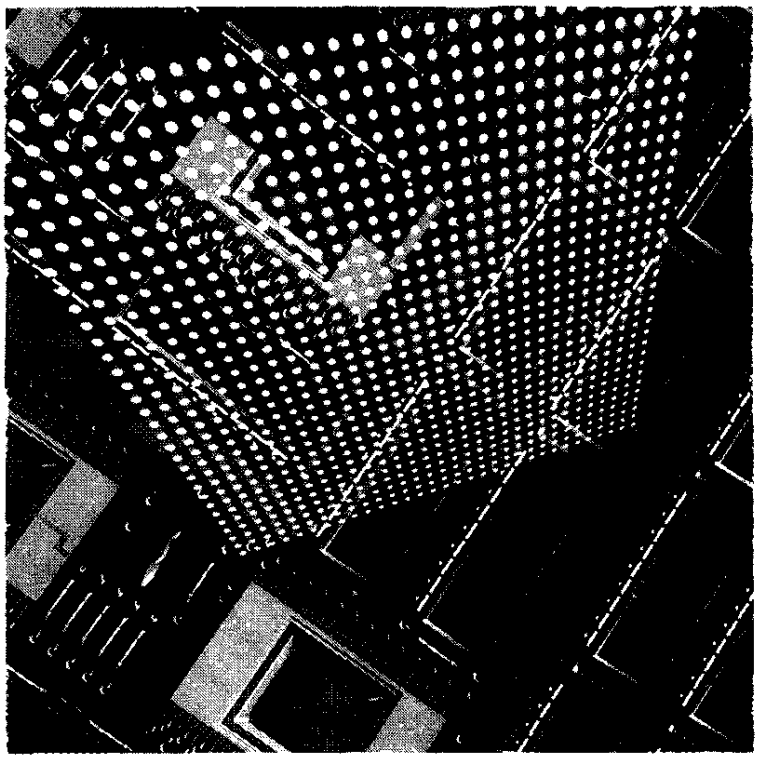
Ю.В. Романец П.А. Тимофеев В.Ф. Шаньгин

Crypton КРИПТОН

ЗАЩИТА информации в компьютерных системах и сетях

Радио и связь

Ю.В. Романец П.А. Тимофеев В.Ф. Шаньгин



ЗАЩИТА информации в компьютерных системах и сетях

Радио и связь

Ю.В. Романец, П.А. Тимофеев, В.Ф. Шаньгин

ЗАЩИТА
информации в компьютерных системах и сетях

Под редакцией доктора технических наук профессора В.Ф. Шаньгина

Издание второе, переработанное и дополненное

Москва
"Радио и связь"
2001

УДК 681.322
ББК 32.973
Р69

Романец Ю.В., Тимофеев П.А., Шаньгин В.Ф.

Р69 Защита информации в компьютерных системах и сетях / Под ред. В.Ф. Шаньгина. – 2-е изд., перераб. и доп. – М.: Радио и связь, 2001. – 376 с.: ил.

ISBN 5-256-01518-4.

Изложены как классические методы и средства шифрования, созданные в докомпьютерную эпоху, так и современные криптографические методы, алгоритмы, протоколы и средства защиты информации. Описаны методы и средства защиты локальных и корпоративных сетей от удаленных атак через сеть Internet. Рассмотрены вопросы обеспечения безопасности электронных платежных систем. Приводятся подробные сведения о серии эффективных отечественных аппаратно-программных средств криптографической защиты информации КРИПТОН.

Во втором издании книги учтены новые публикации в отечественной и зарубежной литературе и результаты новых разработок авторов.

Для разработчиков и пользователей компьютерных систем и сетей, а также для студентов, аспирантов и преподавателей вузов соответствующих специальностей.

ББК 32.973

Производственное издание

Романец Юрий Васильевич
Тимофеев Петр Александрович
Шаньгин Владимир Федорович

ЗАЩИТА ИНФОРМАЦИИ В КОМПЬЮТЕРНЫХ СИСТЕМАХ И СЕТЯХ

Редактор *Н.Г. Давыдова*
Художественный и технический редактор *Т.Н. Зыкина*

Компьютерная верстка *Р.А. Сафиной*

ИБ № 2943

Подписано в печать с оригинал-макета 19.02.2001 г.
Формат 60×90/16 Гарнитура Arial Печать офсетная
Усл. печ. л. 23,5 Усл. кр.-отт. 24,0 Уч.-изд. л. 21,98 Тираж 3000 экз.
Изд. № 24235 Зак. 49

Издательство "Радио и связь", 103473 Москва, 2-й Щемиловский пер., 4/5

Типография издательства "Радио и связь", 103473 Москва, 2-й Щемиловский пер., 4/5

ISBN 5-256-01518-4

Предисловие

Быстро развивающиеся компьютерные информационные технологии вносят заметные изменения в нашу жизнь. Все чаще понятие "информация" используется как обозначение специального товара, который можно приобрести, продать, обменять на что-то другое и т. п. При этом стоимость информации часто превосходит в сотни и тысячи раз стоимость компьютерной системы, в которой она находится. Поэтому вполне естественно возникает необходимость в защите информации от несанкционированного доступа, умышленного изменения, кражи, уничтожения и других преступных действий.

Проблемы защиты информации привлекают все большее внимание как специалистов в области компьютерных систем и сетей, так и многочисленных пользователей современных компьютерных средств. В то же время эта актуальная проблематика компьютерной науки и практики пока недостаточно освещена в отечественной научно-технической и учебной литературе.

Предлагаемая вниманию читателя книга посвящена методам и средствам защиты информации в компьютерных системах и сетях. В книге подробно излагаются классические методы шифрования и современные криптографические методы, алгоритмы, протоколы и системы. Описаны методы и средства защиты локальных и корпоративных сетей от удаленных атак через сеть Internet, а также рассмотрена защита информации в электронных платежных системах. Приводятся подробные данные об эффективных отечественных аппаратно-программных средствах криптографической защиты компьютерной информации.

По сравнению с первым изданием книги, осуществленным издательством "Радио и связь" в 1999 г., предлагаемое вниманию читателя второе издание книги "Защита информации в компьютерных системах и сетях" представляет собой существенно дополненный и переработанный вариант, в котором учтены новые открытые публикации в отечественной и зарубежной литературе, а также результаты новых разработок авторов.

В частности, во второе издание книги включены следующие новые разделы:

- Криптосистема с депонированием ключа.
- Типовые схемы идентификации и аутентификации пользователя. Средства биометрической идентификации.
- Цифровые подписи с дополнительными функциональными возможностями (схемы слепой цифровой подписи и неоспоримой цифровой подписи).
- Носители ключевой информации.
- Особенности протокола аутентификации и распределения ключей Kerberos.

Кроме того, значительно расширены и обновлены разделы, посвященные отечественным аппаратно-программным средствам криптографической защиты информации серии КРИПТОН/Crypton.

Книга содержит предисловие, введение, 12 глав, приложение и список литературы.

В главе 1 разъясняются основные понятия информационной безопасности компьютерных систем, рассматривается классификация угроз безопасности автоматизированных систем обработки информации (АСОИ), систематизируются меры обеспечения информационной безопасности АСОИ, формулируются концепции криптографической защиты информации, приводится классификация аппаратно-программных средств защиты компьютерной информации.

В главе 2 описываются классические методы симметричного шифрования от первых систем шифрования Спарты, Древней Греции и Рима до методов шифрования Вижинера и Вернама. Глава завершается разделом, посвященным шифрованию методом гаммирования, который является развитием метода Вернама и широко используется в настоящее время.

Глава 3 посвящена современным симметричным криптосистемам. Подробно рассматриваются американский стандарт шифрования данных DES, основные режимы работы алгоритма DES, приобретающий все большую популярность алгоритм шифрования данных IDEA и отечественный стандарт шифрования данных. Показано, как путем комбинирования блочных алгоритмов можно повысить их криптостойкость. Глава заканчивается разделом, в котором описана разработанная в США криптосистема с депонированием ключа. Эта криптосистема обеспечивает высокий уровень безопасности при передаче сообщений по открытым каналам связи и в то же время отвечает требованиям национальной безопасности, позволяя компетентным службам осуществлять расшифрование перехваченной информации после получения соответствующих санкций.

В главе 4 обсуждаются современные асимметричные криптосистемы (с открытыми ключами). Формулируется концепция по-

строения асимметричной криптосистемы, описываются однонаправленные (односторонние) функции, на которых базируется криптография с открытыми ключами. Особое внимание уделено криптосистеме RSA, получившей широкое распространение как в США, так и в других странах. Рассмотрены и другие схемы асимметричного шифрования. Показано, как, используя комбинированный метод шифрования, можно построить криптосистему, объединяющую достоинства симметричных и асимметричных криптосистем.

В главе 5 приводятся процедуры идентификации и аутентификации (проверки подлинности) объектов компьютерной системы и сети. Рассмотрены типовые схемы идентификации и аутентификации пользователя. Описаны схемы защиты пароля при аутентификации пользователя. Обсуждаются средства биометрической идентификации и аутентификации. Особое внимание уделяется проблеме взаимной проверки подлинности пользователей. Подробно обсуждаются современные протоколы идентификации с нулевой передачей знаний, применение которых особенно целесообразно в микропроцессорных смарт-картах.

Глава 6 посвящена электронной цифровой подписи, позволяющей решить проблему аутентификации электронного документа и его автора. Рассмотрены методы и алгоритмы хэширования исходного подписываемого текста. Подробно обсуждаются основные алгоритмы электронной цифровой подписи: RSA, Эль Гамаля, DSA и отечественный стандарт цифровой подписи. Глава завершается разделом, посвященным цифровым подписям с дополнительными функциональными возможностями. В этом разделе рассмотрены схемы слепой цифровой подписи и неоспоримой цифровой подписи, существенно расширяющие возможности обычной электронной цифровой подписи.

В главе 7 исследуются вопросы реализации таких основных функций управления криптографическими ключами, как генерация, хранение и распределение ключей. Описываются основные варианты носителей ключевой информации. Рассматривается концепция иерархии ключей. Особое внимание уделено самому ответственному процессу в управлении ключами – распределению ключей. Приводятся особенности протокола аутентификации и распределения ключей Kerberos, описываются алгоритм открытого распределения ключей Диффи–Хеллмана и протокол SKIP управления криптоключами.

В главе 8 обсуждаются методы и средства защиты локальных и корпоративных сетей от удаленных атак злоумышленников через сеть Internet. Описываются основные компоненты межсетевых экранов и схемы сетевой защиты на базе межсетевых экранов, приводятся программные методы защиты на базе криптопротоколов SKIP и SSL.

Глава 9 посвящена актуальным проблемам защиты информации в электронных платежных системах. Разбираются варианты

реализации электронных пластиковых карт. Обсуждаются способы генерации и защиты персонального идентификационного номера PIN держателя карты. Рассмотрены способы обеспечения безопасности систем POS и банкоматов. Особое внимание уделено перспективной универсальной электронной платежной системе UEPS, которая ориентирована на применение микропроцессорных смарт-карт. Заключительный раздел главы посвящен средствам обеспечения безопасности электронных платежей через сеть Internet. В частности, в нем рассматриваются особенности функционирования перспективного протокола обеспечения безопасности электронных транзакций SET (Secure Electronic Transaction).

В главе 10 описываются современные отечественные аппаратно-программные средства криптографической защиты компьютерной информации серии КРИПТОН, разработанные фирмой АНКАД в рамках научно-технического сотрудничества с Федеральным агентством правительственной связи и информации (ФАПСИ). Обсуждается концептуальный подход фирмы АНКАД к построению систем, надежно защищающих информацию в компьютере, где модулем безопасности выступают аппаратные средства серии КРИПТОН. Подробно рассмотрены характеристики плат серии КРИПТОН, отличительной особенностью которых является наличие в них отечественных шифропроцессоров. Описываются программные эмуляторы функций шифрования устройств КРИПТОН. Глава завершается разделом, посвященным системам защиты информации от несанкционированного доступа КРИПТОН-ВЕТО, КРИПТОН-ЗАМОК, Secret Disk.

В главе 11 подробно рассматриваются программные средства серии КРИПТОН/Crypton, применяемые для защиты несанкционированного доступа со стороны сети. Описываются программы абонентского шифрования и электронной цифровой подписи для MS-DOS и для ОС Windows 95/98/NT, средства пакетного шифрования. Предлагаются различные варианты использования средств для защиты абонентских пунктов, маршрутизаторов, серверов, отдельных сегментов локальных сетей и корпоративных сетей в целом.

В главе 12 показана возможность комплексного закрытия информации в электронных платежных системах с использованием аппаратно-программных средств серии КРИПТОН.

В основу книги положены материалы лекций, читаемых авторами на кафедре "Информатика и программное обеспечение вычислительных систем" Московского института электронной техники, а также результаты их научных и проектных работ, связанных с созданием аппаратных и программных средств криптографической защиты информации.

Весь материал книги базируется только на открытых публикациях в отечественной и зарубежной печати.

Авторы сознают, что содержание и качество книги могут быть со временем улучшены, и заранее благодарны читателям, которые пришлют им свои замечания и пожелания.

Введение

Проблемы защиты информации волновали человечество с незапамятных времен. Необходимость защиты информации возникла из потребностей тайной передачи, как военных, так и дипломатических сообщений. Например, античные спартанцы шифровали свои военные сообщения. У китайцев простая запись сообщения с помощью иероглифов делала его тайным для чужестранцев.

Для обозначения всей области тайной (секретной) связи используется термин "криптология", который происходит от греческих корней "cryptos"–тайный и "logos"–сообщение. Криптология довольно четко может быть разделена на два направления: криптографию и криптоанализ.

Задача криптографа – обеспечить конфиденциальность (секретность) и аутентичность (подлинность) передаваемых сообщений.

Задача криптоаналитика–"взломать" систему защиты, разработанную криптографами. Он пытается раскрыть зашифрованный текст или выдать поддельное сообщение за настоящее.

Первые каналы связи были очень простыми. Их организовывали, используя надежных курьеров. Безопасность таких систем связи зависела как от надежности курьера, так и от его способности не попадать в ситуации, при которых могло иметь место раскрытие сообщения.

Создание современных компьютерных систем и появление глобальных компьютерных сетей радикально изменило характер и диапазон проблем защиты информации. В широко компьютеризированном и информатизированном современном обществе обладание реальными ценностями, управление ими, передача ценностей или доступ к ним часто основаны на неовеществленной информации, т.е. на информации, существование которой не обязательно связывается с какой-либо записью на физическом носителе. Аналогичным образом иногда определяются и полномочия физических и юридических лиц на использование, модификацию, копирование имеющей большое значение или конфиденциальной информации. Поэтому весьма важно создавать и применять эффективные средства для реализации всех необходимых функций, связанных с обеспечением конфиденциальности и целостности информации.

Поскольку информация может быть очень ценной или особо важной, возможны разнообразные злонамеренные действия по отношению к компьютерным системам, хранящим, обрабатывающим или передающим такую информацию. Например, нарушитель может попытаться выдать себя за другого пользователя системы, подслушать канал связи или перехватить и изменить информацию, которой обмениваются пользователи системы. Нарушителем может быть и пользователь системы, который отказывается от сообщения, в действительности сформированного им, или который пытается утверждать, что им получено сообщение, которое в действительности не передавалось. Он может попытаться расширить свои полномочия, чтобы получить доступ к информации, к которой ему предоставлен только частичный доступ, или попытаться разрушить систему, несанкционированно изменяя права других пользователей.

Для решения указанных и других подобных проблем не существует какого-то одного технического приема или средства. Однако общим в решении многих из них является использование криптографии и криптоподобных преобразований информации.

На протяжении более чем тысячелетней истории криптографии она представляла собой постоянно обновляющийся и совершенствующийся набор технических приемов шифрования и расшифрования, которые сохранялись в строгом секрете.

Период развития криптологии с древних времен до 1949 г. принято называть *эрой донаучной криптологии*, поскольку достижения тех времен были основаны на интуиции и не подкреплялись доказательствами. Криптологией занимались тогда почти исключительно как искусством, а не как наукой. Конечно, это не означает, что история криптологии тех времен не представляет для нас никакого интереса. Более 2000 лет назад Юлий Цезарь писал Цицерону и друзьям в Риме, используя шифр, теперь названный его именем. Лишь с началом второй мировой войны криптологические службы воюющих держав осознали, что математики могут внести весомый вклад в развитие криптологии. В частности, в Англии в это время был призван на службу в качестве специалиста по криптологии Алан Тьюринг.

Публикация в 1949 г. статьи К. Шеннона "Теория связи в секретных системах" [83] стала началом новой *эры научной криптологии* с секретными ключами. В этой блестящей работе Шеннон связал криптографию с теорией информации.

С середины семидесятых годов (в связи с изобретением систем с открытым ключом) криптография не только перестала быть секретным сводом приемов шифрования-расшифрования, но и начала оформляться в новую математическую теорию. На последние двадцать лет пришлось значительное повышение активности в об-

ласти развития криптографии и ее применения для решения проблем защиты информации. Это вызвано широким признанием крайней необходимости в средствах обеспечения защиты информации во всех областях деятельности широко информатизированного человеческого сообщества и обусловлено появлением таких новых фундаментальных идей, как асимметричная (с открытым ключом) криптография, доказательно стойкие протоколы, надежность которых основана на гарантированной сложности решения математических задач, и т.д. [66].

В криптографической системе преобразование шифрования может быть симметричным или асимметричным относительно преобразования расшифрования. Соответственно различают два класса криптосистем:

- симметричные одноключевые криптосистемы (с секретным ключом);
- асимметричные двухключевые криптосистемы (с открытым ключом).

Современные симметричные одноключевые криптоалгоритмы базируются на принципах, изложенных в упомянутой работе Шеннона (1949). К ним относятся зарубежные криптоалгоритмы DES, IDEA и отечественный криптоалгоритм, описанный в стандарте ГОСТ 28147-89 [36]. Схемы реализации этих криптоалгоритмов открыто опубликованы и тщательно проанализированы многими исследователями. В этих криптосистемах секретным является только ключ, с помощью которого осуществляется шифрование и расшифрование информации. Данные криптосистемы могут использоваться не только для шифрования, но и для проверки подлинности (аутентификации) сообщений.

Появлению нового направления в криптологии – асимметричной криптографии с открытым ключом – способствовали две проблемы, которые не удавалось решить в рамках классической симметричной одноключевой криптографии.

Первая из этих проблем связана с распространением секретных ключей. Наличие секретного ключа, известного только получателю сообщения и его отправителю, столетиями считалось непременным условием безопасной передачи информации. Но при использовании симметричных криптосистем с секретными ключами требуют решения следующие вопросы. Как передать участникам обмена информацией сменяемые секретные ключи, которые требуются им для выполнения этого обмена? Как участники обмена смогут убедиться в целостности того, что они получили?

Вторая из этих проблем связана с формированием электронной цифровой подписи. В конце письма или другого авторизованного документа отправитель обычно ставит свою подпись. Подобное действие преследует две цели: во-первых, получатель может убе-

диться в подлинности письма, сличив подпись с имеющимся у него образцом; во-вторых, личная подпись является юридическим гарантом авторства документа. Этот аспект особенно важен при заключении разного рода торговых сделок, составлении обязательств, доверенностей и т.д. Подделать подпись человека на бумаге совсем не просто, а скопировать цепочку цифр на ЭВМ – несложная операция. Как в таком случае гарантировать подлинность и авторство электронных сообщений? В то же время существует много приложений, требующих достоверной цифровой подписи для цифровой информации, которая бы выполняла все те задачи, которые выполняет подпись, поставленная на документе рукой.

Обе эти проблемы казались трудноразрешимыми. Однако они были успешно решены с помощью криптографии с открытыми ключами. В опубликованной в 1976 г. статье "Новые направления в криптографии" У. Диффи и М. Хеллман впервые показали, что секретная связь возможна без передачи секретного ключа между отправителем и получателем. В основе этого криптографического метода лежат так называемые однонаправленные (односторонние) функции: при заданном значении x относительно просто вычислить значение f(x), однако, зная y= f(x), определить по y значение x чрезвычайно трудно.

В асимметричных криптосистемах с открытым ключом используются два ключа, по крайней мере, один из которых невозможно вычислить из другого. Один ключ используется отправителем для шифрования информации; другой – получателем для расшифрования получаемых шифртекстов. Обычно в приложениях один ключ должен быть несекретным, а другой – секретным.

Если ключ расшифрования невозможно получить из ключа зашифрования с помощью вычислений, то секретность информации, зашифрованной на несекретном (открытом) ключе, будет обеспечена. Однако этот ключ должен быть защищен от подмены или модификации, иначе отправитель может быть обманут и будет выполнять зашифрование на поддельном ключе, соответствующий ключ расшифрования которого известен противнику. Для того чтобы обеспечить закрытие информации, ключ расшифрования получателя должен быть секретным и физически защищенным от подмены. Так работает канал обеспечения конфиденциальности (секретности) информации.

Если же, наоборот, вычислительно невозможно получить ключ шифрования из ключа расшифрования, то ключ расшифрования может быть несекретным, а секретный ключ шифрования можно использовать для формирования электронной цифровой подписи под сообщением. В этом случае, если результат расшифрования цифровой подписи содержит аутентификационную информацию (заранее согласованную законным отправителем информации с

потенциальным получателем), эта подпись удостоверяет целостность сообщения, полученного от отправителя. Так работает канал аутентификации сообщения.

Кроме задачи аутентификации сообщения в проблеме аутентификации можно выделить еще две:

- задачу аутентификации пользователя – является ли пользователь, обращающийся к ресурсам компьютерной системы, именно тем, за кого он себя выдает?
- задачу взаимной аутентификации абонентов сети в процессе установления соединения между ними.

Обе эти задачи также успешно решаются с привлечением криптографических методов и средств.

Появление новых информационных технологий и интенсивное развитие компьютерных сетей привлекают все большее внимание пользователей к глобальной сети Internet. Многие компании и организации подключают сегодня свои локальные сети к сети Internet, чтобы воспользоваться ее ресурсами и преимуществами. Бизнесмены и государственные организации используют Internet в различных целях, включая обмен электронной почтой, распространение информации среди заинтересованных лиц и т.п. Подключение к Internet дает большие преимущества, однако при этом возникают серьезные проблемы с обеспечением информационной безопасности подключаемой локальной или корпоративной сети. В силу открытости своей идеологии Internet предоставляет злоумышленникам много возможностей для вторжения во внутренние сети предприятий и организаций с целью хищения, искажения или разрушения важной и конфиденциальной информации. Решение задач по защите внутренних сетей от наиболее вероятных атак через Internet может быть возложено на межсетевые экраны, иногда называемые брандмауэрами или firewall. Применяются и программные методы защиты, к которым относятся защищенные криптопротоколы SSL и SKIP.

Важным приложением, нуждающимся в эффективных средствах защиты информации, являются электронные платежные системы. В этих системах в качестве универсального платежного средства используются банковские пластиковые карты. Для обеспечения надежной работы электронная платежная система должна быть надежно защищена. С точки зрения информационной безопасности в системах электронных платежей существует ряд потенциально уязвимых мест, в частности, пересылка платежных и других сообщений между банками, между банком и банкоматом, между банком и клиентами. Для обеспечения защиты информации на отдельных узлах системы электронных платежей должны быть реализованы следующие механизмы защиты: управление доступом на оконечных системах, обеспечение целостности и конфиденциаль-

ности сообщений, взаимная аутентификация абонентов, гарантии доставки сообщения и т.д. Качество решения указанных проблем существенно зависит от рациональности выбора криптографических средств при реализации механизмов защиты.

Наиболее перспективным видом пластиковых карт являются микропроцессорные смарт-карты, которые благодаря встроенному микропроцессору обеспечивают обширный набор функций защиты и выполнение всех операций взаимодействия владельца карты, банка и торговца.

Все большее значение приобретает электронная торговля через Internet, которая сегодня может рассматриваться как огромный информатизированный рынок, способный охватить практически все население планеты Земля. Интенсивное развитие различных видов коммерческой деятельности в Internet требует принятия надлежащих мер по обеспечению безопасности этого перспективного вида электронной коммерции.

ГЛАВА 1. ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ КОМПЬЮТЕРНЫХ СИСТЕМ

1.1. Основные понятия и определения

Информатизация является характерной чертой жизни современного общества. Новые информационные технологии активно внедряются во все сферы народного хозяйства. Компьютеры управляют космическими кораблями и самолетами, контролируют работу атомных электростанций, распределяют электроэнергию и обслуживают банковские системы. Компьютеры являются основой множества автоматизированных систем обработки информации (АСОИ), осуществляющих хранение и обработку информации, предоставление ее потребителям, реализуя тем самым современные информационные технологии.

По мере развития и усложнения средств, методов и форм автоматизации процессов обработки информации повышается зависимость общества от степени безопасности используемых им информационных технологий, от которых порой зависит благополучие, а иногда и жизнь многих людей.

Актуальность и важность проблемы обеспечения безопасности информационных технологий обусловлены следующими причинами:

- резкое увеличение вычислительной мощности современных компьютеров при одновременном упрощении их эксплуатации;
- резкое увеличение объемов информации, накапливаемой, хранимой и обрабатываемой с помощью компьютеров и других средств автоматизации;
- сосредоточение в единых базах данных информации различного назначения и различной принадлежности;
- высокие темпы роста парка персональных компьютеров, находящихся в эксплуатации в самых разных сферах деятельности;
- резкое расширение круга пользователей, имеющих непосредственный доступ к вычислительным ресурсам и массивам данных;
- бурное развитие программных средств, не удовлетворяющих даже минимальным требованиям безопасности;
- повсеместное распространение сетевых технологий и объединение локальных сетей в глобальные;

- развитие глобальной сети Internet, практически не препятствующей нарушениям безопасности систем обработки информации во всем мире.

Введем и определим основные понятия информационной безопасности компьютерных систем [22, 89].

Под *безопасностью* АСОИ понимают ее защищенность от случайного или преднамеренного вмешательства в нормальный процесс ее функционирования, а также от попыток хищения, изменения или разрушения ее компонентов.

Природа воздействий на АСОИ может быть самой разнообразной. Это и стихийные бедствия (землетрясение, ураган, пожар), и выход из строя составных элементов АСОИ, и ошибки персонала, и попытка проникновения злоумышленника.

Безопасность АСОИ достигается принятием мер по обеспечению конфиденциальности и целостности обрабатываемой ею информации, а также доступности и целостности компонентов и ресурсов системы.

Под *доступом к информации* понимается ознакомление с информацией, ее обработка, в частности копирование, модификация или уничтожение информации.

Различают санкционированный и несанкционированный доступ к информации.

Санкционированный доступ к информации – это доступ к информации, не нарушающий установленные правила разграничения доступа.

Правила разграничения доступа служат для регламентации права доступа субъектов доступа к объектам доступа.

Несанкционированный доступ к информации характеризуется нарушением установленных правил разграничения доступа. Лицо или процесс, осуществляющие несанкционированный доступ к информации, являются нарушителями правил разграничения доступа. Несанкционированный доступ является наиболее распространенным видом компьютерных нарушений.

Конфиденциальность данных – это статус, предоставленный данным и определяющий требуемую степень их защиты. По существу конфиденциальность информации – это свойство информации быть известной только допущенным и прошедшим проверку (авторизированным) субъектам системы (пользователям, процессам, программам). Для остальных субъектов системы эта информация должна быть неизвестной.

Субъект – это активный компонент системы, который может стать причиной потока информации от объекта к субъекту или изменения состояния системы.

Объект – пассивный компонент системы, хранящий, принимающий или передающий информацию. Доступ к объекту означает доступ к содержащейся в нем информации.

Целостность информации обеспечивается в том случае, если данные в системе не отличаются в семантическом отношении от-

данных в исходных документах, т.е. если не произошло их случайного или преднамеренного искажения или разрушения.

Целостность компонента или ресурса системы – это свойство компонента или ресурса быть неизменными в семантическом смысле при функционировании системы в условиях случайных или преднамеренных искажений или разрушающих воздействий.

Доступность компонента или *ресурса* системы – это свойство компонента или ресурса быть доступным для авторизованных законных субъектов системы.

Под *угрозой безопасности* АСОИ понимаются возможные воздействия на АСОИ, которые прямо или косвенно могут нанести ущерб ее безопасности. *Ущерб безопасности* подразумевает нарушение состояния защищенности информации, содержащейся и обрабатывающейся в АСОИ. С понятием угрозы безопасности тесно связано понятие уязвимости АСОИ.

Уязвимость АСОИ – это некоторое неудачное свойство системы, которое делает возможным возникновение и реализацию угрозы.

Атака на компьютерную систему – это действие, предпринимаемое злоумышленником, которое заключается в поиске и использовании той или иной уязвимости системы. Таким образом, атака – это реализация угрозы безопасности.

Противодействие угрозам безопасности является целью защиты систем обработки информации.

Безопасная или *защищенная система* – это система со средствами защиты, которые успешно и эффективно противостоят угрозам безопасности.

Комплекс средств защиты представляет собой совокупность программных и технических средств, создаваемых и поддерживаемых для обеспечения информационной безопасности АСОИ. Комплекс создается и поддерживается в соответствии с принятой в данной организации политикой безопасности.

Политика безопасности – это совокупность норм, правил и практических рекомендаций, регламентирующих работу средств защиты АСОИ от заданного множества угроз безопасности.

1.2. Основные угрозы безопасности АСОИ

По цели воздействия различают три основных типа угроз безопасности АСОИ:

- угрозы нарушения конфиденциальности информации;
- угрозы нарушения целостности информации;
- угрозы нарушения работоспособности системы (отказы в обслуживании) [37].

Угрозы нарушения конфиденциальности направлены на разглашение конфиденциальной или секретной информации. При реализации этих угроз информация становится известной лицам, которые не должны иметь к ней доступ. В терминах компьютерной безопасности угроза нарушения конфиденциальности имеет место всякий раз, когда получен несанкционированный доступ к некоторой закрытой информации, хранящейся в компьютерной системе или передаваемой от одной системы к другой.

Угрозы нарушения целостности информации, хранящейся в компьютерной системе или передаваемой по каналу связи, направлены на ее изменение или искажение, приводящее к нарушению ее качества или полному уничтожению. Целостность информации может быть нарушена умышленно злоумышленником, а также в результате объективных воздействий со стороны среды, окружающей систему. Эта угроза особенно актуальна для систем передачи информации – компьютерных сетей и систем телекоммуникаций. Умышленные нарушения целостности информации не следует путать с ее санкционированным изменением, которое выполняется полномочными лицами с обоснованной целью (например, таким изменением является периодическая коррекция некоторой базы данных).

Угрозы нарушения работоспособности (отказ в обслуживании) направлены на создание таких ситуаций, когда определенные преднамеренные действия либо снижают работоспособность АСОИ, либо блокируют доступ к некоторым ее ресурсам. Например, если один пользователь системы запрашивает доступ к некоторой службе, а другой предпринимает действия по блокированию этого доступа, то первый пользователь получает отказ в обслуживании. Блокирование доступа к ресурсу может быть постоянным или временным.

Нарушения конфиденциальности и целостности информации, а также доступности и целостности определенных компонентов и ресурсов АСОИ могут быть вызваны различными опасными воздействиями на АСОИ.

Современная автоматизированная система обработки информации представляет собой сложную систему, состоящую из большого числа компонентов различной степени автономности, которые связаны между собой и обмениваются данными. Практически каждый компонент может подвергнуться внешнему воздействию или выйти из строя. Компоненты АСОИ можно разбить на следующие группы:

- *аппаратные средства* – ЭВМ и их составные части (процессоры, мониторы, терминалы, периферийные устройства – дисководы, принтеры, контроллеры, кабели, линии связи) и т.д.;

- *программное обеспечение* – приобретенные программы, исходные, объектные, загрузочные модули; операционные системы и системные программы (компиляторы, компоновщики и др.), утилиты, диагностические программы и т.д.;
- *данные* – хранимые временно и постоянно, на магнитных носителях, печатные, архивы, системные журналы и т.д.;
- *персонал* – обслуживающий персонал и пользователи.

Опасные воздействия на АСОИ можно подразделить на случайные и преднамеренные. Анализ опыта проектирования, изготовления и эксплуатации АСОИ показывает, что информация подвергается различным случайным воздействиям на всех этапах цикла жизни и функционирования АСОИ. Причинами *случайных воздействий* при эксплуатации АСОИ могут быть:

- аварийные ситуации из-за стихийных бедствий и отключений электропитания;
- отказы и сбои аппаратуры;
- ошибки в программном обеспечении;
- ошибки в работе обслуживающего персонала и пользователей;
- помехи в линиях связи из-за воздействий внешней среды.

Преднамеренные угрозы связаны с целенаправленными действиями нарушителя. В качестве нарушителя могут выступать служащий, посетитель, конкурент, наемник и т.д. Действия нарушителя могут быть обусловлены разными мотивами: недовольством служащего своей карьерой, сугубо материальным интересом (взятка), любопытством, конкурентной борьбой, стремлением самоутвердиться любой ценой и т.п.

Исходя из возможности возникновения наиболее опасной ситуации, обусловленной действиями нарушителя, можно составить гипотетическую модель потенциального нарушителя [57]:

- квалификация нарушителя может быть на уровне разработчика данной системы;
- нарушителем может быть как постороннее лицо, так и законный пользователь системы;
- нарушителю известна информация о принципах работы системы;
- нарушитель выберет наиболее слабое звено в защите.

В частности, для банковских АСОИ можно выделить следующие преднамеренные угрозы:

- несанкционированный доступ посторонних лиц, не принадлежащих к числу банковских служащих, и ознакомление с хранимой конфиденциальной информацией;
- ознакомление банковских служащих с информацией, к которой они не должны иметь доступ;
- несанкционированное копирование программ и данных;

- кража магнитных носителей, содержащих конфиденциальную информацию;
- кража распечатанных банковских документов;
- умышленное уничтожение информации;
- несанкционированная модификация банковскими служащими финансовых документов, отчетности и баз данных;
- фальсификация сообщений, передаваемых по каналам связи;
- отказ от авторства сообщения, переданного по каналам связи;
- отказ от факта получения информации;
- навязывание ранее переданного сообщения;
- разрушение информации, вызванное вирусными воздействиями;
- разрушение архивной банковской информации, хранящейся на магнитных носителях;
- кража оборудования [39].

Несанкционированный доступ (НСД) является наиболее распространенным и многообразным видом компьютерных нарушений. Суть НСД состоит в получении пользователем (нарушителем) доступа к объекту в нарушение правил разграничения доступа, установленных в соответствии с принятой в организации политикой безопасности. НСД использует любую ошибку в системе защиты и возможен при нерациональном выборе средств защиты, их некорректной установке и настройке. НСД может быть осуществлен как штатными средствами АСОИ, так и специально созданными аппаратными и программными средствами.

Перечислим основные каналы несанкционированного доступа, через которые нарушитель может получить доступ к компонентам АСОИ и осуществить хищение, модификацию и/или разрушение информации:

- все штатные каналы доступа к информации (терминалы пользователей, оператора, администратора системы; средства отображения и документирования информации; каналы связи) при их использовании нарушителями, а также законными пользователями вне пределов их полномочий;
- технологические пульты управления;
- линии связи между аппаратными средствами АСОИ;
- побочные электромагнитные излучения от аппаратуры, линий связи, сетей электропитания и заземления и др.

Из всего разнообразия способов и приемов несанкционированного доступа остановимся на следующих распространенных и связанных между собой нарушениях:

- перехват паролей;
- "маскарад";
- незаконное использование привилегий.

Перехват паролей осуществляется специально разработанными программами. При попытке законного пользователя войти в систему программа-перехватчик имитирует на экране дисплея ввод имени и пароля пользователя, которые сразу пересылаются владельцу программы-перехватчика, после чего на экран выводится сообщение об ошибке и управление возвращается операционной системе. Пользователь предполагает, что допустил ошибку при вводе пароля. Он повторяет ввод и получает доступ в систему. Владелец программы-перехватчика, получивший имя и пароль законного пользователя, может теперь использовать их в своих целях. Существуют и другие способы перехвата паролей.

"Маскарад" – это выполнение каких-либо действий одним пользователем от имени другого пользователя, обладающего соответствующими полномочиями. Целью "маскарада" является приписывание каких-либо действий другому пользователю либо присвоение полномочий и привилегий другого пользователя.

Примерами реализации "маскарада" являются:

- вход в систему под именем и паролем другого пользователя (этому "маскараду" предшествует перехват пароля);
- передача сообщений в сети от имени другого пользователя.

"Маскарад" особенно опасен в банковских системах электронных платежей, где неправильная идентификация клиента из-за "маскарада" злоумышленника может привести к большим убыткам законного клиента банка.

Незаконное использование привилегий. Большинство систем защиты устанавливают определенные наборы привилегий для выполнения заданных функций. Каждый пользователь получает свой набор привилегий: обычные пользователи – минимальный, администраторы – максимальный. Несанкционированный захват привилегий, например посредством "маскарада", приводит к возможности выполнения нарушителем определенных действий в обход системы защиты. Следует отметить, что незаконный захват привилегий возможен либо при наличии ошибок в системе защиты, либо из-за халатности администратора при управлении системой и назначении привилегий.

Особо следует остановиться на угрозах, которым могут подвергаться компьютерные сети. Основная особенность любой компьютерной сети состоит в том, что ее компоненты распределены в пространстве. Связь между узлами (объектами) сети осуществляется физически с помощью сетевых линий связи и программно с помощью механизма сообщений. При этом управляющие сообщения и данные, пересылаемые между объектами сети, передаются в виде пакетов обмена. При вторжении в компьютерную сеть злоумышленник может использовать как пассивные, так и активные методы вторжения [55].

При *пассивном вторжении* (*перехвате* информации) нарушитель только наблюдает за прохождением информации по каналу связи, не вторгаясь ни в информационный поток, ни в содержание передаваемой информации. Как правило, злоумышленник может определить пункты назначения и идентификаторы либо только факт прохождения сообщения, его длину и частоту обмена, если содержимое сообщения не распознаваемо, т.е. выполнить анализ трафика (потока сообщений) в данном канале.

При *активном вторжении* нарушитель стремится подменить информацию, передаваемую в сообщении. Он может выборочно модифицировать, изменить или добавить правильное или ложное сообщение, удалить, задержать или изменить порядок следования сообщений. Злоумышленник может также аннулировать и задержать все сообщения, передаваемые по каналу. Подобные действия можно квалифицировать как отказ в передаче сообщений.

Компьютерные сети характерны тем, что кроме обычных локальных атак, осуществляемых в пределах одной системы, против объектов сетей предпринимают так называемые *удаленные атаки*, что обусловлено распределенностью сетевых ресурсов и информации. Злоумышленник может находиться за тысячи километров от атакуемого объекта, при этом нападению может подвергаться не только конкретный компьютер, но и информация, передающаяся по сетевым каналам связи. Под удаленной атакой понимают информационное разрушающее воздействие на распределенную компьютерную сеть, программно осуществленное по каналам связи [56].

В табл.1.1 показаны основные пути реализации угроз безопасности АСОИ при воздействии на ее компоненты. Конечно, табл.1.1 дает самую общую картину того, что может произойти с системой. Конкретные обстоятельства и особенности должны рассматриваться отдельно. Более подробную классификацию угроз безопасности АСОИ можно найти в [22].

Таблица 1.1

Пути реализации угроз безопасности АСОИ

Объекты воздействия	Нарушение конфиденциальности информации	Нарушение целостности информации	Нарушение работоспособности системы
Аппаратные средства	НСД – подключение; использование ресурсов; хищение носителей	НСД – подключение; использование ресурсов; модификация, изменение режимов	НСД – изменение режимов; вывод из строя; разрушение
Программное обеспечение	НСД – копирование; хищение; перехват	НСД, внедрение "троянского коня", "вирусов", "червей"	НСД – искажение; удаление; подмена
Данные	НСД – копирование; хищение; перехват	НСД – искажение; модификация	НСД – искажение; удаление; подмена
Персонал	Разглашение; передача сведений о защите; халатность	"Маскарад"; вербовка; подкуп персонала	Уход с рабочего места; физическое устранение

В табл.1.1 приведены специфические названия и термины: "троянский конь", "вирус", "червь", которые употребляются для именования некоторых распространенных угроз безопасности АСОИ. Хотя эти названия имеют жаргонный оттенок, они уже вошли в общепринятый компьютерный лексикон. Дадим краткую характеристику этих распространенных угроз безопасности АСОИ.

"Троянский конь" представляет собой программу, которая наряду с действиями, описанными в ее документации, выполняет некоторые другие действия, ведущие к нарушению безопасности системы и деструктивным результатам. Аналогия такой программы с древнегреческим "троянским конем" вполне оправдана, так как в обоих случаях не вызывающая подозрений оболочка таит серьезную угрозу.

Термин "троянский конь" был впервые использован хакером Даном Эдварсом, позднее ставшим сотрудником Агентства Национальной Безопасности США. "Троянский конь" использует в сущности обман, чтобы побудить пользователя запустить программу со скрытой внутри угрозой. Обычно для этого утверждается, что такая программа выполняет некоторые весьма полезные функции. В частности, такие программы маскируются под какие-нибудь полезные утилиты.

Опасность "троянского коня" заключается в дополнительном блоке команд, вставленном в исходную безвредную программу, которая затем предоставляется пользователям АСОИ. Этот блок команд может срабатывать при наступлении какого-либо условия (даты, состояния системы) либо по команде извне. Пользователь, запустивший такую программу, подвергает опасности как свои файлы, так и всю АСОИ в целом. Приведем для примера некоторые деструктивные функции, реализуемые "троянскими конями".

- Уничтожение информации. Выбор объектов и способов уничтожения определяется фантазией автора вредоносной программы.
- Перехват и передача информации. В частности, известна программа, осуществляющая перехват паролей, набираемых на клавиатуре.
- Целенаправленная модификация текста программы, реализующей функции безопасности и защиты системы.

В общем, "троянские кони" наносят ущерб АСОИ посредством хищения информации и явного разрушения программного обеспечения системы. "Троянский конь" является одной из наиболее опасных угроз безопасности АСОИ. Радикальный способ защиты от этой угрозы заключается в создании замкнутой среды исполнения программ, которые должны храниться и защищаться от несанкционированного доступа.

Компьютерный "вирус" представляет собой своеобразное явление, возникшее в процессе развития компьютерной и информационной техники. Суть этого явления состоит в том, что программы-вирусы обладают рядом свойств, присущих живым организмам, – они рождаются, размножаются и умирают.

Термин "вирус" в применении к компьютерам был предложен Фредом Коэном из Университета Южной Калифорнии [88]. Исторически первое определение, данное Ф. Коэном: "Компьютерный вирус – это программа, которая может заражать другие программы, модифицируя их посредством включения в них своей, возможно, измененной копии, причем последняя сохраняет способность к дальнейшему размножению". Ключевыми понятиями в определении компьютерного вируса являются способность вируса к саморазмножению и способность к модификации вычислительного процесса. Указанные свойства компьютерного вируса аналогичны паразитированию в живой природе биологического вируса.

Вирус обычно разрабатывается злоумышленниками таким образом, чтобы как можно дольше оставаться необнаруженным в компьютерной системе. Начальный период "дремоты" вирусов является механизмом их выживания. Вирус проявляется в полной мере в конкретный момент времени, когда происходит некоторое событие вызова, например пятница 13-е, известная дата и т. п.

Компьютерный вирус пытается тайно записать себя на компьютерные диски. Способ функционирования большинства вирусов заключается в таком изменении системных файлов компьютера, чтобы вирус начинал свою деятельность при каждой загрузке. Например, вирусы, поражающие загрузочный сектор, пытаются инфицировать часть дискеты или жесткого диска, зарезервированную только для операционной системы и хранения файлов запуска. Эти вирусы особенно коварны, так как они загружаются в память при каждом включении компьютера. Такие вирусы обладают наибольшей способностью к размножению и могут постоянно распространяться на новые диски.

Другая группа вирусов пытается инфицировать исполняемые файлы, чтобы остаться необнаруженными. Обычно вирусы отдают предпочтение EXE- или COM-файлам, применяемым для выполнения кода программы в компьютерной системе. Некоторые вирусы используют для инфицирования компьютерной системы как загрузочный сектор, так и метод заражения файлов. Это затрудняет выявление и идентификацию таких вирусов специальными программами и ведет к их быстрому распространению. Существуют и другие разновидности вирусов. Компьютерные вирусы наносят ущерб системе за счет многообразного размножения и разрушения среды обитания.

Сетевой "червь" представляет собой разновидность программы-вируса, которая распространяется по глобальной сети и не оставляет своей копии на магнитном носителе. Термин "червь" пришел из научно-фантастического романа Джона Бруннера "По бурным волнам". Этот термин используется для именования программ, которые подобно ленточным червям перемещаются по компьютерной сети от одной системы к другой.

Первоначально "черви" были разработаны для поиска в сети других компьютеров со свободными ресурсами, чтобы получить возможность выполнить распределенные вычисления. При правильном использовании технология "червей" может быть весьма полезной. Например, "червь" World Wide Web Worm формирует индекс поиска участков Web. Однако "червь" легко превращается во вредоносную программу. "Червь" использует механизмы поддержки сети для определения узла, который может быть поражен. Затем с помощью этих же механизмов передает свое тело в этот узел и либо активизируется, либо ждет подходящих условий для активизации.

Наиболее известным представителем этого класса вредоносных программ является "червь" Морриса, который представлял собой программу из 4000 строк на языке Си и входном языке командного интерпретатора системы UNIX. Студент Корнеллского Университета Роберт Таппан Моррис-младший запустил 2 ноября 1988 г. на компьютере Массачусетсского Технологического Института свою программу-червь. Используя ошибки в операционной системе UNIX на компьютерах VAX и Sun, эта программа передавала свой код с машины на машину. Всего за 6 часов были поражены подключенные к сети Internet 6000 компьютеров ряда крупных университетов, институтов и исследовательских лабораторий США. Ущерб от этой акции составил многие миллионы долларов.

Сетевые "черви" являются самым опасным видом вредоносных программ, так как объектом их атаки может стать любой из миллионов компьютеров, подключенных к глобальной сети Internet. Для защиты от "червя" необходимо принять меры предосторожности против несанкционированного доступа к внутренней сети. Следует отметить, что "троянские кони", компьютерные вирусы и сетевые "черви" относятся к наиболее опасным угрозам АСОИ. Для защиты от указанных вредоносных программ необходимо применение ряда мер:

- исключение несанкционированного доступа к исполняемым файлам;
- тестирование приобретаемых программных средств;
- контроль целостности исполняемых файлов и системных областей;
- создание замкнутой среды исполнения программ.

1.3. Обеспечение безопасности АСОИ

Основным назначением АСОИ является переработка (сбор, хранение, обработка и выдача) информации, поэтому проблема обеспечения информационной безопасности является для АСОИ центральной. Обеспечение безопасности АСОИ предполагает организацию противодействия любому несанкционированному вторжению в процесс функционирования АСОИ, а также попыткам модификации, хищения, выведения из строя или разрушения ее компонентов, т.е. защиту всех компонентов АСОИ – аппаратных средств, программного обеспечения, данных и персонала [22].

Существуют два подхода к проблеме обеспечения безопасности АСОИ: "фрагментарный" и комплексный.

"Фрагментарный" подход направлен на противодействие четко определенным угрозам в заданных условиях. В качестве примеров реализации такого подхода можно указать отдельные средства управления доступом, автономные средства шифрования, специализированные антивирусные программы и т.п.

Достоинством такого подхода является высокая избирательность к конкретной угрозе. Существенным недостатком данного подхода является отсутствие единой защищенной среды обработки информации. Фрагментарные меры защиты информации обеспечивают защиту конкретных объектов АСОИ только от конкретной угрозы. Даже небольшое видоизменение угрозы ведет к потере эффективности защиты.

Комплексный подход ориентирован на создание защищенной среды обработки информации в АСОИ, объединяющей в единый комплекс разнородные меры противодействия угрозам. Организация защищенной среды обработки информации позволяет гарантировать определенный уровень безопасности АСОИ, что является несомненным достоинством комплексного подхода. К недостаткам этого подхода относятся: ограничения на свободу действий пользователей АСОИ, большая чувствительность к ошибкам установки и настройки средств защиты, сложность управления.

Комплексный подход применяют для защиты АСОИ крупных организаций или небольших АСОИ, выполняющих ответственные задачи или обрабатывающих особо важную информацию. Нарушение безопасности информации в АСОИ крупных организаций может нанести огромный материальный ущерб как самим организациям, так и их клиентам. Поэтому такие организации вынуждены уделять особое внимание гарантиям безопасности и реализовывать комплексную защиту. Комплексного подхода придерживаются большинство государственных и крупных коммерческих предприятий и учреждений. Этот подход нашел свое отражение в различных стандартах.

Комплексный подход к проблеме обеспечения безопасности основан на разработанной для конкретной АСОИ политике безопасности. *Политика безопасности* представляет собой набор норм, правил и практических рекомендаций, на которых строится управление, защита и распределение информации в АСОИ. Политика безопасности регламентирует эффективную работу средств защиты АСОИ. Она охватывает все особенности процесса обработки информации, определяя поведение системы в различных ситуациях.

Политика безопасности реализуется посредством административно-организационных мер, физических и программно-технических средств и определяет архитектуру системы защиты. Для конкретной организации политика безопасности должна носить индивидуальный характер и зависеть от конкретной технологии обработки информации и используемых программных и технических средств.

Политика безопасности определяется способом управления доступом, определяющим порядок доступа к объектам системы. Различают два основных вида политики безопасности: избирательную и полномочную.

Избирательная политика безопасности основана на избирательном способе управления доступом. *Избирательное (или дискреционное) управление доступом* характеризуется заданным администратором множеством разрешенных отношений доступа (например, в виде троек <объект, субъект, тип доступа>). Обычно для описания свойств избирательного управления доступом применяют математическую модель на основе матрицы доступа.

Матрица доступа представляет собой матрицу, в которой столбец соответствует объекту системы, а строка – субъекту. На пересечении столбца и строки матрицы указывается тип разрешенного доступа субъекта к объекту. Обычно выделяют такие типы доступа субъекта к объекту, как "доступ на чтение", "доступ на запись", "доступ на исполнение" и т.п. Матрица доступа является самым простым подходом к моделированию систем управления доступом. Однако она является основой для более сложных моделей, более адекватно описывающих реальные АСОИ.

Избирательная политика безопасности широко применяется в АСОИ коммерческого сектора, так как ее реализация соответствует требованиям коммерческих организаций по разграничению доступа и подотчетности, а также имеет приемлемую стоимость.

Полномочная политика безопасности основана на полномочном (мандатном) способе управления доступом. *Полномочное (или мандатное) управление доступом* характеризуется совокупностью правил предоставления доступа, определенных на множестве атрибутов безопасности субъектов и объектов, например, в зависи-

мости от метки конфиденциальности информации и уровня допуска пользователя. Полномочное управление доступом подразумевает, что:

- все субъекты и объекты системы однозначно идентифицированы;
- каждому объекту системы присвоена метка конфиденциальности информации, определяющая ценность содержащейся в нем информации;
- каждому субъекту системы присвоен определенный уровень допуска, определяющий максимальное значение метки конфиденциальности информации объектов, к которым субъект имеет доступ.

Чем важнее объект, тем выше его метка конфиденциальности. Поэтому наиболее защищенными оказываются объекты с наиболее высокими значениями метки конфиденциальности.

Основным назначением полномочной политики безопасности является регулирование доступа субъектов системы к объектам с различными уровнями конфиденциальности, предотвращение утечки информации с верхних уровней должностной иерархии на нижние, а также блокирование возможных проникновений с нижних уровней на верхние.

Помимо управления доступом субъектов к объектам системы проблема защиты информации имеет еще один аспект. Для получения информации о каком-либо объекте системы совсем необязательно искать пути несанкционированного доступа к нему. Необходимую информацию можно получить, наблюдая за обработкой требуемого объекта, т.е. используя каналы утечки информации. В системе всегда существуют информационные потоки. Поэтому администратору необходимо определить, какие информационные потоки в системе являются "легальными", т.е. не ведут к утечке информации, а какие – ведут к утечке. Поэтому возникает необходимость разработки правил, регламентирующих *управление информационными потоками* в системе. Обычно управление информационными потоками применяется в рамках избирательной или полномочной политики, дополняя их и способствуя повышению надежности системы защиты.

Избирательное и полномочное управление доступом, а также управление информационными потоками являются тем фундаментом, на котором строится вся система защиты.

Под *системой защиты* АСОИ понимают единую совокупность правовых и морально-этических норм, административно-организационных мер, физических и программно-технических средств, направленных на противодействие угрозам АСОИ с целью сведения к минимуму возможности ущерба.

Процесс построения системы защиты включает следующие этапы:

- анализ возможных угроз АСОИ;
- планирование системы защиты;
- реализация системы защиты;
- сопровождение системы защиты.

Этап анализа возможных угроз АСОИ необходим для фиксации состояния АСОИ (конфигурации аппаратных и программных средств, технологии обработки информации) и определения учитываемых воздействий на компоненты системы. Практически невозможно обеспечить защиту АСОИ от всех воздействий, поскольку невозможно полностью установить все угрозы и способы их реализаций. Поэтому из всего множества вероятных воздействий выбирают только такие воздействия, которые могут реально произойти и нанести серьезный ущерб.

На *этапе планирования* формулируется система защиты как единая совокупность мер противодействия угрозам различной природы [22].

По способам осуществления все меры обеспечения безопасности компьютерных систем подразделяют на:
- правовые (законодательные);
- морально-этические;
- административные;
- физические;
- аппаратно-программные.

Перечисленные меры безопасности АСОИ можно рассматривать как последовательность барьеров или рубежей защиты информации. Для того чтобы добраться до защищаемой информации, нужно последовательно преодолеть несколько рубежей защиты. Рассмотрим их подробнее.

Первый рубеж защиты, встающий на пути человека, пытающегося осуществить НСД к информации, является чисто правовым. Этот аспект защиты информации связан с необходимостью соблюдения юридических норм при передаче и обработке информации. К *правовым мерам* защиты информации относятся действующие в стране законы, указы и другие нормативные акты, регламентирующие правила обращения с информацией ограниченного использования и ответственности за их нарушения. Этим они препятствуют несанкционированному использованию информации и являются сдерживающим фактором для потенциальных нарушителей.

Второй рубеж защиты образуют *морально-этические меры*. Этический момент в соблюдении требований защиты имеет весьма большое значение. Очень важно, чтобы люди, имеющие доступ к компьютерам, работали в здоровом морально-этическом климате.

К морально-этическим мерам противодействия относятся всевозможные нормы поведения, которые традиционно сложились или складываются в обществе по мере распространения компьютеров в стране. Эти нормы большей частью не являются обязательными, как законодательно утвержденные, но их несоблюдение обычно ведет к падению престижа человека, группы лиц или организации. Морально-этические нормы бывают как неписаными (например, общепризнанные нормы честности, патриотизма и т.д.), так и оформленными в некий свод правил или предписаний. Например, "Кодекс профессионального поведения членов Ассоциации пользователей ЭВМ США" рассматривает как неэтичные действия, которые умышленно или неумышленно:

- нарушают нормальную работу компьютерных систем;
- вызывают неоправданные затраты ресурсов (машинного времени, памяти, каналов связи и т.п.);
- нарушают целостность информации (хранимой и обрабатываемой);
- нарушают интересы других законных пользователей и т.п.

Третьим рубежом, препятствующим неправомочному использованию информации, являются административные меры. Администраторы всех рангов с учетом правовых норм и социальных аспектов определяют административные меры защиты информации.

Административные меры защиты относятся к мерам организационного характера. Они регламентируют:

- процессы функционирования АСОИ;
- использование ресурсов АСОИ;
- деятельность ее персонала;
- порядок взаимодействия пользователей с системой, с тем чтобы в наибольшей степени затруднить или исключить возможность реализации угроз безопасности.

Административные меры включают:

- разработку правил обработки информации в АСОИ;
- совокупность действий при проектировании и оборудовании вычислительных центров и других объектов АСОИ (учет влияния стихии, пожаров, охрана помещений и т.п.);
- совокупность действий при подборе и подготовке персонала (проверка новых сотрудников, ознакомление их с порядком работы с конфиденциальной информацией, с мерами ответственности за нарушение правил ее обработки; создание условий, при которых персоналу было бы невыгодно допускать злоупотребления и т.д.);
- организацию надежного пропускного режима;
- организацию учета, хранения, использования и уничтожения документов и носителей с конфиденциальной информацией;

- распределение реквизитов разграничения доступа (паролей, полномочий и т. п.);
- организацию скрытого контроля за работой пользователей и персонала АСОИ;
- совокупность действий при проектировании, разработке, ремонте и модификации оборудования и программного обеспечения (сертификация используемых технических и программных средств, строгое санкционирование, рассмотрение и утверждение всех изменений, проверка на удовлетворение требованиям защиты, документальная фиксация изменений и т. п.).

Важно отметить, что, пока не будут реализованы действенные меры административной защиты ЭВМ, прочие меры будут, несомненно, неэффективны. Административно-организационные меры защиты могут показаться скучными и рутинными по сравнению с морально-этическими и лишенными конкретности по сравнению с аппаратно-программными. Однако они представляют собой мощный барьер на пути незаконного использования информации и надежную базу для других уровней защиты.

Четвертым рубежом являются *физические меры защиты*. К физическим мерам защиты относятся разного рода механические, электро- и электронно-механические устройства или сооружения, специально предназначенные для создания физических препятствий на возможных путях проникновения и доступа потенциальных нарушителей к компонентам системы и защищаемой информации.

Пятым рубежом являются *аппаратно-программные средства защиты*. К ним относятся различные электронные устройства и специальные программы, которые реализуют самостоятельно или в комплексе с другими средствами следующие способы защиты:

- идентификацию (распознавание) и аутентификацию (проверка подлинности) субъектов (пользователей, процессов) АСОИ;
- разграничение доступа к ресурсам АСОИ;
- контроль целостности данных;
- обеспечение конфиденциальности данных;
- регистрацию и анализ событий, происходящих в АСОИ;
- резервирование ресурсов и компонентов АСОИ.

Большинство из перечисленных способов защиты реализуется криптографическими методами защиты информации.

При проектировании эффективной системы защиты следует учитывать ряд принципов, отображающих основные положения по безопасности информации [39]. К числу этих принципов относятся следующие:

- Экономическая эффективность. Стоимость средств защиты должна быть меньше, чем размеры возможного ущерба.

- Минимум привилегий. Каждый пользователь должен иметь минимальный набор привилегий, необходимый для работы.
- Простота. Защита тем более эффективна, чем легче пользователю с ней работать.
- Отключаемость защиты. При нормальном функционировании защита не должна отключаться. Только в особых случаях сотрудник со специальными полномочиями может отключить систему защиты.
- Открытость проектирования и функционирования механизмов защиты. Специалисты, имеющие отношение к системе защиты, должны полностью представлять себе принципы ее функционирования и в случае возникновения затруднительных ситуаций адекватно на них реагировать.
- Всеобщий контроль. Любые исключения из множества контролируемых субъектов и объектов защиты снижают защищенность автоматизированного комплекса обработки информации.
- Независимость системы защиты от субъектов защиты. Лица, занимавшиеся разработкой системы защиты, не должны быть в числе тех, кого эта система будет контролировать.
- Отчетность и подконтрольность. Система защиты должна предоставлять доказательства корректности своей работы.
- Ответственность. Подразумевается личная ответственность лиц, занимающихся обеспечением безопасности информации.
- Изоляция и разделение. Объекты защиты целесообразно разделять на группы таким образом, чтобы нарушение защиты в одной из групп не влияло на безопасность других групп.
- Полнота и согласованность. Надежная система защиты должна быть полностью специфицирована, протестирована и согласована.
- Параметризация. Защита становится более эффективной и гибкой, если она допускает изменение своих параметров со стороны администратора.
- Принцип враждебного окружения. Система защиты должна проектироваться в расчете на враждебное окружение. Разработчики должны исходить из предположения, что пользователи имеют наихудшие намерения, что они будут совершать серьезные ошибки и искать пути обхода механизмов защиты.
- Привлечение человека. Наиболее важные и критические решения должны приниматься человеком.
- Отсутствие излишней информации о существовании механизмов защиты. Существование механизмов защиты должно быть по возможности скрыто от пользователей, работа которых должна контролироваться.

Результатом *этапа планирования* является развернутый план защиты АСОИ, содержащий перечень защищаемых компонентов АСОИ и возможных воздействий на них, цель защиты информации в АСОИ, правила обработки информации в АСОИ, обеспечивающие ее защиту от различных воздействий, а также описание планируемой системы защиты информации.

Сущность *этапа реализации* системы защиты заключается в установке и настройке средств защиты, необходимых для реализации запланированных правил обработки информации.

Заключительный *этап сопровождения* заключается в контроле работы системы, регистрации происходящих в ней событий, их анализе с целью обнаружения нарушений безопасности, коррекции системы защиты.

1.4. Принципы криптографической защиты информации

Криптография представляет собой совокупность методов преобразования данных, направленных на то, чтобы сделать эти данные бесполезными для противника. Такие преобразования позволяют решить две главные проблемы защиты данных: *проблему конфиденциальности* (путем лишения противника возможности извлечь информацию из канала связи) и *проблему целостности* (путем лишения противника возможности изменить сообщение так, чтобы изменился его смысл, или ввести ложную информацию в канал связи).

Проблемы конфиденциальности и целостности информации тесно связаны между собой, поэтому методы решения одной из них часто применимы для решения другой.

Обобщенная схема криптографической системы, обеспечивающей шифрование передаваемой информации, показана на рис.1.1. *Отправитель* генерирует *открытый текст* исходного сообщения M, которое должно быть передано законному *получателю* по незащищенному каналу. За каналом следит *перехватчик* с целью перехватить и раскрыть передаваемое сообщение. Для того чтобы перехватчик не смог узнать содержание сообщения M, отправитель шифрует его с помощью обратимого преобразования E_K и получает *шифртекст* (или *криптограмму*) $C=E_K(M)$, который отправляет получателю.

Законный получатель, приняв шифртекст C, расшифровывает его с помощью обратного преобразования $D=E_K^{-1}$ и получает исходное сообщение в виде открытого текста M:

$$D_K(C) = E_K^{-1}(E_K(M)) = M.$$

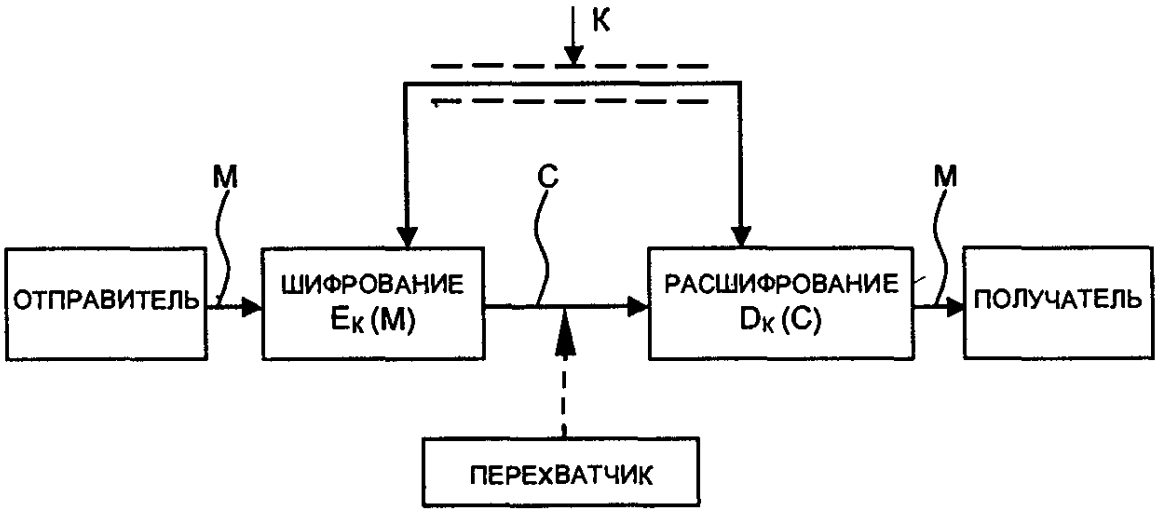


Рис.1.1. Обобщенная схема криптосистемы

Преобразование E_K выбирается из семейства криптографических преобразований, называемых криптоалгоритмами. Параметр, с помощью которого выбирается отдельное используемое преобразование, называется криптографическим ключом К. Криптосистема имеет разные варианты реализации: набор инструкций, аппаратные средства, комплекс программ компьютера, которые позволяют зашифровать открытый текст и расшифровать шифртекст различными способами, один из которых выбирается с помощью конкретного ключа К.

Говоря более формально, криптографическая система – это однопараметрическое семейство $(E_K)_{K\in\overline{K}}$ обратимых преобразований

$$E_K : \overline{M} \to \overline{C}$$

из пространства $\overline{M}$ сообщений открытого текста в пространство $\overline{C}$ шифрованных текстов. Параметр К (ключ) выбирается из конечного множества $\overline{K}$, называемого *пространством ключей.*

Вообще говоря, преобразование шифрования может быть симметричным или асимметричным относительно преобразования расшифрования. Это важное свойство функции преобразования определяет два класса криптосистем:

- симметричные (одноключевые) криптосистемы;
- асимметричные (двухключевые) криптосистемы (с открытым ключом).

Схема симметричной криптосистемы с одним секретным ключом была показана на рис.1.1. В ней используются одинаковые секретные ключи в блоке шифрования и блоке расшифрования.

Обобщенная схема асимметричной криптосистемы с двумя разными ключами K_1 и K_2 показана на рис.1.2. В этой криптосистеме один из ключей является открытым,а другой – секретным.

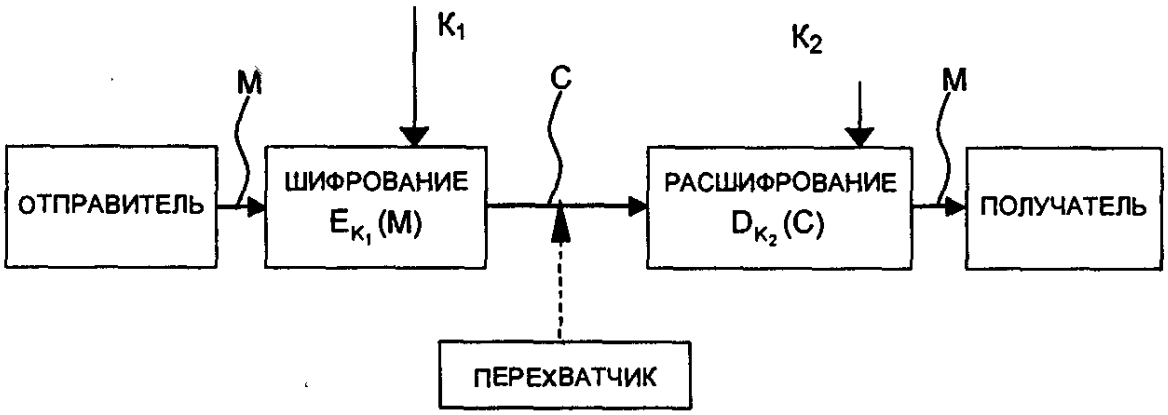


Рис.1.2. Обобщенная схема асимметричной криптосистемы с открытым ключом

В симметричной криптосистеме секретный ключ надо передавать отправителю и получателю по защищенному каналу распространения ключей, например такому, как курьерская служба. На рис.1.1 этот канал показан "экранированной" линией. Существуют и другие способы распределения секретных ключей, они будут рассмотрены позднее. В асимметричной криптосистеме передают по незащищенному каналу только открытый ключ, а секретный ключ сохраняют на месте его генерации.

На рис.1.3 показан поток информации в криптосистеме в случае активных действий перехватчика. Активный перехватчик не только считывает все шифртексты, передаваемые по каналу, но может также пытаться изменять их по своему усмотрению.

Любая попытка со стороны перехватчика расшифровать шифртекст C для получения открытого текста M или зашифровать свой собственный текст M′ для получения правдоподобного шифртекста C′, не имея подлинного ключа, называется *криптоаналитической атакой*.

Если предпринятые криптоаналитические атаки не достигают поставленной цели и криптоаналитик не может, не имея подлинного ключа, вывести M из C или C′ из M′, то полагают, что такая криптосистема является *криптостойкой*.

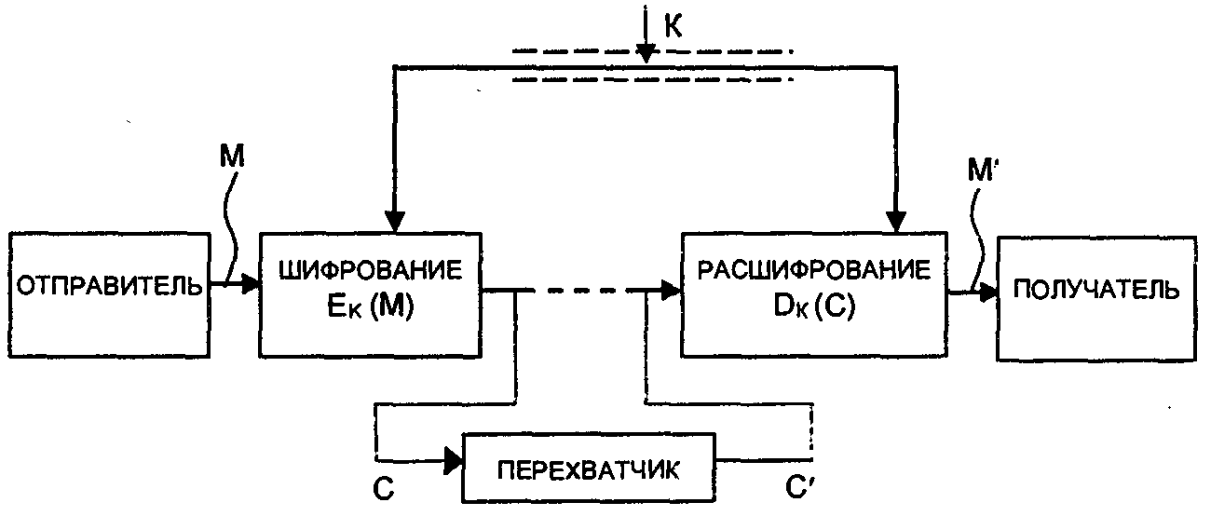


Рис.1.3. Поток информации в криптосистеме при активном перехвате сообщений

Криптоанализ – это наука о раскрытии исходного текста зашифрованного сообщения без доступа к ключу. Успешный анализ может раскрыть исходный текст или ключ. Он позволяет также обнаружить слабые места в криптосистеме, что, в конечном счете, ведет к тем же результатам.

