е	В

Якщо під час шифрування уважно стежити за рівномірним використанням усіх варіантів шифрпозначень, то дійсно частоти кожного з 100 шифрпозначень будуть близькими одна до одної і дешифрування сильно ускладнюється.

Такий же ефект досягається і «книжковими шифрами», в яких шифрпозначення складається з номера сторінки, рядка і місця в рядку потрібної букви в обраній книзі, яка є на обох кінцях лінії зв'язку. Можливий і «економний» варіант книжкового шифру, у якому відрізок тексту книги виписується в квадрат 10×10 кліток. При цьому криптограма зазвичай забезпечується посиланням, яке вказує координати початку тексту в книзі.

	1	2	3	4	5	6	7	8	9	0
1	К	Р	И	П	Т	О	Г	Р	А	М
2	М	А	О	Б	Ы	Ч	Н	О	С	Н
3	А	Б	Ж	А	Е	Т	С	Я	М	А
4	Р	К	А	Н	Т	О	М	У	К	А
5	З	Ы	В	А	Ю	Щ	И	М	К	О
6	О	Р	Д	И	Н	А	Т	Ы	Н	А
7	Ч	А	Л	А	Т	Е	К	С	Т	А
8	В	К	Н	И	Г	Е	Н	А	П	Р
9	И	М	Е	Р	М	А	Р	К	А	Н
0	Т	У	С	О	О	Ш	Ф	Х	Ц	Я

Останні місця в таблиці відводяться буквам, що не зустрілися в даному відрізку тексту. Схожі ключі будують іноді і за гаслами:

	1	2	3	4	5	6	7	8	9	0
1	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ы
2	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ы	Ь
3	А	Б	В	Г	Д	Е	Ж	З	И	К
4	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ы	Ь
5	И	К	Л	М	Н	О	П	Р	С	Т
6	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ы
7	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ы	Ь
8	И	К	Л	М	Н	О	П	Р	С	Т
9	К	Л	М	Н	О	П	Р	С	Т	У
0	А	Б	В	Г	Д	Е	Ж	Э	Ю	Я

		1	2	3	4	5	6	7	8	9	0
М	7	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ы
А	1	Т	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ы
Т	9	Ю	Я	А	Б	В	Г	Д	Е	Ж	З
Е	4	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш
М	8	Д	Е	Ж	З	И	К	Л	М	Н	О
А	2	М	Н	О	П	Р	С	Т	У	Ф	Х
Т	0	М	Н	О	П	Р	С	Т	У	Ф	Х
И	5	Б	В	Г	Д	Е	Ж	З	И	К	Л
К	6	Б	В	Г	Д	Е	Ж	З	И	К	Л
А	3	Ц	Ч	Ш	Щ	Ы	Ь	Э	Ю	Я	А

Ключ довільного шифру парної заміни з двозначним шифрпозначенням являє собою досить громіздку таблицю, скажімо, $z \times z$ рядків і стовпців:

	А	Б	У	...	Я
А	РФ	ЛЖ	ЮН		БВ
Б	НГ	АН	ЗК		ЦЮ
в	РГ	ЬУ	НГ		МК
..					
Я	ЛЩ	АВ	ЭУ		ЗО

При цифрових тризначних шифрпозначеннях ситуація залишається такою ж. Виготовлення випадкових пар таблиць шифрант-дешифрант виявляється досить складною справою.

У розглянутих нами прикладах шифрів перетворення повідомлення проводилось поетапно: спочатку перетворювався перший елемент відкритого тексту (буква), потім другий і так далі. Неважко узагальнити цю ситуацію на сполучення трьох, чотирьох і більше знаків. У криптографії послідовні етапи перетворення відкритого тексту називаються тактами шифрування. Елемент, що перетворюється в одному такті шифрування, є найменшою складовою відкритого тексту, яку можна зашифрувати за допомогою даного шифру. Наприклад, під час заміни пар букв на пари, можна зашифрувати сполучення AB, але неможливо зашифрувати букву A або букву B окремо.

На кожному такті шифрування, у випадку простої заміни, діє та ж сама таблиця. У більш складних шифрах на кожному такті може діяти своє перетворення. Це не означає, що для кожного такту необхідний свій ключ. Просто в інших видах шифрсистем ключ визначає і закони формування таблиць перетворень і послідовність вибору цих таблиць.

Протягом одного такту шифрування шифрсистема може перетворити один або кілька знаків, деяку ділянку відкритого тексту, склад, фразу, слово і, в принципі, ціле повідомлення. Шифри заміни перетворять на кожному такті групу символів. Кількість знаків у групі при цьому фіксована і називається значністю групи. Групи значності 2 називаються біграмами, значності 3 – триграмами, значності 4 – чотириграмами і так далі. У загальному випадку групи значності v називаються v-грамами.

Можлива побудова шифру, аналогічного шифрові заміни, коли в такті шифрування можуть перетворюватися групи різної значності. Такою властивістю володіють шифрсистеми, які називаються кодами. Специфічною особливістю кодів є те, що вони оперують не з довільними комбінаціями символів, а зі словами, складами і фразами. У найпростішому випадку код являє собою список, який нагадує словник, у якому кожній відкритій величині (слову, фразі) відповідає кодова група: комбінація символів, яка замінює відповідну величину під час зашифрування. Значність кодових груп постійна. Зазвичай до складу коду входять також деякі допоміжні величини – цифри, розділові знаки та інше.

Очевидно, що код, який дозволяє зашифрувати на рівні слів довільний відкритий текст, повинен мати об'єм, порівняний з докладним словником природної мови, що для практичного застосування незручно, тим більше, якщо врахувати, що коди традиційно відносяться до ручних систем шифрування, тобто не розраховані на використання засобів автоматизації. В цьому зв'язку, під час складання коду основним етапом є вивчення словникового складу майбутнього листування, у результаті чого визначається набір слів і фраз, які підлягають занесенню в список словникових величин. Чим яскравіше виражена специфіка листування, тим компактніше та зручніше у використанні код. На практиці це приводить до «спеціалізації» кодів, тобто до того, що не будь-який відкритий текст може бути якісно зашифрований за допомогою конкретного коду. З іншого боку, перевагою кодів є ущільнення інформації під час зашифрування, оскільки кодові групи, як правило, коротші величин, які вони заміняють. Як і будь-який інший шифр, код може бути тим або іншим чином модифікований. Він може бути ускладнений, наприклад, за рахунок використання різних кодових груп для зашифрування однієї і тієї ж відкритої величини. Вибір кодової групи з відповідного списку здійснюється випадковим чином.

Аналогічна процедура, яка називається рандомізацією, може застосовуватися для перетворення відкритого тексту перед зашифруванням незалежно від системи шифру. Перетворення полягає в тому, що знаки відкритого тексту замінюються на символи іншого алфавіту, більшого за об'ємом, ніж вихідний. У процесі заміни конкретна буква переходить випадковим чином в один із зв'язаних з нею символів. Кількість таких символів для кожної букви різна і пропорційна частоті її зустрічності у вихідному відкритому тексті. В результаті отримується

послідовність, усі знаки якої зустрічаються приблизно з однаковою частотою. Рандомізація може бути введена в будь-яку криптографічну систему і її використання ускладнює розкриття шифру на основі статистичного аналізу.

Порядок виконання роботи

1. Вивчити короткі теоретичні відомості про шифри заміни.
2. Створити ключ і пряму підстановку (таблицю зашифрування) для шифру простої заміни.
3. Створити і зашифрувати деяке повідомлення.
4. Перевірити правильність зашифрування за допомогою програми SIMSUB.
5. Отримати з прямої підстановки (таблиці зашифрування) обернену (таблицю розшифрування).
6. Розшифрувати зашифроване повідомлення.
7. Перевірити правильність розшифрування за допомогою програми SIMSUB.
8. Скласти звіт, додавши туди отримані результати.

Вимоги до звіту

У звіті повинні бути наведені:

– короткі відомості про вивчені види шифру;
– вибраний ключ;
– відкрите повідомлення;
– зашифроване повідомлення;
– порядок зашифрування і розшифрування;
– висновки про властивості і якість шифрперетворення заміною;
– відповіді на контрольні запитання.

Контрольні питання

1. У чому полягає основна суть шифрів заміни?
2. Які відмінності між рівнозначним і різнозначним шифрами заміни? Порівняєте їх за складністю і криптографічною якістю.

3. Які відмітні риси кодів, порівняйте їх із шифрами простої і парної заміни?

4. Поясніть, наскільки криптографічно стійкими є шифри заміни?

5. Якими могли б бути способи спрощення побудови ключів шифрів простої заміни? Чи знижують ці особливості побудови ключів загальну стійкість застосованого шифру?

Лабораторна робота № 2
ШИФРИ ПЕРЕСТАНОВКИ

Мета роботи – вивчити теоретичні основи побудови шифрів перестановки, на практиці здійснити створення ключа шифру перестановки, провести зашифрування відкритого і розшифрування шифрованого повідомлення. Усвідомити сильні і слабкі сторони шифрів перестановки.

Короткі теоретичні відомості

Розглянуті вище шифри здійснювали перетворення відкритого тексту за методом заміни його знаків деякими шифрпозначеннями. Тим часом можливий й інший метод перетворення – перестановка знаків відкритого тексту за заданим правилом. Шифри типу перестановки застосовувались ще з античних часів. Відмінність цього типу шифру від шифрів заміни полягає в тому, що під час зашифрування буква a_i відкритого тексту переходить не у фіксований знак алфавіту, а в іншу букву того ж відкритого тексту, скажемо a_j, у результаті чого букви розташовуються на нових місцях, тобто переставляються. Ключем для даного шифру також служить таблиця заміни, тільки не букв алфавіту, а їхніх індексів (номерів місць) у тексті, який підлягає зашифруванню. У загальному випадку розмір таблиці заміни дорівнює довжині відкритого тексту. Такі таблиці зручно формувати (і записувати) у вигляді так званих підстановок, тому нагадаємо коротенько їхні деякі властивості.

Підстановкою називається взаємно однозначне відображення скінченної множини на себе. Зазвичай підстановки записують у вигляді двох рядків. Верхній рядок є операндом підстановки, а нижній – результатом її дії на операнд. Наприклад, одна з підстановок, що діють на множині з п'яти елементів, може бути записана в так $T = \begin{pmatrix} a & b & c & d & e \\ b & a & d & e & c \end{pmatrix}$.

Елементи скінченної множини завжди можна перенумерувати, отже й операнд будь-якої підстановки можна записати у вигляді $1 \quad 2 \quad \ldots \quad N$, де N – кількість елементів в операнді. Число N називається степенем

підстановки. Зазначена підстановка T є, таким чином, підстановкою п'ятого степеня і може бути записана так: $T=\begin{pmatrix}1&2&3&4&5\\2&1&4&5&3\end{pmatrix}$.

Очевидно, різними підстановками одного степеня можна впливати на вихідну множину послідовно одна за одною. Результат послідовної дії підстановок T_1, T_2 називається їхнім добутком і записується зазвичай T_1T_2. Добуток двох підстановок визначається виходячи з того, що якщо перша підстановка переводить j на місце i, а друга підстановка, переводить k на місце j, то в добутку k переходить на місце i. Таким чином, якщо $T_1=\begin{pmatrix}i\\ \alpha_i\end{pmatrix}$, $T_2=\begin{pmatrix}\alpha_i\\ \beta_i\end{pmatrix}$, то $T_1T_2=\begin{pmatrix}i\\ \beta_i\end{pmatrix}$. Наприклад, при $T_1=\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$ і $T_2=\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$ $T_1T_2=\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$. Легко бачити, що $T_1T_2\neq T_2T_1$, тобто добуток підстановок некомутативний, однак дужки в добутках можна розставляти довільно. Існує «одинична» підстановка I така, що для всіх T: $IT=TI=T$. Для нашого прикладу $I=\begin{pmatrix}1&2&3&4\\1&2&3&4\end{pmatrix}$. Також неважко бачити, що для будь-якої підстановки T існує обернена підстановка T^{-1} така, що $T^{-1}T=TT^{-1}=I$. Дійсно, для побудови T^{-1} досить переставити місцями рядки в підстановці T, а потім упорядкувати стовпці так, щоб числа у верхньому рядку йшли в зростаючому порядку. Перерахованих властивостей цілком достатньо, щоб класифікувати множину підстановок степеня N із зазначеною операцією множення як групу – одну з найважливіших математичних моделей. Відмітимо також, що в деякому розумінні, підстановки степеня N можуть бути сконструйовані з підстановок меншого степеня. Справа в тому, що підстановки степеня $n<N$ можуть діяти на операнді з N елементів, якщо вважати, що кожний з $N-n$ додаткових елементів не переміщується (переходять у себе). Наприклад, під час доповнення кожного операнда елементами $n+1$, $n+2$ ... N можна вважати, що

$$\begin{pmatrix}1&2&\dots&n\\ \alpha_1&\alpha_2&\dots&\alpha_n\end{pmatrix}=\begin{pmatrix}1&2&\dots&n&n+1&\dots&N\\ \alpha_1&\alpha_2&\dots&\alpha_n&n+1&\dots&N\end{pmatrix}.$$

Виходячи з цього, можна говорити про те, що підстановки меншого степеня включаються в множину підстановок більшого степеня і, в свою

чергу, утворять тепер уже підгрупу за вихідною операцією множення. Виявляється, кожну підстановку T можна представити у вигляді добутку $T = P_1 ... P_k$ деяких спеціальних підстановок, які називаються циклами, причому, цикли $P_1 ... P_k$ попарно незалежні. Останнє означає, що підстановки P_i і P_j, при $P_i \neq P_j$, діють на неперетинних підмножинах операнда підстановки T, якщо не брати до уваги елементи, які залишаються нерухомими. Пояснимо зазначену властивість підстановок більш конкретно.

Нехай $1 < k < n$ і P – підстановка степеня N, причому $P \neq I$. Підстановка P називається *k-членним* циклом, якщо вона не переміщує $N - k$ елементів, а її дію на k елементів, що залишилися, $i_1, i_2, ..., i_k$ можна представити у вигляді циклічної діаграми переходів:

$$i_1 \to i_2 \to i_3 \to ... \to i_k \to i_1.$$

У цій діаграмі допускається тільки один перехід від елемента з більшим індексом до елемента з меншим індексом, а саме: $i_k \to i_1$. Наприклад, тричленний цикл п'ятого степеня може виглядати так: $P = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 1 & 2 & 4 & 5 \end{pmatrix}$.

Тут елементи 4 і 5 нерухомі, причому $i_1 = 1$, $i_2 = 3$, $i_3 = 2$. Ясно, що циклічна діаграма переходів може бути виписана, починаючи з будь-якого свого елемента, однак усі ці записи відповідають одній і тій же підстановці. Зазвичай цикл записують у вигляді, аналогічному діаграмі переходів: $(i_1, i_2, ..., i_k)$, а одиничну підстановку розглядають як добуток одночленних циклів вигляду $I = (i)$, де під i мається на увазі будь-який елемент (від 1 до N). Для запису підстановки у вигляді добутку циклів достатньо виписати всі різні діаграми переходів. Наприклад, підстановка може бути представлена як $T = (1679)(2354)(8)$, втім, одночленні цикли під час записування часто опускаються. Найбільш простим циклом, мабуть, є підстановка, яка переставляє місцями тільки два елементи. Такий двочленний цикл називається транспозицією. Виявляється, можна показати, що якщо підстановка степеня N розкладається в добуток r попарно незалежних циклів (включаючи й одночленні цикли), то її можна представити у вигляді добутку $n - r$ транспозицій. Відзначимо, що вказані транспозиції не обов'язково є незалежними циклами.

Повертаючись до шифрів перестановки, припустимо, що номери букв відкритого тексту довжиною в N знаків розбиті яким-небудь способом на множини $M_1,...,M_r$, які не пересікаються. Впливаючи на кожну множину M_i своєю підстановкою (циклом) T_i, ми реалізуємо процес зашифрування шифром перестановки, причому відповідна шифрперетворенню підстановка дорівнює $T_1,...T_r$. Змінюючи розбивку множини індексів і вибираючи різними способами відповідні підстановки T_j', ми будемо отримувати різні перетворення відкритого тексту вигляду $T_1',...,T_s'$. Причому, рано чи пізно, ми побудуємо всі підстановки степеня N, оскільки будь-яка підстановка розкладається в добуток циклів. Те ж саме можна сказати і стосовно деякого іншого, більш складного процесу генерації підстановок, який, скажімо, включає вищевказану процедуру як один з етапів. Звідси випливає, що вдалий спосіб генерації підстановок міг би, в принципі, забезпечити досить якісний шифр перестановки, при якому використовувалися би підстановки степеня меншого, ніж довжина відкритого тексту. Останнє досить важливо, оскільки зберігати і передавати ключі, довжина яких дорівнює довжині відкритого тексту, в даному випадку невигідно (існують більш досконалі шифри з аналогічними вимогами) і, крім того, неможливо зашифрувати шифром перестановки текст, довжина якого менше степеня відповідної підстановки. Саме способами генерації підстановок і різняться різні види шифрів перестановки.

Для ручного використання таких шифрів дуже важливим є зручність зашифрування і легкість запам'ятовування ключа. Розглянемо приклад перестановної криптограми, для якої підстановка задається неявно, як ключове слово – «гасло».

Відкритий текст криптограми такий: *"В связи с создавшимся положением отодвигаем сроки возвращения домой Рамзай"*.

Для зашифрування шифром вертикальної перестановки побудуємо прямокутну таблицю, кількість рядків якої визначається довжиною тексту, а кількість колонок дорівнює шести. У якості «гасла» виберемо слово "ЗАПИСЬ" (кількість букв у ключовому слові повинна дорівнювати кількості стовпців у нашій таблиці).

Замінимо тепер кожну букву ключового слова на число від 1 до 6 таким чином, щоб буква, яка має менший порядковий номер в алфавіті,

замінялася на менше число. Отримані числа $(2,1,4,3,5,6)$ проставимо підряд на початку відповідних стовпців таблиці і будемо надалі вважати їх номерами цих стовпців. Впишемо відкритий текст у таблицю, переходячи звичайним чином з рядка на рядок. У результаті отримаємо:

2	1	4	3	5	6
в	с	в	я	з	и
с	с	о	з	д	а
в	ш	и	м	с	я
п	о	л	о	ж	е
н	и	е	м	о	т
о	д	в	и	г	а
е	м	с	р	о	к
и	в	о	з	в	р
а	щ	е	н	и	я
д	о	м	о	й	р
а	м	з	а	й	

Випишемо тепер букви зі стовпців таблиці: спочатку весь стовпець, на початку якого стоїть одиниця, потім – стовпець, позначений двійкою і т.д.

У підсумку, отримаємо такий шифртекст (представивши його п'ятизначними групами знаків):

ссшои дмвщо мвсвп ноеиа даязм омирз ноаво илевс оемзз дсжог овийй иаяет акряр

Ясно, що шифртекст відрізняється від відкритого тексту лише перестановкою букв і ми, таким чином, сумістили процес генерації підстановок із процесом зашифрування.

У криптографічній практиці результат дії підстановки на послідовність номерів знаків відкритого тексту, який є рядом чисел, що вказують місце чергових букв шифртексту у відкритому тексті, називається шкалою рознесення. Неважко бачити, що для розшифрування, тобто перестановки знаків шифртексту на вихідне місце, необхідно скористатися

підстановкою, оберненою шкалі рознесення, яка називається шкалою набору.

У залежності від окремого вигляду закону побудови шкали рознесення, розрізняють окремі види шифру перестановки – горизонтальну, вертикальну перестановки. У деяких випадках шкала рознесення будується як функція від двох незалежних перестановок невеликого степеня (подвійна перестановка) або від перестановки і геометричної фігури (так звані шифри-грати і лабіринти).

На рис.2.1 наведено приклад зашифровування подвійною перестановкою.

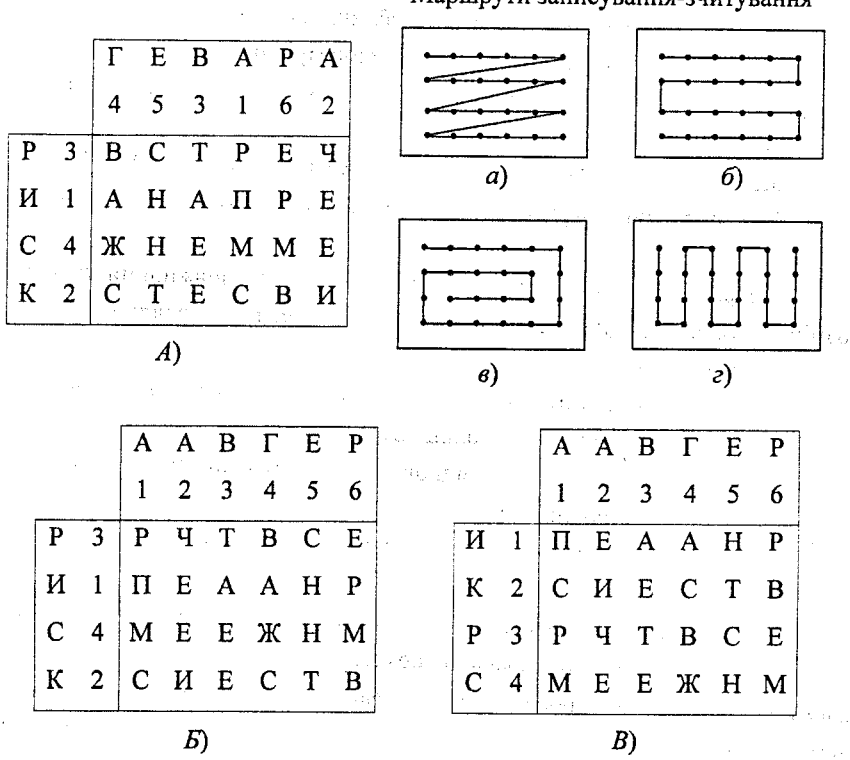


Рис. 2.1. Приклад зашифровування подвійною перестановкою

При цьому необхідно виконати такі дії:

– побудувати якусь таблицю, формати якої визначаються розмірами двох ключових слів (скажімо, "гевара" і "риск"), які виписуються при цьому зверху і збоку таблиці;

– у таблицю за певним маршрутом (приміром, "*а*") заноситься вихідний текст (таблиця *А*), а невикористані місця повністю заповнюються будь-якими буквами, але краще такими, які зустрічаються часто (тут: "з", "у", "и");

– перемістити стовпці в порядку, що відповідає розташуванню букв у верхнього ключа ("гевара") в звичайному алфавіті (таблиця *Б*);

– перемістити всі рядки у відповідності з послідовністю букв другого ключового слова ("риск") в алфавіті (таблиця *В*);

– виписати послідовно букви з таблиці, яка вийшла, стандартно розбиваючи їх на п'ятизнакові групи, причому, якщо остання з них виявиться неповною, вона дописується будь-якими буквами, що часто зустрічаються.

Наш шифртекст – **ПЕААН РСИЕС ТВРЧТ ВСЕМЕ ЕЖНМИ.**

Під час розшифровування криптограми варто діяти в зворотному порядку:

– шифртекст вписується в таблицю визначеного довжинами ключів розміру; стовпці і рядки в ній послідовно нумеруються, а надлишок букв відкидається (так виходить таблиця *В*);

– рядки розташовують відповідно до порядку номерів букв бічного ключового слова (так виходить таблиця *Б*);

– стовпці переставляються відповідно до нумерації букв верхнього ключа (так виходить таблиця *А*);

– букви виписуються в рядок, слідуючи обумовленим маршрутом заповнення–читання.

Насамкінець відзначимо, що зашифрування випадковою, такою, що не має закономірностей, шкалою рознесення при достатньо великій довжині повідомлення робить дешифрування такої криптограми досить проблематичним.

Порядок виконання роботи

1. Вивчити короткі теоретичні відомості про шифри перестановки.
2. Створити ключ (наприклад, на основі деякого гасла).

3. Створити і зашифрувати шифром вертикальної перестановки деяке повідомлення.

4. Перевірити правильність зашифрування за допомогою програми VERPER.

5. Побудувати шкалу рознесення і по ній шкалу набору.

6. Розшифрувати зашифроване повідомлення.

7. Перевірити правильність розшифрування за допомогою програми VERPER.

8. Скласти звіт, додавши туди отримані результати.

Вимоги до звіту

У звіті повинні бути наведені:

– короткі відомості про вивчений вид шифру;

– вибраний ключ;

– відкрите повідомлення;

– зашифроване повідомлення;

– порядок зашифрування і розшифрування;

– висновки про властивості і якість вивченого шифрперетворення;

– відповіді на контрольні запитання.

Контрольні питання

1. У чому полягає основна суть шифрів перестановки?

2. Які відмінності між шифрами вертикальної, горизонтальної і подвійної перестановок? Порівняйте їх за складністю і криптографічною якістю.

3. Які відмітні риси шифрів перестановки? Порівняйте їх із шифрами заміни, кодами.

4. Поясніть, наскільки криптографічно стійкими є шифри перестановки?

5. Якими могли б бути способи спрощення побудови ключів шифрів перестановки? Чи знижують ці особливості побудови ключів загальну стійкість застосованого шифру?

Лабораторна робота № 3
ГАМУВАННЯ

Мета роботи – вивчити теоретичні основи шифру гамування, на практиці здійснити створення послідовності буквеної гами, провести зашифрування відкритого і розшифрування шифрованого повідомлення. Усвідомити сильні і слабкі сторони шифрів гамування.

Короткі теоретичні відомості

Як уже відзначалося вище, перетворення відкритого тексту часто виконується за допомогою обчислень, здійснюваних над буквами алфавіту, яким попередньо присвоєні деякі числові значення. Наприклад, букви алфавіту нумеруються з нуля, а їхні числові значення збігаються з цими номерами. Для латинського алфавіту букві A можна приписати значення 0, букві B – значення 1, букві C – значення 2 і так далі до букви Z, якій приписується значення, рівне 25. Для того, щоб скласти букви B і D складемо їхні числові значення: $1+3=4$. Розглянемо суму як числове значення деякої букви латинського алфавіту. Легко бачити, що такою буквою є буква E. Вважаємо тому: $B+D=E$. При додаванні букви Z з буквою C числове значення дорівнює 27 і, мабуть, не відповідає ніякій букві алфавіту. В таких випадках вважають, що в алфавіті за буквою Z йде буква A, потім B і т.д. У другому алфавіті букві A приписане числове значення рівне 26, букві B – 27 і так до букви Z. Потім йде третій алфавіт, четвертий алфавіт і так далі необхідну кількість разів. Таким чином, можна додавати кілька букв в одному виразі, виконувати множення букв або множити букви на константи. В даному випадку: $Z+C=B$. Зазначені дії над числовими значеннями букв відповідають операціям, виконуваним над числами за модулем m, де модуль дорівнює кількості знаків в алфавіті. Часто величину m називають модулем алфавіту. Під час виконання модульних операцій однаковим буквам, які знаходяться в різних, послідовно записаних, алфавітах повинні відповідати однакові числові значення. Наприклад, значення 1, 27, 53 задають ту саму букву B і вони, у цьому розумінні, еквівалентні. Неважко бачити, що ці числа відрізняються на величину, кратну m, тобто мають один і той же залишок під час ділення на модуль алфавіту. Такі числа називаються порівнянними

за модулем m, що записується у вигляді так званих порівнянь: $a \equiv b(\mathrm{mod}\, m)$, тобто $1 \equiv 27(\mathrm{mod}\, 26)$, або $1 \equiv 27(26)$. При переході до порівнянь, числові значення і модуль алфавіту маються на увазі, а самі порівняння часто записуються як рівності: $Z + C \equiv B$, замість $Z + C \equiv B(\mathrm{mod}\, 26)$.

Таким чином, тепер ми готові розглянути такий розповсюджений спосіб зашифрування відкритого тексту, як гамування. Шифрувальною гамою будемо називати послідовність символів з деякої скінченної множини. Такою послідовністю можуть виступати букви, цифри, біти і т.д. Процес зашифрування відкритого тексту за допомогою гами називають накладанням гами, або гамуванням. Перед зашифруванням формується дворядковий запис, де в одному рядку послідовно виписані знаки відкритого тексту, а в іншому – відповідні знаки гами. Кожному знакові відкритого тексту відповідає свій знак гами, тобто вони утворюють вертикальні біграми знаків. Розрізняють два види гамування – модульне і табличне. Модульне накладання було описано вище. Під час табличного гамування вертикальні пари, складені з відповідних знаків відкритого тексту і гами, заміняються на знаки шифртексту за деякою таблицею. Для реалізації взаємооднозначного перетворення така таблиця повинна обов'язково бути так званим «латинським квадратом», тобто будь-який її рядок і будь-який стовпець повинні являти собою перестановку знаків заданого алфавіту (або чисел від 1 до m), і в кожнім стовпці і рядку даної таблиці всі елементи повинні бути різні. Наприклад, для $m = 5$:

	0	1	2	3	4
0	0	1	2	3	4
1	2	3	4	0	1
2	1	2	3	4	0
3	4	0	1	2	3
4	3	4	0	1	2

Для отримання шифрованого тексту (S) існує три способи накладання гами (Γ) на відкритий текст (O): додавання гами і тексту ($S = \Gamma + O$), віднімання гами з тексту ($S = O - \Gamma$) і віднімання тексту з гами ($S = \Gamma - O$). Під операціями додавання і віднімання ми тут розуміємо як звичайні операції за модулем m, так і застосування замість них

відповідних таблиць. Процедура розшифрування, очевидно, будується природним чином, використовуючи обернені перетворення $O = S - \Gamma$, $O = S + \Gamma$, $O = \Gamma - S$ або обернені таблиці, відповідно.

За довжиною гаму можна класифікувати на коротку і довгу. Коротка гама виникає у випадку, коли її довжина менша довжини відкритого тексту, який підлягає зашифруванню. У цьому випадку, для отримання шифрувальної послідовності необхідної довжини, гама дописується в кінець до самої себе повторно необхідну кількість разів. Довгою гама буває у випадку, якщо її довжина завжди більша будь-якого зашифровуваного повідомлення.

З розвитком засобів телекомунікацій і обчислювальної техніки, відкриті повідомлення стали представлятися у вигляді послідовностей бітів, заміняючи знаки вихідного алфавіту повідомлення на їхні бітові комбінації у відповідному кодуванні. Користувачам комп'ютерів добре відомі кодування ASCII, EBCDIC та інші. У минулому практично всі засоби зв'язку мали можливість використовувати так званий міжнародний телеграфний код (МТК-2). Цей код містить у собі всі тридцять дві п'ятибітові комбінації. Його особливість полягає в тому, що конкретна кодова комбінація може відповідати то одному, то іншому символу тексту, в залежності від наявності перед нею спеціальних кодових слів, які називаються регістровими комбінаціями. Усього в коді передбачені три регістрових комбінації (позначення наші): <РУС> – російський регістр (регістр кирилиці), <ЛАТ> – латинський регістр, <ЦИФ> – цифровий регістр.

Ці комбінації мають відповідно такий вигляд: 00000, 11111, 11011.

В принципі, кожен знак тексту можна представляти парою кодових слів, з яких перше слово є регістровою комбінацією:

"Л" = 00000 01001, "(" = 11011 01001, "L" = 11111 01001.

З метою скорочення довжини повідомлення у коді МТК-2 можна використовувати регістри економніше, задаючи регістрову комбінацію лише при необхідності скасування дії попередньої регістрової комбінації. Іншими словами, регістрова комбінація впливає на всі наступні за нею кодові слова, крім іншої регістрової комбінації.

Таким чином, послідовності "11111 01001 01001" і "00000 11111 11011 11111 01001 01001" при друкуванні на телетайпі дадуть той самий

результат: *LL*. Крім регістрів у кількість службових кодових слів входять ще три комбінації: пропуск, повернення каретки і переведення рядка. Повернення каретки приводить до друкування чергового символу з першої позиції поточного рядка (не виключено, що поверх уже раніше надрукованих символів). Ми позначимо повернення каретки знаком <. Переведення рядка (позначимо ≡) приводить до друкування чергового знаку слідом за попереднім, але на один рядок нижче. Комбінація "пропуск" (позначимо його незайнятою позицією) приводить до пропуску однієї позиції рядка.

Позначення кодових комбінацій зручні для скороченого запису бітової послідовності. Так, наприклад, тексту в коді МТК-2 "11111 01001 00100 01001 11011 01001" відповідає запис "<РУС>L L<ЦИФ>)".

Нижче приводиться таблиця коду МТК-2, яка складається з двох підтаблиць.

	лат	рус	циф	1	2	3	4	5
A	A	Ј	–	1	1	0	0	0
B	B	Б	?	1	0	0	1	1
C	C	Ц	:	0	1	1	0	0
D	D	Д		1	0	0	1	0
E	E	Е	3	1	0	0	0	0
F	F	Ф	Э	1	0	1	1	0
G	G	Г	Ш	0	1	0	1	1
H	H	Х	Щ	0	0	1	0	1
I	I	И	8	0	1	1	0	0
J	J	Й	Ю	1	1	0	1	0
K	K	К	(	1	1	1	1	0
L	L	Л	)	0	1	0	0	1
M	M	М	.	0	0	1	1	1
N	N	Н	,	0	0	1	1	0
O	O	О	9	0	0	0	1	1
P	P	П	0	0	1	1	0	1

	лат	рус	циф	1	2	3	4	5
Q	Q	Я	1	1	1	1	0	1
R	R	Р	4	0	1	0	1	0
S	S	С	'	1	0	1	0	0
T	T	Т	5	0	0	0	0	1
U	U	У	7	1	1	1	0	0
V	V	Ж	=	0	1	1	1	1
W	W	В	2	1	1	0	0	1
X	X	Ь	/	1	0	1	1	1
Y	Y	Ы	6	1	0	1	0	1
Z	Z	З	+	1	1	1	1	1
<РУС>				0	0	0	0	0
<				0	0	0	1	0
пробел				0	0	1	0	0
≡				0	1	0	0	0
<ЦИФ>				1	1	0	1	1
<ЛАТ>				1	1	1	1	1

У першому стовпці підтаблиці записані позначення (назви) кодових комбінацій. В другому, третьому, четвертому стовпцях наводиться графічний еквівалент (зображення) кодових слів на латинському, російському і цифровому регістрах. Службові комбінації (їхні позначення приведені наприкінці першого стовпця другої підтаблиці) графічного еквівалента не мають. Таким чином, знаки <ЛАТ>, <ЦИФ>, <РУС> означають операції, виконувані телеграфним апаратом. У правій частині підтаблиць, по горизонталі, в стовпцях з п'ятого по дев'ятий, розташовані кодові слова, які відповідають знакам першого стовпця. Цифри (12345) поза підтаблицею задають порядок номерів бітів у кодовому слові. Нумерація є умовною. Часто біти в кодовому слові нумеруються навпаки: (54321), оскільки молодший розряд числа розташований праворуч.

Під час записування тексту довжини N за допомогою кодових комбінацій виходить бітова послідовність, яка складається з $5N$ бітів. Той же текст, очевидно, може бути записаний у вигляді послідовності з N колонок по п'ять бітів, якщо кожну кодову комбінацію записувати по вертикалі. У першому випадку будемо говорити, що текст записаний у лінію, а в другому – що текст записаний побланково.

Наприклад, фраза "I AM 21 YEARS OLD" буде виглядати так:

ла	I		A	M		*ци*	W	Q		*ла*	Y	E	A	R	S		O	L	D
т						*ф*				*т*									
1	0	0	1	0	0	1	1	1	0	1	1	1	1	0	1	0	0	0	1
1	1	0	1	0	0	1	1	1	0	1	0	0	1	1	0	0	0	1	0
1	1	1	0	1	1	0	0	1	1	1	1	0	0	0	1	1	0	0	0
1	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	1	0	1
1	0	0	0	1	0	1	1	1	0	1	1	0	0	0	0	0	1	1	0

Утворені вказаним чином рядки називаються бланками. Неважко уявити собі шифрсистему, яка під час зашифрування перетворює кожен бланк окремо, тобто шифрує не стовпці нашого тексту, а рядки.