Фундаментальное правило криптоанализа, впервые сформулированное голландцем А. Керкхоффом еще в XIX веке заключается в том, что стойкость шифра (криптосистемы) должна определяться только секретностью ключа. Иными словами, правило Керкхоффа состоит в том, что весь алгоритм шифрования, кроме значения секретного ключа, известен криптоаналитику противника. Это обусловлено тем, что криптосистема, реализующая семейство криптографических преобразований, обычно рассматривается как открытая система. Такой подход отражает очень важный принцип технологии защиты информации: защищенность системы не должна зависеть от секретности чего-либо такого, что невозможно быстро изменить в случае утечки секретной информации. Обычно криптосистема представляет собой совокупность аппаратных и программных средств, которую можно изменить только при значительных затратах времени и средств, тогда как ключ является легко изменяемым объектом. Именно поэтому стойкость криптосистемы определяется только секретностью ключа.

Другое почти общепринятое допущение в криптоанализе состоит в том, что криптоаналитик имеет в своем распоряжении шифртексты сообщений.

Существует четыре основных типа криптоаналитических атак. Конечно, все они формулируются в предположении, что криптоаналитику известны применяемый алгоритм шифрования и шифртексты сообщений. Перечислим эти криптоаналитические атаки.

1. Криптоаналитическая атака при наличии только известного шифртекста. Криптоаналитик имеет только шифртексты C_1, C_2, ..., C_i нескольких сообщений, причем все они зашифрованы с использованием одного и того же алгоритма шифрования E_K. Работа криптоаналитика заключается в том, чтобы раскрыть исходные тексты M_1, M_2, ..., M_i по возможности большинства сообщений или, еще лучше, вычислить ключ K, использованный для шифрования этих сообщений, с тем, чтобы расшифровать и другие сообщения, зашифрованные этим ключом.

2. Криптоаналитическая атака при наличии известного открытого текста. Криптоаналитик имеет доступ не только к шифртекстам C_1, C_2, ..., C_i нескольких сообщений, но также к открытым текстам M_1, M_2, ..., M_i этих сообщений. Его работа заключается в нахождении ключа K, используемого при шифровании этих сообщений, или алгоритма расшифрования D_K любых новых сообщений, зашифрованных тем же самым ключом.

3. Криптоаналитическая атака при возможности выбора открытого текста. Криптоаналитик не только имеет доступ к шифртекстам $C_1, C_2, \ldots, C_i$ и связанным с ними открытым текстам $M_1, M_2, \ldots, M_i$ нескольких сообщений, но и может по желанию выбирать открытые тексты, которые затем получает в зашифрованном виде. Такой криптоанализ получается более мощным по сравнению с криптоанализом с известным открытым текстом, потому что криптоаналитик может выбрать для шифрования такие блоки открытого текста, которые дадут больше информации о ключе. Работа криптоаналитика состоит в поиске ключа К, использованного для шифрования сообщений, или алгоритма расшифрования D_K новых сообщений, зашифрованных тем же ключом.

4. Криптоаналитическая атака с адаптивным выбором открытого текста. Это – особый вариант атаки с выбором открытого текста. Криптоаналитик может не только выбирать открытый текст, который затем шифруется, но и изменять свой выбор в зависимости от результатов предыдущего шифрования. При криптоанализе с простым выбором открытого текста криптоаналитик обычно может выбирать несколько крупных блоков открытого текста для их шифрования; при криптоанализе с адаптивным выбором открытого текста он имеет возможность выбрать сначала более мелкий пробный блок открытого текста, затем выбрать следующий блок в зависимости от результатов первого выбора, и т.д. Эта атака предоставляет криптоаналитику еще больше возможностей, чем предыдущие типы атак.

Кроме перечисленных основных типов криптоаналитических атак, можно отметить, по крайней мере, еще два типа.

5. Криптоаналитическая атака с использованием выбранного шифртекста. Криптоаналитик может выбирать для расшифрования различные шифртексты $C_1, C_2, \ldots, C_i$ и имеет доступ к расшифрованным открытым текстам $M_1, M_2, \ldots, M_i$. Например, криптоаналитик получил доступ к защищенному от несанкционированного вскрытия блоку, который выполняет автоматическое расшифрование. Работа криптоаналитика заключается в нахождении ключа. Этот тип криптоанализа представляет особый интерес для раскрытия алгоритмов с открытым ключом.

6. Криптоаналитическая атака методом полного перебора всех возможных ключей. Эта атака предполагает использование криптоаналитиком известного шифртекста и осуществляется посредством полного перебора всех возможных ключей с проверкой, является ли осмысленным получающийся открытый текст. Такой подход требует привлечения предельных вычислительных ресурсов и иногда называется силовой атакой.

Существуют и другие, менее распространенные, криптоаналитические атаки, некоторые из них будут описаны в соответствующих разделах книги.

1.5. Аппаратно-программные средства защиты компьютерной информации

Первые операционные системы (ОС) для персональных компьютеров (MS-DOS и Windows версий до 3.1 включительно) не имели собственных средств защиты, что и породило проблему создания дополнительных средств защиты. Актуальность этой проблемы практически не уменьшилась с появлением более мощных ОС с развитыми подсистемами защиты, например Windows NT и Windows 2000. Это обусловлено тем, что большинство систем не способны защитить данные, находящиеся за ее пределами, например при использовании сетевого информационного обмена [73].

Аппаратно-программные средства, обеспечивающие повышенный уровень защиты, можно разбить на пять основных групп (рис.1.4).

Первую группу образуют *системы идентификации и аутентификации пользователей.* Такие системы применяются для ограничения доступа случайных и незаконных пользователей к ресурсам компьютерной системы. Общий алгоритм работы этих систем заключается в том, чтобы получить от пользователя информацию, удостоверяющую его личность, проверить ее подлинность и затем предоставить (или не предоставить) этому пользователю возможность работы с системой.

При построении подобных систем возникает проблема выбора информации, на основе которой осуществляются процедуры идентификации и аутентификации пользователя. Можно выделить следующие типы:

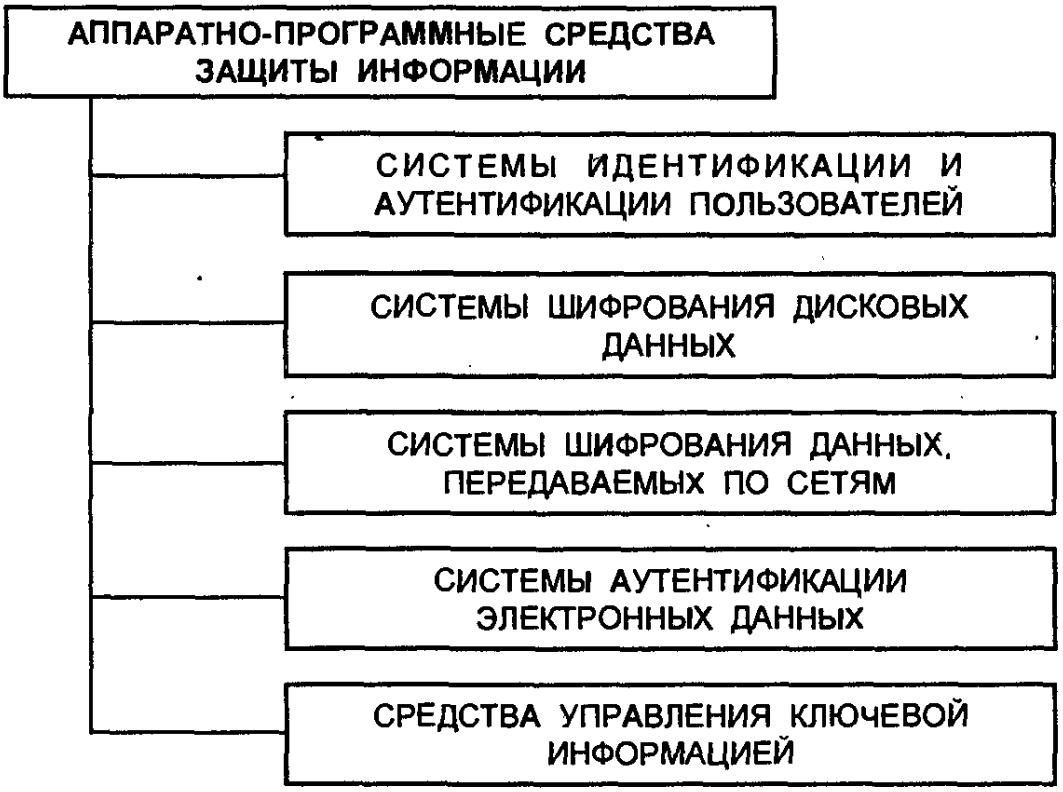


Рис.1.4. Аппаратно-программные средства защиты компьютерной информации

1) секретная информация, которой обладает пользователь (пароль, персональный идентификатор, секретный ключ и т.п.); эту информацию пользователь должен запомнить или же могут быть применены специальные средства хранения этой информации;

2) физиологические параметры человека (отпечатки пальцев, рисунок радужной оболочки глаза и т.п.) или особенности поведения человека (особенности работы на клавиатуре и т.п.).

Системы идентификации, основанные на первом типе информации, принято считать *традиционными*. Системы идентификации, использующие второй тип информации, называются *биометрическими*. Следует отметить наметившуюся тенденцию опережающего развития биометрических систем идентификации [117].

Процедуры идентификации и аутентификации пользователей подробно рассматриваются в гл. 5.

Вторую группу средств, обеспечивающих повышенный уровень защиты, составляют *системы шифрования дисковых данных*. Основная задача, решаемая такими системами, состоит в защите от несанкционированного использования данных, расположенных на магнитных носителях.

Обеспечение конфиденциальности данных, располагаемых на магнитных носителях, осуществляется путем их шифрования с использованием симметричных алгоритмов шифрования. Основным классификационным признаком для комплексов шифрования служит уровень их встраивания в компьютерную систему [73].

Работа прикладных программ с дисковыми накопителями состоит из двух этапов –"логического" и "физического".

Логический этап соответствует уровню взаимодействия прикладной программы с операционной системой (например, вызов сервисных функций чтения/записи данных). На этом уровне основным объектом является файл.

Физический этап соответствует уровню взаимодействия операционной системы и аппаратуры. В качестве объектов этого уровня выступают структуры физической организации данных – сектора диска.

В результате системы шифрования данных могут осуществлять криптографические преобразования данных на уровне файлов (защищаются отдельные файлы) и на уровне дисков (защищаются диски целиком).

К программам первого типа можно отнести архиваторы типа arj, которые позволяют использовать криптографические методы для защиты архивных файлов. Примером систем второго типа может служить программа шифрования Diskreet, входящая в состав популярного программного пакета Norton Utilities.

Другим классификационным признаком систем шифрования дисковых данных является способ их функционирования. По способу функционирования системы шифрования дисковых данных делят на два класса:

1) системы "прозрачного" шифрования;

2) системы, специально вызываемые для осуществления шифрования.

В системах *прозрачного шифрования* (шифрования "на лету") криптографические преобразования осуществляются в режиме реального времени, незаметно для пользователя. Например, пользователь записывает подготовленный в текстовом редакторе документ на защищаемый диск, а система защиты в процессе записи выполняет его шифрование.

Системы второго класса обычно представляют собой утилиты, которые необходимо специально вызывать для выполнения шифрования. К ним относятся, например, архиваторы со встроенными средствами парольной защиты.

Алгоритмы шифрования и системы шифрования дисковых данных подробно рассматриваются в гл. 3 и 10.

К третьей группе средств, обеспечивающих повышенный уровень защиты, относятся *системы шифрования данных, передаваемых по компьютерным сетям*. Различают два основных способа шифрования: канальное шифрование и оконечное (абонентское) шифрование.

В случае *канального шифрования* защищается вся передаваемая по каналу связи информация, включая служебную. Соответствующие процедуры шифрования реализуются с помощью протокола канального уровня семиуровневой эталонной модели взаимодействия открытых систем OSI (Open System Interconnection). Этот способ шифрования обладает следующим достоинством – встраивание процедур шифрования на канальный уровень позволяет использовать аппаратные средства, что способствует повышению производительности системы.

Однако, у данного подхода имеются существенные недостатки:

- шифрованию на данном уровне подлежит вся информация, включая служебные данные транспортных протоколов; это осложняет механизм маршрутизации сетевых пакетов и требует расшифрования данных в устройствах промежуточной коммутации (шлюзах, ретрансляторах и т. п.);
- шифрование служебной информации, неизбежное на данном уровне, может привести к появлению статистических закономерностей в шифрованных данных; это влияет на надежность защиты и накладывает ограничения на использование криптографических алгоритмов.

Оконечное (абонентское) шифрование позволяет обеспечить конфиденциальность данных, передаваемых между двумя прикладными объектами (абонентами). Оконечное шифрование реализуется с помощью протокола прикладного или представительного уровня эталонной модели OSI. В этом случае защищенным оказывается только содержание сообщения, вся служебная информация остается открытой. Данный способ позволяет избежать проблем, связанных с шифрованием служебной информации, но при этом возникают другие проблемы. В частности, злоумышленник, имеющий доступ к каналам связи компьютерной сети, получает возможность анализировать информацию о структуре обмена сообщениями, например об отправителе и получателе, о времени и условиях передачи данных, а также об объеме передаваемых данных. Системы шифрования данных, передаваемых по компьютерным сетям, рассматриваются в гл. 3, 4 и 11.

Четвертую группу средств защиты составляют *системы аутентификации электронных данных*. При обмене электронными данными по сетям связи возникает проблема аутентификации автора документа и самого документа, т.е. установление подлинности автора и проверка отсутствия изменений в полученном документе. Для аутентификации электронных данных применяют код аутентификации сообщения (имитовставку) или электронную цифровую подпись. При формировании кода аутентификации сообщения и электронной цифровой подписи используются разные типы систем шифрования.

Код аутентификации сообщения формируют с помощью симметричных систем шифрования данных. В частности, симметричный алгоритм шифрования данных DES при работе в режиме сцепления блоков шифра CBC позволяет сформировать с помощью секретного ключа и начального вектора IV код аутентификации сообщения MAC (Message Authentication Code). Проверка целостности принятого сообщения осуществляется путем проверки кода MAC получателем сообщения.

Аналогичные возможности предоставляет отечественный стандарт симметричного шифрования данных ГОСТ 28147-89. В этом алгоритме предусмотрен режим выработки имитовставки, обеспечивающий *имитозащиту*, т.е. защиту системы шифрованной связи от навязывания ложных данных.

Имитовставка вырабатывается из открытых данных посредством специального преобразования шифрования с использованием секретного ключа и передается по каналу связи в конце зашифрованных данных. Имитовставка проверяется получателем сообщения, владеющим секретным ключом, путем повторения процедуры, выполненной ранее отправителем, над полученными открытыми данными.

Алгоритмы формирования кода аутентификации сообщения MAC и имитовставки подробно рассмотрены в гл. 3.

Электронная цифровая подпись (ЭЦП) представляет собой относительно небольшое количество дополнительной аутентифицирующей цифровой информации, передаваемой вместе с подписываемым текстом. Для реализации ЭЦП используются принципы асимметричного шифрования. Система ЭЦП включает процедуру формирования цифровой подписи отправителем с использованием секретного ключа отправителя и процедуру проверки подписи получателем с использованием открытого ключа отправителя. Алгоритмы и системы электронной цифровой подписи подробно разбираются в гл. 6 и 11.

Пятую группу средств, обеспечивающих повышенный уровень защиты, *образуют средства управления ключевой информацией.* Под ключевой информацией понимается совокупность всех используемых в компьютерной системе или сети криптографических ключей. Безопасность любого криптографического алгоритма определяется используемыми криптографическими ключами. В случае ненадежного управления ключами злоумышленник может завладеть ключевой информацией и получить полный доступ ко всей информации в компьютерной системе или сети.

Основным классификационным признаком средств управления ключевой информацией является вид функции управления ключами. Различают следующие основные виды функций управления ключами: генерация ключей, хранение ключей и распределение ключей.

Способы *генерации ключей* различаются для симметричных и асимметричных криптосистем. Для генерации ключей симметричных криптосистем используются аппаратные и программные средства генерации случайных чисел, в частности схемы с применением блочного симметричного алгоритма шифрования. Генерация ключей для асимметричных криптосистем представляет существенно более сложную задачу в связи с необходимостью получения ключей с определенными математическими свойствами.

Функция *хранения ключей* предполагает организацию безопасного хранения, учета и удаления ключей. Для обеспечения безопасного хранения и передачи ключей применяют их шифрование с помощью других ключей. Такой подход приводит к *концепции иерархии ключей.* В иерархию ключей обычно входят главный ключ (мастер-ключ), ключ шифрования ключей и ключ шифрования данных. Следует отметить, что генерация и хранение мастер-ключей являются критическими вопросами криптографической защиты.

Распределение ключей является самым ответственным процессом в управлении ключами. Этот процесс должен гарантировать скрытность распределяемых ключей, а также оперативность и точность их распределения. Различают два основных способа распределения ключей между пользователями компьютерной сети:

- применение одного или нескольких центров распределения ключей;
- прямой обмен сеансовыми ключами между пользователями.

Аппаратно-программные средства управления криптографическими ключами подробно обсуждаются в гл.7 и 10.

ГЛАВА 2. ТРАДИЦИОННЫЕ СИММЕТРИЧНЫЕ КРИПТОСИСТЕМЫ

2.1. Основные понятия и определения

Большинство средств защиты информации базируется на использовании криптографических шифров и процедур шифрования-расшифрования. В соответствии со стандартом ГОСТ 28147-89 под шифром понимают совокупность обратимых преобразований множества открытых данных на множество зашифрованных данных, задаваемых ключом и алгоритмом криптографического преобразования.

Ключ – это конкретное секретное состояние некоторых параметров алгоритма криптографического преобразования данных, обеспечивающее выбор только одного варианта из всех возможных для данного алгоритма [81].

Основной характеристикой шифра является *криптостойкость*, которая определяет его стойкость к раскрытию методами криптоанализа. Обычно эта характеристика определяется интервалом времени, необходимым для раскрытия шифра.

К шифрам, используемым для криптографической защиты информации, предъявляется ряд требований:

- достаточная криптостойкость (надежность закрытия данных);
- простота процедур шифрования и расшифрования;
- незначительная избыточность информации за счет шифрования;
- нечувствительность к небольшим ошибкам шифрования и др.

В той или иной мере этим требованиям отвечают:

- шифры перестановок;
- шифры замены;
- шифры гаммирования;
- шифры, основанные на аналитических преобразованиях шифруемых данных.

Шифрование перестановкой заключается в том, что символы шифруемого текста переставляются по определенному правилу в пределах некоторого блока этого текста. При достаточной длине

блока, в пределах которого осуществляется перестановка, и сложном неповторяющемся порядке перестановки можно достигнуть приемлемой для простых практических приложений стойкости шифра.

Шифрование заменой (подстановкой) заключается в том, что символы шифруемого текста заменяются символами того же или другого алфавита в соответствии с заранее обусловленной схемой замены.

Шифрование гаммированием заключается в том, что символы шифруемого текста складываются с символами некоторой случайной последовательности, именуемой *гаммой шифра*. Стойкость шифрования определяется в основном длиной (периодом) неповторяющейся части гаммы шифра. Поскольку с помощью ЭВМ можно генерировать практически бесконечную гамму шифра, то данный способ является одним из основных для шифрования информации в автоматизированных системах.

Шифрование аналитическим преобразованием заключается в том, что шифруемый текст преобразуется по некоторому аналитическому правилу (формуле).

Например, можно использовать правило умножения вектора на матрицу, причем умножаемая матрица является ключом шифрования (поэтому ее размер и содержание должны храниться в секрете), а символами умножаемого вектора последовательно служат символы шифруемого текста. Другим примером может служить использование так называемых однонаправленных функций для построения криптосистем с открытым ключом (см. гл. 5).

Процессы шифрования и расшифрования осуществляются в рамках некоторой криптосистемы. Характерной особенностью симметричной криптосистемы является применение одного и того же секретного ключа как при шифровании, так и при расшифровании сообщений.

Как открытый текст, так и шифртекст образуются из букв, входящих в конечное множество символов, называемых алфавитом. Примерами алфавитов являются конечное множество всех заглавных букв, конечное множество всех заглавных и строчных букв и цифр и т. п. В общем виде некоторый алфавит Σ можно представить так:

$$\Sigma = \{a_0, a_1, a_2, \ldots, a_{m-1}\}.$$

Объединяя по определенному правилу буквы из алфавита Σ, можно создать новые алфавиты:

- алфавит Σ^2, содержащий m^2 биграмм a_0a_0, a_0a_1, ..., $a_{m-1}a_{m-1}$;
- алфавит Σ^3, содержащий m^3 триграмм $a_0a_0a_0, a_0a_0a_1, \ldots, a_{m-1}a_{m-1}a_{m-1}$.

В общем случае, объединяя по n букв, получаем алфавит Σ^n, содержащий m^n n-грамм [95].

Например, английский алфавит

$$\Sigma = \{ABCDEFGH \ldots WXYZ\}$$

объемом m = 26 букв позволяет сгенерировать посредством операции конкатенации алфавит из $26^2 = 676$ биграмм

$$AA, AB, \ldots, XZ, ZZ,$$

алфавит из $26^3 = 17576$ триграмм

$$AAA, AAB, \ldots, ZZX, ZZZ \quad \text{и т.д.}$$

При выполнении криптографических преобразований полезно заменить буквы алфавита целыми числами 0, 1, 2, 3,.... Это позволяет упростить выполнение необходимых алгебраических манипуляций. Например, можно установить взаимно однозначное соответствие между русским алфавитом

$$\Sigma_{рус.} = \{АБВГДЕ \ldots ЮЯ\}$$

и множеством целых

$$\bar{Z}_{32} = \{0, 1, 2, 3, \ldots, 31\};$$

между английским алфавитом

$$\Sigma_{англ.} = \{ABCDEF \ldots YZ\}$$

и множеством целых

$$\bar{Z}_{26} = \{0, 1, 2, 3, \ldots, 25\}$$

(см. табл. 2.1 и 2.2).

В дальнейшем будет обычно использоваться алфавит

$$\bar{Z}_m = \{0, 1, 2, 3, \ldots, m-1\},$$

содержащий m "букв" (в виде чисел).

Замена букв традиционного алфавита числами позволяет более четко сформулировать основные концепции и приемы криптографических преобразований. В то же время в большинстве иллюстраций будет использоваться алфавит естественного языка.

Таблица 2.1

Соответствие между русским алфавитом и множеством целых

$\bar{Z}_{32} = \{0, 1, 2, 3, \ldots, 31\}$

Буква	Число	Буква	Число	Буква	Число	Буква	Число
А	0	И	8	Р	16	Ш	24
Б	1	Й	9	С	17	Щ	25
В	2	К	10	Т	18	Ь	26
Г	3	Л	11	У	19	Ы	27
Д	4	М	12	Ф	20	Ъ	28
Е	5	Н	13	Х	21	Э	29
Ж	6	О	14	Ц	22	Ю	30
З	7	П	15	Ч	23	Я	31

Таблица 2.2

Соответствие между английским алфавитом и множеством целых

$\bar{Z}_{26} = \{0, 1, 2, 3, ..., 25\}$

Буква	Число	Буква	Число	Буква	Число
A	0	J	9	S	18
B	1	K	10	T	19
C	2	L	11	U	20
D	3	M	12	V	21
E	4	N	13	W	22
F	5	O	14	X	23
G	6	P	15	Y	24
H	7	Q	16	Z	25
I	8	R	17		

Текст с п буквами из алфавита $\bar{Z}_m$ можно рассматривать как п-грамму

$$\bar{x} = (x_0, x_1, x_2, \ldots, x_{n-1}),$$

где $x_i \in \bar{Z}_m$, $0 \le i < n$, для некоторого целого $n = 1, 2, 3, \ldots$.

Через $\bar{Z}_{m,n}$ будем обозначать множество п-грамм, образованных из букв множества $\bar{Z}_m$.

Криптографическое преобразование E представляет собой совокупность преобразований

$$E = \{E^{(n)} : 1 \le n < \infty\},$$

$$E^{(n)} : \bar{Z}_{m,n} \to \bar{Z}_{m,n}. \qquad (2.1)$$

Преобразование $E^{(n)}$ определяет, как каждая п-грамма открытого текста $\bar{x} \in \bar{Z}_{m,n}$ заменяется п-граммой шифртекста $\bar{y}$, т.е.

$$\bar{y} = E^{(n)}(\bar{x}), \quad \text{причем} \quad \bar{x}, \bar{y} \in \bar{Z}_{m,n};$$

при этом обязательным является требование взаимной однозначности преобразования $E^{(n)}$ на множестве $\bar{Z}_{m,n}$.

Криптографическая система может трактоваться как семейство криптографических преобразований

$$\bar{E} = \{E_K : K \in \bar{K}\}, \qquad (2.2)$$

помеченных параметром K, называемым ключом.

Множество значений ключа образует ключевое пространство $\bar{K}$.

Далее рассматриваются традиционные (классические) методы шифрования, отличающиеся симметричной функцией шифрования. К ним относятся шифры перестановки, шифры простой и сложной замены, а также некоторые их модификации и комбинации. Следует отметить, что комбинации шифров перестановок и шифров замены образуют все многообразие применяемых на практике симметричных шифров.

Приводимые сведения о шифрах каждой группы даются по возможности в хронологическом порядке, что позволяет постепенно вводить читателя в сферу криптографии. Как известно, довольно трудно понять концептуальную схему науки, ее модели и методы исследования, если не иметь хотя бы общего представления об истории развития этой науки.

2.2. Шифры перестановки

При шифровании перестановкой символы шифруемого текста переставляются по определенному правилу в пределах блока этого текста. Шифры перестановки являются самыми простыми и, вероятно, самыми древними шифрами.

Шифрующие таблицы

С начала эпохи Возрождения (конец XIV столетия) начала возрождаться и криптография. Наряду с традиционными применениями криптографии в политике, дипломатии и военном деле появляются и другие задачи – защита интеллектуальной собственности от преследований инквизиции или заимствований злоумышленников. В разработанных шифрах перестановки того времени применяются шифрующие таблицы, которые в сущности задают правила перестановки букв в сообщении.

В качестве ключа в шифрующих таблицах используются:

- размер таблицы;
- слово или фраза, задающие перестановку;
- особенности структуры таблицы.

Одним из самых примитивных табличных шифров перестановки является простая перестановка, для которой ключом служит размер таблицы. Этот метод шифрования сходен с шифром скитала. Например, сообщение

ТЕРМИНАТОР ПРИБЫВАЕТ СЕДЬМОГО В ПОЛНОЧЬ

записывается в таблицу поочередно по столбцам. Результат заполнения таблицы из 5 строк и 7 столбцов показан на рис. 2.1.

Т	Н	П	В	Е	Г	Л
Е	А	Р	А	Д	О	Н
Р	Т	И	Е	Ь	В	О
М	О	Б	Т	М	П	Ч
И	Р	Ы	С	О	О	Ь

Рис. 2.1. Заполнение таблицы из 5 строк и 7 столбцов

После заполнения таблицы текстом сообщения по столбцам для формирования шифртекста считывают содержимое таблицы по строкам. Если шифртекст записывать группами по пять букв, получается такое шифрованное сообщение:

ТНПВЕ ГЛЕАР АДОНР ТИЕЬВ ОМОБТ МПЧИР ЫСООЬ

Естественно, отправитель и получатель сообщения должны заранее условиться об общем ключе в виде размера таблицы. Следует заметить, что объединение букв шифртекста в 5-буквенные группы не входит в ключ шифра и осуществляется для удобства записи несмыслового текста. При расшифровании действия выполняют в обратном порядке.

Несколько большей стойкостью к раскрытию обладает метод шифрования, называемый одиночной перестановкой по ключу. Этот метод отличается от предыдущего тем, что столбцы таблицы переставляются по ключевому слову, фразе или набору чисел длиной в строку таблицы.

Применим в качестве ключа, например, слово

ПЕЛИКАН,

а текст сообщения возьмем из предыдущего примера. На рис. 2.2 показаны две таблицы, заполненные текстом сообщения и ключевым словом, при этом левая таблица соответствует заполнению до перестановки, а правая таблица – заполнению после перестановки.

Ключ →

П	Е	Л	И	К	А	Н
7	2	5	3	4	1	6
Т	Н	П	В	Е	Г	Л
Е	А	Р	А	Д	О	Н
Р	Т	И	Е	Ь	В	О
М	О	Б	Т	М	П	Ч
И	Р	Ы	С	О	О	Ь

До перестановки

А	Е	И	К	Л	Н	П
1	2	3	4	5	6	7
Г	Н	В	Е	П	Л	Т
О	А	А	Д	Р	Н	Е
В	Т	Е	Ь	И	О	Р
П	О	Т	М	Б	Ч	М
О	Р	С	О	Ы	Ь	И

После перестановки

Рис. 2.2. Таблицы, заполненные ключеаым словом и текстом сообщения

В верхней строке левой таблицы записан ключ, а номера под буквами ключа определены в соответствии с естественным порядком соответствующих букв ключа в алфавите. Если бы в ключе встретились одинаковые буквы, они бы были понумерованы слева направо. В правой таблице столбцы переставлены в соответствии с упорядоченными номерами букв ключа.

При считывании содержимого правой таблицы по строкам и записи шифртекста группами по пять букв получим шифрованное сообщение:

ГНВЕП ЛТООА ДРНЕВ ТЕЬИО РПОТМ БЧМОР СОЫЬИ

Для обеспечения дополнительной скрытности можно повторно зашифровать сообщение, которое уже прошло шифрование. Такой метод шифрования называется *двойной перестановкой*. В случае двойной перестановки столбцов и строк таблицы перестановки определяются отдельно для столбцов и отдельно для строк. Сначала в таблицу записывается текст сообщения, а потом поочередно переставляются столбцы, а затем строки. При расшифровании порядок перестановок должен быть обратным.

Пример выполнения шифрования методом двойной перестановки показан на рис. 2.3. Если считывать шифртекст из правой таблицы построчно блоками по четыре буквы, то получится следующее:

ТЮАЕ ООГМ РЛИП ОЬСВ

Ключом к шифру двойной перестановки служит последовательность номеров столбцов и номеров строк исходной таблицы (в нашем примере последовательности 4132 и 3142 соответственно).

	4	1	3	2
3	П	Р	И	Л
1	Е	Т	А	Ю
4	В	О	С	Ь
2	М	О	Г	О

Исходная таблица

	1	2	3	4
3	Р	Л	И	П
1	Т	Ю	А	Е
4	О	Ь	С	В
2	О	О	Г	М

Перестановка столбцов

	1	2	3	4
1	Т	Ю	А	Е
2	О	О	Г	М
3	Р	Л	И	П
4	О	Ь	С	В

Перестановка строк

Рис. 2.3. Пример выполнения шифрования методом двойной перестановки

Число вариантов двойной перестановки быстро возрастает при увеличении размера таблицы:

- для таблицы 3×3 36 вариантов;
- для таблицы 4×4 576 вариантов;
- для таблицы 5×5 14400 вариантов.

Однако двойная перестановка не отличается высокой стойкостью и сравнительно просто "взламывается" при любом размере таблицы шифрования.

Применение магических квадратов

В средние века для шифрования перестановкой применялись и магические квадраты.

Магическими квадратами называют квадратные таблицы с вписанными в их клетки последовательными натуральными числами, начиная от 1, которые дают в сумме по каждому столбцу, каждой строке и каждой диагонали одно и то же число.

Шифруемый текст вписывали в магические квадраты в соответствии с нумерацией их клеток. Если затем выписать содержимое такой таблицы по строкам, то получится шифртекст, сформированный благодаря перестановке букв исходного сообщения. В те времена считалось, что созданные с помощью магических квадратов шифртексты охраняет не только ключ, но и магическая сила.

Пример магического квадрата и его заполнения сообщением

ПРИЛЕТАЮ ВОСЬМОГО

показан на рис. 2.4.

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

О	И	Р	М
Е	О	С	Ю
В	Т	А	Ь
Л	Г	О	П

Рис. 2.4. Пример магического квадрата 4×4 и его заполнения сообщением ПРИЛЕТАЮ ВОСЬМОГО

Шифртекст, получаемый при считывании содержимого правой таблицы по строкам, имеет вполне загадочный вид:

ОИРМ ЕОСЮ ВТАЬ ЛГОП

Число магических квадратов быстро возрастает с увеличением размера квадрата. Существует только один магический квадрат размером 3×3 (если не учитывать его повороты). Количество магических квадратов 4×4 составляет уже 880, а количество магических квадратов 5×5 – около 250000.

Магические квадраты средних и больших размеров могли служить хорошей базой для обеспечения нужд шифрования того времени, поскольку практически нереально выполнить вручную перебор всех вариантов для такого шифра.

2.3. Шифры простой замены

При шифровании заменой (подстановкой) символы шифруемого текста заменяются символами того же или другого алфавита с заранее установленным правилом замены. В шифре простой замены каждый символ исходного текста заменяется символами того же алфавита одинаково на всем протяжении текста. Часто шифры простой замены называют шифрами одноалфавитной подстановки.

Система шифрования Цезаря

Шифр Цезаря является частным случаем шифра простой замены (одноалфавитной подстановки). Свое название этот шифр получил по имени римского императора Гая Юлия Цезаря, который использовал этот шифр при переписке с Цицероном (около 50 г. до н. э.).

При шифровании исходного текста каждая буква заменялась на другую букву того же алфавита по следующему правилу. Заменяющая буква определялась путем смещения по алфавиту от исходной буквы на К букв. При достижении конца алфавита выполнялся циклический переход к его началу. Цезарь использовал шифр замены при смещении К= 3. Такой шифр замены можно задать таблицей подстановок, содержащей соответствующие пары букв открытого текста и шифртекста. Совокупность возможных подстановок для К= 3 показана в табл. 2.3.

Например, послание Цезаря

VENI VIDI VICI

(в переводе на русский означает "Пришел, Увидел, Победил"), направленное его другу Аминтию после победы над понтийским царем Фарнаком, сыном Митридата, выглядело бы в зашифрованном виде так:

YHQL YLGL YLFL

Таблица 2.3

Одноалфавитные подстановки (К = 3, m = 26)

A → D	J → M	S → V
B → E	K → N	T → W
C → F	L → O	U → X
D → G	M → P	V → Y
E → H	N → Q	W → Z
F → I	O → R	X → A
G → J	P → S	Y → B
H → K	Q → T	Z → C
I → L	R → U	

Выполним математический анализ шифра простой замены (подстановки) на основе понятий, введенных в начале гл. 2 [113].

Подстановка в алфавите $\overline{Z}_m$ является взаимно однозначным отображением π из $\overline{Z}_m$ на $\overline{Z}_m$:

$$\pi : t \to \pi(t),$$

которое заменяет букву t открытого текста на букву $\pi(t)$ шифртекста. Множество всех подстановок на $\overline{Z}_m$ называется симметричной группой $\overline{Z}_m$ и обозначается $\overline{SYM}(\overline{Z}_m)$.

Симметричная группа $\overline{SYM}(\overline{Z}_m)$ обладает следующими свойствами:

1. *Замкнутость.* Произведение подстановок $\pi_1\pi_2$ является подстановкой:

$$\pi : \bar{Z}_m \xrightarrow{\pi_2} \bar{Z}_m \xrightarrow{\pi_1} \bar{Z}_m,$$

$$\pi : t \to \pi_1(\pi_2(t)).$$

2. *Ассоциативность.* Оба способа заключения в скобки произведения подстановок $\pi_1\pi_2\pi_3$:

$$\pi_1(\pi_2\pi_3) = (\pi_1\pi_2)\,\pi_3$$

дают одинаковый результат.

3. *Существование единичного элемента.* Подстановка δ, определенная как

$$\delta(t) = t, \quad 0 \le t < m,$$

является единственным единичным элементом группы $\overline{SYM}(\bar{Z}_m)$ по умножению:

$$\delta\pi = \delta\pi \quad \text{для всех} \quad \pi \in \overline{SYM}(\bar{Z}_m).$$

4. *Существование обратных элементов.* Для каждой подстановки π имеется взаимно однозначно определенная обратная подстановка, обозначаемая π^{-1}, которая удовлетворяет соотношению:

$$\pi\pi^{-1} = \delta.$$

Указанные свойства являются аксиомами группы.

Ключ К *подстановки* для алфавита $\bar{Z}_m$ представляет собой последовательность элементов симметричной группы из $\bar{Z}_m$:

$$K = (\pi_0, \pi_1, \ldots, \pi_{n-1}, \ldots), \ \pi_n \in \overline{SYM}(\bar{Z}_m), \quad 0 \le n < \infty.$$

Подстановка, определяемая ключом К, является криптографическим преобразованием E_K, которое шифрует n-грамму $(x_0, x_1, x_2, \ldots, x_{n-1})$ открытого текста в n-грамму $(y_0, y_1, y_2, \ldots, y_{n-1})$ шифртекста, где

$$y_i = \pi_i(x_i), \quad 0 \le i < n,$$

для каждого n, n = 1, 2, 3, ...

Криптографическое преобразование E_K называется *одноалфавитной подстановкой*, если значение π_i одинаково для каждого i, i = 0, 1, 2, ...; в противном случае преобразование E_K называется *многоалфавитной подстановкой.*

На рис. 2.5 представлена схема реализации подстановки E_K.

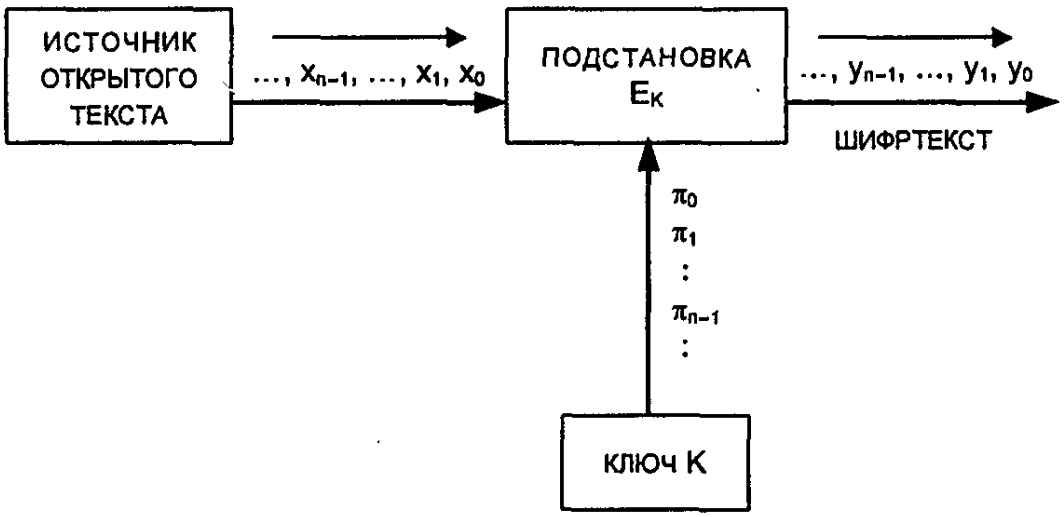


Рис. 2.5. Схема подстановки E_K

Отметим характерные особенности подстановки E_K:

- открытый текст шифруется побуквенно (буква за буквой);
- i-я буква y_i шифртекста является функцией только i-й компоненты π_i ключа К и i-й буквы x_i открытого текста;
- шифрование n-граммы $(x_0, x_1, x_2, \ldots, x_{n-1})$ производится в соответствии с формулой

$$(y_0, y_1, y_2, \ldots, y_{n-1}) = E_K (x_0, x_1, x_2, \ldots, x_{n-1}).$$

Система Цезаря представляет собой одноалфавитную подстановку, которая шифрует n-грамму $(x_0, x_1, x_2, \ldots, x_{n-1})$ открытого текста в n-грамму $(y_0, y_1, y_2, \ldots, y_{n-1})$ шифртекста согласно следующему правилу:

$$y_i = E_K (x_i), \quad 0 \le i < n, \tag{2.3}$$

$$E_K : j \to (j + K) \pmod n, \quad 0 \le K < m,$$

где j – числовой код буквы открытого текста; j + K – числовой код соответствующей буквы шифртекста.

В отличие от шифра Цезаря, описанного в начале этого подраздела, система шифрования Цезаря образует по существу семейство одноалфавитных подстановок для выбираемых значений ключа К, причем $0 \le K < m$.

Достоинством системы шифрования Цезаря является простота шифрования и расшифрования. К недостаткам системы Цезаря следует отнести следующие:

- подстановки, выполняемые в соответствии с системой Цезаря, не маскируют частот появления различных букв исходного открытого текста;
- сохраняется алфавитный порядок в последовательности заменяющих букв; при изменении значения К изменяются только начальные позиции такой последовательности;

- число возможных ключей K мало;
- шифр Цезаря легко вскрывается на основе анализа частот появления букв в шифртексте.

Криптоаналитическая атака против системы одноалфавитной замены начинается с подсчета частот появления символов: определяется число появлений каждой буквы в шифртексте. Затем полученное распределение частот букв в шифртексте сравнивается с распределением частот букв в алфавите исходных сообщений, например в английском. Буква с наивысшей частотой появления в шифртексте заменяется на букву с наивысшей частотой появления в английском языке и т.д. Вероятность успешного вскрытия системы шифрования повышается с увеличением длины шифртекста.

Концепция, заложенная в систему шифрования Цезаря, оказалась весьма плодотворной, о чем свидетельствуют ее многочисленные модификации. Несколько таких модификаций будут рассмотрены ниже.

Аффинная система подстановок Цезаря

В системе шифрования Цезаря использовались только аддитивные свойства множества целых $\overline{Z}_m$. Однако символы множества $\overline{Z}_m$ можно также умножать по модулю m. Применяя одновременно операции сложения и умножения по модулю m над элементами множества $\overline{Z}_m$, можно получить систему подстановок, которую называют аффинной системой подстановок Цезаря.

Определим преобразование в такой системе:

$$E_{a,b}: \overline{Z}_m \to \overline{Z}_m,$$

$$E_{a,b}: t \to E_{a,b}(t),$$

$$E_{a,b}(t) = at + b \pmod m, \qquad (2.4)$$

где a, b – целые числа, $0 \le a, b < m$, НОД $(a, m) = 1$.

В данном преобразовании буква, соответствующая числу t, заменяется на букву, соответствующую числовому значению $(at + b)$ по модулю m.

Следует заметить, что преобразование $E_{a,b}(t)$ является взаимно однозначным отображением на множестве $\overline{Z}_m$ только в том случае, если наибольший общий делитель чисел a и m, обозначаемый как НОД (a, m), равен единице, т.е. a и m должны быть взаимно простыми числами.

Например, пусть $m = 26$, $a = 3$, $b = 5$. Тогда, очевидно, НОД $(3, 26) = 1$, и мы получаем следующее соответствие между числовыми кодами букв:

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
3t+5	5	8	11	14	17	20	23	0	3	6	9	12	15	18	21	24	1	4	7	10	13	16	19	22	25	2

Преобразуя числа в буквы английского языка, получаем следующее соответствие для букв открытого текста и шифртекста:

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

Исходное сообщение HOPE преобразуется

в шифртекст AVYR

Достоинством аффинной системы является удобное управление ключами – ключи шифрования и расшифрования представляются в компактной форме в виде пары чисел (a, b). Недостатки аффинной системы аналогичны недостаткам системы шифрования Цезаря.

Аффинная система использовалась на практике несколько веков назад, а сегодня ее применение ограничивается большей частью иллюстрациями основных криптологических положений.

Система Цезаря с ключевым словом

Система шифрования Цезаря с ключевым словом является одноалфавитной системой подстановки. Особенностью этой системы является использование ключевого слова для смещения и изменения порядка символов в алфавите подстановки [79].

Выберем некоторое число k, $0 \le k < 25$, и слово или короткую фразу в качестве *ключевого слова*. Желательно, чтобы все буквы ключевого слова были различными. Пусть выбраны слово DIPLOMAT в качестве ключевого слова и число k = 5.

Ключевое слово записывается под буквами алфавита, начиная с буквы, числовой код которой совпадает с выбранным числом k:

```
0 1 2 3 4 5              10             15             20             25
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
        D I P L O M A T
```

Оставшиеся буквы алфавита подстановки записываются после ключевого слова в алфавитном порядке:

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

Теперь мы имеем подстановку для каждой буквы произвольного сообщения.

Исходное сообщение SEND MORE MONEY

шифруется как HZBY TCGZ TCBZS

Следует отметить, что требование о различии всех букв ключевого слова не обязательно. Можно просто записать ключевое слово (или фразу) без повторения одинаковых букв. Например, ключевая фраза

КАК ДЫМ ОТЕЧЕСТВА НАМ СЛАДОК И ПРИЯТЕН

и число k = 3 порождают следующую таблицу подстановок:

```
0   3
А Б В Г Д Е Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ь Ы Ъ Э Ю Я
Ъ Э Ю К А Д Ы М О Т Е Ч С В Н Л И П Р Я Б Г Ж З Й У Ф Х Ц Ш Щ Ь
```

Несомненным достоинством системы Цезаря с ключевым словом является то, что количество возможных ключевых слов практически неисчерпаемо. Недостатком этой системы является возможность взлома шифртекста на основе анализа частот появления букв.

Шифрующие таблицы Трисемуса

В 1508 г. аббат из Германии Иоганн Трисемус написал печатную работу по криптологии под названием "Полиграфия". В этой книге он впервые систематически описал применение шифрующих таблиц, заполненных алфавитом в случайном порядке [32]. Для получения такого шифра замены обычно использовались таблица для записи букв алфавита и ключевое слово (или фраза). В таблицу сначала вписывалось по строкам ключевое слово, причем повторяющиеся буквы отбрасывались. Затем эта таблица дополнялась не вошедшими в нее буквами алфавита по порядку. Поскольку ключевое слово или фразу легко хранить в памяти, то такой подход упрощал процессы шифрования и расшифрования.

Поясним этот метод шифрования на примере. Для русского алфавита шифрующая таблица может иметь размер 4×8. Выберем в качестве ключа слово БАНДЕРОЛЬ. Шифрующая таблица с таким ключом показана на рис. 2.6.

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

Рис. 2.6. Шифрующая таблица с ключевым словом БАНДЕРОЛЬ

Как и в случае полибианского квадрата, при шифровании находят в этой таблице очередную букву открытого текста и записывают в шифртекст букву, расположенную ниже ее в том же столбце. Если буква текста оказывается в нижней строке таблицы, тогда для шифртекста берут самую верхнюю букву из того же столбца.

Например, при шифровании с помощью этой таблицы сообщения

ВЫЛЕТАЕМПЯТОГО

получаем шифртекст

ПДКЗЫВЗЧШЛЫЙЙСЙ

Такие табличные шифры называются монограммными, так как шифрование выполняется по одной букве. Трисемус первым заметил, что шифрующие таблицы позволяют шифровать сразу по две буквы. Такие шифры называются *биграммными*.

Биграммный шифр Плейфейра

Шифр Плейфейра, изобретенный в 1854 г., является наиболее известным биграммным шифром замены. Он применялся Великобританией во время первой мировой войны. Основой шифра Плейфейра является шифрующая таблица со случайно расположенными буквами алфавита исходных сообщений.

Для удобства запоминания шифрующей таблицы отправителем и получателем сообщений можно использовать ключевое слово (или фразу) при заполнении начальных строк таблицы. В целом структура шифрующей таблицы системы Плейфейра полностью аналогична структуре шифрующей таблицы Трисемуса. Поэтому для пояснения процедур шифрования и расшифрования в системе Плейфейра воспользуемся шифрующей таблицей Трисемуса из предыдущего раздела (см. рис. 2.6).

Процедура шифрования включает следующие шаги.

1. Открытый текст исходного сообщения разбивается на пары букв (биграммы). Текст должен иметь четное количество букв и в нем не должно быть биграмм, содержащих две одинаковые буквы. Если эти требования не выполнены, то текст модифицируется даже из-за незначительных орфографических ошибок.
2. Последовательность биграмм открытого текста преобразуется с помощью шифрующей таблицы в последовательность биграмм шифртекста по следующим правилам:
   2а. Если обе буквы биграммы открытого текста не попадают на одну строку или столбец (как, например, буквы А и Й в табл. на рис. 2.6), тогда находят буквы в углах прямоугольника, определяемого данной парой букв. (В нашем примере это – буквы АЙОВ. Пара букв АЙ отображается в пару ОВ. Последовательность букв в биграмме шифртекста должна быть зеркально расположенной по отношению к последовательности букв в биграмме открытого текста.)
   2б. Если обе буквы биграммы открытого текста принадлежат одному столбцу таблицы, то буквами шифртекста считаются буквы, которые лежат под ними. (Например, биграмма НС

дает биграмму шифртекста ГЩ.) Если при этом буква открытого текста находится в нижней строке, то для шифртекста берется соответствующая буква из верхней строки того же столбца. (Например, биграмма ВШ дает биграмму шифртекста ПА.)

2в. Если обе буквы биграммы открытого текста принадлежат одной строке таблицы, то буквами шифртекста считаются буквы, которые лежат справа от них. (Например, биграмма НО дает биграмму шифртекста ДЛ.) Если при этом буква открытого текста находится в крайнем правом столбце, то для шифра берут соответствующую букву из левого столбца в той же строке. (Например, биграмма ФЦ дает биграмму шифртекста ХМ.)

Зашифруем текст

ВСЕ ТАЙНОЕ СТАНЕТ ЯВНЫМ

Разбиение этого текста на биграммы дает

ВС ЕТ АЙ НО ЕС ТА НЕ ТЯ ВН ЫМ

Данная последовательность биграмм открытого текста преобразуется с помощью шифрующей таблицы (см. рис. 2.6) в следующую последовательность биграмм шифртекста:

ГП ДУ ОВ ДЛ НУ ПД ДР ЦЫ ГА ЧТ

При расшифровании применяется обратный порядок действий.

Следует отметить, что шифрование биграммами резко повышает стойкость шифров к вскрытию. Хотя книга И. Трисемуса "Полиграфия" была относительно доступной, описанные в ней идеи получили признание лишь спустя три столетия. По всей вероятности, это было обусловлено плохой осведомленностью криптографов о работах богослова и библиофила Трисемуса в области криптографии [32].

Криптосистема Хилла

Алгебраический метод, обобщающий аффинную подстановку Цезаря

$$E_{a,b} : \overline{Z}_m \rightarrow \overline{Z}_m ,$$

$$E_{a,b}(t) = t \rightarrow at + b \ (\mathrm{mod}\ m)$$

для определения n-грамм, был сформулирован Лестером С. Хиллом [79].

Множество целых $\overline{Z}_m$, для которого определены операции сложения, вычитания и умножения по модулю m, является примером кольца. Кольцо R представляет собой алгебраическую систему,

в которой определены операции сложения, вычитания и умножения пар элементов. Эта алгебраическая система обладает рядом свойств:

- элементы кольца R образуют коммутативную группу относительно операции сложения; кроме того, существуют единичный и обратный элементы относительно операции сложения;
- умножение и сложение удовлетворяют ассоциативному и дистрибутивному законам.

Мультипликативное обратное α^{-1} элемента α кольца может существовать не всегда. Например, если модуль $m = 26$, то значения $2^{-1} \pmod{26}$ и $13^{-1} \pmod{26}$ не могут существовать.

Если модуль m является простым числом p, то существует обратная величина любого ненулевого элемента t из $\bar{Z}_p$ (при $m = p$), поскольку значения

$$t \pmod{m},\ 2t \pmod{m},\ 3t \pmod{m},\ \ldots,\ (p-1)\,t \pmod{m}$$

различаются, если $1 \le t \le p-1$.

Множество $\bar{Z}_p$, где p – простое число, является примером алгебраической системы, называемой конечным полем. Ненулевые элементы $\bar{Z}_p$ образуют мультипликативную группу.

Множество всех n-грамм $\bar{x} = (x_0, x_1, x_2, \ldots, x_{n-1})$ с компонентами из кольца $\bar{Z}_m$ образует векторное пространство $\bar{Z}_{m,n}$ над кольцом $\bar{Z}_m$. Каждая n-грамма $\bar{x}$ называется вектором. В векторном пространстве $\bar{Z}_{m,n}$ для векторов $\bar{x}$ определены операции сложения и вычитания по модулю n, а также скалярное умножение вектора на элемент t кольца $\bar{Z}_m$. Сложение и скалярное умножение являются операциями, удовлетворяющими коммутативному, ассоциативному и дистрибутивному законам. Вектор $\bar{x}$ является линейной комбинацией векторов

$$\{\bar{x}^{(i)} : 0 \le i < L\}, \quad \text{если}$$

$$\bar{x} = t_0\bar{x}^{(0)} + t_1\bar{x}^{(1)} + \ldots + t_{L-1}\bar{x}^{(L-1)} \pmod{m}. \qquad (2.5)$$

Линейное преобразование $\bar{T}$ является отображением:

$$\bar{T} : \bar{Z}_{m,n} \to \bar{Z}_{m,n},$$

$$\bar{T} : \bar{x} \to \bar{y}, \quad \bar{y} = \bar{T}(\bar{x}), \qquad (2.6)$$

которое удовлетворяет условию линейности

$$\bar{T} = (t * \bar{x} + s * \bar{y}) = t * \bar{T}(\bar{x}) + s * \bar{T}(\bar{y}) \pmod{m}$$

для всех s, t в $\bar{Z}_m$ и $\bar{x}, \bar{y}$ в $\bar{Z}_{m,n}$.

Линейное преобразование $\bar{T}$ может быть представлено матрицей размером n×n вида

$$\bar{T} = \begin{bmatrix} \gamma_{0,0} & \gamma_{0,1} & \cdots & \gamma_{0,j} & \cdots & \gamma_{0,n-1} \\ \gamma_{1,0} & \gamma_{1,1} & \cdots & \gamma_{1,j} & \cdots & \gamma_{1,n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \gamma_{i,0} & \gamma_{i,1} & \cdots & \gamma_{i,j} & \cdots & \gamma_{j,n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \gamma_{n-1,0} & \gamma_{n-1,1} & \cdots & \gamma_{n-1,j} & \cdots & \gamma_{n-1,n-1} \end{bmatrix}, \qquad (2.7)$$

причем

$$y_i = \gamma_{i,0} * x_0 + \gamma_{i,1} * x_1 + \ldots + \gamma_{ij} * x_j + \ldots + \gamma_{i,n-1} * x_{n-1} \pmod m$$

или

$$y_i = \sum_{j=0}^{n-1} \gamma_{ij} * x_j \pmod m, \quad 0 \le i \le n-1.$$

Базисом для векторного пространства $\bar{Z}_{m,n}$ является набор векторов из $\{\bar{x}^{(i)}: 0 \le i < L\}$, которые линейно независимы и порождают $\bar{Z}_{m,n}$. Каждый базис для $\bar{Z}_{m,n}$ содержит n линейно независимых векторов. Любой набор из n векторов, которые линейно независимы над $\bar{Z}_{m,n}$, является базисом.

Пусть $\bar{T}$ является линейным преобразованием, описываемым матрицей (2.7), причем

$$\bar{T}: \bar{Z}_{m,n} \to \bar{Z}_{m,n}.$$

Если векторы $\{\bar{x}^{(i)}: 0 \le i < n\}$ линейно независимы над $\bar{Z}_{m,n}$, тогда их образы $\{\bar{T}(\bar{x}^{(i)}): 0 \le i < n\}$ линейно независимы над $\bar{Z}_{m,n}$ только в том случае, если определитель матрицы $\bar{T}$, обозначаемый как det $(\bar{T})$, не делится на любое простое p, которое делит m. В этом случае преобразование $\bar{T}$ называется обратимым (или невырожденным) линейным преобразованием, имеющим обратное преобразование $\bar{T}^{-1}$:

$$\bar{T}^{-1}: \bar{Z}_{m,n} \to \bar{Z}_{m,n},$$

$$\bar{T}\bar{T}^{-1} = \bar{T}^{-1}T = \bar{I}, \qquad (2.8)$$

где $\bar{I}$ – единичная матрица. Кроме того, $\bar{T}^{-1}$ также является линейным преобразованием.

Например, когда m = 26 и матрица преобразования

$$\bar{T} = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix},$$

то определитель этой матрицы

$$\det(\bar{T}) = 9 = 1 (\mathrm{mod}\ 2),$$

$$\det(\bar{T}) = 9 = 9 (\mathrm{mod}\ 13).$$

Поэтому существует обратное преобразование $\bar{T}^{-1}$. Нетрудно убедиться, что

$$\bar{T} = \begin{bmatrix} 15 & 17 \\ 20 & 9 \end{bmatrix}$$

удовлетворяет соотношению

$$\bar{T}\bar{T}^{-1} = \bar{T}^{-1}\bar{T} = \bar{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Пусть $\bar{T}$ является линейным преобразованием на $\bar{Z}_{26,2}$ с матрицей

$$\bar{T} = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix}.$$

Используем это преобразование $\bar{T}$ для определения биграммной подстановки в английском алфавите {ABCDEFGH..XYZ}. Сначала разобьем n-грамму открытого текста на биграммы, причем выберем n кратным 2. Например, 12-грамма PAYMOREMONEY делится на шесть биграмм:

PA YM OR EM ON EY

Затем в каждой биграмме открытого текста заменим каждую букву ее числовым эквивалентом из таблицы:

PA ⇄ 15 0; YM ⇄ 24 12; OR ⇄ 14 17;

EM ⇄ 4 12; ON ⇄ 14 13; EY ⇄ 4 24;

Преобразование биграмм $\bar{x}_j$ открытого текста в биграммы $\bar{y}_i$ шифртекста осуществляется в соответствии с уравнением

$$\bar{y}_j = \bar{T} * \bar{x}_j (\mathrm{mod}\ 26)$$

или

$$\bar{y}_j = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} * \bar{x}_j (\mathrm{mod}\ 26),$$

где $\bar{x}_j$ и $\bar{y}_i$ – вектор-столбцы биграмм шифртекста и открытого текста соответственно.

Получаем

$$\bar{y}_1 = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} * \begin{vmatrix} 15 \\ 0 \end{vmatrix} = \begin{vmatrix} 19 \\ 4 \end{vmatrix},$$

$$\bar{y}_2 = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} * \begin{vmatrix} 24 \\ 12 \end{vmatrix} = \begin{vmatrix} 4 \\ 4 \end{vmatrix},$$

$$\bar{y}_3 = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} * \begin{vmatrix} 14 \\ 17 \end{vmatrix} = \begin{vmatrix} 15 \\ 9 \end{vmatrix},$$

$$\bar{y}_4 = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} * \begin{vmatrix} 4 \\ 12 \end{vmatrix} = \begin{vmatrix} 22 \\ 16 \end{vmatrix},$$

$$\bar{y}_5 = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} * \begin{vmatrix} 14 \\ 13 \end{vmatrix} = \begin{vmatrix} 3 \\ 15 \end{vmatrix},$$

$$\bar{y}_6 = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} * \begin{vmatrix} 4 \\ 24 \end{vmatrix} = \begin{vmatrix} 6 \\ 24 \end{vmatrix}.$$

Заменяя в биграммах шифртекста числа на соответствующие буквы согласно табл. 2.2, получаем 12-грамму шифртекста

TE EE PJ WQ DP GY

Для расшифрования биграмм $\bar{y}_i$ шифртекста и восстановления биграмм $\bar{x}_j$ открытого текста необходимо выполнить обратное преобразование $\bar{T}^{-1}$ согласно уравнению

$$\bar{x}_j = \bar{T}^{-1} * \bar{y}_j.$$

В рассмотренном примере матрицы преобразования имели размер 2×2 и шифровались биграммы (пары) букв. Хотя буква Е может быть зашифрована по-разному в различных парах исходного сообщения, одна и та же пара, например ЕМ, будет шифроваться всегда одинаково на протяжении всего исходного текста.

Система Хилла является одноалфавитной в широком смысле слова.

2.4. Шифры сложной замены

Шифры сложной замены называют многоалфавитными, так как для шифрования каждого символа исходного сообщения применяют свой шифр простой замены. Многоалфавитная подстановка последовательно и циклически меняет используемые алфавиты.

При r-алфавитной подстановке символ x_0 исходного сообщения заменяется символом y_0 из алфавита B_0, символ x_1 – символом y_1 из алфавита B_1, и так далее, символ x_{r-1} заменяется символом y_{r-1} из алфавита B_{r-1}, символ x_r заменяется символом y_r снова из алфавита B_0, и т.д.

Общая схема многоалфавитной подстановки для случая $r = 4$ показана на рис. 2.7.

Входной символ:	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9
Алфавит подстановки:	B_0	B_1	B_2	B_3	B_0	B_1	B_2	B_3	B_0	B_1

Рис. 2.7. Схема r-алфавитной подстановки для случая r = 4

Эффект использования многоалфавитной подстановки заключается в том, что обеспечивается маскировка естественной статистики исходного языка, так как конкретный символ из исходного алфавита A может быть преобразован в несколько различных символов шифровальных алфавитов B_j. Степень обеспечиваемой защиты теоретически пропорциональна длине периода r в последовательности используемых алфавитов B_j.

Многоалфавитные шифры замены предложил и ввел в практику криптографии Леон Батист Альберти, который также был известным архитектором и теоретиком искусства. Его книга "Трактат о шифре", написанная в 1566 г., представляла собой первый в Европе научный труд по криптологии. Кроме шифра многоалфавитной замены, Альберти также подробно описал устройства из вращающихся колес для его реализации. Криптологи всего мира почитают Л.Альберти основоположником криптологии [32].

Система шифрования Вижинера

Система Вижинера впервые была опубликована в 1586 г. и является одной из старейших и наиболее известных многоалфавитных систем. Свое название она получила по имени французского дипломата XVI века Блеза Вижинера, который развивал и совершенствовал криптографические системы.

Система Вижинера подобна такой системе шифрования Цезаря, у которой ключ подстановки меняется от буквы к букве. Этот шифр многоалфавитной замены можно описать таблицей шифрования, называемой таблицей (квадратом) Вижинера. На рис. 2.8 и 2.9 показаны таблицы Вижинера для русского и английского алфавитов соответственно.

Таблица Вижинера используется для зашифрования и расшифрования. Таблица имеет два входа:

- верхнюю строку подчеркнутых символов, используемую для считывания очередной буквы исходного открытого текста;.
- крайний левый столбец ключа.

Последовательность ключей обычно получают из числовых значений букв ключевого слова.

При шифровании исходного сообщения его выписывают в строку, а под ним записывают ключевое слово (или фразу). Если ключ оказался короче сообщения, то его циклически повторяют. В процессе шифрования находят в верхней строке таблицы очеред-

Ключ	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я
0	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я
1	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а
2	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б
3	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в
4	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г
5	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д
6	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е
7	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж
8	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з
9	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и
10	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й
11	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к
12	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л
13	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м
14	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н
15	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о
16	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
17	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р
18	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с
19	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т
20	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у
21	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф
22	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х
23	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц
24	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч
25	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш
26	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ
27	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь
28	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы
29	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ
30	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э
31	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю

Рис. 2.8. Таблица Вижинера для русского алфавита

Ключ	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
2	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
3	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
4	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
6	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
6	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
7	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
8	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
9	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
10	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
11	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
12	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
13	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
14	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
15	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
16	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
17	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
18	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
19	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
20	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
21	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
22	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
23	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
24	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
25	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Рис. 2.9. Таблица Вижинера для английского алфавита

ную букву исходного текста и в левом столбце очередное значение ключа. Очередная буква шифртекста находится на пересечении столбца, определяемого шифруемой буквой, и строки, определяемой числовым значением ключа.

Пусть ключевая последовательность имеет длину r, тогда ключ r-алфавитной подстановки есть r-строка

$$\bar{\pi} = (\pi_0, \pi_1, \ldots, \pi_{r-1}). \qquad (2.9)$$

Система шифрования Вижинера преобразует открытый текст $\bar{x} = (x_0, x_1, \ldots, x_{n-1})$ в шифртекст $\bar{y} = (y_0, y_1, \ldots, y_{n-1})$ с помощью ключа $\bar{\pi} = (\pi_0, \pi_1, \ldots, \pi_{n-1})$ согласно правилу

$$T_\pi : \bar{x} = (x_0, x_1, \ldots, x_{n-1}) \to \bar{y} = (y_0, y_1, \ldots, y_{n-1}),$$

$$(y_0, y_1, \ldots, y_{n-1}) = (\pi_0(x_0), \pi_1(x_1), \ldots, \pi_{n-1}(x_{n-1})), \qquad (2.10)$$

где $\pi_i = \pi_{(i \bmod r)}$.

Рассмотрим пример получения шифртекста с помощью таблицы Вижинера. Пусть выбрано ключевое слово АМБРОЗИЯ. Необходимо зашифровать сообщение ПРИЛЕТАЮ СЕДЬМОГО.

Выпишем исходное сообщение в строку и запишем под ним ключевое слово с повторением. В третью строку будем выписывать буквы шифртекста, определяемые из таблицы Вижинера.

Сообщение	П	Р	И	Л	Е	Т	А	Ю	С	Е	Д	Ь	М	О	Г	О
Ключ	А	М	Б	Р	О	З	И	Я	А	М	Б	Р	О	З	И	Я
Шифртекст	П	Ъ	Й	Ы	У	Щ	И	Э	С	С	Е	К	Ь	Х	Л	Н

Шифр "двойной квадрат" Уитстона

В 1854 г. англичанин Чарльз Уитстон разработал новый метод шифрования биграммами, который называют "двойным квадратом". Свое название этот шифр получил по аналогии с полибианским квадратом. Шифр Уитстона открыл новый этап в истории развития криптографии. В отличие от полибианского шифр "двойной квадрат" использует сразу две таблицы, размещенные по одной горизонтали, а шифрование идет биграммами, как в шифре Плейфейра. Эти не столь сложные модификации привели к появлению на свет качественно новой криптографической системы ручного шифрования. Шифр "двойной квадрат" оказался очень надежным и удобным и применялся Германией даже в годы второй мировой войны.

Поясним процедуру шифрования этим шифром на примере. Пусть имеются две таблицы со случайно расположенными в них русскими алфавитами (рис.2.10). Перед шифрованием исходное сообщение разбивают на биграммы. Каждая биграмма шифруется отдельно. Первую букву биграммы находят в левой таблице, а вто-

вую букву – в правой таблице. Затем мысленно строят прямоугольник так, чтобы буквы биграммы лежали в его противоположных вершинах. Другие две вершины этого прямоугольника дают буквы биграммы шифртекста.

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

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

Рис. 2.10. Две таблицы со случайно расположенными символами русского алфавита для шифра "двойной квадрат"

Предположим, что шифруется биграмма исходного текста ИЛ. Буква И находится в столбце 1 и строке 2 левой таблицы. Буква Л находится в столбце 5 и строке 4 правой таблицы. Это означает, что прямоугольник образован строками 2 и 4, а также столбцами 1 левой таблицы и 5 правой таблицы. Следовательно, в биграмму шифртекста входят буква О, расположенная в столбце 5 и строке 2 правой таблицы, и буква В, расположенная в столбце 1 и строке 4 левой таблицы, т. е. получаем биграмму шифртекста ОВ.

Если обе буквы биграммы сообщения лежат в одной строке, то и буквы шифртекста берут из этой же строки. Первую букву биграммы шифртекста берут из левой таблицы в столбце, соответствующем второй букве биграммы сообщения. Вторая же буква биграммы шифртекста берется из правой таблицы в столбце, соответствующем первой букве биграммы сообщения. Поэтому биграмма сообщения ТО превращается в биграмму шифртекста ЖБ. Аналогичным образом шифруются все биграммы сообщения:

Сообщение	ПР	ИЛ	ЕТ	АЮ	_Ш	ЕС	ТО	ГО
Шифртекст	ПЕ	ОВ	ЩН	ФМ	ЕШ	РФ	БЖ	ДЦ

Шифрование методом "двойного квадрата" дает весьма устойчивый к вскрытию и простой в применении шифр. Взламывание шифртекста "двойного квадрата" требует больших усилий, при этом длина сообщения должна быть не менее тридцати строк.

Одноразовая система шифрования

Почти все применяемые на практике шифры характеризуются как условно надежные, поскольку они могут быть в принципе раскрыты при наличии неограниченных вычислительных возможностей. Абсолютно надежные шифры нельзя разрушить даже при ис-

пользовании неограниченных вычислительных возможностей. Существует единственный такой шифр, применяемый на практике,– одноразовая система шифрования. Характерной особенностью одноразовой системы шифрования является одноразовое использование ключевой последовательности.

Одноразовая система шифрует исходный открытый текст

$$\overline{X} = (X_0, X_1, \ldots, X_{n-1})$$

в шифртекст

$$\overline{Y} = (Y_0, Y_1, \ldots, Y_{n-1})$$

посредством подстановки Цезаря

$$Y_i = (X_i + K_i) \bmod m, \quad 0 \le i < n, \tag{2.11}$$

где K_i – i-й элемент случайной ключевой последовательности.

Ключевое пространство $\overline{K}$ одноразовой системы представляет собой набор дискретных случайных величин из $\overline{Z}_m$ и содержит m^n значений.

Процедура расшифрования описывается соотношением

$$X_i = (Y_i - K_i) \bmod m, \tag{2.12}$$

где K_i – i-й элемент той же самой случайной ключевой последовательности.

Одноразовая система изобретена в 1917 г. американцами Дж. Моборном и Г. Вернамом [113]. Для реализации этой системы подстановки иногда используют одноразовый блокнот. Этот блокнот составлен из отрывных страниц, на каждой из которых напечатана таблица со случайными числами (ключами) K_i. Блокнот выполняется в двух экземплярах: один используется отправителем, а другой – получателем. Для каждого символа X_i сообщения используется свой ключ K_i из таблицы только один раз. После того как таблица использована, она должна быть удалена из блокнота и уничтожена. Шифрование нового сообщения начинается с новой страницы.

Этот шифр абсолютно надежен, если набор ключей K_i действительно случаен и непредсказуем. Если криптоаналитик попытается использовать для заданного шифртекста все возможные наборы ключей и восстановить все возможные варианты исходного текста, то они все окажутся равновероятными. Не существует способа выбрать исходный текст, который был действительно послан. Теоретически доказано, что одноразовые системы являются нераскрываемыми системами, поскольку их шифртекст не содержит достаточной информации для восстановления открытого текста.

Казалось бы, что благодаря данному достоинству одноразовые системы следует применять во всех случаях, требующих абсолютной информационной безопасности. Однако возможности применения одноразовой системы ограничены чисто практическими аспектами. Существенным моментом является требование одноразового использования случайной ключевой последовательности. Ключевая последовательность с длиной, не меньшей длины сообщения, должна передаваться получателю сообщения заранее или отдельно по некоторому секретному каналу. Это требование не будет слишком обременительным для передачи действительно важных одноразовых сообщений, например, по горячей линии Вашингтон – Москва. Однако такое требование практически неосуществимо для современных систем обработки информации, где требуется шифровать многие миллионы символов.

В некоторых вариантах одноразового блокнота прибегают к более простому управлению ключевой последовательностью, но это приводит к некоторому снижению надежности шифра. Например, ключ определяется указанием места в книге, известной отправителю и получателю сообщения. Ключевая последовательность начинается с указанного места этой книги и используется таким же образом, как в системе Вижинера. Иногда такой шифр называют шифром с бегущим ключом. Управление ключевой последовательностью в таком варианте шифра намного проще, так как длинная ключевая последовательность может быть представлена в компактной форме. Но с другой стороны, эти ключи не будут случайными. Поэтому у криптоаналитика появляется возможность использовать информацию о частотах букв исходного естественного языка.

Шифрование методом Вернама

Система шифрования Вернама является в сущности частным случаем системы шифрования Вижинера при значении модуля $m = 2$. Конкретная версия этого шифра, предложенная в 1926 г. Гилбертом Вернамом, сотрудником фирмы AT&T США, использует двоичное представление символов исходного текста.

Каждый символ исходного открытого текста из английского алфавита {A, B, C, D, ..., Z}, расширенного шестью вспомогательными символами (пробел, возврат каретки и т. п.), сначала кодировался в 5-битовый блок (b_0, b_1, ..., b_4) телеграфного кода Бодо.

Случайная последовательность двоичных ключей $k_0, k_1, k_2, \ldots$ заранее записывалась на бумажной ленте.

Схема передачи сообщений с использованием шифрования методом Вернама показана на рис. 2.11. Шифрование исходного текста, предварительно преобразованного в последовательность двоичных символов x, осуществлялось путем сложения по модулю 2 символов x с последовательностью двоичных ключей k.

Символы шифртекста

$$y = x \oplus k. \qquad (2.13)$$

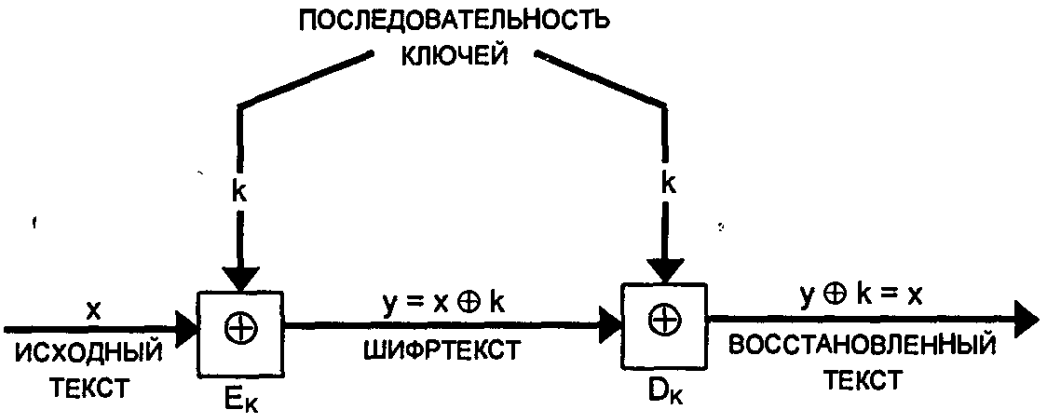


Рис. 2.11. Схема шифрования и расшифрования сообщений по методу Вернама

Расшифрование состоит в сложении по модулю 2 символов y шифртекста с той же последовательностью ключей k:

$$y \oplus k = x \oplus k \oplus k = x. \qquad (2.14)$$

При этом последовательности ключей, использованные при шифровании и расшифровании, компенсируют друг друга (при сложении по модулю 2), и в результате восстанавливаются символы x исходного текста.

При разработке своей системы Вернам проверял ее с помощью закольцованных лент, установленных на передатчике и приемнике для того, чтобы использовалась одна и та же последовательность ключей.

Следует отметить, что метод Вернама не зависит от длины последовательности ключей и, кроме того, он позволяет использовать случайную последовательность ключей. Однако при реализации метода Вернама возникают серьезные проблемы, связанные с необходимостью доставки получателю такой же последовательности ключей, как у отправителя, либо с необходимостью безопасного хранения идентичных последовательностей ключей у отправителя и получателя. Эти недостатки системы шифрования Вернама преодолены при шифровании методом гаммирования.

2.5. Шифрование методом гаммирования

Под *гаммированием* понимают процесс наложения по определенному закону гаммы шифра на открытые данные. *Гамма шифра* – это псевдослучайная последовательность, выработанная по заданному алгоритму для зашифрования открытых данных и расшифрования зашифрованных данных.

Процесс зашифрования заключается в генерации гаммы шифра и наложении полученной гаммы на исходный открытый текст обратимым образом, например с использованием операции сложения по модулю 2.

Следует отметить, что перед зашифрованием открытые данные разбивают на блоки $T_0^{(i)}$ одинаковой длины, обычно по 64 бита. Гамма шифра вырабатывается в виде последовательности блоков $Г_ш^{(i)}$ аналогичной длины.

Уравнение зашифрования можно записать в виде

$$T_ш^{(i)} = Г_ш^{(i)} \oplus T_0^{(i)}, \quad i = 1 \ldots M, \qquad (2.15)$$

где $T_ш^{(i)}$ – i-й блок шифртекста; $Г_ш^{(i)}$ – i-й блок гаммы шифра; $T_0^{(i)}$ – i-й блок открытого текста; M – количество блоков открытого текста.

Процесс расшифрования сводится к повторной генерации гаммы шифра и наложению этой гаммы на зашифрованные данные. Уравнение расшифрования имеет вид

$$T_0^{(i)} = Г_ш^{(i)} \oplus T_ш^{(i)}. \qquad (2.16)$$

Получаемый этим методом шифртекст достаточно труден для раскрытия, поскольку теперь ключ является переменным. По сути дела гамма шифра должна изменяться случайным образом для каждого шифруемого блока. Если период гаммы превышает длину всего шифруемого текста и злоумышленнику неизвестна никакая часть исходного текста, то такой шифр можно раскрыть только прямым перебором всех вариантов ключа. В этом случае криптостойкость шифра определяется длиной ключа.

Методы генерации псевдослучайных последовательностей чисел

При шифровании методом гаммирования в качестве ключа используется случайная строка битов, которая объединяется с открытым текстом, также представленным в двоичном виде (например, A = 00000, B = 00001, C = 00010 и т.д.), с помощью побитового сложения по модулю 2, и в результате получается шифрованный текст. Генерирование непредсказуемых двоичных последовательностей большой длины является одной из важных проблем классической криптографии. Для решения этой проблемы широко используются генераторы двоичных псевдослучайных последовательностей.

Генерируемые псевдослучайные ряды чисел часто называют гаммой шифра или просто гаммой (по названию буквы γ греческого алфавита, часто используемой в математических формулах для обозначения случайных величин).

Обычно для генерации последовательности псевдослучайных чисел применяют компьютерные программы, которые, хотя и называются генераторами случайных чисел, на самом деле выдают детерминированные числовые последовательности, которые по своим свойствам очень похожи на случайные [32].

К криптографически стойкому генератору псевдослучайной последовательности чисел (гаммы шифра) предъявляются три основных требования:

- период гаммы должен быть достаточно большим для шифрования сообщений различной длины;
- гамма должна быть практически непредсказуемой, что означает невозможность предсказать следующий бит гаммы, даже если известны тип генератора и предшествующий кусок гаммы;
- генерирование гаммы не должно вызывать больших технических сложностей.

Длина периода гаммы является самой важной характеристикой генератора псевдослучайных чисел. По окончании периода числа начнут повторяться, и их можно будет предсказать. Требуемая длина периода гаммы определяется степенью закрытости данных. Чем длиннее ключ, тем труднее его подобрать. Длина периода гаммы зависит от выбранного алгоритма получения псевдослучайных чисел.

Второе требование связано со следующей проблемой: как можно достоверно убедиться, что псевдослучайная гамма конкретного генератора является действительно непредсказуемой? Пока не существуют такие универсальные и практически проверяемые критерии и методики. Чтобы гамма считалась непредсказуемой, т.е. истинно случайной, необходимо, чтобы ее период был очень большим, а различные комбинации битов определенной длины были равномерно распределены по всей ее длине.

Третье требование обусловливает возможность практической реализации генератора программным или аппаратным путем с обеспечением необходимого быстродействия.

Один из первых способов генерации псевдослучайных чисел на ЭВМ предложил в 1946 г. Джон фон Нейман. Суть этого способа состоит в том, что каждое последующее случайное число образуется возведением в квадрат предыдущего числа с отбрасыванием цифр младших и старших разрядов. Однако этот способ оказался ненадежным и от него вскоре отказались.

Из известных процедур генерации последовательности псевдослучайных целых чисел наиболее часто применяется так называемый линейный конгруэнтный генератор. Этот генератор вырабатывает последовательность псевдослучайных чисел $Y_1, Y_2, \ldots, Y_{i-1}, Y_i, \ldots$, используя соотношение

$$Y_i = (a * Y_{i-1} + b) \bmod m, \qquad (2.17)$$

где Y_i – i-е (текущее) число последовательности; Y_{i-1} – предыдущее число последовательности; a, b и m – константы; m – модуль; a – множитель (коэффициент); b – приращение; Y_0 – порождающее число (исходное значение).

Текущее псевдослучайное число Y_i получают из предыдущего числа Y_{i-1} умножением его на коэффициент a, сложением с приращением b и вычислением остатка от деления на модуль m. Данное уравнение генерирует псевдослучайные числа с периодом повторения, который зависит от выбираемых значений параметров a, b и m и может достигать значения m. Значение модуля m берется равным 2^n либо равным простому числу, например $m = 2^{31} - 1$. Приращение b должно быть взаимно простым с m, коэффициент a должен быть нечетным числом.

Конгруэнтные генераторы, работающие по алгоритму, предложенному Национальным бюро стандартов США, используются, в частности, в системах программирования. Эти генераторы имеют длину периода 2^{24} и обладают хорошими статистическими свойствами. Однако такая длина периода мала для криптографических применений. Кроме того, доказано, что последовательности, генерируемые конгруэнтными генераторами, не являются криптографически стойкими.

Существует способ генерации последовательностей псевдослучайных чисел на основе линейных рекуррентных соотношений [69].

Рассмотрим рекуррентные соотношения и их разностные уравнения:

$$\sum_{j=0}^{k} h_j a_{i+j} = 0,$$

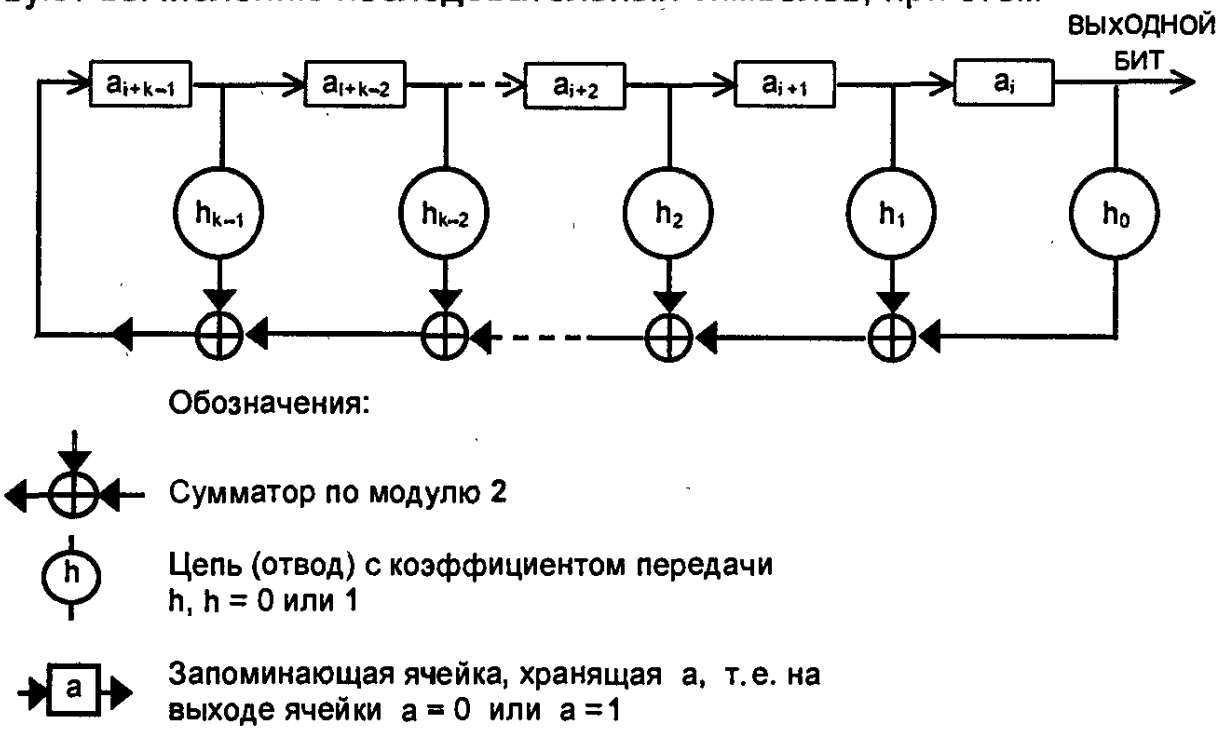
$$a_{i+k} = -\sum_{j=0}^{k-1} h_j a_{i+j}, \qquad (2.18)$$

где $h_0 \neq 0$, $h_k = 1$ и каждое h_i принадлежит полю GF(q).

Решением этих уравнений является последовательность элементов $a_0, a_1, a_2, \ldots$ поля GF(q) (см. Приложение). Соотношение (2.18) определяет правило вычисления a_k по известным значениям величин $a_0, a_1, a_2, \ldots, a_{k-1}$. Затем по известным значениям $a_0, a_1, a_2, \ldots, a_k$ находят a_{k+1} и т.д. В результате по начальным значениям

$a_0, a_1, a_2, \ldots, a_{k-1}$ можно построить бесконечную последовательность, причем каждый ее последующий член определяется из k предыдущих. Последовательности такого вида легко реализуются на компьютере, при этом реализация получается особенно простой, если все h_i и a_i принимают значения 0 и 1 из поля GF(2).

На рис. 2.12 показана линейная последовательная переключательная схема, которая может быть использована для вычисления суммы (2.18) и, следовательно, для вычисления значения a_k по значениям k предыдущих членов последовательности. Исходные величины $a_0, a_1, a_2, \ldots, a_{k-1}$ помещаются в разряды сдвигового регистра, последовательные сдвиги содержимого которого соответствуют вычислению последовательных символов, при этом



Обозначения:

Сумматор по модулю 2

Цепь (отвод) с коэффициентом передачи h, h = 0 или 1

Запоминающая ячейка, хранящая a, т.е. на выходе ячейки a = 0 или a = 1

Рис. 2.12. Генератор с регистром сдвига

выход после i-го сдвига равен a_i. Данное устройство называют генератором последовательности чисел, построенным на базе сдвигового регистра с линейной обратной связью.

Решения линейных рекуррентных соотношений, реализуемые генератором с регистром сдвига, описываются следующей теоремой. Пусть многочлен

$$h(X) = \sum_{j=0}^{k} h_j X^j,$$

где X – формальная переменная; h_j – коэффициент при X^j, принимающий значение 0 или 1; $h_0 \neq 0$, $h_k = 1$, и пусть n – наименьшее целое положительное число, для которого многочлен $X^n - 1$ делится на h(X). Кроме того, многочлен

$$g(X) = (X^n - 1)/h(X).$$

Тогда решения рекуррентных соотношений

$$\sum_{j=0}^{k} h_j a_{i+j} = 0$$

в виде последовательности элементов $a_0, a_1, a_i, \ldots, a_{n-1}$ периодичны с периодом n и совокупность, составленная из первых периодов всех возможных решений, рассматриваемых как многочлены по модулю $(X^n - 1)$, т.е.

$$a(X) = a_0 * X^{n-1} + a_1 * X^{n-2} + \ldots + a_{n-2} * X + a_{n-1},$$

совпадает с идеалом, порожденным многочленом g(X) в алгебре многочленов по модулю $(X^n - 1)$. Доказательство этой теоремы можно найти в [69].

Заметим, что если при таком определении многочлена a(X) элементы $a_0, a_1, a_2, \ldots$ вычисляются в порядке возрастания номеров, то коэффициенты многочлена a(X) вычисляются, начиная с коэффициентов при степенях высших порядков. Следует также отметить, что вид многочлена $h(X) = \sum_{j=0}^{k} h_j X^j$ определяет конфигурацию обратных связей (отводов) h_j в генераторе со сдвиговым регистром. Другими словами, если у многочлена h(X) коэффициент $h_j = 1$, это означает, что отвод h_j в схеме генератора присутствует, если же у многочлена h(X) коэффициент $h_j = 0$, то отвод h_j в схеме генератора отсутствует. В [58] показано, что в качестве h(X) необходимо выбирать неприводимый примитивный многочлен (см. также приложение). При таком выборе многочлена h(X) со старшей степенью m генератор обеспечивает выдачу псевдослучайной последовательности двоичных чисел с максимально возможным периодом $2^m - 1$.

Рассмотрим в качестве примера трехразрядный сдвиговый регистр с линейной обратной связью (рис.2.13), построенный в соответствии с неприводимым примитивным многочленом

$$h(X) = X^3 + X^2 + 1,$$

где коэффициенты $h_3 = 1$, $h_2 = 1$, $h_1 = 0$, $h_0 = 1$.

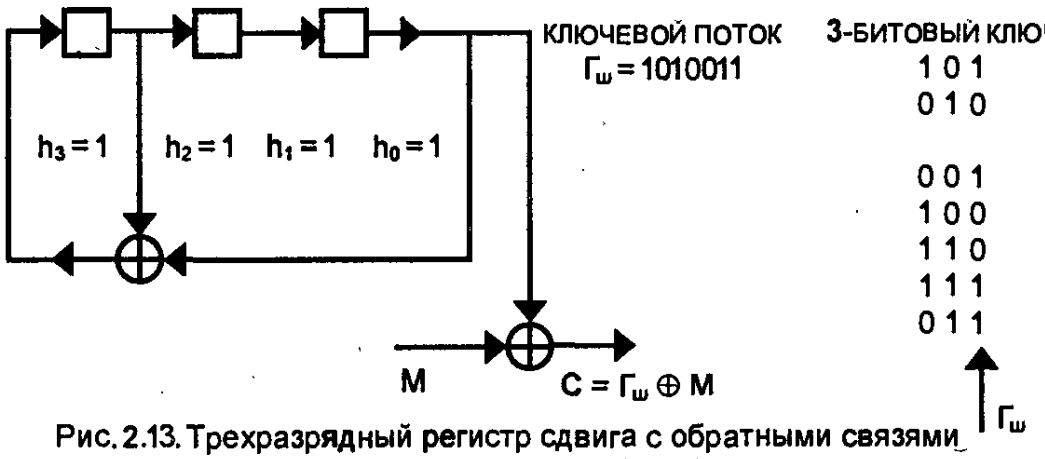


Рис. 2.13. Трехразрядный регистр сдвига с обратными связями (генератор гаммы шифра $Г_ш$)

Пусть ключом является 101. Регистр начинает работать с этого состояния; последовательность состояний регистра приведена на рис. 2.13. Регистр проходит через все семь ненулевых состояний и снова возвращается в свое исходное состояние 101. Это – наиболее длинный период данного регистра с линейной обратной связью. Такая последовательность называется *последовательностью максимальной длины* для сдвигового регистра (Maximal Lenght Shift Register Sequence – MLSRS). Питерсон и Уэлдон [69] показали, что при любом целом m существует m-битовая последовательность MLSRS с периодом $2^m - 1$. В частности, при $m = 100$ последовательность будет иметь период $2^{100} - 1$ и не повторится 10^{16} лет при передаче ее по линии связи со скоростью 1 Мбит/с.

В нашем примере выходной последовательностью (гаммой шифра) $Г_ш$ сдвигового регистра с обратной связью является последовательность 1010011, которая циклически повторяется. В этой последовательности имеется четыре единицы и три нуля, и их распределение настолько близко к равномерному, насколько это возможно в последовательности, имеющей длину 7. Если рассмотреть пары последовательных битов, то пары 10 и 01 появляются по два раза, а пары 00 и 11 – один раз, что опять оказывается настолько близким к равномерному распределению, насколько это возможно. В случае последовательности максимальной длины для m-разрядного регистра это свойство равнораспределенности распространяется на тройки, четверки и т.д. битов, вплоть до m-битовых групп. Благодаря такой близости к равномерному распределению последовательности максимальной длины часто используются в качестве псевдослучайных последовательностей в криптографических системах, которые имитируют работу криптостойкой системы одноразового шифрования.

Хотя такая криптографическая система осуществляет имитацию заведомо криптостойкой системы одноразового шифрования, сама она не отличается стойкостью и может быть раскрыта за несколько секунд работы компьютера при условии наличия известного открытого текста [29].

Если отводы регистра с обратной связью зафиксированы, то для нахождения начального состояния регистра достаточно знать m битов открытого текста. Чтобы найти m битов ключевого потока, m битов известного открытого текста складывают по модулю 2 с соответствующими m битами шифртекста. Полученные m битов дают состояние сдвигового регистра с обратной связью в обратном направлении на некоторый момент времени. Затем, моделируя работу регистра назад, можно определить его исходное состояние.

Если отводы регистра с обратной связью не фиксированы, а являются частью ключа, то достаточно 2m битов известного открытого текста, чтобы сравнительно быстро определить расположение отводов регистра и его начальное состояние. Пусть S(i) – вектор-столбец, состоящий из m символов 0 и 1, который определяет состояние регистра в i-й момент времени. Тогда

$$S(i+1) = A * S(i) \bmod 2,$$

где A – матрица размером m×m, определяющая положение отводов регистра с обратной связью.

Для трехразрядного регистра (рис. 2.13)

$$A = \begin{vmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{vmatrix}.$$

Матрица A всегда имеет следующую структуру: в ее первой строке отражена последовательность отводов в регистре, непосредственно под главной диагональю располагаются единицы, а в остальных позициях располагаются нули.

2m битов известного открытого текста позволяют вычислить 2m последовательных битов ключевого потока. Для упрощения обозначений предположим, что это – первые 2m битов ключевого потока. Следовательно,

- S(1) – первая группа m известных битов ключевого потока;
- S(2) – следующая группа (начиная с номера 2) из m известных битов ключевого потока;
- S(m+1) – последняя группа из m известных битов ключевого потока.

Далее можно образовать две матрицы размером m×m:

$$X(1) = [S(1), S(2), \ldots, S(m)],$$

$$X(2) = [S(2), S(3), \ldots, S(m+1)],$$

которые связаны соотношением

$$X(2) = A * X(1) \bmod 2.$$

Можно показать, что для любой последовательности максимальной длины матрица X(1) всегда несингулярна, поэтому матрицу A можно вычислить как

$$A = X(2)\,[X(1)]^{-1} \bmod 2.$$

Обращение матрицы X(1) требует (самое большее) порядка m^3 операций, поэтому легко выполняется при любом разумном значении m.

Для криптографии последовательности максимальной длины MLSRS можно сделать более криптостойкими, используя нелинейную логику. В частности, предложен вариант [29], в котором в качестве ключевого потока используется нелинейно "фильтрованное" содержимое сдвигового регистра, а для получения последовательности максимальной длины – линейная обратная связь, как показано на рис. 2.14.

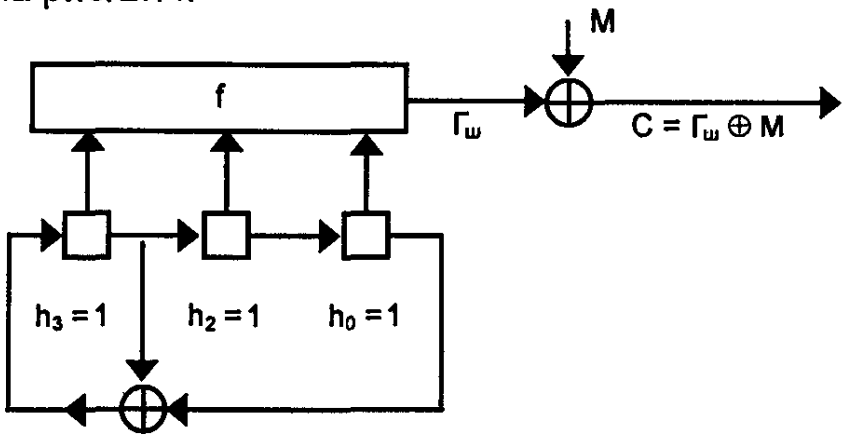


Рис. 2.14. Линейный сдвиговый регистр с нелинейными логическими цепями на выходе

Функция f должна выбираться так, чтобы обеспечить хороший баланс между нулями и единицами, а фильтрованная последовательность имела распределение, близкое к равномерному. Необходимо также, чтобы фильтрованная последовательность имела большой период. Если $(2^m - 1)$ является простым числом (как в примере: при $m = 3$ имеем $2^3 - 1 = 7$), то фильтрованная последовательность может иметь период $(2^m - 1)$ (при выборе структуры сдвигового регистра в соответствии с неприводимым примитивным многочленом $h(X)$ степени m).

К таким значениям m относятся, в частности, следующие:

2, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127, 521, 607, 1279, 2203, 2281, а полученные таким образом простые числа называются *простыми числами Мерсенна*.

Несмотря на то, что фильтрованную выходную последовательность обычно нельзя получить с помощью m-разрядного сдвигового регистра с линейной обратной связью, ее всегда можно получить с помощью сдвигового регистра большей длины с линейной обратной связью [59]. Регистр длиной $(2^m - 1)$ всегда позволит это сделать, а иногда пригоден и более короткий регистр.

Еще более привлекательно использование в цепи обратной связи нелинейной логики, однако теория таких схем недостаточно хорошо освещена (в открытой литературе).

ГЛАВА 3. СОВРЕМЕННЫЕ СИММЕТРИЧНЫЕ КРИПТОСИСТЕМЫ

По мнению К. Шеннона [101], в практических шифрах необходимо использовать два общих принципа: рассеивание и перемешивание.

Рассеивание представляет собой распространение влияния одного знака открытого текста на много знаков шифртекста, что позволяет скрыть статистические свойства открытого текста.

Перемешивание предполагает использование таких шифрующих преобразований, которые усложняют восстановление взаимосвязи статистических свойств открытого и шифрованного текстов. Однако шифр должен не только затруднять раскрытие, но и обеспечивать легкость зашифрования и расшифрования при известном пользователю секретном ключе.

Распространенным способом достижения эффектов рассеивания и перемешивания является использование *составного шифра*, т.е. такого шифра, который может быть реализован в виде некоторой последовательности простых шифров, каждый из которых вносит свой вклад в значительное суммарное рассеивание и перемешивание.

В составных шифрах в качестве простых шифров чаще всего используются простые перестановки и подстановки. При *перестановке* просто перемешивают символы открытого текста, причем конкретный вид перемешивания определяется секретным ключом. При *подстановке* каждый символ открытого текста заменяют другим символом из того же алфавита, а конкретный вид подстановки также определяется секретным ключом. Следует заметить, что в современном блочном шифре блоки открытого текста и шифртекста представляют собой двоичные последовательности обычно длиной 64 бита. В принципе каждый блок может принимать 2^{64} значений. Поэтому подстановки выполняются в очень большом алфавите, содержащем до $2^{64} \approx 10^{19}$ "символов".

При многократном чередовании простых перестановок и подстановок, управляемых достаточно длинным секретным ключом, можно получить очень стойкий шифр с хорошим рассеиванием и перемешиванием. Рассмотренные ниже криптоалгоритмы DES, IDEA и отечественный стандарт шифрования данных построены в полном соответствии с указанной методологией.

3.1. Американский стандарт шифрования данных DES

Стандарт шифрования данных DES (Data Encryption Standard) опубликован в 1977 г. Национальным бюро стандартов США. Стандарт DES предназначен для защиты от несанкционированного доступа к важной, но несекретной информации в государственных и коммерческих организациях США. Алгоритм, положенный в основу стандарта, распространялся достаточно быстро, и уже в 1980 г. был одобрен Национальным институтом стандартов и технологий США (НИСТ). С этого момента DES превращается в стандарт не только по названию (Data Encryption Standard), но и фактически. Появляются программное обеспечение и специализированные микроЭВМ, предназначенные для шифрования и расшифрования информации в сетях передачи данных.

К настоящему времени DES является наиболее распространенным алгоритмом, используемым в системах защиты коммерческой информации. Более того, реализация алгоритма DES в таких системах становится признаком хорошего тона.

Основные достоинства алгоритма DES:

- используется только один ключ длиной 56 бит;
- зашифровав сообщение с помощью одного пакета программ, для расшифровки можно использовать любой другой пакет программ. соответствующий стандарту DES;
- относительная простота алгоритма обеспечивает высокую скорость обработки;
- достаточно высокая стойкость алгоритма.

Первоначально метод, лежащий в основе стандарта DES, был разработан фирмой IBM для своих целей и реализован в виде системы "Люцифер". Система "Люцифер" основана на комбинировании методов подстановки и перестановки и состоит из чередующейся последовательности блоков перестановки и подстановки. В ней использовался ключ длиной 128 бит, управлявший состояниями

блоков перестановки и подстановки. Система "Люцифер" оказалась весьма сложной для практической реализации из-за относительно малой скорости шифрования (2190 байт/с – программная реализация, 96970 байт/с – аппаратная реализация).

Алгоритм DES также использует комбинацию подстановок и перестановок. DES осуществляет шифрование 64-битовых блоков данных с помощью 64-битового ключа, в котором значащими являются 56 бит (остальные 8 бит – проверочные биты для контроля на четность). Дешифрование в DES является операцией, обратной шифрованию, и выполняется путем повторения операций шифрования в обратной последовательности. Обобщенная схема процесса шифрования в алгоритме DES показана на рис.3.1. Процесс шифрования заключается в начальной перестановке битов 64-битового блока, шестнадцати циклах шифрования и, наконец, в конечной перестановке битов.

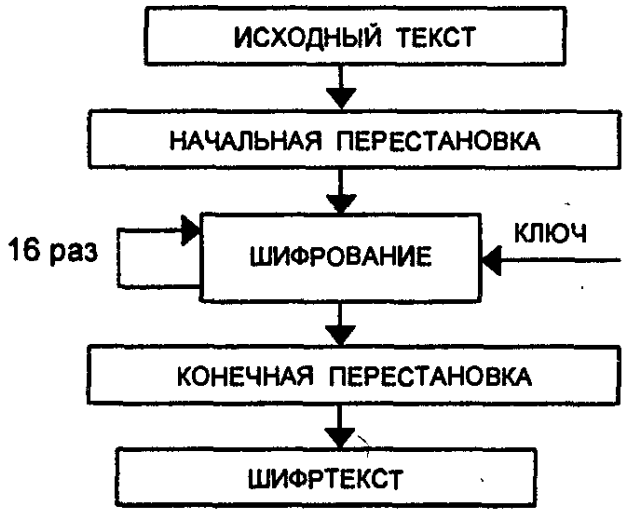


Рис. 3.1. Обобщенная схема шифроввния в алгоритме DES

Следует сразу отметить, что все приводимые таблицы являются стандартными и должны включаться в реализацию алгоритма DES в неизменном виде.

Все перестановки и коды в таблицах подобраны разработчиками таким образом, чтобы максимально затруднить процесс расшифровки путем подбора ключа. При описании алгоритма DES (рис. 3.2) применены следующие обозначения:

L и R – последовательности битов (левая (left) и правая (right));

LR – конкатенация последовательностей L и R, т.е. такая последовательность битов, длина которой равна сумме длин L и R; в последовательности LR биты последовательности R следуют за битами последовательности L;

⊕ – операция побитового сложения по модулю 2.

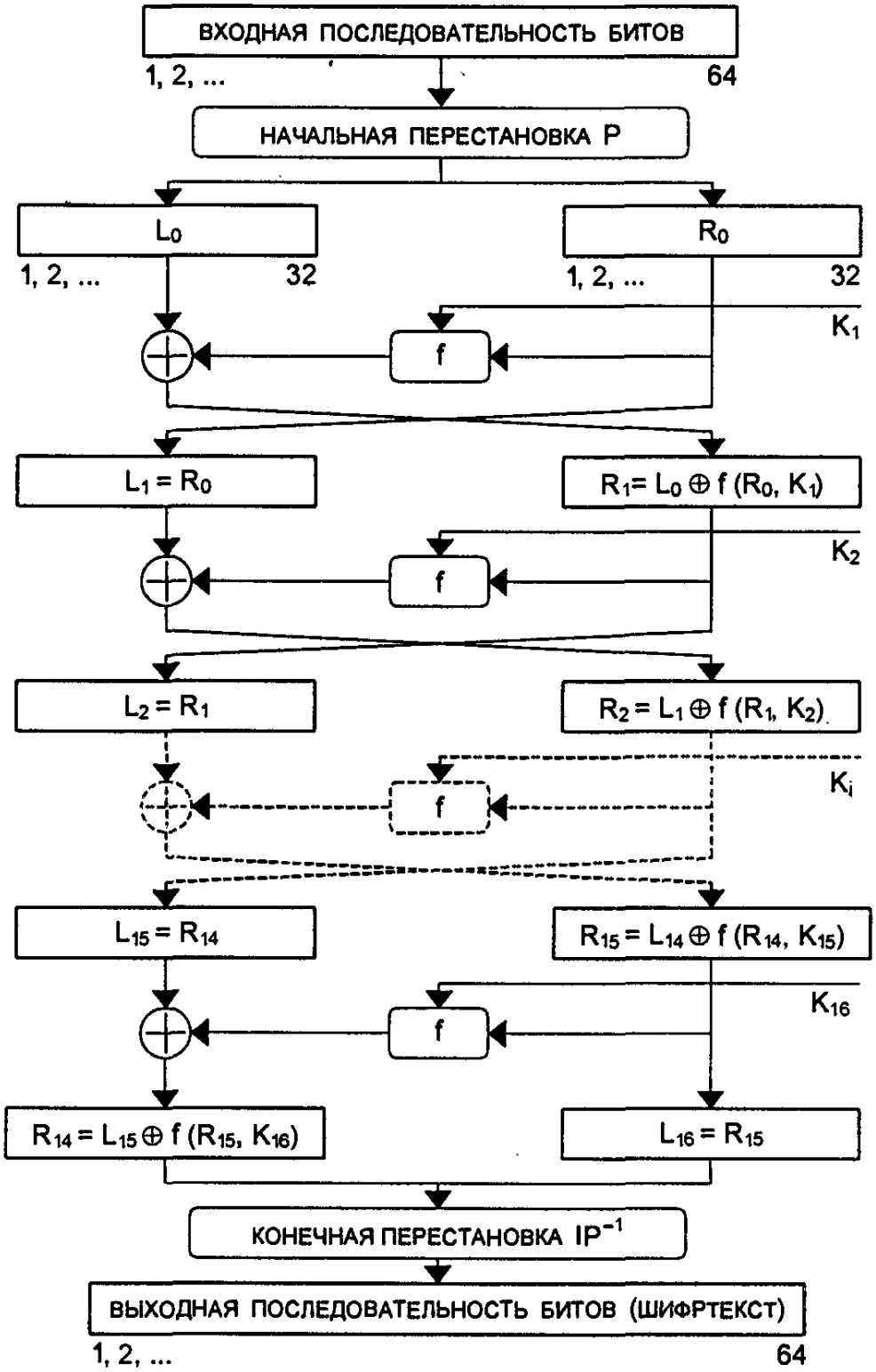


Рис. 3.2. Структура алгоритма DES

Пусть из файла исходного текста считан очередной 64-битовый (8-байтовый) блок T. Этот блок T преобразуется с помощью *матрицы начальной перестановки* IP (табл. 3.1).

Биты входного блока T (64 бита) переставляются в соответствии с матрицей IP: бит 58 входного блока T становится битом 1, бит 50 – битом 2 и т.д. Эту перестановку можно описать выражением $T_0 = IP(T)$. Полученная последовательность битов T_0 разделяется на две последовательности: L_0 – левые или старшие биты, R_0 – правые или младшие биты, каждая из которых содержит 32 бита.

Таблица 3.1

Матрица начальной перестановки 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

Таблица 3.2

Матрицв обратной перестановки IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Затем выполняется итеративный процесс шифрования, состоящий из 16 шагов (циклов). Пусть T_i – результат i-й итерации:

$$T_i = L_i R_i,$$

где $L_i = t_1 t_2 \ldots t_{32}$ (первые 32 бита); $R_i = t_{33} t_{34} \ldots t_{64}$ (последние 32 бита). Тогда результат i-й итерации описывается следующими формулами:

$$L_i = R_{i-1}, \quad i = 1, 2, \ldots, 16;$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i), \quad i = 1, 2, \ldots, 16.$$

Функция f называется функцией шифрования. Ее аргументами являются последовательность R_{i-1}, получаемая на предыдущем шаге итерации, и 48-битовый ключ K_i, который является результатом преобразования 64-битового ключа шифра K. (Подробнее функция шифрования f и алгоритм получения ключа K_i описаны ниже.)

На последнем шаге итерации получают последовательности R_{16} и L_{16} (без перестановки местами), которые конкатенируются в 64-битовую последовательность $R_{16}L_{16}$.

По окончании шифрования осуществляется восстановление позиций битов с помощью матрицы обратной перестановки IP^{-1} (табл. 3.2).

Пример того, как соотносятся элементы первой строки матрицы IP^{-1} с элементами матрицы IP приведен в табл. 3.3.

Таблица 3.3

Связь элементов матриц

Элемент матрицы IP^{-1}	Элемент матрицы IP
40	01
8	02
48	03
16	04
56	05
...	...

Процесс расшифрования данных является инверсным по отношению к процессу шифрования. Все действия должны быть выполнены в обратном порядке. Это означает, что расшифровываемые данные сначала переставляются в соответствии с матрицей IP^{-1}, а затем над последовательностью битов $R_{16}L_{16}$ выполняются те же действия, что и в процессе шифрования, но в обратном порядке.

Итеративный процесс расшифрования может быть описан следующими формулами:

$$R_{i-1} = L_i, \quad i = 1, 2, ..., 16;$$

$$L_{i-1} = R_i \oplus f(L_i, K_i), \quad i = 1, 2, ..., 16.$$

Таким образом, для процесса расшифрования с переставленным входным блоком $R_{16}L_{16}$ на первой итерации используется ключ K_{16}, на второй итерации – K_{15} и т.д. На 16-й итерации используется ключ K_1. На последнем шаге итерации будут получены последовательности L_0 и R_0, которые конкатенируются в 64-битовую последовательность L_0R_0. Затем в этой последовательности 64 бита переставляются в соответствии с матрицей IP. Результат такого преобразования – исходная последовательность битов (расшифрованное 64-битовое значение).

Теперь рассмотрим, что скрывается под преобразованием, обозначенным буквой f. Схема вычисления функции шифрования $f(R_{i-1},K_i)$ показана на рис. 3.3.

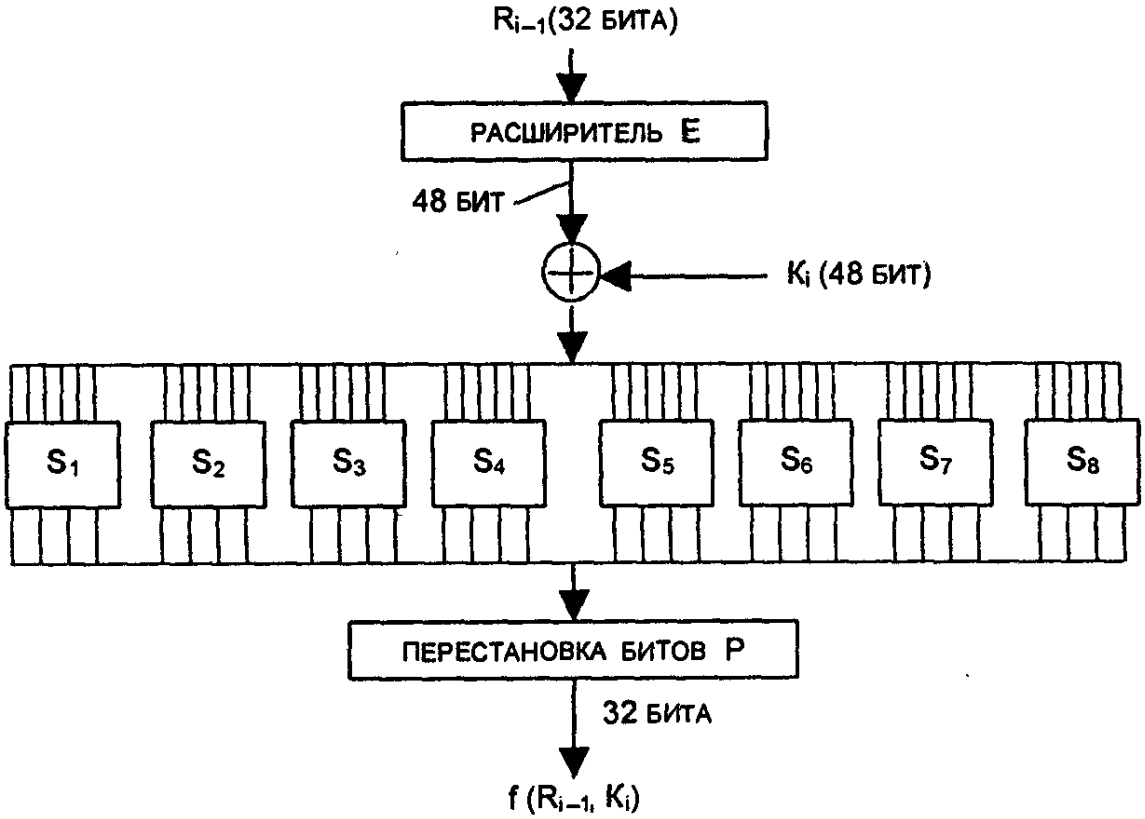


Рис. 3.3. Схема вычисления функции шифрования f

Для вычисления значения функции f используются:

- функция E (расширение 32 бит до 48);
- функция $S_1, S_2, ..., S_8$ (преобразование 6-битового числа в 4-битовое);
- функция P (перестановка битов в 32-битовой последовательности).

Приведем определения этих функций.

Аргументами функции шифрования f являются R_{i-1} (32 бита) и K_i (48 бит). Результат функции $E(R_{i-1})$ есть 48-битовое число. Функция расширения E, выполняющая расширение 32 бит до 48 (принимает блок из 32 бит и порождает блок из 48 бит), определяется табл. 3.4.

В соответствии с табл. 3.4 первые три бита $E(R_{i-1})$ – это биты 32, 1 и 2, а последние – 31, 32, 1. Полученный результат (обозначим его $E(R_{i-1})$) складывается по модулю 2 (операция XOR) с текущим значением ключа K_i и затем разбивается на восемь 6-битовых блоков $B_1, B_2, ..., B_8$:

Таблица 3.4

Функция расширения E

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

$$E(R_{i-1}) \oplus K_i = B_1B_2 ... B_8.$$

Далее каждый из этих блоков используется как номер элемента в функциях-матрицах $S_1, S_2, ..., S_8$, содержащих 4-битовые значения (табл. 3.5).

Следует отметить, что выбор элемента в матрице S_j осуществляется достаточно оригинальным образом. Пусть на вход матрицы S_j поступает 6-битовый блок $B_j = b_1b_2b_3b_4b_5b_6$, тогда двухбитовое число b_1b_6 указывает номер строки матрицы, а четырехбитовое число $b_2b_3b_4b_5$ – номер столбца. Например, если на вход матрицы S_1 поступает 6-битовый блок $B_1 = b_1b_2b_3b_4b_5b_6 = 100110$, то 2-битовое число $b_1b_6 = 10_{(2)} = 2_{(10)}$ указывает строку с номером 2 матрицы S_1, а 4-битовое число $b_2b_3b_4b_5 = 0011_{(2)} = 3_{(10)}$ указывает столбец с номером 3 матрицы S_1. Это означает, что в матрице S_1 блок $B_1 = 100110$ выбирает элемент на пересечении строки с номером 2 и столбца с номером 3, т.е. элемент $8_{(10)} = 1000_{(2)}$. Совокупность 6-битовых блоков $B_1, B_2, ..., B_8$ обеспечивает выбор четырехбитового элемента в каждой из матриц $S_1, S_2, ..., S_8$.

Таблица 3.5

Функции преобразования S_1, S_2, ..., S_8

		Номер столбца																
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Номер строки	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	S_1
	2	4	1	4	8	13	6	2	11	15	12	9	7	3	10	5	0	
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	S_2
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	S_3
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	S_4
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	S_5
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	S_6
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	1	6	
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	
	0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	S_7
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
	0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	S_8
	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	

Таблица 3.6

Функция P перестановки битов

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

Таблица 3.7

Функция G первоначальной подготовки ключа (переставленная выборка 1)

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

В результате получаем $S_1(B_1)\,S_2(B_2)\,S_3(B_3)\ldots S_8(B_8)$, т.е. 32-битовый блок (поскольку матрицы S_j содержат 4-битовые элементы). Этот 32-битовый блок преобразуется с помощью функции перестановки битов P (табл. 3.6).

Таким образом, функция шифрования

$$f(R_{i-1}, K_i) = P(S_1(B_1), \ldots, S_8(B_8)).$$

Как нетрудно заметить, на каждой итерации используется новое значение ключа K_i (длиной 48 бит). Новое значение ключа K_i вычисляется из начального ключа K (рис. 3.4). Ключ K представляет собой 64-битовый блок с 8 битами контроля по четности, расположенными в позициях 8, 16, 24, 32, 40, 48, 56, 64. Для удаления контрольных битов и подготовки ключа к работе используется функция G первоначальной подготовки ключа (табл. 3.7).

Табл. 3.7 разделена на две части. Результат преобразования G(K) разбивается на две половины C_0 и D_0 по 28 бит каждая. Первые четыре строки матрицы G определяют, как выбираются биты последовательности C_0 (первым битом C_0 будет бит 57 ключа шифра, затем бит 49 и т.д., а последними битами – биты 44 и 36 ключа).

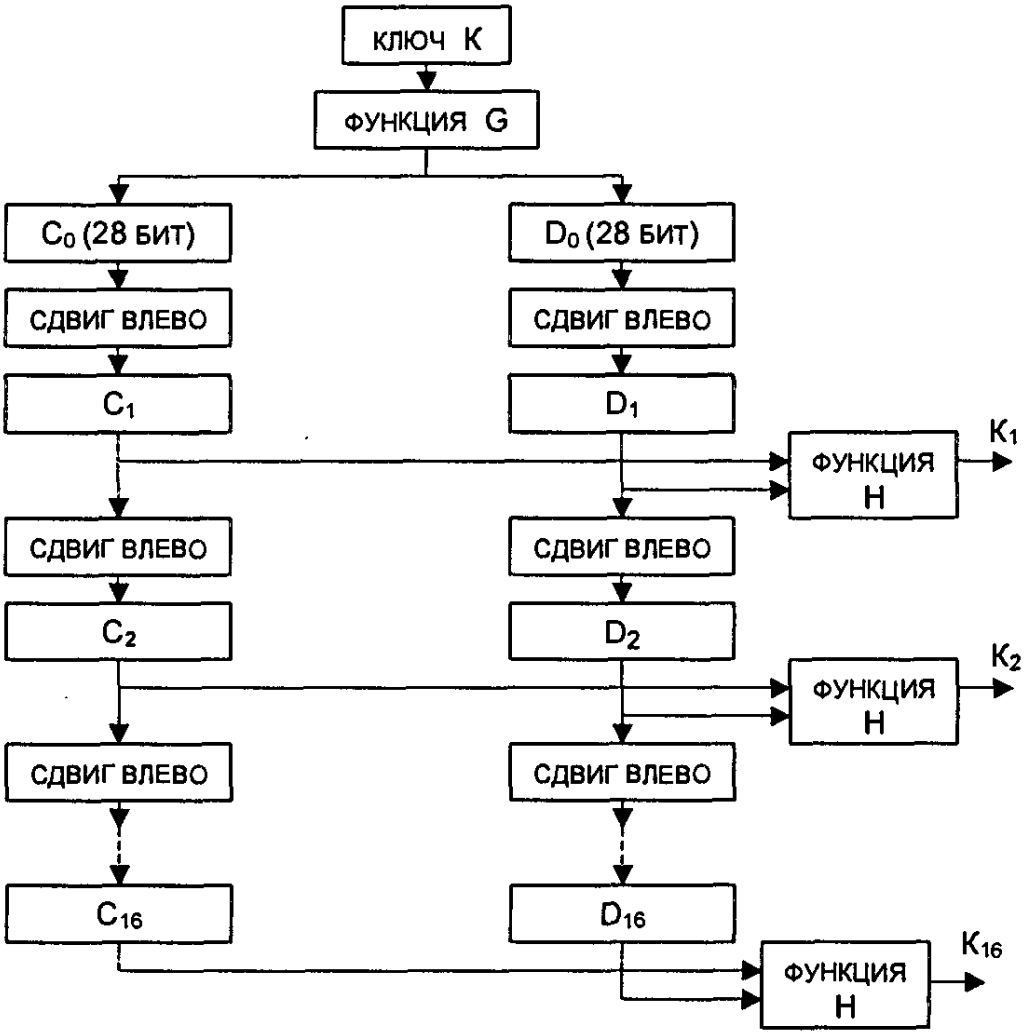


Рис. 3.4. Схема алгоритма вычисления ключей K_i

Следующие четыре строки матрицы G определяют, как выбираются биты последовательности D_0 (т.е. последовательность D_0 будет состоять из битов 63, 55, 47, ..., 12, 4 ключа шифра).

Как видно из табл. 3.7, для генерации последовательностей C_0 и D_0 не используются биты 8, 16, 24, 32, 40, 48, 56 и 64 ключа шифра. Эти биты не влияют на шифрование и могут служить для других целей (например, для контроля по четности). Таким образом, в действительности ключ шифра является 56-битовым.

После определения C_0 и D_0 рекурсивно определяются C_i и D_i, $i = 1, 2, ..., 16$. Для этого применяются операции циклического сдвига влево на один или два бита в зависимости от номера шага итерации, как показано в табл. 3.8.

Операции сдвига выполняются для последовательностей C_i и D_i независимо. Например, последовательность C_3 получается посредством циклического сдвига влево на две позиции последовательности C_2, а последовательность D_3 – посредством сдвига влево на две позиции последовательности D_2, C_{16} и D_{16} получаются из C_{15} и D_{15} посредством сдвига влево на одну позицию.

Таблица 3.8

Таблица сдвигов s_i для вычисления ключа

Номер итерации	Количество s_i сдвигов влево, бит	Номер итврации	Количество s_i сдвигов влеео, бит
1	1	9	1
2	1	10	2
3	2	11	2
4	2	12	2
5	2	13	2
6	2	14	2
7	2	15	2
8	2	16	1

Ключ K_i, определяемый на каждом шаге итерации, есть результат выбора конкретных битов из 56-битовой последовательности C_iD_i и их перестановки. Другими словами, ключ $K_i = H(C_iD_i)$, где функция H определяется матрицей, завершающей обработку ключа (табл. 3.9).

Как следует из табл. 3.9, первым битом ключа K_i будет 14-й бит последовательности C_iD_i, вторым – 17-й бит, 47-м битом ключа K_i будет 29-й бит C_iD_i, а 48-м битом – 32-й бит C_iD_i.

Таблица 3.9

Функция H завершающей обработки ключа (переставленная выборка 2)

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

3.2.Основные режимы работы алгоритма DES

Алгоритм DES вполне подходит как для шифрования, так и для аутентификации данных. Он позволяет непосредственно преобразовывать 64-битовый входной открытый текст в 64-битовый выходной шифрованный текст, однако данные редко ограничиваются 64 разрядами.

Чтобы воспользоваться алгоритмом DES для решения разнообразных криптографических задач, разработаны четыре рабочих режима:

- электронная кодовая книга ECB (Electronic Code Book);
- сцепление блоков шифра CBC (Cipher Block Chaining);
- обратная связь по шифртексту CFB (Cipher Feed Back);
- обратная связь по выходу OFB (Output Feed Back).

Режим "Электронная кодовая книга"

Длинный файл разбивают на 64-битовые отрезки (блоки) по 8 байтов. Каждый из этих блоков шифруют независимо с использованием одного и того же ключа шифрования (рис. 3.5).

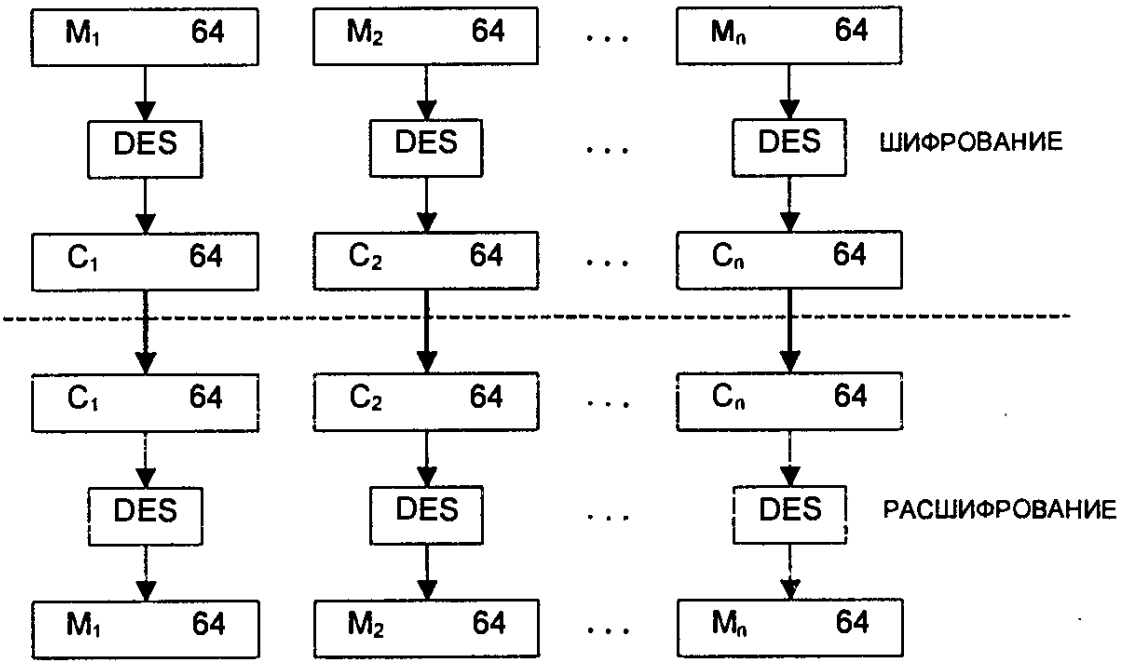


Рис. 3.5. Схема алгоритма DES в режиме электронной кодовой книги

Основное достоинство – простота реализации. Недостаток – относительно слабая устойчивость против квалифицированных криптоаналитиков. Из-за фиксированного характера шифрования при ограниченной длине блока 64 бита возможно проведение криптоанализа "со словарем". Блок такого размера может повториться в сообщении вследствие большой избыточности в тексте на естественном языке. Это приводит к тому, что идентичные блоки открытого текста в сообщении будут представлены идентичными блоками шифртекста, что дает криптоаналитику некоторую информацию о содержании сообщения.

Режим "Сцепление блоков шифра"

В этом режиме исходный файл M разбивается на 64-битовые блоки: $M = M_1 M_2 \ldots M_n$. Первый блок M_1 складывается по модулю 2 с 64-битовым начальным вектором IV, который меняется ежедневно и держится в секрете (рис. 3.6). Полученная сумма затем шифруется с использованием ключа DES, известного и отправителю, и получателю информации. Полученный 64-битовый шифр C_1 складывается по модулю 2 со вторым блоком текста, результат шифруется и получается второй 64-битовый шифр C_2, и т.д. Процедура повторяется до тех пор, пока не будут обработаны все блоки текста.

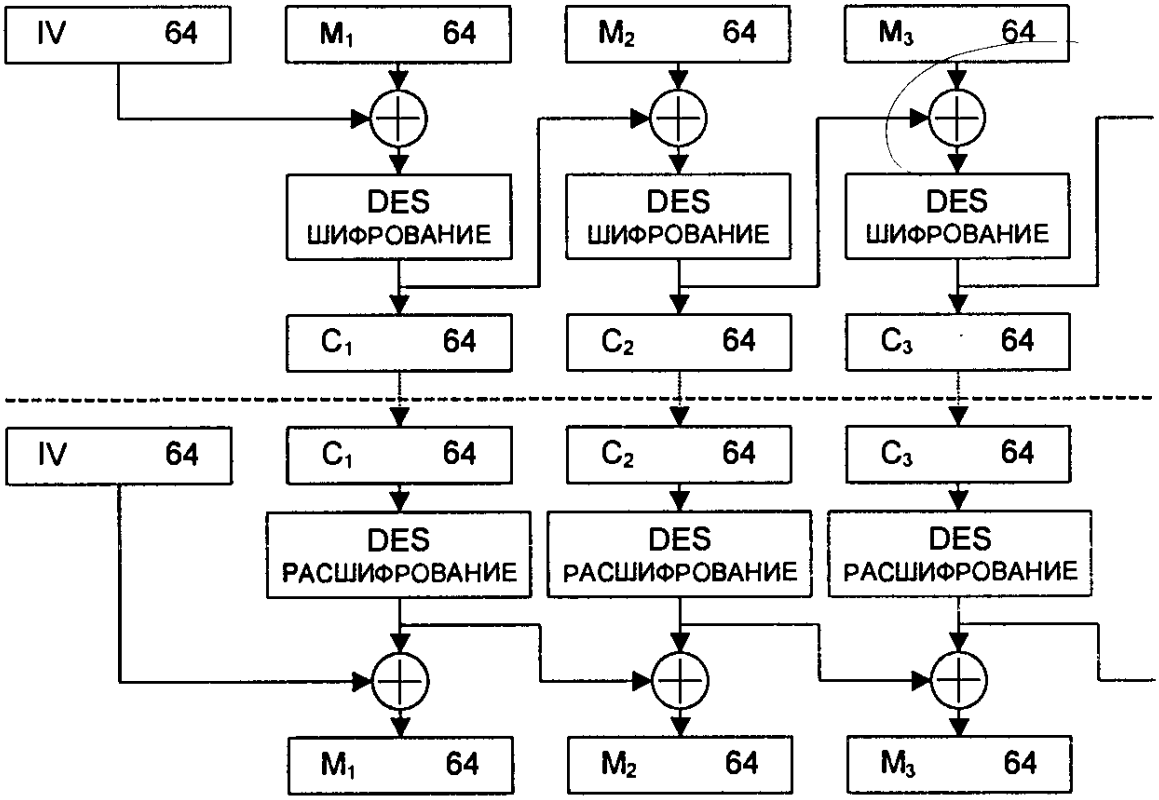


Рис. 3.6. Схема алгоритма DES в режиме сцепления блоков шифра

Таким образом, для всех $i = 1 \ldots n$ (n – число блоков) результат шифрования C_i определяется следующим образом: $C_i = DES(M_i \oplus C_{i-1})$, где $C_0 = IV$ – начальное значение шифра, равное начальному вектору (вектору инициализации).

Очевидно, что последний 64-битовый блок шифртекста является функцией секретного ключа, начального вектора и каждого бита открытого текста независимо от его длины. Этот блок шифртекста называют кодом аутентификации сообщения (КАС).

Код КАС может быть легко проверен получателем, владеющим секретным ключом и начальным вектором, путем повторения процедуры, выполненной отправителем. Посторонний, однако, не может осуществить генерацию КАС, который воспринялся бы получателем как подлинный, чтобы добавить его к ложному сообщению, либо отделить КАС от истинного сообщения для использования его с измененным или ложным сообщением.

Достоинство данного режима в том, что он не позволяет накапливаться ошибкам при передаче.

Блок M_i является функцией только C_{i-1} и C_i. Поэтому ошибка при передаче приведет к потере только двух блоков исходного текста.

Режим "Обратная связь по шифру"

В этом режиме размер блока может отличаться от 64 бит (рис. 3.7). Файл, подлежащий шифрованию (расшифрованию), считывается последовательными блоками длиной k битов (k = 1 ... 64).

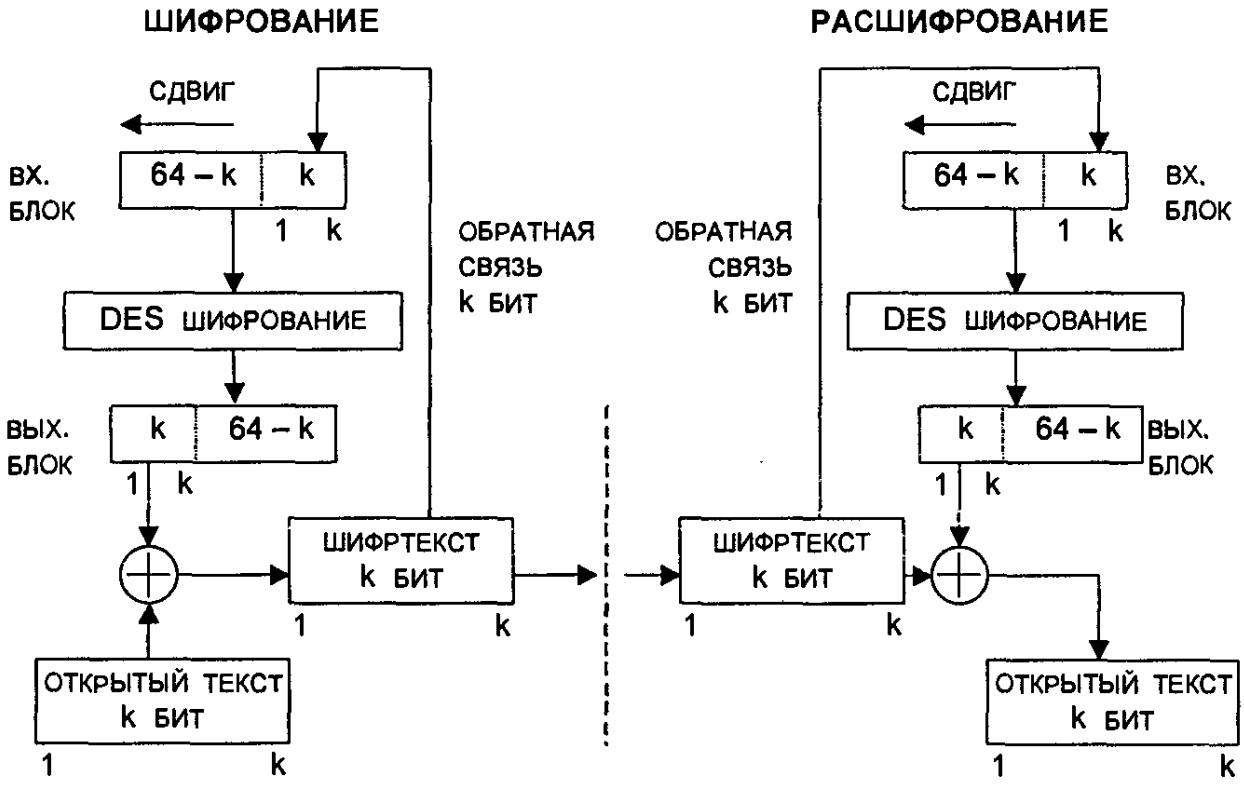


Рис. 3.7. Схема алгоритма DES в режиме обратной связи по шифртексту

Входной блок (64-битовый регистр сдвига) вначале содержит вектор инициализации, выровненный по правому краю.

Предположим, что в результате разбиения на блоки мы получили n блоков длиной k битов каждый (остаток дописывается нулями или пробелами). Тогда для любого i = 1 ... n блок шифртекста

$$C_i = M_i \oplus P_{i-1},$$

где P_{i-1} обозначает k старших битов предыдущего зашифрованного блока.

Обновление сдвигового регистра осуществляется путем удаления его старших k битов и записи C_i в регистр. Восстановление зашифрованных данных также выполняется относительно просто: P_{i-1} и C_i вычисляются аналогичным образом и

$$M_i = C_i \oplus P_{i-1}.$$

Режим "Обратная связь по выходу"

Этот режим тоже использует переменный размер блока и сдвиговый регистр, инициализируемый так же, как в режиме CFB, а именно – входной блок вначале содержит вектор инициализации IV, выровненный по правому краю (рис. 3.8). При этом для каждого сеанса шифрования данных необходимо использовать новое начальное состояние регистра, которое должно пересылаться по каналу открытым текстом.

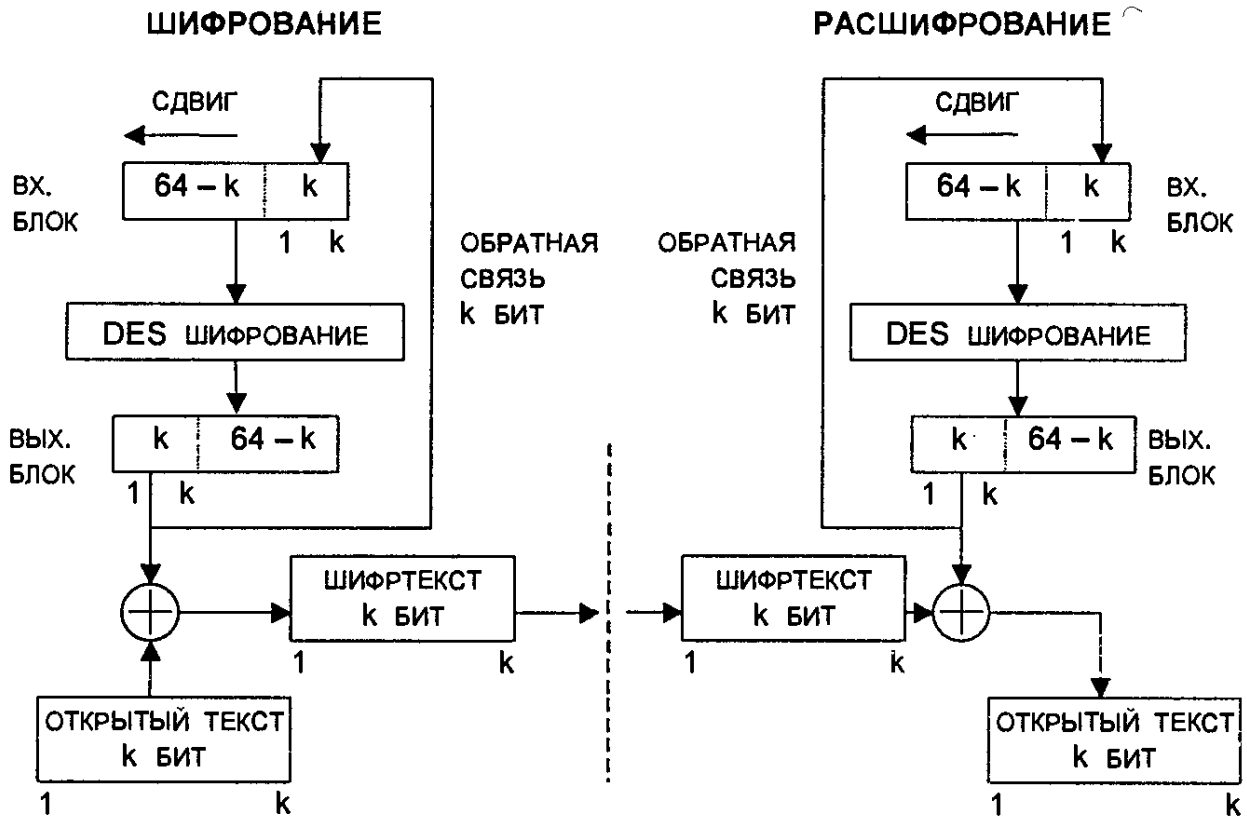


Рис. 3.8. Схема алгоритма DES в режиме обратной связи по выходу

Положим

$$M = M_1 M_2 \ldots M_n.$$

Для всех $i = 1 \ldots n$

$$C_i = M_i \oplus P_i,$$

где P_i – старшие k битов операции DES (C_{i-1}).

Отличие от режима обратной связи по шифртексту состоит в методе обновления сдвигового регистра.

Это осуществляется путем отбрасывания старших k битов и дописывания справа P_i.

Области применения алгоритма DES

Каждому из рассмотренных режимов (ECB, CBC, CFB, OFB) свойственны свои достоинства и недостатки, что обусловливает области их применения.

Режим ECB хорошо подходит для шифрования ключей: режим CFB, как правило, предназначается для шифрования отдельных символов, а режим OFB нередко применяется для шифрования в спутниковых системах связи.

Режимы CBC и CFB пригодны для аутентификации данных. Эти режимы позволяют использовать алгоритм DES для:

- интерактивного шифрования при обмене данными между терминалом и главной ЭВМ;
- шифрования криптографического ключа в практике автоматизированного распространения ключей;
- шифрования файлов, почтовых отправлений, данных спутников и других практических задач.

Первоначально стандарт DES предназначался для шифрования и расшифрования данных ЭВМ. Однако его применение было обобщено и на аутентификацию.

В системах автоматической обработки данных человек не в состоянии просмотреть данные, чтобы установить, не внесены ли в них какие-либо изменения. При огромных объемах данных, проходящих в современных системах обработки, просмотр занял бы слишком много времени. К тому же избыточность данных может оказаться недостаточной для обнаружения ошибок. Даже в тех случаях, когда просмотр человеком возможен, данные могут быть изменены таким образом, что обнаружить эти изменения человеку очень трудно. Например, "do" может быть заменено на "do not", "$1900"– на "$9100". Без дополнительной информации человек при просмотре может легко принять измененные данные за подлинные. Такие опасности могут существовать даже при использовании шифрования данных. Поэтому желательно иметь автоматическое средство обнаружения преднамеренных и непреднамеренных изменений данных.

Обыкновенные коды, обнаруживающие ошибки, непригодны, так как если алгоритм образования кода известен, противник может выработать правильный код после внесения изменений в данные. Однако с помощью алгоритма DES можно образовать криптографическую контрольную сумму, которая может защитить как от случайных, так и преднамеренных, но несанкционированных изменений данных.