Під час передавання повідомлень, у реальній системі секретного зв'язку, до відкритого або до шифрованого тексту можуть застосовуватися додаткові перетворення. Наприклад, може статися, що повідомлення, яке передається каналом зв'язку, повинно за технічними умовами складатися тільки з букв латинського алфавіту, а шифртекст, проте, являє собою

бітову послідовність. Тоді можна вчинити так. Нехай п'ятибітові комбінації яким-небудь чином упорядковані. Позначимо перші 25 комбінацій буквами латинського алфавіту від A до Y. 7 комбінацій, що залишилися, позначимо парами, які починаються з букви Z: $ZA, ZB, ZC, ZD, ZE, ZF, ZG$. Можна зробити й інакше. Наприклад, досить розбити бітовий потік на чотирибітові комбінації, доповнити кожну комбінацію нулем і замінити її на букву латинського алфавіту відповідно до коду МТК-2. У підсумку, бітове представлення шифртексту стане довшим, алфавіт тексту буде складатися тільки з 16 букв, однак поставлена задача буде вирішена.

Додатково відзначимо, що перетворений у такий спосіб текст може бути зашифрований шифрами, розглянутими в лабораторних роботах № 1 і 2.

Тепер коротенько розглянемо можливі способи одержання гамових послідовностей. У загальному випадку це досить складний і відповідальний процес. Гама може вироблятися попередньо. Можлива і безпосередня генерація гами в процесі зашифрування, для чого створюються відповідні генератори гамових послідовностей (механічні, електронні). Такі елементи є складеними елементами апаратів, що їх називають шифраторами. Можливо також і застосування деяких підручних засобів для цих цілей. Наприклад, під час шифрування з застосуванням книги можна діяти в такий спосіб:

- береться екземпляр певної книги і розкривається на ключовій сторінці (подібний ключ може прив'язуватися до дати, що додається до якогось числа, вказуватися десь у шифрованому повідомленні, або передаватися якимсь іншим шляхом);
- проводять розрахунки, з позначенням номера рядка і номера необхідної букви в цьому рядку;
- починаючи звідси, послідовність букв (текст книги) і виступає як гама.

Під час шифрування можна застосовувати мікрокалькулятор, отримуючи з його допомогою послідовність псевдовипадкових чисел, які виступають як послідовність гами. Наприклад, такий класичний прийом: беруть яке-небудь чотиризначне число (скажімо, 5997), підносять його мікрокалькулятором до квадрату (35964009), а з отриманого результату (35964009) виписують середні цифри (9640), розглядаючи їх як випадкові;

тепер уже ці «випадкові» числа підносять до квадрату і знову виділяють з отриманого результату середину і т.д. Побудована таким чином послідовність через десятки або сотні знаків, у залежності від вихідного числа – ключа, звичайно, повторюється, але при коротких текстах цілком може розглядатися як випадкова. Можна використовувати мікрокалькулятор або ПЕОМ із функцією генерування псевдовипадкової числової послідовності: у регістри калькулятора тут вводять комбінацію якихось чисел (вони виступають як ключ шифру), після чого під час натискання потрібної клавіші на індикаторі висвічуються знаки псевдовипадкової числової послідовності – гами.

Одним з найбільш надійних способів шифрування є застосування одноразового блокнота. Такий блокнот складений з відривних сторінок, на кожній з яких надрукована таблиця з випадковими цифрами або буквами. Таблиця має тільки два екземпляри: один у передавача, інший – у приймача. Для кожного символу повідомлення кожен символ з блокнота використовується тільки один раз (тому блокнот і називається одноразовим). Коли таблиця використана, вона відривається і знищується.

Насамкінець відзначимо, що представлення інформації, яка підлягає зашифруванню, у бітовому вигляді вимагає, відповідно, і бітової послідовності гами. Накладання гами, у даному випадку, найчастіше здійснюється з застосуванням операції додавання за модулем 2, яка виконується досить швидко. У 1917 році, інженер фірми AT&T Вернам запропонував версію шифру, який використовував двійкове представлення символів у рамках використовуваного тоді телеграфного коду. У сучасних умовах шифри гамування використовуються для шифрування значних об'ємів інформації і вимагають наявності високошвидкісних генераторів послідовностей гами.

Порядок виконання роботи.

1. Вивчити короткі теоретичні відомості про шифрування гамуванням.
2. Створити послідовність гами в алфавіті 26 латинських букв.
3. Створити відкрите повідомлення, записавши його латинськими буквами. В якості пропусків, ком, крапок використовувати скорочення BLN, CMA, PNT. Цифри, за необхідністю, записувати словами.

4. Зашифрувати гамуванням сформоване повідомлення, використовуючи формулу $S = \Gamma - O(\mathrm{mod}\,26)$.

5. Перевірити правильність зашифрування за допомогою програми GAMMA.

6. Розшифрувати зашифроване повідомлення.

7. Перевірити правильність розшифрування за допомогою програми GAMMA.

8. Скласти звіт, додавши туди отримані результати.

Вимоги до звіту

У звіті повинні бути наведені:

- короткі відомості про вивчений вид шифру;
- вибраний ключ – послідовність гами та описаний спосіб її отримання;
- відкрите повідомлення;
- зашифроване повідомлення;
- порядок зашифрування і розшифрування;
- висновки про властивості і якість вивченого шифрперетворення;
- відповіді на контрольні запитання.

Контрольні питання

1. У чому полягає основна суть шифрування гамуванням?

2. Наведіть класифікацію видів гамувальних послідовностей за різними ознаками. Які, у криптографічному розумінні, розбіжності між ними?

3. Які відмітні риси шифрування гамуванням? Порівняйте його із шифрами заміни, кодами, шифрами перестановки.

4. Поясніть, наскільки криптографічно стійким є шифрування гамуванням і від чого залежить його стійкість?

5. Якими могли б бути способи спрощення побудови послідовностей гами? Порівняйте їх за складністю і криптографічною якістю. Чи знижують їхні особливості загальну стійкість застосованого способу шифрування?

Лабораторна робота № 4
СТАНДАРТ ШИФРУВАННЯ ДАНИХ DES

Мета роботи – вивчити структуру алгоритму шифрування DES, на практиці здійснити формування ключа, зашифрування і розшифрування фрагмента інформації даним алгоритмом. Усвідомити сильні і слабкі сторони в застосуванні даного виду шифру.

Короткі теоретичні відомості

Федеральний стандарт шифрування даних (DES) прийнятий у США в листопаді 1976 року. В стандарт входить опис «блокового шифру типу Файстеля» а також різних режимів його роботи як складової частини декількох процедур криптографічного перетворення даних. Зазвичай під абревіатурою DES розуміється саме «блоковий шифр» (див. нижче), який у стандарті відповідає процедурі шифрування в режимі ECB (Electronic Codebook Mode). Назва викликана тим, що будь-який «блоковий шифр» є шифром простої заміни й у цьому відношенні подібний кодовій таблиці.

Пояснимо суть блокового шифру. Входом у блоковий шифр і результатом його роботи є блок довжини n – послідовність, яка складається з n бітів. Число n постійне. При необхідності зашифрування повідомлення довжини більшої n воно розбивається на блоки, кожен з яких шифрується окремо. Різні режими роботи пов'язані з додатковими ускладненнями блокового шифру під час переходів від блоку до блоку. В стандарті DES довжина блоку $n = 64$.

У режимі ECB зашифрування блоку відкритого тексту B проходить за 16 однотипних ітерацій, іменованих циклами. Схема перетворення приведена на рис.3.1. Блок розглядається як конкатенація (зчеплення) двох підблоків: $B = (L, R)$. Довжини підблоків L і R рівні. В кожному циклі застосовується свій ключ (X_i), зазвичай він виробляється з деякого основного ключа (X). Ключі, які використовуються в циклах, називаються підключами.

Основним елементом шифру є несекретна циклова функція вигляду $Y = f(R, X)$. Входом у цикл є вихід з попереднього циклу. Якщо згаданий вхід має вигляд (L, R), то вихід має вигляд $(R, L \oplus f(R, X))$, де $\oplus$ –

порозрядне додавання за модулем 2. Більш точно, для виходу циклу з номером i, це означає

$$R_i = L_{i-1} \oplus f(R_{i-1}, X_i), L_i = R_{i-1} (i = 1,...,16).$$

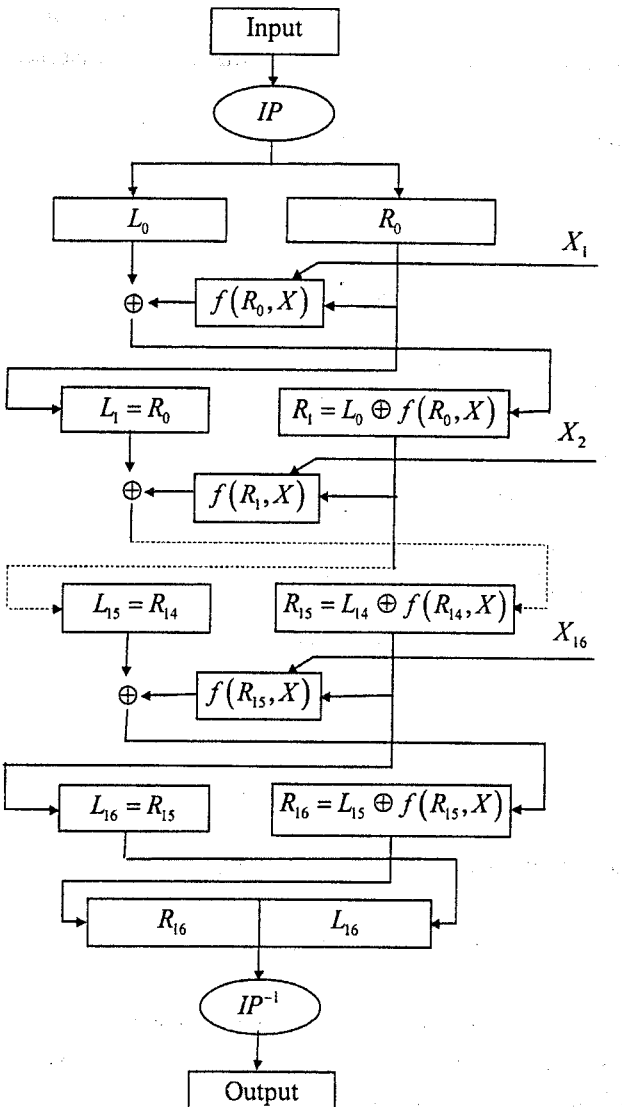


Рис 3.1. Блок-схема роботи алгоритму DES

У режимі ECB алгоритм DES зашифровує 64-бітовий блок за 16 циклів. Біти вхідного блоку перед першим циклом переставляються за підстановкою IP (таб.3.1). Після виходу з останнього циклу L і R переставляються місцями, після чого з'єднуються в блок. Біти отриманого блоку переставляються за IP^{-1}. Результат приймається як блок шифртексту.

На рис 3.2 наведено процедуру формування підключів.

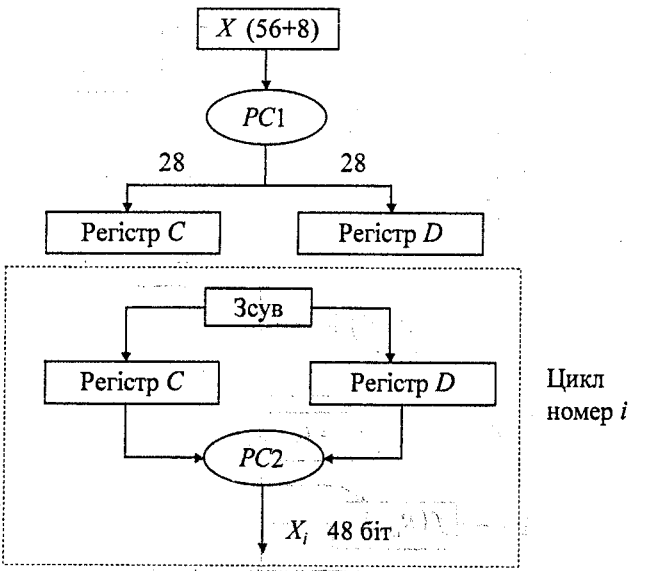


Рис. 3.2. Формування підключів

В кожному циклі ключ X_i має розмір у 48 бітів і є підвибіркою з ключа X довжиною в 56 бітів. Сам ключ X розміщується у вигляді восьмибайтового слова. Восьмі розряди байтів є контрольними й у ключ не входять. Перед зашифруванням, відповідно до процедури вибору $PC1$ (таб. 3.2), з X вибираються 56 бітів, якими заповнюються два регістри (C,D) довжиною в 28 бітів кожний. Надалі, під час входження в черговий цикл з номером i, регістри зсуваються циклічно вліво. Величина зсуву залежить від номера циклу, але є фіксованою і відома. Після зсуву обидва підблоки поєднуються в порядку C, D. Потім, відповідно до функції

вибору $PC2$ (таб.3.3), з них вибираються 48 бітів подключа X_i. Зашифрування і розшифрування відрізняються послідовністю зсувів (таб.3.4).

Таблиця 3.1. Перетворення IP

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

Таблиця 3.2. Перетворення $PC1$

Заповнення C							Заповнення D						
57	49	41	33	25	17	9	63	55	47	39	31	23	15
1	58	50	42	34	26	18	7	62	54	46	38	30	22
10	2	59	51	43	35	27	14	6	61	53	45	37	29
19	11	3	60	52	44	36	21	13	5	28	20	12	4

Таблиця 3.3. Перетворення $PC2$

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	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

Таблиця 3.4. Відповідність зсувів номерам циклів DES

Номер циклу	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Зсув вліво (зашифр.)	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1
Зсув вправо (розшифр.)	0	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Вибір бітів за таблицями 3.1–3.3 з відповідних блоків здійснюється в такий спосіб. Таблиця розглядається як послідовність її рядків, записаних один за одним, починаючи з першого рядка. Біти блоку даних відповідної довжини нумеруються зліва направо, починаючи з одиниці. Кожен елемент s таблиці розглядається як номер біта b_s в блоці даних. Перетворення полягає в заміні всіх елементів s на біти b_s.

Циклова функція робить такі дії:

а) розширення блоку R_{i-1} до 48 бітів за рахунок повторення його бітів за допомогою функції розширення EP;

б) порозрядне додавання результату з ключем X_i;

в) перетворення отриманої суми за допомогою заміни (використовуючи так звані S-блоки), у результаті якого виходить блок довжини 32 біта;

г) перестановку результату заміни за постійною і відомою підстановкою P, що дає значення функції $Y = f(R, X)$.

Механізм дії *S-блоків*. Перетворення, за допомогою якого 48-розрядний блок перетвориться в 32-розрядний, полягає у вибірці восьми тетрад з 8 таблиць (S-блоків) розміром 4 на 16. З кожного S-блоку вибирається одна тетрада. Для цього 48-розрядний блок ділиться послідовно на 8 комбінацій по 6 бітів кожна. Перша комбінація (зліва) є входом у перший S-блок, друга – у другий і т.д. При цьому перший і останній біти комбінації задають номер рядка, а інші 4 біти – номер стовпчика S-блоку, на перетині яких розташована відповідна тетрада. Конкретні значення $S_i\,(i = 1,...,8)$ наведені в таблиці 3.7. Перетворення EP, P задані таблицями 3.5, 3.6 і використовуються аналогічно таблицям 3.1–3.3.

Таблиця 3.5. Перетворення EP

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

Таблиця 3.6. Перетворення P

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

Таблиця 3.7. S-блоки для DES

S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
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

S_2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
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

S_3	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
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

S_4	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
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

Таблиця 3.7. (продовження)

S_5	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
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

S_6	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
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

S_7	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
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

S_8	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
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

Розглянемо інші режими використання алгоритму шифрування DES.

Режим зчеплення блоків шифртексту CBC (Cipher Block Chaining Mode). Суть цього режиму полягає в тому, що повідомлення розбивається на блоки по 64 біти, і їхня послідовність зашифровується. Перед зашифруванням (у режимі ECB) блок відкритого тексту порозрядно складається з попереднім блоком шифртексту. Для зашифрування першого блоку шифртексту потрібен унікальний для повідомлення блок (вектор ініціалізації, IV). Останній не є секретним. Даний режим не дозволяє накопичуватися помилкам під час передавання, оскільки помилка під час

передавання призведе до втрати тільки двох блоків вихідного тексту. Існують також режим шифрування зі зворотним зв'язком CFB (Cipher Feedback) і режим шифрування з зовнішнім зворотним зв'язком OFB (Output Feedback). Розгляд цих режимів виходить за рамки даної роботи.

Порядок виконання роботи

1. Вивчити короткі теоретичні відомості про стандарт шифрування DES.
2. Вибрати ключ шифрування (64 біти, кожен 8-й не використовується) і записати його в шістнадцятковому вигляді.
3. Вибрати вхідний блок інформації – 64 біта і записати його в шістнадцятковому вигляді.
4. Вручну зашифрувати блок інформації.
5. Використовуючи програму DES провести зашифрування цього ж блоку.
6. Порівняти результати.
7. Розшифрувати результат. Переконатися в правильності зашифрування отриманням вихідного блоку.
8. Скласти звіт, додавши туди отримані результати.

Вимоги до звіту

У звіті повинні бути наведені:

- криптографічна схема алгоритму DES для режиму ECB і короткі відомості про вивчений вид шифру;
- вибраний ключ;
- відкритий блок повідомлення;
- зашифрований блок повідомлення;
- проміжні результати процесу зашифрування;
- висновки про властивості і якість вивченого шифрперетворення;
- відповіді на контрольні питання.

Контрольні питання

1. У чому полягає основна суть і необхідність появи «блокових» шифрів?

2. Що є основним елементом шифру DES?

3. Поясніть суть циклової функції $Y = f(R, X)$.

4. Які головні вимоги до S-блоків Ви б сформулювали, якби постала задача про їх заміну?

5. Порівняйте вивчений шифр із шифрами заміни, кодами, шифрами перестановки, гамуванням.

6. Наскільки криптографічно стійким є шифрування даним блоковим шифром? Що впливає і може впливати на його стійкість? Як?

7. Наскільки, на Ваш погляд, простими є програмна і апаратна реалізації даного шифру? Чи враховані можливості і тенденції розвитку мікроелектроніки для апаратних реалізацій DES на момент його створення і на перспективу?

8. Які режими використання алгоритму шифрування DES.