Этот процесс описывает стандарт для аутентификации данных ЭВМ (FIPS 113). Суть стандарта состоит в том, что данные зашифровываются в режиме обратной связи по шифртексту (режим CFB) или в режиме сцепления блоков шифра (режим CBC), в результате чего получается окончательный блок шифра, представляющий собой функцию всех разрядов открытого текста. После этого сообщение, которое содержит открытый текст, может быть

передано с использованием вычисленного окончательного блока шифра, служащего в качестве криптографической контрольной суммы.

Одни и те же данные можно защитить, пользуясь как шифрованием, так и аутентификацией. Данные защищаются от ознакомления шифрованием, а изменения обнаруживаются посредством аутентификации. Алгоритм аутентификации можно применить как к открытому, так и к зашифрованному тексту. При финансовых операциях, когда в большинстве случаев реализуются и шифрование, и аутентификация, последняя применяется и к открытому тексту.

Шифрование и аутентификацию используют для защиты данных, хранящихся в ЭВМ. Во многих ЭВМ пароли зашифровывают необратимым образом и хранят в памяти машины. Когда пользователь обращается к ЭВМ и вводит пароль, последний зашифровывается и сравнивается с хранящимся значением. Если обе зашифрованные величины одинаковы, пользователь получает доступ к машине, в противном случае следует отказ.

Нередко зашифрованный пароль вырабатывают с помощью алгоритма DES, причем ключ полагается равным паролю, а открытый текст – коду идентификации пользователя.

С помощью алгоритма DES можно также зашифровать файлы ЭВМ для их хранения [82].

Одним из *наиболее важных применений* алгоритма *DES* является *защита сообщений электронной системы платежей (ЭСП) при операциях с широкой клиентурой и между банками.*

Алгоритм DES реализуется в банковских автоматах, терминалах в торговых точках, автоматизированных рабочих местах и главных ЭВМ. Диапазон защищаемых им данных весьма широк – от оплат $50 до переводов на многие миллионы долларов. Гибкость основного алгоритма DES позволяет использовать его в самых разнообразных областях применения электронной системы платежей.

3.3. Комбинирование блочных алгоритмов

В настоящее время блочный алгоритм DES считается относительно безопасным алгоритмом шифрования. Он подвергался тщательному криптоанализу в течение 20 лет, и самым практичным способом его взламывания является метод перебора всех возможных вариантов ключа. Ключ DES имеет длину 56 бит, поэтому существует 2^{56} возможных вариантов такого ключа. Если предположить, что суперкомпьютер может испытать миллион вариантов ключа за секунду, то потребуется 2285 лет для нахождения правильного ключа. Если бы ключ имел длину 128 бит, то потребовалось бы 10^{25} лет (для сравнения: возраст Вселенной около 10^{10} лет).

Нетрудно представить себе, что при постоянном прогрессе возможностей компьютерной техники недалеко то время, когда машины поиска ключа DES методом полного перебора станут экономичными для мощных в финансовом отношении государственных и коммерческих организаций.

Возникает естественный вопрос: нельзя ли использовать DES в качестве строительного блока для создания другого алгоритма с более длинным ключом?

В принципе существует много способов комбинирования блочных алгоритмов для получения новых алгоритмов. Одним из таких способов комбинирования является многократное шифрование, т.е. использование блочного алгоритма несколько раз с разными ключами для шифрования одного и того же блока открытого текста. Двухкратное шифрование блока открытого текста одним и тем же ключом не приводит к положительному результату. При использовании одного и того же алгоритма такое шифрование не влияет на сложность криптоаналитической атаки полного перебора.

Рассмотрим эффективность двухкратного шифрования блока открытого текста с помощью двух разных ключей. Сначала шифруют блок P ключом K_1, а затем получившийся шифртекст $E_{K1}(P)$ шифруют ключом K_2. В результате двухкратного шифрования получают криптограмму

$$C = E_{K2}(E_{K1}(P)).$$

Расшифрование является обратным процессом:

$$P = D_{K1}(D_{K2}(C)).$$

Если блочный алгоритм обладает свойствами группы [125], то всегда найдется такой ключ K_3, что

$$C = E_{K2}(E_{K1}(P)) = E_{K3}(P).$$

Если же блочный алгоритм не является группой, то результирующий двухкратно шифрованный блок текста окажется намного сложнее для взламывания методом полного перебора вариантов. Вместо 2^n попыток, где n – длина ключа в битах, потребуется 2^{2n} попыток. В частности, если $n = 64$, то двухкратно зашифрованный блок текста потребует 2^{128} попыток для нахождения ключа.

Однако Р. Меркль и М. Хеллман показали на примере DES, что, используя метод "обмена времени на память" и криптоаналитическую атаку "встреча посредине", можно взломать такую схему двухкратного шифрования за 2^{n+1} попыток [34]. Хотя эта атака потребует очень большого объема памяти (для алгоритма с 56-битовым ключом потребуется 2^{56} 64-битовых блоков или 10^{17} бит памяти).

Более привлекательную идею предложил У. Тачмен [125]. Суть этой идеи состоит в том, чтобы шифровать блок открытого текста P три раза с помощью двух ключей K_1 и K_2 (рис. 3.9). Процедура шифрования:

$$C = E_{K1}(D_{K2}(E_{K1}(P))),$$

т. е. блок открытого текста P сначала шифруется ключом K_1, затем расшифровывается ключом K_2 и окончательно зашифровывается ключом K_1.

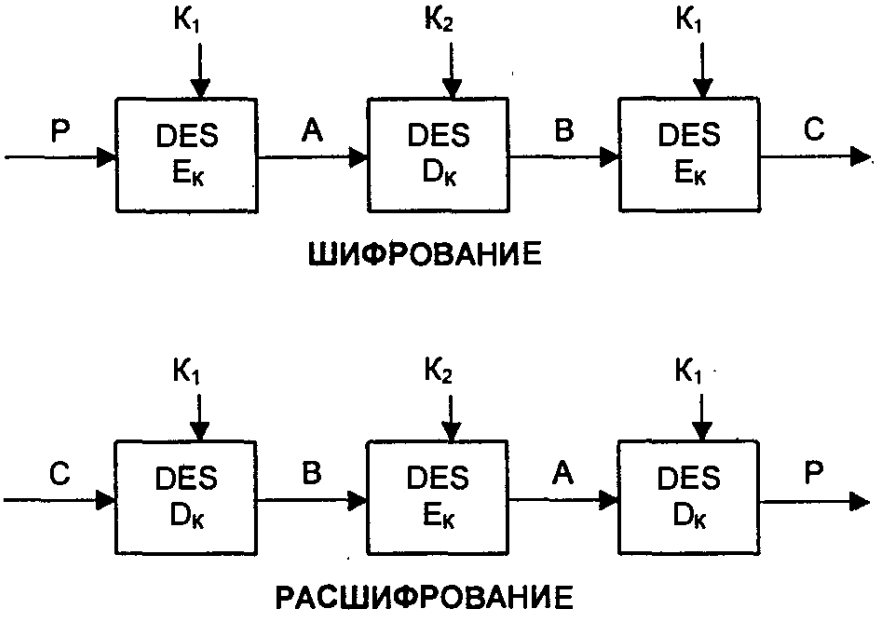


Рис. 3.9. Схемы трехкратного применения алгоритма DES с двумя разными ключами

Этот режим иногда называют режимом EDE (encrypt-decrypt-encrypt). Введение в данную схему операции расшифрования D_{K2} позволяет обеспечить совместимость этой схемы со схемой однократного использования алгоритма DES. Если в схеме трехкратного использования DES выбрать все ключи одинаковыми, то эта схема превращается в схему однократного использования DES. Процедура расшифрования выполняется в обратном порядке:

$$P = D_{K1}(E_{K2}(D_{K1}(C))),$$

т. е. блок шифртекста C сначала расшифровывается ключом K_1, затем зашифровывается ключом K_2 и окончательно расшифровывается ключом K_1.

Если исходный блочный алгоритм имеет n-битовый ключ, то схема трехкратного шифрования имеет 2n-битовый ключ. Чередование ключей K_1 и K_2 позволяет предотвратить криптоаналитическую атаку "встреча посредине". Данная схема приводится в стандартах X9.17 и ISO 8732 в качестве средства улучшения характеристик алгоритма DES.

При трехкратном шифровании можно применить три различных ключа. При этом возрастает общая длина результирующего ключа. Процедуры шифрования и расшифрования описываются выражениями:

$$C = E_{K3}(D_{K2}(E_{K1}(P))),$$

$$P = D_{K1}(E_{K2}(D_{K3}(C))).$$

Трехключевой вариант имеет еще большую стойкость. Очевидно, что если требуется повысить безопасность большого парка оборудования, использующего DES, то гораздо дешевле переключиться на схемы трехкратных DES, чем переходить на другой тип криптосхем.

3.4. Алгоритм шифрования данных IDEA

Алгоритм IDEA (International Data Encryption Algorithm) является блочным шифром. Он оперирует 64-битовыми блоками открытого текста. Несомненным достоинством алгоритма IDEA является то, что его ключ имеет длину 128 бит. Один и тот же алгоритм используется и для шифрования, и для расшифрования.

Первая версия алгоритма IDEA была предложена в 1990 г., ее авторы – Х. Лей и Дж. Мэсси. Первоначальное название алгоритма PES (Proposed Encryption Standard). Улучшенный вариант этого алгоритма, разработанный в 1991 г., получил название IPES (Improved Proposed Encryption Standard). В 1992 г. IPES изменил свое имя на IDEA. Как и большинство других блочных шифров, алгоритм IDEA использует при шифровании процессы смешивания и рассеивания, причем все процессы легко реализуются аппаратными и программными средствами.

В алгоритме IDEA используются следующие математические операции:

- поразрядное сложение по модулю 2 (операция "исключающее ИЛИ"); операция обозначается как $\oplus$;
- сложение беззнаковых целых по модулю 2^{16} (модуль 65536); операция обозначается как $\boxplus$;
- умножение целых по модулю $(2^{16}+1)$ (модуль 65537), рассматриваемых как беззнаковые целые, за исключением того, что блок из 16 нулей рассматривается как 2^{16}; операция обозначается как $\odot$.

Все операции выполняются над 16-битовыми субблоками.

Эти три операции несовместимы в том смысле, что:

- никакая пара из этих трех операций не удовлетворяет ассоциативному закону, например $a \boxplus (b \oplus c) \neq (a \boxplus b) \oplus c$;
- никакая пара из этих трех операций не удовлетворяет дистрибутивному закону, например $a \boxplus (b \odot c) \neq (a \boxplus b) \odot (a \boxplus c)$.

Комбинирование этих трех операций обеспечивает комплексное преобразование входа, существенно затрудняя крипто-анализ IDEA по сравнению с DES, который базируется исключительно на операции "исключающее ИЛИ".

Общая схема алгоритма IDEA приведена на рис.3.10. 64-битовый блок данных делится на четыре 16-битовых субблока. Эти четыре субблока становятся входом в первый цикл алгоритма. Всего выполняется восемь циклов. Между циклами второй и третий субблоки меняются местами. В каждом цикле имеет место следующая последовательность операций:

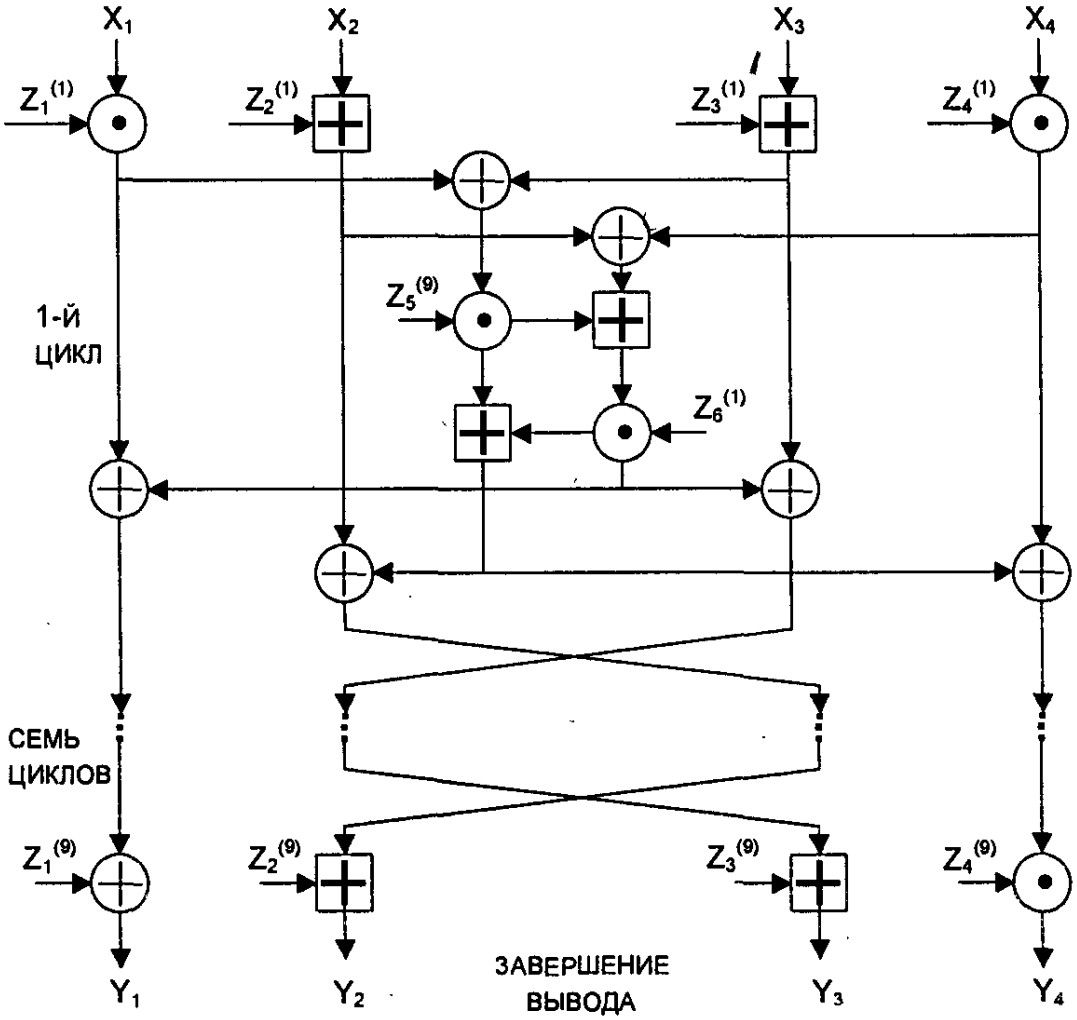
(1) $\odot$ – умножение субблока X_1 и первого подключа.
(2) $\boxplus$ – сложение субблока X_2 и второго подключа.
(3) $\boxplus$ – сложение субблока X_3 и третьего подключа.
(4) $\odot$ – умножение субблока X_4 и четвертого подключа.
(5) $\oplus$ – сложение результатов шагов (1) и (3).
(6) $\oplus$ – сложение результатов шагов (2) и (4).
(7) $\odot$ – умножение результата шага (5) и пятого подключа.
(8) $\boxplus$ – сложение результатов шагов (6) и (7).
(9) $\odot$ – умножение результата шага (8) с шестым подключом.
(10) $\boxplus$ – сложение результатов шагов (7) и (9).
(11) $\oplus$ – сложение результатов шагов (1) и (9).
(12) $\oplus$ – сложение результатов шагов (3) и (9).
(13) $\oplus$ – сложение результатов шагов (2) и (10).
(14) $\oplus$ – сложение результатов шагов (4) и (10).

Выходом цикла являются четыре субблока, которые получают как результаты выполнения шагов (11), (12), (13) и (14). В завершение цикла переставляют местами два внутренних субблока (за исключением последнего цикла), и в результате формируется вход для следующего цикла.

После восьмого цикла осуществляют заключительное преобразование выхода:

(1) $\odot$ – умножение субблока X_1 и первого подключа.
(2) $\boxplus$ – сложение субблока X_2 и второго подключа.
(3) $\boxplus$ – сложение субблока X_3 и третьего подключа.
(4) $\odot$ – умножение субблока X_4 и четвертого подключа.

Наконец, эти результирующие четыре субблока $Y_1 \ldots Y_4$ вновь объединяют для получения блока шифртекста.



Обозначения:
X_i – 16-битовый субблок открытого текста, i = 1... 4
Y_i – 16-битовый субблок шифртекста, i = 1... 4
$Z_i^{(r)}$ – 16-битовый подключ (субблок ключа), j = 1...6, r = 1...8

⊕ – поразрядное суммирование по модулю 2 16-битовых субблоков

⊞ – сложение по модулю 2^{16} 16-битовых целых

⊙ – умножение по модулю 2^{16} 16-битовых целых (с нулевым субблоком, соответствующим 2^{16})

Рис. 3.10. Схема алгоритма IDEA (режим шифрования)

Создание подключей Z_j также относительно несложно. Алгоритм использует всего 52 подключа (по шесть для каждого из восьми циклов и еще четыре для преобразования выхода). Сначала 128-битовый ключ делят на восемь 16-битовых подключей. Это – первые восемь подключей для алгоритма (шесть подключей – для первого цикла и первые два подключа – для второго цикла). Затем 128-битовый ключ циклически сдвигается влево на 25 бит и снова делится на восемь подключей. Первые четыре из них используют во втором цикле; последние четыре – в третьем цикле. Ключ снова

циклически сдвигается влево еще на 25 бит для получения следующих восьми подключей и т.д., пока выполнение алгоритма не завершится.

Расшифрование осуществляют аналогичным образом, за исключением того, что порядок использования подключей становится обратным, причем ряд значений подключей заменяется на обратные значения. Подключи расшифрования являются в основном либо аддитивными, либо мультипликативными обратными величинами подключей шифрования (табл. 3.10).

Таблица 3.10

Подключи шифрования и расшифрования алгоритма IDEA

Цикл	Подключи шифрования	Подключи расшифрования
1	$Z_1^{(1)}\ Z_2^{(1)}\ Z_3^{(1)}\ Z_4^{(1)}\ Z_5^{(1)}\ Z_6^{(1)}$	$Z_1^{(9)-1}\ -Z_2^{(9)}\ -Z_3^{(9)}\ Z_4^{(9)-1}\ Z_5^{(8)}\ Z_6^{(8)}$
2	$Z_1^{(2)}\ Z_2^{(2)}\ Z_3^{(2)}\ Z_4^{(2)}\ Z_5^{(2)}\ Z_6^{(2)}$	$Z_1^{(8)-1}\ -Z_3^{(8)}\ -Z_2^{(8)}\ Z_4^{(8)-1}\ Z_5^{(7)}\ Z_6^{(7)}$
3	$Z_1^{(3)}\ Z_2^{(3)}\ Z_3^{(3)}\ Z_4^{(3)}\ Z_5^{(3)}\ Z_6^{(3)}$	$Z_1^{(7)-1}\ -Z_3^{(7)}\ -Z_2^{(7)}\ Z_4^{(7)-1}\ Z_5^{(6)}\ Z_6^{(6)}$
4	$Z_1^{(4)}\ Z_2^{(4)}\ Z_3^{(4)}\ Z_4^{(4)}\ Z_5^{(4)}\ Z_6^{(4)}$	$Z_1^{(6)-1}\ -Z_3^{(6)}\ -Z_2^{(6)}\ Z_4^{(6)-1}\ Z_5^{(5)}\ Z_6^{(5)}$
5	$Z_1^{(5)}\ Z_2^{(5)}\ Z_3^{(5)}\ Z_4^{(5)}\ Z_5^{(5)}\ Z_6^{(5)}$	$Z_1^{(5)-1}\ -Z_3^{(5)}\ -Z_2^{(5)}\ Z_4^{(5)-1}\ Z_5^{(4)}\ Z_6^{(4)}$
6	$Z_1^{(6)}\ Z_2^{(6)}\ Z_3^{(6)}\ Z_4^{(6)}\ Z_5^{(6)}\ Z_6^{(6)}$	$Z_1^{(4)-1}\ -Z_3^{(4)}\ -Z_2^{(4)}\ Z_4^{(4)-1}\ Z_5^{(3)}\ Z_6^{(3)}$
7	$Z_1^{(7)}\ Z_2^{(7)}\ Z_3^{(7)}\ Z_4^{(7)}\ Z_5^{(7)}\ Z_6^{(7)}$	$Z_1^{(3)-1}\ -Z_3^{(3)}\ -Z_2^{(3)}\ Z_4^{(3)-1}\ Z_5^{(2)}\ Z_6^{(2)}$
8	$Z_1^{(8)}\ Z_2^{(8)}\ Z_3^{(8)}\ Z_4^{(8)}\ Z_5^{(8)}\ Z_6^{(8)}$	$Z_1^{(2)-1}\ -Z_3^{(2)}\ -Z_2^{(2)}\ Z_4^{(2)-1}\ Z_5^{(1)}\ Z_6^{(1)}$
Преобразование выхода	$Z_1^{(9)}\ Z_2^{(9)}\ Z_3^{(9)}\ Z_4^{(9)}$	$Z_1^{(1)-1}\ -Z_2^{(1)}\ -Z_3^{(1)}\ Z_4^{(1)-1}$

Для реализации алгоритма IDEA было принято предположение, что нулевой субблок равен $2^{16}=-1$; при этом мультипликативная обратная величина от 0 равна 0 [121]. Вычисление значений мультипликативных обратных величин требует некоторых затрат, но это приходится делать только один раз для каждого ключа расшифрования.

Алгоритм IDEA может работать в любом режиме блочного шифра, предусмотренном для алгоритма DES. Алгоритм IDEA обладает рядом преимуществ перед алгоритмом DES. Он значительно безопаснее алгоритма DES, поскольку 128-битовый ключ алгоритма IDEA вдвое больше ключа DES. Внутренняя структура алгоритма IDEA обеспечивает лучшую устойчивость к криптоанализу. Существующие программные реализации алгоритма IDEA примерно вдвое быстрее реализаций алгоритма DES. Алгоритм IDEA шифрует данные на IBM PC/486 со скоростью 2,4 Мбит/с. Реализация IDEA на СБИС шифрует данные со скоростью 177 Мбит/с при частоте 25 Мгц. Алгоритм IDEA запатентован в Европе и США.

3.5. Отечественный стандарт шифрования данных

В нашей стране установлен единый алгоритм криптографического преобразования данных для систем обработки информации в сетях ЭВМ, отдельных вычислительных комплексах и ЭВМ, кото-

рый определяется ГОСТ 28147-89 [81]. Стандарт обязателен для организаций, предприятий и учреждений, применяющих криптографическую защиту данных, хранимых и передаваемых в сетях ЭВМ, в отдельных вычислительных комплексах и ЭВМ.

Этот алгоритм криптографического преобразования данных предназначен для аппаратной и программной реализации, удовлетворяет криптографическим требованиям и не накладывает ограничений на степень секретности защищаемой информации. Алгоритм шифрования данных представляет собой 64-битовый блочный алгоритм с 256-битовым ключом.

При описании алгоритма используются следующие обозначения:

L и R – последовательности битов;

LR – конкатенация последовательностей L и R, в которой биты последовательности R следуют за битами последовательности L;

$\oplus$ – операция побитового сложения по модулю 2;

$\boxplus$ – операция сложения по модулю 2^{32} двух 32-разрядных двоичных чисел;

$\boxplus'$ – операция сложения двух 32-разрядных чисел по модулю $2^{32}-1$.

Два целых числа a, b, где $0 \le a, b \le 2^{32}-1$,

$$a = (a_{32}a_{31} \ldots a_2a_1), \quad b = (b_{32}, b_{31}, \ldots, b_2, b_1),$$

представленные в двоичном виде, т. е.

$$a = a_{32}\cdot 2^{31} + a_{31}\cdot 2^{30} + \ldots + a_2\cdot 2^1 + a_1,$$

$$b = b_{32}\cdot 2^{31} + b_{31}\cdot 2^{30} + \ldots + b_2\cdot 2^1 + b_1,$$

суммируются по модулю 2^{32} (операция $\boxplus$) по следующему правилу:

$$a \boxplus b = a + b, \quad \text{если} \quad a + b < 2^{32},$$

$$a \boxplus b = a + b - 2^{32}, \quad \text{если} \quad a + b \ge 2^{32}.$$

Правила суммирования чисел по модулю $2^{32}-1$:

$$a \boxplus' b = a + b, \quad \text{если} \quad a + b < 2^{32}-1,$$

$$a \boxplus' b = a + b - (2^{32}-1), \quad \text{если} \quad a + b \ge 2^{32}-1.$$

Алгоритм предусматривает четыре режима работы:

шифрование данных в режиме простой замены;

шифрование данных в режиме гаммирования;

шифрование данных в режиме гаммирования с обратной связью;

выработка имитовставки.

Режим простой замены

Для реализации алгоритма шифрования данных в режиме простой замены используется только часть блоков общей криптосистемы (рис. 3.11). Обозначения на схеме:

N_1, N_2 – 32-разрядные накопители;

$СМ_1$ – 32-разрядный сумматор по модулю 2^{32} (⊞);

$СМ_2$ – 32-разрядный сумматор по модулю 2 (⊕);

R – 32-разрядный регистр циклического сдвига;

КЗУ – ключевое запоминающее устройство на 256 бит, состоящее из восьми 32-разрядных накопителей X_0, X_1, X_2, ..., X_7;

S – блок подстановки, состоящий из восьми узлов замены (S-блоков замены) S_1, S_2, S_3, ..., S_7, S_8.

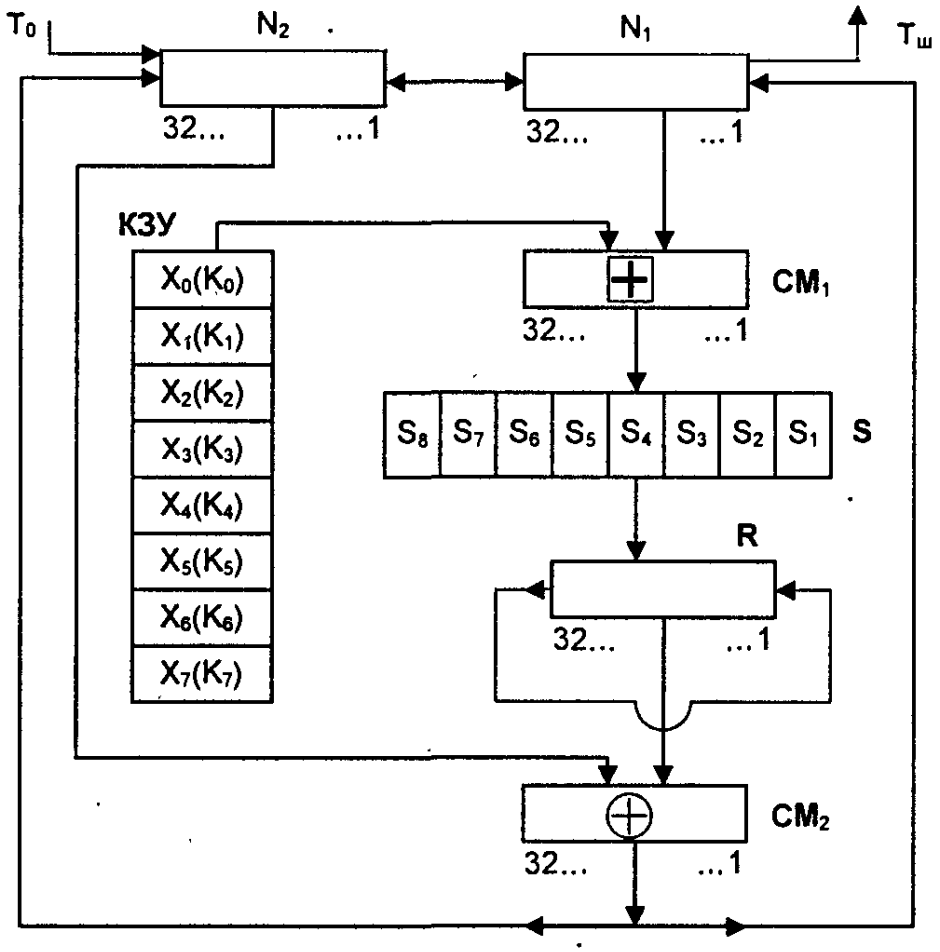


Рис. 3.11. Схема реализации режима простой замены

Зашифрование открытых данных в режиме простой замены. Открытые данные, подлежащие зашифрованию, разбивают на 64-разрядные блоки T_0. Процедура зашифрования 64-разрядного блока T_0 в режиме простой замены включает 32 цикла (j = 1 ... 32). В ключевое запоминающее устройство вводят 256 бит ключа К в виде восьми 32-разрядных подключей (чисел) K_i:

$$K=K_7K_6K_5K_4K_3K_2K_1K_0.$$

Последовательность битов блока

$$T_0 = (a_1(0), a_2(0), ..., a_{31}(0), a_{32}(0), \quad b_1(0), b_2(0), ..., b_{31}(0), b_{32}(0))$$

разбивают на две последовательности по 32 бита: b(0) a(0), где b(0) – левые или старшие биты, a(0) – правые или младшие биты.

Эти последовательности вводят в накопители N_1 и N_2 перед началом первого цикла зашифрования. В результате начальное заполнение накопителя N_1

$$a(0) = (a_{32}(0), a_{31}(0), ..., a_2(0), a_1(0)),$$
$$32, \quad 31, \quad ... \quad 2, \quad 1 \quad \leftarrow \text{номер разряда } N_1$$

начальное заполнение накопителя N_2

$$b(0) = (b_{32}(0), b_{31}(0), ..., b_2(0), b_1(0)).$$
$$32, \quad 31, \quad ... \quad 2, \quad 1 \quad \leftarrow \text{номер разряда } N_2$$

Первый цикл (j = 1) процедуры зашифрования 64-разрядного блока открытых данных можно описать уравнениями:

$$\begin{cases} a(1) = f(a(0) \boxplus K_0) \oplus b(0), \\ b(1) = a(0). \end{cases}$$

Здесь a (1) – заполнение N_1 после 1-го цикла зашифрования; b(1) – заполнение N_2 после 1-го цикла зашифрования; f – функция шифрования.

Аргументом функции f является сумма по модулю 2^{32} числа a(0) (начального заполнения накопителя N_1) и числа K_0 – подключа, считываемого из накопителя X_0 КЗУ. Каждое из этих чисел равно 32 битам.

Функция f включает две операции над полученной 32-разрядной суммой $(a(0) \boxplus K_0)$.

Первая операция называется *подстановкой (заменой)* и выполняется блоком подстановки S. Блок подстановки S состоит из восьми узлов замены (S-блоков замены) $S_1, S_2, ..., S_8$ с памятью 64 бит каждый. Поступающий из $СМ_1$ на блок подстановки S 32-разрядный вектор разбивают на восемь последовательно идущих 4-разрядных векторов, каждый из которых преобразуется в четырехразрядный вектор соответствующим узлом замены. Каждый узел замены можно представить в виде таблицы-перестановки шестнадцати четырехразрядных двоичных чисел в диапазоне 0000 ... 1111. Входной вектор указывает адрес строки в таблице, а число в этой строке является выходным вектором. Затем четырехразрядные выходные векторы последовательно объединяют в 32-разрядный вектор. Узлы замены (таблицы-перестановки) представляют собой ключевые элементы, которые являются общими для сети ЭВМ и редко изменяются. Эти узлы замены должны сохраняться в секрете.

Вторая операция – *циклический сдвиг влево* (на 11 разрядов) 32-разрядного вектора, полученного с выхода блока подстановки S. Циклический сдвиг выполняется регистром сдвига R.

Далее результат работы функции шифрования f суммируют поразрядно по модулю 2 в сумматоре $СМ_2$ с 32-разрядным начальным заполнением b(0) накопителя N_2. Затем полученный на выходе $СМ_2$ результат (значение a(1)) записывают в накопитель N_1, а старое значение N_1 (значение a(0)) переписывают в накопитель N_2 (значение b(1) = a(0)). Первый цикл завершен.

Последующие циклы осуществляются аналогично, при этом во втором цикле из КЗУ считывают заполнение X_1 – подключ $К_1$, в третьем цикле – подключ $К_2$ и т.д., в восьмом цикле – подключ $К_7$. В циклах с 9-го по 16-й, а также в циклах с 17-го по 24-й подключи из КЗУ считываются в том же порядке: $К_0, К_1, К_2, ..., К_6, К_7$. В последних восьми циклах с 25-го по 32-й порядок считывания подключей из КЗУ обратный: $К_7, К_6, ..., К_2, К_1, К_0$. Таким образом, при зашифровании в 32 циклах осуществляется следующий порядок выборки из КЗУ подключей:

$$К_0, К_1, К_2, К_3, К_4, К_5, К_6, К_7, К_0, К_1, К_2, К_3, К_4, К_5, К_6, К_7,$$

$$К_0, К_1, К_2, К_3, К_4, К_5, К_6, К_7, К_7, К_6, К_5, К_4, К_3, К_2, К_1, К_0.$$

В 32-м цикле результат из сумматора $СМ_2$ вводится в накопитель N_2, а в накопителе N_1 сохраняется прежнее заполнение. Полученные после 32-го цикла зашифрования заполнения накопителей N_1 и N_2 являются блоком зашифрованных данных $Т_ш$, соответствующим блоку открытых данных $Т_0$.

Уравнения зашифрования в режиме простой замены имеют вид:

$$\begin{cases} a(j) = f(a(j-1) \boxplus K_{j-1(\mathrm{mod}8)}) \oplus b(j-1) \\ b(j) = a(j-1) \end{cases} \quad \text{при } j = 1 \ldots 24,$$

$$\begin{cases} a(j) = f(a(j-1) \boxplus K_{32-j}) \oplus b(j-1) \\ b(j) = a(j-1) \end{cases} \quad \text{при } j = 25 \ldots 31,$$

$$\begin{cases} a(32) = a(31) \\ b(32) = f(a(31) \boxplus K_0) \oplus b(31) \end{cases} \quad \text{при } j = 32,$$

где $a(j) = (a_{32}(j), a_{31}(j), ..., a_1(j))$ – заполнение N_1 после j-го цикла зашифрования; $b(j) = (b_{32}(j), b_{31}(j), ..., b_1(j))$ – заполнение N_2 после j-го цикла зашифрования, $j = 1 \ldots 32$.

Блок зашифрованных данных $Т_ш$ (64 разряда) выводится из накопителей N_1, N_2 в следующем порядке: из разрядов 1... 32 накопителя N_1, затем из разрядов 1... 32 накопителя N_2, т.е. начиная с младших разрядов:

$$T_ш = (a_1(32), a_2(32), ..., a_{32}(32), b_1(32), b_2(32), ..., b_{32}(32)).$$

Остальные блоки открытых данных зашифровываются в режиме простой замены аналогично.

Расшифрование в режиме простой замены. Криптосхема, реализующая алгоритм расшифрования в режиме простой замены, имеет тот же вид, что и при зашифровании (см. рис.3.11).

В КЗУ вводят 256 бит ключа, на котором осуществлялось зашифрование. Зашифрованные данные, подлежащие расшифрованию, разбиты на блоки $T_ш$ по 64 бита в каждом. Ввод любого блока

$$T_ш = (a_1(32), a_2(32), ..., a_{32}(32), b_1(32), b_2(32), ..., b_{32}(32))$$

в накопители N_1 и N_2 производят так, чтобы начальное значение накопителя N_1 имело вид

$$\underset{32,}{(a_{32}(32)}, \underset{31,}{a_{31}(32)}, ..., \underset{2,}{a_2(32)}, \underset{1}{a_1(32)}),$$

← номер разряда N_1

а начальное заполнение накопителя N_2 – вид

$$\underset{32,}{(b_{32}(32)}, \underset{31,}{b_{31}(32)}, ..., \underset{2,}{b_2(32)}, \underset{1}{b_1(32)}).$$

← номер разряда N_2

Расшифрование осуществляется по тому же алгоритму, что и зашифрование, с тем изменением, что заполнения накопителей X_0, X_1, ..., X_7 считываются из КЗУ в циклах расшифрования в следующем порядке:

$$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0,$$

$$K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0.$$

Уравнения расшифрования имеют вид:

$$\begin{cases} a(32-j) = f(a(32-j+1) \boxplus K_{j-1}) \oplus b(32-j+1) \\ b(32-j) = a(32-j+1) \end{cases} \quad \text{при } j = 1...8;$$

$$\begin{cases} a(32-j) = f(a(32-j+1) \boxplus K_{32-j(\mathrm{mod}\,8)}) \oplus b(32-j+1) \\ b(32-j) = a(32-j+1) \end{cases} \quad \text{при } j = 9...31;$$

$$\begin{cases} a(0) = a(1) \\ b(0) = f(a(1) \boxplus K_0) \oplus b(1) \end{cases} \quad \text{при } j = 32.$$

Полученные после 32 циклов работы заполнения накопителей N_1 и N_2 образуют блок открытых данных

$$T_0 = (a_1(0), a_2(0), ..., a_{32}(0), b_1(0), b_2(0), ..., b_{32}(0)),$$

соответствующий блоку зашифрованных данных $T_ш$. При этом состояние накопителя N_1

$$\underset{32,}{(a_{32}(0)}, \underset{31,}{a_{31}(0)}, ..., \underset{2,}{a_2(0)}, \underset{1}{a_1(0)}),$$

← номер разряда N_1

состояние накопителя N_2

$$(b_{32}(0), b_{31}(0), ..., b_2(0), b_1(0)).$$
$$32, \quad 31, \quad ..., \quad 2, \quad 1 \quad \leftarrow \text{номер разряда } N_2$$

Аналогично расшифровываются остальные блоки зашифрованных данных.

Если алгоритм зашифрования в режиме простой замены 64-битового блока T_0 обозначить через A, то

$$A(T_0) = A(a(0), b(0)) = (a(32), b(32)) = T_ш.$$

Следует иметь в виду, что режим простой замены допустимо использовать для шифрования данных только в ограниченных случаях – при выработке ключа и зашифровании его с обеспечением имитозащиты для передачи по каналам связи или для хранения в памяти ЭВМ.

Режим гаммирования

Зашифрование открытых данных в режиме гаммирования. Криптосхема, реализующая алгоритм зашифрования в режиме гаммирования, показана на рис. 3.12. Открытые данные разбивают на 64-разрядные блоки

$$T_0^{(1)}, T_0^{(2)}, ..., T_0^{(i)}, ..., T_0^{(m)},$$

где $T_0^{(i)}$ – i-й 64-разрядный блок открытых данных, i = 1... m, m определяется объемом шифруемых данных.

Эти блоки поочередно зашифровываются в режиме гаммирования путем поразрядного сложения по модулю 2 в сумматоре $СМ_5$ с гаммой шифра $Г_ш$, которая вырабатывается блоками по 64 бита, т. е.

$$Г_ш = (Г_ш^{(1)}, Г_ш^{(2)}, ..., Г_ш^{(i)}, Г_ш^{(m)}),$$

где $Г_ш^{(i)}$ – i-й 64-разрядный блок, i = 1... m.

Число двоичных разрядов в блоке $T_0^{(m)}$ может быть меньше 64, при этом неиспользованная для зашифрования часть гаммы шифра из блока $Г_ш^{(m)}$ отбрасывается.

Уравнение зашифрования данных в режиме гаммирования имеет вид

$$T_ш^{(i)} = T_0^{(i)} \oplus Г_ш^{(i)},$$

где $Г_ш^{(i)} = A(Y_{i-1} \boxplus C_2, Z_{i-1} \boxplus C_1)$, i = 1... m; $T_ш^{(i)}$ – i-й блок 64-разрядного блока зашифрованного текста; $A(\cdot)$ – функция зашифрования в режиме простой замены; C_1, C_2 – 32-разрядные двоичные константы; Y_i, Z_i – 32-разрядные двоичные последовательности.

Величины Y_i, Z_i определяются итерационно по мере формирования гаммы $Г_ш$ следующим образом:

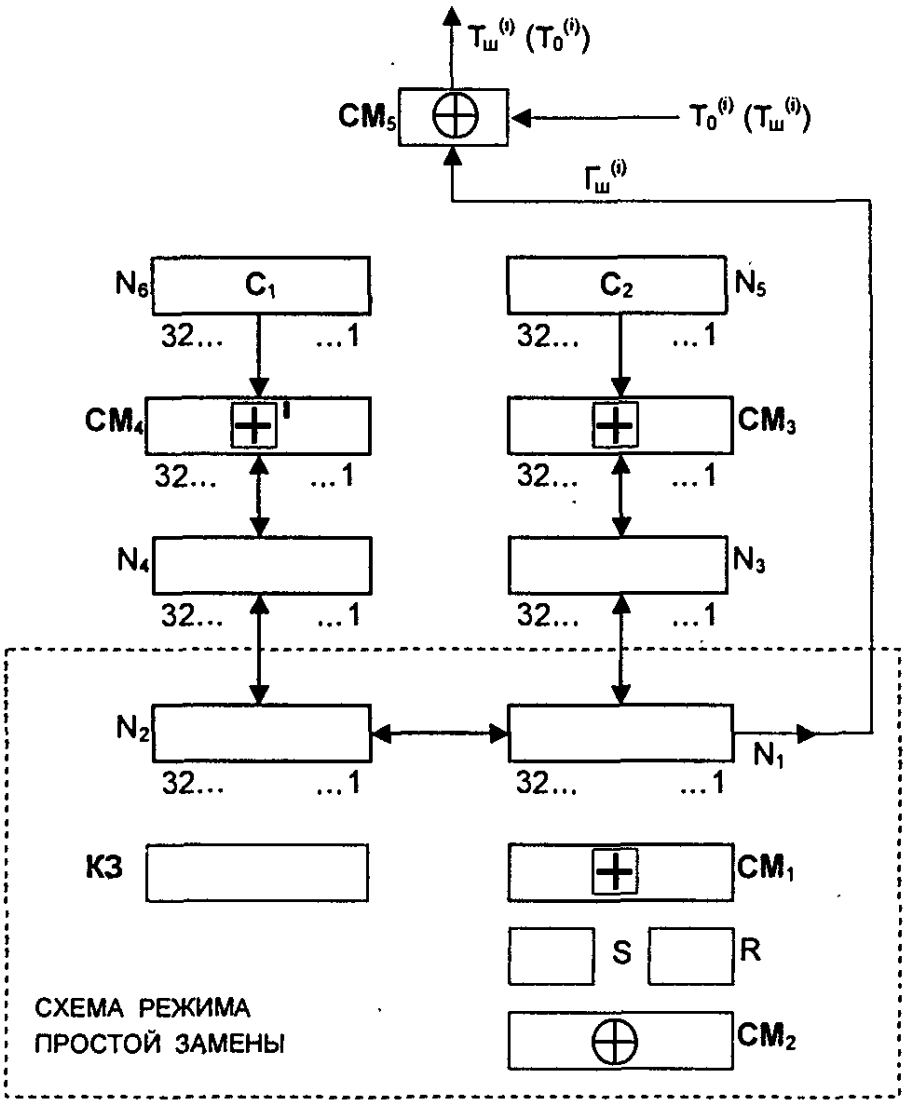


Рис. 3.12. Схема реализации режима гаммирования

$$(Y_0, Z_0) = A(\tilde{S}),$$

где $\tilde{S}$ – синхропосылка (64-разрядная двоичная последовательность),

$$(Y_i, Z_i) = (Y_{i-1} \boxplus C_2, Z_{i-1} \boxplus C_1), \quad i = 1 \ldots m.$$

Рассмотрим реализацию процедуры зашифрования в режиме гаммирования. В накопители N_6 и N_5 заранее записаны 32-разрядные двоичные константы C_1 и C_2, имеющие следующие значения (в шестнадцатеричной форме):

$$C_1 = 01010104_{(16)}, \quad C_2 = 01010101_{(16)}.$$

В КЗУ вводится 256 бит ключа; в накопители N_1 и N_2 – 64-разрядная двоичная последовательность (синхропосылка)

$$\tilde{S} = (S_1, S_2, \ldots, S_{64}).$$

Синхропосылка $\tilde{S}$ является исходным заполнением накопителей N_1 и N_2 для последовательной выработки m блоков гаммы шифра.

Исходное заполнение накопителя N_1:

$$(S_{32}, S_{31}, \ldots, S_2, S_1);$$
$$32, \quad 31, \ldots, \quad 2, \quad 1 \leftarrow \text{номер разряда } N_1$$

исходное заполнение накопителя N_2:

$$(S_{64}, S_{63}, \ldots, S_{34}, S_{33}).$$
$$64, \quad 63, \ldots, \quad 34, \quad 33 \leftarrow \text{номер разряда } N_2$$

Исходное заполнение N_1 и N_2 (синхропосылка $\tilde{S}$) зашифровывается в режиме простой замены. Результат зашифрования

$$A(\tilde{S}) = (Y_0, Z_0)$$

переписывается в 32-разрядные накопители N_3 и N_4 так, что заполнение N_1 переписывается в N_3, а заполнение N_2 – в N_4.

Заполнение накопителя N_4 суммируют по модулю $(2^{32}-1)$ в сумматоре $СМ_4$ с 32-разрядной константой C_1 из накопителя N_6. Результат записывается в N_4. Заполнение накопителя N_3 суммируется по модулю 2^{32} в сумматоре $СМ_3$ с 32-разрядной константой C_2 из накопителя N_5. Результат записывается в N_3. Заполнение N_3 переписывают в N_1, а заполнение N_4 – в N_2, при этом заполнения N_3, N_4 сохраняются. Заполнение накопителей N_1 и N_2 зашифровывается в режиме простой замены.

Полученное в результате зашифрования заполнение накопителей N_1, N_2 образует первый 64-разрядный блок гаммы шифра $\Gamma_ш^{(1)} = (\gamma_1^{(1)}, \gamma_2^{(1)}, \ldots, \gamma_{63}^{(1)}, \gamma_{64}^{(1)})$, который суммируют поразрядно по модулю 2 в сумматоре $СМ_5$ с первым 64-разрядным блоком открытых данных

$$T_0^{(1)} = (t_1^{(1)}, t_2^{(1)}, \ldots, t_{63}^{(1)}, t_{64}^{(1)}).$$

В результате суммирования по модулю 2 значений $\Gamma_ш^{(1)}$ и $T_0^{(1)}$ получают первый 64-разрядный блок зашифрованнных данных:

$$T_ш^{(1)} = \Gamma_ш^{(1)} \oplus T_0^{(1)} = (\tau_1^{(1)}, \tau_2^{(1)}, \ldots, \tau_{63}^{(1)}, \tau_{64}^{(1)}),$$

где $\tau_i^{(1)} = t_i^{(1)} \oplus \gamma_i^{(1)}$, $i = 1 \ldots 64$.

Для получения следующего 64-разрядного блока гаммы шифра $\Gamma_ш^{(2)}$ заполнение N_4 суммируется по модулю $(2^{32}-1)$ в сумматоре $СМ_4$ с константой C_1 из N_6. Результат записывается в N_4. Заполнение N_3 суммируется по модулю 2^{32} в сумматоре $СМ_3$ с константой C_2 из N_5. Результат записывается в N_3. Новое заполнение N_3 переписывают в N_1, а новое заполнение N_4 – в N_2, при этом заполнения N_3 и N_4 сохраняют. Заполнения N_1, N_2 зашифровывают в режиме простой замены.

Полученное в результате зашифрования заполнение накопителей N_1 и N_2 образует второй 64-разрядный блок гаммы шифра $\Gamma_ш^{(2)}$, который суммируется поразрядно по модулю 2 в сумматоре $СМ_5$ со вторым блоком открытых данных $T_0^{(2)}$:

$$T_ш^{(2)} = Г_ш^{(2)} \oplus T_0^{(2)}.$$

Аналогично вырабатываются блоки гаммы шифра $Г_ш^{(3)}$, $Г_ш^{(4)}$, ..., $Г_ш^{(m)}$ и зашифровываются блоки открытых данных $T_0^{(3)}$, $T_0^{(4)}$, ..., $T_0^{(m)}$.

В канал связи или память ЭВМ передаются синхропосылка $\tilde{S}$ и блоки зашифрованных данных

$$T_ш^{(1)}, T_ш^{(2)}, ..., T_ш^{(m)}.$$

Расшифрование в режиме гаммирования. При расшифровании криптосхема имеет тот же вид, что и при зашифровании (см. рис. 3.12).

Уравнение расшифрования:

$$T_0^{(i)} = T_ш^{(i)} \oplus Г_ш^{(i)} = T_ш^{(i)} \oplus A\,(Y_{i-1} \boxplus C_2,\ Z_{i-1} \boxplus' C_1),\quad i = 1 \ldots m.$$

Следует отметить, что расшифрование данных возможно только при наличии синхропосылки, которая не является секретным элементом шифра и может храниться в памяти ЭВМ или передаваться по каналам связи вместе с зашифрованными данными.

Рассмотрим реализацию процедуры расшифрования. В КЗУ вводят 256 бит ключа, с помощью которого осуществляется зашифрование данных $T_0^{(1)}, T_0^{(2)}, ..., T_0^{(m)}$. В накопители N_1 и N_2 вводится синхропосылка, и осуществляется процесс выработки m блоков гаммы шифра $Г_ш^{(1)}, Г_ш^{(2)}, ..., Г_ш^{(m)}$. Блоки зашифрованных данных $T_ш^{(1)}, T_ш^{(2)}, ..., T_ш^{(m)}$ суммируются поразрядно по модулю 2 в сумматоре $СМ_5$ с блоками гаммы шифра $Г_ш^{(1)}, ..., Г_ш^{(m)}$. В результате получаются блоки открытых данных

$$T_0^{(1)}, T_0^{(2)}, ..., T_0^{(m)};$$

при этом $T_0^{(m)}$ может содержать меньше 64 разрядов.

Режим гаммирования с обратной связью

Зашифрование открытых данных в режиме гаммирования с обратной связью. Криптосхема, реализующая алгоритм зашифрования в режиме гаммирования с обратной связью, имеет вид, показанный на рис. 3.13.

Открытые данные, разбитые на 64-разрядные блоки $T_0^{(1)}$, $T_0^{(2)}$, ..., $T_0^{(m)}$, зашифровываются в режиме гаммирования с обратной связью путем поразрядного сложения по модулю 2 с гаммой шифра $Г_ш$, которая вырабатывается блоками по 64 бита:

$$Г_ш = (Г_ш^{(1)}, Г_ш^{(2)}, ..., Г_ш^{(m)}).$$

Число двоичных разрядов в блоке $T_0^{(m)}$ может быть меньше 64, при этом неиспользованная для шифрования часть гаммы шифра из блока $Г_ш^{(m)}$ отбрасывается.

Уравнения зашифрования в режиме гаммирования с обратной связью имеют вид:

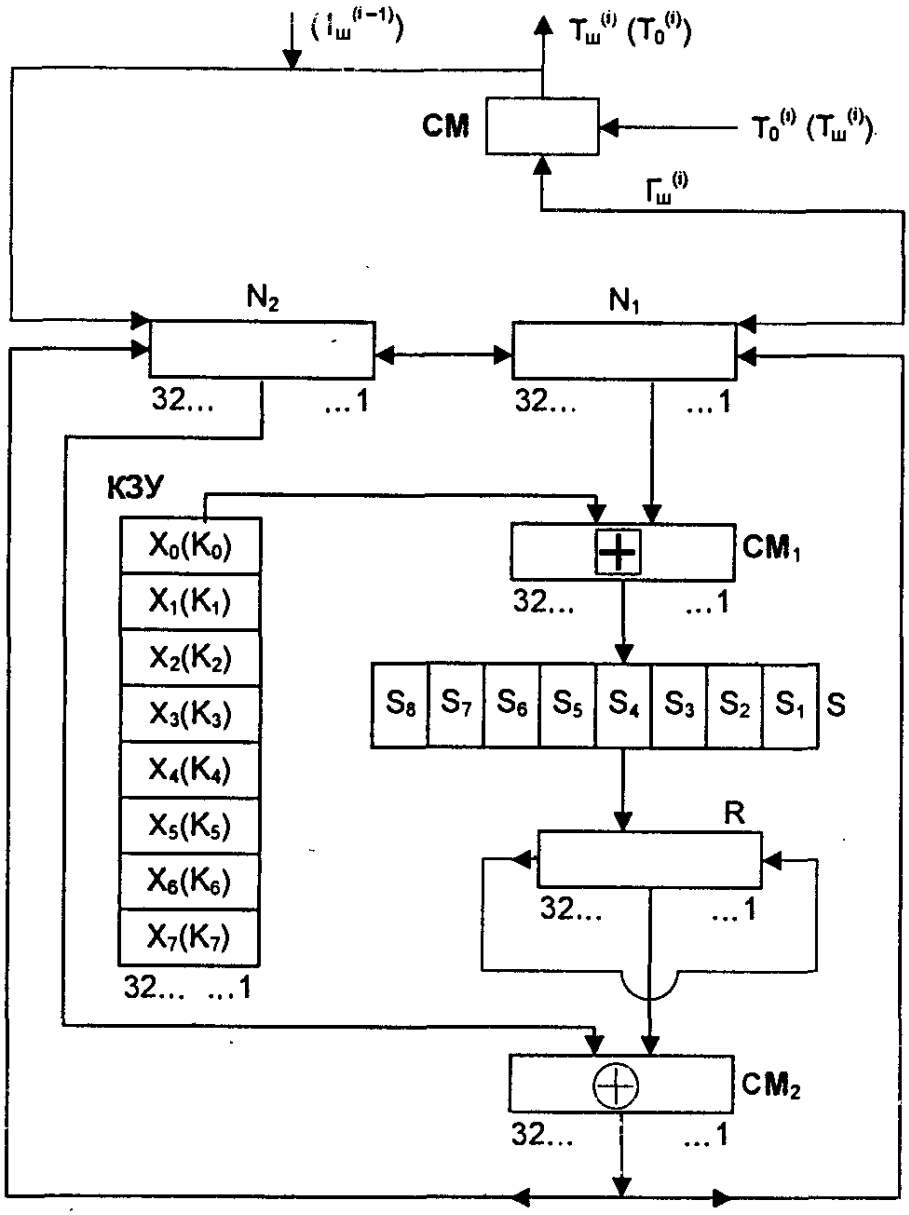


Рис. 3.13. Схема реализации режима гаммирования с обратной связью

$$T_ш^{(1)} = A(\tilde{S}) \oplus T_0^{(1)} = Г_ш^{(1)} \oplus T_0^{(1)},$$

$$T_ш^{(i)} = A(T_ш^{(i-1)}) \oplus T_0^{(i)} = Г_ш^{(i)} \oplus T_0^{(i)}, \quad i = 2 \ldots m.$$

Здесь $T_ш^{(i)}$ – i-й 64-разрядный блок зашифрованного текста; $A(\cdot)$ – функция зашифрования в режиме простой замены; m – определяется объемом открытых данных.

Аргументом функции $A(\cdot)$ на первом шаге итеративного алгоритма является 64-разрядная синхропосылка $\tilde{S}$, а на всех последующих шагах – предыдущий блок зашифрованных данных $T_ш^{(i-1)}$.

Процедура зашифрования данных в режиме гаммирования с обратной связью реализуется следующим образом. В КЗУ вводятся 256 бит ключа. В накопители N_1 и N_2 вводится синхропосылка $\tilde{S} = (S_1, S_2, \ldots, S_{64})$ из 64 бит. Исходное заполнение накопителей N_1

и N_2 зашифровывается в режиме простой замены. Полученное в результате зашифрования заполнение накопителей N_1 и N_2 образует первый 64-разрядный блок гаммы шифра $Г_ш^{(1)} = A(\tilde{S})$, который суммируется поразрядно по модулю 2 в сумматоре $СМ_5$ с первым 64-разрядным блоком открытых данных

$$T_0^{(1)} = (t_1^{(1)}, t_2^{(1)}, ..., t_{64}^{(1)}).$$

В результате получают первый 64-разрядный блок зашифрованных данных

$$T_ш^{(1)} = Г_ш^{(1)} \oplus T_0^{(1)},$$

где $T_ш^{(1)} = (\tau_1^{(1)}, \tau_2^{(1)}, ..., \tau_{64}^{(1)})$.

Блок зашифрованных данных $T_ш^{(1)}$ одновременно является также исходным состоянием накопителей N_1, N_2 для выработки второго блока гаммы шифра $Г_ш^{(2)}$, и поэтому по обратной связи $T_ш^{(1)}$ записывается в указанные накопители N_1 и N_2.

Заполнение накопителя N_1

$$(\tau_{32}^{(1)}, \tau_{31}^{(1)}, ..., \tau_2^{(1)}, \tau_1^{(1)}).$$
$$32, \quad 31, \quad ..., \quad 2, \quad 1 \quad \leftarrow \text{номер разряда } N_1$$

Заполнение накопителя N_2

$$(\tau_{64}^{(1)}, \tau_{63}^{(1)}, ..., \tau_{34}^{(1)}, \tau_{33}^{(1)}).$$
$$32, \quad 31, \quad ..., \quad 2, \quad 1 \quad \leftarrow \text{номер разряда } N_2$$

Заполнение накопителей N_1 и N_2 зашифровывается в режиме простой замены. Полученное в результате зашифрования заполнение накопителей N_1 и N_2 образует второй 64-разрядный блок гаммы шифра $Г_ш^{(2)}$, который суммируется поразрядно по модулю 2 в сумматоре $СМ_5$ со вторым блоком открытых данных $T_0^{(2)}$:

$$Г_ш^{(2)} \oplus T_0^{(2)} = T_ш^{(2)}.$$

Выработка последующих блоков гаммы шифра $Г_ш^{(i)}$ и зашифрование соответствующих блоков открытых данных $T_0^{(i)}$ $(i = 3 ... m)$ производится аналогично.

Если длина последнего m-го блока открытых данных $T_0^{(m)}$ меньше 64 разрядов, то из $Г_ш^{(m)}$ используется только соответствующее число разрядов гаммы шифра, остальные разряды отбрасываются.

В канал связи или память ЭВМ передаются синхропосылка $\tilde{S}$ и блоки зашифрованных данных $T_ш^{(1)}, T_ш^{(2)}, ..., T_ш^{(m)}$.

Расшифрование в режиме гаммирования с обратной связью. При расшифровании криптосхема имеет тот же вид, что и при зашифровании (см. рис. 3.13).

Уравнения расшифрования:

$$T_0^{(1)} = A(\tilde{S}) \oplus T_ш^{(1)} = Г_ш^{(1)} \oplus T_ш^{(1)},$$

$$T_0^{(i)} = Г_ш^{(i)} \oplus T_ш^{(i)} = A(T_ш^{(i-1)}) \oplus T_ш^{(i)}, \quad i = 2 \ldots m.$$

Реализация процедуры расшифрования зашифрованных данных в режиме гаммирования с обратной связью происходит следующим образом. В КЗУ вводят 256 бит того же ключа, на котором осуществлялось зашифрование открытых блоков $T_0^{(1)}, T_0^{(2)}, \ldots, T_0^{(m)}$. В накопители N_1 и N_2 вводится синхропосылка $\tilde{S}$. Исходное заполнение накопителей N_1 и N_2 (синхропосылка $\tilde{S}$) зашифровывается в режиме простой замены. Полученное в результате зашифрования заполнение N_1 и N_2 образует первый блок гаммы шифра

$$Г_ш^{(1)} = A(\tilde{S}),$$

который суммируется поразрядно по модулю 2 в сумматоре $СМ_5$ с блоком зашифрованных данных $T_ш^{(1)}$.

В результате получается первый блок открытых данных

$$T_0^{(1)} = Г_ш^{(1)} \oplus T_ш^{(1)}.$$

Блок зашифрованных данных $T_ш^{(1)}$ является исходным заполнением накопителей N_1 и N_2 для выработки второго блока гаммы шифра $Г_ш^{(2)}$: $Г_ш^{(2)} = A(T_ш^{(1)})$. Полученное заполнение накопителей N_1 и N_2 зашифровывается в режиме простой замены. Образованный в результате зашифрования блок $Г_ш^{(2)}$ суммируется поразрядно по модулю 2 в сумматоре $СМ_5$ со вторым блоком зашифрованных данных $T_ш^{(2)}$. В результате получают второй блок открытых данных. Аналогично в N_1, N_2 последовательно записывают блоки зашифрованных данных $T_ш^{(2)}, T_ш^{(3)}, \ldots, T_ш^{(m)}$, из которых в режиме простой замены вырабатываются блоки гаммы шифра $Г_ш^{(3)}, Г_ш^{(4)}, \ldots, Г_ш^{(m)}$.

Блоки гаммы шифра суммируются поразрядно по модулю 2 в сумматоре $СМ_5$ с блоками зашифрованных данных $T_ш^{(3)}, T_ш^{(4)}, \ldots, T_ш^{(m)}$

В результате получают блоки открытых данных

$$T_0^{(3)}, T_0^{(4)}, \ldots, T_0^{(m)},$$

при этом последний блок открытых данных $T_0^{(m)}$ может содержать меньше 64 разрядов.

Режим выработки имитовставки

Имитовставка – это блок из P бит, который вырабатывают по определенному правилу из открытых данных с использованием ключа и затем добавляют к зашифрованным данным для обеспечения их имитозащиты.

Имитозащита – это защита системы шифрованной связи от навязывания ложных данных.

В стандарте ГОСТ 28147-89 определяется процесс выработки имитовставки, который единообразен для любого из режимов шифрования данных. Имитовставка $И_р$ вырабатывается из блоков открытых данных либо перед шифрованием всего сообщения, ли-

бо параллельно с шифрованием по блокам. Первые блоки открытых данных, которые участвуют в выработке имитовставки, могут содержать служебную информацию (например, адресную часть, время, синхропосылку) и не зашифровываются.

Значение параметра P (число двоичных разрядов в имитовставке) определяется криптографическими требованиями с учетом того, что вероятность навязывания ложных помех равна $1/2^p$.

Для выработки имитовставки открытые данные представляют в виде последовательности 64-разрядных блоков $T_0^{(i)}$, $i = 1 \ldots m$.

Первый блок открытых данных $T_0^{(1)}$ подвергают преобразованию $\tilde{A}(\cdot)$, соответствующему первым 16 циклам алгоритма зашифрования в режиме простой замены. В качестве ключа для выработки имитовставки используют ключ длиной 256 бит, по которому шифруют данные.

Полученное после 16 циклов 64-разрядное число $\tilde{A}(T_0^{(1)})$ суммируют по модулю 2 со вторым блоком открытых данных $T_0^{(2)}$. Результат суммирования $(\tilde{A}(T_0^{(1)}) \oplus T_0^{(2)})$ снова подвергают преобразованию $\tilde{A}(\cdot)$.

Полученное 64-разрядное число $\tilde{A}(\tilde{A}(T_0^{(1)}) \oplus T_0^{(2)})$ суммируют по модулю 2 с третьим блоком $T_0^{(3)}$ и снова подвергают преобразованию $\tilde{A}(\cdot)$, получая 64-разрядное число

$$\tilde{A}(\tilde{A}(\tilde{A}(T_0^{(1)}) \oplus T_0^{(2)}) \oplus T_0^{(3)}), \quad \text{и т.д.}$$

Последний блок $T_0^{(m)}$ (при необходимости дополненный нулями до полного 64-разрядного блока) суммируют по модулю 2 с результатом вычислений на шаге $(m-1)$, после чего зашифровывают в режиме простой замены, используя преобразование $\tilde{A}(\cdot)$.

Из полученного 64-разрядного числа выбирают отрезок $И_p$ (имитовставку) длиной P бит:

$$И_p = [a_{32-p+1}^{(m)}(16), a_{32-p+2}^{(m)}(16), \ldots, a_{32}^{(m)}(16)],$$

где $a_i^{(m)}$ – i-й бит 64-разрядного числа, полученного после 16-го цикла последнего преобразования $\tilde{A}(\cdot)$, $32 - p + 1 \le i \le 32$.

Имитовставка $И_p$ передается по каналу связи или в память ЭВМ в конце зашифрованных данных, т.е.

$$T_ш^{(1)}, T_ш^{(2)}, \ldots, T_ш^{(m)}, И_p.$$

Поступившие к получателю зашифрованные данные

$$T_ш^{(1)}, T_ш^{(2)}, \ldots, T_ш^{(m)}$$

расшифровываются, и из полученных блоков открытых данных $T_0^{(1)}, T_0^{(2)}, \ldots, T_0^{(m)}$ аналогичным образом вырабатывается имитовставка $И_p'$. Эта имитовставка $И_p'$ сравнивается с имитовставкой $И_p$, полученной вместе с зашифрованными данными из канала связи или из памяти ЭВМ. В случае несовпадения имитовставок полученные при расшифровании блоки открытых данных $T_0^{(1)}$, $T_0^{(2)}$, ..., $T_0^{(m)}$ считают ложными.

3.6. Блочные и поточные шифры

Проектирование алгоритмов шифрования данных основано на рациональном выборе функций, преобразующих исходные (незашифрованные) сообщения в шифртекст. Идея непосредственного применения такой функции ко всему сообщению реализуется очень редко. Практически все применяемые криптографические методы связаны с разбиением сообщения на большое число фрагментов (или знаков) фиксированного размера, каждый из которых шифруется отдельно. Такой подход существенно упрощает задачу шифрования, так как сообщения обычно имеют различную длину.

Различают три основных способа шифрования: поточные шифры, блочные шифры и блочные шифры с обратной связью. Для классификации методов шифрования данных следует выбрать некоторое количество характерных признаков, которые можно применить для установления различий между этими методами. Будем полагать, что каждая часть или каждый знак сообщения шифруется отдельно в заданном порядке.

Можно выделить следующие характерные признаки методов шифрования данных [72].

- Выполнение операций с отдельными битами или блоками. Известно, что для некоторых методов шифрования знаком сообщения, над которым производят операции шифрования, является отдельный бит, тогда как другие методы оперируют конечным множеством битов, обычно называемым блоком.
- Зависимость или независимость функции шифрования от результатов шифрования предыдущих частей сообщения.
- Зависимость или независимость шифрования отдельных знаков от их положения в тексте. В некоторых методах знаки шифруются с использованием одной и той же функции независимо от их положения в сообщении, а в других методах, например при поточном шифровании, различные знаки сообщения шифруются с учетом их положения в сообщении. Это свойство называют позиционной зависимостью или независимостью шифра.
- Симметрия или асимметрия функции шифрования. Эта важная характеристика определяет существенное различие между обычными симметричными (одноключевыми) криптосистемами и асимметричными (двухключевыми) криптосистемами с открытым ключом. Основное различие между ними состоит в том, что в асимметричной криптосистеме знания ключа шифрования (или расшифрования) недостаточно для раскрытия соответствующего ключа расшифрования (или шифрования).

В табл. 3.11 приведены типы криптосистем и их основные характеристики.

Таблица 3.11

Основные характеристики криптосистем

Тип криптосистемы	Операции с битами или блоками	Зависимость от предыдущих знаков	Позиционная зависимость	Наличие симметрии функции шифрования
Поточного шифрования	Биты	Не зависит	Зависит	Симметричная
Блочного шифрования	Блоки	Не зависит	Не зависит	Симметричная или несимметричная
С обратной связью от шифртекста	Биты или блоки	Зависит	Не зависит	Симметричная

Поточное шифрование состоит в том, что биты открытого текста складываются по модулю 2 с битами псевдослучайной последовательности. К достоинствам поточных шифров относятся высокая скорость шифрования, относительная простота реализации и отсутствие размножения ошибок. Недостатком является необходимость передачи информации синхронизации перед заголовком сообщения, которая должна быть принята до расшифрования любого сообщения. Это обусловлено тем, что если два различных сообщения шифруются на одном и том же ключе, то для расшифрования этих сообщений требуется одна и та же псевдослучайная последовательность. Такое положение может создать угрозу криптостойкости системы. Поэтому часто используют дополнительный, случайно выбираемый ключ сообщения, который передается в начале сообщения и применяется для модификации ключа шифрования. В результате разные сообщения будут шифроваться с помощью различных последовательностей.

Поточные шифры широко применяются для шифрования преобразованных в цифровую форму речевых сигналов и цифровых данных, требующих оперативной доставки потребителю информации. До недавнего времени такие применения были преобладающими для данного метода шифрования. Это обусловлено, в частности, относительной простотой проектирования и реализации генераторов хороших шифрующих последовательностей. Но самым важным фактором, конечно, является отсутствие размножения ошибок в поточном шифре. Стандартным методом генерирования последовательностей для поточного шифрования является метод, применяемый в стандарте шифрования DES в режиме обратной связи по выходу (режим OFB).

При *блочном шифровании* открытый текст сначала разбивается на равные по длине блоки, затем применяется зависящая от ключа функция шифрования для преобразования блока открытого текста длиной m бит в блок шифртекста такой же длины. Достоинством блочного шифрования является то, что каждый бит блока шифртекста зависит от значений всех битов соответствующего

блока открытого текста, и никакие два блока открытого текста не могут быть представлены одним и тем же блоком шифртекста. Алгоритм блочного шифрования может использоваться в различных режимах. Четыре режима шифрования алгоритма DES фактически применимы к любому блочному шифру: режим прямого шифрования или шифрования с использованием электронной книги кодов ECB (Electronic code Book), шифрование со сцеплением блоков шифртекста CBC (Cipher block chaining), шифрование с обратной связью по шифртексту CFB (Cipher feedback) и шифрование с обратной связью по выходу OFB (Output feedback).

Основным достоинством прямого блочного шифрования ECB является то, что в хорошо спроектированной системе блочного шифрования небольшие изменения в шифртексте вызывают большие и непредсказуемые изменения в соответствующем открытом тексте, и наоборот. Вместе с тем применение блочного шифра в данном режиме имеет серьезные недостатки. Первый из них заключается в том, что вследствие детерминированного характера шифрования при фиксированной длине блока 64 бита можно осуществить криптоанализ шифртекста "со словарем" в ограниченной форме. Это обусловлено тем, что идентичные блоки открытого текста длиной 64 бита в исходном сообщении представляются идентичными блоками шифртекста, что позволяет криптоаналитику сделать определенные выводы о содержании сообщения. Другой потенциальный недостаток этого шифра связан с размножением ошибок. Результатом изменения только одного бита в принятом блоке шифртекста будет неправильное расшифрование всего блока. Это, в свою очередь, приведет к появлению искаженных битов (от 1 до 64) в восстановленном блоке исходного текста.

Из-за отмеченных недостатков блочные шифры редко применяются в указанном режиме для шифрования длинных сообщений. Однако в финансовых учреждениях, где сообщения часто состоят из одного или двух блоков, блочные шифры широко используют в режиме прямого шифрования. Такое применение обычно связано с возможностью частой смены ключа шифрования, поэтому вероятность шифрования двух идентичных блоков открытого текста на одном и том же ключе очень мала.