Лабораторна робота № 5
СТАНДАРТ КРИПТОГРАФІЧНОГО ПЕРЕТВОРЕННЯ ДАНИХ ГОСТ 28147-89

Мета роботи – вивчити загальну структуру алгоритму криптографічного перетворення даних ГОСТ 28147-89, на практиці здійснити формування ключа, зашифрування і розшифрування фрагменту інформації даним алгоритмом. Усвідомити сильні і слабкі сторони в застосуванні даного виду шифру.

Короткі теоретичні відомості

Стандарт криптографічного перетворення даних ГОСТ 28147-89 рекомендований до використання для захисту будь-яких даних, представлених у вигляді двійкового коду. Даний стандарт формувався з урахуванням світового досвіду і, зокрема, під час його розробки були взяті до уваги недоліки алгоритму DES. Стандарт досить складний, тому приведемо лише його концептуальний опис.

Алгоритм криптографічного перетворення, встановлений ГОСТ 28147-89 (далі – ГОСТ), використовується для зашифрування даних у двох режимах і для вироблення імітовставки, яка є засобом контролю цілісності даних і залежить від ключів. Під час зашифрування алгоритм ГОСТ зводиться до шифру гамування. Блок гами являє собою 64-бітову комбінацію, яка складається з двох послідовних 32-бітових блоків. Виходячи зі зручності викладення, далі будемо називати будь-який 64-бітовий блок комбінацією а також вважати, що блок складається з двох зчеплених підблоків з 32-х бітів кожний.

Гама накладається порозрядно за модулем 2. Кожна комбінація гами являє собою результат шифрперетворення за допомогою шифру простої заміни на множині 64-бітових комбінацій. Вхідні комбінації для вказаного шифру, у загальному випадку, формуються в залежності від ключів, псевдовипадкового відкритого параметра S (синхропосилка), відомих констант $c1$, $c2$ і попереднього блоку шифртексту. Фактично, задача кожного з режимів шифрування – це формування 64-бітових комбінацій для входу в основний режим роботи ГОСТ, який називається режимом простої заміни. По суті, ключі необхідні для роботи ГОСТ саме в цьому

режимі. Комбінація гами є результатом роботи алгоритму в режимі простої заміни.

Алгоритм ГОСТ як вихідні дані використовує три параметри: K, X і Z – 64-бітовий блок даних. Перший параметр є довгостроковим, а другий – сеансовим ключем.

Параметри незалежні і мають розмір 512, 256 і 64 біти відповідно. K являє собою відображення множини блоків у себе. Це відображення реалізує потетрадну заміну 32-розрядних блоків у 32-розрядні і складається з 8 підключів K. Підключ $K_i, (i=1,...,8)$, який входить у K, є таблицею заміни для i-ої (зліва) тетради, тобто складається з 16 тетрад. У стандарті ключ K називається блоком підстановки, а підключі K вузлами заміни.

Сеансовый ключ X складається з восьми 32-розрядних підключів X_i, кожен з яких у відповідний момент використовується для підсумовування з деяким блоком за модулем 2. Режим простої заміни алгоритму ГОСТ реалізований у вигляді шифру Файстеля.

Зашифрування блоку відкритого тексту Z за алгоритмом ГОСТ проходить за 32 цикли. В кожному циклі відбувається перетворення вхідної комбінації у вихідну. Шифртекстом є результат роботи (вихід) тридцять другого циклу, підданий дуже простому додатковому перетворенню.

Процес зашифрування в режимі простої заміни (рис.4.1), який позначимо через $T = ГОСТ(S)$, можна представити у вигляді послідовності 34 блоків $u = (U_{-2}, U_{-1}, U_0, U_1, U_2, ..., U_{30}, U_{31})$, де $U_{-2} \| U_{-1} = S$ і $U_{31} \| U_{30} = T$.

Тут $U_{-1} \| U_0$ – результат роботи циклу номер 0, $U_0 \| U_1$ – результат роботи циклу номер 1 і так далі, $U_{30} \| U_{31}$ – результат роботи циклу номер 31. Додаткове перетворення змінює порядок проходження блоків: $U_{31} \| U_{30} = T$.

В циклі з номером i використовується підключ $X_{t(i)}$. Послідовність вибору підключів, від початкового і до останнього циклу, під час зашифрування така:

$$t(i) = \{0,1,2,3,4,5,6,7;\ 0,1,2,3,4,5,6,7;\ 0,1,2,3,4,5,6,7;\ 7,6,5,4,3,2,1,0\}.$$

Під час розшифрування використовується зворотний порядок проходження підключів.

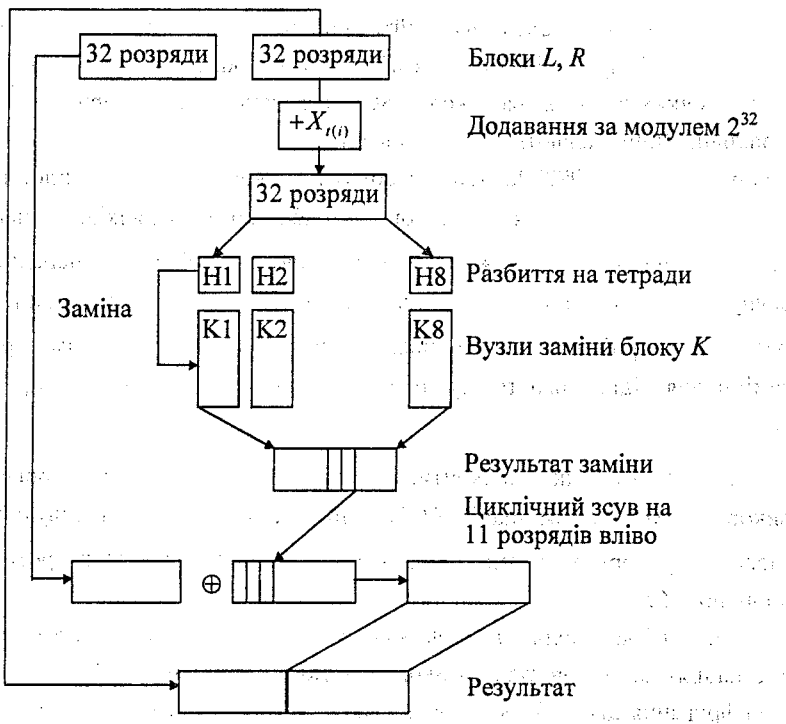


Рис. 4.1. Цикл шифрування в режимі простої заміни

У режимі гамування послідовність 64-бітових комбінацій гами має вигляд: $\gamma_k = ГОСТ(\varphi(\sigma_{k-1}))$, $k = 1,2...$, де $\sigma_0 = ГОСТ(S)$. При цьому, для $s_1 \| s_2$, $\varphi(\sigma)$ складається з двох блоків: $s_1 \circ c1$, $s_2 + c2$. Тут додавання з $c2$ проходить за $\bmod 2^{32}$, а $s_1 \circ c1 = s_1 + c1 \bmod(2^{32} - 1)$, за винятком випадку $s_1 + c1 = 2^{32} - 1$, коли результат приймається рівним $2^{32} - 1$. Шістнадцяткове представлення $c1$, $c2$, відповідно, таке: $x01010101$, $x01010104$.

У режимі гамування зі зворотним зв'язком $\gamma_1 = ГОСТ(S)$, $\gamma_{k+1} = ГОСТ(\gamma_k \oplus t_k)$, $k = 1,2,...$, t – комбінація відкритого тексту.

Порядок виконання роботи

1. Вивчіть загальні відомості про Стандарт криптографічного перетворення ГОСТ 28147-89 і дайте відповідь на контрольні питання.

2. Побудуйте 64-бітову комбінацію Z, яка складається з 8 байтів, кожен з яких представляє в коді ASCII десяткову цифру. Перші дві цифри довільні. Інші задають число, місяць і рік Вашого дня народження. Наприклад: $Z = 9927021918$. Побудуйте вручну блок підстановки $K(Z) = K_1,...,K_8$, розглядаючи його як таблицю з 16 рядків, де перший рядок дорівнює Z, а кожен наступний рядок дорівнює попередньому, зсунутому циклічно на 4 розряди вліво. Побудуйте комбінацію $T(Z)$, яка є результатом роботи другого циклу алгоритму ГОСТ у режимі простої заміни для відкритого тексту, рівного Z, при $K = K(Z)$ і $X_0 = (11...1)$, $X_1 = (00...0)$.

3. Нехай як відкритий текст для вказаної процедури використовується комбінація $T(Z)$, а підключі X_0, X_1 застосовуються в зворотному порядку. Знайти відповідь на питання: чи завжди результат дорівнює Z?

4. Вибрати вхідний блок інформації – 64 біти і записати його в шістнадцятковому вигляді. Використовуючи програму GOST, провести зашифрування цього блоку. Вручну зашифрувати цей же блок інформації, контролюючи свої дії за проміжними результатами, виданими програмою GOST. Порівняти результати. Розшифрувати результат програмою GOST. Переконатися в правильності зашифрування одержанням вихідного блоку. Скласти звіт, додавши відповіді на контрольні запитання й отримані результати.

Вимоги до звіту

У звіті повинні бути наведені:

- криптосхема алгоритму ГОСТ для режиму простої заміни і короткі відомості про вивчений вид шифру;
- відкритий блок повідомлення;
- зашифрований блок повідомлення;
- проміжні результати процесу зашифрування;

– висновки про властивості і якість вивченого шифрперетворення;

– відповіді на контрольні питання.

Контрольні питання

1. Для режиму простої заміни входом у цикл шифрування з номером $i=1,2...$ є вихід з попереднього циклу. Якщо вхід у цикл має вигляд (L_{i-1},R_{i-1}), то на виході циклу $L_i=R_{i-1}$, $R_i=L_{i-1}\oplus f(R_{i-1},X_i)$, де $\oplus$ – порозрядне додавання за модулем 2. Описати перетворення $f(R,X)$.

2. Чи є сама функція $f(R,X)$ незмінною для будь-яких користувачів?

3. Чи є функція $f(R,X)$ секретною при несекретному X?

4. Після зашифрування в режимі гамування зі зворотним зв'язком отримані три блоки шифртексту: T_1, T_2, T_3. Якою була вхідна комбінація для режиму простої заміни під час формування гами, необхідної для зашифрування третього блоку відкритого тексту?

5. Порівняйте вивчений шифр з американським стандартом шифрування даних DES. У чому подібність і основні розбіжності даних алгоритмів?

6. Нехай результат зашифрування синхропосилки в режимі простої заміни відомий. Чи можна в такому випадку (без ключів) отримати всі комбінації, результатом зашифрування яких, у режимі простої заміни, були б комбінації гами для режиму гамування (без зворотного зв'язку)?

Лабораторна робота № 6
КРИПТОГРАФІЧНА СИСТЕМА RSA

Мета роботи – вивчити криптографічну систему RSA, на практиці здійснити формування ключа, зашифрування і розшифрування фрагмента інформації в даній системі. Усвідомити сильні і слабкі сторони в застосуванні систем шифрування з відкритим ключем.

Короткі теоретичні відомості

Якими б не були складними і надійними класичні криптографічні системи – їх слабким місцем, під час практичної реалізації, є проблема розподілу ключів. Для того, щоб був можливий обмін конфіденційною інформацією між двома абонентами, ключ повинен бути сгенерований одним з них, а потім якимось чином переданий іншому в конфіденційному порядку. У загальному випадку, для передавання ключа каналами зв'язку потрібне використання ще однієї криптосистеми, для якої знову виникає проблема розподілу ключів і т.д.

Для вирішення цієї і ряду інших проблем були запропоновані криптосистеми з відкритим ключем, які називають також асиметричними криптосистемами.

Перед відправленням повідомлення адресатові вихідний текст шифрується відкритим (загальнодоступним) ключем адресата. Алгоритм шифрування побудований таким чином, що розшифрування повідомлення можливе тільки з використанням особистого (секретного) ключа адресата.

Вперше модель системи секретного зв'язку з відкритим ключем була запропонована У.Діффі і М.Хеллманом у 1976 році.

Суть цієї моделі в тому, що ключ відомий повністю тільки приймачеві повідомлення і являє собою трійку чисел $k=(e,d,n)$, де підключ e служить ключем зашифрування, а ключ d – для розшифрування. При цьому, тільки d є секретним (особистим). Стійкість системи забезпечується за рахунок особливих властивостей шифрперетворення, яке являє собою так звану однобічну функцію з лазівкою. Обчислення значення такої функції (від відкритого тексту і параметра e) повинно бути нескладним, у той же час її обернення повинно

бути обчислювально нереалізовним без знання секретної інформації, «лазівки», зв'язаної із секретним ключем d.

У криптосистемі з відкритим ключем повідомлення, призначене абонентові, зашифровується передавачем за допомогою ключа e і розшифровується приймачем за допомогою ключа d. Якщо шифрперетворення дійсно є однобічною функцією, то сам передавач не в змозі розшифрувати сформовану ним криптограму.

Широко відомим прикладом криптосистеми з відкритим ключем є криптосистема RSA, розроблена в 1977 році і отримавша назву на честь її авторів: Рональда Рівеста, Аді Шаміра і Леонарда Ейдельмана. Стійкість цієї системи ґрунтується на складності оберненості степеневої функції в кільці лишків цілих чисел за складеним модулем n (при належному виборі модуля).

Наведемо необхідні відомості з елементарної теорії чисел.

1. Простим числом є натуральне число, яке має в точності два нерівних натуральних дільники.
2. Кожне натуральне число єдиним чином, з точністю до порядку запису співмножників, представляється у вигляді добутку ступенів простих чисел.
3. Найбільшим спільним дільником двох цілих чисел $НСД(a,b)$ (або (a,b)) називається найбільше ціле, яке ділить як a, так і b.
4. Нехай $a>b$ і $d=(a,b)$. Тоді існують цілі x,y, які є розв'язком рівняння $xa+yb=d$. Якщо $d=1$, то a і b називаються взаємно простими.
5. Найбільший спільний дільник двох чисел можна знайти за допомогою алгоритму Евкліда. Для цього a ділиться з залишком на b, тобто $a=q_1b+r_1$. Далі, замість a і b, розглядаємо відповідно b і r_1: $b=q_2r_1+r_2$. На наступному кроці роль b і r_1 відіграють r_1 і r_2: $r_1=q_3r_2+r_3$ і т.д. Процес закінчується на деякому кроці $k+1$, для якого $r_{k+1}=0$. Тоді $НСД(a,b)=r_k$.

Приклад. Знайти $(1547,560)$.

Розв'язання:

$1547=2\cdot560+427$

$560=1\cdot427+133$

$427=3\cdot133+28$

$133 = 4 \cdot 28 + 21$

$28 = 1 \cdot 21 + 7$

$21 = 3 \cdot 7 + 0$. $НСД(1547,560) = 7$.

6. Для розв'язання рівняння $xa + yb = d$ можна використовувати дані, отримані на кожному кроці алгоритму Евкліда, рухаючись знизу вгору, за допомогою вираження залишку через інші елементи, використовувані на відповідному кроці. Наприклад, з $r_2 = q_4 r_3 + r_4$ випливає $r_4 = r_2 - q_4 r_3$. В останній рівності r_3 можна замінити, виходячи зі співвідношення $r_1 = q_3 r_2 + r_3$, тобто $r_4 = r_2 - q_4(q_3 r_2 - r_1)$. Тому $r_4 = (1 - q_4 q_3) r_2 + q_4 r_1$. Таким чином, ми виразили r_4 у вигляді цілочислової комбінації залишків з меншими номерами, які, у свою чергу, можуть бути виражені аналогічно. Просуваючись «знизу вгору» r_4, зрештою, буде виражено через вихідні числа a і b. Якби ми почали не з r_4 а з r_k, то отримали б $r_k = xa + yb = d$.

Приклад. Розв'язати $1547x + 560y = 7$.

Розв'язання:

$7 = 28 - 1 \cdot 21 = 28 - 1 \cdot (133 - 4 \cdot 28) = 5 \cdot 28 - 1 \cdot 133 =$

$= 5 \cdot (427 - 3 \cdot 133) - 1 \cdot 133 = 5 \cdot 427 - 16 \cdot 133 = 5 \cdot 427 - 16 \cdot (560 - 1 \cdot 427) =$

$= 21 \cdot 427 - 16 \cdot 560 = 21 \cdot (1547 - 2 \cdot 560) - 16 \cdot 560 = 21 \cdot 1547 - 58 \cdot 560$.

$x = 21$, $y = -58$.

7. Порівняння. Число a порівнянне з числом b за модулем n, якщо $a - b$ ділиться на n. Запис: $a = b(\mathrm{mod}\, n)$. Найменше невід'ємне число a, таке, що $a = A(\mathrm{mod}\, n)$, називається лишком числа A за модулем n. Якщо $(a,n) = 1$, то існує x, таке, що $xa = 1(\mathrm{mod}\, n)$. Дійсно, $(a,n) = 1 = d = ax + ny$, тому $ax = 1(\mathrm{mod}\, n)$. Таке число x називається оберненим до a за модулем n і записується у вигляді $a^{-1}(\mathrm{mod}\, n)$.

8. Нехай функція $\varphi(n)$, де n – натуральне, дорівнює кількості натуральних чисел, менших n, для яких $(a,n) = 1$. Така функція

називається функцією Ейлера. Для чисел n вигляду $n=\prod_i p_i$ (p_i – просте) $\varphi(n)=\prod_i (p_i-1)$.

9. Теорема Ейлера. Нехай $(a,n)=1$. Тоді $a^{\phi(n)}=1(\mathrm{mod}\, n)$.

Висновок. Якщо $ed=1(\mathrm{mod}\,\varphi(n))$ і $(a,n)=1$, то $(a^e)^d=a(\mathrm{mod}\, n)$.

10. Для більшості лишків за модулем $n=pq$ показник степеня у співвідношенні $a^{\phi(n)}=1(\mathrm{mod}\, n)$ може бути зменшений, але в цьому випадку він залежить від a. Найменший показник $k(a)$, для якого $a^{k(a)}=1(\mathrm{mod}\, n)$, називається порядком числа a за модулем n. Він позначається через $ord_n(a)$.

Для кожного a, $ord_n(a)$ ділить $\varphi(n)$.

Криптосистема RSA. Криптосистема RSA на кожному такті шифрування перетворює двійковий блок відкритого тексту m довжини $size(n)$, який розглядається як ціле число, відповідно до формули: $c=m^e(\mathrm{mod}\, n)$.

При цьому $n=pq$, де p, q – випадкові прості числа великої розрядності, які знищуються після формування модуля і ключів. Відкритий ключ складається з пари чисел e, n. Підключ e вибирається як достатньо велике число з діапазону $2\le e\le\varphi(n)-2$, з умовою: $НСД(e,\varphi(n))=1$, де $\varphi(n)$– найменше спільне кратне чисел $p-1$ і $q-1$. Далі, розв'язуючи в цілих числах x, y рівняння $xe+y\phi(n)=1$, вважається $d=x$, тобто $ed=1(\varphi(n))$. При цьому для усіх m виконується співвідношення $m^{ed}=m(n)$, тому знання d дозволяє розшифровувати криптограми.

Щоб гарантувати надійний захист інформації, до систем з відкритим ключем пред'являються дві природних вимоги:

1. Перетворення вихідного тексту повинно виключати його відновлення на основі відкритого ключа;

2. Визначення закритого ключа на основі відкритого також повинно бути обчислювально нереалізовним, при цьому бажана точна нижня оцінка складності (кількості операцій) розкриття шифру.

Алгоритми шифрування з відкритим ключем отримали широке розповсюдження в сучасних інформаційних системах.

Приклад побудови криптосистеми RSA.

1. Виберемо $p=3$ і $q=11$.

2. Визначимо $n=3\cdot 11=33$.

3. Знайдемо $\varphi(n)=(p-1)(q-1)=20$.

4. Виберемо e, взаємно просте з 20, наприклад, $e=7$.

5. Виберемо число d, яке задовольняє $7d=1(\mathrm{mod}\,20)$.

Легко побачити, що $d=3(\mathrm{mod}\,20)$.

Представимо повідомлення, що шифрується, як послідовність цілих чисел за допомогою відповідності: $A=1$, $B=2$, $C=3$, ... , $Z=26$. Оскільки $size(n)=6$, то наша криптосистема в змозі зашифровувати букви латинського алфавіту, що розглядаються як блоки. Опублікуємо відкритий ключ $(e,n)=(7,33)$ і запропонуємо іншим учасникам системи секретного зв'язку зашифровувати на ньому повідомлення, що направляються на нашу адресу. Нехай таким повідомленням буде CAB, яке набуває вигляду (3,1,2). Передавач повинен зашифрувати кожен блок і відправити зашифроване повідомлення на нашу адресу.

$$RSA(C)=RSA(3)=3^7=2187=9(\mathrm{mod}\,33),$$
$$RSA(A)=RSA(1)=1^7=1(\mathrm{mod}\,33),$$
$$RSA(B)=RSA(2)=2^7=128=29(\mathrm{mod}\,33).$$

Одержавши зашифроване повідомлення (9,1,29), ми зможемо його розшифрувати на основі секретного ключа $(d,n)=(3,33)$, підносячи кожен блок до степеня $d=3$:

$$9^3=729=3(\mathrm{mod}\,33),\quad 1^3=1(\mathrm{mod}\,33),\quad 29^3=24389=2(\mathrm{mod}\,33).$$

Для нашого прикладу легко знайти секретний ключ перебором. На практиці це неможливо, тому що в даний час для практичного використання рекомендуються такі значення $size(n)$:

– 512–768 біт – для приватних осіб;
– 1024 біт – для комерційної інформації;
– 2048 біт – для секретної інформації.

Цифровий (електронний) підпис на основі криптосистеми RSA. Асиметрична криптографія дозволяє принципово вирішити задачу підтвердження істинності електронного документа. Ця можливість базується на тому, що зашифрувати дані, використовуючи секретний ключ d замість відкритого ключа e, може тільки той, кому секретний ключ відомий. При цьому, існує можливість перевірки застосування секретного ключа до даних без його розкриття.

Дійсно, нехай нам необхідно завірити блок m відкритого тексту. Сам відкритий текст не є секретним. Зашифруємо m, використовуючи d замість e: $c = m^d \pmod n$. Відправимо повідомлення подвійної довжини вигляду $m \| c$. Приймач має можливість перевірити наш підпис, оскільки, після піднесення c до степеня e повинно отримуватись значення $s = m$ (при істинному підписі) і значення $s \neq m$ в протилежному випадку. Для нашого прикладу $m = (3,1,2)$, $c = (27,1,8)$, $m \| c = (3,1,2,27,1,8)$.

На практиці подвоєння довжини повідомлення, очевидно, є небажаним. Це є однією з причин, за якою замість $c = m^d \pmod n$ використовуються дані вигляду $c = \left(h(m)\right)^d \pmod n$, де функція h відображає повідомлення довільної довжини в короткі блоки фіксованої довжини, причому так, що крім блоку m підібрати інший блок z, із властивістю $h(m) = h(z)$, практично неможливо. Функція $h(m)$ називається геш-функцією.

Порядок виконання роботи

1. Вивчіть опис криптосистеми RSA і відомості з елементарної теорії чисел.

2. Скористайтеся демонстраційною програмою RSA для ознайомлення з за(роз)шифруванням інформації в криптографічній системі з відкритим ключем.

3. Побудуйте криптосистему RSA для $p = 5$, $q = 7$. Використовуйте перекодування алфавіту: $A = 1$, $B = 2$, $C = 3$, ... , $Z = 26$. Зашифруйте повідомлення *CRDHQS*, представивши результат у вигляді послідовності цифр.

4. Розшифруйте повідомлення $ASDFG$, отримавши послідовність цифр, скажімо, $C = c_1, c_2, c_3, c_4, c_5$. Сформуйте для передавання абонентові повідомлення C, завірене цифровим підписом.

5. Нехай задана криптосистема RSA з відкритим ключем (e,n) і блок вигляду $m = h^e \pmod n$. Сформуйте для передавання абонентові повідомлення m, завірене цифровим підписом.

6. Скласти звіт, додавши відповіді на контрольні питання й отримані результати.

Вимоги до звіту

У звіті повинні бути наведені:

- загальна структура і короткі відомості про криптосистему RSA;
- вхідні дані, проміжні і кінцеві результати завдань, отримані в ході виконання лабораторної роботи;
- значення p, q, n, $size(n)$, $\varphi(n)$, e, d, отримані при побудові криптосистеми RSA;
- висновки про властивості і якість вивченої криптосистеми;
- відповіді на контрольні питання.

Контрольні питання

1. У чому полягає основна суть і необхідність появи криптографічних систем з відкритим ключем?

2. На яких ідеях базується криптосистема RSA?

3. Порівняйте вивчений шифр із вивченими раніше шифрами.

4. Чому під час побудови криптосистеми RSA не використовують $e = 2$? Для $p = 3$, $q = 5$, $e = 2$ спробуйте знайти перебором значення d, обернене до e за $\mathrm{mod}(\varphi(n))$.

5. Скористайтеся програмою RSA для зашифрування деякого блоку m за допомогою криптосистеми RSA з випадковими p, q. Ця програма видає також проміжні дані для побудови відповідної криптосистеми. Побудуйте замість значення d величину D, яка задовольняє порівняння

$De = 1(\mathrm{mod}\,\lambda(n))$, де $\lambda(n) = \dfrac{\varphi(n)}{НСД(p-1, q-1)}$. Перевірте, що $m^{eD} = m(\mathrm{mod}\,n)$. Чи вірно це для будь-яких значень m?

6. Наскільки, на Ваш погляд, складними є програмна й апаратна реалізації вивченої криптосистеми?

Лабораторна робота № 7
РОЗПОДІЛ КЛЮЧІВ. ПРОТОКОЛ ДІФФІ-ХЕЛЛМАНА

Мета роботи – вивчити криптографічний протокол розподілу ключів Діффі-Хеллмана, на практиці здійснити формування загального ключа між двома користувачами. Усвідомити сильні і слабкі сторони в застосуванні даного протоколу.

Короткі теоретичні відомості

Під час побудови секретного зв'язку системи дуже важливою задачею є побудова системи керування ключами. Якою б складною і надійною не була обрана криптосистема, вона основана на використанні ключів. Якщо для забезпечення конфіденційного обміну інформацією між двома користувачами організувати процес обміну ключами легко, то в системі, де кількість користувачів складає десятки і сотні абонентів, керування ключами – серйозна проблема. Якщо не забезпечено досить надійне керування ключовою інформацією, то, заволодівши нею, зловмисник отримає несанкціонований доступ до секретних даних.

Під ключовою інформацією розуміється сукупність усіх діючих в інформаційній системі ключів.

Керування ключами – процес обробки і передавання інформації, який містить три складові:

– генерацію ключів;

– зберігання ключів;

– розподіл ключів.

Розглянемо, яким чином ці складові повинні бути реалізовані для того, щоб забезпечити безпеку ключової інформації в інформаційній системі.

У реальних криптосистемах використовуються спеціальні апаратні і програмні методи генерації випадкових ключів, необхідною умовою роботи яких є наявність деякого «випадкового фактора», наприклад, послідовності випадкових чисел. Ідеальними генераторами подібних послідовностей є пристрої на основі «природних» випадкових процесів. Наприклад, є генератори ключів на основі білого радіошуму. Програмні методи моделювання «випадкового фактора» являють собою так звані

програмні датчики псевдовипадкових послідовностей, які виробляють послідовності великих об'ємів, з необхідними статистичними властивостями, виходячи з відносно невеликих відрізків випадкових даних.

Під зберіганням ключів розуміється організація їхнього зберігання, обліку та видалення.

У достатньо складній системі один користувач може працювати з великим об'ємом ключової інформації. Іноді виникає необхідність організації спеціальних баз даних для ключової інформації. Такі бази даних забезпечують прийняття, зберігання, облік та видалення використовуваних ключів. *Секретні ключі ніколи не повинні записуватися в явному вигляді на носії, які можуть бути зчитаними або скопійованими.*

Отже, інформація про використовувані ключі повинна зберігатися в зашифрованому вигляді. Ключі, які зашифровують ключову інформацію, називаються майстер-ключами.

Дуже важливою умовою безпеки інформації є періодичне поновлення ключової інформації в системі. При цьому перепризначуватися повинні як звичайні ключі, так і майстер-ключі. В особливо відповідальних системах поновлення ключової інформації бажано робити щодня.

Питання поновлення ключової інформації пов'язано і з третьою складовою системи керування ключами – розподілом ключів.

Розподіл ключів – найвідповідальніший процес у керуванні ключами. До нього пред'являються дві вимоги:

– оперативність і точність розподілу;

– скритність розподілюваних ключів.

Для цих цілей ефективно використання криптосистем з відкритим ключем як засобу зашифрування ключів класичних (симетричних) криптосистем під час їхнього передавання каналами зв'язку. У цьому випадку за шифрування даних «відповідає» симетрична криптосистема, а за розповсюдження ключів – криптосистема з відкритим ключем. Подібним чином організовані криптосистеми називаються змішаними.

У цьому випадку проблема зводиться до того, щоб надійно засвідчити справжність ключів, тобто підтвердити ту обставину, що ключ, оголошений як відкритий, належить законному користувачеві мережі, а не сторонній особі. Іншими словами, повинна бути гарантована справжність

сеансу зв'язку, що приводить до необхідності попереднього обміну даними відповідно до встановлених правил. Подібні процедури називаються криптографічними протоколами.

Для обміну ключами можна використовувати алгоритм RSA, але досить ефективним виявився так званий протокол Діффі-Хеллмана, який дозволяє двом користувачам без посередників безпечно обмінятися ключем, який може бути використаний для симетричного шифрування.

У.Діффі і М.Хеллман запропонували для створення криптографічних систем з відкритим ключем використовувати однобічну функцію дискретного піднесення до степеня.

Необоротність перетворення в цьому випадку забезпечується тим, що досить легко обчислити показникову функцію вигляду $f(x)=a^x \pmod p$, де p – велике просте число (наприклад, розміру 256 бітів), однак знайти x з даного співвідношення, при коректному виборі p, практично неможливо.

Протокол Діффі-Хеллмана здійснює так званий експоненціальний ключовий обмін, з використанням функції $f(x)$, при заздалегідь вибраних несекретних параметрах a, p. Ці параметри залежні, у тому розумінні, що a вибирається так, щоб послідовність степенів

$$a^1 \bmod p,\ a^2 \bmod p,\ldots,a^{p-1} \bmod p$$

збігалася, з точністю до перестановки, з послідовністю $1,2,\ldots p-1$. Число a називається примітивним елементом або первісним коренем. Примітивні елементи за модулем простого числа завжди існують.

Протокол Діффі-Хеллмана вирішує задачу безпечної побудови загального ключа вигляду $k=a^{xy} \bmod p$, де x і y – випадкові лишки за модулем $p-1$, які вибираються учасниками ключового обміну незалежно і зберігаються в секреті. При цьому, $2 \le x,\ y \le p-2$. У ході протоколу учасники обмінюються значеннями $a^x \pmod p$ і $a^y \pmod p$. Оскільки одному з них відоме x, а іншому – y, то кожен учасник у змозі обчислити значення $k=a^{xy} \pmod p$. При перехопленні знання $a^x \pmod p$ не дозволяє знайти x (використовується однобічна функція), крім того, в даний час,

для знаходження $a^{xy}(\mathrm{mod}\,p)$ за $a^x(\mathrm{mod}\,p)$ і $a^y(\mathrm{mod}\,p)$, підходів, не еквівалентних оберненню однобічної функції, не знайдено.

Приклад організації системи експоненціального ключового обміну.

1. Виберемо просте число: $p=13$.
2. Виберемо первісний корінь:

   $a=7$, оскільки: $\{7^1,7^2,...,7^{12}\}=\{7,10,5,9,11,12,6,3,8,4,2,1\}$.
3. Опублікуємо a,p.

Приклад експоненціального ключового обміну між абонентами A, B.

1. A генерує псевдовипадкове число, наприклад, $x=8(\mathrm{mod}13)$ і передає B значення $a^x=7^8=3(\mathrm{mod}13)$.
2. B генерує псевдовипадкове число, наприклад, $y=5(\mathrm{mod}13)$ і передає A значення $a^y=7^5=11(\mathrm{mod}13)$.
3. A обчислює $k=\left(a^y\right)^x=11^8=9(\mathrm{mod}13)$.
4. B обчислює $k=\left(a^x\right)^e=3^5=9(\mathrm{mod}13)$.
5. Псевдовипадкове число k можна використовувати як ключ для симетричної криптосистеми.

Порядок виконання роботи

1. Вивчіть загальні відомості про системи керування ключами та опис протоколу експоненціального ключового обміну.
2. Скористайтеся демонстраційною програмою DIFHEL для засвоєння протоколу Діффі-Хеллмана на конкретному прикладі.
3. Знайдіть найменше від'ємне число, яке є первісним коренем за модулем 13.
4. Знайдіть число x, яке задовольняє порівняння $2^x=9(\mathrm{mod}\,p)$.
5. Побудуйте параметри для протоколу експоненціального ключового обміну при $p=11$ та опишіть хід протоколу для вибраних Вами x,y.
6. Здійсніть експоненціальний ключовий обмін за допомогою відповідної програми. Отримайте значення $a^{x-y}(\mathrm{mod}\,p)$, без знання x,y,

але за умови, що дані, якими абоненти обмінювалися в ході протоколу, відомі. *Підказка*: оскільки $a^{p-1} = 1 (\mathrm{mod}\, p)$, то $a^{-1} = a^{p-2} (\mathrm{mod}\, p)$.

7. Складіть звіт, додавши туди відповіді на контрольні питання та отримані результати.

Вимоги до звіту

У звіті повинні бути наведені:

– опис протоколу експоненціального ключового обміну;

– результати виконання завдань, отримані в ході виконання лабораторної роботи;

– відповіді на запитання (у тому числі, на контрольні).

Контрольні питання

1. Чому під час експоненціального ключового обміну необхідно уникати значень x, y рівних 0, 1, $p-1$?

2. Чи розв'язується порівняння $2^x = c (\mathrm{mod}\, p)$, при $p = 13$ і $1 \le c \le p-1$?

3. Чи розв'язується порівняння $5^x = 9 (\mathrm{mod} 13)$?

4. Який повинен бути розмір p (тобто $size(p)$), щоб можна було б під час експоненціального ключового обміну формувати ключі для алгоритму DES?

5. Вкажіть мінімальне значення $size(p)$ для формування сеансових ключів для алгоритму ГОСТ 28147-89.

6. З погляду загальної стійкості криптосистеми дайте відповідь, для якого алгоритму – DES або ГОСТ експоненціальний ключовий обмін більш прийнятний?

7. Наскільки, на ваш погляд, складними є програмна і апаратна реалізації вивченого протоколу ключового обміну?

Лабораторна робота № 8
ЕЛІПТИЧНІ КРИВІ В КРИПТОГРАФІЇ

Мета роботи – ознайомитися з основними поняттями про еліптичні криві й еліптичні групи за модулем простого числа, вивчити основні підходи до використання еліптичних кривих у криптографічних алгоритмах. Усвідомити сильні та слабкі сторони криптографічних алгоритмів, які базуються на еліптичних кривих.

Короткі теоретичні відомості

Останнім часом кількість бітів ключа, необхідних для надійної захищеності алгоритму RSA, різко зросла. Це породило безліч проблем, у тому числі під час вирішення задач електронної торгівлі, де потрібен захист великої кількості транзакцій. Задача пошуку надійних криптографічних методів захисту інформації з меншою кількістю бітів ключа є досить актуальною.

В цьому зв'язку, все частіше починають застосовувати криптографічні методи на основі еліптичних кривих. Привабливість такого підходу, у порівнянні з криптосистемою RSA, полягає в тому, що з використанням еліптичних кривих забезпечується еквівалентний захист при значно меншій кількості розрядів ключа, внаслідок чого зменшується кількість операцій під час виконання криптографічних операцій.

Загальні відомості про еліптичні криві. Еліптичні криві називаються так тому, що вони описуються кубічними рівняннями, подібними тим, які описують обчислення кривої еліпса. У загальному випадку кубічні рівняння для еліптичних кривих мають вигляд $y^2 + axy + by = x^3 + cx^2 + dx + e$, де дійсні числа a, b, c, d і e задовольняють деякі прості умови. Якщо точки еліптичної кривої лежать на прямій лінії, то їхня сума дорівнює деякому елементу O, який називається невласним (нескінченним, нульовим) елементом. Для точок еліптичної кривої справедливі такі правила додавання.