Криптосистема с открытым ключом также является системой блочного шифрования и должна оперировать блоками довольно большой длины. Это обусловлено тем, что криптоаналитик знает открытый ключ шифрования и мог бы заранее вычислить и составить таблицу соответствия блоков открытого текста и шифртекста. Если длина блоков мала, например 30 бит, то число возможных блоков не слишком большое (при длине 30 бит это $2^{30} = \sim 10^9$), и

может быть составлена полная таблица, позволяющая моментально расшифровать любое сообщение с использованием известного открытого ключа. Асимметричные криптосистемы с открытым ключом подробно разбираются в следующей главе.

Наиболее часто блочные шифры применяются *в системах шифрования с обратной связью.* Системы шифрования с обратной связью встречаются в различных практических вариантах. Как и при блочном шифровании, сообщения разбивают на ряд блоков, состоящих из m бит. Для преобразования этих блоков в блоки шифртекста, которые также состоят из m бит, используются специальные функции шифрования. Однако если в блочном шифре такая функция зависит только от ключа, то в блочных шифрах с обратной связью она зависит как от ключа, так и от одного или более предшествующих блоков шифртекста.

Практически важным шифром с обратной связью является шифр со сцеплением блоков шифртекста CBC. В этом случае m бит предыдущего шифртекста суммируются по модулю 2 со следующими m битами открытого текста, а затем применяется алгоритм блочного шифрования под управлением ключа для получения следующего блока шифртекста. Еще один вариант шифра с обратной связью получается из стандартного режима CFB алгоритма DES, т. е. режима с обратной связью по шифртексту.

Достоинством криптосистем блочного шифрования с обратной связью является возможность применения их для обнаружения манипуляций сообщениями, производимых активными перехватчиками. При этом используется факт размножения ошибок в таких шифрах, а также способность этих систем легко генерировать код аутентификации сообщений. Поэтому системы шифрования с обратной связью используют не только для шифрования сообщений, но и для их аутентификации. Криптосистемам блочного шифрования с обратной связью свойственны некоторые недостатки. Основным из них является размножение ошибок, так как один ошибочный бит при передаче может вызвать ряд ошибок в расшифрованном тексте. Другой недостаток связан с тем, что разработка и реализация систем шифрования с обратной связью часто оказываются более трудными, чем систем поточного шифрования.

На практике для шифрования длинных сообщений применяют поточные шифры или шифры с обратной связью. Выбор конкретного типа шифра зависит от назначения системы и предъявляемых к ней требований.

3.7. Криптосистема с депонированием ключа

Общие сведения

Криптосистема с депонированием ключа предназначена для шифрования пользовательского трафика (например, речевого или передачи данных) таким образом, чтобы сеансовые ключи, используемые для зашифрования и расшифрования трафика, были доступны при определенных чрезвычайных обстоятельствах авторизованной третьей стороне [97, 117, 121].

По существу, криптосистема с депонированием ключа реализует новый метод криптографической защиты информации, обеспечивающий высокий уровень информационной безопасности при передаче по открытым каналам связи и отвечающий требованиям национальной безопасности. Этот метод основан на применении специальной шифрующей/дешифрующей микросхемы типа Clipper и процедуры депонирования ключа, определяющей дисциплину раскрытия уникального ключа этой микросхемы. Микросхема Clipper разработана по технологии TEMPEST, препятствующей считыванию информации с помощью внешних воздействий.

Генерация и запись уникального ключа в микросхему выполняется до встраивания микросхемы в конечное устройство. Следует отметить, что не существует способа, позволяющего непосредственно считывать этот ключ как во время, так и по завершении технологического процесса производства и программирования данной микросхемы.

Ключ разделяется на два компонента, каждый из которых шифруется и затем передаётся на хранение доверенным Агентам Депозитной Службы, которые представляют собой правительственные организации, обеспечивающие надёжное хранение компонентов ключа в течение срока его действия. Агенты Депозитной Службы выдают эти компоненты ключа только по соответствующему запросу, подтверждённому решением Федерального Суда. Полученные компоненты ключа позволяют службам, отвечающим за национальную безопасность, восстановить уникальный ключ и выполнить расшифрование пользовательского трафика.

В 1994 г. в США был введён новый стандарт шифрования с депонированием ключа EES (Escrowed Encryption Standard) [97]. Стандарт EES предназначен для защиты информации, передаваемой по коммутируемым телефонным линиям связи ISDN (Integrated Services Digital Network) и радиоканалам, включая голосовую информацию, факс и передачу данных со скоростями стандартных коммерческих модемов. Стандарт EES специфицирует алгоритм криптографического преобразования SkipJack с 80-битовым ключом и метод вычисления специального поля доступа LEAF

(Law Enforcement Access Field), позволяющего впоследствии раскрыть секретный ключ в целях контроля трафика при условии соблюдения законности. Алгоритм SkipJack и метод вычисления поля LEAF должны быть реализованы на базе микросхемы типа Clipper. Этот стандарт специфицирует уникальный идентификатор (серийный номер) микросхемы UID (Device Unique Identifier), уникальный ключ микросхемы KU (Device Unique Key) и общий для семейства микросхем ключ KF (Family Key). Вся эта информация записывается в микросхему после её производства, но до встраивания в конкретное устройство. Хотя ключ KU не используется непосредственно для шифрования информационных потоков между отправителем и получателем, объединение его с полем доступа LEAF и ключом KF позволяет восстановить сеансовый ключ KS и выполнить дешифрование.

Криптоалгоритм SkipJack. Криптоалгоритм SkipJack преобразует 64-битовый входной блок в 64-битовый выходной блок с помощью 80-битового секретного ключа. Поскольку один и тот же ключ используется для шифрования и расшифрования, этот алгоритм относится к классу симметричных криптосистем. Отметим, что размер блока в алгоритме SkipJack такой же, как и в DES, но при этом используется более длинный ключ.

Алгоритм SkipJack может функционировать в одном из четырёх режимов, введённых для стандарта DES:

- электронной кодовой книги (ECB);
- сцепления блоков шифртекста (CBC);
- 64-битовой обратной связи по выходу (OFB);
- обратной связи по шифртексту (CFB) для 1-, 8-, 16-, 32- или 64-битовых блоков.

Алгоритм SpikJack был разработан непосредственно Агентством Национальной Безопасности (АНБ) США и засекречен, чтобы сделать невозможной разработку программной или аппаратной реализации процедур шифрования и расшифрования отдельно от процедуры депонирования ключа. Практическая криптостойкость алгоритма SkipJack была подтверждена группой независимых экспертов-криптографов.

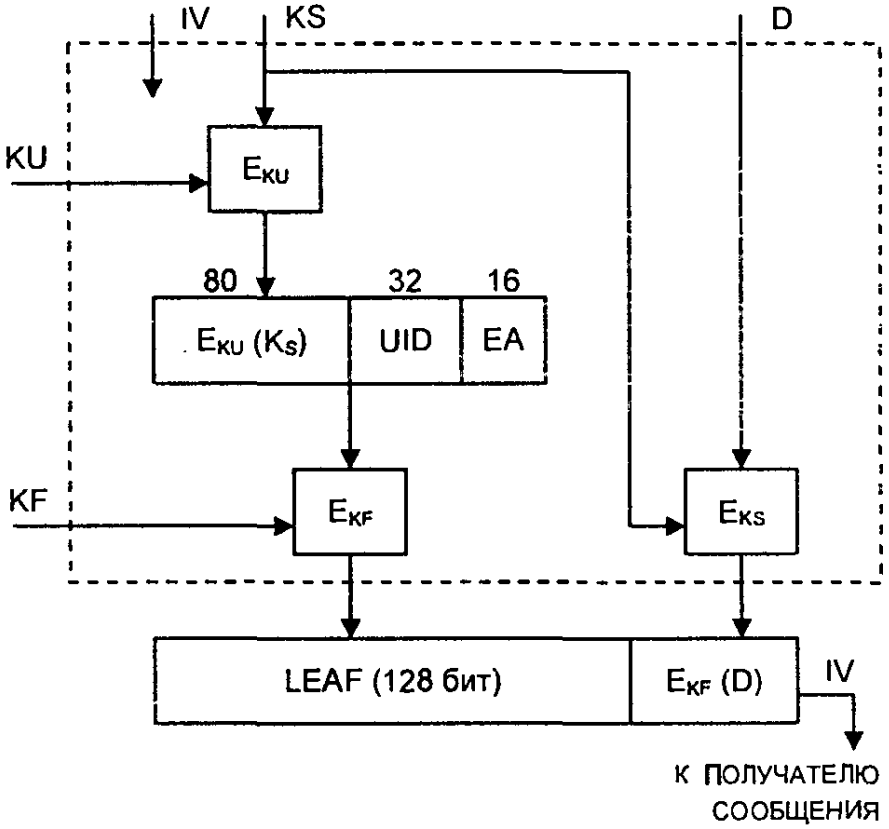
Метод вычисления поля LEAF. Поле доступа LEAF передаётся получателю в начале каждого сеанса связи и содержит секретный сеансовый ключ шифрования/расшифрования KS (Session Key). Только официальные лица, имеющие законное разрешение, могут получить сеансовый ключ KS и дешифровать все ранее зашифрованные на нём сообщения. Хотя ключ KS передаётся в поле LEAF, последнее используется исключительно для контроля над соблюдением законности и не предназначено для распределения ключей абонентам. Микросхема Clipper, установленная в принимающем устройстве, не позволяет извлечь ключ KS из информации в поле LEAF.

Поле доступа LEAF вычисляется как функция от сеансового ключа KS, вектора инициализации IV (Initialization Vector), идентификатора микросхемы UID и уникального ключа KU. Поле LEAF состоит из ключа KS, зашифрованного на ключе KU (80 бит), идентификатора UID (32 бита) и 16-битового аутентификатора EA (Escrow Authenticator). Формирование содержимого поля LEAF завершается шифрованием указанной информации на ключе семейства микросхем KF, т.е.

$$LEAF = E_{KF}(E_{KU}(KS), UID, EA).$$

Таким образом, сеансовый ключ KS закрыт двойным шифрованием (рис. 3.14) и может быть раскрыт в результате последовательного расшифрования на ключах KF и KU.

Детали алгоритма вычисления LEAF засекречены, включая режим шифрования алгоритма SkipJack и метод вычисления аутентификатора EA. Аутентификатор EA позволяет контролировать целостность и защищает поле LEAF от навязывания ложной информации.



Обозначения:

KS – сеансовый ключ
IV – вектор инициализации
D – передаваемые данные
KU – уникальный ключ микросхемы
UID – идентификатор микросхемы
EA – аутентификатор поля LEAF
KF – ключ семейства

Рис. 3.14. Вычисление LEAF и формирование зашифрованного сообщения

Способ применения. Для защиты телефонных переговоров каждый из абонентов должен иметь специальное криптографическое устройство, содержащее Clipper, Capstone или другую аналогичную микросхему (рис. 3.15). В этом устройстве должен быть реализован протокол, позволяющий абонентам обмениваться секретным сеансовым ключом KS, например, с помощью известного метода "цифрового конверта" (digital envelope).

Данный метод применяется в устройстве защиты телефонных переговоров TSD (3600 Telephone Security Device) компании AT&T. Оно подключается встык между телефонной трубкой и основным блоком и активизируется нажатием кнопки. После установления ключевого синхронизма ключ KS вместе с вектором инициализации IV подаётся на вход микросхемы для вычисления LEAF. Затем LEAF вместе с IV передается принимающей стороне для проверки и синхронизации микросхем на передающем и приёмном концах. После синхронизации микросхем сеансовый ключ KS используется для шифрования и расшифрования данных (речь предварительно оцифровывается) в обоих направлениях.

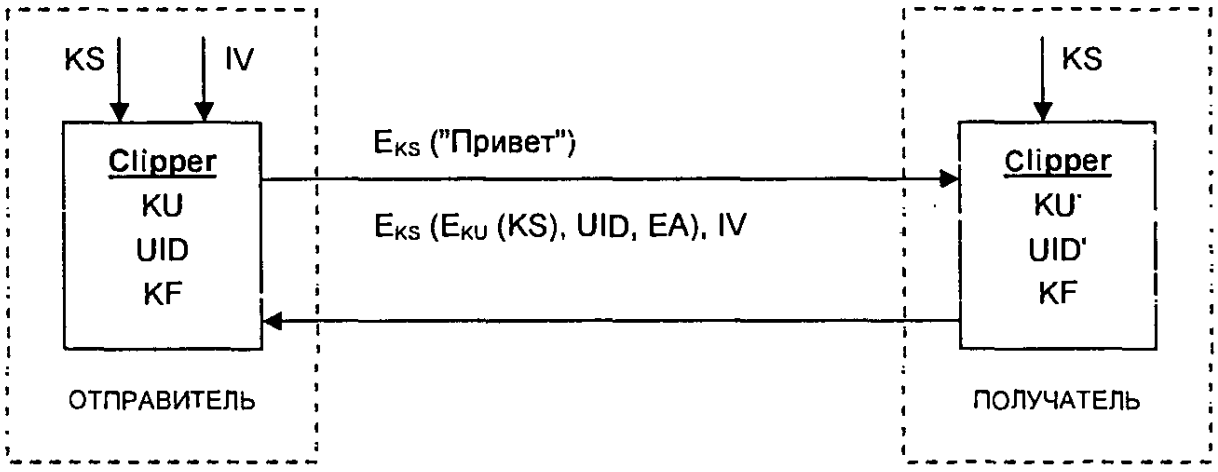


Рис. 3.15. Схема защиты телефонных переговоров с использованием микросхем Clipper

В дуплексном и полудуплексном режимах связи каждое криптографическое устройство передаёт свою уникальную пару IV и LEAF. При этом оба устройства используют один и тот же сеансовый ключ KS для шифрования и расшифрования.

Первая партия микросхем для криптографических устройств с депонированием ключа была изготовлена компанией VLSI Technology Inc. и запрограммирована фирмой Mykotronx. В устройстве TSD используется микросхема МУК-78Т (Mykotronx) с быстродействием 21 Мбит/с.

Рассмотрим подробнее функционирование криптосистемы с депонированием ключа и необходимую для её работы инфраструктуру.

Процедура генерации ключей

В процедуре генерации ключей участвуют Национальный Менеджер Программы; два Агента Депозитной Службы; два Агента KF-Службы, отвечающие за формирование и хранение ключа семейства KF; представитель Службы Программирования микросхем (Programming Facility) [97, 117].

До генерации уникального ключа микросхемы KU и соответствующих ему ключевых компонентов KC1 и KC2 необходимо сгенерировать вспомогательные ключи KN1 и KN2 и случайные числа RS1 и RS2 (Random Seeds). В процедуре генерации используется специальная смарт-карта, на которой в соответствии со стандартом X9.17 (FIPS 171) реализован генератор псевдослучайных чисел. Начальное значение генератора формируется как результат вычисления хэш-функции от последовательности случайных символов, введённых с клавиатуры, временных интервалов между нажатиями при вводе символов с клавиатуры и текущего времени суток.

Описанная выше процедура используется Агентами Депозитной Службы для генерации ключевых чисел KN1, KN2 и случайных чисел RS1, RS2. Назначение вспомогательных ключевых чисел и случайных чисел будет пояснено в последующих разделах.

Компоненты ключа семейства KF. Агенты KF-Службы, отвечающие за формирование ключа семейства KF, генерируют компоненты KFC1 и KFC2 на отдельных операционных пунктах.

По две копии каждого компонента записываются на магнитные носители. Каждый магнитный носитель помещают в специальный пронумерованный контейнер. Контейнер с копией каждого компонента на магнитном носителе помещается в отдельный сейф.

Ключевые и случайные числа. Каждый Агент Депозитной Службы генерирует и записывает на магнитные носители четыре копии ключевых чисел KN1 и KN2. Каждый магнитный носитель помещается в специальный пронумерованный контейнер. Контейнер с копией каждого компонента на магнитном носителе помещают в отдельный сейф, установленный в операционном пункте Агента. Кроме того, каждый Агент генерирует и записывает на магнитный носитель одно случайное число RS. Магнитный носитель в контейнере помещается в сейф.

Каждый Офицер Депозитной Службы (представитель Агента) доставляет в Службу Программирования микросхем две копии ключевого числа (KN1 или KN2) и одну копию случайного числа (RS1 или RS2).

Программирование микросхемы

Программирование микросхемы осуществляется внутри специального бокса SCIF (Sensitive Compartmented Information Facility). Операция программирования выполняется с санкции Национального Менеджера Программы и требует участия:

- Офицеров Депозитной Службы (по одному от каждого Агента);
- двух Офицеров, представляющих Агентов KF-Службы (далее – Офицеров KF-Службы);
- Представителя Службы Программирования микросхем [97].

Служба Программирования использует:

- автоматизированное рабочее место (под управлением ОС UNIX) для генерации уникального ключа микросхемы KU и его компонентов KC1 и KC2;
- персональный компьютер для контроля процесса программирования микросхемы;
- устройство программирования микросхем IMS (Integrated Measurement System) с производительностью порядка 120 микросхем в час.

Инициализация. Подготовительные действия перед каждой процедурой включают передачу (секретной почтой) в Службу Программирования по одной копии каждого компонента (KFC1, KFC2) ключа семейства KF. Контейнеры с магнитными носителями помещают в сейф с двойным замком Службы Программирования.

Офицеры Депозитной Службы доставляют в Службу Программирования свои контейнеры с ключевой информацией (KN1, KN2, RS1, RS2). Затем два Офицера Депозитной Службы (по одному от каждого Агента) отпирают сейф.

Представитель Службы Программирования, действующий по доверенности от Агентов KF-Службы, извлекает из сейфа магнитные носители с компонентами KFC1 и KFC2. Следуя установленной процедуре, магнитные носители с компонентами KFC1 и KFC2 вставляются в считывающее устройство автоматизированного рабочего места.

Ключ семейства KF вычисляется путем побитового сложения по модулю 2 компонент KFC1 и KFC2:

$$KF = KFC1 \oplus KFC2,$$

Затем Офицеры Депозитной Службы вводят значения KN1, KN2, RS1, RS2 и произвольные последовательности символов AI1 и AI2 с клавиатуры.

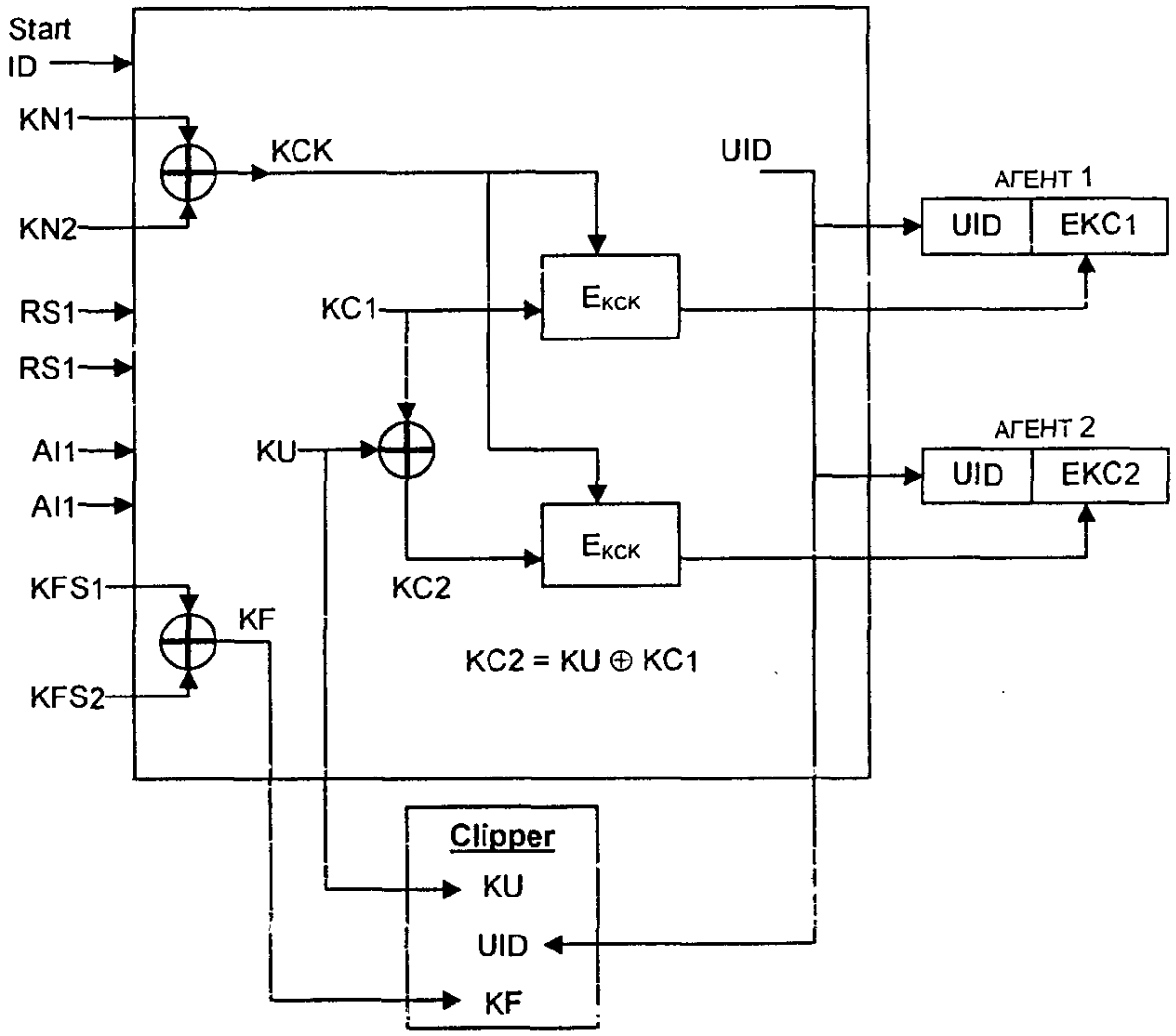
Ключевые числа KN1 и KN2 побитово суммируются по модулю 2 для вычисления ключа шифрования KCK (Key Component Enciphering Key):

$$KCK = KN1 \oplus KN2.$$

Ключ КСК предназначен для шифрования компонентов КС1 и КС2 ключа KU.

Офицеры Депозитной Службы вводят также начальный серийный номер (Start UID) для формирования уникального идентификатора микросхемы UID. Процедура генерации и программирования микросхемы иллюстрируется на рис. 3.16.

Генерация ключа KU. Случайные числа RS1, RS2 и произвольные последовательности символов AI1, AI2 используются для вычисления пары значений. Одно из этих значений служит для формирования ключа KU, а другое – для формирования компоненты КС1.



Обозначения:

UID	– уникальный идентификатор микросхемы
KU	– уникальный ключ микросхемы
KN1, KN2	– ключевые числа для генерации КСК
КСК	– ключ шифрования компонент КС1 и КС2
КС1, КС2	– ключевые компоненты
EKC1, EKC2	– зашифрованные ключевые компоненты
RS1, RS2	– случайные числа
AI1, AI2	– произвольные последовательности
KF	– ключ семейства
KFC1, KFC2	– компоненты ключа семейства

Рис. 3.16. Генерация ключа и программирование микросхемы

Далее ключ KU и компонента KC1 побитово суммируются по модулю 2 для вычисления компонента KC2:

$$KC2 = KU \oplus KC1.$$

Таким образом, ключ KU может быть вычислен как сумма компонентов KC1 и KC2:

$$KU = KC1 \oplus KC2.$$

Затем компоненты KC1 и KC2 шифруются на ключе KCK.

Пара KU/UID подаётся на вход устройства программирования IMS и вместе с ключом KF записывается в микросхему.

Зашифрованные компоненты KC1 и KC2 (т.е. EKC1, EKC2) вместе с UID записываются на четыре магнитных носителя (по две копии каждого компонента), упаковываются в пронумерованные контейнеры и помещаются в сейф с двойным замком до момента завершения процедуры программирования микросхемы.

Уничтожение и транспортировка ключевых компонентов. В обязанности Офицеров входит также активизация специальной программы, стирающей ключевую информацию с магнитных накопителей и оперативной памяти. По завершении этой процедуры Офицеры Депозитной Службы независимо друг от друга доставляют в депозитарий первого Агента контейнер с магнитным носителем, содержащим компонент EKC1 и UID, и в депозитарий второго Агента – компонент EKC2 и UID. До того как покинуть SCIF, Офицеры Депозитной Службы регистрируют свои действия в специальном журнале.

Обслуживание ключей

Агенты Депозитной Службы помещают копии ключевых компонентов в отдельный сейф с двойным замком. Для отпирания такого сейфа требуется участие двух лиц. Таким образом, надежность депозитария обеспечивается за счет двойного контроля, физической безопасности, криптографических средств и резервирования.

После доставки ключевых компонентов EKC1 и EKC2 в депозитарий каждый из двух Офицеров проверяет целостность контейнеров и их номеров. Если контейнеры не были скомпрометированы, Офицеры выполняют их регистрацию и помещают копию регистрационной записи вместе с контейнерами в сейф. Эти контейнеры с ключевыми компонентами хранятся в сейфах до тех пор, пока не будет получена санкция на их извлечение [97].

Процедура выдачи ключевых компонентов. Ключевые компоненты выдаются только с санкции Федерального Суда и в соответствии с процедурой, установленной Генеральным Прокурором. Эта процедура предполагает формирование специальных

запросов и представление их Агентам Депозитной Службы. Назначение запроса заключается в установлении факта легальности расследования со стороны запрашивающего органа, законности расследования, определения сроков и т.д. Запрос включает также идентификатор UID и серийный номер Процессора Дешифрования (Key Escrow Decryption Processor). В случае, если запрос принят, Агенты Депозитной Службы выдают ключевые компоненты, соответствующие заданному UID. Следует отметить, что должна быть обеспечена гарантия того, что по истечении срока расследования эти ключевые компоненты не смогут быть повторно использованы в тех же целях.

Извлечение и транспортировка ключевых компонентов. Получив официальное разрешение на выдачу ключевого компонента, соответствующего одному или более UID, Агент Депозитной Службы дает указание своим Офицерам открыть один из сейфов и извлечь ключевой компонент. Поскольку сейф имеет двойной замок, для его отпирания необходимо участие двух Офицеров. Помимо ключевого компонента (EKC1 или EKC2) из сейфа извлекаются ключевые числа (KN1, KN2), необходимые для формирования ключа дешифрования КСК. Факт извлечения ключевого компонента регистрируется в журнале.

Офицеры извлекают магнитные носители из контейнеров и в соответствии с запросом Программы Извлечения Ключа (Key Extract Program) вставляют их в считывающее устройство персонального компьютера. Эта Программа идентифицирует ключевой компонент по заданному UID и копирует его на отдельный магнитный носитель. По завершении процесса копирования все магнитные носители убираются в контейнеры. Все контейнеры, кроме контейнера с копией зашифрованного ключевого компонента и ключевым числом, помещаются в сейф.

В результате два Офицера от каждого Агента Депозитной Службы доставляют контейнеры с копией зашифрованного ключевого компонента и ключевыми числами на специальный операционный пункт Службы Дешифрования. Права доступа на операционный пункт Службы Дешифрования подтверждаются процедурой авторизации.

Процедура дешифрования

Поставщики телекоммуникационных услуг обязаны предоставлять компетентным органам доступ к каналам связи в том случае, если необходимость этого подтверждается соответствующим судебным решением. Обычной практикой является предоставление выделенной линии связи для передачи перехваченных шифртекстов на операционный пункт Службы Дешифрования.

Дешифрующий Процессор, установленный на операционном пункте, представляет собой персональный компьютер со специально разработанной платой. Запуск этого компьютера выполняется только после ввода ключа с Touch Memory. Дешифрующий Процессор узко специализирован, функционально ограничен и предназначен для решения конкретных задач дешифрования [97, 117].

При обработке речевой информации необходимо дополнительное оборудование для преобразования цифрового сигнала в аналоговый.

Инициализация дешифрующего процессора. Перед тем как Дешифрующий Процессор будет использован по назначению, необходимо выполнить его инициализацию – ввести ключ семейства KF. Для этого два Офицера от каждого Агента KF-Службы доставляют компоненты ключа семейства на операционный пункт Службы Дешифрования. Далее компоненты KFC1 и KFC2 вводятся в Дешифрующий Процессор для формирования ключа KF путём их суммирования по модулю 2:

$$KF = KFC1 \oplus KFC2.$$

Извлечение LEAF и UID. Процедура дешифрования информации отправителя иллюстрируется на рис. 3.17. Дешифрующий Процессор выделяет LEAF отправителя и получателя из зашифрованного информационного потока и затем выполняет его дешифрование на ключе семейства KF с целью получения UID. Несмотря на то, что Дешифрующий Процессор выделяет два, возможно различных, идентификатора UID микросхем на приёмном и передающем концах, сеансовый ключ KS используется для шифрования/дешифрования в обоих направлениях.

Полученный в результате дешифрования UID вместе с запросом передаются Агентам Депозитной Службы с целью получения ключевых компонентов.

Загрузка ключевых компонентов и ключевых чисел. После доставки магнитных носителей с ключевыми числами (KN1, KN2) и копий зашифрованного ключевого компонента (EKC1, EKC2) Офицеры Депозитной Службы проверяют соответствие серийного номера Дешифрующего Процессора номеру, указанному в запросе (см. раздел "Процедура выдачи ключевых компонент"). Если номера идентичны, Офицеры извлекают магнитные носители из контейнеров и в соответствии с процедурой вставляют их в считывающее устройство Дешифрующего Процессора. Кроме того, в Дешифрующий Процессор вводится информация о временном интервале, в течение которого ключевой материал может быть использован на законном основании.

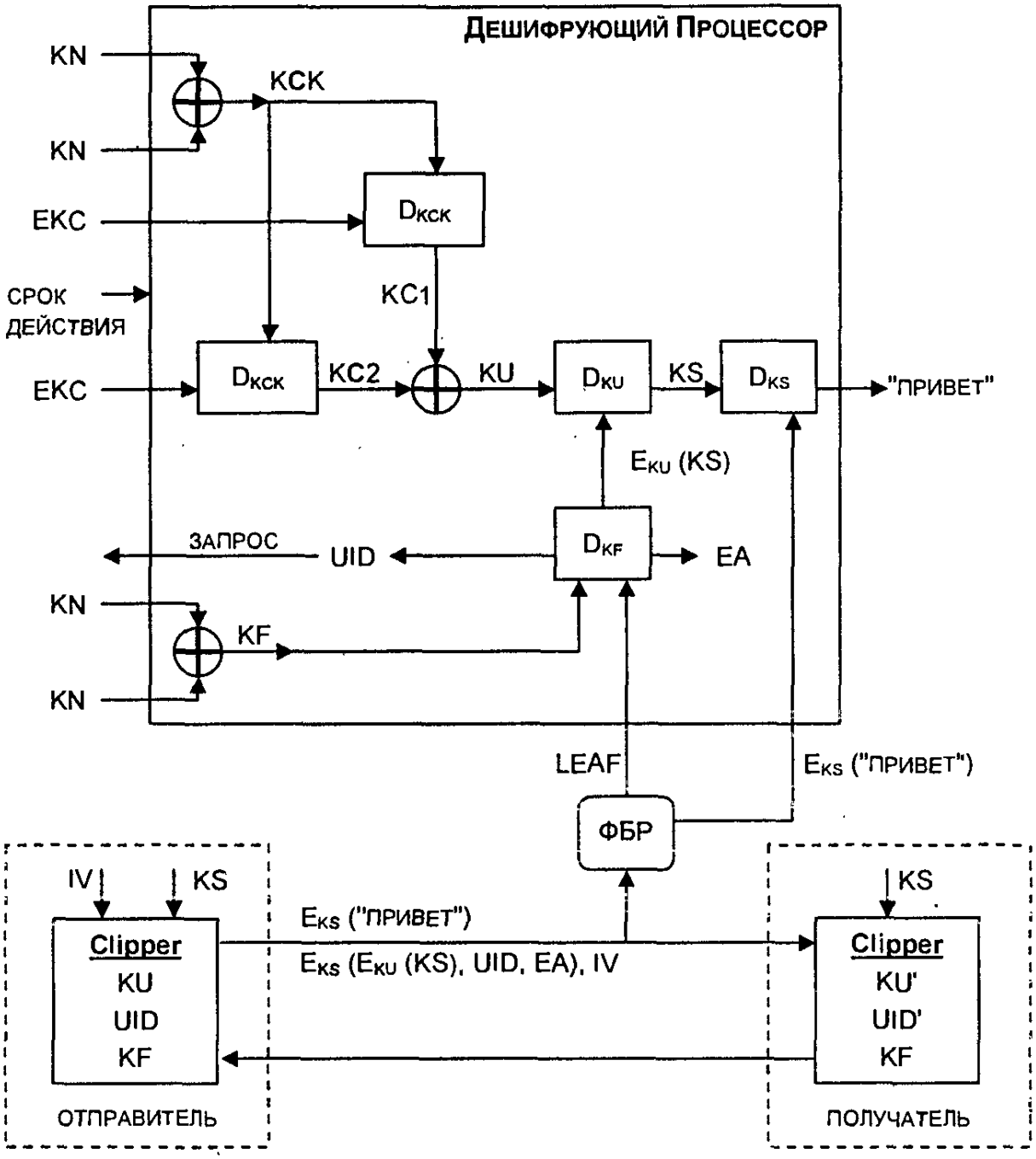


Рис. 3.17. Схема процедуры дешифрования

Дешифрующий Процессор выполняет суммирование по модулю 2 ключевых чисел KN1 и KN2 для вычисления значения ключа КСК. После дешифрования ЕКС1 и ЕКС2 на ключе КСК и получения компонентов КС1 и КС2 последние суммируются по модулю 2 для получения ключа KU.

После завершения процедуры загрузки копии ключевых компонентов, доставленные Офицерами Депозитной Службы, уничтожаются, а контейнер с ключевыми числами (KN1, KN2) доставляется обратно в депозитарий Агента.

Дешифрование. Раскрытие ключа KU конкретной микросхемы позволяет дешифровать любой шифртекст, полученный с помощью этой микросхемы. Для этого достаточно перехватить LEAF.

передаваемое в начале каждого сеанса связи, затем дешифровать LEAF на ключе KF и получить UID и зашифрованный сеансовый ключ KS.

Следующий шаг заключается в раскрытии сеансового ключа KS путём дешифрования на ключе KU и проверке аутентификатора EA. Правильность EA свидетельствует о том, что ключ KS восстановлен корректно и может быть использован для дешифрования информации в обоих направлениях.

Полученные в результате дешифрования речевые данные в цифровой форме преобразуются в сигнал тональной частоты с помощью цифроаналогового преобразователя. Ранее перехваченная зашифрованная информация (до раскрытия KU) также может быть дешифрована. Если ключ KU известен, быстродействие аппаратуры позволяет осуществлять прослушивание телефонных переговоров в реальном масштабе времени.

По истечении установленного срока действия выдаётся команда уничтожения ключа KU, хранящегося в памяти Дешифрующего Процессора. Уничтожение этого ключа подтверждается аутентичным сообщением, посылаемым каждому Агенту Депозитной Службы. Поэтому применение ключа после истечения срока будет обнаружено при аудиторской проверке.

Развитие этой криптосистемы заключается в автоматизации большинства ручных процедур, в первую очередь транспортировки ключей и регистрации.

ГЛАВА 4. АСИММЕТРИЧНЫЕ КРИПТОСИСТЕМЫ

4.1. Концепция криптосистемы с открытым ключом

Эффективными системами криптографической защиты данных являются асимметричные криптосистемы, называемые также криптосистемами с открытым ключом. В таких системах для зашифрования данных используется один ключ, а для расшифрования – другой ключ (отсюда и название – асимметричные). Первый ключ является открытым и может быть опубликован для использования всеми пользователями системы, которые зашифровывают данные. Расшифрование данных с помощью открытого ключа невозможно.

Для расшифрования данных получатель зашифрованной информации использует второй ключ, который является *секретным*. Разумеется, ключ расшифрования не может быть определен из ключа зашифрования.

Обобщенная схема асимметричной криптосистемы с открытым ключом показана на рис. 4.1. В этой криптосистеме применяют два различных ключа: $K_в$ – открытый ключ отправителя A; $k_в$ – секретный ключ получателя B. Генератор ключей целесообразно располагать на стороне получателя B (чтобы не пересылать секретный ключ $k_в$ по незащищенному каналу). Значения ключей $K_в$ и $k_в$ зависят от начального состояния генератора ключей.

Раскрытие секретного ключа $k_в$ по известному открытому ключу $K_в$ должно быть вычислительно неразрешимой задачей.

Характерные особенности асимметричных криптосистем:

1. Открытый ключ $K_в$ и криптограмма C могут быть отправлены по незащищенным каналам, т.е. противнику известны $K_в$ и C.

2. Алгоритмы шифрования и расшифрования

$$E_в : M \to C,$$

$$D_в : C \to M$$

являются открытыми.

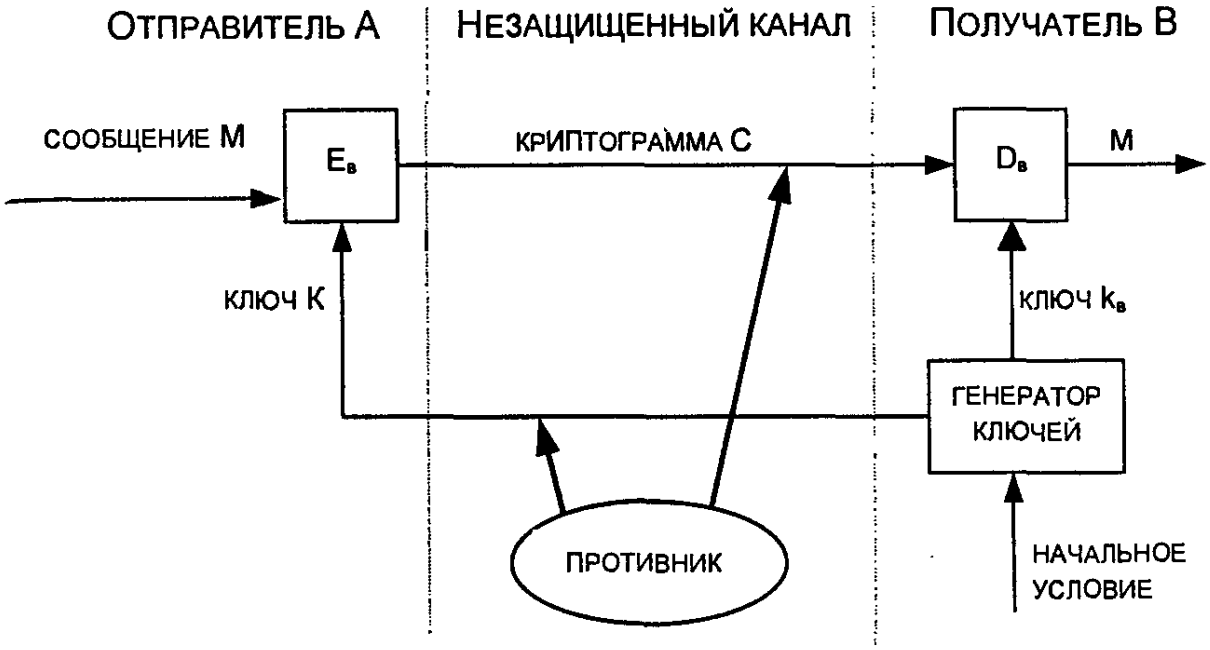


Рис. 4.1. Обобщенная схема асимметричной криптосистемы с открытым ключом

Защита информации в асимметричной криптосистеме основана на секретности ключа $k_в$.

У. Диффи и М. Хеллман сформулировали требования, выполнение которых обеспечивает безопасность асимметричной криптосистемы:

1. Вычисление пары ключей ($К_в$, $k_в$) получателем В на основе начального условия должно быть простым.

2. Отправитель А, зная открытый ключ $К_в$ и сообщение М, может легко вычислить криптограмму

$$C = E_{K_в}(M) = E_в(M). \qquad (4.1)$$

3. Получатель В, используя секретный ключ $k_в$ и криптограмму С, может легко восстановить исходное сообщение

$$M = D_{k_в}(C) = D_в(C) = D_в[E_в(M)]. \qquad (4.2)$$

4. Противник, зная открытый ключ $К_в$, при попытке вычислить секретный ключ $k_в$ наталкивается на непреодолимую вычислительную проблему.

5. Противник, зная пару ($К_в$, С), при попытке вычислить исходное сообщение М наталкивается на непреодолимую вычислительную проблему [28].

4.2. Однонаправленные функции

Концепция асимметричных криптографических систем с открытым ключом основана на применении однонаправленных функций. Неформально однонаправленную функцию можно опре-

делить следующим образом. Пусть X и Y – некоторые произвольные множества. Функция

$$f : X \to Y$$

является однонаправленной, если для всех $x \in X$ можно легко вычислить функцию

$$y = f(x), \quad \text{где} \quad y \in Y.$$

И в то же время для большинства $y \in Y$ достаточно сложно получить значение $x \in X$, такое, что $f(x) = y$ (при этом полагают, что существует по крайней мере одно такое значение x).

Основным критерием отнесения функции f к классу однонаправленных функций является отсутствие эффективных алгоритмов обратного преобразования $Y \to X$.

В качестве первого примера однонаправленной функции рассмотрим целочисленное умножение. Прямая задача – вычисление произведения двух очень больших целых чисел P и Q, т.е. нахождение значения

$$N = P * Q, \tag{4.3}$$

является относительно несложной задачей для ЭВМ.

Обратная задача – разложение на множители большого целого числа, т.е. нахождение делителей P и Q большого целого числа $N = P * Q$, является практически неразрешимой задачей при достаточно больших значениях N. По современным оценкам теории чисел при целом $N \approx 2^{664}$ и $P \approx Q$ для разложения числа N потребуется около 10^{23} операций, т.е. задача практически неразрешима на современных ЭВМ.

Следующий характерный пример однонаправленной функции – это модульная экспонента с фиксированными основанием и модулем. Пусть A и N – целые числа, такие, что $1 \le A < N$. Определим множество Z_N:

$$Z_N = \{0, 1, 2, \ldots, N-1\}.$$

Тогда модульная экспонента с основанием A по модулю N представляет собой функцию

$$f_{A,N} : Z_N \to Z_N,$$

$$f_{A,N}(x) = A^x \pmod N, \tag{4.4}$$

где X – целое число, $1 \le x \le N-1$.

Существуют эффективные алгоритмы, позволяющие достаточно быстро вычислить значения функции $f_{A,N}(x)$.

Если $y = A^x$, то естественно записать $x = \log_A(y)$.

Поэтому задачу обращения функции $f_{A,N}(x)$ называют задачей нахождения дискретного логарифма или задачей дискретного логарифмирования.

Задача дискретного логарифмирования формулируется следующим образом. Для известных целых A, N, y найти целое число x, такое, что

$$A^x \bmod N = y.$$

Алгоритм вычисления дискретного логарифма за приемлемое время пока не найден. Поэтому модульная экспонента считается однонаправленной функцией.

По современным оценкам теории чисел при целых числах $A \approx 2^{664}$ и $N \approx 2^{664}$ решение задачи дискретного логарифмирования (нахождение показателя степени x для известного y) потребует около 10^{26} операций, т.е. эта задача имеет в 10^3 раз большую вычислительную сложность, чем задача разложения на множители. При увеличении длины чисел разница в оценках сложности задач возрастает.

Следует отметить, что пока не удалось доказать, что не существует эффективного алгоритма вычисления дискретного логарифма за приемлемое время. Исходя из этого, модульная экспонента отнесена к однонаправленным функциям условно, что, однако, не мешает с успехом применять ее на практике.

Вторым важным классом функций, используемых при построении криптосистем с открытым ключом, являются так называемые однонаправленные функции с "потайным ходом" (с лазейкой). Дадим неформальное определение такой функции. Функция

$$f : X \to Y$$

относится к классу однонаправленных функций с "потайным ходом" в том случае, если она является однонаправленной и, кроме того, возможно эффективное вычисление обратной функции, если известен "потайной ход" (секретное число, строка или другая информация, ассоциирующаяся с данной функцией).

В качестве примера однонаправленной функции с "потайным ходом" можно указать используемую в криптосистеме RSA модульную экспоненту с фиксированными модулем и показателем степени. Переменное основание модульной экспоненты используется для указания числового значения сообщения M либо криптограммы C (см. § 4.3.).

4.3. Криптосистема шифрования данных RSA

Алгоритм RSA предложили в 1978 г. три автора: Р. Райвест (Rivest), А. Шамир (Shamir) и А. Адлеман (Adleman). Алгоритм получил свое название по первым буквам фамилий его авторов. Алгоритм RSA стал первым полноценным алгоритмом с открытым

ключом, который может работать как в режиме шифрования данных, так и в режиме электронной цифровой подписи [118].

Надежность алгоритма основывается на трудности факторизации больших чисел и трудности вычисления дискретных логарифмов.

В криптосистеме RSA открытый ключ $K_в$, секретный ключ $k_в$, сообщение M и криптограмма C принадлежат множеству целых чисел

$$Z_N = \{0, 1, 2, ..., N-1\}, \quad (4.5)$$

где N – модуль:

$$N = P * Q. \quad (4.6)$$

Здесь P и Q – случайные большие простые числа. Для обеспечения максимальной безопасности выбирают P и Q равной длины и хранят в секрете.

Множество Z_N с операциями сложения и умножения по модулю N образует арифметику по модулю N.

Открытый ключ $K_в$ выбирают случайным образом так, чтобы выполнялись условия:

$$1 < K_в \le \varphi(N), \quad НОД\ (K_в, \varphi(N)) = 1, \quad (4.7)$$

$$\varphi(N) = (P-1)(Q-1), \quad (4.8)$$

где $\varphi(N)$ – функция Эйлера.

Функция Эйлера $\varphi(N)$ указывает количество положительных целых чисел в интервале от 1 до N, которые взаимно просты с N.

Второе из указанных выше условий означает, что открытый ключ $K_в$ и функция Эйлера $\varphi(N)$ должны быть взаимно простыми.

Далее, используя расширенный алгоритм Евклида, вычисляют секретный ключ $k_в$, такой, что

$$k_в * K_в \equiv 1 (\mathrm{mod}\ \varphi(N)) \quad (4.9)$$

или

$$k_в = K_в^{-1} (\mathrm{mod}\ (P-1)(Q-1)).$$

Это можно осуществить, так как получатель B знает пару простых чисел (P, Q) и может легко найти $\varphi(N)$. Заметим, что $k_в$ и N должны быть взаимно простыми.

Открытый ключ $K_в$ используют для шифрования данных, а секретный ключ $k_в$ – для расшифрования.

Преобразование шифрования определяет криптограмму C через пару (открытый ключ $K_в$, сообщение M) в соответствии со следующей формулой:

$$C = E_{K_в}(M) = E_B(M) = M^{K_в} (\mathrm{mod}\ N). \quad (4.10)$$

В качестве алгоритма быстрого вычисления значения C используют ряд последовательных возведений в квадрат целого M и умножений на M с приведением по модулю N.

Обращение функции $C = M^{K_в} (\text{mod } N)$, т.е. определение значения M по известным значениям C, $K_в$ и N, практически не осуществимо при $N \approx 2^{512}$.

Однако обратную задачу, т.е. задачу расшифрования криптограммы C, можно решить, используя пару (секретный ключ $k_в$, криптограмма C) по следующей формуле:

$$M = D_{k_в}(C) = D_B(C) = C^{k_в} (\text{mod } N). \quad (4.11)$$

Процесс расшифрования можно записать так:

$$D_B(E_B(M)) = M. \quad (4.12)$$

Подставляя в (4.12) значения (4.10) и (4.11), получаем:

$$(M^{K_в})^{k_в} = M (\text{mod } N)$$

или

$$M^{K_в k_в} = M (\text{mod } N). \quad (4.13)$$

Величина $\varphi(N)$ играет важную роль в теореме Эйлера, которая утверждает, что если НОД $(x, N) = 1$, то

$$x^{\varphi(N)} \equiv 1 (\text{mod } N),$$

или в несколько более общей форме

$$x^{n \cdot \varphi(N)+1} \equiv x (\text{mod } N). \quad (4.14)$$

Сопоставляя выражения (4.13) и (4.14), получаем

$$K_в * k_в = n * \varphi(N) + 1$$

или, что то же самое,

$$K_в * k_в \equiv 1 (\text{mod } \varphi(N)).$$

Именно поэтому для вычисления секретного ключа $k_в$ используют соотношение (4.9).

Таким образом, если криптограмму

$$C = M^{K_в} (\text{mod } N)$$

возвести в степень $k_в$, то в результате восстанавливается исходный открытый текст M, так как

$$(M^{K_в})^{k_в} = M^{K_в k_в} = M^{n \cdot \varphi(N)+1} \equiv M (\text{mod } N).$$

Таким образом, получатель B, который создает криптосистему, защищает два параметра: 1) секретный ключ $k_в$ и 2) пару чисел (P,Q), произведение которых дает значение модуля N. С другой стороны, получатель B открывает значение модуля N и открытый ключ $K_в$.

Противнику известны лишь значения $K_в$ и N. Если бы он смог разложить число N на множители P и Q, то он узнал бы "потайной ход"–тройку чисел $\{P, Q, K_в\}$, вычислил значение функции Эйлера

$$\varphi(N) = (P - 1)(Q - 1)$$

и определил значение секретного ключа $k_в$.

Однако, как уже отмечалось, разложение очень большого N на множители вычислительно не осуществимо (при условии, что длины выбранных P и Q составляют не менее 100 десятичных знаков).

Процедуры шифрования и расшифрования в криптосистеме RSA

Предположим, что пользователь A хочет передать пользователю B сообщение в зашифрованном виде, используя криптосистему RSA. В таком случае пользователь A выступает в роли отправителя сообщения, а пользователь B – в роли получателя. Как отмечалось выше, криптосистему RSA должен сформировать получатель сообщения, т.е. пользователь B. Рассмотрим последовательность действий пользователя B и пользователя A.

1. Пользователь B выбирает два произвольных больших простых числа P и Q.

2. Пользователь B вычисляет значение модуля $N = P * Q$.

3. Пользователь B вычисляет функцию Эйлера

$$\varphi(N) = (P-1)(Q-1)$$

и выбирает случайным образом значение открытого ключа $K_в$ с учетом выполнения условий:

$$1 < K_в \le \varphi(N), \quad НОД(K_в, \varphi(N)) = 1.$$

4. Пользователь B вычисляет значение секретного ключа $k_в$, используя расширенный алгоритм Евклида при решении сравнения

$$k_в \equiv K_в^{-1} \pmod{\varphi(N)}.$$

5. Пользователь B пересылает пользователю A пару чисел $(N, K_в)$ по незащищенному каналу.

Если пользователь A хочет передать пользователю B сообщение M, он выполняет следующие шаги.

6. Пользователь A разбивает исходный открытый текст M на блоки, каждый из которых может быть представлен в виде числа

$$M_i = 0, 1, 2, \ldots, N-1.$$

7. Пользователь A шифрует текст, представленный в виде последовательности чисел M_i по формуле

$$C_i = M_i^{K_в} \pmod{N}$$

и отправляет криптограмму

$$C_1, C_2, C_3, \ldots, C_i, \ldots$$

пользователю B.

8. Пользователь B расшифровывает принятую криптограмму

$$C_1, C_2, C_3, \ldots, C_i, \ldots,$$

используя секретный ключ $k_в$, по формуле

$$M_i = C_i^{k_в} \pmod N.$$

В результате будет получена последовательность чисел M_i, которые представляют собой исходное сообщение M. Чтобы алгоритм RSA имел практическую ценность, необходимо иметь возможность без существенных затрат генерировать большие простые числа, уметь оперативно вычислять значения ключей $K_в$ и $k_в$.

Пример. *Шифрование сообщения CAB.* Для простоты вычислений будут использоваться небольшие числа. На практике применяются очень большие числа (см. следующий раздел).

Действия пользователя B.

1. Выбирает P = 3 и Q = 11.
2. Вычисляет модуль $N = P * Q = 3 * 11 = 33$.
3. Вычисляет значение функции Эйлера для N = 33:

$$\varphi(N) = \varphi(33) = (P-1)(Q-1) = 2 * 10 = 20.$$

Выбирает в качестве открытого ключа $K_в$ произвольное число с учетом выполнения условий:

$$1 < K_в \le 20, \quad НОД\ (K_в, 20) = 1.$$

Пусть $K_в = 7$.

4. Вычисляет значение секретного ключа $k_в$, используя расширенный алгоритм Евклида (см. приложение) при решении сравнения

$$k_в \equiv 7^{-1} \pmod{20}.$$

Решение дает $k_в = 3$.

5. Пересылает пользователю A пару чисел (N = 33, $K_в = 7$).

Действия пользователя A.

6. Представляет шифруемое сообщение как последовательность целых чисел в диапазоне 0...32. Пусть буква A представляется как число 1, буква B – как число 2, буква C – как число 3. Тогда сообщение CAB можно представить как последовательность чисел 312, т.е. $M_1 = 3$, $M_2 = 1$, $M_3 = 2$.

7. Шифрует текст, представленный в виде последовательности чисел M_1, M_2 и M_3, используя ключ $K_в = 7$ и N = 33, по формуле

$$C_i = M_i^{K_в} \pmod N = M_i^7 \pmod{33}.$$

Получает

$$C_1 = 3^7 \pmod{33} = 2187 \pmod{33} = 9,$$

$$C_2 = 1^7 \pmod{33} = 1 \pmod{33} = 1,$$

$$C_3 = 2^7 \pmod{33} = 128 \pmod{33} = 29.$$

Отправляет пользователю B криптограмму

$$C_1, C_2, C_3 = 9, 1, 29.$$

Действия пользователя B.

8. Расшифровывает принятую криптограмму C_1, C_2, C_3, используя секретный ключ $k_в = 3$, по формуле

$$M_i = C_i^{k_в} \pmod N = C_i^3 \pmod{33}.$$

Получает

$$M_1 = 9^3 \pmod{33} = 729 \pmod{33} = 3,$$

$$M_2 = 1^3 \pmod{33} = 1 \pmod{33} = 1,$$

$$M_3 = 29^3 \pmod{33} = 24389 \pmod{33} = 2.$$

Таким образом, восстановлено исходное сообщение: C A B
3 1 2

Безопасность и быстродействие криптосистемы RSA

Безопасность алгоритма RSA базируется на трудности решения задачи факторизации больших чисел, являющихся произведениями двух больших простых чисел. Действительно, криптостойкость алгоритма RSA определяется тем, что после формирования секретного ключа $k_в$ и открытого ключа $K_в$ "стираются" значения простых чисел P и Q, и тогда исключительно трудно определить секретный ключ $k_в$ по открытому ключу $K_в$, поскольку для этого необходимо решить задачу нахождения делителей P и Q модуля N.

Разложение величины N на простые множители P и Q позволяет вычислить функцию $\varphi(N) = (P-1)(Q-1)$ и затем определить секретное значение $k_в$, используя уравнение

$$K_в * k_в \equiv 1 (\mathrm{mod}\ \varphi(N)).$$

Другим возможным способом криптоанализа алгоритма RSA является непосредственное вычисление или подбор значения функции $\varphi(N)=(P-1)(Q-1)$. Если установлено значение $\varphi(N)$, то сомножители P и Q вычисляются достаточно просто. В самом деле, пусть

$$x = P + Q = N + 1 - \varphi(N),$$

$$y = (P - Q)^2 = (P + Q)^2 - 4 * N.$$

Зная $\varphi(N)$, можно определить x и затем y; зная x и y, можно определить числа P и Q из следующих соотношений:

$$P = 1/2\ (x + \sqrt{y}\ ), \quad Q = 1/2\ (x - \sqrt{y}\ ).$$

Однако эта атака не проще задачи факторизации модуля N [28].

Задача факторизации является трудно разрешимой задачей для больших значений модуля N.

Сначала авторы алгоритма RSA предлагали для вычисления модуля N выбирать простые числа P и Q случайным образом, по 50 десятичных разрядов каждое. Считалось, что такие большие числа N очень трудно разложить на простые множители. Один из авторов алгоритма RSA, P. Райвест, полагал, что разложение на простые множители числа из почти 130 десятичных цифр, приведенного в их публикации, потребует более 40 квадриллионов лет машинного времени. Однако этот прогноз не оправдался из-за сравнительно быстрого прогресса компьютеров и их вычислительной мощности, а также улучшения алгоритмов факторизации.

Ряд алгоритмов факторизации приведен в [45]. Один из наиболее быстрых алгоритмов, известных в настоящее время, алгоритм NFS (Number Field Sieve) может выполнить факторизацию большого числа N (с числом десятичных разрядов больше 120) за число шагов, оцениваемых величиной

$$e^{2(\ln n)^{1/3}(\ln(\ln n))^{2/3}}.$$

В 1994 г. было факторизовано число со 129 десятичными цифрами. Это удалось осуществить математикам А. Ленстра и М. Манасси посредством организации распределенных вычислений на 1600 компьютерах, объединенных сетью, в течение восьми месяцев. По мнению А. Ленстра и М. Манасси, их работа компрометирует криптосистемы RSA и создает большую угрозу их дальнейшим применениям. Теперь разработчикам криптоалгоритмов с открытым ключом на базе RSA приходится избегать применения чисел длиной менее 200 десятичных разрядов. Самые последние публикации предлагают применять для этого числа длиной не менее 250–300 десятичных разрядов.

В [121] сделана попытка расчета оценок безопасных длин ключей асимметричных криптосистем на ближайшие 20 лет исходя из прогноза развития компьютеров и их вычислительной мощности, а также возможного совершенствования алгоритмов факторизации. Эти оценки (табл.4.1) даны для трех групп пользователей (индивидуальных пользователей, корпораций и государственных организаций), в соответствии с различием требований к их информационной безопасности. Конечно, данные оценки следует рассматривать как сугубо приблизительные, как возможную тенденцию изменений безопасных длин ключей асимметричных криптосистем со временем.

Таблица 4.1

Оценки длин ключей для асимметричных криптосистем, бит

Год	Отдельные пользователи	Корпорации	Государственные организации
1995	768	1280	1536
2000	1024	1280	1536
2005	1280	1536	2048
2010	1280	1536	2048
2015	1536	2048	2048

Криптосистемы RSA реализуются как аппаратным, так и программным путем.

Для аппаратной реализации операций зашифрования и расшифрования RSA разработаны специальные процессоры. Эти процессоры, реализованные на сверхбольших интегральных схемах (СБИС), позволяют выполнять операции RSA, связанные с возведением больших чисел в колоссально большую степень по модулю N, за относительно короткое время. И все же аппаратная реализация RSA примерно в 1000 раз медленнее аппаратной реализации симметричного криптоалгоритма DES.

Одна из самых быстрых аппаратных реализаций RSA с модулем 512 бит на сверхбольшой интегральной схеме имеет быстродействие 64 Кбит/с. Лучшими из серийно выпускаемых СБИС являются процессоры фирмы CYLINK, выполняющие 1024-битовое шифрование RSA.

Программная реализация RSA примерно в 100 раз медленнее программной реализации DES. С развитием технологии эти оценки могут несколько изменяться, но асимметричная криптосистема RSA никогда не достигнет быстродействия симметричных криптосистем.

Следует отметить, что малое быстродействие криптосистем RSA ограничивает область их применения, но не перечеркивает их ценность.

4.4. Схема шифрования Полига–Хеллмана

Схема шифрования Полига–Хеллмана [121] сходна со схемой шифрования RSA. Она представляет собой несимметричный алгоритм, поскольку используются различные ключи для шифрования и расшифрования. В то же время эту схему нельзя отнести к классу криптосистем с открытым ключом, так как ключи шифрования и расшифрования легко выводятся один из другого. Оба ключа (шифрования и расшифрования) нужно держать в секрете.

Аналогично схеме RSA криптограмма C и открытый текст P определяются из соотношений:

$$C = P^e \bmod n,$$

$$P = C^d \bmod n,$$

где $e*d \equiv 1$ (по модулю некоторого составного числа).

В отличие от алгоритма RSA в этой схеме число n не определяется через два больших простых числа; число n должно оставаться частью секретного ключа. Если кто-либо узнает значения e и n, он сможет вычислить значение d.

Не зная значений e или d, противник будет вынужден вычислять значение

$$e = \log_P C \pmod n.$$

Известно, что это является трудной задачей.

Схема шифрования Полига–Хеллмана запатентована в США и Канаде.

4.5. Схема шифрования Эль Гамаля

Схема Эль Гамаля, предложенная в 1985 г., может быть использована как для шифрования, так и для цифровых подписей. Безопасность схемы Эль Гамаля обусловлена сложностью вычисления дискретных логарифмов в конечном поле.

Для того чтобы генерировать пару ключей (открытый ключ – секретный ключ), сначала выбирают некоторое большое простое число P и большое целое число G, причем G<P. Числа P и G могут быть распространены среди группы пользователей.

Затем выбирают случайное целое число X, причем X<P. Число X является секретным ключом и должно храниться в секрете.

Далее вычисляют $Y = G^X \bmod P$. Число Y является открытым ключом.

Для того чтобы зашифровать сообщение M, выбирают случайное целое число K, 1<K<P–1, такое, что числа K и (P–1) являются взаимно простыми.

Затем вычисляют числа

$$a = G^K \bmod P,$$

$$b = Y^K M \bmod P.$$

Пара чисел (a, b) является шифртекстом. Заметим, что длина шифртекста вдвое больше длины исходного открытого текста M.

Для того чтобы расшифровать шифртекст (a, b), вычисляют

$$M = b/a^X \bmod P. \qquad (*)$$

Поскольку

$$a^X \equiv G^{KX} \bmod P,$$

$$b/a^X \equiv Y^K M/a^X \equiv G^{KX} M/G^{KX} \equiv M \pmod P,$$

то соотношение (*) справедливо.

Пример. Выберем P = 11, G = 2, секретный ключ X = 8.

Вычисляем

$$Y = G^X \bmod P = 2^8 \bmod 11 = 256 \bmod 11 = 3.$$

Итак, открытый ключ Y = 3.

Пусть сообщение M = 5. Выберем некоторое случайное число K = 9. Убедимся, что НОД (K, P–1) = 1. Действительно, НОД (9, 10) = 1. Вычисляем пару чисел a и b:

$$a = G^K \bmod P = 2^9 \bmod 11 = 512 \bmod 11 = 6,$$

$$b = Y^K M \bmod P = 3^9 * 5 \bmod 11 = 19683 * 5 \bmod 11 = 9.$$

Получим шифртекст (a, b) = (6, 9).

Выполним расшифрование этого шифртекста. Вычисляем сообщение M, используя секретный ключ X:

$$M = b/a^X \bmod P = 9/6^8 \bmod 11.$$

Выражение $M = 9/6^8 \bmod 11$ можно представить в виде

$$6^8 * M \equiv 9 \bmod 11$$

или

$$1679616 * M \equiv 9 \bmod 11.$$

Решая данное сравнение, находим M = 5.

В реальных схемах шифрования необходимо использовать в качестве модуля P большое целое простое число, имеющее в двоичном представлении длину 512...1024 бит.

При программной реализации схемы Эль Гамаля [123] скорость ее работы (на SPARC-II) в режимах шифрования и расшифрования при 160-битовом показателе степени для различных длин модуля P определяется значениями, приведенными в табл. 4.2.

Таблица 4.2

Скорости работы схемы Эль Гамаля

Режим работы	Длина модуля, бит		
	512	768	1024
Шифрование	0,33 с	0,80 с	1,09 с
Расшифрование	0,24 с	0,58 с	0,77 с

4.6. Комбинированный метод шифрования

Главным достоинством криптосистем с открытым ключом является их потенциально высокая безопасность: нет необходимости ни передавать, ни сообщать кому бы то ни было значения секретных ключей, ни убеждаться в их подлинности. В симметричных криптосистемах существует опасность раскрытия секретного ключа во время передачи.

Однако алгоритмы, лежащие в основе криптосистем с открытым ключом, имеют следующие недостатки:

- генерация новых секретных и открытых ключей основана на генерации новых больших простых чисел, а проверка простоты чисел занимает много процессорного времени;
- процедуры шифрования и расшифрования, связанные с возведением в степень многозначного числа, достаточно громоздки.

Поэтому быстродействие криптосистем с открытым ключом обычно в сотни и более раз меньше быстродействия симметричных криптосистем с секретным ключом.

Комбинированный (гибридный) метод шифрования позволяет сочетать преимущества высокой секретности, предоставляемые асимметричными криптосистемами с открытым ключом, с преимуществами высокой скорости работы, присущими симметричным криптосистемам с секретным ключом. При таком подходе криптосистема с открытым ключом применяется для шифрования, передачи и последующего расшифрования только секретного ключа симметричной криптосистемы. А симметричная криптосистема применяется для шифрования и передачи исходного открытого текста. В результате криптосистема с открытым ключом не заменяет симметричную криптосистему с секретным ключом, а лишь дополняет ее, позволяя повысить в целом защищенность передаваемой информации. Такой подход иногда называют схемой электронного цифрового конверта.

Если пользователь А хочет передать зашифрованное комбинированным методом сообщение М пользователю В, то порядок его действий будет таков.

1. Создать (например, сгенерировать случайным образом) симметричный ключ, называемый в этом методе сеансовым ключом K_S.

2. Зашифровать сообщение М на сеансовом ключе K_S.

3. Зашифровать сеансовый ключ K_S на открытом ключе K_B пользователя В.

4. Передать по открытому каналу связи в адрес пользователя В зашифрованное сообщение вместе с зашифрованным сеансовым ключом.

Действия пользователя В при получении зашифрованного сообщения и зашифрованного сеансового ключа должны быть обратными:

5. Расшифровать на своем секретном ключе k_B сеансовый ключ K_S.

6. С помощью полученного сеансового ключа K_S расшифровать и прочитать сообщение М.

При использовании комбинированного метода шифрования можно быть уверенным в том, что только пользователь В сможет правильно расшифровать ключ K_S и прочитать сообщение М.

Таким образом, при комбинированном методе шифрования применяются криптографические ключи как симметричных, так и асимметричных криптосистем. Очевидно, выбор длин ключей для каждого типа криптосистемы следует осуществлять таким образом, чтобы злоумышленнику было одинаково трудно атаковать любой механизм защиты комбинированной криптосистемы.

В табл.4.3. приведены распространенные длины ключей симметричных и асимметричных криптосистем, для которых трудность атаки полного перебора примерно равна трудности факторизации соответствующих модулей асимметричных криптосистем [121].

Таблица 4.3

Длины ключей для симметричных и асимметричных криптосистем при одинаковой их криптостойкости

Длина ключа симметричной криптосистемы, бит	Длина ключа асимметричной криптосистемы, бит
56	384
64	512
80	768
112	1792
128	2304

Комбинированный метод допускает возможность выполнения процедуры аутентификации, т.е. проверки подлинности передаваемого сообщения. Для этого пользователь А на основе функции хэширования сообщения и своего секретного ключа k_A с помощью известного алгоритма электронной цифровой подписи (ЭЦП) генерирует свою подпись и записывает ее, например, в конец передаваемого файла.

Пользователь В, прочитав принятое сообщение, может убедиться в подлинности цифровой подписи абонента А. Используя тот же алгоритм ЭЦП и результат хэширования принятого сообщения, пользователь В проверяет полученную подпись (см. гл.6). Комбинированный метод шифрования является наиболее рациональным, объединяя в себе высокое быстродействие симметричного шифрования и высокую криптостойкость, гарантируемую системами с открытым ключом.

ГЛАВА 5. ИДЕНТИФИКАЦИЯ И ПРОВЕРКА ПОДЛИННОСТИ

5.1. Основные понятия и концепции

С каждым объектом компьютерной системы (КС) связана некоторая информация, однозначно идентифицирующая его. Это может быть *число, строка символов, алгоритм,* определяющий данный объект. Эту информацию называют *идентификатором объекта.* Если объект имеет некоторый идентификатор, зарегистрированный в сети, он называется законным (легальным) объектом; остальные объекты относятся к незаконным (нелегальным).

Идентификация объекта – одна из функций подсистемы защиты. Эта функция выполняется в первую очередь, когда объект делает попытку войти в сеть. Если процедура идентификации завершается успешно, данный объект считается законным для данной сети.

Следующий шаг – аутентификация объекта (проверка подлинности объекта). Эта процедура устанавливает, является ли данный объект именно таким, каким он себя объявляет.

После того как объект идентифицирован и подтверждена его подлинность, можно установить сферу его действия и доступные ему ресурсы КС. Такую процедуру называют *предоставлением полномочий (авторизацией).*

Перечисленные три процедуры инициализации являются процедурами защиты и относятся к одному объекту КС [55].

При защите каналов передачи данных *подтверждение подлинности* (аутентификация) *объектов* означает взаимное установление подлинности объектов, связывающихся между собой по линиям связи. Процедура подтверждения подлинности выполняется обычно в начале сеанса в процессе установления соединения абонентов. (Термин "соединение" указывает на логическую связь (потенциально двустороннюю) между двумя объектами сети. Цель данной процедуры – обеспечить уверенность, что соединение установлено с законным объектом и вся информация дойдет до места назначения.

После того как соединение установлено, необходимо обеспечить выполнение требований защиты при обмене сообщениями:

(а) получатель должен быть уверен в подлинности источника данных;

(б) получатель должен быть уверен в подлинности передаваемых данных;

(в) отправитель должен быть уверен в доставке данных получателю;

(г) отправитель должен быть уверен в подлинности доставленных данных.

Для выполнения требований (а) и (б) средством защиты является *цифровая подпись*. Для выполнения требований (в) и (г) отправитель должен получить *уведомление о вручении* с помощью удостоверяющей почты (certified mail). Средством защиты в такой процедуре является цифровая подпись подтверждающего ответного сообщения, которое в свою очередь является доказательством пересылки исходного сообщения.

Если эти четыре требования реализованы в КС, то гарантируется защита данных при их передаче по каналу связи и обеспечивается функция защиты, называемая функцией подтверждения (неоспоримости) передачи. В этом случае отправитель не может отрицать ни факта посылки сообщения, ни его содержания, а получатель не может отрицать ни факта получения сообщения, ни подлинности его содержания.

5.2. Идентификация и аутентификация пользователя

Прежде чем получить доступ к ресурсам компьютерной системы, пользователь должен пройти процесс представления компьютерной системе, который включает две стадии:

- идентификацию – пользователь сообщает системе по ее запросу свое имя (идентификатор);
- аутентификацию – пользователь подтверждает идентификацию, вводя в систему уникальную, не известную другим пользователям информацию о себе (например, пароль).