1. Об'єкт O виступає в ролі нульового елемента під час додавання. Так, $O = -O$ і для будь-якої точки P на еліптичній кривій $P + O = P$.

2. Вертикальна лінія перетинає криву в двох точках з однією й тією ж координатою x $P_1=(x,y)$ і $P_2=(x,-y)$. Ця лінія перетинає криву й у нескінченній точці. Тому $P_1+P_2+O=O$ і $-P_1=P_2$. Таким чином, точкою зі знаком «мінус» є точка з тією ж координатою x, але з протилежною за знаком координатою y (рис. 8.1).

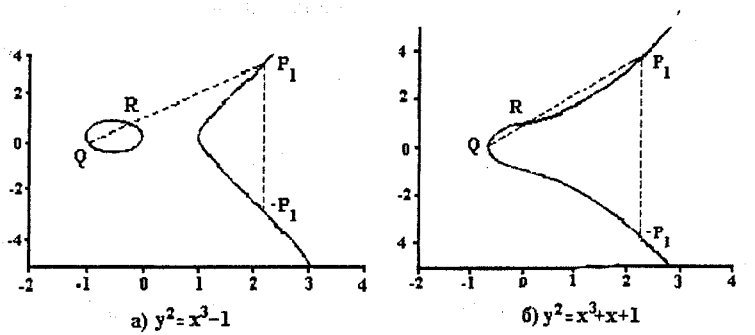


Рис. 8.1. Приклади еліптичних кривих

3. Щоб додати дві точки еліптичної кривої Q і R з різними координатами x, необхідно провести через ці точки пряму лінію і знайти третю точку перетину P_1 цієї прямої з еліптичною кривою (якщо пряма не є дотичною до кривої, то існує тільки одна така точка). Потім, використовуючи те, що $Q+R+P_1=O$, отримаємо $Q+R=-P_1$ (рис. 8.1).

4. Щоб подвоїти точку Q, необхідно провести дотичну в точці Q і знайти іншу точку перетину S. Тоді $Q+Q=2Q=-S$.

Вищеперераховані правила додавання підкоряються всім звичайним властивостям додавання, наприклад комутативному і асоціативному законам. Множення точки P еліптичної кривої на додатне ціле число k визначається як сума k копій точки P.

Еліптичні криві над скінченними полями. У задачах криптографії цікаві еліптичні криві, які визначаються над скінченними полями. Особливий інтерес викликають еліптичні групи за модулем простого числа p. Подібна група визначається таким чином. Виберемо два невід'ємних цілих числа a і b, які менші p і задовольняють, наприклад, умові

$$4a^3+27b^2 \pmod p \neq 0.$$

Тоді $E_p(a,b)$ означає еліптичну групу за модулем p, елементами якої є пари невід'ємних цілих чисел (x,y), які менші p і задовольняють умову

$$y^2 \equiv x^3 + ax + b \pmod{p} \qquad (1)$$

разом із точкою в нескінченності O.

Для еліптичної групи розглядаються тільки цілі значення від $(0,0)$ до (p,p) у квадранті невід'ємних чисел, які задовольняють рівняння за модулем p.

Наприклад, розглянемо еліптичну криву $y^2 = x^3 + x + 1 \pmod{23}$, тобто $a = b = 1$ і $p = 23$. У цьому випадку маємо

$$4 \times 1^3 + 27 \times 1^2 \pmod{23} = 8 \neq 0,$$

що задовольняє умови еліптичної групи за модулем 23. Зображення даної еліптичної кривої приведено на рис.8.1,*б*.

Приведемо всі точки, які відмінні від O і є елементами $E_{23}(1,1)$: (0,1); (0,22); (1,7); (1,16); (3,10); (3,13); (4,0); (5,4); (5,19); (6,4); (6,19); (7,11); (7,12); (9,7); (9,16); (11,3); (11,20); (12,4); (12,19); (13,7); (13,16); (17,3); (17,20); (18,3); (18,20); (19,5); (19,18).

У загальному випадку такий список створюється за правилами.

1. Для кожного значення x, де $0 \le x < p$, обчислюється $x^3 + ax + b \pmod{p}$.
2. Для кожного з отриманих на попередньому кроці значень з'ясовується, чи має це значення квадратний корінь за модулем p. Якщо ні, то в $E_p(a,b)$ немає точок з цим значенням x. Якщо ж корінь існує, є два значення y існуючих операцій витягнення квадратного кореня (винятком є випадок, коли єдиним таким значенням виявляється $y = 0$). Ці значення (x,y) і будуть точками $E_p(a,b)$.

Правила для додавання в $E_p(a,b)$ відповідають геометричним прийомам, показаним на рис.8.1. Формально ці прийоми для всіх точок $E_p(a,b)$ можуть бути записані так.

1. $P + O = P$.

2. Якщо $P=(x,y)$, то $P+(x,-y)=O$. Точка $(x,-y)$ є від'ємним значенням точки P і позначається $-P$. Відзначимо, що $(x,-y)$ лежить на еліптичній кривій і належить $E_p(a,b)$.

3. Якщо $P=(x_1,y_1)$ і $Q=(x_2,y_2)$, де $P\neq Q$, то $P+Q=(x_3,y_3)$ визначається відповідно до правил

$$x_3 \equiv \lambda^2 - x_1 - x_2 \pmod p,$$
$$y_3 \equiv \lambda(x_1 - x_3) - y_1 \pmod p,$$

де $$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, \text{ якщо } P \neq Q \\ \dfrac{3x_1^2 + a}{2x_1}, \text{ якщо P = Q} \end{cases}.$$

В криптографічних функціях на основі еліптичних кривих операція додавання є аналогом операції множення за модулем простого числа в алгоритмі RSA, а багаторазове повторне додавання – аналогом піднесення до степеня. Щоб побудувати криптографічну систему на основі еліптичних кривих, необхідно вирішити задачу, яка за складністю відповідає розкладанню на множники добутку двох простих чисел або дискретному логарифмуванню. Дійсно, якщо є рівняння $Q=kP$, де $Q,P\in E_p(a,b)$ і $k<p$, то відносно легко обчислити Q за даними k і P, але значно складніше визначити k, маючи Q і P.

Розглянемо два приклади використання еліптичних кривих у криптографії.

Обмін ключами з використанням еліптичних кривих може бути виконаний таким чином. Спочатку вибирається просте число, бажано $p\approx 2^{180}$, і параметри a і b для еліптичної кривої в рівнянні (1). Цим задається еліптична група точок $E_p(a,b)$. Потім в $E_p(a,b)$ вибирається генерувальна точка $G=(x,y)$. Під час вибору G важливо, щоб найменше значення n, при якому $nG=0$, виявилося дуже великим простим числом. Параметри $E_p(a,b)$ і G криптосистеми є параметрами, які відомі всім користувачам.

Обмін ключами між користувачами A і B можна провести за такою схемою.

1. Сторона A вибирає ціле число $n_A < n$, яке буде її особистим ключем. Потім вона генерує відкритий ключ $P_A = n_A \times G$, який являє собою деяку точку з $E_p(a,b)$.

2. Користувач B аналогічним способом вибирає особистий ключ n_B і обчислює відкритий ключ P_B.

3. Далі користувач A генерує секретний ключ $K_A = n_A \times P_B$, а користувач B – секретний ключ $K_B = n_B \times P_A$. Ці ключі рівні, оскільки $K_A = n_A \times P_B = n_A \times (n_B \times G) = n_B \times (n_A \times G) = n_B \times P_A = K_B$.

Загальний секретний ключ являє собою пару чисел. Якщо цей ключ передбачається використовувати як сеансовий ключа для традиційного шифрування, то з цієї пари чисел необхідно вибрати одне придатне значення. Можна, наприклад, використовувати просто координату x або деяку просту функцію від x. Щоб зламати описану схему, противник повинен буде обчислити значення k за відомими величинами G і kG, що являє собою важковирішувану задачу.

Як приклад візьмемо $p = 211$, $E_p(0,-4)$, що відповідає кривій $y^2 = x^3 - 4$, і $G = (2,2)$. Можна підрахувати, що значення $n = 241$, оскільки $241 \cdot G = 0$. Нехай особистим ключем користувача A є $n_A = 121$, тоді відкритим ключем A буде $P_A = 121(2,2) = (115,48)$. Нехай особистим ключем користувача B є $n_B = 203$, тоді його відкритим ключем буде $203(2,2) = (130,203)$. Загальний секретний ключ для цих користувачів обчислюється як $121(130,203) = 203(115,48) = (161,169)$.

Шифрування з використанням еліптичних кривих. Розглянемо процес шифрування відкритого повідомлення – тексту m. Шифрований текст буде представляти собою точку P_m на еліптичній кривій у вигляді координати $x - y$. Слід зазначити, що повідомлення кодується не будь-якою координатою x або y точки еліптичної кривої, оскільки не всі координати належать еліптичній групі $E_p(a,b)$. Як і у випадку розглянутої раніше системи обміну ключами, під час шифрування (дешифрування) в якості параметрів розглядається точка G і еліптична група $E_p(a,b)$.

Користувач A вибирає особистий ключ n_A і генерує відкритий ключ $P_B = n_A \times G$. Щоб зашифрувати і послати повідомлення P_m користувачеві B, користувач A вибирає випадкове додатне ціле число k і обчислює шифрований текст C_m, який складається з пари точок $C_m = \{kG, P_m + kP_B\}$. Відзначимо, що сторона A використовує відкритий ключ P_B сторони B. Щоб розшифрувати цей шифрований текст, користувач B множить першу точку в парі на секретний ключ B і віднімає від другої точки: $P_m + kP_B - n_B(kG) = P_m + k(n_B G) - n_B(kG) = P_m$. Користувач A замаскував повідомлення P_m за допомогою додавання до нього kP_B. Ніхто, крім цього користувача, не знає значення k, тому, хоча P_B і є відкритим ключем, ніхто не зможе забрати маску kP_B. Однак користувач A включає в повідомлення «підказку», якої буде достатньо, щоб маску забрав той, хто має особистий ключ n_B. Противнику для відновлення повідомлення необхідно обчислити k за заданими G і kG, що представляється важковирішуваною задачею.

Як приклад шифрування розглянемо випадок $p = 751$, $E_p(-1,188)$, що відповідає кривій $y^2 = x^3 - x + 188$, і $G = (0,376)$. Припустимо, що користувач A збирається відправити користувачеві B повідомлення, яке кодується еліптичною точкою $P_m = (562,201)$, і що користувач A вибирає випадкове число $k = 386$. Відкритим ключем B є $P_B = (201,5)$. Ми знаємо $386(0,376) = (678,558)$ і $(562,201) + 386(201,5) = (385,328)$. Таким чином, користувач A повинен послати шифрований текст $\{(676,558),(385,328)\}$.

Безпека, яка забезпечується криптографічними алгоритмами на основі еліптичних кривих, залежить від того, наскільки важкою для вирішення буде задача визначення k за даними kP і P. Цю задачу зазвичай називають проблемою логарифмування на еліптичній кривій. Найбільш швидким із відкрито опублікованих на сьогодні методів логарифмування на еліптичній кривій є ρ-метод Полларда. У таблиці 8.1 приводиться порівняльна ефективність цього методу і методу розкладання на прості множники за допомогою решета в полі чисел загального вигляду. Одиницею складності задачі в даному випадку є MIPS-рік – об'єм

роботи, виконаної протягом року процесором, який обробляє один мільйон команд в секунду.

Таблиця 8.1. Порівняння ефективності методів

Розмір ключа (у бітах)	Складність задачі (у MIPS-роках)
1. Логарифмування на еліптичній кривій за допомогою ρ-методу Полларда	
150	$3{,}8\times10^{10}$
205	$7{,}1\times10^{18}$
234	$1{,}6\times10^{28}$
2. Розкладання на множники в цілих числах за допомогою методу решета в полі чисел загального вигляду	
512	3×10^{4}
768	2×10^{8}
1024	3×10^{11}
1280	1×10^{14}
1536	3×10^{16}
2048	3×10^{20}

З даних, наведених у таблиці, видно, що в порівнянні з алгоритмом RSA застосування криптографічних методів на основі еліптичних кривих дає приблизно той же рівень захисту при значно менших значеннях довжини ключа. До того ж при рівних довжинах ключів обчислювальні витрати, необхідні під час використання алгоритму RSA і алгоритму на основі еліптичних кривих, не сильно відрізняються. Таким чином, у порівнянні з RSA, при рівних умовах захисту явна обчислювальна перевага належить криптографічним алгоритмам на основі еліптичних кривих з більш короткою довжиною ключа.

Порядок виконання роботи

1. Вивчіть загальні відомості про еліптичні криві.
2. Вивчіть особливості еліптичної групи за модулем простого числа.

3. Для еліптичної кривої $E_{11}(1,6)$, тобто заданої рівнянням $y^2 = x^3 + x + 6 (\mathrm{mod} 11)$, визначте всі точки $E_{11}(1,6)$. *Підказка*: Почніть з обчислення значень правої частини рівняння для всіх значень x.

4. Для випадку $E_{11}(1,6)$ обчисліть числа, які кратні точці $G = (2,7)$, починаючи від $2G$ і закінчуючи $13G$.

5. Для випадку еліптичної групи $E_{23}(1,1)$ знайдіть від'ємне значення $-P$ для точки $P = (13,7)$ і суму двох точок $P = (3,10)$ і $Q = (9,7)$.

6. Користувач B має секретний ключ $n_B = 7$ для криптосистеми, яка базується на еліптичних кривих і має параметри $E_{11}(1,6)$ і $G = (2,7)$. Необхідно:

- знайти відкритий ключ P_B користувача B;
- визначити шифрований текст C_m, який сформований користувачеві B під час зашифрування повідомлення $P_m = (10,9)$ з вибраним випадковим числом $k = 3$;
- записати формули, за допомогою яких користувач B відновлює P_m з C_m.

7. Складіть звіт, додавши туди отримані результати.

Вимоги до звіту

У звіті повинні бути наведені:

- короткі відомості про еліптичні криві й еліптичні групи за модулем простого числа;
- блок-схеми алгоритмів обміну ключами і шифрування (розшифрування), що базуються на еліптичних кривих;
- результати завдань, отримані в ході виконання лабораторної роботи;
- висновки про властивості і якості вивчених криптосистем.

Контрольні питання

1. Дайте визначення еліптичної кривої і її властивостей.
2. Які особливості мають точки еліптичної групи за модулем p. Чому число p повинно бути простим?

3. Порівняйте між собою алгоритм ключового обміну на основі еліптичних кривих і алгоритм Діффі-Хеллмана. У чому їхня схожість і відмінності?

4. У чому полягає суть шифрування з використанням еліптичних кривих?

5. Які переваги дає використання в криптографії алгоритмів на основі еліптичних кривих?

Лабораторна робота № 9
ГЕНЕРУВАННЯ ВИПАДКОВИХ ЧИСЕЛ

Мета роботи – усвідомити важливість проблеми генерування випадкових чисел під час вирішення задач захисту інформації, ознайомитися з деякими способами генерування псевдовипадкових чисел, усвідомити сильні і слабкі сторони алгоритмічних методів генерування випадкових чисел.

Короткі теоретичні відомості

Випадкові числа відіграють важливу роль при вирішенні різних задач захисту інформації. Їх використовують під час генерування сеансових ключів, отримання ключів для алгоритму RSA, у протоколах взаємної ідентифікації учасників інформаційного обміну та ін. Більшість застосувань в галузі захисту інформації висувають до випадкової послідовності дві вимоги: випадковість і непередбачуваність.

Випадковість. Традиційно під час генерування послідовності нібито випадкових чисел потрібно, щоб послідовність отримуваних чисел була випадковою в деякому цілком визначеному статистичному розумінні. Для перевірки будь-якої послідовності на випадковість зазвичай служать два такі критерії:

– однорідність розподілу: розподіл чисел у послідовності повинен бути однорідним, тобто частота появи в послідовності конкретного значення повинна бути однаковою для всіх значень;

– незалежність: жодне зі значень послідовності не повинно логічно виводитися з інших значень.

Є цілком чіткі алгоритми для перевірки відповідності послідовності заданому розподілу (наприклад, однорідному), але не існує алгоритмів, які дозволили б довести незалежність випадкової послідовності. У цьому випадку застосовується ряд тестів, які дозволяють продемонструвати, що послідовність не є незалежною. Загальна стратегія полягає в застосуванні такого набору тестів, що незалежність послідовності не стане правдоподібною.

Непередбачуваність. У деяких криптографічних застосуваннях (наприклад, під час взаємної ідентифікації абонентів або під час

генерування сеансових ключів) вимога статистичної випадковості послідовності чисел виявляється менш важливою, ніж вимога непередбачуваності елементів послідовності. В істинно випадковій послідовності кожне число статистично незалежне від інших чисел послідовності й таким чином непередбачуване. Однак істинно випадкові числа використовуються дуже рідко. Частіше застосовуються послідовності чисел, які виглядають випадковими, але насправді генеруються за допомогою деякого алгоритму. Такі числа називаються псевдовипадковими. У цьому випадку доводиться турбуватись, щоб противник не мав можливості передбачити наступні елементи послідовності на основі попередніх.

Джерелами істинно випадкових чисел можуть бути фізичні генератори шумів, такі як імпульсні детектори іонізуючого випромінювання, газорозрядні лампи і конденсатори з витоком струму. Однак такі пристрої досить обмежені для використання в задачах захисту інформації, тому що існують проблеми як з випадковістю, так і з точністю отримуваних чисел, не говорячи вже про проблеми підключення таких пристроїв у засоби захисту. Внаслідок цього в криптографічних застосуваннях зазвичай використовуються алгоритмічні методи генерування псевдовипадкових чисел. Відповідні алгоритми є детермінованими і тому породжують послідовності чисел, які статистично не випадкові. Однак якщо алгоритм досить хороший, то породжувані ним послідовності чисел витримують багато розумних тестів на випадковість.

Генератори псевдовипадкових чисел. Найпопулярнішим алгоритмом для генерування псевдовипадкових чисел є алгоритм, який називається лінійним конгруентним генератором. Лінійний конгруентний генератор породжує псевдовипадкову послідовність $x_1, x_2, ... \in \{0,1,...,N-1\}$ за допомогою рекурентного співвідношення

$$x_{t+1} = (ax_t + c) \bmod N, \text{ для } t = 0,1,... \qquad (1)$$

де N – модуль порівняння, $N > 0$; a – множник, $0 < a < N$; c – приріст, $0 \le c < N$; x_0 – початкове (породне) число, $0 \le x_0 < N$. Якщо числа N, a, c і x_0 є цілими, то буде отримана послідовність цілих чисел з діапазону $0 \le x_t < N$. Бажано, щоб число N було дуже великим, щоб потенційно могли генеруватися дуже довгі серії різних випадкових чисел.

Вибір значень для N, a і c виявляється дуже важливим з погляду розробки хорошого генератора псевдовипадкових чисел. Розглянемо, наприклад, випадок $a=c=1$. Породжувана при цьому послідовність, очевидно, не буде задовільною. Тепер розглянемо значення $a=7$, $c=0$, $N=32$ і $x_0=1$. У цьому випадку генерується послідовність $\{7,17,23,1,7,...\}$, яка також, очевидно, не буде задовільною. З 32 можливих значень тут виявляються задіяними тільки 4 (тобто існує період довжиною 4). Якщо змінити значення $a=7$, то результуючою послідовністю буде $\{1,5,25,29,17,21,9,13,1,...\}$ і її період буде дорівнювати 8.

Псевдовипадкова послідовність, породжувана лінійним конгруентним генератором, досягає максимального значення періоду $T_{\max}=N$ тоді і тільки тоді, коли виконані такі три умови:

– числа c і N повинні бути взаємно прості, тобто $НСД(c,N)=1$;

– число $b=a-1$ повинно бути кратне p для будь-якого простого числа $p<N$, яке є дільником N;

– число b кратне 4, якщо N кратне 4.

Узагальненням конгруентного генератора є лінійна рекурентна послідовність порядку $k\geq 1$ над скінченним полем F_N

$$x_{t+1}=(a_1x_t+a_2x_{t-1}+...+a_kx_{t-k+1})\bmod N \text{ для } t=0,1,..., \qquad (2)$$

де $a_1,a_2,...,a_k\in\{0,1,...,N-1\}$ – коефіцієнти рекурентної послідовності,

$x_0,x_{-1},...,x_{-k+1}\in\{0,1,...,N-1\}$ – початкові значення рекурентної послідовності.

Параметрами генератора псевдовипадкової послідовності вигляду (2) є значення N; k; $a_1,...,a_k$; $x_0,x_{-1},...,x_{-k+1}$. Початкові значення $x_0,x_{-1},...,x_{-k+1}$ вибираються довільно, так, щоб не перетворювалися в нуль одночасно. Коефіцієнти рекурентної послідовності $a_1,...,a_k$ вибираються таким чином, щоб породний поліном

$$f(x)=x^k-a_1x^{k-1}-...-a_{k-1}x-a_k \qquad (3)$$

був примітивним многочленом за модулем N, тобто многочлен (3) мав корінь $x*$, який є первісним елементом поля F_Nk. При такому виборі параметрів досягається максимально можливий період псевдовипадкової послідовності (1).

Як приклади ефективно реалізовних на комп'ютері алгоритмів генерації послідовностей вигляду (1) приведемо чотири генератори псевдовипадкових послідовностей з періодом $T_{\max} \approx 2^{96}$:

$$- x_{t+1} = \left(1176x_t + 1476x_{t-1} + ...1776x_{t-k+1}\right) \bmod \left(2^{35} - 5\right);$$

$$- x_{t+1} = 2^{13}\left(x_t + x_{t-1} + x_{t-2}\right) \bmod \left(2^{32} - 5\right);$$

$$- x_{t+1} = \left(1995x_t + 1998x_{t-1} + ...2001x_{t-k+1}\right) \bmod \left(2^{35} - 849\right);$$

$$- x_{t+1} = 2^{19}\left(x_t + x_{t-1} + x_{t-2}\right) \bmod \left(2^{32} - 1629\right).$$

Перевагою лінійного конгруентного генератора (лінійної рекурентної послідовності) є те, що якщо вибрати підходящий множник і модуль порівняння, то генеровна послідовність чисел виявляється статистично невідмінною від послідовності чисел, які вибираються випадково (але безповоротно) з множини чисел $1,2,...,N-1$. Разом з тим, у самому алгоритмі немає нічого випадкового взагалі, крім вибору початкового значення x_0. Якщо це значення вибрано, то інші числа послідовності визначаються однозначно. Крім цього, якщо противник зможе визначити послідовність x_0, x_1, x_2 і x_3, то він зможе вирішити відносно a, c і N таку систему рівнянь:

$$\begin{cases} x_1 = (ax_0 + c) \bmod N \\ x_2 = (ax_1 + c) \bmod N \\ x_3 = (ax_2 + c) \bmod N \end{cases}.$$

Отже, хоча і зручно використовувати хороший генератор псевдовипадкових чисел, бажано подбати про те, щоб генерована послідовність була в дійсності невідтворюваною, тобто знання частини послідовності не давало б противникові можливості визначити наступні елементи послідовності.

У криптографічних застосуваннях, для генерування випадкових чисел, є сенс використовувати вже наявну в системі захисту логіку шифрування. Розглянемо два таких підходи.

Генератор псевдовипадкових чисел ANSI X9.17. Один із кращих (з погляду криптографії) генераторів псевдовипадкових чисел визначається стандартом ANSI X9.17. Цей алгоритм використовується цілим рядом застосувань, серед яких безпека фінансових платежів і широко

розповсюджена програма криптографічного захисту PGP. На рис.9.1 показана схема алгоритму, у якому для шифрування використовується «потрійний» DES-алгоритм.

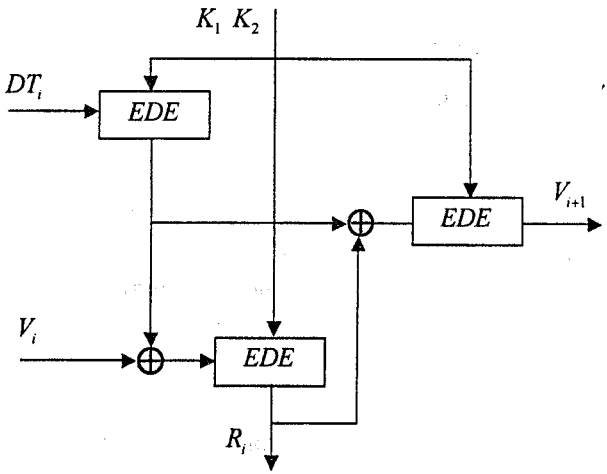


Рис. 9.1. Генератор псевдовипадкових чисел ANSI X9.17

Алгоритм має такі складові.

– **Введення**. На вхід генератора подається два псевдовипадкових значення. Одне з них є 64-бітовим представленням поточної дати і часу і змінюється для кожного нового генеровного числа. Інше являє собою 64-бітове початкове значення, яке ініціалізується до деякого довільного значення і поновлюється в процесі обчислень.

– **Ключі**. Генератор використовує три модулі шифрування «потрійного» DES. Кожний з цих трьох модулів використовує ту саму пару 56-бітових ключів, які повинні зберігатися в секреті і використовуватися тільки для генерування псевдовипадкових чисел.

– **Виведення**. Значеннями, які виводяться, є 64-бітове псевдовипадкове число і 64-бітове початкове значення.

Формулу генерації псевдовипадкових чисел можна представити таким чином:

$$R_i = EDE_{K1,K2}\left[V_i \oplus EDE_{K1,K2}\left[DT_i\right]\right],$$

$$V_{i+1} = EDE_{K1,K2}\left[R_i \oplus EDE_{K1,K2}\left[DT_i\right]\right],$$

де EDE – послідовність операцій шифрування–розшифрування–шифрування з використанням алгоритму DES із трьома ключами;

DT_i – значення дати–часу на початку i-ого циклу генерування псевдовипадкового числа;

V_i – початкове значення для i-ого циклу генерування псевдовипадкового числа;

R_i – псевдовипадкове число, яке отримується в результаті i-ого циклу генерування;

K_1, K_2 – ключі DES, які використовуються в кожному циклі.

Криптографічна надійність цього методу визначається декількома факторами. Тут використовується 112-бітовий ключ і три блоки шифрування EDE, які в сумі дають дев'ятикратне шифрування DES. Схема керується двома псевдовипадковими значеннями, які вводяться: значенням дати і часу та початковим значенням, яке виробляється генератором, відмінним від вироблюваного генератором псевдовипадкового значення. Таким чином, об'єм даних, що його повинен аналізувати противник, виявляється величезним. Навіть якщо йому стане відомим псевдовипадкове значення R_i, з цього факту неможливо буде вивести V_{i+1}, оскільки для генерування V_{i+1} використовується додаткова операція EDE.

Генератор BBS. Один з популярних підходів до генерування надійних послідовностей псевдовипадкових чисел полягає у використанні генератора BBS. Доведення його криптографічної надійності є найбільш строгим з опублікованих. Закладена в його алгоритмі процедура виглядає так.

1. Вибираються два великих простих числа p і q такі, що $N = p \times q$ і $p \equiv q \equiv 3 (\mathrm{mod}\, 4)$.
2. Вибирається випадкове число s, взаємно просте з N (у даному випадку це означає, що ні p, ні q не є дільниками s) і обчислюється значення $x_0 = s^2 \bmod N$.
3. Для генерування послідовності $B = \{b_i\}$ з l псевдовипадкових бітів необхідно для i від 1 до l обчислити: $b_i = x_{i-1} \bmod 2$ і

$x_i = (x_{i-1})^2 \bmod N$. Таким чином, на кожній ітерації в послідовність вибирається молодший біт.

Розглянемо приклад отримання псевдовипадкової послідовності в результаті використання алгоритму BBS. Нехай $p = 19$ і $q = 23$, тоді $N = 423$. Для $x_0 = 233$ отримуємо:

i	0	1	2	3	4	5	6	7	8	9	10	11	12
x_i	101	150	213	358	123	271	25	188	384	187	9	81	6
b_i	1	0	1	0	1	1	1	0	0	1	1	1	0

Генератор BBS зазвичай називають криптографічно захищеним генератором псевдовипадкових бітів. Цей генератор є одним з генераторів, які задовольняють так званий критерій наступного біта. Іншими словами, для псевдовипадкових бітів генератора BBS не існує практично ефективного алгоритму, який дозволив би за першими k бітами з'ясувати з ймовірністю більшою, ніж $1/2$, що наступний біт буде дорівнювати 1 (або 0). З погляду будь-якого практичного підходу, послідовність буде непередбачуваною. Захищеність генератора BBS базується на складності розкладання значення N на множники, тобто на складності задачі знаходження простих множників p і q за відомим числом N.

Порядок виконання роботи

1. Вивчіть короткі відомості про генерування випадкових чисел і наведені алгоритми.
2. Напишіть програмну реалізацію одного з представлених генераторів псевдовипадкових чисел. Виберіть правильні параметри для генератора. Отримайте послідовність псевдовипадкових чисел.
3. Дайте відповідь на контрольні питання.
4. Складіть звіт, додавши туди отримані результати.

Вимоги до звіту

У звіті повинні бути наведені:

- короткі відомості про значимість генераторів випадкових чисел у задачах захисту інформації;

– текст програми одного з генераторів псевдовипадкових чисел і результат її роботи (послідовність з 50 псевдовипадкових чисел);
– висновки про властивості і якість вивченого генератора псевдовипадкових чисел;
– відповіді на контрольні питання.

Контрольні питання

1. Якщо в рекурентному співвідношенні (1) лінійного конгруентного генератора приріст $c=0$, тобто співвідношення набуде вигляду $x_{t+1}=(ax_t)\bmod N$, то такий конгруентний генератор називається мультиплікативним. Можна показати, що якщо в мультиплікативному конгруентному генераторі N є простим і дане значення a породжує послідовності максимального періоду $N-1$, то a^k теж буде породжувати послідовності максимального періоду, якщо тільки k менше N і $N-1$ не ділиться на k. Продемонструйте це, розглянувши $x_0=1$ і $N=1$ та обчисливши послідовності для $a=3$, 3^2, 3^3 і 3^4.

2. В алгоритмі конгруентного генератора вибір параметрів, які забезпечують повний період, не обов'язково гарантує хорошу рандомізацію. Розглянемо, наприклад, два такі генератори: $x_{n+1}=(6x_n)\bmod 13$ і $x_{n+1}=(7x_n)\bmod 13$. Випишіть дві послідовності і переконайтеся, що обидві вони є повноперіодичними послідовностями. Яка з них виглядає більш випадковою?

3. Який максимальний період можна отримати від генератора $x_{n+1}=(ax_n \bmod 2^4)$? Яким при цьому повинно бути значення a? Які обмеження повинні накладатися на початкове значення?

4. Алгоритм генератора BBS має такі параметри: $p=383$, $q=503$ і $s=101355$. Отримайте перші 20 бітів псевдовипадкової послідовності цього генератора.

Лабораторна робота № 10
СТЕГАНОГРАФІЧНИЙ ЗАХИСТ ІНФОРМАЦІЇ

Мета роботи – усвідомити принципи, на яких базуються стеганографічні методи захисту інформації в комп'ютерних системах і мережах; ознайомитися з деякими способами приховування даних у текстових, звукових і графічних файлах; усвідомити сильні і слабкі сторони методів комп'ютерної стеганографії.

Короткі теоретичні відомості

Слово "стеганографія" у перекладі з грецької буквально означає "тайнопис" (steganos – секрет, таємниця; graphy – запис). До неї відноситься безліч секретних засобів зв'язку, таких, як невидиме чорнило, мікрофотознімки, умовне розташування знаків, таємні канали і засоби зв'язку на плаваючих частотах і т.д.

Сучасний прогрес в галузі глобальних комп'ютерних мереж і засобів мультимедіа привів до розробки нових методів, призначених для забезпечення безпеки передавання даних каналами телекомунікацій. Ці методи, з огляду на природні неточності пристроїв оцифровування і надлишковість аналогового відео- або аудіосигналу, дозволяють приховувати повідомлення в комп'ютерних файлах (контейнерах).

Комп'ютерна стеганографія – науковий напрямок захисту інформації, який вивчає принципи, засоби і методи організації прихованого передавання і зберігання даних з використанням сучасних комп'ютерних і комунікаційних технологій. На відміну від криптографії, де «противник» може точно визначити чи є передаване повідомлення зашифрованим текстом, методи стеганографії дозволяють приховати сам факт існування секретного повідомлення.

У зв'язку зі зростанням ролі глобальних комп'ютерних мереж стає усе більш важливим значення стеганографії. В даний час стеганографічні системи активно використовуються для вирішення таких основних задач:

1. **Захист конфіденційної інформації від несанкціонованого доступу.** Ця область використання комп'ютерної стеганографії є найбільш ефективною під час вирішення проблеми захисту конфіденційної інформації. Так, наприклад, тільки одна секунда оцифрованого звуку з

частотою дискретизації 44100 Гц і рівнем відліку 8 бітів у стереорежимі дозволяє приховати за рахунок заміни найменш значимих молодших розрядів у приховуваному повідомленні близько 10 Кбайт інформації. При цьому зміна значень відліків складає менше 1 %. Така зміна практично не виявляється під час прослуховування файлу більшістю людей.

2. **Подолання систем моніторингу і керування мережними ресурсами.** Стеганографічні методи, спрямовані на протидію системам моніторингу і керування мережними ресурсами промислового шпигунства, дозволяють протистояти спробам контролю над інформаційним простором під час проходження інформації через сервери керування локальних і глобальних обчислювальних мереж.

3. **Камуфлювання програмного забезпечення.** Іншою важливою задачею стеганографії є камуфлювання програмного забезпечення. У тих випадках, коли використання програм незареєстрованими користувачами є небажаним, воно може бути закамуфльовано під стандартні універсальні програмні продукти (наприклад, текстові редактори) або приховано у файлах мультимедіа (наприклад, у звуковому супроводі комп'ютерних ігор).

4. **Захист авторського права на деякі види інтелектуальної власності.** Ще одною областю використання стеганографії є захист авторського права від піратства. На комп'ютерні графічні зображення наноситься спеціальна мітка, яка залишається невидимою для очей, але розпізнається спеціальним програмним забезпеченням. Такі програми уже використовуються в комп'ютерних версіях деяких журналів. Даний напрямок стеганографії призначений не тільки для обробки зображень, але і для файлів з аудіо- та відеоінформацією і покликаний забезпечити захист інтелектуальної власності.

Стеганографічна система (або стегосистема) – сукупність засобів і методів, які використовуються для формування прихованого каналу передавання інформації. Узагальнена модель стегосистеми представлена на рис. 10.1.

У комп'ютерних стегосистемах розрізняють два основних типи файлів: файл-повідомлення, який необхідно приховати, і файл-контейнер, який може бути використаний для приховування в ньому повідомлення.

Вбудоване (приховане) повідомлення – повідомлення, яке вбудовується в контейнер.

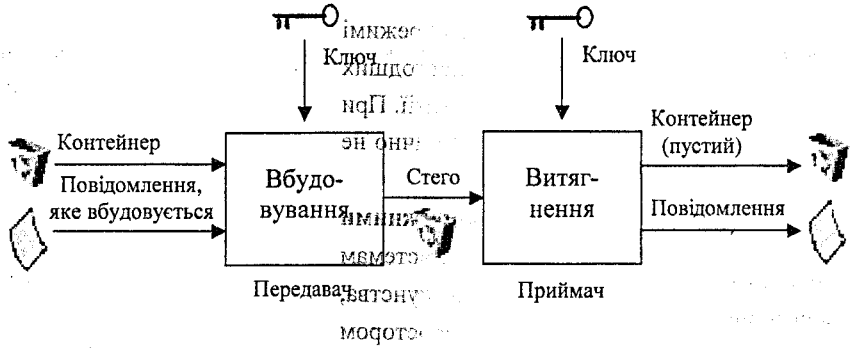


Рис. 10.1. Узагальнена модель стегосистеми

Контейнер – будь-яка інформація, призначена для приховання таємних повідомлень (текст, повідомлення, зображення і т.п.). Контейнери бувають двох типів. Контейнер-оригінал ("порожній" контейнер) — це контейнер, який не містить прихованої інформації. Контейнер-результат ("заповнений" контейнер або стеганограма) — це контейнер, який містить приховану інформацію.