Для проведения процедур идентификации и аутентификации пользователя необходимы:

- наличие соответствующего *субъекта (модуля) аутентификации*;
- наличие *аутентифицирующего объекта*, хранящего уникальную информацию для аутентификации пользователя.

Различают две формы представления объектов, аутентифицирующих пользователя:

- внешний аутентифицирующий объект, не принадлежащий системе;

• внутренний объект, принадлежащий системе, в который переносится информация из внешнего объекта.

Внешние объекты могут быть технически реализованы на различных носителях информации – магнитных дисках, пластиковых картах и т.п. Естественно, что внешняя и внутренняя формы представления аутентифицирующего объекта должны быть семантически тождественны.

Типовые схемы идентификации и аутентификации пользователя

Рассмотрим структуры данных и протоколы идентификации и аутентификации пользователя [73]. Допустим, что в компьютерной системе зарегистрировано n пользователей. Пусть i-й аутентифицирующий объект i-го пользователя содержит два информационных поля:

ID_i – неизменный идентификатор i-го пользователя, который является аналогом имени и используется для идентификации пользователя;

K_i – аутентифицирующая информация пользователя, которая может изменяться и служит для аутентификации (например, пароль $P_i = K_i$).

Описанная структура соответствует практически любому ключевому носителю информации, используемому для опознания пользователя. Например, для носителей типа пластиковых карт выделяется неизменяемая информация ID_i первичной персонализации пользователя и объект в файловой структуре карты, содержащий K_i.

Совокупную информацию в ключевом носителе можно назвать первичной аутентифицирующей информацией i-го пользователя. Очевидно, что внутренний аутентифицирующий объект не должен существовать в системе длительное время (больше времени работы конкретного пользователя). Для длительного хранения следует использовать данные в защищенной форме.

Рассмотрим две типовые схемы идентификации и аутентификации [73].

Схема 1. В компьютерной системе выделяется объект-эталон для идентификации и аутентификации пользователей. Структура объекта-эталона для схемы 1 показана в табл.5.1. Здесь $E_i = F(ID_i, K_i)$, где F – функция, которая обладает свойством "невосстановимости" значения K_i по E_i и ID_i."Невосстановимость" K_i оценивается некоторой пороговой трудоемкостью T_o решения задачи восстановления аутентифицирующей информации K_i по E_i и ID_i. Кроме того, для пары K_i и K_j возможно совпадение соответствующих значений E. В связи с этим вероятность ложной аутентификации пользователя не должна быть больше некоторого порогового значения P_o. На практике задают $T_o = 10^{20}...10^{30}$, $P_o = 10^{-7}...10^{-9}$ [73].

Таблица 5.1

Структура объекта-эталона для схемы 1

Номер пользователя	Информация для идентификации	Информация для аутентификации
1	ID_1	E_1
2	ID_2	E_2
...	...	...
N	ID_n	E_n

Протокол идентификации и аутентификации (для схемы 1).

1. Пользователь предъявляет свой идентификатор ID.

2. Если ID не совпадает ни с одним ID_i, зарегистрированным в компьютерной системе, то идентификация отвергается – пользователь не допускается к работе, иначе (существует $ID_i = ID$) устанавливается, что пользователь, назвавшийся пользователем i, прошел идентификацию.

3. Субъект аутентификации запрашивает у пользователя его аутентификатор K.

4. Субъект аутентификации вычисляет значение $Y = F(ID_i, K)$.

5. Субъект аутентификации производит сравнение значений Y и E_i. При совпадении этих значений устанавливается, что данный пользователь успешно аутентифицирован в системе. Информация об этом пользователе передается в программные модули, использующие ключи пользователей (т.е. в систему шифрования, разграничения доступа и т.д.). В противном случае аутентификация отвергается – пользователь не допускается к работе.

Данная схема идентификации и аутентификации пользователя может быть модифицирована. Модифицированная схема 2 обладает лучшими характеристиками по сравнению со схемой 1.

Схема 2. В компьютерной системе выделяется модифицированный объект-эталон, структура которого показана в табл. 5.2.

Таблица 5.2

Структура модифицированного объекта-эталона

Номер пользователя	Информация для идентификации	Информация для аутентификации
1	ID_1, S_1	E_1
2	ID_2, S_2	E_2
...	...	...
N	ID_n, S_n	E_n

В отличие от схемы 1, в схеме 2 значение E_i равно $F(S_i, K_i)$, где S_i – случайный вектор, задаваемый при создании идентификатора пользователя, т.е. при создании строки, необходимой для идентификации и аутентификации пользователя; F – функция, которая обладает свойством "невосстановимости" значения K_i по E_i и S_i.

Протокол идентификации и аутентификации (для схемы 2).

1. Пользователь предъявляет свой идентификатор ID.

2. Если ID не совпадает ни с одним ID_i, зарегистрированным в компьютерной системе, то идентификация отвергается – пользователь не допускается к работе, иначе (существует $ID_i = ID$) устанавливается, что пользователь, называвшийся пользователем i, прошел идентификацию.

3. По идентификатору ID_i выделяется вектор S_i.

4. Субъект аутентификации запрашивает у пользователя аутентификатор K.

5. Субъект аутентификации вычисляет значение $Y = F(S_i, K)$.

6. Субъект аутентификации производит сравнение значений Y и E_i. При совпадении этих значений устанавливается, что данный пользователь успешно аутентифицирован в системе. В противном случае аутентификация отвергается – пользователь не допускается к работе.

Вторая схема аутентификации применяется в ОС UNIX. В качестве идентификатора ID используется имя пользователя (запрошенное по Loqin), в качестве аутентификатора K_i – пароль пользователя (запрошенный по Password), функция F представляет собой алгоритм шифрования DES. Эталоны для идентификации и аутентификации содержатся в файле Etc/passwd.

Следует отметить, что необходимым требованием устойчивости схем аутентификации к восстановлению информации K_i является случайный равновероятный выбор K_i из множества возможных значений.

Системы парольной аутентификации имеют пониженную стойкость, поскольку в них выбор аутентифицирующей информации происходит из относительно небольшого множества осмысленных слов. Мощность этого множества определяется энтропией соответствующего языка.

Особенности применения пароля для аутентификации пользователя

Традиционно каждый законный пользователь компьютерной системы получает идентификатор и/или пароль. В начале сеанса работы пользователь предъявляет свой идентификатор системе, которая затем запрашивает у пользователя пароль.

Простейший метод подтверждения подлинности с использованием пароля основан на сравнении представляемого пользователем пароля P_A с исходным значением P_A', хранящимся в компьютерном центре (рис. 5.1). Поскольку пароль должен храниться в тайне, он должен шифроваться перед пересылкой по незащищенному каналу. Если значения P_A и P_A' совпадают, то пароль P_A считается подлинным, а пользователь – законным [123].

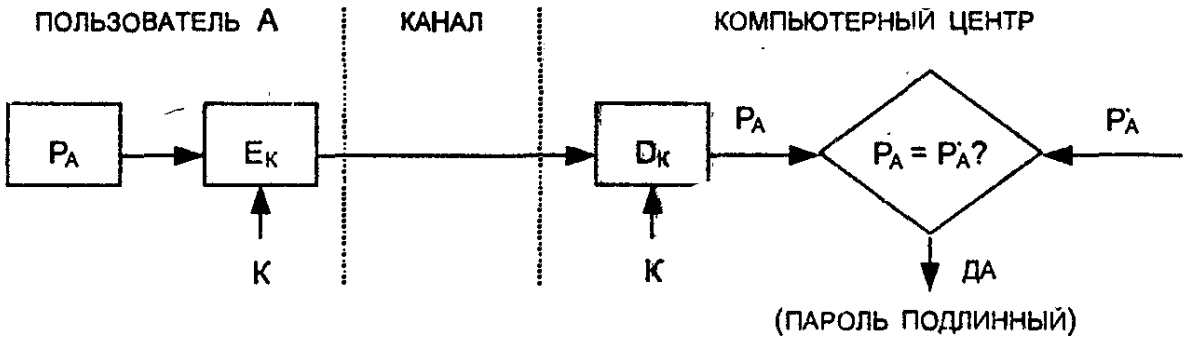


Рис. 5.1. Схема простой аутентификации с помощью пароля

Если кто-нибудь, не имеющий полномочий для входа в систему, узнает каким-либо образом пароль и идентификационный номер законного пользователя, он получает доступ в систему.

Иногда получатель не должен раскрывать исходную открытую форму пароля. В этом случае отправитель должен пересылать вместо открытой формы пароля отображение пароля, получаемое с использованием односторонней функции $\alpha(\cdot)$ пароля. Это преобразование должно гарантировать невозможность раскрытия противником пароля по его отображению, так как противник наталкивается на неразрешимую числовую задачу.

Например, функция $\alpha(\cdot)$ может быть определена следующим образом:

$$\alpha(P) = E_P(ID),$$

где P – пароль отправителя; ID – идентификатор отправителя; E_P – процедура шифрования, выполняемая с использованием пароля P в качестве ключа.

Такие функции особенно удобны, если длина пароля и ключа одинаковы. В этом случае подтверждение подлинности с помощью пароля состоит из пересылки получателю отображения $\alpha(P)$ и сравнения его с предварительно вычисленным и хранимым эквивалентом $\alpha'(P)$.

На практике пароли состоят только из нескольких букв, чтобы дать возможность пользователям запомнить их. Короткие пароли уязвимы к атаке полного перебора всех вариантов. Для того чтобы предотвратить такую атаку, функцию $\alpha(P)$ определяют иначе, а именно:

$$\alpha(P) = E_{P \oplus K}(ID),$$

где K и ID – соответственно ключ и идентификатор отправителя.

Очевидно, значение $\alpha(P)$ вычисляется заранее и хранится в виде $\alpha'(P)$ в идентификационной таблице у получателя (рис. 5.2). Подтверждение подлинности состоит из сравнения двух отображений пароля $\alpha(P_A)$ и $\alpha'(P_A)$ и признания пароля P_A, если эти отобра-

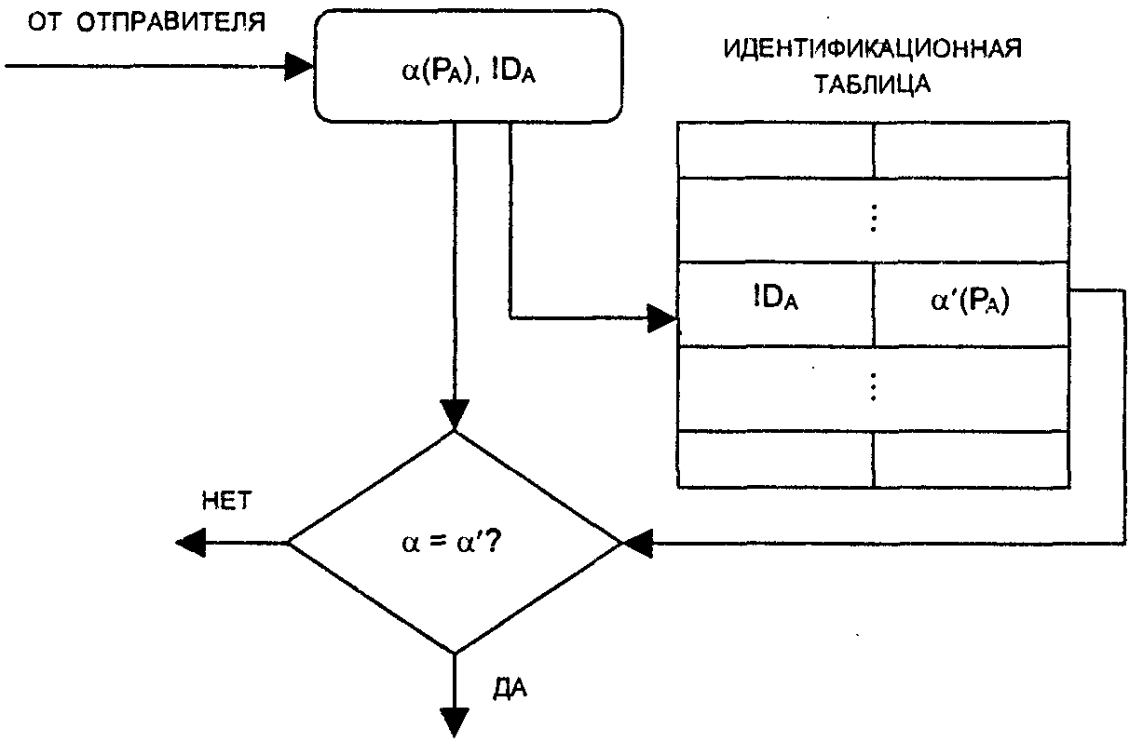


Рис. 5.2. Схема аутентификации с помощью пароля с использованием идентификационной таблицы

жения равны. Конечно, любой, кто получит доступ к идентификационной таблице, может незаконно изменить ее содержимое, не опасаясь, что эти действия будут обнаружены.

Применение для целей идентификации и аутентификации персонального идентификационного номера PIN рассматривается в гл. 9.

Биометрическая идентификация и аутентификация пользователя

Процедуры идентификации и аутентификации пользователя могут базироваться не только на секретной информации, которой обладает пользователь (пароль, секретный ключ, персональный идентификатор и т. п.). В последнее время все большее распространение получает биометрическая идентификация и аутентификация пользователя, позволяющая уверенно идентифицировать потенциального пользователя путем измерения физиологических параметров и характеристик человека, особенностей его поведения.

Отметим основные достоинства биометрических методов идентификации и аутентификации пользователя по сравнению с традиционными [73]:

- высокая степень достоверности идентификации по биометрическим признакам из-за их уникальности;
- неотделимость биометрических признаков от дееспособной личности;
- трудность фальсификации биометрических признаков.

В качестве биометрических признаков, которые могут быть использованы при идентификации потенциального пользователя, можно выделить следующие:

- узор радужной оболочки и сетчатки глаз;
- отпечатки пальцев;
- геометрическая форма руки;
- форма и размеры лица;
- особенности голоса;
- биомеханические характеристики рукописной подписи;
- биомеханические характеристики "клавиатурного почерка".

При регистрации пользователь должен продемонстрировать один или несколько раз свои характерные биометрические признаки. Эти признаки (известные как подлинные) регистрируются системой как контрольный "образ" законного пользователя. Этот образ пользователя хранится в электронной форме и используется для проверки идентичности каждого, кто выдает себя за соответствующего законного пользователя. В зависимости от совпадения или несовпадения совокупности предъявленных признаков с зарегистрированными в контрольном образе их предъявивший признается законным пользователем (при совпадении) или нет (при несовпадении).

Системы идентификации по узору радужной оболочки и сетчатки глаз могут быть разделены на два класса:

- использующие рисунок радужной оболочки глаза;
- использующие рисунок кровеносных сосудов сетчатки глаза.

Поскольку вероятность повторения данных параметров равна 10^{-78}, эти системы являются наиболее надежными среди всех биометрических систем. Такие средства идентификации применяются там, где требуется высокий уровень безопасности (например, в США в зонах военных и оборонных объектов).

Системы идентификации по отпечаткам пальцев являются самыми распространенными. Одна из основных причин широкого распространения таких систем заключается в наличии больших банков данных по отпечаткам пальцев. Основными пользователями подобных систем во всем мире являются полиция, различные государственные и некоторые банковские организации.

Системы идентификации по геометрической форме руки используют сканеры формы руки, обычно устанавливаемые на стенах. Следует отметить, что подавляющее большинство пользователей предпочитают системы именно этого типа, а не описанные выше.

Системы идентификации по лицу и голосу являются наиболее доступными из-за их дешевизны, поскольку большинство со-

временных компьютеров имеют видео- и аудиосредства. Системы данного класса широко применяются при удаленной идентификации субъекта доступа в телекоммуникационных сетях.

Системы идентификации личностей по динамике рукописной подписи учитывают интенсивность каждого усилия подписывающего, частотные характеристики написания каждого элемента подписи и начертание подписи в целом.

Системы идентификации по биомеханическим характеристикам "клавиатурного почерка" основываются на том, что моменты нажатия и отпускания клавиш при наборе текста на клавиатуре существенно различаются у разных пользователей. Этот динамический ритм набора ("клавиатурный почерк") позволяет построить достаточно надежные средства идентификации. В случае обнаружения изменения клавиатурного почерка пользователя ему автоматически запрещается работа на ЭВМ.

Следует отметить, что применение биометрических параметров при идентификации субъектов доступа автоматизированных систем пока не получило надлежащего нормативно-правового обеспечения, в частности в виде стандартов. Поэтому применение систем биометрической идентификации допускается только в автоматизированных системах, обрабатывающих и хранящих персональные данные, составляющие коммерческую и служебную тайну [73].

5.3. Взаимная проверка подлинности пользователей

Обычно стороны, вступающие в информационный обмен, нуждаются во взаимной проверке подлинности (аутентификации) друг друга. Этот процесс взаимной аутентификации выполняют в начале сеанса связи.

Для проверки подлинности применяют следующие способы [55]:

- механизм запроса-ответа;
- механизм отметки времени ("временной штемпель").

Механизм запроса-ответа состоит в следующем. Если пользователь А хочет быть уверенным, что сообщения, получаемые им от пользователя В, не являются ложными, он включает в посылаемое для В сообщение непредсказуемый элемент – запрос X (например, некоторое случайное число). При ответе пользователь В должен выполнить над этим элементом некоторую операцию (например, вычислить некоторую функцию f(X)). Это невозможно осуществить заранее, так как пользователю В неизвестно, какое случайное число X придет в запросе. Получив ответ с результатом действий В, пользователь может быть уверен, что В – подлинный. Недостаток этого метода – возможность установления закономерности между запросом и ответом.

Механизм отметки времени подразумевает регистрацию времени для каждого сообщения. В этом случае каждый пользователь сети может определить, насколько "устарело" пришедшее сообщение, и решить не принимать его, поскольку оно может быть ложным.

В обоих случаях для защиты механизма контроля следует применять шифрование, чтобы быть уверенным, что ответ послан не злоумышленником.

При использовании отметок времени возникает проблема *допустимого временного интервала задержки* для подтверждения подлинности сеанса. Ведь сообщение с "временным штемпелем" в принципе не может быть передано мгновенно. Кроме того, компьютерные часы получателя и отправителя не могут быть абсолютно синхронизированы. Какое запаздывание "штемпеля" является подозрительным?

Для взаимной проверки подлинности обычно используют *процедуру "рукопожатия"* [55, 123]. Эта процедура базируется на указанных выше механизмах контроля и заключается во взаимной проверке ключей, используемых сторонами. Иначе говоря, стороны признают друг друга законными партнерами, если докажут друг другу, что обладают правильными ключами. Процедуру рукопожатия обычно применяют в компьютерных сетях при организации сеанса связи между пользователями, пользователем и хост-компьютером, между хост-компьютерами и т.д.

Рассмотрим в качестве примера процедуру рукопожатия для двух пользователей А и В. (Это допущение не влияет на общность

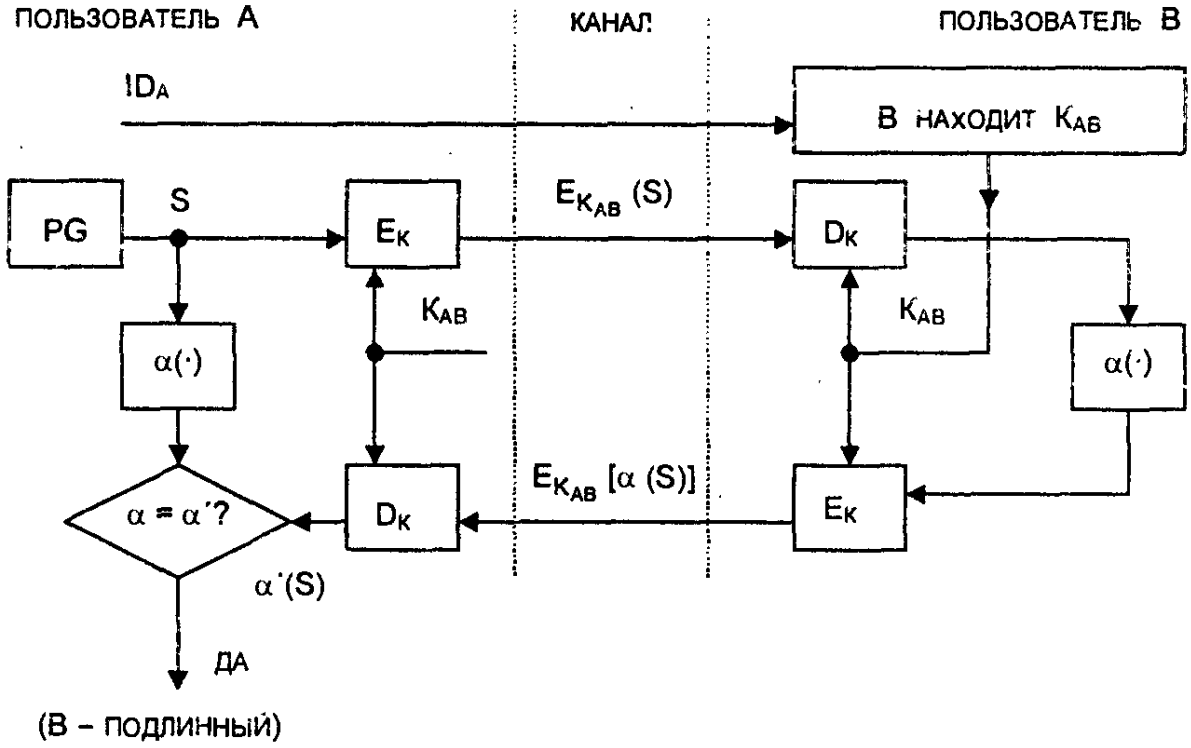


Рис. 5.3. Схема процедуры рукопожатия (пользователь А проверяет подлинность пользователя В)

рассмотрения. Такая же процедура используется, когда вступающие в связь стороны не являются пользователями). Пусть применяется симметричная криптосистема. Пользователи A и B разделяют один и тот же секретный ключ K_{AB}. Вся процедура показана на рис. 5.3.

- Пусть пользователь A инициирует процедуру рукопожатия, отправляя пользователю B свой идентификатор ID_A в открытой форме.
- Пользователь B, получив идентификатор ID_A, находит в базе данных секретный *ключ* K_{AB} и вводит его в свою криптосистему.
- Тем временем *пользователь A* генерирует случайную *последовательность S* с помощью псевдослучайного генератора PG и отправляет ее *пользователю B* в виде криптограммы

$$E_{K_{AB}}(S).$$

- Пользователь B расшифровывает эту криптограмму и раскрывает исходный вид последовательности S.
- Затем оба пользователя A и B преобразуют последовательность S, используя открытую одностороннюю функцию $\alpha(\cdot)$.
- Пользователь B шифрует сообщение $\alpha(S)$ и отправляет эту криптограмму *пользователю A*.
- Наконец, пользователь A расшифровывает эту криптограмму и сравнивает полученное сообщение $\alpha'(S)$ с исходным $\alpha(S)$. Если эти сообщения равны, пользователь A признает подлинность пользователя B.

Очевидно, пользователь B проверяет подлинность пользователя A таким же способом. Обе эти процедуры образуют процедуру рукопожатия, которая обычно выполняется в самом начале любого сеанса связи между любыми двумя сторонами в компьютерных сетях.

Достоинством модели рукопожатия является то, что ни один из участников сеанса связи не получает никакой секретной информации во время процедуры подтверждения подлинности.

Иногда пользователи хотят иметь непрерывную проверку подлинности отправителей в течение всего сеанса связи. Один из простейших способов непрерывной проверки подлинности показан на рис. 5.4 [123]. Передаваемая криптограмма имеет вид

$$E_K(ID_A, M),$$

где ID_A – идентификатор отправителя A; M – сообщение.

Получатель B, принявший эту криптограмму, расшифровывает ее и раскрывает пару (ID_A, M). Если принятый идентификатор ID_A совпадает с хранимым значением ID'_A, получатель B признает эту криптограмму.

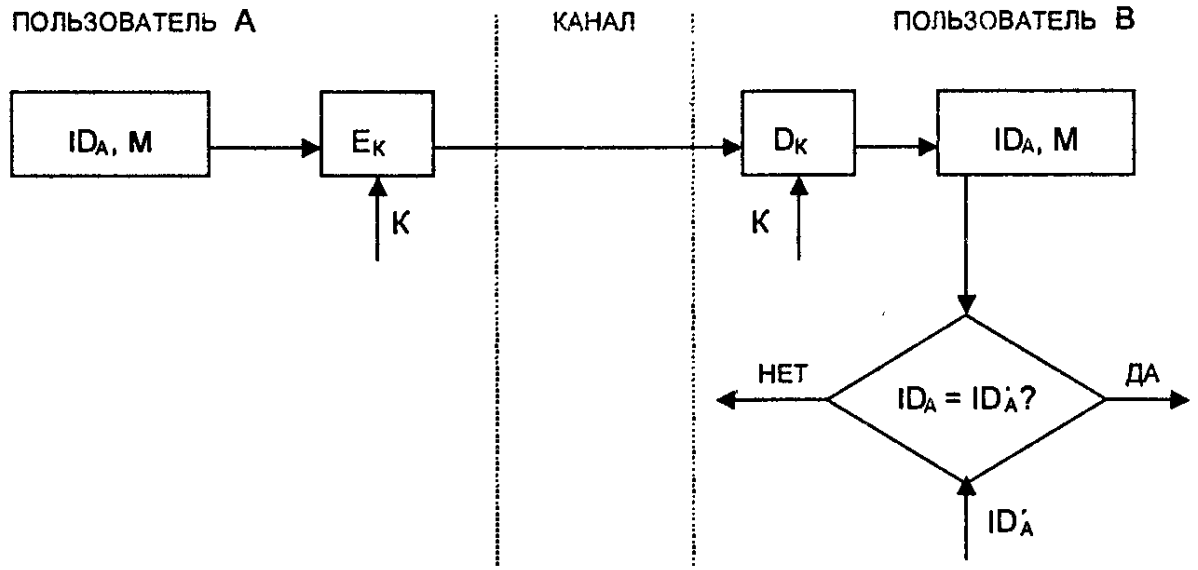


Рис. 5.4. Схема непрерывной проверки подлинности отправителя

Другой вариант непрерывной проверки подлинности использует вместо идентификатора отправителя его секретный пароль. Заранее подготовленные пароли известны обеим сторонам. Пусть P_A и P_B – пароли пользователей А и В соответственно. Тогда пользователь А создает криптограмму

$$C = E_K(P_A, M).$$

Получатель криптограммы расшифровывает ее и сравнивает пароль, извлеченный из этой криптограммы, с исходным значением. Если они равны, получатель признает эту криптограмму.

Процедура рукопожатия была рассмотрена в предположении, что пользователи А и В уже имеют общий *секретный сеансовый ключ*. Реальные процедуры предназначены для распределения ключей между подлинными партнерами и включает как этап распределения ключей, так и этап собственно подтверждения подлинности партнеров по информационному обмену. Такие процедуры будут рассмотрены в гл.7.

5.4. Протоколы идентификации с нулевой передачей знаний

Широкое распространение интеллектуальных карт (смарт-карт) для разнообразных коммерческих, гражданских и военных применений (кредитные карты, карты социального страхования, карты доступа в охраняемое помещение, компьютерные пароли и ключи, и т.п.) потребовало обеспечения безопасной идентификации таких карт и их владельцев. Во многих приложениях главная проблема заключается в том, чтобы при предъявлении интеллектуальной карты оперативно обнаружить обман и отказать обманщику в допуске, ответе или обслуживании.

Для безопасного использования интеллектуальных карт разработаны протоколы идентификации с нулевой передачей знаний [121]. Секретный ключ владельца карты становится неотъемлемым признаком его личности. Доказательство знания этого секретного ключа с нулевой передачей этого знания служит доказательством подлинности личности владельца карты.

Упрощенная схема идентификации с нулевой передачей знаний

Схему идентификации с нулевой передачей знаний предложили в 1986 г. У. Фейге, А. Фиат и А. Шамир. Она является наиболее известным доказательством идентичности с нулевой передачей конфиденциальной информации.

Рассмотрим сначала упрощенный вариант схемы идентификации с нулевой передачей знаний для более четкого выявления ее основной концепции. Прежде всего выбирают случайное значение модуля n, который является произведением двух больших простых чисел. Модуль n должен иметь длину 512...1024 бит. Это значение n может быть представлено группе пользователей, которым придется доказывать свою подлинность. В процессе идентификации участвуют две стороны:

- сторона A, доказывающая свою подлинность,
- сторона B, проверяющая представляемое стороной A доказательство.

Для того чтобы сгенерировать открытый и секретный ключи для стороны A, доверенный арбитр (Центр) выбирает некоторое число V, которое является квадратичным вычетом по модулю n. Иначе говоря, выбирается такое число V, что сравнение

$$x^2 \equiv V (\mathrm{mod}\ n)$$

имеет решение и существует целое число

$$V^{-1} \mathrm{mod}\ n.$$

Выбранное значение V является *открытым ключом* для A. Затем вычисляют наименьшее значение S, для которого

$$S \equiv \mathrm{sqrt}\,(V^{-1})\ (\mathrm{mod}\ n).$$

Это значение S является *секретным ключом* для A.

Теперь можно приступить к выполнению протокола идентификации.

1. Сторона A выбирает некоторое случайное число r, $r < n$. Затем она вычисляет

$$x = r^2 \mathrm{mod}\ n$$

и отправляет x стороне B.

2. Сторона B посылает A случайный бит b.

3. Если b = 0, тогда A отправляет r стороне B. Если b = 1, то A отправляет стороне B

$$y = r * S \bmod n.$$

4. Если b = 0, сторона B проверяет, что

$$x = r^2 \bmod n,$$

чтобы убедиться, что A знает sqrt(x). Если b = 1, сторона B проверяет, что

$$x = y^2 * V \bmod n,$$

чтобы быть уверенной, что A знает $\text{sqrt}(V^{-1})$.

Эти шаги образуют один цикл протокола, называемый *аккредитацией*. Стороны A и B повторяют этот цикл t раз при разных случайных значениях r и b до тех пор, пока B не убедится, что A знает значение S.

Если сторона A не знает значения S, она может выбрать такое значение r, которое позволит ей обмануть сторону B, если B отправит ей b = 0, либо A может выбрать такое r, которое позволит обмануть B, если B отправит ей b = 1. Но этого невозможно сделать в обоих случаях. Вероятность того, что A обманет B в одном цикле, составляет 1/2. Вероятность обмануть B в t циклах равна $(1/2)^t$.

Для того чтобы этот протокол работал, сторона A никогда не должна повторно использовать значение r. Если A поступила бы таким образом, а сторона B отправила бы стороне A на шаге 2 другой случайный бит b, то B имела бы оба ответа A. После этого B может вычислить значение S, и для A все закончено.

Параллельная схема идентификации с нулевой передачей знаний

Параллельная схема идентификации позволяет увеличить число аккредитаций, выполняемых за один цикл, и тем самым уменьшить длительность процесса идентификации.

Как и в предыдущем случае, сначала генерируется число n как произведение двух больших чисел. Для того, чтобы сгенерировать открытый и секретный ключи для стороны A, сначала выбирают K различных чисел $V_1, V_2, \ldots, V_K$, где каждое V_i является квадратичным вычетом по модулю n. Иначе говоря, выбирают значение V_i таким, что сравнение

$$x^2 \equiv V_i \bmod n$$

имеет решение и существует $V_i^{-1} \bmod n$. Полученная строка $V_1, V_2, \ldots, V_K$ является *открытым ключом*.

Затем вычисляют такие наименьшие значения S_i, что

$$S_i = \text{sqrt}\,(V_i^{-1}) \bmod n.$$

Эта строка $S_1, S_2, ..., S_K$ является *секретным ключом* стороны А.

Протокол процесса идентификации имеет следующий вид:

1. Сторона А выбирает некоторое случайное число r, $r < n$. Затем она вычисляет $x = r^2 \bmod n$ и посылает x стороне В.

2. Сторона В отправляет стороне А некоторую случайную двоичную строку из K бит: $b_1, b_2, ..., b_K$.

3. Сторона А вычисляет

$$y = r * (S_1^{b_1} * S_2^{b_2} * ... * S_K^{b_K}) \bmod n.$$

Перемножаются только те значения S_i, для которых $b_i = 1$. Например, если $b_1 = 1$, то сомножитель S_1 входит в произведение, если же $b_1 = 0$, то S_1 не входит в произведение, и т.д. Вычисленное значение y отправляется стороне В.

4. Сторона В проверяет, что

$$x = y^2 * (V_1^{b_1} * V_2^{b_2} * ... * V_K^{b_K}) \bmod n.$$

Фактически сторона В перемножает только те значения V_i, для которых $b_i = 1$. Стороны А и В повторяют этот протокол t раз, пока В не убедится, что А знает $S_1, S_2, ..., S_K$.

Вероятность того, что А может обмануть В, равна $(1/2)^{Kt}$. Авторы рекомендуют в качестве контрольного значения брать вероятность обмана В равной $(1/2)^{20}$ при $K = 5$ и $t = 4$.

Пример. Рассмотрим работу этого протокола для небольших числовых значений [102]. Если n = 35 (n – произведение двух простых чисел 5 и 7), то возможные квадратичные вычеты будут следующими:

1: $x^2 \equiv 1 \pmod{35}$ имеет решения: x = 1, 6, 29, 34;

4: $x^2 \equiv 4 \pmod{35}$ имеет решения: x = 2, 12, 23, 33;

9: $x^2 \equiv 9 \pmod{35}$ имеет решения: x = 3, 17, 18, 32;

11: $x^2 \equiv 11 \pmod{35}$ имеет решения: x = 9, 16, 19, 26;

14: $x^2 \equiv 14 \pmod{35}$ имеет решения: x = 7, 28;

15: $x^2 \equiv 15 \pmod{35}$ имеет решения: x = 15, 20;

16: $x^2 \equiv 16 \pmod{35}$ имеет решения: x = 4, 11, 24, 31;

21: $x^2 \equiv 21 \pmod{35}$ имеет решения: x = 14, 21;

25: $x^2 \equiv 25 \pmod{35}$ имеет решения: x = 5, 30;

29: $x^2 \equiv 29 \pmod{35}$ имеет решения: x = 8, 13, 22, 27;

30: $x^2 \equiv 30 \pmod{35}$ имеет решения: x = 10, 25.

Заметим, что 14, 15, 21, 25 и 30 не имеют обратных значений по модулю 35, потому что они не являются взаимно простыми с 35. Следует также отметить, что число квадратичных вычетоа по модулю 35, взаимно простых с $n = p * q = 5 * 7 = 35$ (для которых НОД (x, 35) = 1), равно

$$(p - 1)(q - 1)/4 = (5 - 1)(7 - 1)/4 = 6.$$

Состааим таблицу каадратичных вачетов по модулю 35, обратных к ним значений по модулю 35 и их каадратных корней.

V	V^{-1}	$S = \text{sqrt}(V^{-1})$
1	1	1
4	9	3
9	4	2
11	16	4
16	11	9
29	29	8

Итак, сторона A получает открытый ключ, состоящий из K = 4 значений V:

[4, 11, 16, 29].

Соответствующий секретный ключ, состоящий из K = 4 значений S:

[3 4 9 8].

Рассмотрим один цикл протокола.

1. Сторона A выбирает некоторое случайное число r = 16, вычисляет

$$x = 16^2 \bmod 35 = 11$$

и посылает это значение x стороне B.

2. Сторона B отправляет стороне A некоторую случайную двоичную строку

[1, 1, 0, 1].

3. Сторона A вычисляет значение

$$y = r * (S_1^{b_1} * S_2^{b_2} * ... * S_K^{b_K}) \bmod n = 16 * (3^1 * 4^1 * 9^0 * 8^1) \bmod 35 = 31$$

и отправляет это значение y стороне B.

4. Сторона B проверяет, что

$$x = y^2 * (V_1^{b_1} * V_2^{b_2} * ... * V_K^{b_K}) \bmod n = 31^2 * (4^1 * 11^1 * 16^0 * 29^1) \bmod 35 = 11.$$

Стороны A и B повторяют этот протокол t раз, каждый раз с разным случайным числом r, пока сторона B не будет удовлетворена.

При малых значениях величин, как в данном примере, не достигается настоящей безопасности. Но если n представляет собой число длиной 512 бит и более, сторона B не сможет узнать ничего о секретном ключе стороны A, кроме того факта, что сторона A знает этот ключ.

В этот протокол можно включить идентификационную информацию [123].

Пусть I – некоторая двоичная строка, представляющая идентификационную информацию о владельце карты (имя, адрес, персональный идентификационный номер, физическое описание) и о карте (дата окончания действия и т. п.). Эту информацию I формируют в Центре выдачи интеллектуальных карт по заявке пользователя A.

Далее используют одностороннюю функцию $f(\cdot)$ для вычисления $f(I, j)$, где j – некоторое двоичное число, сцепляемое со строкой I. Вычисляют значения

$$V_j = f(I, j)$$

для небольших значений j, отбирают K разных значений j, для которых V_j являются квадратичными вычетами по модулю n. Затем для отобранных квадратичных вычетов V_j вычисляют наименьшие квадратные корни из $V_j^{-1}(\mathrm{mod}\, n)$. Совокупность из K значений V_j образует открытый ключ, а совокупность из K значений S_j – секретный ключ пользователя A.

Схема идентификации Гиллоу–Куискуотера

Алгоритм идентификации с нулевой передачей знания, разработанный Л. Гиллоу и Ж. Куискуотером [121], имеет несколько лучшие характеристики, чем предыдущая схема идентификации. В этом алгоритме обмены между сторонами A и B и аккредитации в каждом обмене доведены до абсолютного минимума – для каждого доказательства требуется только один обмен с одной аккредитацией. Однако объем требуемых вычислений для этого алгоритма больше, чем для схемы Фейге–Фиата–Шамира.

Пусть сторона A – интеллектуальная карточка, которая должна доказать свою подлинность проверяющей стороне B. Идентификационная информация стороны A представляет собой битовую строку I, которая включает имя владельца карточки, срок действия, номер банковского счета и др. Фактически идентификационные данные могут занимать достаточно длинную строку, и тогда их хэшируют к значению I.

Строка I является аналогом открытого ключа. Другой открытой информацией, которую используют все карты, участвующие в данном приложении, являются модуль n и показатель степени V. Модуль n является произведением двух секретных простых чисел.

Секретным ключом стороны A является величина G, выбираемая таким образом, чтобы выполнялось соотношение

$$I * G^V \equiv 1(\mathrm{mod}\, n).$$

Сторона A отправляет стороне B свои идентификационные данные I. Далее ей нужно доказать стороне B, что эти идентификационные данные принадлежат именно ей. Чтобы добиться этого, сторона A должна убедить сторону B, что ей известно значение G.

Вот протокол доказательства подлинности A без передачи стороне B значения G:

1. Сторона A выбирает случайное целое r, такое, что $1<r\le n-1$. Она вычисляет

$$T = r^V \bmod n$$

и отправляет это значение стороне B.

2. Сторона B выбирает случайное целое d, такое, что $1<d\le n-1$, и отправляет это значение d стороне A.

3. Сторона A вычисляет

$$D = r * G^d \bmod n$$

и отправляет это значение стороне B.

4. Сторона B вычисляет значение

$$T' = D^V I^d \bmod n.$$

Если $T \equiv T' (\bmod n)$,

то проверка подлинности успешно завершена.

Математические выкладки, использованные в этом протоколе не очень сложны:

$$T' = D^V I^d = (rG^d)^V I^d = r^V G^{dV} I^d = r^V (IG^V)^d = r^V \equiv T (\bmod n);$$

поскольку G вычислялось таким образом, чтобы выполнялось со отношение

$$IG^V \equiv 1 (\bmod n).$$

ГЛАВА 6. ЭЛЕКТРОННАЯ ЦИФРОВАЯ ПОДПИСЬ

6.1. Проблема аутентификации данных и электронная цифровая подпись

При обмене электронными документами по сети связи существенно снижаются затраты на обработку и хранение документов, убыстряется их поиск. Но при этом возникает проблема аутентификации автора документа и самого документа, т.е. установления подлинности автора и отсутствия изменений в полученном документе. В обычной (бумажной) информатике эти проблемы решаются за счет того, что информация в документе и рукописная подпись автора жестко связаны с физическим носителем (бумагой). В электронных документах на машинных носителях такой связи нет.

Целью аутентификации злектронных документов является их защита от возможных видов злоумышленных действий, к которым относятся:

- активный перехват – нарушитель, подключившийся к сети, перехватывает документы (файлы) и изменяет их;
- маскарад – абонент С посылает документ абоненту В от имени абонента А;
- ренегатство – абонент А заявляет, что не посылал сообщения абоненту В, хотя на самом деле послал;
- подмена – абонент В изменяет или формирует новый документ и заявляет, что получил его от абонента А;
- повтор – абонент С повторяет ранее переданный документ, который абонент А посылал абоненту В.

Эти виды злоумышленных действий могут нанести существенный ущерб банковским и коммерческим структурам, государственным предприятиям и организациям, частным лицам, применяющим в своей деятельности компьютерные информационные технологии.

При обработке документов в электронной форме совершенно непригодны традиционные способы установления подлинности по рукописной подписи и оттиску печати на бумажном документе. Принципиально новым решением является электронная цифровая подпись (ЭЦП).

Электронная цифровая подпись используется для аутентификации текстов, передаваемых по телекоммуникационным каналам. Функционально она аналогична обычной рукописной подписи и обладает ее основными достоинствами:

- удостоверяет, что подписанный текст исходит от лица, поставившего подпись;
- не дает самому этому лицу возможности отказаться от обязательств, связанных с подписанным текстом;
- гарантирует целостность подписанного текста.

Цифровая подпись представляет собой относительно небольшое количество дополнительной цифровой информации, передаваемой вместе с подписываемым текстом.

Система ЭЦП включает две процедуры: 1) процедуру постановки подписи; 2) процедуру проверки подписи. В процедуре постановки подписи используется секретный ключ отправителя сообщения, в процедуре проверки подписи – открытый ключ отправителя.

При формировании ЭЦП отправитель прежде всего вычисляет хэш-функцию h(M) подписываемого текста M. Вычисленное значение хэш-функции h(M) представляет собой один короткий блок информации m, характеризующий весь текст M в целом. Затем число m шифруется секретным ключом отправителя. Получаемая при этом пара чисел представляет собой ЭЦП для данного текста M.

При проверке ЭЦП получатель сообщения снова вычисляет хэш-функцию m = h(M) принятого по каналу текста M, после чего при помощи открытого ключа отправителя проверяет, соответствует ли полученная подпись вычисленному значению m хэш-функции.

Принципиальным моментом в системе ЭЦП является невозможность подделки ЭЦП пользователя без знания его секретного ключа подписывания.

В качестве подписываемого документа может быть использован любой файл. Подписанный файл создается из неподписанного путем добавления в него одной или более электронных подписей.

Каждая подпись содержит следующую информацию:

- дату подписи;
- срок окончания действия ключа данной подписи;
- информацию о лице, подписавшем файл (Ф.И.О., должность, краткое наименование фирмы);
- идентификатор подписавшего (имя открытого ключа);
- собственно цифровую подпись.

6.2. Однонаправленные хэш-функции

Хэш-функция предназначена для сжатия подписываемого документа M до нескольких десятков или сотен бит. Хэш-функция h(·) принимает в качестве аргумента сообщение (документ) M произ-

вольной длины и возвращает хэш-значение h(M) = H фиксированной длины. Обычно хэшированная информация является сжатым двоичным представлением основного сообщения произвольной длины. Следует отметить, что значение хэш-функции h(M) сложным образом зависит от документа M и не позволяет восстановить сам документ M.

Хэш-функция должна удовлетворять целому ряду условий:

- хэш-функция должна быть чувствительна к всевозможным изменениям в тексте M, таким как вставки, выбросы, перестановки и т.п.;
- хэш-функция должна обладать свойством необратимости, то есть задача подбора документа M′, который обладал бы требуемым значением хэш-функции, должна быть вычислительно неразрешима;
- вероятность того, что значения хэш-функций двух различных документов (вне зависимости от их длин) совпадут, должна быть ничтожно мала [123].

Большинство хэш-функций строится на основе однонаправленной функции f(·), которая образует выходное значение длиной n при задании двух входных значений длиной n. Этими входами являются блок исходного текста M_i и хэш-значение H_{i-1} предыдущего блока текста (рис. 6.1):

$$H_i = f(M_i, H_{i-1}).$$

Хэш-значение, вычисляемое при вводе последнего блока текста, становится хэш-значением всего сообщения M.

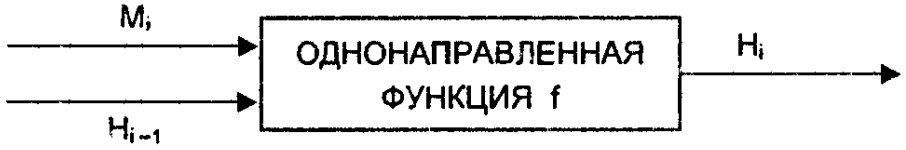


Рис. 6.1. Построение однонаправленной хэш-функции

В результате однонаправленная хэш-функция всегда формирует выход фиксированной длины n (независимо от длины входного текста).

Однонаправленные хэш-функции на основе симметричных блочных алгоритмов

Однонаправленную хэш-функцию можно построить, используя симметричный блочный алгоритм. Наиболее очевидный подход состоит в том, чтобы шифровать сообщение M посредством блочного алгоритма в режиме CBC или CFB с помощью фиксированного ключа и некоторого вектора инициализации IV. Последний блок шифртекста можно рассматривать в качестве хэш-значения сооб-

щения M. При таком подходе не всегда возможно построить безопасную однонаправленную хэш-функцию, но всегда можно получить код аутентификации сообщения MAC (Message Authentication Code).

Более безопасный вариант хэш-функции можно получить, используя блок сообщения в качестве ключа, предыдущее хэш-значение – в качестве входа, а текущее хэш-значение – в качестве выхода. Реальные хэш-функции проектируются еще более сложными. Длина блока обычно определяется длиной ключа, а длина хэш-значения совпадает с длиной блока.

Поскольку большинство блочных алгоритмов являются 64-битовыми, некоторые схемы хэширования проектируют так, чтобы хэш-значение имело длину, равную двойной длине блока.

Если принять, что получаемая хэш-функция корректна, безопасность схемы хэширования базируется на безопасности лежащего в ее основе блочного алгоритма. Схема хэширования, у которой длина хэш-значения равна длине блока, показана на рис.6.2. Ее работа описывается выражениями:

$$H_0 = I_H,$$

$$H_i = E_A(B) \oplus C,$$

где I_H – некоторое случайное начальное значение; A, B и C могут принимать значения M_i, H_{i-1}, $(M_i \oplus H_{i-1})$ или быть константами.

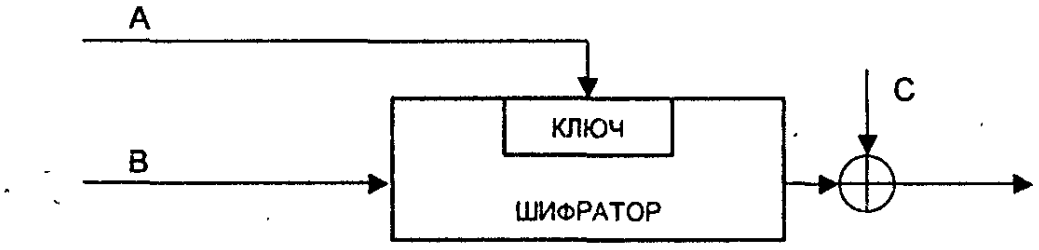


Рис.6.2. Обобщенная схема формирования хэш-функции

Сообщение M разбивается на блоки M_i принятой длины, которые обрабатываются поочередно.

Три различные переменные A, B и C могут принимать одно из четырех возможных значений, поэтому в принципе можно получить 64 варианта общей схемы этого типа. Из них 52 варианта являются либо тривиально слабыми, либо небезопасными. Остальные 12 безопасных схем хэширования перечислены в табл.6.1[121].

Таблица 6.1

Схемы безопасного хэширования, у которых длина хэш-значения равна длине блока

Номер схемы	Функция хэширования
1	$H_i = E_{H_{i-1}}(M_i) \oplus M_i$
2	$H_i = E_{H_{i-1}}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1}$
3	$H_i = E_{H_{i-1}}(M_i) \oplus H_{i-1} \oplus M_i$
4	$H_i = E_{H_{i-1}}(M_i \oplus H_{i-1}) \oplus M_i$
5	$H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1}$
6	$H_i = E_{M_i}(M_i \oplus H_{i-1})\ M_i \oplus H_{i-1}$
7	$H_i = E_{M_i}(H_{i-1})\ M_i \oplus H_{i-1}$
8	$H_i = E_{M_i}(M_i \oplus H_{i-1}) \oplus H_{i-1}$
9	$H_i = E_{M_i \oplus H_{i-1}}(M_i) \oplus M_i$
10	$H_i = E_{M_i \oplus H_{i-1}}(H_{i-1}) \oplus H_{i-1}$
11	$H_i = E_{M_i \oplus H_{i-1}}(M_i) \oplus H_{i-1}$
12	$H_i = E_{M_i \oplus H_{i-1}}(H_{i-1}) \oplus M_i$

Первые четыре схемы хэширования, являющиеся безопасными при всех атаках, приведены на рис. 6.3.

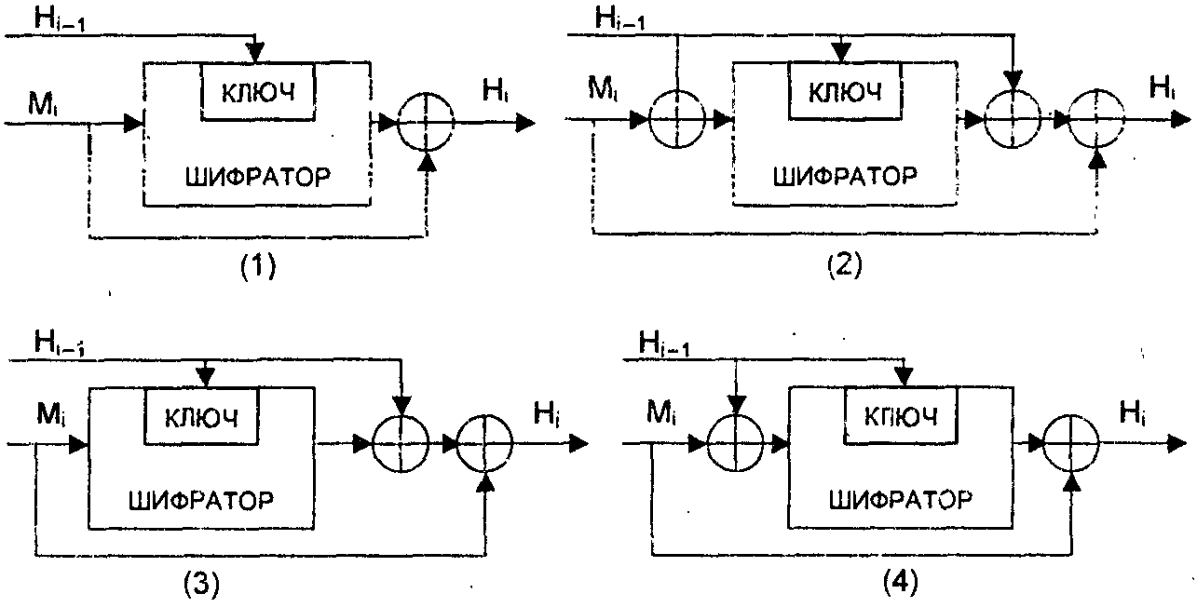


Рис. 6.3. Четыре схемы безопасного хэширования

Отечественный стандарт хэш-функции

Российский стандарт ГОСТ Р 34.11-94 определяет алгоритм и процедуру вычисления хэш-функции для любых последовательностей двоичных символов, применяемых в криптографических методах обработки и защиты информации. Этот стандарт базируется на блочном алгоритме шифрования ГОСТ 28147-89, хотя в принципе можно было бы использовать и другой блочный алгоритм шифрования с 64-битовым блоком и 256-битовым ключом.

Данная хэш-функция формирует 256-битовое хэш-значение.

Функция сжатия $H_i = f(M_i, H_{i-1})$ (оба операнда M_i и H_{i-1} являются 256-битовыми величинами) определяется следующим образом:

1. Генерируются 4 ключа шифрования K_j, $j = 1...4$, путем линейного смешивания M_i, H_{i-1} и некоторых констант C_j.

2. Каждый ключ K_j, используют для шифрования 64-битовых подслов h_j слова H_{i-1} в режиме простой замены: $S_j = E_{K_j}(h_j)$. Результирующая последовательность S_4, S_3, S_2, S_1 длиной 256 бит запоминается во временной переменной S.

3. Значение H_i является сложной, хотя и линейной функцией смешивания S, M_i и H_{i-1}.

При вычислении окончательного хэш-значения сообщения M учитываются значения трех связанных между собой переменных:

H_n – хэш-значение последнего блока сообщения;

Z – значение контрольной суммы, получаемой при сложении по модулю 2 всех блоков сообщения;

L – длина сообщения.

Эти три переменные и дополненный последний блок M′ сообщения объединяются в окончательное хэш-значение следующим образом:

$$H = f(Z \oplus M', f(L, f(M', H_n))).$$

Данная хэш-функция определена стандартом ГОСТ Р 34.11-94 для использования совместно с российским стандартом электронной цифровой подписи [40, 41].

6.3. Алгоритмы электронной цифровой подписи

Технология применения системы ЭЦП предполагает наличие сети абонентов, посылающих друг другу подписанные электронные документы. Для каждого абонента генерируется пара ключей: секретный и открытый. Секретный ключ хранится абонентом в тайне и используется им для формирования ЭЦП. Открытый ключ известен всем другим пользователям и предназначен для проверки ЭЦП получателем подписанного электронного документа. Иначе говоря, открытый ключ является необходимым инструментом, позволяющим проверить подлинность электронного документа и автора подписи. Открытый ключ не позволяет вычислить секретный ключ.

Для генерации пары ключей (секретного и открытого) в алгоритмах ЭЦП, как и в асимметричных системах шифрования, используются разные математические схемы, основанные на применении однонаправленных функций. Эти схемы разделяются на две группы. В основе такого разделения лежат известные сложные вычислительные задачи:

- задача факторизации (разложения на множители) больших целых чисел;
- задача дискретного логарифмирования.

Алгоритм цифровой подписи RSA

Первой и наиболее известной во всем мире конкретной системой ЭЦП стала система RSA, математическая схема которой была разработана в 1977 г. в Массачуссетском технологическом институте США.

Сначала необходимо вычислить пару ключей (секретный ключ и открытый ключ). Для этого отправитель (автор) электронных документов вычисляет два больших простых числа P и Q, затем находит их произведение

$$N = P * Q$$

и значение функции

$$\varphi(N) = (P-1)(Q-1).$$

Далее отправитель вычисляет число E из условий:

$$E \le \varphi(N), \quad \text{НОД}(E, \varphi(N)) = 1$$

и число D из условий:

$$D < N, \quad E * D \equiv 1 (\text{mod } \varphi(N)).$$

Пара чисел (E,N) является открытым ключом. Эту пару чисел автор передает партнерам по переписке для проверки его цифровых подписей. Число D сохраняется автором как секретный ключ для подписывания.

Обобщенная схема формирования и проверки цифровой подписи RSA показана на рис. 6.4.

Допустим, что отправитель хочет подписать сообщение M перед его отправкой. Сначала сообщение M (блок информации, файл, таблица) сжимают с помощью хэш-функции $h(\cdot)$ в целое число m:

$$m = h(M).$$

Затем вычисляют цифровую подпись S под электронным документом M, используя хэш-значение m и секретный ключ D:

$$S = m^D (\text{mod } N).$$

Пара (M, S) передается партнеру-получателю как электронный документ M, подписанный цифровой подписью S, причем подпись S сформирована обладателем секретного ключа D.

После приема пары (M, S) получатель вычисляет хэш-значение сообщения M двумя разными способами. Прежде всего он

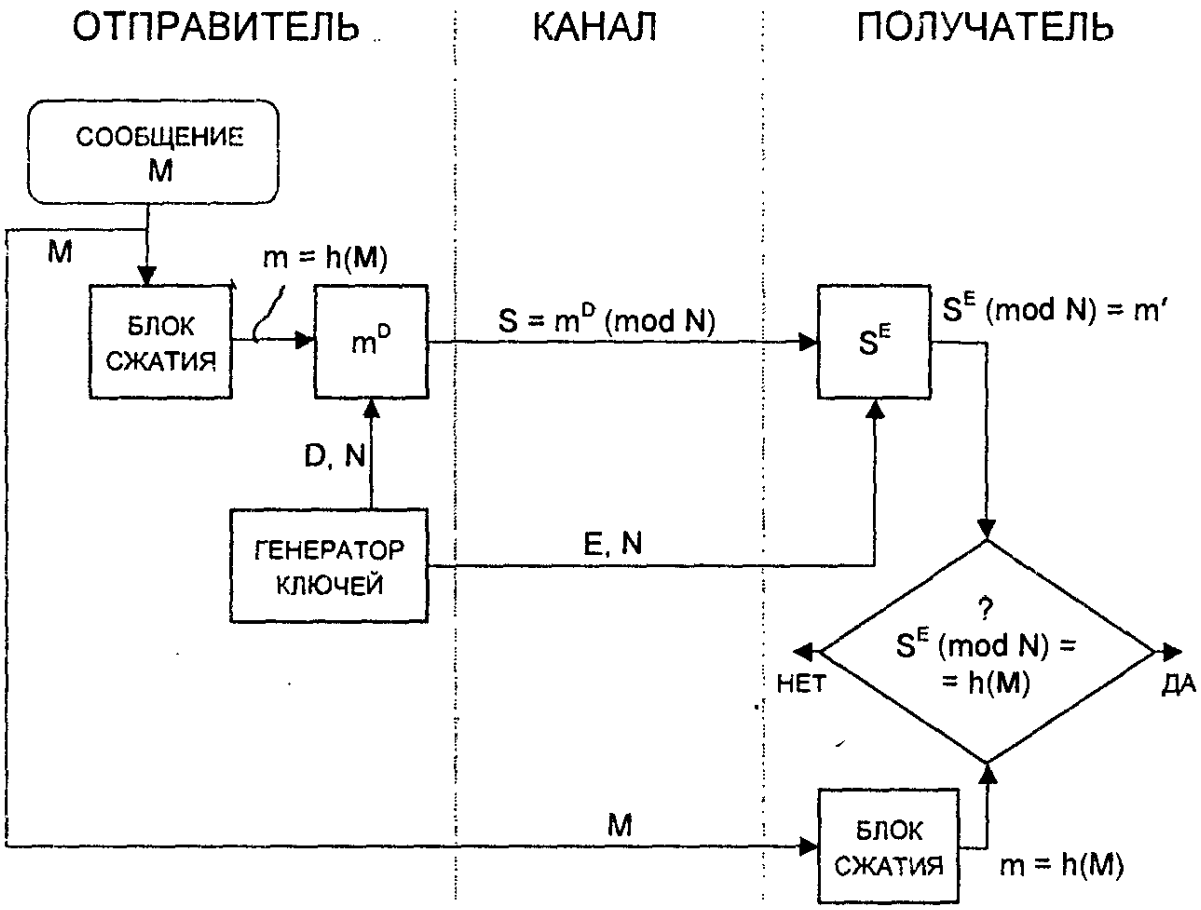


Рис. 6.4. Обобщенная схема цифровой подписи RSA

восстанавливает хэш-значение m', применяя криптографическое преобразование подписи S с использованием открытого ключа E:

$$m' = S^E \pmod N.$$

Кроме того, он находит результат хэширования принятого сообщения M с помощью такой же хэш-функции h(·):

$$m = h(M).$$

Если соблюдается равенство вычисленных значений, т. е.

$$S^E \pmod N = h(M),$$

то получатель признает пару (M, S) подлинной. Доказано, что только обладатель секретного ключа D может сформировать цифровую подпись S по документу M, а определить секретное число D по открытому числу E не легче, чем разложить модуль N на множители.

Кроме того, можно строго математически доказать, что результат проверки цифровой подписи S будет положительным только в том случае, если при вычислении S был использован секретный ключ D, соответствующий открытому ключу E. Поэтому открытый ключ E иногда называют "идентификатором" подписавшего.

Недостатки алгоритма цифровой подписи RSA.

1. При вычислении модуля N, ключей E и D для системы цифровой подписи RSA необходимо проверять большое количество дополнительных условий, что сделать практически трудно. Невыполнение любого из этих условий делает возможным фальсификацию цифровой подписи со стороны того, кто обнаружит такое невыполнение. При подписании важных документов нельзя допускать такую возможность даже теоретически.

2. Для обеспечения криптостойкости цифровой подписи RSA по отношению к попыткам фальсификации на уровне, например, национального стандарта США на шифрование информации (алгоритм DES), т.е. 10^{18}, необходимо использовать при вычислениях N, D и E целые числа не менее 2^{512} (или около 10^{154}) каждое, что требует больших вычислительных затрат, превышающих на 20...30 % вычислительные затраты других алгоритмов цифровой подписи при сохранении того же уровня криптостойкости.

3. Цифровая подпись RSA уязвима к так называемой мультипликативной атаке. Иначе говоря, алгоритм цифровой подписи RSA позволяет злоумышленнику без знания секретного ключа D сформировать подписи под теми документами, у которых результат хэширования можно вычислить как произведение результатов хэширования уже подписанных документов.

Пример. Допустим, что злоумышленник может сконструировать три сообщения M_1, M_2 и M_3, у которых хэш-значения

$$m_1 = h(M_1), \quad m_2 = h(M_2), \quad m_3 = h(M_3),$$

причем

$$m_3 = m_1 * m_2 \pmod N.$$

Допустим также, что для двух сообщений M_1 и M_2 получены законные подписи

$$S_1 = m_1^D \pmod N \quad \text{и} \quad S_2 = m_2^D \pmod N.$$

Тогда злоумышленник может легко вычислить подпись S_3 для документа M_3, даже не зная секретного ключа D:

$$S_3 = S_1 * S_2 \pmod N.$$

Действительно,

$$S_1 * S_2 \pmod N = m_1^D * m_2^D \pmod N = (m_1 m_2)^D \pmod N = m_3^D \pmod N = S_3.$$

Более надежный и удобный для реализации на персональных компьютерах алгоритм цифровой подписи был разработан в 1984 г. американцем арабского происхождения Тахером Эль Гамалем. В 1991 г. НИСТ США обосновал перед комиссией Конгресса США выбор алгоритма цифровой подписи Эль Гамаля в качестве основы для национального стандарта.

Алгоритм цифровой подписи Эль Гамаля (EGSA)

Название EGSA происходит от слов El Gamal Signature Algorithm (алгоритм цифровой подписи Эль Гамаля). Идея EGSA основана на том, что для обоснования практической невозможности фальсификации цифровой подписи может быть использована

более сложная вычислительная задача, чем разложение на множители большого целого числа, – задача дискретного логарифмирования. Кроме того, Эль Гамалю удалось избежать явной слабости алгоритма цифровой подписи RSA, связанной с возможностью подделки цифровой подписи под некоторыми сообщениями без определения секретного ключа.

Рассмотрим подробнее алгоритм цифровой подписи Эль Гамаля. Для того чтобы генерировать пару ключей (открытый ключ – секретный ключ), сначала выбирают некоторое большое простое целое число P и большое целое число G, причем $G<P$. Отправитель и получатель подписанного документа используют при вычислениях одинаковые большие целые числа P ($\sim 10^{308}$ или $\sim 2^{1024}$) и G ($\sim 10^{154}$ или $\sim 2^{512}$), которые не являются секретными.

Отправитель выбирает случайное целое число $X, 1<X\leq(P-1)$, и вычисляет

$$Y = G^X \bmod P.$$

Число Y является открытым ключом, используемым для проверки подписи отправителя. Число Y открыто передается всем потенциальным получателям документов.

Число X является секретным ключом отправителя для подписывания документов и должно храниться в секрете.

Для того чтобы подписать сообщение M, сначала отправитель хэширует его с помощью хэш-функции $h(\cdot)$ в целое число m:

$$m = h(M), \quad 1< m < (P-1),$$

и генерирует случайное целое число K, $1< K< (P-1)$, такое, что K и $(P-1)$ являются взаимно простыми. Затем отправитель вычисляет целое число a:

$$a = G^K \bmod P$$

и, применяя расширенный алгоритм Евклида, вычисляет с помощью секретного ключа X целое число b из уравнения

$$m = X * a + K * b \ (\mathrm{mod}\ (P-1)).$$

Пара чисел (a, b) образует цифровую подпись S:

$$S = (a, b),$$

проставляемую под документом M.

Тройка чисел (M, a, b) передается получателю, в то время как пара чисел (X,K) держится в секрете.

После приема подписанного сообщения (M, a, b) получатель должен проверить, соответствует ли подпись $S = (a, b)$ сообщению M. Для этого получатель сначала вычисляет по принятому сообщению M число

$$m = h(M),$$

т. е. хэширует принятое сообщение M.

Затем получатель вычисляет значение

$$A = Y^a a^b \pmod P$$

и признает сообщение M подлинным, если, и только если

$$A = G^m \pmod P.$$

Иначе говоря, получатель проверяет справедливость соотношения

$$Y^a a^b \pmod P = G^m \pmod P.$$

Можно строго математически доказать, что последнее равенство будет выполняться тогда, и только тогда, когда подпись $S = (a, b)$ под документом M получена с помощью именно того секретного ключа X, из которого был получен открытый ключ Y. Таким образом, можно надежно удостовериться, что отправителем сообщения M был обладатель именно данного секретного ключа X, не раскрывая при этом сам ключ, и что отправитель подписал именно этот конкретный документ M.

Следует отметить, что выполнение каждой подписи по методу Эль Гамаля требует нового значения K, причем это значение должно выбираться случайным образом. Если нарушитель раскроет когда-либо значение K, повторно используемое отправителем, то он сможет раскрыть секретный ключ X отправителя.

Пример. Выберем: числа $P=11$, $G=2$ и секретный ключ $X=8$. Вычисляем значение открытого ключа:

$$Y = G^X \bmod P = Y = 2^8 \bmod 11 = 3.$$

Предположим, что исходное сообщение M характеризуется хэш-значением $m = 5$.

Для того чтобы вычислить цифровую подпись для сообщения M, имеющего хэш-значение $m=5$, сначала выберем случайное целое число $K=9$. Убедимся, что числа K и $(P-1)$ являются взаимно простыми. Действительно,

$$\text{НОД}\,(9, 10) = 1.$$

Далее вычисляем элементы a и b подписи:

$$a = G^K \bmod P = 2^9 \bmod 11 = 6,$$

элемент b определяем, используя расширенный алгоритм Евклида:

$$m = X * a + K * b \pmod{(P-1)}.$$

При $m=5$, $a=6$, $X=8$, $K=9$, $P=11$ получаем

$$5 = (6 * 8 + 9 * b) \pmod{10}$$

или

$$9 * b \equiv -43 \pmod{10}.$$

Решение: $b=3$. Цифровая подпись представляет собой пару: $a=6$, $b=3$.

Далее отправитель передает подписанное сообщение. Приняв подписанное сообщение и открытый ключ $Y=3$, получатель вычисляет хэш-значение для сообщения M: $m=5$, а затем вычисляет два числа:

1) $Y^a a^b \pmod P = 3^6 * 6^3 \pmod{11} = 10 \pmod{11}$;

2) $G^m \pmod P = 2^5 \pmod{11} = 10 \pmod{11}$.

Так как эти два целых числа равны, принятое получателем сообщение признается подлинным.

Следует отметить, что схема Эль Гамаля является характерным примером подхода, который допускает пересылку сообщения M в открытой форме вместе с присоединенным аутентификатором (a, b). В таких случаях процедура установления подлинности принятого сообщения состоит в проверке соответствия аутентификатора сообщению.

Схема цифровой подписи Эль Гамаля имеет ряд преимуществ по сравнению со схемой цифровой подписи RSA:

1. При заданном уровне стойкости алгоритма цифровой подписи целые числа, участвующие в вычислениях, имеют запись на 25 % короче, что уменьшает сложность вычислений почти в два раза и позволяет заметно сократить объем используемой памяти.

2. При выборе модуля P достаточно проверить, что это число является простым и что у числа $(P-1)$ имеется большой простой множитель (т.е. всего два достаточно просто проверяемых условия).

3. Процедура формирования подписи по схеме Эль Гамаля не позволяет вычислять цифровые подписи под новыми сообщениями без знания секретного ключа (как в RSA).

Однако алгоритм цифровой подписи Эль Гамаля имеет и некоторые недостатки по сравнению со схемой подписи RSA. В частности, длина цифровой подписи получается в 1,5 раза больше, что, в свою очередь, увеличивает время ее вычисления.

Алгоритм цифровой подписи DSA

Алгоритм цифровой подписи DSA (Digital Signature Algorithm) предложен в 1991 г. в НИСТ США для использования в стандарте цифровой подписи DSS (Digital Signature Standard). Алгоритм DSA является развитием алгоритмов цифровой подписи Эль Гамаля и К. Шнорра [121].

Отправитель и получатель электронного документа используют при вычислении большие целые числа: G и P – простые числа, L бит каждое ($512 \le L \le 1024$); q – простое число длиной 160 бит (делитель числа $(P-1)$). Числа G, P, q являются открытыми и могут быть общими для всех пользователей сети.

Отправитель выбирает случайное целое число $X, 1<X<q$. Число X является секретным ключом отправителя для формирования электронной цифровой подписи.

Затем отправитель вычисляет значение

$$Y = G^X \bmod P.$$

Число Y является открытым ключом для проверки подписи отправителя. Число Y передается всем получателям документов.

Этот алгоритм также предусматривает использование односторонней функции хэширования $h(\cdot)$. В стандарте DSS определен алгоритм безопасного хэширования SHA (Secure Hash Algorithm).

Для того чтобы подписать документ M, отправитель хэширует его в целое хэш-значение m:

$$m = h(M), \quad 1 < m < q,$$

затем генерирует случайное целое число K, $1<K<q$, и вычисляет число r:

$$r = (G^K \bmod P) \bmod q.$$

Затем отправитель вычисляет с помощью секретного ключа X целое число s:

$$s = \frac{m + r * X}{K} \bmod q.$$

Пара чисел r и s образует цифровую подпись

$$S = (r, s)$$

под документом M.

Таким образом, подписанное сообщение представляет собой тройку чисел [M, r, s].

Получатель подписанного сообщения [M, r, s] проверяет выполнение условий

$$0 < r < q, \quad 0 < s < q$$

и отвергает подпись, если хотя бы одно из этих условий не выполнено.

Затем получатель вычисляет значение

$$w = \frac{1}{s} \bmod q,$$

хэш-значение

$$m = h(M)$$

и числа

$$u_1 = (m * w) \bmod q,$$

$$u_2 = (r * w) \bmod q.$$

Далее получатель с помощью открытого ключа Y вычисляет значение

$$v = ((G^{u_1} * Y^{u_2}) \bmod P) \bmod q$$

и проверяет выполнение условия

$$v = r.$$

Если условие v = r выполняется, тогда подпись S = (r, s) под документом M признается получателем подлинной.

Можно строго математически доказать, что последнее равенство будет выполняться тогда, и только тогда, когда подпись S = = (r, s) под документом M получена с помощью именно того секрет-

ного ключа X, из которого был получен открытый ключ Y. Таким образом, можно надежно удостовериться, что отправитель сообщения владеет именно данным секретным ключом X (не раскрывая при этом значения ключа X) и что отправитель подписал именно данный документ M.

По сравнению с алгоритмом цифровой подписи Эль Гамаля алгоритм DSA имеет следующие основные преимущества:

1. При любом допустимом уровне стойкости, т.е. при любой паре чисел G и P (от 512 до 1024 бит), числа q, X, r, s имеют длину по 160 бит, сокращая длину подписи до 320 бит.

2. Большинство операций с числами K, r, s, X при вычислении подписи производится по модулю числа q длиной 160 бит, что сокращает время вычисления подписи.

3. При проверке подписи большинство операций с числами u_1, u_2, v, w также производится по модулю числа q длиной 160 бит, что сокращает объем памяти и время вычисления.

Недостатком алгоритма DSA является то, что при подписывании и при проверке подписи приходится выполнять сложные операции деления по модулю q:

$$s = \frac{m + rX}{K} \pmod q, \quad w = \frac{1}{s} \pmod q,$$

что не позволяет получать максимальное быстродействие.

Следует отметить, что реальное исполнение алгоритма DSA может быть ускорено с помощью выполнения предварительных вычислений. Заметим, что значение r не зависит от сообщения M и его хэш-значения m. Можно заранее создать строку случайных значений K и затем для каждого из этих значений вычислить значения r. Можно также заранее вычислить обратные значения K^{-1} для каждого из значений K. Затем, при поступлении сообщения M, можно вычислить значение s для данных значений r и K^{-1}. Эти предварительные вычисления значительно ускоряют работу алгоритма DSA.

Отечественный стандарт цифровой подписи

Отечественный стандарт цифровой подписи обозначается как ГОСТ Р 34.10-94 [40]. Алгоритм цифровой подписи, определяемый этим стандартом, концептуально близок к алгоритму DSA. В нем используются следующие параметры:

p – большое простое число длиной от 509 до 512 бит либо от 1020 до 1024 бит;

q – простой сомножитель числа (p – 1), имеющий длину 254...256 бит.

a – любое число, меньшее (p – 1), причем такое, что $a^q \bmod p = 1$;

x – некоторое число, меньшее q;

$y = a^x \bmod p$.

Кроме того, этот алгоритм использует однонаправленную хэш-функцию H(x). Стандарт ГОСТ Р 34.11-94 определяет хэш-функцию, основанную на использовании стандартного симметричного алгоритма ГОСТ 28147-89.

Первые три параметра p, q и a являются открытыми и могут быть общими для всех пользователей сети. Число x является секретным ключом. Число y является открытым ключом.

Чтобы подписать некоторое сообщение m, а затем проверить подпись, выполняются следующие шаги.

1. Пользователь A генерирует случайное число k, причем $k<q$.

2. Пользователь A вычисляет значения

$$r = (a^k \bmod p) \bmod q,$$

$$s = (x * r + k\,(H(m))) \bmod q.$$

Если $H(m) \bmod q = 0$, то значение $H(m) \bmod q$ принимают равным единице. Если $r = 0$, то выбирают другое значение k и начинают снова.

Цифровая подпись представляет собой два числа:

$$r \bmod 2^{256} \quad \text{и} \quad s \bmod 2^{256}.$$

Пользователь A отправляет эти числа пользователю B.

3. Пользователь B проверяет полученную подпись, вычисляя

$$v = H(m)^{q-2} \bmod q,$$

$$z_1 = (s * v) \bmod q,$$

$$z_2 = ((q - r) * v) \bmod q,$$

$$u = ((a^{z_1} * y^{z_2}) \bmod p) \bmod q.$$

Если $u = r$, то подпись считается верной.

Различие между этим алгоритмом и алгоритмом DSA заключается в том, что в DSA

$$s = (k^{-1}\,(x * r + (H(m)))) \bmod q,$$

что приводит к другому уравнению верификации.

Следует также отметить, что в отечественном стандарте ЭЦП параметр q имеет длину 256 бит. Западных криптографов вполне устраивает q длиной примерно 160 бит. Различие в значениях параметра q является отражением стремления разработчиков отечественного стандарта к получению более безопасной подписи.

Этот стандарт вступил в действие с начала 1995 г.

6.4. Цифровые подписи с дополнительными функциональными свойствами

Рассматриваемые в этом разделе цифровые подписи обладают дополнительными функциональными возможностями, помимо обычных свойств аутентификации сообщения и невозможности

отказа подписавшего лица от обязательств, связанных с подписанным текстом. В большинстве случаев они объединяют базовую схему цифровой подписи, например на основе алгоритма RSA, со специальным протоколом, обеспечивающим достижение тех дополнительных свойств, которыми базовая схема цифровой подписи не обладает [117, 121].

К схемам цифровой подписи с дополнительными функциональными свойствами относятся:

- схемы слепой (blind) подписи,
- схемы неоспоримой (undeniable) подписи.

Схемы слепой подписи

В отличие от обычных схем цифровой подписи, описанных в § 6.3, *схемы слепой подписи* (иногда называемые схемами подписи вслепую) являются двусторонними протоколами между отправителем A и стороной B, подписывающей документ.

Основная идея этих схем заключается в следующем. Отправитель A посылает порцию информации стороне B, которую B подписывает и возвращает A. Используя полученную подпись, сторона A может вычислить подпись стороны B на более важном для себя сообщении m. По завершении этого протокола сторона B ничего не знает ни о сообщении m, ни о подписи под этим сообщением [108].

Цель слепой подписи состоит в том, чтобы воспрепятствовать подписывающему лицу B ознакомиться с сообщением стороны A, которое он подписывает, и с соответствующей подписью под этим сообщением. Поэтому в дальнейшем подписанное сообщение невозможно связать со стороной A.

Приведем пример применения слепой подписи. Схема слепой подписи может найти применение в тех случаях, когда отправитель A (клиент банка) не хочет, чтобы подписывающая сторона B (банк) имела возможность в дальнейшем связать сообщение m и подпись $s_B(m)$ с определенным шагом выполненного ранее протокола.

В частности, это может быть важно при организации анонимных безналичных расчетов, когда сообщение m могло бы представлять денежную сумму, которую A хочет потратить. Когда сообщение m с подписью $s_B(m)$ предъявляется банку B для оплаты, банк B не может проследить, кто именно из клиентов предъявляет подписанный документ. Это позволяет пользователю A остаться анонимным. Принципы организации системы анонимных безналичных расчетов с использованием так называемой "электронной наличности" ("цифровых денег") на базе протоколов слепой подписи рассмотрены в [49, 67].

Для построения протокола слепой подписи необходимы следующие компоненты [108, 117]:

1. Механизм обычной цифровой подписи для подписывающей стороны B. Пусть $s_B(X)$ обозначает подпись стороны B на документе X.

2. Функции $f(\cdot)$ и $g(\cdot)$ (известные только отправителю) такие, что

$$g(s_B(f(m))) = s_m(m),$$

где $f(\cdot)$ – маскирующая (blinding) функция; $g(\cdot)$ – демаскирующая (unblinding) функция; $f(m)$ – замаскированное (blinded) сообщение m.

При выборе s_B, f и g существует ряд ограничений.

Выберем в качестве алгоритма подписи s_B для стороны B схему цифровой подписи RSA (см. § 6.3) с открытым ключом (N, E) и секретным ключом D, причём $N = P * Q$ – произведение двух больших случайных простых чисел.

Пусть k – некоторое фиксированное целое число, взаимно простое с N, т. е. НОД $(N, k) = 1$.

Маскирующая функция $f: Z_n \to Z_n$ определяется как $f(m) = m * k^E \bmod N$, а демаскирующая функция $g: Z_n \to Z_n$ – как $g(m) = k^{-1} m \bmod N$. При таком выборе f, g и s получаем

$$g(s_B(f(m))) = g(s_B(mk^E \bmod N)) = g(m^D k \bmod N) = m^D \bmod N = s_B(m),$$

что соответствует требованию 2.

Согласно протоколу слепой подписи, который предложил Д. Чом [121], отправитель A сначала получает подпись стороны B на замаскированном сообщении m^*. Используя эту подпись, сторона A вычисляет подпись B на заранее выбранном сообщении m, где $0 \le m \le N-1$. При этом стороне B ничего неизвестно ни о значении m, ни о подписи, связанной с m.

Пусть сторона B имеет для подписи по схеме RSA открытый ключ (N, E) и секретный ключ D. Пусть k – случайное секретное целое число, выбранное стороной A и удовлетворяющее условиям $0 \le k \le N-1$ и НОД (N, k).

Протокол слепой подписи Д. Чома включает следующие шаги:

1. Отправитель A вычисляет замаскированное сообщение $m^* = mk^E \bmod N$ и посылает его стороне B.

2. Подписывающая сторона B вычисляет подпись $s^* = (m^*)^D \bmod N$ и отправляет эту подпись стороне A.

3. Сторона A вычисляет подпись $s = k^{-1} s^* \bmod N$, которая является подписью B на сообщении m.

Нетрудно видеть, что

$$(m^*)^D \equiv (mk^E)^D \equiv m^D k \pmod N,$$

поэтому

$$k^{-1} s^* \equiv m^D k\, k^{-1} \equiv m^D \pmod N.$$

Д. Чом разработал несколько алгоритмов слепой подписи для создания системы анонимных безналичных электронных расчетов eCash [49, 108].

Схемы неоспоримой подписи

Неоспоримая подпись, как и обычная цифровая подпись, зависит от подписанного документа и секретного ключа. Однако в отличие от обычных цифровых подписей неоспоримая подпись не может быть верифицирована без участия лица, поставившего эту подпись. Возможно, более подходящим названием для этих подписей было бы "подписи, не допускающие подлога".

Рассмотрим два возможных сценария применения неоспоримой подписи [107, 117].

Сценарий 1. Сторона А (клиент) хочет получить доступ в защищенную зону, контролируемую стороной В (банком). Этой защищенной зоной может быть, например, депозитарий (хранилище ценностей клиентов). Сторона В требует от А поставить до предоставления клиенту доступа на заявке о допуске в защищенную зону подпись, время и дату. Если А применит неоспоримую подпись, тогда сторона В не сможет впоследствии доказать кому-либо, что А получил допуск без непосредственного участия А в процессе верификации подписи.

Сценарий 2. Предположим, что известная корпорация А разработала пакет программного обеспечения. Чтобы гарантировать подлинность пакета и отсутствие в нем вирусов, сторона А подписывает этот пакет неоспоримой подписью и продает его стороне В. Сторона В решает сделать копии этого пакета программного обеспечения и перепродать его третьей стороне С. При использовании стороной А неоспоримой подписи сторона С не сможет убедиться в подлинности этого пакета программного обеспечения и отсутствии в нем вирусов без участия стороны А.

Конечно, этот сценарий не препятствует стороне В поставить на пакете свою подпись, но тогда для стороны В будут утрачены все маркетинговые преимущества, связанные с использованием торговой марки корпорации А. Кроме того, будет легче раскрыть мошенническую деятельность стороны В.

Рассмотрим алгоритм неоспоримой цифровой подписи, разработанный Д. Чомом [107]. Сначала опишем алгоритм генерации ключей, с помощью которого каждая сторона А, подписывающая документ, выбирает секретный ключ и соответствующий открытый ключ.

Каждая сторона А должна выполнить следующее:

1. Выбрать случайное простое число $p=2q+1$, где q – также простое число.

2. Выбрать генераторное число α для подгруппы порядка q в циклической группе Z_p^*:

2.1. Выбрать случайный элемент $\beta \in Z_p$ и вычислить $\alpha = \beta^{(p-1)/q} \bmod p$.

2.2. Если $\alpha = 1$, тогда возвратиться к шагу 2.1.

3. Выбрать случайное целое $x \in \{1, 2, ..., q-1\}$ и вычислить $y = \alpha^x \bmod p$.

4. Для стороны A открытый ключ равен (p, α, y), секретный ключ равен x.

Согласно алгоритму неоспоримой подписи Д. Чома, сторона A подписывает сообщение m, принадлежащее подгруппе порядка q в Z_p^*. Любая сторона B может проверить эту подпись при участии A.

В работе алгоритма неоспоримой подписи можно выделить два этапа:

- генерация подписи;
- верификация подписи.

На этапе генерации подписи сторона A вычисляет $s = m^x \bmod p$, где s – подпись стороны A на сообщении m. Сообщение m с подписью s отсылается стороне B.

Этап верификации подписи выполняется стороной B с участием стороны A и включает следующие шаги:

1. B получает подлинный открытый ключ (p, α, y) стороны A.

2. B выбирает два случайных секретных целых числа a, $b \in \{1, 2, ..., q-1\}$.

3. B вычисляет $z = s^a y^b \bmod p$ и отправляет значение z стороне A.

4. A вычисляет $w = (z)^{1/x} \bmod p$, где $xx^{-1} \equiv 1 (\bmod q)$, и отправляет значение w стороне B.

5. B вычисляет $w' = m^a \alpha^b \bmod p$ и признает подпись s подлинной, если и только если $w = w'$.

Убедимся, что проверка подписи s работает:

$$w \equiv (z)^{1/x} \equiv (s^a y^b)^{1/x} \equiv (m^{xa} \alpha^{xb})^{1/x} \equiv m^a \alpha^b \equiv w' \bmod p.$$

Можно показать, что с высокой степенью вероятности злоумышленник не сможет заставить B принять фальшивую подпись. Предположим, что s представляет собой подделку подписи стороны A на сообщении m, т.е. $s \neq m^x \bmod p$. Тогда вероятность принятия стороной этой подписи в данном алгоритме составляет только 1/q, причем эта вероятность не зависит от вычислительных ресурсов злоумышленника.

Подписавшая сторона A при некоторых обстоятельствах могла бы попытаться отказаться от своей подлинной подписи одним из трех способов:

(а) отказаться от участия в протоколе верификации;

(б) некорректно выполнить протокол верификации;

(в) объявить подпись фальшивой, даже если протокол верификации оказался успешным.

Отречение от подписи способом (а) рассматривалось бы как очевидная попытка неправомерного отказа. Против способов (б) и (в) бороться труднее, здесь требуется специальный протокол дезавуирования. Этот протокол определяет, пытается ли подписавшая сторона А дезавуировать правильную подпись s или эта подпись является фальшивой. В этом протоколе по существу дважды применяется протокол верификации и затем производится проверка с целью убедиться, что сторона А выполняет этот протокол корректно [107, 117].

Протокол дезавуирования для схемы неоспоримой подписи Д. Чома включает следующие шаги:

1. В принимает от стороны А сообщение m с подписью s и получает подлинный открытый ключ (p, α, y) стороны А.

2. В выбирает случайные секретные целые числа $a, b \in \{1, 2, ..., q-1\}$, вычисляет $z = s^a y^b \bmod p$ и отправляет значение z стороне А.

3. А вычисляет $w = (z)^{1/x} \bmod p$, где $xx^{-1} \equiv 1 (\bmod q)$, и отправляет значение w стороне В.

4. Если $w = m^a \alpha^b \bmod p$, тогда В признает подпись s подлинной и выполнение протокола прекращается.

5. В выбирает случайные секретные целые числа $a', b' \in \{1, 2, ..., q-1\}$, вычисляет $z' = s^{a'} y^{b'} \bmod p$ и отправляет значение z' стороне А.

6. А вычисляет $w' = (z')^{1/x} \bmod p$ и отправляет значение w' стороне В.

7. Если $w' = m^{a'} \alpha^{b'} \bmod p$, тогда В принимает подпись s и выполнение протокола останавливается.

8. В вычисляет $c = (w\alpha^{-b})^{a'} \bmod p$, $c' = (w'\alpha^{-b'})^a \bmod p$. Если $c = c'$, тогда В заключает, что подпись s фальшивая; в противном случае В делает вывод, что подпись s подлинная, а сторона А пытается дезавуировать подпись s.

Нетрудно убедиться в том, что этот протокол достигает поставленной цели. Пусть m – сообщение и предположим, что s – подпись стороны А под сообщением m. Если подпись s фальшивая, т.е. $s \neq m^x \bmod p$, и если стороны А и В следуют протоколу должным образом, тогда $w = w'$ (и поэтому справедливо заключение В, что подпись s фальшивая). Пусть s на самом деле является подписью стороны А под сообщением m, т.е. $s = m^x \bmod p$. Предположим, что В точно следует протоколу, а А не следует. Тогда вероятность того, что $w = w'$ (и А преуспевает в дезавуировании подписи), составляет только $1/q$.

Следует отметить, что третья сторона C никогда не должна принимать в качестве доказательства подлинности подписи s запись стороной B протокола верификации, поскольку сторона B может выдумать успешную запись шага 2 и последующих шагов протокола верификации без участия подписывающей стороны A.

Неоспоримая подпись может быть верифицирована только путем непосредственного взаимодействия с подписывающей стороной A.

Разработан также алгоритм для обратимой неоспоримой подписи [105], которая может быть верифицирована, дезавуирована, а также преобразована в обычную цифровую подпись. Этот алгоритм основан на использовании алгоритма цифровой подписи Эль Гамаля.

ГЛАВА 7. УПРАВЛЕНИЕ КРИПТОГРАФИЧЕСКИМИ КЛЮЧАМИ

Любая криптографическая система основана на использовании криптографических ключей. В симметричной криптосистеме отправитель и получатель сообщения используют один и тот же секретный ключ. Этот ключ должен быть неизвестен всем остальным и должен периодически обновляться одновременно у отправителя и получателя. Процесс распределения (рассылки) секретных ключей между участниками информационного обмена в симметричных криптосистемах имеет весьма сложный характер.

Асимметричная криптосистема предполагает использование двух ключей – открытого и личного (секретного). Открытый ключ можно разглашать, а личный надо хранить в тайне. При обмене сообщениями необходимо пересылать только открытый ключ. Важным требованием является обеспечение подлинности отправителя сообщения. Это достигается путем взаимной аутентификации участников информационного обмена.

Под *ключевой информацией* понимают совокупность всех действующих в АСОИ ключей. Если не обеспечено достаточно надежное управление ключевой информацией, то, завладев ею, злоумышленник получает неограниченный доступ ко всей информации.

Управление ключами – информационный процесс, включающий реализацию следующих основных функций:

- генерация ключей;
- хранение ключей;
- распределение ключей.

7.1. Генерация ключей

Безопасность любого криптографического алгоритма определяется используемым криптографическим ключом. Добротные криптографические ключи должны иметь достаточную длину и случайные значения битов. В табл. 4.3 приведены длины ключей симметричной и асимметричной криптосистем, обеспечивающие одинаковую стойкость к атаке полного перебора (атаке "грубой силы") [123].

Для получения ключей используются аппаратные и программные средства генерации случайных значений ключей. Как правило, применяют датчики псевдослучайных чисел (ПСЧ). Однако степень случайности генерации чисел должна быть достаточно высокой. Идеальными генераторами являются устройства на основе "натуральных" случайных процессов, например на основе *белого радиошума*.

В АСОИ со средними требованиями защищенности вполне приемлемы программные генераторы ключей, которые вычисляют ПСЧ как сложную функцию от текущего времени и (или) числа, введенного пользователем.

Один из методов генерации сеансового ключа для симметричных криптосистем описан в стандарте ANSI X 9.17. Он предполагает использование криптографического алгоритма DES (хотя можно применить и другие симметричные алгоритмы шифрования).

Обозначения:

$E_K(X)$ – результат шифрования алгоритмом DES значения X;

K – ключ, зарезервированный для генерации секретных ключей;

V_0 – секретное 64-битовое начальное число;

T – временная отметка.

Схема генерации случайного сеансового ключа R_i в соответствии со стандартом ANSI X 9.17 показана на рис.7.1. Случайный ключ R_i генерируют, вычисляя значение

$$R_i = E_K(E_K(T_i) \oplus V_i).$$

Следующее значение V_{i+1} вычисляют так:

$$V_{i+1} = E_K(E_K(T_i) \oplus R_i).$$

Если необходим 128-битовый случайный ключ, генерируют пару ключей R_i, R_{i+1} и объединяют их вместе.

Если ключ не меняется регулярно, это может привести к его раскрытию и утечке информации. Регулярную замену ключа можно осуществить, используя процедуру модификации ключа.

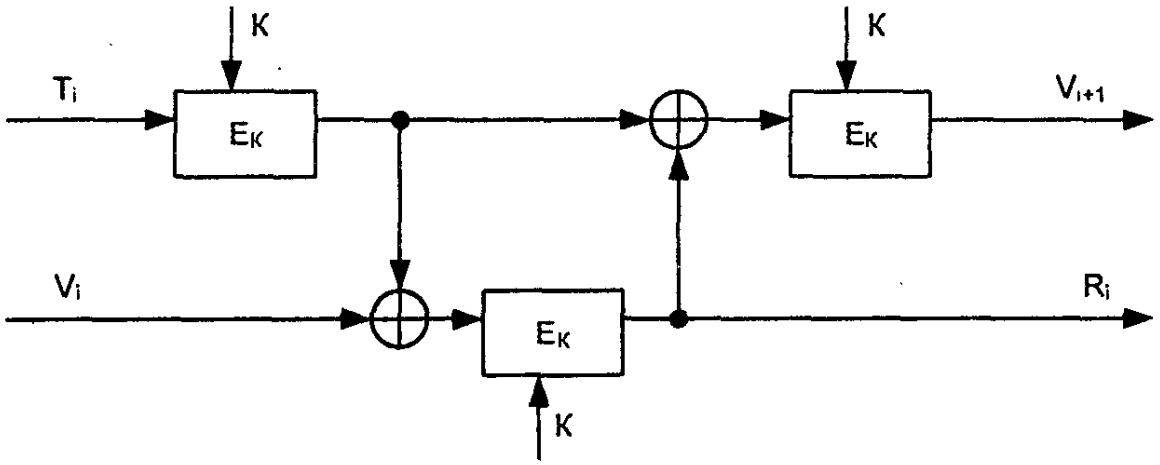


Рис.7.1. Схема генерации случайного ключа R_i в соответствии со стандартом ANSI X 9.17

Модификация ключа – это генерирование нового ключа из предыдущего значения ключа с помощью односторонней (однонаправленной) функции. Участники информационного обмена разделяют один и тот же ключ и одновременно вводят его значение в качестве аргумента в одностороннюю функцию, получая один и тот же результат. Затем они берут определенные биты из этих результатов, чтобы создать новое значение ключа.

Процедура модификации ключа работоспособна, но надо помнить, что новый ключ безопасен в той же мере, в какой был безопасен прежний ключ. Если злоумышленник сможет добыть прежний ключ, то он сможет выполнить процедуру модификации ключа.

Генерация ключей для асимметричных криптосистем с открытыми ключами много сложнее, потому что эти ключи должны обладать определенными математическими свойствами (они должны быть очень большими и простыми и т.д.).

7.2. Хранение ключей

Под *функцией хранения ключей* понимают организацию их безопасного хранения, учета и удаления. Ключ является самым привлекательным для злоумышленника объектом, открывающим ему путь к конфиденциальной информации. Поэтому вопросам безопасного хранения ключей следует уделять особое внимание. Секретные ключи никогда не должны записываться в явном виде на носителе, который может быть считан или скопирован.

Носители ключевой информации

Ключевой носитель может быть технически реализован различным образом на разных носителях информации – магнитных дисках, устройствах хранения ключей типа Touch Memory, пластиковых картах и т.д.

Магнитные диски представляют собой распространенный тип носителя ключевой информации. Применение магнитного диска (МД) в качестве носителя ключа позволяет реализовать необходимое свойство отчуждаемости носителя ключа от защищенной компьютерной системы, т.е. осуществить временное изъятие МД из состава технических средств компьютерной системы. Особенно целесообразно использование в качестве ключевых носителей съемных накопителей – гибких магнитных дисков, съемных магнитооптических носителей и т.д. [73].

Основное преимущество МД по сравнению с другими носителями ключевой информации заключается в том, что оборудование для взаимодействия с МД (дисковод) входит в состав штатных

средств компьютера. Другая важная особенность, определяющая широкое распространение МД,– стандартный формат хранения информации на дисках и стандартные программные средства доступа к дискам. Кроме того, из всех средств хранения ключевой информации гибкие магнитные диски имеют самую низкую стоимость.

Для обеспечения надежного хранения ключевой информации на МД применяют как минимум двукратное резервирование объектов хранения. Это позволяет защитить ключевую информацию от ошибок при считывании с МД и от сбоев программной и аппаратной части.

Для предотвращения возможности перехвата ключевой информации в процессе ее чтения с МД используют хранение ключевой информации на МД в зашифрованном виде.

Устройство хранения ключей типа Touch Memory является относительно новым носителем ключевой информации, предложенным американской компанией Dallas Semiconductor. Носитель информации Touch Memory (TM) представляет собой энергонезависимую память, размещенную в металлическом корпусе, с одним сигнальным контактом и одним контактом земли. Корпус TM имеет диаметр 16,25 мм и толщину 3,1 или 5,89 мм (в зависимости от модификации прибора).

В структуру TM входят следующие основные блоки [73]:

- Постоянное запоминающее устройство (ПЗУ) хранит 64-разрядный код, состоящий из байтового кода типа прибора, 48-битового уникального серийного номера и 8-битовой контрольной суммы. Содержимое ПЗУ уникально и не может быть изменено в течение всего срока службы прибора.
- Оперативное запоминающее устройство (ОЗУ) емкостью от 128 до 8192 байт содержат практически все модификации TM. В одной из модификаций оперативная память аппаратно защищена от несанкционированного доступа.
- Встроенная миниатюрная литиевая батарейка со сроком службы не менее 10 лет обеспечивает питанием все блоки устройства.

Особенностью технологии хранения и обмена ключевой информации между носителем TM и внешними устройствами является сравнительно низкая скорость (обусловленная последовательной передачей данных) и высокая вероятность сбоя в тракте чтения-записи, обусловленная тем, что контакт устройства TM с устройством чтения осуществляется пользователем вручную без дополнительной фиксации (простое касание, что и определило название прибора TM). В связи с этим особое значение приобретают вопросы надежного обмена между программами обработки ключевой информации пользователей и носителем TM.

В устройстве ТМ конструктивно отработаны вопросы надежности функционирования и вопросы интерфейса со считывающим устройством на основе одного сигнального контакта. Для обеспечения достоверного чтения применяются корректирующие коды, для обеспечения достоверной записи в приборе предусмотрена технология буферизации. При проведении операции записи первоначально вектор передаваемой в ТМ информации помещается в буфер, далее выполняется операция чтения из буфера, затем прочтенная из буфера информация сравнивается с записываемой и в случае совпадения подается сигнал переноса информации из буфера в память долговременного хранения.

Таким образом, носитель ТМ является микроконтроллерным устройством без собственной вычислительной мощности и с ограниченным объемом хранения, но с достаточно высокими надежностными характеристиками. Поэтому применение ТМ вполне обосновано в случае повышенных требований к надежности носителя ключа и небольшого объема ключевой информации, хранимой в ТМ.

Электронные пластиковые карты становятся в настоящее время наиболее распространенным и универсальным носителем конфиденциальной информации, который позволяет идентифицировать и аутентифицировать пользователей, хранить криптографические ключи, пароли и коды.

Интеллектуальные карты (смарт-карты), обладающие наибольшими возможностями, не только эффективно применяются для хранения ключевой информации, но и широко используются в электронных платежных системах, в комплексных решениях для медицины, транспорта, связи, образования и т. п. Более подробные сведения об электронных пластиковых картах приводятся в § 9.4.

Концепция иерархии ключей

Любая информация об используемых ключах должна быть защищена, в частности храниться в зашифрованном виде.

Необходимость в хранении и передаче ключей, зашифрованных с помощью других ключей, приводит к концепции *иерархии ключей*. В стандарте ISO 8532 (Banking-Key Management) подробно изложен метод главных/сеансовых ключей (master/session keys). Суть метода состоит в том, что вводится иерархия ключей: главный ключ (ГК), ключ шифрования ключей (КК), ключ шифрования данных (КД).

Иерархия ключей может быть:

- двухуровневой (КК/КД);
- трехуровневой (ГК/КК/КД).

Самым нижним уровнем являются *рабочие или сеансовые КД*, которые применяются для шифрования данных, персональных идентификационных номеров (PIN) и аутентификации сообщений.

Когда эти ключи надо зашифровать с целью защиты при передаче или хранении, используют ключи следующего уровня – *ключи шифрования ключей*. Ключи шифрования ключей никогда не должны использоваться как сеансовые (рабочие) КД, и наоборот.

Такое разделение функций необходимо для обеспечения максимальной безопасности. Фактически стандарт устанавливает, что различные типы рабочих ключей (например, для шифрования данных, для аутентификации и т.д.) должны всегда шифроваться с помощью различных версий ключей шифрования ключей. В частности, ключи шифрования ключей, используемые для пересылки ключей между двумя узлами сети, известны также как *ключи обмена между узлами сети* (cross domain keys). Обычно в канале используются два ключа для обмена между узлами сети, по одному в каждом направлении. Поэтому каждый узел сети будет иметь *ключ отправления* для обмена с узлами сети и *ключ получения* для каждого канала, поддерживаемого другим узлом сети.

На верхнем уровне иерархии ключей располагается *главный ключ, мастер-ключ*. Этот ключ применяют для шифрования КК, когда требуется сохранить их на диске. Обычно в каждом компьютере используется только один мастер-ключ.

Мастер-ключ распространяется между участниками обмена неэлектронным способом – при личном контакте, чтобы исключить его перехват и/или компрометацию. Раскрытие противником значения мастер-ключа полностью уничтожает защиту компьютера.

Значение мастер-ключа фиксируется на длительное время (до нескольких недель или месяцев). Поэтому генерация и хранение мастер-ключей являются критическими вопросами криптографической защиты. На практике мастер-ключ компьютера создается истинно случайным выбором из всех возможных значений ключей. Мастер-ключ помещают в защищенный от считывания и записи и от механических воздействий блок криптографической системы таким образом, чтобы раскрыть значение этого ключа было невозможно. Однако все же должен существовать способ проверки, является ли значение ключа правильным.

Проблема аутентификации мастер-ключа может быть решена различными путями. Один из способов аутентификации показан на рис.7.2 [123].

Администратор, получив новое значение мастер-ключа $K_н$ хост-компьютера, шифрует некоторое сообщение М ключом $K_н$. Пара (криптограмма $E_{K_н}(M)$, сообщение М) помещается в память компьютера. Всякий раз, когда требуется аутентификация мастер-ключа хост-компьютера, берется сообщение М из памяти и подается в криптографическую систему. Получаемая криптограмма сравнивается с криптограммой, хранящейся в памяти. Если они совпадают, считается, что данный ключ является правильным.

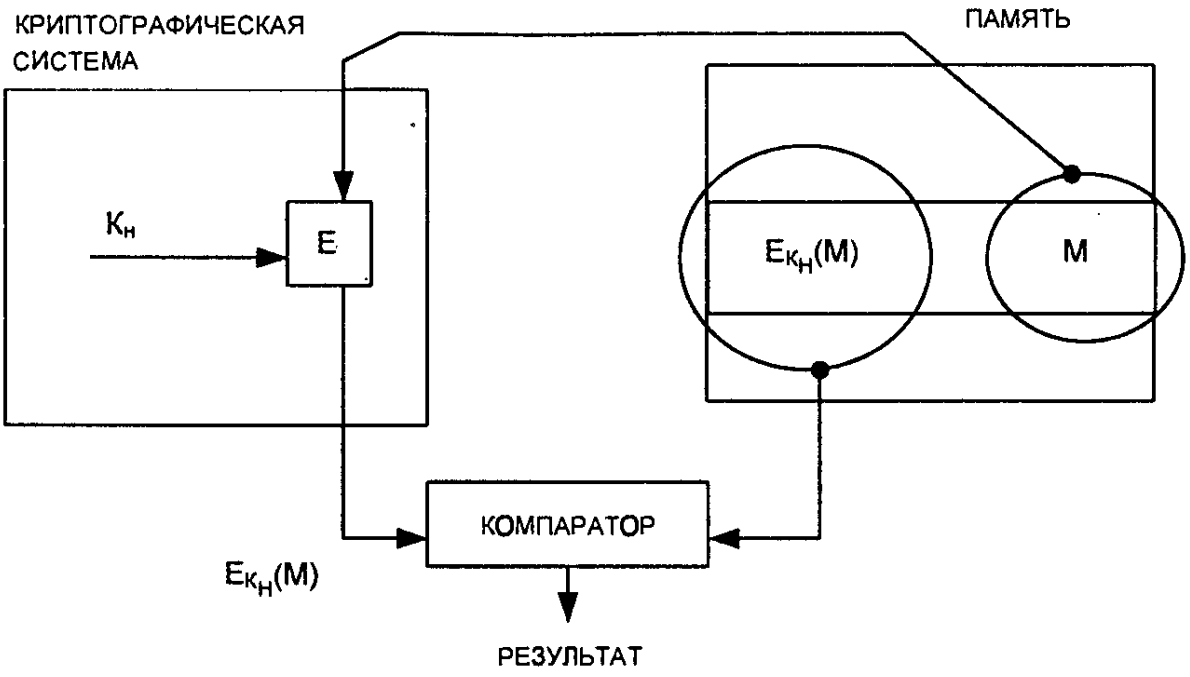


Рис.7.2. Схема аутентификации мастер-ключа хост-компьютера

Рабочие ключи (например, сеансовый) обычно создаются с помощью псевдослучайного генератора и могут храниться в незащищенном месте. Это возможно, поскольку такие ключи генерируются в форме соответствующих криптограмм, т.е. генератор ПСЧ выдает вместо ключа K_S его криптограмму $E_{K_н}(K_S)$, получаемую с помощью мастер-ключа хост-компьютера. Расшифровывание такой криптограммы выполняется только перед использованием ключа K_S.

Схема защиты рабочего (сеансового) ключа показана на рис.7.3. Чтобы зашифровать сообщение M ключом K_S, на соответствующие входы криптографической системы подается крипто-

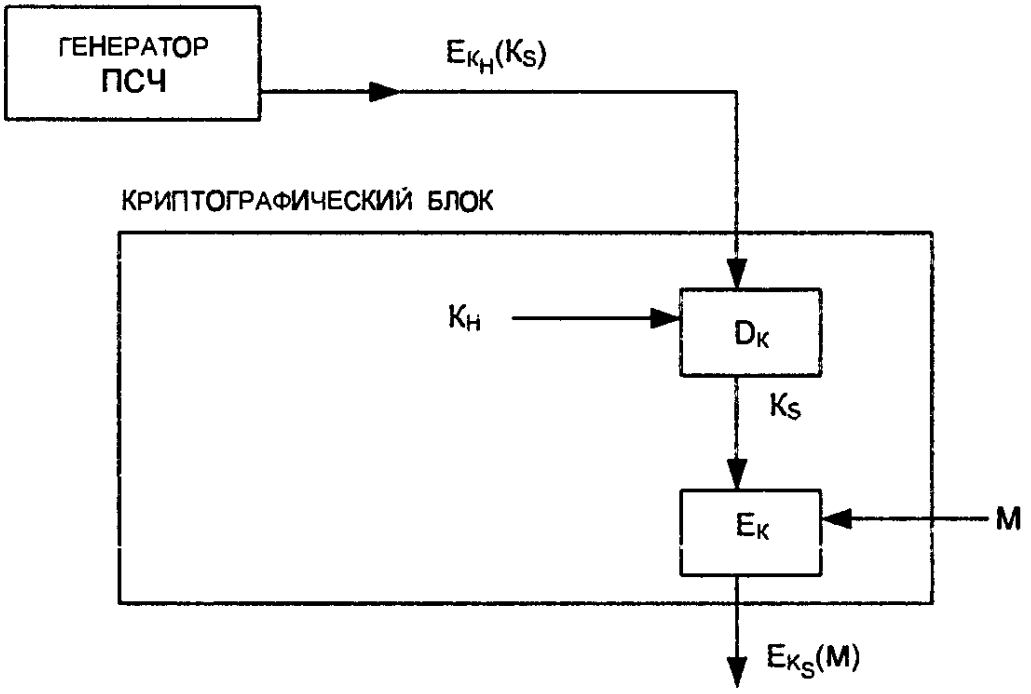


Рис.7.3. Схема защиты ключа K_S

грамма $E_{K_H}(K_S)$ и сообщение М. Криптографическая система сначала восстанавливает ключ K_S, а затем шифрует сообщение М, используя открытую форму сеансового ключа K_S.

Таким образом, безопасность сеансовых ключей зависит от безопасности криптографической системы. Криптографический блок может быть спроектирован как единая СБИС и помещен в физически защищенное место.

Очень важным условием безопасности информации является периодическое обновление ключевой информации в АСОИ. При этом должны переназначаться как рабочие ключи, так и мастер-ключи. В особо ответственных АСОИ обновление ключевой информации (сеансовых ключей) желательно делать ежедневно. Вопрос обновления ключевой информации тесно связан с третьим элементом управления ключами – распределением ключей.

7.3. Распределение ключей

Распределение ключей – самый ответственный процесс в управлении ключами. К нему предъявляются следующие требования:

- оперативность и точность распределения;
- скрытность распределяемых ключей.

Распределение ключей между пользователями компьютерной сети реализуется двумя способами [55]:

1) использованием одного или нескольких центров распределения ключей;

2) прямым обменом сеансовыми ключами между пользователями сети.

Недостаток первого подхода состоит в том, что центру распределения ключей известно, кому и какие ключи распределены, и это позволяет читать все сообщения, передаваемые по сети. Возможные злоупотребления существенно влияют на защиту. При втором подходе проблема состоит в том, чтобы надежно удостоверить подлинность субъектов сети.

В обоих случаях должна быть обеспечена подлинность сеанса связи. Это можно осуществить, используя механизм запроса-ответа или механизм отметки времени.

Механизм запроса-ответа заключается в следующем. Пользователь А включает в посылаемое сообщение (запрос) для пользователя В непредсказуемый элемент (например, случайное число). При ответе пользователь В должен выполнить некоторую операцию с этим элементом (например, добавить единицу), что невозможно осуществить заранее, поскольку неизвестно, какое случайное число придет в запросе. После получения результата действий пользователя В (ответ) пользователь А может быть уверен, что сеанс является подлинным.

Механизм отметки времени предполагает фиксацию времени для каждого сообщения. Это позволяет каждому субъекту сети определить, насколько старо пришедшее сообщение, и отвергнуть его, если появится сомнение в его подлинности. При использовании отметок времени необходимо установить допустимый временной интервал задержки.

В обоих случаях для защиты элемента контроля используют шифрование, чтобы быть уверенным, что ответ отправлен не злоумышленником и не изменен штемпель отметки времени.

Задача распределения ключей сводится к построению протокола распределения ключей, обеспечивающего:

- взаимное подтверждение подлинности участников сеанса;
- подтверждение достоверности сеанса механизмом запроса-ответа или отметки времени;
- использование минимального числа сообщений при обмене ключами;
- возможность исключения злоупотреблений со стороны центра распределения ключей (вплоть до отказа от него).

В основу решения задачи распределения ключей целесообразно положить принцип отделения процедуры подтверждения подлинности партнеров от процедуры собственно распределения ключей. Цель такого подхода состоит в создании метода, при котором после установления подлинности участники сами формируют сеансовый ключ без участия центра распределения ключей с тем, чтобы распределитель ключей не имел возможности выявить содержание сообщений.

Распределение ключей с участием центра распределения ключей

При распределении ключей между участниками предстоящего информационного обмена должна быть гарантирована подлинность сеанса связи. Для взаимной проверки подлинности партнеров приемлема *модель рукопожатия.* В этом случае ни один из участников не будет получать никакой секретной информации во время процедуры установления подлинности [55].

Взаимное установление подлинности гарантирует вызов нужного субъекта с высокой степенью уверенности, что связь установлена с требуемым адресатом и никаких попыток подмены не было. Реальная процедура организации соединения между участниками информационного обмена включает как этап распределения, так и этап подтверждения подлинности партнеров.

При включении в процесс распределения ключей центра распределения ключей (ЦРК) осуществляется его взаимодействие с одним или обоими участниками сеанса с целью распределения секретных или открытых ключей, предназначенных для использования в последующих сеансах связи [125].

Следующий этап – подтверждение подлинности участников – содержит обмен удостоверяющими сообщениями, чтобы иметь возможность выявить любую подмену или повтор одного из предыдущих вызовов.

Рассмотрим протоколы для симметричных криптосистем с секретными ключами и для асимметричных криптосистем с открытыми ключами. Вызывающий (исходный объект) обозначается через A, а вызываемый (объект назначения) – через B. Участники сеанса A и B имеют уникальные идентификаторы Id_A и Id_B соответственно.

Протокол аутентификации и распределения ключей для симметричных криптосистем. Рассмотрим в качестве примера протокол аутентификации и распределения ключей Kerberos (по-русски – Цербер). Первоначально протокол Kerberos был разработан в Массачусетском технологическом институте (США) для проекта Athena. Протокол Kerberos спроектирован для работы в сетях TCP/IP и предполагает участие в аутентификации и распределении ключей третьей доверенной стороны. Kerberos обеспечивает надежную аутентификацию в сети, разрешая законному пользователю доступ к различным машинам в сети. Протокол Kerberos основывается на симметричной криптографии (реализован алгоритм DES, хотя возможно применение и других симметричных криптоалгоритмов). Kerberos разделяет отдельный секретный ключ с каждым субъектом сети. Знание такого секретного ключа равносильно доказательству подлинности субъекта сети [117, 125].

Основной протокол Kerberos является вариантом протокола аутентификации и распределения ключей Нидхема–Шредера [117]. В основном протоколе Kerberos (версия 5) участвуют две взаимодействующие стороны A и B и доверенный сервер KS (Kerberos Server). Стороны A и B, каждая по отдельности, разделяют свой секретный ключ с сервером KS. Доверенный сервер KS выполняет роль центра распределения ключей ЦРК.

Пусть сторона A хочет получить сеансовый ключ для информационного обмена со стороной B.

Сторона A инициирует фазу распределения ключей, посылая по сети серверу KS идентификаторы Id_A и Id_B:

(1) $A \rightarrow KS$: Id_A, Id_B.

Сервер KS генерирует сообщение с временной отметкой T, сроком действия L, случайным сеансовым ключом K и идентификатором Id_A. Он шифрует это сообщение секретным ключом, который разделяет со стороной B.

Затем сервер KS берет временную отметку T, срок действия L, сеансовый ключ K, идентификатор Id_B стороны B и шифрует все это секретным ключом, который разделяет со стороной A. Оба эти зашифрованные сообщения он отправляет стороне A:

(2) $KS \rightarrow A$: $E_A(T, L, K, Id_B)$, $E_B(T, L, K, Id_A)$.

Сторона А расшифровывает первое сообщение своим секретным ключом, проверяет отметку времени Т, чтобы убедиться, что это сообщение не является повторением предыдущей процедуры распределения ключей.

Затем сторона А генерирует сообщение со своим идентификатором Id_A и отметкой времени Т, шифрует его сеансовым ключом К и отправляет стороне В. Кроме того, А отправляет для В сообщение от KS, зашифрованное ключом стороны В:

(3) $A \rightarrow B$: $E_K(Id_A, T)$, $E_B(T, L, K, Id_A)$.

Только сторона В может расшифровать сообщения (3). Сторона В получает отметку времени Т, срок действия L, сеансовый ключ К и идентификатор Id_A. Затем сторона В расшифровывает сеансовым ключом К вторую часть сообщения (3). Совпадение значений Т и Id_A в двух частях сообщения подтверждают подлинность А по отношению к В.

Для взаимного подтверждения подлинности сторона В создает сообщение, состоящее из отметки времени Т плюс 1, шифрует его ключом К и отправляет стороне А:

(4) $B \rightarrow A$: $E_K(T+1)$.

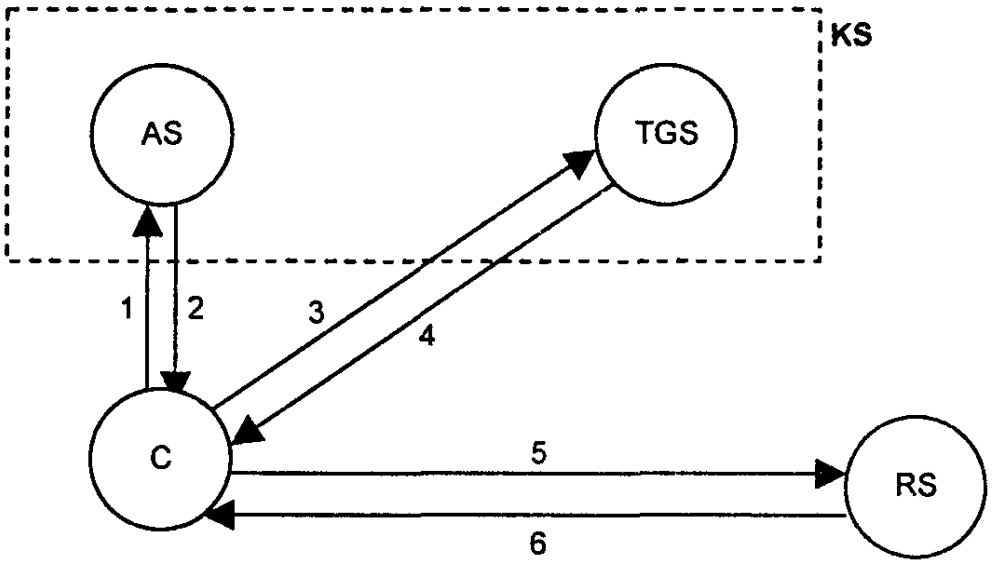
Если после расшифрования сообщения (4) сторона А получает ожидаемый результат, она знает, что на другом конце линии связи находится действительно В.

Этот протокол успешно работает при условии, что часы каждого участника синхронизированы с часами сервера KS. Следует отметить, что в этом протоколе необходим обмен с KS для получения сеансового ключа каждый раз, когда А желает установить связь с В. Протокол обеспечивает надежное соединение объектов А и В при условии, что ни один из ключей не скомпрометирован и сервер KS защищен.

Система Kerberos обеспечивает защиту сети от несанкционированного доступа, базируясь исключительно на программных решениях, и предполагает многократное шифрование передаваемой по сети управляющей информации.

Система Kerberos имеет структуру типа клиент-сервер и состоит из клиентских частей С, установленных на все машины сети (рабочие станции пользователей и серверы), и Kerberos-сервера KS, располагающегося на каком-либо (не обязательно выделенном) компьютере.

Kerberos-сервер, в свою очередь, можно разделить на две части: сервер идентификации AS (Authentication Server) и сервер выдачи разрешений TGS (Ticket Granting Server). Информационными ресурсами, необходимыми клиентам С, управляет сервер информационных ресурсов RS (рис.7.4).



Обозначения:

KS	– сервер системы Kerberos
AS	– сервер идентификации
TGS	– сервер выдачи разрешений
RS	– сервер информационных ресурсов
C	– клиент системы Kerberos
1 : C → AS:	– запрос разрешить обратиться к TGS
2 : AS → C:	– разрешение обратиться к TGS
3 : C → TGS:	– запрос на допуск к RS
4 : TGS → C:	– разрешение на допуск к RS
5 : C → RS:	– запрос на получение информационного ресурса от RS
6 : RS → C:	– подтверждение подлинности сервера RS и предоставление информационного ресурса

Рис.7.4. Схема и шаги протокола Kerberos

Область действия системы Kerberos распространяется на тот участок сети, все пользователи которого зарегистрированы под своими именами и паролями в базе данных Kerberos-сервера.

Укрупненно процесс идентификации и аутентификации пользователя в системе Kerberos можно описать следующим образом. Пользователь (клиент) С, желая получить доступ к ресурсу сети, направляет запрос серверу идентификации AS. Последний идентифицирует пользователя с помощью его имени и пароля и выдает разрешение на доступ к серверу выдачи разрешений TGS, который, в свою очередь, по запросу клиента С разрешает использование необходимых ресурсов сети с помощью целевого сервера информационных ресурсов RS.

Данная модель взаимодействия клиента с серверами может функционировать только при условии обеспечения конфиденциальности и целостности передаваемой управляющей информации. Без строгого обеспечения информационной безопасности клиент не может отправлять серверам AS, TGS и RS свои запросы и получать разрешения на доступ к обслуживанию в сети. Чтобы избе-

жать возможности перехвата и несанкционированного использования информации, Kerberos применяет при передаче любой управляющей информации в сети сложную систему многократного шифрования с использованием комплекса секретных ключей (секретный ключ клиента, секретный ключ сервера, секретные сеансовые ключи, клиент-сервер) [125].

Протокол для асимметричных криптосистем с использованием сертификатов открытых ключей. В этом протоколе используется идея сертификатов открытых ключей [55].

Сертификатом открытого ключа C называется сообщение ЦРК, удостоверяющее целостность некоторого открытого ключа объекта. Например, сертификат открытого ключа для пользователя A, обозначаемый C_A, содержит отметку времени T, идентификатор Id_A и открытый ключ K_A, зашифрованные секретным ключом ЦРК $k_{ЦРК}$, т.е.

$$C_A = E_{k_{ЦРК}}(T, Id_A, K_A).$$

Отметка времени T используется для подтверждения актуальности сертификата и тем самым предотвращает повторы прежних сертификатов, которые содержат открытые ключи и для которых соответствующие секретные ключи несостоятельны.

Секретный ключ $k_{ЦРК}$ известен только менеджеру ЦРК. Открытый ключ $K_{ЦРК}$ известен участникам A и B. ЦРК поддерживает таблицу открытых ключей всех объектов сети, которые он обслуживает.

Вызывающий объект A инициирует стадию установления ключа, запрашивая у ЦРК сертификат своего открытого ключа и открытого ключа участника B:

(1) A → ЦРК: Id_A, Id_B, "Вышлите сертификаты ключей A и B".

Здесь Id_A и Id_B – уникальные идентификаторы соответственно участников A и B.

Менеджер ЦРК отвечает сообщением

(2) ЦРК → A: $E_{k_{ЦРК}}(T, Id_A, K_A)$, $E_{k_{ЦРК}}(T, Id_B, K_B)$.

Участник A, используя открытый ключ ЦРК $K_{ЦРК}$, расшифровывает ответ ЦРК, проверяет оба сертификата. Идентификатор Id_B убеждает A, что личность вызываемого участника правильно зафиксирована в ЦРК и K_B – действительно открытый ключ участника B, поскольку оба зашифрованы ключом $k_{ЦРК}$.

Хотя открытые ключи предполагаются известными всем, посредничество ЦРК позволяет подтвердить их целостность. Без такого посредничества злоумышленник может снабдить A своим открытым ключом, который A будет считать ключом участника B. Затем злоумышленник может подменить собой B и установить связь с A, и его никто не сможет выявить.

Следующий шаг протокола включает установление связи A с B:

(3) $A \rightarrow B$: C_A, $E_{k_A}(T)$, $E_{K_B}(r_1)$.

Здесь C_A – сертификат открытого ключа пользователя A; $E_{k_A}(T)$ – отметка времени, зашифрованная секретным ключом участника A и являющаяся подписью участника A, поскольку никто другой не может создать такую подпись; r_1 – случайное число, генерируемое A и используемое для обмена с B в ходе процедуры подлинности.

Если сертификат C_A и подпись A верны, то участник B уверен, что сообщение пришло от A. Часть сообщения $E_{k_B}(r_1)$ может расшифровать только B, поскольку никто другой не знает секретного ключа k_B, соответствующего открытому ключу K_B. Участник B расшифровывает значение числа r_1 и, чтобы подтвердить свою подлинность, посылает участнику A сообщение

(4) $B \rightarrow A$: $E_{K_A}(r_1)$.

Участник A восстанавливает значение r_1, расшифровывая это сообщение с использованием своего секретного ключа k_A. Если это ожидаемое значение r_1, то A получает подтверждение, что вызываемый участник действительно B.

Протокол, основанный на симметричном шифровании, функционирует быстрее, чем протокол, основанный на криптосистемах с открытыми ключами. Однако способность систем с открытыми ключами генерировать цифровые подписи, обеспечивающие различные функции защиты, компенсирует избыточность требуемых вычислений.

Прямой обмен ключами между пользователями

При использовании для информационного обмена криптосистемы с симметричным секретным ключом два пользователя, желающие обменяться криптографически защищенной информацией, должны обладать общим секретным ключом. Пользователи должны обменяться общим ключом по каналу связи безопасным образом. Если пользователи меняют ключ достаточно часто, то доставка ключа превращается в серьезную проблему.

Для решения этой проблемы можно применить два способа:

1) использование криптосистемы с открытым ключом для шифрования и передачи секретного ключа симметричной криптосистемы;

2) использование системы открытого распределения ключей Диффи–Хеллмана.

Первый способ был подробно изложен в § 4.6. Второй способ основан на применении системы открытого распределения ключей. Эта система позволяет пользователям обмениваться ключами по незащищенным каналам связи. Интересно отметить, что система

открытого распределения ключей базируется на тех же принципах, что и система шифрования с открытыми ключами [24].

Алгоритм открытого распределения ключей Диффи–Хеллмана. Алгоритм Диффи–Хеллмана был первым алгоритмом с открытыми ключами (предложен в 1976 г.). Его безопасность обусловлена трудностью вычисления дискретных логарифмов в конечном поле, в отличие от легкости дискретного возведения в степень в том же конечном поле.

Предположим, что два пользователя A и B хотят организовать защищенный коммуникационный канал.

1. Обе стороны заранее уславливаются о модуле N (N должен быть простым числом) и примитивном элементе $g \in Z_N$, $(1 \le g \le N-1)$, который образует все ненулевые элементы множества Z_N, т.е.

$$\{g, g^2, \ldots, g^{N-1} = 1\} = Z_N - \{0\}.$$

Эти два целых числа N и g могут не храниться в секрете. Как правило, эти значения являются общими для всех пользователей системы.

2. Затем пользователи A и B независимо друг от друга выбирают собственные секретные ключи k_A и k_B (k_A и k_B – случайные большие целые числа, которые хранятся пользователями A и B в секрете).

3. Далее пользователь A вычисляет открытый ключ

$$y_A = g^{k_A} (\mathrm{mod}\ N),$$

а пользователь B – открытый ключ

$$y_B = g^{k_B} (\mathrm{mod}\ N).$$

4. Затем стороны A и B обмениваются вычисленными значениями открытых ключей y_A и y_B по незащищенному каналу. (Мы считаем, что все данные, передаваемые по незащищенному каналу связи, могут быть перехвачены злоумышленником.)

5. Далее пользователи A и B вычисляют общий секретный ключ, используя следующие сравнения:

пользователь A: $K = (y_B)^{k_A} = (g^{k_B})^{k_A} (\mathrm{mod}\ N)$;

пользователь B: $K' = (y_A)^{k_B} = (g^{k_A})^{k_B} (\mathrm{mod}\ N)$.

При этом $K = K'$, так как $(g^{k_B})^{k_A} = (g^{k_A})^{k_B} (\mathrm{mod}\ N)$.

Схема реализации алгоритма Диффи–Хеллмана показана на рис.7.5.

Ключ K может использоваться в качестве общего секретного ключа (ключа шифрования ключей) в симметричной криптосистеме.

Кроме того, обе стороны A и B могут шифровать сообщения, используя следующее преобразование шифрования (типа RSA): $C = E_K(M) = M^K (\mathrm{mod}\ N)$.

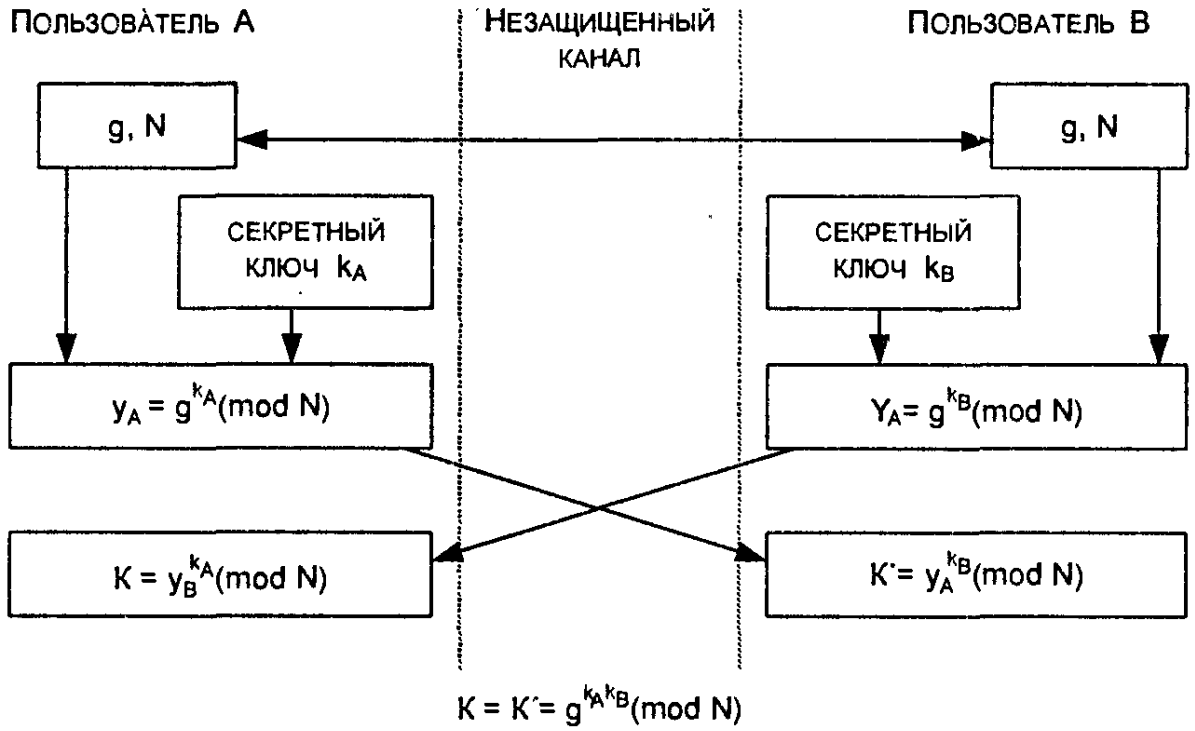


Рис.7.5. Схема реализации алгоритма Диффи–Хеллмана

Для выполнения расшифрования получатель сначала находит ключ расшифрования K^* с помощью сравнения

$$K * K^* \equiv 1 (\mathrm{mod}\ N - 1),$$

а затем восстанавливает сообщение

$$M = D_K (C) = C^{K^*} (\mathrm{mod}\ N).$$

Пример. Допустим, модуль $N=47$, а примитивный элемент $g=23$. Предположим, что пользователи А и В выбрали свои секретные ключи: $k_A=12 (\mathrm{mod}\ 47)$ и $k_B=33 (\mathrm{mod}\ 47)$.

Для того чтобы иметь общий секретный ключ K, они вычисляют сначала значения частных открытых ключей:

$$y_A = g^{k_A} = 23^{12} = 27 (\mathrm{mod}\ 47),$$

$$y_B = g^{k_B} = 23^{33} = 33 (\mathrm{mod}\ 47).$$

После того, как пользователи А и В обменяются своими значениями y_A и y_B, они вычисляют общий секретный ключ

$$K = (y_B)^{k_A} = (y_A)^{k_B} = 33^{12} = 27^{33} = 23^{12*33} = 25 (\mathrm{mod}\ 47).$$

Кроме того, они находят секретный ключ расшифрования, используя следующее сравнение:

$$K * K^* \equiv 1 (\mathrm{mod}\ N - 1),$$

откуда $K^*=35 (\mathrm{mod}\ 46)$.

Теперь, если сообщение $M=16$, то криптограмма

$$C = M^K = 16^{25} = 21 (\mathrm{mod}\ 47).$$

Получатель восстанавливает сообщение так:

$$M = C^{K^*} = 21^{35} = 16 (\mathrm{mod}\ 47).$$

Злоумышленник, перехватив значения N, g, y_A и y_B, тоже хотел бы определить значение ключа K. Очевидный путь для решения этой задачи состоит в вычислении такого значения k_A по N, g, y_A, что $g^{k_A} \bmod N = y_A$ (поскольку в этом случае, вычислив k_A, можно найти $K = (y_B)^{k_A} \bmod N$). Однако нахождение k_A по N, g и y_A – задача нахождения дискретного логарифма в конечном поле, которая считается неразрешимой.

Выбор значений N и g может иметь существенное влияние на безопасность этой системы. Модуль N должен быть большим и простым числом. Число (N – 1)/2 также должно быть простым числом. Число g желательно выбирать таким, чтобы оно было примитивным элементом множества Z_N. (В принципе достаточно, чтобы число g генерировало большую подгруппу мультипликативной группы по mod N).

Алгоритм открытого распределения ключей Диффи–Хеллмана позволяет обойтись без защищенного канала для передачи ключей. Однако, работая с этим алгоритмом, необходимо иметь гарантию того, что пользователь A получил открытый ключ именно от пользователя B, и наоборот. Эта проблема решается с помощью электронной подписи, которой подписываются сообщения об открытом ключе.

Метод Диффи–Хеллмана дает возможность шифровать данные при каждом сеансе связи на новых ключах. Это позволяет не хранить секреты на дискетах или других носителях. Не следует забывать, что любое хранение секретов повышает вероятность попадания их в руки конкурентов или противника.

Преимущество метода Диффи–Хеллмана по сравнению с методом RSA заключается в том, что формирование общего секретного ключа происходит в сотни раз быстрее. В системе RSA генерация новых секретных и открытых ключей основана на генерации новых простых чисел, что занимает много времени.

Протокол SKIP управления криптоключами. Протокол SKIP (Simple Key management for Internet Protocol) может использоваться в качестве интегрирующей среды и системы управления криптоключами.

Протокол SKIP базируется на криптографии открытых ключей Диффи–Хеллмана и обладает рядом достоинств:

- обеспечивает высокую степень защиты информации;
- обеспечивает быструю смену ключей;
- поддерживает групповые рассылки защищенных сообщений;
- допускает модульную замену систем шифрования;
- вносит минимальную избыточность.

Концепция SKIP-протокола основана на организации множества двухточечных обменов (по алгоритму Диффи–Хеллмана) в компьютерной сети.

- Узел I имеет секретный ключ $i(i=k_I)$ и сертифицированный открытый ключ $g^i \bmod N$.
- Подпись сертификата открытого ключа производится при помощи надежного алгоритма (ГОСТ, DSA и др.). Открытые ключи свободно распространяются центром распределения ключей из общей базы данных.
- Для каждой пары узлов I, J вычисляется совместно используемый секрет (типичная длина 1024 бита): $g^{ij} \bmod N$.
- Разделяемый ключ K_{ij} вычисляется из этого секрета путем уменьшения его до согласованной в рамках протокола длины 64...128 бит.
- Узел вычисляет ключ K_{ij} (используемый как ключ шифрования ключей) для относительно длительного применения и размещает его в защищенной памяти.

Следует отметить, что если сеть содержит n узлов, то в каждом узле должно храниться $(n-1)$ ключей, используемых исключительно для организации связи с соответствующими узлами.

ГЛАВА 8. МЕТОДЫ И СРЕДСТВА ЗАЩИТЫ ОТ УДАЛЕННЫХ АТАК ЧЕРЕЗ СЕТЬ Internet

8.1. Особенности функционирования межсетевых экранов

Интенсивное развитие глобальных компьютерных сетей, появление новых технологий поиска информации привлекают все больше внимания к сети Internet со стороны частных лиц и различных организаций. Многие организации принимают решение об интеграции своих локальных и корпоративных сетей в глобальную сеть. Использование глобальных сетей в коммерческих целях, а также при передаче информации, содержащей сведения конфиденциального характера, влечет за собой необходимость построения эффективной системы защиты информации. В настоящее время в России глобальные сети применяются для передачи коммерческой информации различного уровня конфиденциальности, например для связи с удаленными офисами из головной штабквартиры организации или создания Web-страницы организации с размещенной на ней рекламой и деловыми предложениями [66].

Вряд ли нужно перечислять все преимущества, которые получает современное предприятие, имея доступ к глобальной сети Internet. Но, как и многие другие новые технологии, использование Internet имеет и негативные последствия. Развитие глобальных сетей привело к многократному увеличению количества пользователей и увеличению количества атак на компьютеры, подключенные к сети Internet. Ежегодные потери, обусловленные недостаточным уровнем защищенности компьютеров, оцениваются десятками миллионов долларов. При подключении к Internet локальной или корпоративной сети необходимо позаботиться об обеспечении информационной безопасности этой сети.

Глобальная сеть Internet создавалась как открытая система, предназначенная для свободного обмена информацией. В силу открытости своей идеологии Internet предоставляет для злоумышленников значительно большие возможности по сравнению с традиционными информационными системами. Через Internet нарушитель может:

вторгнуться во внутреннюю сеть предприятия и получить несанкционированный доступ к конфиденциальной информации;
незаконно скопировать важную и ценную для предприятия информацию;
получить пароли, адреса серверов, а подчас и их содержимое;
входить в информационную систему предприятия под именем зарегистрированного пользователя и т.д.

С помощью полученной злоумышленником информации может быть серьезно подорвана конкурентоспособность предприятия доверие его клиентов.

Ряд задач по отражению наиболее вероятных угроз для внутренних сетей способны решать *межсетевые экраны*. В отечественной литературе до последнего времени использовались вместо этого термина другие термины иностранного происхождения: брандмауэр и firewall. Вне компьютерной сферы брандмауэром или firewall) называют стену, сделанную из негорючих материалов и препятствующую распространению пожара. В сфере компьютерных сетей межсетевой экран представляет собой барьер, защищающий от фигурального пожара – попыток злоумышленников вторгнуться во внутреннюю сеть для того, чтобы скопировать, изменить или стереть информацию либо воспользоваться памятью или вычислительной мощностью работающих в этой сети компьютеров. Межсетевой экран призван обеспечить безопасный доступ к внешней сети и ограничить доступ внешних пользователей к внутренней сети.

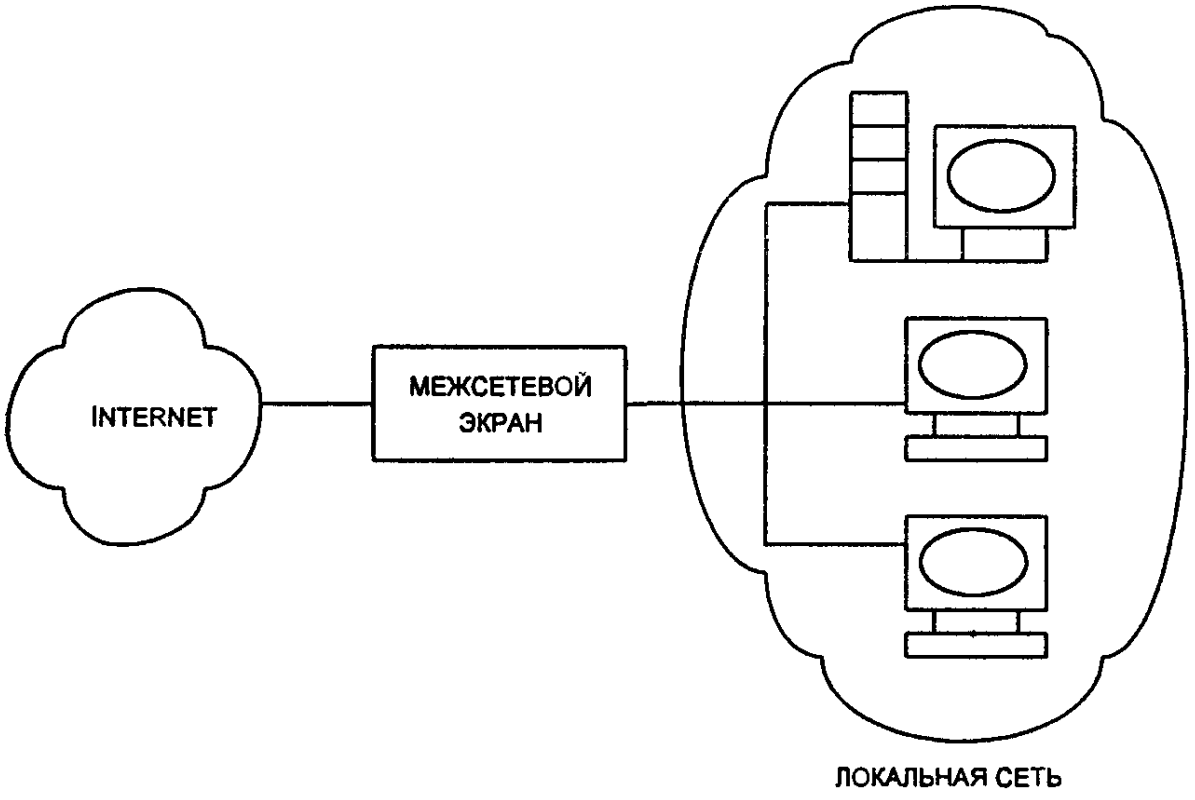


Рис. 8.1. Схема установления межсетевого экрана

Межсетевой экран (МЭ) – это система межсетевой защиты, позволяющая разделить общую сеть на две части или более и реализовать набор правил, определяющих условия прохождения пакетов с данными через границу из одной части общей сети в другую (рис. 8.1). Как правило, эта граница проводится между корпоративной (локальной) сетью предприятия и глобальной сетью Internet, хотя ее можно провести и внутри корпоративной сети предприятия. МЭ пропускает через себя весь трафик, принимая для каждого проходящего пакета решение – пропускать его или отбросить. Для того чтобы МЭ мог осуществить это, ему необходимо определить набор правил фильтрации.