Стеганографічний канал (або стегоканал) – це канал передавання стеганограми.

Стегоключ (ключ) – секретний елемент, який визначає порядок занесення повідомлення в контейнер. У залежності від кількості рівнів захисту (наприклад, вбудовування попередньо зашифрованого повідомлення) у стегосистемі може бути один або декілька стегоключів. За аналогією з криптографією, за типом стегоключа стегосистеми можна підрозділити на два типи: із секретним ключем і з відкритим ключем. У стегосистемі з секретним ключем використовується один ключ, який повинен бути визначений або до початку обміну секретними повідомленнями або переданий захищеним каналом. У стегосистемі з відкритим ключем для вбудовування та витягнення повідомлення використовуються різні ключі, які розрізняються таким чином, що за допомогою обчислень неможливо вивести один ключ з іншого. Тому один ключ (відкритий) може передаватися вільно незахищеним каналом зв'язку.

Основними положеннями сучасної комп'ютерної стеганографії є такі:

– стеганографічні методи повинні забезпечувати автентичність і цілісність файлу;

– передбачається, що противнику повністю відомі можливі стеганографічні методи, і він має повне уявлення про стеганографічну систему і деталі її реалізації;
– безпека методів ґрунтується на збереженні стеганографічним перетворенням основних властивостей відкрито передаваного файлу під час внесення в нього секретного повідомлення і деякої невідомої противнику інформації – ключа, за допомогою якого тільки його власник може встановити факт присутності і зміст прихованого повідомлення;
– якщо противник якимось чином довідається про факт існування прихованого повідомлення, це не повинно дозволити йому витягнути подібні повідомлення з інших даних доти, поки ключ зберігається в таємниці;
– витягнення самого секретного повідомлення представляє складну обчислювальну задачу, потенційний противник повинен бути позбавлений яких-небудь технічних і інших переваг у розпізнаванні або розкритті змісту таємних повідомлень.

Будь-яка стегосистема повинна відповідати таким вимогам.

1. Властивості контейнера повинні бути модифіковані так, щоб зміну неможливо було виявити під час візуального контролю. Ця вимога визначає якість приховання вбудовуваного повідомлення: для забезпечення безперешкодного проходження стегоповідомлення каналом зв'язку воно ніяким чином не повинно привернути увагу противника.
2. Стегоповідомлення повинно бути стійким до перекручувань, у тому числі і зловмисних. У процесі передавання зображення (звук або інший контейнер) може по-різному трансформуватись: зменшуватися або збільшуватися, перетворюватися в інший формат і т.д. Крім того, воно може бути ущільнено, у тому числі і з використанням алгоритмів ущільнення з втратою даних.
3. Для збереження цілісності вбудовуваного повідомлення необхідно використовувати код з виправленням помилки.
4. Для підвищення надійності вбудовуване повідомлення повинно бути продубльоване.

В наш час можна виділити два тісно пов'язаних між собою і маючих одні корені напрямку застосування комп'ютерної стеганографії:

приховування даних (повідомлень) і цифрові водяні знаки. Приховування даних, які у більшості випадків мають великий об'єм, висуває серйозні вимоги до контейнера: розмір контейнера в кілька разів повинен перевищувати розмір вбудовуваних даних. Цифрові водяні знаки використовуються для захисту авторських або майнових прав на цифрові зображення, фотографії або інші оцифровані твори мистецтва. Основними вимогами, які пред'являються до таких вбудованих даних, є надійність і стійкість до перекручувань. Цифрові водяні знаки мають невеликий об'єм, однак, з врахуванням зазначених вище вимог, для їхнього вбудовування використовуються більш складні методи, ніж для вбудовування просто повідомлень.

Кожне з перерахованих вище застосувань вимагає певного співвідношення між стійкістю вбудованого повідомлення до зовнішніх впливів (у тому числі і стегоаналізу) і розміром самого вбудовуваного повідомлення. Для більшості сучасних методів, які використовуються для приховування повідомлення в цифрових контейнерах, має місце залежність надійності системи від об'єму вбудовуваних даних, яка показана на рис.10.2. Дана залежність показує, що при збільшенні об'єму вбудовуваних даних знижується надійність системи (при незмінності розміру контейнера). Таким чином, використовуваний у стегосистемі контейнер накладає обмеження на розмір вбудовуваних даних.

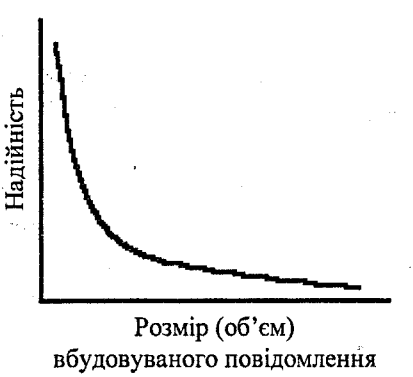


Рис. 10.2. Залежність надійності стегосистеми
від об'єму вбудовуваних даних

Істотно впливає на надійність стегосистеми і можливість виявлення факту передавання прихованого повідомлення вибір контейнеру. За довжиною контейнери можна підрозділити на два типи: безперервні (потокові) і обмеженої (фіксованої) довжини.

Особливістю потокового контейнера є те, що неможливо визначити його початок або кінець. Більш того, немає можливості довідатися заздалегідь, якими будуть наступні шумові біти, що приводить до необхідності включати біти, які приховують повідомлення, у потік в реальному масштабі часу, а самі приховуючі біти вибираються за допомогою спеціального генератора, який задає відстань між послідовними бітами в потоці. У безперервному потоці даних найбільші труднощі для приймача – визначити, коли починається приховуване повідомлення. При наявності в потоковому контейнері сигналів синхронізації або границь пакета приховане повідомлення починається відразу після одного з них. У свою чергу, для передавача можливі проблеми, якщо він не впевнений в тому, що потік контейнера буде досить довгим для розміщення цілого таємного повідомлення.

Під час використання контейнерів фіксованої довжини передавач заздалегідь знає розмір файлу і може вибрати приховуючі біти у придатній псевдовипадковій послідовності. З іншого боку, контейнери фіксованої довжини, як це уже відзначалося вище, мають обмежений об'єм і вбудовуване повідомлення іноді може не поміститися у файл-контейнер. Інший недолік полягає в тому, що відстані між приховуваними бітами рівномірно розподілені між найкоротшими і найдовшими заданими відстанями, у той час як істинний випадковий шум буде мати експоненціальний розподіл довжин інтервалу. Звичайно, можна породити псевдовипадкові експоненціально розподілені числа, але цей шлях зазвичай занадто трудомісткий. Однак на практиці найчастіше використовуються саме контейнери фіксованої довжини, як найбільш розповсюджені і доступні.

Можливі такі варіанти вибору контейнерів: контейнер генерується самою стегосистемою; контейнер вибирається з деякої множини контейнерів з метою вибору найбільш придатного для приховування повідомлення або ж контейнер надходить ззовні, що не завжди придатно до вбудовуваного повідомлення.

В наш час існує досить багато різних методів (та їх варіантів) вбудовування повідомлень у комп'ютерні файли. Найбільший розвиток отримали методи комп'ютерної стеганографії за двома основними напрямками:

– методи, які ґрунтуються на використанні спеціальних властивостей комп'ютерних форматів;

– методи, які використовують надлишковість аудіо і візуальної інформації.

Перший напрямок базується на використанні спеціальних властивостей комп'ютерних форматів представлення даних, а не на надлишковості самих даних. Спеціальні властивості форматів вибираються з врахуванням захисту приховуваного повідомлення від безпосереднього прослуховування, перегляду або прочитання.

Наприклад, для приховування інформації можна використовувати зарезервовані поля для розширення комп'ютерних форматів даних. Такі зарезервовані поля розширення є в багатьох мультимедійних форматах, вони зазвичай заповнюються нулями і не враховуються програмою. Перевагою такого методу є простота використання, а недоліком – низький ступінь скритності, передавання невеликих обмежених об'ємів. Іншим яскравим прикладом можуть служити методи спеціального форматування текстових файлів. Наприклад, методи використання відомого зміщення слів, речень та абзаців базуються на зміні положення рядків і розміщення слів у реченні, що забезпечується вставкою додаткових пропусків або зсувів між елементами тексту. Нижче наведена ілюстрація стеганографічної схеми, яка використовує зсуви слів у відформатованому тексті: у залежності від приховуваних даних змінюється горизонтальна позиція початку слів.

Приклад приховування даних у тексті

Приклад приховування даних у тексті

Приклад приховування даних у тексті

Найбільш популярними в комп'ютерній стеганографії є методи, які використовують надлишковість цифрових фотографій, звуку або

візуальної інформації. Цифрові фотографії, цифрова музика, цифрове відео – представляються матрицями чисел, які кодують інтенсивність у дискретні моменти в просторі і/або в часі. Цифрова фотографія – це матриця чисел, які представляють інтенсивність світла у певний момент часу. Цифровий звук – це матриця чисел, які представляють інтенсивність звукового сигналу в моменти часу, що йдуть послідовно. Усі ці числа неточні, тому що неточні пристрої оцифровування аналогових сигналів, є шуми квантування. Молодші розряди цифрових відліків містять дуже мало корисної інформації про поточні параметри звуку і візуального образу. Тому модифікація молодших бітів у більшості випадків не викликає значної трансформації зображення або звуку і не виявляється візуально.

Наприклад, графічні кольорові файли зі схемою змішання RGB кодують кожну точку малюнка трьома байтами. Кожна така точка складається з адитивних складових: червоного, зеленого, синього. Зміна кожного з трьох найменш значимих бітів приводить до зміни менше 1% інтенсивності даної точки. Це дозволяє приховати в стандартній графічній картинці об'ємом 800 Кбайт близько 100 Кбайт інформації, що не помітно під час перегляду зображення.

Стеганографічний метод, який для приховування інформації використовує молодші розряди цифрового представлення елементів зображення або звуку, називається методом найменшого значущого біта (LSB-метод). Цей метод є найбільш поширеним, але найменш стійким – він сильно чутливий до перекручувань. Його перевагою є можливість приховування великого об'єму інформації і можливість захисту авторського права, прихованого зображення торговельної марки, реєстраційних номерів і т.п. До недоліків цього методу слід віднести те, що за рахунок введення додаткової інформації спотворюються статистичні характеристики цифрових потоків, тому для зниження компрометувальних ознак потрібна корекція статистичних характеристик.

У мережі Інтернет можна вільно знайти програмні стегозасоби, призначені для роботи в різних операційних системах. Наприклад, для операційного середовища **Windows** існує програмний комплекс **Steganos**, який є легкою у використанні потужною програмою для шифрування файлів і приховування їх всередині BMP, DIB, VOC, WAV, ASCII, HTML файлів, і програмне забезпечення **Contraband**, яке дозволяє приховувати будь-які файли в 24 бітових графічних файлах формату BMP. Для

операційного середовища **DOS** доступна програма **Jsteg** для приховування інформації в популярному форматі JPG; пакет програм **StegoDos**, який дозволяє вибирати зображення, приховувати в ньому повідомлення, відображати і зберігати зображення в іншому графічному форматі; пакет програм **Wnstorm**, який дозволяє шифрувати повідомлення і приховувати його всередині графічного файлу PCX формату. Для операційного середовища **OS/2** доступна програма **Hide4PGP**, яка дозволяє приховувати інформацію у файлах формату BMP, WAV і VOC, при цьому для приховування можна використовувати будь-яку кількість наймолодших бітів; стеганографічна програма **Texto**, яка перетворює дані у беззмістовний англійський текст але який є достатньо близьким до нормального тексту; програма **Stego**, яка дозволяє вбудовувати дані у файли формату PICT без зміни зовнішнього вигляду і розміру PICT-файлу; а також програма **Paranoid**, яка дозволяє шифрувати дані за алгоритмами IDEA і DES, а потім приховувати файл у файлі звукового формату.

Порядок виконання роботи

1. Вивчіть короткі відомості про комп'ютерну стеганографію.
2. Знайдіть в Інтернеті одну з вільно розповсюджуваних програм стеганографічного захисту інформації. Дослідіть її можливості, вивчіть та опишіть метод приховування, який реалізує стеганографічний алгоритм.

Підказка. Програмне забезпечення стеганографічних засобів (включаючи вихідні тексти), наприклад, можна знайти за такими адресами в мережі Інтернет: ftp.crl.com; www.netlink.co.uk; www.rugeley.demon.co.uk; www.stego.com; www.demcom.com/english/steganos; www.cypher.net; ftp.funet.fi/pub/crypt/steganography і багатьох інших.

3. Напишіть програму, яка реалізує стеганографічне приховування інформації методом найменшого значущого біта в графічному файлі формату BMP або звуковому файлі формату WAV. Проведіть ряд експериментів з цією програмою, щоразу приховуючи в одному пікселі усе більше додаткової інформації (почніть з одного біта на піксел зображення, потім по два біти і т.д.). Визначте залежність величини перекручувань, які виникають у файлі-контейнері (звуковому або графічному) від кількості бітів, використовуваних для приховування інформації. Знайдіть границю

для припустимої кількості бітів, які можна приховати в кожному пікселі зображення (звуковому відліку), при яких перекручування несуттєві.

Підказка. Для того, щоб акуратно приховати інформацію у файлі-контейнері і не спотворити необхідні дані, що зберігаються в заголовку файлу (растра), необхідно попередньо вивчити формат графічних файлів BMP і формат звукових файлів WAV.

4. Відповісти на контрольні запитання.
5. Складіть звіт, додавши туди отримані результати.

Вимоги до звіту

У звіті повинні бути наведені:

- результати дослідження знайденої в мережі Інтернет стеганографічної програми (її можливості, з якими видами файлів-контейнерів вона працює, якого виду інформацію може приховувати, за яким принципом реалізований стеганографічний алгоритм, чи передбачена можливість попереднього шифрування приховуваної інформації та ін.), висновки про властивості і якість даної програми;
- текст складеної стеганографічної програми, яка реалізує приховування даних у графічних (звукових) файлах методом найменшого значущого біта, і результати експерименту за кількістю внесених додаткових бітів у кожен піксел зображення;
- відповіді на контрольні запитання.

Контрольні питання

1. Поясніть основні положення комп'ютерної стеганографії. У чому її відмінність від криптографії?
2. Для вирішення яких задач використовуються сучасні методи стеганографії?
3. Які типи контейнерів існують, у чому їхні переваги і недоліки?
4. Від яких параметрів залежить надійність стеганографічного методу?
5. В чому полягає стеганографічний метод найменшого значущого біта?

ЛІТЕРАТУРА

1. Организация и современные методы защиты информации. – М.: Концерн «Банковский Деловой Центр», 1998. – 465 с.

2. Мельников В.В. Защита информации в компьютерных системах. – М.: Финансы и статистика, Электроинформ, 1997. – 364 с.

3. Шеннон К.Э. Теория связи в секретных системах / Работы по теории информации и кибернетики. – М.:И.Л., 1963. – С.333–402.

4. Введение в криптографию / Под общей ред. В. В. Ященко. - СПб.: Питер, 2001. - 288 с.

5. Брюс Шнайер. Прикладная криптография. Протоколы алгоритмы и исходные тексты на языке С. 2-е издание. Пер. с англ. - М.: Мир, 1996. – 562 с.

6. Manazes A., van Oorschot, S. Vanstone. Handbook of Applied Cryptography. – CRC Press, 1996. – 782 p.

7. Жельников В. Криптография от папируса до компьютера. – М.: ABF, 1997. – 335 с.

8. Задірака В.К., Олексюк О.С. Комп'ютерна криптологія: Підручник. - Київ, 2002. – 504 с.

9. А. Саломаа. Криптография с открытым ключом: Пер. с англ. - М.: Мир, 1995. - 318 с.

10. Анохин М.И., Варнавский Н.П., Сидельников В.М., Ященко В.В. Криптография в банковском деле. – М.: МИФИ, 1997. – 358 с.

11. Дориченко С.А., Ященко В.В. 25 этюдов о шифрах. – М.: ТЭИС, 1994. – 203 с.

12. Хоффман Л.Д. Современные методы защиты информации. - М.: Сов.радио, 1980. - 264 с.

13. Спесивцев А.В., Вегнер В.А., Крутяков А.Ю. Защита информации в персональных ЭВМ. – М.: Радио и связь, 1992. – 192 с.

14. Болотов А.А., Гашков С.Б., Фролов А.Б., Часовских А.А. Алгоритмические основы эллиптической криптографии. –М.: МЭИ, 2000. –100 с.

15. Клопов В.А., Мотуз О.В. Основы компьютерной стеганографии // Информационно-методический журнал «Защита информации. Конфидент». – 1997. – №4. – С.43–48.

16. Хорошко В.О., Азаров О.Д., Шелест М.Є., Яремчук Ю.Є. Основи комп'ютерної стеганографії. Навчальний посібник. – Вінниця: ВДТУ, 2003. – 143 с.

17. Закон України “Про електронний цифровій підпис”.
18. Закон України “Про електронні документи та електронний документообіг”.

Навчальне видання

Хорошко Володимир Олексійович
Азаров Олексій Дмитрович
Шелест Михайло Євгенович
Мухачьов Вячеслав Андрійович
Андрєєв Володимир Іванович
Щербина Володимир Парфирович
Яремчук Юрій Євгенович

КОМП'ЮТЕРНА КРИПТОГРАФІЯ

Лабораторний практикум

Оригінал-макет підготовлено Яремчуком Ю.Є.
Редактор В.О. Дружиніна
Технічний редактор Н.К. Ніколаєва

Видавництво НАУ
Свідоцтво Держкомінформу України
серія ДК № 977 від 05.07.2002 р.
03058, м. Київ, проспект Космонавта Комарова, 1, НАУ

Підписано до друку 27.06.2003 р.
Формат 60×84/16
Друк різографічний
Тираж 300 прим.
Зам. №287/111

Гарнітура Times New Roman
Папір офсетний
Умовн. друк. арк. 6,04

Віддруковано в видавництві НАУ
серія ДК № 977 від 05.07.2002 р.
03058, м. Київ, проспект Космонавта Комарова, 1