Обычно межсетевые экраны защищают внутреннюю сеть предприятия от "вторжений" из глобальной сети Internet, однако они могут использоваться и для защиты от "нападений" из корпоративной интрасети, к которой подключена локальная сеть предприятия. Ни один межсетевой экран не может гарантировать полной защиты внутренней сети при всех возможных обстоятельствах. Однако для большинства коммерческих организаций установка межсетевого экрана является необходимым условием обеспечения безопасности внутренней сети. Главный довод в пользу применения межсетевого экрана состоит в том, что без него системы внутренней сети подвергаются опасности со стороны слабо защищенных служб сети Internet, а также зондированию и атакам с каких-либо других хост-компьютеров внешней сети [89].

Проблемы недостаточной информационной безопасности являются "врожденными" практически для всех протоколов и служб Internet. Большая часть этих проблем связана с исторической зависимостью Internet от операционной системы UNIX. Известно, что сеть Arpanet (прародитель Internet) строилась как сеть, связывающая исследовательские центры, научные, военные и правительственные учреждения, крупные университеты США. Эти структуры использовали операционную систему UNIX в качестве платформы для коммуникаций и решения собственных задач. Поэтому особенности методологии программирования в среде UNIX и ее архитектуры наложили отпечаток на реализацию протоколов обмена и политики безопасности в сети. Из-за открытости и распространенности система UNIX стала любимой добычей хакеров. Поэтому совсем не удивительно, что набор протоколов TCP/IP, который обеспечивает коммуникации в глобальной сети Internet и в получающих все большую популярность интрасетях, имеет "врожденные" недостатки защиты. То же самое можно сказать и о ряде служб Internet.

Набор протоколов управления передачей сообщений в Internet (Transmission Control Protocol/Internet Protocol – TCP/IP) используется для организации коммуникаций в неоднородной сетевой сре-

де, обеспечивая совместимость между компьютерами разных типов. Совместимость – одно из основных преимуществ TCP/IP, поэтому большинство локальных компьютерных сетей поддерживает эти протоколы. Кроме того, протоколы TCP/IP предоставляют доступ к ресурсам глобальной сети Internet. Поскольку TCP/IP поддерживает маршрутизацию пакетов, он обычно используется в качестве межсетевого протокола. Благодаря своей популярности TCP/IP стал стандартом де-факто для межсетевого взаимодействия.

В заголовках пакетов TCP/IP указывается информация, которая может подвергнуться нападениям хакеров. В частности, хакер может подменить адрес отправителя в своих "вредоносных" пакетах, после чего они будут выглядеть, как пакеты, передаваемые авторизированным клиентом.

Отметим "врожденные слабости" некоторых распространенных служб Internet [46].

Простой протокол передачи электронной почты (Simple Mail Transfer Protocol – SMTP) позволяет осуществлять почтовую транспортную службу Internet. Одна из проблем безопасности, связанная с этим протоколом, заключается в том, что пользователь не может проверить адрес отправителя в заголовке сообщения электронной почты. В результате хакер может послать во внутреннюю сеть большое количество почтовых сообщений, что приведет к перегрузке и блокированию работы почтового сервера.

Популярная в Internet *программа электронной почты Sendmail* использует для работы некоторую сетевую информацию – IP-адрес отправителя. Перехватывая сообщения, отправляемые с помощью Sendmail, хакер может употребить эту информацию для нападений, например для спуфинга (подмены адресов).

Протокол передачи файлов (File Transfer Protocol – FTP) обеспечивает передачу текстовых и двоичных файлов, поэтому его часто используют в Internet для организации совместного доступа к информации. Его обычно рассматривают как один из методов работы с удаленными сетями. На FTP-серверах хранятся документы, программы, графика и другие виды информации. К данным этих файлов на FTP-серверах нельзя обратиться напрямую. Это можно сделать, только переписав их целиком с FTP-сервера на локальный сервер. Некоторые FTP-серверы ограничивают доступ пользователей к своим архивам данных с помощью пароля, другие же предоставляют свободный доступ (так называемый анонимный FTP-сервер). При использовании опции анонимного FTP для своего сервера пользователь должен быть уверен, что на нем хранятся только файлы, предназначенные для свободного распространения.

Служба сетевых имен (Domain Name System – DNS) представляет собой распределенную базу данных, которая преобразует имена пользователей и хост-компьютеров в IP-адреса, указываемые в заголовках пакетов, и наоборот. DNS также хранит информацию о структуре сети компании, например количестве компьютеров с IP-адресами в каждом домене. Одной из проблем DNS является то, что эту базу данных очень трудно "скрыть" от неавторизированных пользователей. В результате DNS часто используется хакерами как источник информации об именах доверенных хост-компьютеров.

Служба эмуляции удаленного терминала (TELNET) употребляется для подключения к удаленным системам, присоединенным к сети; применяет базовые возможности по эмуляции терминала. При использовании этого сервиса Internet пользователи должны регистрироваться на сервере TELNET, вводя свои имя и пароль. После аутентификации пользователя его рабочая станция функционирует в режиме "тупого" терминала, подключенного к внешнему хост-компьютеру. С этого терминала пользователь может вводить команды, которые обеспечивают ему доступ к файлам и запуск программ. Подключившись к серверу TELNET, хакер может сконфигурировать его программу таким образом, чтобы она записывала имена и пароли пользователей.

Всемирная паутина (World Wide Web – WWW) – это система, основанная на сетевых приложениях, которые позволяют пользователям просматривать содержимое различных серверов в Internet или интрасетях. Самым полезным свойством WWW является использование гипертекстовых документов, в которые встроены ссылки на другие документы и Web-узлы, что дает пользователям возможность легко переходить от одного узла к другому. Однако это же свойство является и наиболее слабым местом системы WWW, поскольку ссылки на Web-узлы, хранящиеся в гипертекстовых документах, содержат информацию о том, как осуществляется доступ к соответствующим узлам. Используя эту информацию, хакеры могут разрушить Web-узел или получить доступ к хранящейся в нем конфиденциальной информации.

К уязвимым службам и протоколам Internet относятся также протокол копирования UUCP, протокол маршрутизации RIP, графическая оконная система X Windows и др.

Решение о том, фильтровать ли с помощью межсетевого экрана конкретные протоколы и адреса, зависит от принятой в защищаемой сети политики безопасности. Межсетевой экран является набором компонентов, настраиваемых таким образом, чтобы реализовать выбранную политику безопасности. В частности, необходимо решить, будет ли ограничен доступ пользователей к определенным службам Internet на базе протоколов TCP/IP и если будет, то до какой степени.

Политика сетевой безопасности каждой организации должна включать две составляющие [20]:

- политику доступа к сетевым сервисам;
- политику реализации межсетевых экранов.

В соответствии с политикой доступа к сетевым сервисам определяется список сервисов Internet, к которым пользователи должны иметь ограниченный доступ. Задаются также ограничения на методы доступа, например, на использование протоколов SLIP (Serial Line Internet Protocol) и PPP (Point-to-Point Protocol). Ограничение методов доступа необходимо для того, чтобы пользователи не могли обращаться к "запрещенным" сервисам Internet обходными путями. Например, если для ограничения доступа в Internet сетевой администратор устанавливает специальный шлюз, который не дает возможности пользователям работать в системе WWW, они могли бы установить PPP-соединения с Web-серверами по коммутируемой линии.

Политика доступа к сетевым сервисам обычно основывается на одном из следующих принципов:

1) запретить доступ из Internet во внутреннюю сеть, но разрешить доступ из внутренней сети в Internet;

2) разрешить ограниченный доступ во внутреннюю сеть из Internet, обеспечивая работу только отдельных "авторизированных" систем, например почтовых серверов.

В соответствии с политикой реализации межсетевых экранов определяются правила доступа к ресурсам внутренней сети. Прежде всего необходимо установить, насколько "доверительной" или "подозрительной" должна быть система защиты. Иными словами, правила доступа к внутренним ресурсам должны базироваться на одном из следующих принципов:

1) запрещать все, что не разрешено в явной форме;

2) разрешать все, что не запрещено в явной форме.

Реализация межсетевого экрана на основе первого принципа обеспечивает значительную защищенность. Однако правила доступа, сформулированные в соответствии с этим принципом, могут доставлять большие неудобства пользователям, а кроме того, их реализация обходится достаточно дорого. При реализации второго принципа внутренняя сеть оказывается менее защищенной от нападений хакеров, однако пользоваться ей будет удобнее и потребуется меньше затрат.

Эффективность защиты внутренней сети с помощью межсетевых экранов зависит не только от выбранной политики доступа к сетевым сервисам и ресурсам внутренней сети, но и от рациональности выбора и использования основных компонентов межсетевого экрана.

Функциональные требования к межсетевым экранам включают:
- требования к фильтрации на сетевом уровне;
- требования к фильтрации на прикладном уровне;
- требования по настройке правил фильтрации и администрированию;
- требования к средствам сетевой аутентификации;
- требования по внедрению журналов и учету.

8.2. Основные компоненты межсетевых экранов

Большинство компонентов межсетевых экранов можно отнести к одной из трех категорий:
- фильтрующие маршрутизаторы;
- шлюзы сетевого уровня;
- шлюзы прикладного уровня.

Эти категории можно рассматривать как базовые компоненты реальных межсетевых экранов. Лишь немногие межсетевые экраны включают только одну из перечисленных категорий. Тем не менее эти категории отражают ключевые возможности, отличающие межсетевые экраны друг от друга.

Фильтрующие маршрутизаторы

Фильтрующий маршрутизатор представляет собой маршрутизатор или работающую на сервере программу, сконфигурированные таким образом, чтобы фильтровать входящие и исходящие пакеты. Фильтрация пакетов осуществляется на основе информации, содержащейся в TCP- и IP-заголовках пакетов. Процесс инкапсуляции передаваемых данных и формирования TCP- и IP-заголовков пакетов с данными в стеке протоколов TCP/IP показан на рис. 8.2.

Фильтрующий маршрутизатор обычно может фильтровать IP-пакеты на основе группы следующих полей заголовка пакета [20]:
- IP-адрес отправителя (адрес системы, которая послала пакет);
- IP-адрес получателя (адрес системы, которая принимает пакет);
- порт отправителя (порт соединения в системе-отправителе);
- порт получателя (порт соединения в системе-получателе).

Порт – это программное понятие, которое используется клиентом или сервером для посылки или приема сообщений; порт идентифицируется 16-битовым числом.

В настоящее время не все фильтрующие маршрутизаторы фильтруют пакеты по TCP/UDP-порту отправителя, однако многие производители маршрутизаторов начали обеспечивать такую воз-

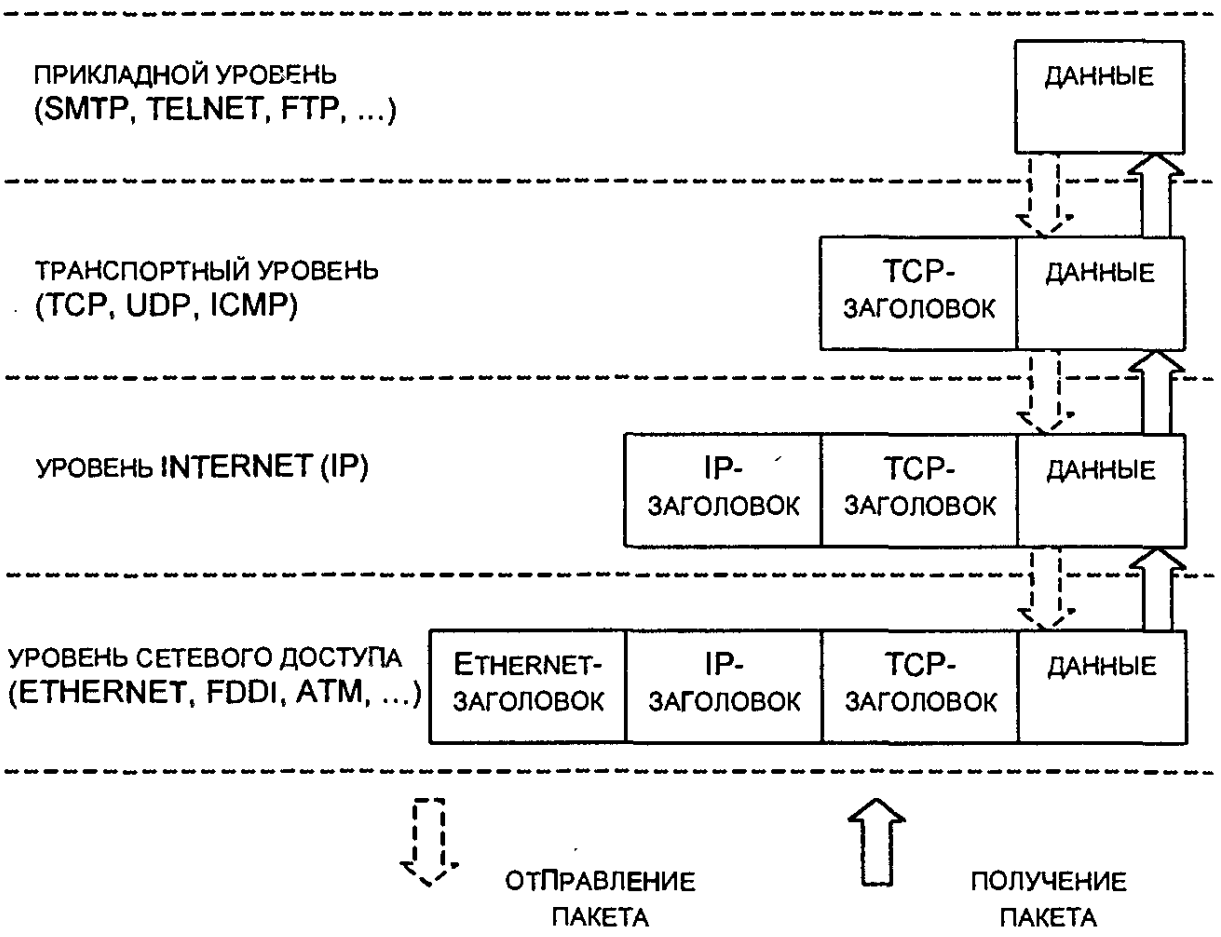


Рис. 8.2. Схема инкапсуляции данных в стеке протоколов TCP/IP

можность. Некоторые маршрутизаторы проверяют, с какого сетевого интерфейса маршрутизатора пришел пакет, и затем используют эту информацию как дополнительный критерий фильтрации.

Фильтрация может быть реализована различным образом для блокирования соединений с определенными хост-компьютерами или портами. Например, можно блокировать соединения, идущие от конкретных адресов тех хост-компьютеров и сетей, которые считаются враждебными или ненадежными.

Добавление фильтрации по портам TCP и UDP к фильтрации по IP-адресам обеспечивает большую гибкость. Известно, что такие серверы, как демон TELNET, обычно связаны с конкретными портами (например, порт 23 протокола TELNET). Если межсетевой экран может блокировать соединения TCP или UDP с определенными портами или от них, то можно реализовать политику безопасности, при которой некоторые виды соединений устанавливаются только с конкретными хост-компьютерами.

Например, внутренняя сеть может блокировать все входные соединения со всеми хост-компьютерами за исключением нескольких систем. Для этих систем могут быть разрешены только определенные сервисы (SMTP для одной системы и TELNET или FTP – для другой). При фильтрации по портам TCP и UDP эта политика

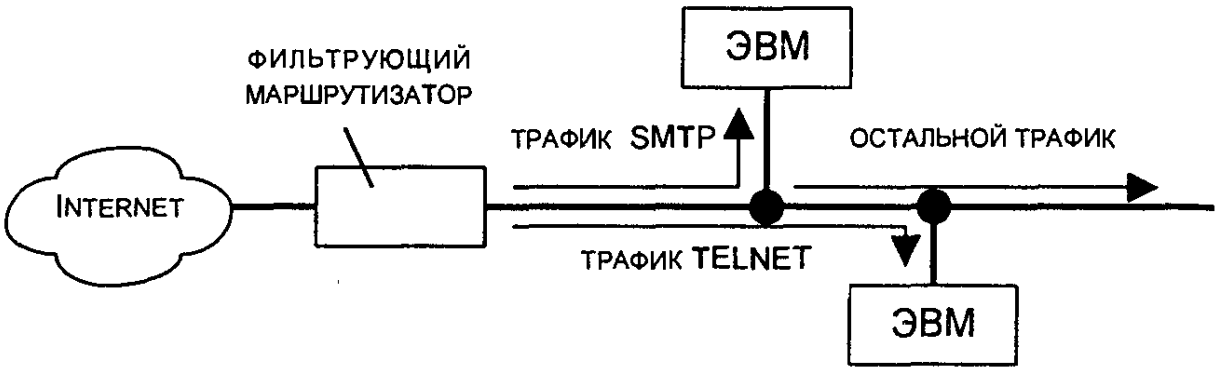


Рис. 8.3. Схема фильтрации трафика SMTP и TELNET

может быть реализована фильтрующим маршрутизатором или хост-компьютером с возможностью фильтрации пакетов (рис. 8.3).

В качестве примера работы фильтрующего маршрутизатора рассмотрим реализацию политики безопасности, допускающей определенные соединения с внутренней сетью с адресом 123.4.*.* Соединения TELNET разрешаются только с одним хост-компьютером с адресом 123.4.5.6, который может быть прикладным TELNET-шлюзом, а SMTP-соединения – только с двумя хост-компьютерами с адресами 123.4.5.7 и 123.4.5.8, которые могут быть двумя шлюзами электронной почты. Обмен по NNTP (Network News Transfer Protocol) разрешается только от сервера новостей с адресом 129.6.48.254 и только с NNTP-сервером сети с адресом 123.4.5.9, а протокол NTP (сетевого времени) – для всех хост-компьютеров. Все другие серверы и пакеты блокируются [16]. Соответствующий набор правил сведен в табл. 8.1.

Таблица 8.1

Правила фильтрации

Тип	Адрес отправителя	Адрес получателя	Порт отправителя	Порт получателя	Действие
TCP	*	123.4.5.6	>1023	23	Разрешить
TCP	*	123.4.5.7	>1023	25	Разрешить
TCP	*	123.4.5.8	>1023	25	Разрешить
TCP	129.6.48.254	123.4.5.9	>1023	119	Разрешить
UDP	*	123.4.*.*	>1023	123	Разрешить
*	*	*	*	*	Запретить

Первое правило позволяет пропускать пакеты TCP из сети Internet от любого источника с номером порта большим, чем 1023, к получателю с адресом 123.4.5.6 в порт 23. Порт 23 связан с сервером TELNET, а все клиенты TELNET должны иметь непривилегированные порты с номерами не ниже 1024.

Второе и третье правила работают аналогично и разрешают передачу пакетов к получателям с адресами 123.4.5.7 и 123.4.5.8 в порт 25, используемый SMTP.

Четвертое правило пропускает пакеты к NNTP-серверу сети, но только от отправителя с адресом 129.6.48.254 к получателю с адресом 123.4.5.9 с портом назначения 119 (129.6.48.254 – единственный NNTP-сервер, от которого внутренняя сеть получает новости, поэтому доступ к сети для выполнения протокола NNTP ограничен только этой системой).

Пятое правило разрешает трафик NTP, который использует протокол UDP вместо TCP, от любого источника к любому получателю внутренней сети.

Наконец, шестое правило блокирует все остальные пакеты. Если бы этого правила не было, маршрутизатор мог бы блокировать, а мог бы и не блокировать другие типы пакетов. Выше был рассмотрен очень простой пример фильтрации пакетов. Реально используемые правила позволяют осуществить более сложную фильтрацию и являются более гибкими.

Правила фильтрации пакетов формулируются сложно, и обычно нет средств для тестирования их корректности, кроме медленного ручного тестирования. У некоторых фильтрующих маршрутизаторов нет средств протоколирования, поэтому, если правила фильтрации пакетов все-таки позволят опасным пакетам пройти через маршрутизатор, такие пакеты не смогут быть выявлены до обнаружения последствий проникновения.

Даже если администратору сети удастся создать эффективные правила фильтрации, их возможности остаются ограниченными. Например, администратор задает правило, в соответствии с которым маршрутизатор будет отбраковывать все пакеты с неизвестным адресом отправителя. Однако хакер может использовать в качестве адреса отправителя в своем "вредоносном" пакете реальный адрес доверенного (авторизированного) клиента. В этом случае фильтрующий маршрутизатор не сумеет отличить поддельный пакет от настоящего и пропустит его. Практика показывает, что подобный вид нападения, называемый *подменой адреса*, довольно широко распространен в сети Internet и часто оказывается эффективным.

Межсетевой экран с фильтрацией пакетов, работающий только на сетевом уровне эталонной модели взаимодействия открытых систем OSI-ISO, обычно проверяет информацию, содержащуюся только в IP-заголовках пакетов. Поэтому обмануть его несложно: хакер создает заголовок, который удовлетворяет разрешающим правилам фильтрации. Кроме заголовка пакета, никакая другая содержащаяся в нем информация межсетевыми экранами данной категории не проверяется.

К положительным качествам фильтрующих маршрутизаторов следует отнести:

- сравнительно невысокую стоимость;
- гибкость в определении правил фильтрации;
- небольшую задержку при прохождении пакетов.

Недостатками фильтрующих маршрутизаторов являются:

- внутренняя сеть видна (маршрутизируется) из сети Internet;
- правила фильтрации пакетов трудны в описании и требуют очень хороших знаний технологий TCP и UDP;
- при нарушении работоспособности межсетевого экрана с фильтрацией пакетов все компьютеры за ним становятся полностью незащищенными либо недоступными;
- аутентификацию с использованием IP-адреса можно обмануть путем подмены IP-адреса (атакующая система выдает себя за другую, используя ее IP-адрес);
- отсутствует аутентификация на пользовательском уровне.

Шлюзы сетевого уровня

Шлюз сетевого уровня иногда называют системой трансляции сетевых адресов или шлюзом сеансового уровня модели OSI. Такой шлюз исключает прямое взаимодействие между авторизированным клиентом и внешним хост-компьютером.

Шлюз сетевого уровня принимает запрос доверенного клиента на конкретные услуги и после проверки допустимости запрошенного сеанса устанавливает соединение с внешним хост-компьютером. После этого шлюз копирует пакеты в обоих направлениях, не осуществляя их фильтрации.

Шлюз следит за подтверждением (квитированием) связи между авторизированным клиентом и внешним хост-компьютером, определяя, является ли запрашиваемый сеанс связи допустимым. Чтобы выявить допустимость запроса на сеанс связи, шлюз выполняет следующую процедуру.

Когда авторизированный клиент запрашивает некоторый сервис, шлюз принимает этот запрос, проверяя, удовлетворяет ли этот клиент базовым критериям фильтрации (например, может ли DNS-сервер определить IP-адрес клиента и ассоциированное с ним имя). Затем, действуя от имени клиента, шлюз устанавливает соединение с внешним хост-компьютером и следит за выполнением процедуры квитирования связи по протоколу TCP. Эта процедура состоит из обмена TCP-пакетами, которые помечаются флагами SYN (синхронизировать) и ACK (подтвердить) (рис.8.4).

Первый пакет сеанса TCP, помеченный флагом SYN и содержащий произвольное число, например 1000, является запросом клиента на открытие сеанса. Внешний хост-компьютер, получивший этот пакет, посылает в ответ пакет, помеченный флагом ACK и содержащий число, на единицу большее, чем в принятом пакете

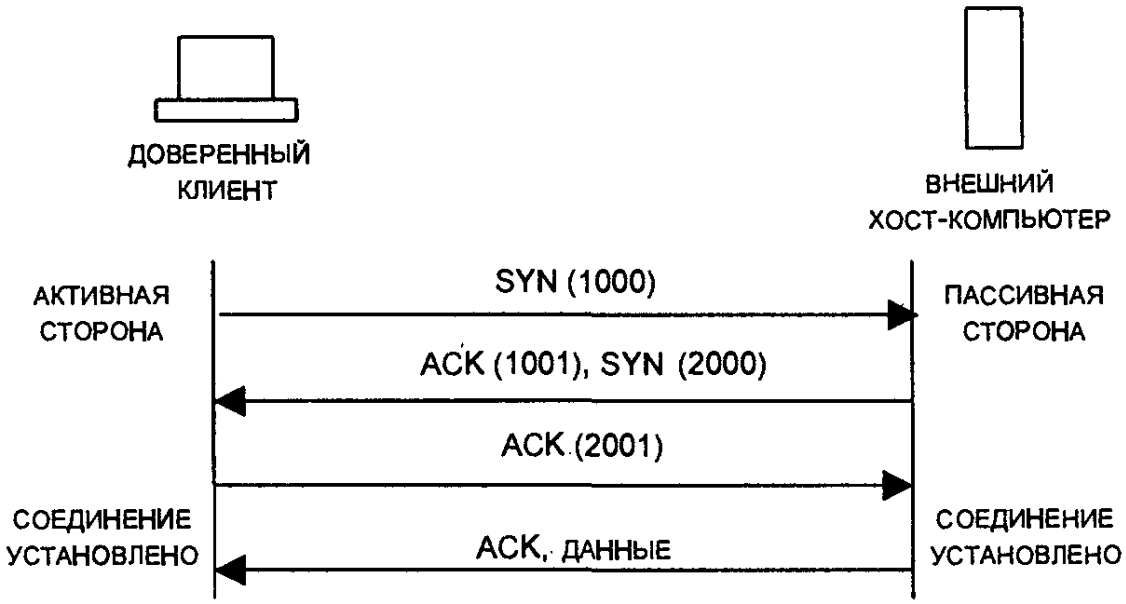


Рис. 8.4. Последовательность передачи пакетоа SYN и ACK в процессе квитирования саязи по протоколу TCP

(в нашем случае 1001), подтверждая тем самым прием пакета SYN от клиента.

Далее осуществляется обратная процедура: хост-компьютер посылает клиенту пакет SYN с исходным числом (например, 2000), а клиент подтверждает его получение передачей пакета ACK, содержащего число 2001. На этом процесс квитирования связи завершается.

Шлюз сетевого уровня признает запрошенное соединение допустимым только в том случае, если при выполнении процедуры квитирования связи флаги SYN и ACK, а также числа, содержащиеся в TCP-пакетах, оказываются логически связанными между собой.

После того как шлюз определил, что доверенный клиент и внешний хост-компьютер являются авторизированными участниками сеанса TCP, и проверил допустимость этого сеанса, он устанавливает соединение. Начиная с этого момента, шлюз копирует и перенаправляет пакеты туда и обратно, не проводя никакой фильтрации. Он поддерживает таблицу установленных соединений, пропуская данные, относящиеся к одному из сеансов связи, зафиксированных в этой таблице. Когда сеанс завершается, шлюз удаляет соответствующий элемент из таблицы и разрывает цепь, использовавшуюся в данном сеансе.

Для копирования и перенаправления пакетов в шлюзах сетевого уровня применяются специальные приложения, которые называют *канальными посредниками*, поскольку они устанавливают между двумя сетями виртуальную цепь или канал, а затем разрешают пакетам, которые генерируются приложениями TCP/IP, проходить по этому каналу. Канальные посредники поддерживают не-

сколько служб TCP/IP, поэтому шлюзы сетевого уровня могут использоваться для расширения возможностей шлюзов прикладного уровня, работа которых основывается на программах-посредниках конкретных приложений.

Фактически большинство шлюзов сетевого уровня не являются самостоятельными продуктами, а поставляются в комплекте со шлюзами прикладного уровня. Примерами таких шлюзов являются Gauntlet Internet Firewall компании Trusted Information Systems, Alta Vista Firewall компании DEC и ANS Interlock компании ANS. Например, Alta Vista Firewall использует канальные посредники прикладного уровня для каждой из шести служб TCP/IP, к которым относятся, в частности, FTP, HTTP (Hyper Text Transport Protocol) и TELNET. Кроме того, межсетевой экран компании DEC обеспечивает шлюз сетевого уровня, поддерживающий другие общедоступные службы TCP/IP, такие как Gopher и SMTP, для которых межсетевой экран не предоставляет посредников прикладного уровня.

Шлюз сетевого уровня выполняет еще одну важную функцию защиты: он используется в качестве *сервера-посредника*. Этот сервер-посредник выполняет *процедуру трансляции адресов*, при которой происходит преобразование внутренних IP-адресов в один "надежный" IP-адрес. Этот адрес ассоциируется с межсетевым экраном, из которого передаются все исходящие пакеты. В результате в сети со шлюзом сетевого уровня все исходящие пакеты оказываются отправленными из этого шлюза, что исключает прямой контакт между внутренней (авторизированной) сетью и потенциально опасной внешней сетью. IP-адрес шлюза сетевого уровня становится единственно активным IP-адресом, который попадает во внешнюю сеть. Таким образом шлюз сетевого уровня и другие серверы-посредники защищают внутренние сети от нападений типа подмены адресов.

После установления связи шлюзы сетевого уровня фильтруют пакеты только на сеансовом уровне модели OSI, т.е. не могут проверять содержимое пакетов, передаваемых между внутренней и внешней сетью на уровне прикладных программ. И поскольку эта передача осуществляется "вслепую", хакер, находящийся во внешней сети, может "протолкнуть" свои "вредоносные" пакеты через такой шлюз. После этого хакер обратится напрямую к внутреннему Web-серверу, который сам по себе не может обеспечивать функции межсетевого экрана. Иными словами, если процедура квитирования связи успешно завершена, шлюз сетевого уровня установит соединение и будет "слепо" копировать и перенаправлять все последующие пакеты независимо от их содержимого.

Чтобы фильтровать пакеты, генерируемые определенными сетевыми службами, в соответствии с их содержимым необходим шлюз прикладного уровня.

Шлюзы прикладного уровня

Для устранения ряда недостатков, присущих фильтрующим маршрутизаторам, межсетевые экраны должны использовать дополнительные программные средства для фильтрации сообщений сервисов типа TELNET и FTP. Такие программные средства называются *полномочными серверами (серверами-посредниками)*, а хост-компьютер, на котором они выполняются, – *шлюзом прикладного уровня* [20].

Шлюз прикладного уровня исключает прямое взаимодействие между авторизированным клиентом и внешним хост-компьютером. Шлюз фильтрует все входящие и исходящие пакеты на прикладном уровне. Связанные с приложениями серверы-посредники перенаправляют через шлюз информацию, генерируемую конкретными серверами.

Для достижения более высокого уровня безопасности и гибкости шлюзы прикладного уровня и фильтрующие маршрутизаторы могут быть объединены в одном межсетевом экране. В качестве примера рассмотрим сеть, в которой с помощью фильтрующего маршрутизатора блокируются входящие соединения TELNET и FTP [89]. Этот маршрутизатор допускает прохождение пакетов TELNET или FTP только к одному хост-компьютеру – шлюзу прикладного уровня TELNET/FTP. Внешний пользователь, который хочет соединиться с некоторой системой в сети, должен сначала соединиться со шлюзом прикладного уровня, а затем уже с нужным внутренним хост-компьютером. Это осуществляется следующим образом:

1) сначала внешний пользователь устанавливает TELNET-соединение со шлюзом прикладного уровня с помощью протокола TELNET и вводит имя интересующего его внутреннего хост-компьютера;

2) шлюз проверяет IP-адрес отправителя и разрешает или запрещает соединение в соответствии с тем или иным критерием доступа;

3) пользователю может потребоваться аутентификация (возможно, с помощью одноразовых паролей);

4) сервер-посредник устанавливает TELNET-соединение между шлюзом и внутренним хост-компьютером;

5) сервер-посредник осуществляет передачу информации между этими двумя соединениями;

6) шлюз прикладного уровня регистрирует соединение.

Этот пример наглядно показывает преимущества использования полномочных серверов-посредников.

- Полномочные серверы-посредники пропускают только те службы, которые им поручено обслуживать. Иначе говоря, если шлюз прикладного уровня наделен полномочиями (и полномочными серверами-посредниками) для служб FTP и TELNET, то в защищаемой сети будут разрешены только FTP и TELNET, а все другие службы будут полностью блокированы. Для некоторых организаций такой вид безопасности имеет большое значение, так как он гарантирует, что через межсетевой экран будут пропускаться только те службы, которые считаются безопасными.
- Полномочные серверы-посредники обеспечивают возможность фильтрации протокола. Например, некоторые межсетевые экраны, использующие шлюзы прикладного уровня, могут фильтровать FTP-соединения и запрещать использование команды FTP *put*, что гарантированно не позволяет пользователям записывать информацию на анонимный FTP-сервер.

В дополнение к фильтрации пакетов многие шлюзы прикладного уровня регистрируют все выполняемые сервером действия и, что особенно важно, предупреждают сетевого администратора о возможных нарушениях защиты. Например, при попытках проникновения в сеть извне BorderWare Firewall Server компании Secure Computing позволяет фиксировать адреса отправителя и получателя пакетов, время, в которое эти попытки были предприняты, и используемый протокол. Межсетевой экран Black Hole компании Milkyway Networks регистрирует все действия сервера и предупреждает администратора о возможных нарушениях, посылая ему сообщение по электронной почте или на пейджер. Аналогичные функции выполняют и ряд других шлюзов прикладного уровня.

Шлюзы прикладного уровня позволяют обеспечить наиболее высокий уровень защиты, поскольку взаимодействие с внешним миром реализуется через небольшое число прикладных полномочных программ-посредников, полностью контролирующих весь входящий и выходящий трафик.

Шлюзы прикладного уровня имеют ряд серьезных преимуществ по сравнению с обычным режимом, при котором прикладной трафик пропускается непосредственно к внутренним хост-компьютерам. Перечислим эти преимущества.

- *Невидимость структуры защищаемой сети* из глобальной сети Internet. Имена внутренних систем можно не сообщать внешним системам через DNS, поскольку шлюз прикладного уровня может быть единственным хост-компьютером, имя которого должно быть известно внешним системам.
- *Надежная аутентификация и регистрация*. Прикладной трафик может быть аутентифицирован, прежде чем он достигнет внутренних хост-компьютеров, и может быть зарегистрирован более эффективно, чем с помощью стандартной регистрации.

- *Оптимальное соотношение между ценой и эффективностью.* Дополнительные программные или аппаратные средства для аутентификации или регистрации нужно устанавливать только на шлюзе прикладного уровня.
- *Простые правила фильтрации.* Правила на фильтрующем маршрутизаторе оказываются менее сложными, чем они были бы, если бы маршрутизатор сам фильтровал прикладной трафик и отправлял его большому числу внутренних систем. Маршрутизатор должен пропускать прикладной трафик, предназначенный только для шлюза прикладного уровня, и блокировать весь остальной трафик.
- *Возможность организации большого числа проверок.* Защита на уровне приложений позволяет осуществлять большое количество дополнительных проверок, что снижает вероятность взлома с использованием "дыр" в программном обеспечении.

К недостаткам шлюзов прикладного уровня относятся:

- более низкая производительность по сравнению с фильтрующими маршрутизаторами; в частности, при использовании клиент-серверных протоколов, таких как TELNET, требуется двухшаговая процедура для входных и выходных соединений;
- более высокая стоимость по сравнению с фильтрующими маршрутизаторами.

Помимо TELNET и FTP шлюзы прикладного уровня обычно используются для электронной почты, X Windows и некоторых других служб.

Усиленная аутентификация

Одним из важных компонентов концепции межсетевых экранов является аутентификация (проверка подлинности пользователя). Прежде чем пользователю будет предоставлено право воспользоваться тем или иным сервисом, необходимо убедиться, что он действительно тот, за кого себя выдает.

Одним из способов аутентификации является использование стандартных UNIX-паролей. Однако эта схема наиболее уязвима с точки зрения безопасности – пароль может быть перехвачен и использован другим лицом. Многие инциденты в сети Internet произошли отчасти из-за уязвимости традиционных паролей. Злоумышленники могут наблюдать за каналами в сети Internet и перехватывать передающиеся в них открытым текстом пароли, поэтому схему аутентификации с традиционными паролями следует признать устаревшей.

Для преодоления этого недостатка разработан ряд средств усиленной аутентификации: смарт-карты, персональные жетоны, биометрические механизмы и т. п. Хотя в них задействованы раз-

ные механизмы аутентификации, общим для них является то, что пароли, генерируемые этими устройствами, не могут быть повторно использованы нарушителем, наблюдающим за установлением связи. Поскольку проблема с паролями в сети Internet является постоянной, межсетевой экран для соединения с Internet, не располагающий средствами усиленной аутентификации или не использующий их, теряет всякий смысл [20].

Ряд наиболее популярных средств усиленной аутентификации, применяемых в настоящее время, называются *системами с одноразовыми паролями*. Например, смарт-карты или жетоны аутентификации генерируют информацию, которую хост-компьютер использует вместо традиционного пароля. Поскольку смарт-карта или жетон работают вместе с аппаратным и программным обеспечением хост-компьютера, генерируемый пароль уникален для каждого установления сеанса. Результатом является одноразовый пароль, который, даже если он перехватывается, не может быть использован злоумышленником под видом пользователя для установления сеанса с хост-компьютером.

Так как межсетевые экраны могут централизовать управление доступом в сети, они являются подходящим местом для установки программ или устройств усиленной аутентификации. Хотя средства усиленной аутентификации могут использоваться на каждом хост-компьютере, более практично их размещение на межсетевом экране. На рис.8.5 показано, что в сети без межсетевого экрана, использующего меры усиленной аутентификации, неаутентифицированный трафик таких приложений, как TELNET или FTP, может напрямую проходить к системам в сети. Если хост-компьютеры не применяют мер усиленной аутентификации, злоумышленник может попытаться взломать пароли или перехватить сетевой трафик с целью найти в нем сеансы, в ходе которых передаются пароли.

На рис.8.5 показана также сеть с межсетевым экраном, использующим усиленную аутентификацию. В этом случае сеансы

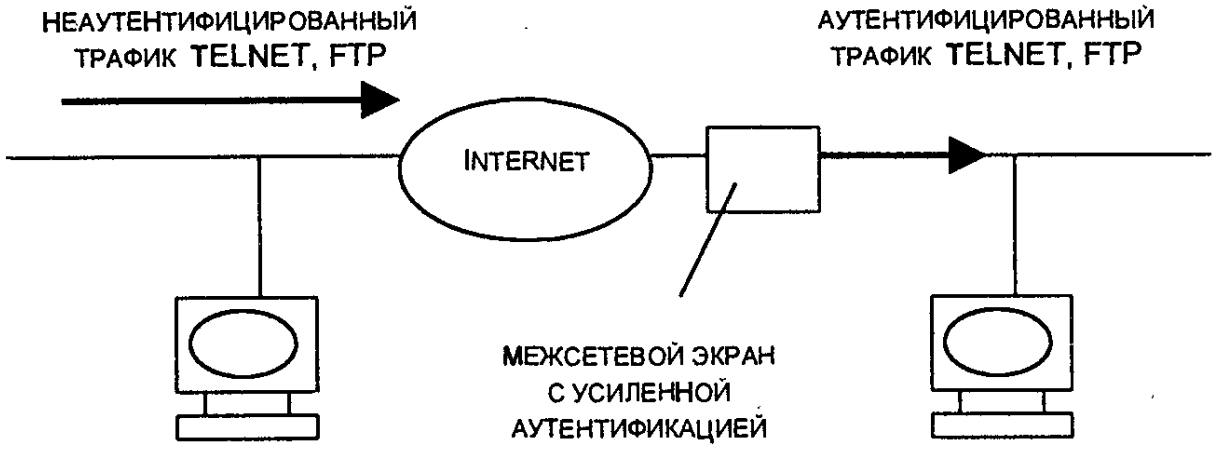


Рис.8.5. Схема использования усиленной аутентификации в межсетевом экране для предварительной аутентификации трафика TELNET, FTP

TELNET или FTP, устанавливаемые со стороны сети Internet с системами сети, должны проходить проверку с помощью средств усиленной аутентификации, прежде чем они будут разрешены. Системы сети могут запрашивать для разрешения доступа и статические пароли, но эти пароли, даже если они будут перехвачены злоумышленником, нельзя будет использовать, так как средства усиленной аутентификации и другие компоненты межсетевого экрана предотвращают проникновение злоумышленников или обход ими межсетевого экрана.

8.3. Основные схемы сетевой защиты на базе межсетевых экранов

При подключении корпоративной или локальной сети к глобальным сетям администратор сетевой безопасности должен решать следующие задачи:

- защита корпоративной или локальной сети от несанкционированного удаленного доступа со стороны глобальной сети;
- скрытие информации о структуре сети и ее компонентов от пользователей глобальной сети;
- разграничение доступа в защищаемую сеть из глобальной сети и из защищаемой сети в глобальную сеть.

Необходимость работы с удаленными пользователями требует установления жестких ограничений доступа к информационным ресурсам защищаемой сети. При этом часто возникает потребность в организации в составе корпоративной сети нескольких сегментов с разными уровнями защищенности:

- свободно доступные сегменты (например, рекламный WWW-сервер),
- сегмент с ограниченным доступом (например, для доступа сотрудникам организации с удаленных узлов),
- закрытые сегменты (например, финансовая локальная сеть организации).

Для защиты корпоративной или локальной сети применяются следующие основные схемы организации межсетевых экранов:

- межсетевой экран – фильтрующий маршрутизатор;
- межсетевой экран на основе двупортового шлюза;
- межсетевой экран на основе экранированного шлюза;
- межсетевой экран – экранированная подсеть.

Межсетевой экран – фильтрующий маршрутизатор

Межсетевой экран, основанный на фильтрации пакетов, является самым распространенным и наиболее простым в реализации. Он состоит из фильтрующего маршрутизатора, расположенного

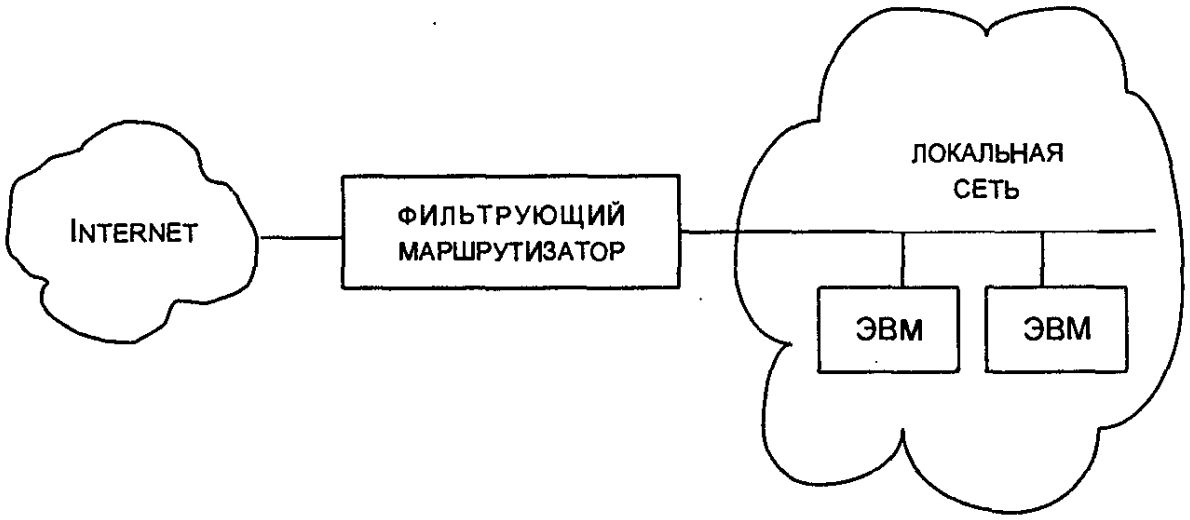


Рис. 8.6. Межсетевой экран на основе фильтрующего маршрутизатора

между защищаемой сетью и сетью Internet (рис. 8.6). Фильтрующий маршрутизатор сконфигурирован для блокирования или фильтрации входящих и исходящих пакетов на основе анализа их адресов и портов [89].

Компьютеры, находящиеся в защищаемой сети, имеют прямой доступ в сеть Internet, в то время как большая часть доступа к ним из Internet блокируется. Часто блокируются такие опасные службы, как X Windows, NIS и NFS. В принципе фильтрующий маршрутизатор может реализовать любую из политик безопасности, описанных ранее. Однако если маршрутизатор не фильтрует пакеты по порту источника и номеру входного и выходного порта, то реализация политики "запрещено все, что не разрешено в явной форме" может быть затруднена.

Межсетевые экраны, основанные на фильтрации пакетов, имеют такие же недостатки, что и фильтрующие маршрутизаторы, причем эти недостатки становятся более ощутимыми при ужесточении требований к безопасности защищаемой сети. Отметим некоторые из них:

- сложность правил фильтрации; в некоторых случаях совокупность этих правил может стать неуправляемой;
- невозможность полного тестирования правил фильтрации; это приводит к незащищенности сети от непротестированных атак;
- практически отсутствующие возможности регистрации событий; в результате администратору трудно определить, подвергался ли маршрутизатор атаке и скомпрометирован ли он;
- каждый хост-компьютер, связанный с сетью Internet, нуждается в своих средствах усиленной аутентификации.

Межсетевой экран на основе двупортового шлюза

Межсетевой экран на базе двупортового прикладного шлюза включает двудомный хост-компьютер с двумя сетевыми интерфейсами. При передаче информации между этими интерфейсами и осуществляется основная фильтрация. Для обеспечения дополнительной защиты между прикладным шлюзом и сетью Internet обычно размещают фильтрующий маршрутизатор (рис. 8.7). В результате между прикладным шлюзом и маршрутизатором образуется внутренняя экранированная подсеть. Эту подсеть можно использовать для размещения доступных извне информационных серверов [106].

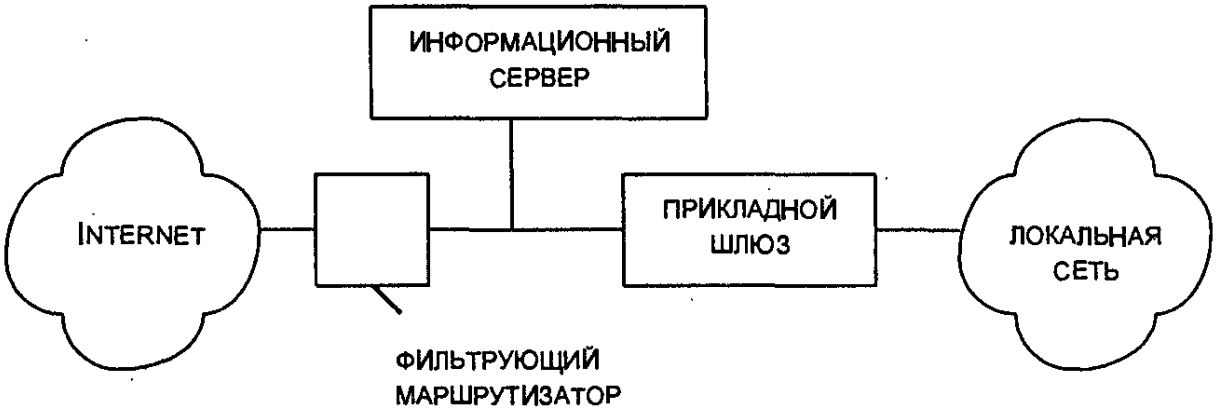


Рис. 8.7. Межсетевой экран с прикладным шлюзом и фильтрующим маршрутизатором

В отличие от схемы межсетевого экрана с фильтрующим маршрутизатором прикладной шлюз полностью блокирует трафик IP между сетью Internet и защищаемой сетью. Только полномочные сервера-посредники, располагаемые на прикладном шлюзе, могут предоставлять услуги и доступ пользователям.

Данный вариант межсетевого экрана реализует политику безопасности, основанную на принципе "запрещено все, что не разрешено в явной форме", при этом пользователю недоступны все службы, кроме тех, для которых определены соответствующие полномочия. Такой подход обеспечивает высокий уровень безопасности, поскольку маршруты к защищенной подсети известны только межсетевому экрану и скрыты от внешних систем.

Рассматриваемая схема организации межсетевого экрана является довольно простой и достаточно эффективной.

Следует отметить, что безопасность двудомного хост-компьютера, используемого в качестве прикладного шлюза, должна поддерживаться на высоком уровне. Любая брешь в его защите может серьезно ослабить безопасность защищаемой сети. Если шлюз окажется скомпрометированным, у злоумышленника появится возможность проникнуть в защищаемую сеть.

Этот межсетевой экран может требовать от пользователей применения средств усиленной аутентификации, а также регистрации доступа, попыток зондирования и атак системы нарушителем.

Для некоторых сетей может оказаться неприемлемой недостаточная гибкость схемы межсетевого экрана с прикладным шлюзом.

Межсетевой экран на основе экранированного шлюза

Межсетевой экран на основе экранированного шлюза объединяет фильтрующий маршрутизатор и прикладной шлюз, размещаемый со стороны внутренней сети. Прикладной шлюз реализуется на хост-компьютере и имеет только один сетевой интерфейс (рис. 8.8).

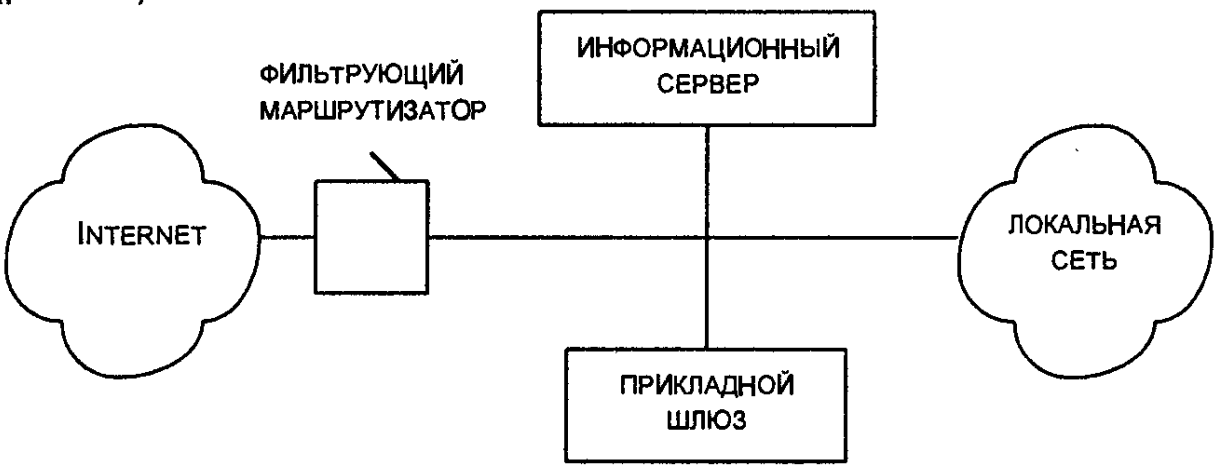


Рис. 8.8. Межсетевой экран с экранированным шлюзом

В этой схеме первичная безопасность обеспечивается фильтрующим маршрутизатором. Пакетная фильтрация в фильтрующем маршрутизаторе может быть реализована одним из следующих способов:

- позволять внутренним хост-компьютерам открывать соединения с хост-компьютерами в сети Internet для определенных сервисов (разрешая доступ к ним средствами пакетной фильтрации);
- запрещать все соединения от внутренних хост-компьютеров (заставляя их использовать полномочные серверы-посредники на прикладном шлюзе).

Эти подходы можно комбинировать для различных сервисов, разрешая некоторым сервисам соединение непосредственно через пакетную фильтрацию, в то время как другим только непрямое соединение через полномочные серверы-посредники. Все зависит от конкретной политики безопасности, принятой во внутренней сети. В частности, пакетная фильтрация на фильтрующем маршрутизаторе может быть организована таким образом, чтобы прикладной шлюз, используя свои полномочные серверы-посредники, обеспечивал для систем защищаемой сети такие сервисы, как TELNET, FTP, SMTP.

Межсетевой экран, выполненный по данной схеме, получается более гибким, но менее безопасным по сравнению с межсетевым экраном с прикладным шлюзом на базе двудомного хост-компьютера. Это обусловлено тем, что в схеме межсетевого экрана с экранированным шлюзом существует потенциальная возможность передачи трафика в обход прикладного шлюза непосредственно к системам локальной сети.

Основной недостаток схемы межсетевого экрана с экранированным шлюзом заключается в том, что если атакующий нарушитель сумеет проникнуть в хост-компьютер, то перед ним окажутся незащищенные системы внутренней сети. Другой недостаток связан с возможной компрометацией маршрутизатора. Если маршрутизатор окажется скомпрометированным, внутренняя сеть станет доступна атакующему нарушителю.

По этим причинам в настоящее время все более популярной становится схема межсетевого экрана с экранированной подсетью.

Межсетевой экран – экранированная подсеть

Межсетевой экран, состоящий из экранированной подсети, представляет собой развитие схемы межсетевого экрана на основе экранированного шлюза. Для создания экранированной подсети используются два экранирующих маршрутизатора (рис. 8.9). Внешний маршрутизатор располагается между сетью Internet и экранируемой подсетью, а внутренний – между экранируемой подсетью и защищаемой внутренней сетью. Экранируемая подсеть содержит прикладной шлюз, а также может включать информационные серверы и другие системы, требующие контролируемого доступа. Эта схема межсетевого экрана обеспечивает хорошую безопасность благодаря организации экранированной подсети, которая еще лучше изолирует внутреннюю защищаемую сеть от Internet.

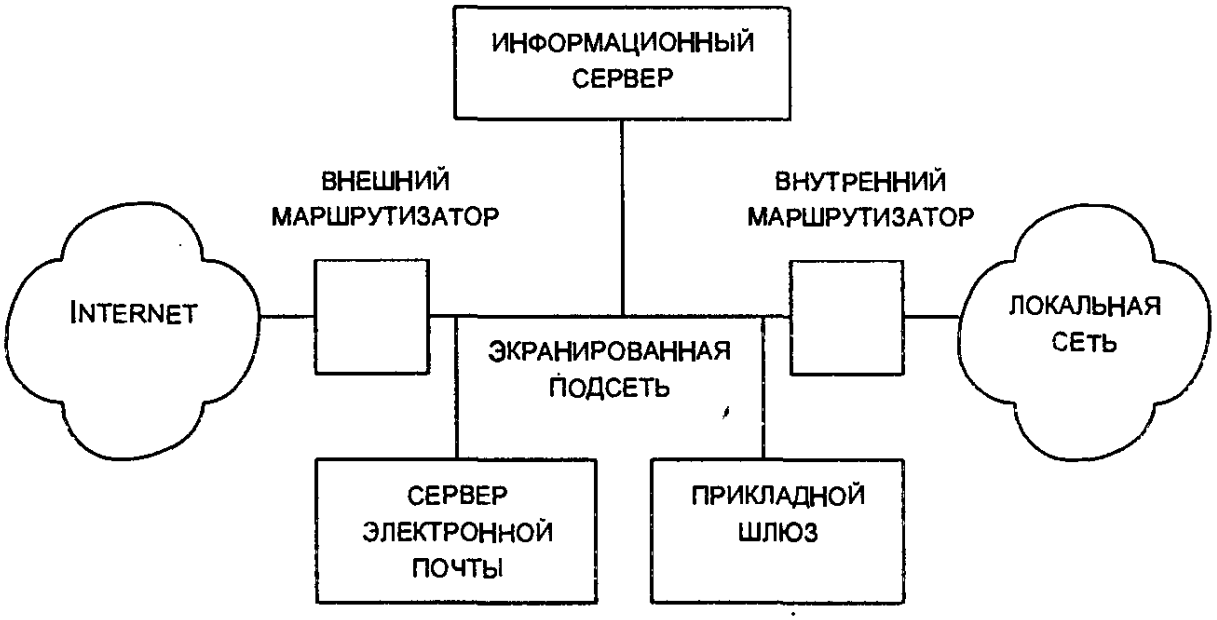


Рис. 8.9. Межсетевой экран – экранированная подсеть

Внешний маршрутизатор защищает от сети Internet как экранированную подсеть, так и внутреннюю сеть. Он должен пересылать трафик согласно следующим правилам:

- разрешается трафик от объектов Internet к прикладному шлюзу;
- разрешается трафик от прикладного шлюза к Internet;
- разрешается трафик электронной почты от Internet к серверу электронной почты;
- разрешается трафик электронной почты от сервера электронной почты к Internet;
- разрешается трафик FTP, Gopher и т.д. от Internet к информационному серверу;
- запрещается остальной трафик.

Внешний маршрутизатор запрещает доступ из Internet к системам внутренней сети и блокирует весь трафик к Internet, идущий от систем, которые не должны являться инициаторами соединений (в частности, информационный сервер и др.). Этот маршрутизатор может быть использован также для блокирования других уязвимых протоколов, которые не должны передаваться к хост-компьютерам внутренней сети или от них [89].

Внутренний маршрутизатор защищает внутреннюю сеть как от Internet, так и от экранированной подсети (в случае ее компрометации). Внутренний маршрутизатор осуществляет большую часть пакетной фильтрации. Он управляет трафиком к системам внутренней сети и от них в соответствии со следующими правилами:

- разрешается трафик от прикладного шлюза к системам сети;
- разрешается прикладной трафик от систем сети к прикладному шлюзу;
- разрешается трафик электронной почты от сервера электронной почты к системам сети;
- разрешается трафик электронной почты от систем сети к серверу электронной почты;
- разрешается трафик FTP, Gopher и т.д. от систем сети к информационному серверу;
- запрещается остальной трафик.

Чтобы проникнуть во внутреннюю сеть при такой схеме межсетевого экрана, атакующему нужно пройти два фильтрующих маршрутизатора. Даже если атакующий каким-то образом проник в хост-компьютер прикладного шлюза, он должен еще преодолеть внутренний фильтрующий маршрутизатор. Таким образом, ни одна система внутренней сети не достижима непосредственно из Internet, и наоборот. Кроме того, четкое разделение функций между маршрутизаторами и прикладным шлюзом позволяет достигнуть более высокой пропускной способности.

Прикладной шлюз может включать программы усиленной аутентификации.

Межсетевой экран с экранированной подсетью хорошо подходит для защиты сетей с большими объемами трафика или с высокими скоростями обмена.

Межсетевой экран с экранированной подсетью имеет и недостатки:

- пара фильтрующих маршрутизаторов нуждается в большом внимании для обеспечения необходимого уровня безопасности, поскольку из-за ошибок при их конфигурировании могут возникнуть провалы в безопасности всей сети;
- существует принципиальная возможность доступа в обход прикладного шлюза.

Применение межсетевых экранов для организации виртуальных корпоративных сетей

Некоторые межсетевые экраны позволяют организовать виртуальные корпоративные сети. Несколько локальных сетей, подключенных к глобальной сети, объединяются в одну виртуальную корпоративную сеть. Схема применения межсетевых экранов в составе виртуальных корпоративных сетей показана на рис. 8.10 [66]. Передача данных между этими локальными сетями производится прозрачным образом для пользователей локальных сетей.

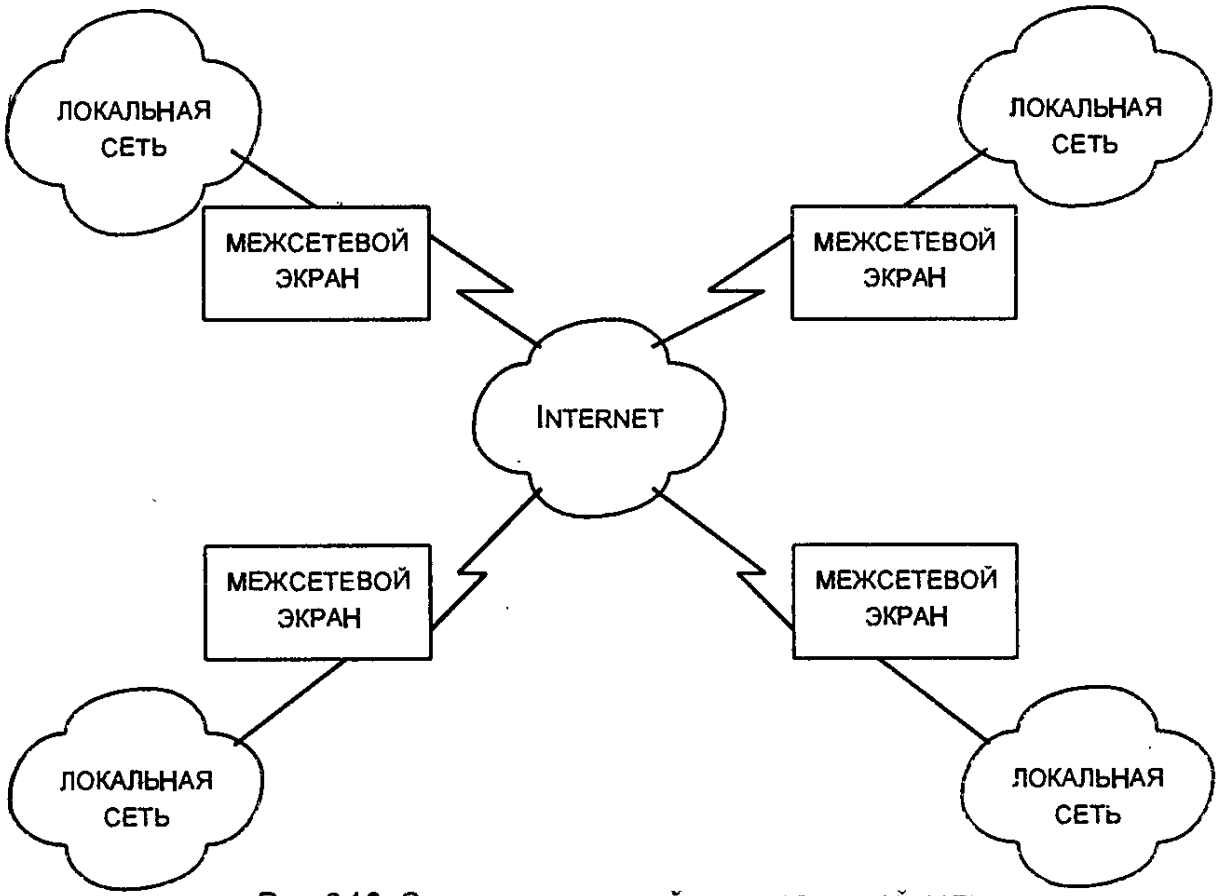


Рис. 8.10. Схема виртуальной корпоративной сети

Конфиденциальность и целостность передаваемой информации должны обеспечиваться при помощи средств шифрования, использования цифровых подписей и т.п. При передаче данных может шифроваться не только содержимое пакета, но и некоторые поля заголовка.

8.4. Программные методы защиты

К программным методам защиты в сети Internet могут быть отнесены защищенные криптопротоколы, которые позволяют надежно защищать соединения [56]. В процессе развития Internet были созданы различные защищенные сетевые протоколы, использующие как симметричную криптографию с закрытым ключом, так и асимметричную криптографию с открытым ключом. К основным на сегодняшний день подходам и протоколам, обеспечивающим защиту соединений, относятся SKIP-технология и протокол защиты соединения SSL.

SKIP (Secure Key Internet Protocol)-технологией называется стандарт защиты трафика IP-пакетов, позволяющий на сетевом уровне обеспечить защиту соединения и передаваемых по нему данных.

Возможны два способа реализации SKIP-защиты трафика IP-пакетов:

- шифрование блока данных IP-пакета;
- инкапсуляция IP-пакета в SKIP-пакет.

Шифрование блока данных IP-пакета иллюстрируется рис.8.11. В этом случае шифруются методом симметричной криптографии только данные IP-пакета, а его заголовок, содержащий помимо прочего адреса отправителя и получателя, остается открытым, и пакет маршрутизируется в соответствии с истинными адресами.

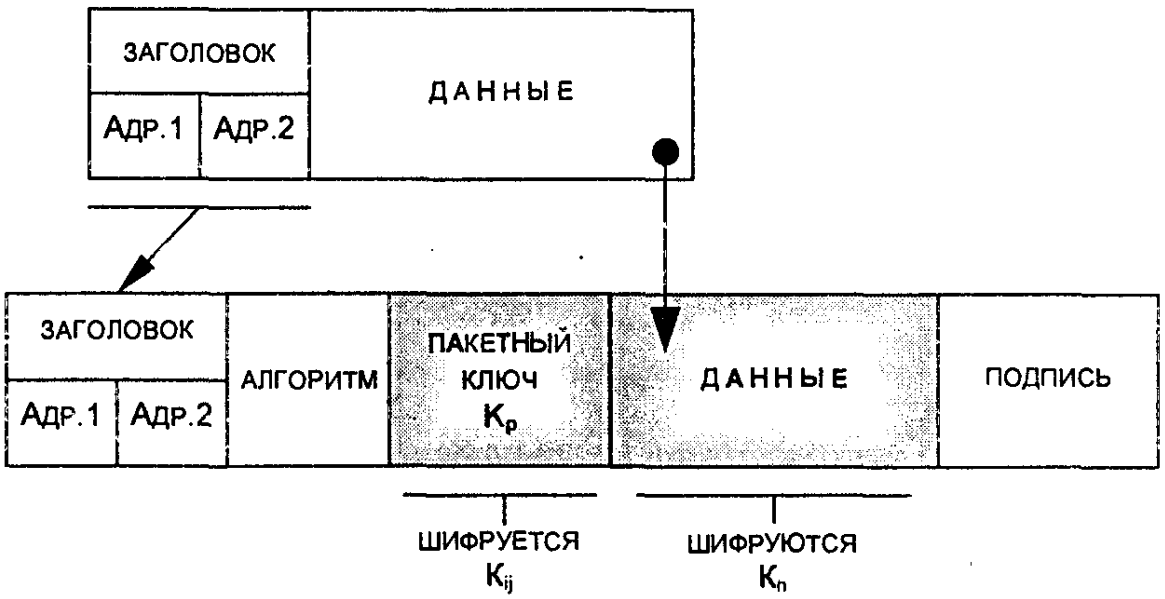


Рис.8.11. Схема шифрования блока данных IP-пакетов

Закрытый ключ K_{ij}, разделяемый парой узлов сети I и J, вычисляется по схеме Диффи–Хеллмана (см. § 7.3).

Инкапсуляция IP-пакета в SKIP-пакет показана на рис. 8.12. SKIP-пакет внешне похож на обычный IP-пакет. В поле данных SKIP-пакета полностью размещается в зашифрованном виде исходный IP-пакет. В этом случае в новом заголовке вместо истинных адресов могут быть помещены некоторые другие адреса. Такая структура SKIP-пакета позволяет беспрепятственно направлять его любому хост-компьютеру в сети Internet, при этом межсетевая адресация осуществляется по обычному IP-заголовку в SKIP-пакете. Конечный получатель SKIP-пакета по заранее определенному разработчиками алгоритму расшифровывает криптограмму и формирует обычный TCP- или UDP-пакет, который и передает соответствующему модулю (TCP или UDP) ядра операционной системы.

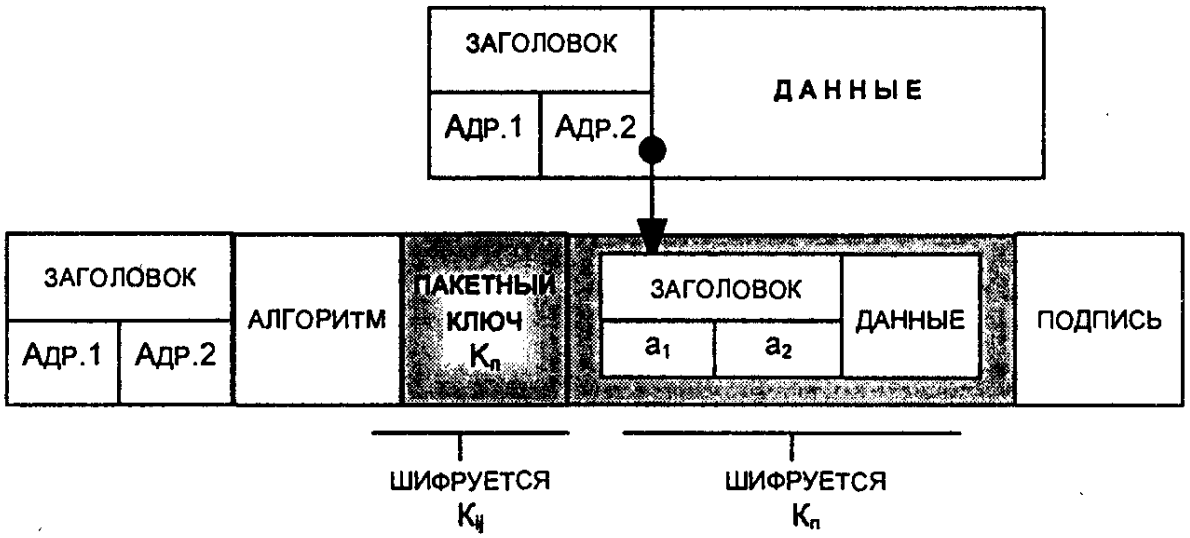


Рис. 8.12. Схема инкапсуляции IP-пакетов

Универсальный протокол защиты соединения SSL (Secure Socket Layer) функционирует на сеансовом уровне эталонной модели OSI. Протокол SSL, разработанный компанией Netscape, использует криптографию с открытым ключом. Этот протокол является действительно универсальным средством, позволяющим динамически защищать соединение при использовании любого прикладного протокола (FTP, TELNET, SMTP, DNS и т.д.). Протокол SSL поддерживают такие ведущие компании, как IBM, Digital Equipment Corporation, Microsoft Corporation, Motorola, Novell Inc., Sun Microsystems, MasterCard International Inc. и др.

ГЛАВА 9. ЗАЩИТА ИНФОРМАЦИИ В ЭЛЕКТРОННЫХ ПЛАТЕЖНЫХ СИСТЕМАХ

Современную практику банковских операций, торговых сделок и взаимных платежей невозможно представить без расчетов с применением пластиковых карт. Благодаря надежности, универсальности и удобству пластиковые карты завоевали прочное место среди других платежных средств и обещают занять лидирующее положение по отношению к наличным платежам уже к 2000 г.

9.1. Принципы функционирования электронных платежных систем

Электронной платежной системой называют совокупность методов и реализующих их субъектов, обеспечивающих в рамках системы использование банковских пластиковых карт в качестве платежного средства [52].

Пластиковая карта – это персонифицированный платежный инструмент, предоставляющий пользующемуся этой картой лицу возможность безналичной оплаты товаров и услуг, а также получения наличных средств в банковских автоматах и отделениях банков. Предприятия торговли и сервиса и отделения банков, принимающие карту в качестве платежного инструмента, образуют *приемную сеть точек обслуживания* карты.

При создании платежной системы одной из основных решаемых задач является выработка и соблюдение общих правил обслуживания карт, выпущенных входящими в платежную систему эмитентами, проведения взаиморасчетов и платежей. Эти правила охватывают как чисто технические аспекты операций с картами – стандарты данных, процедуры авторизации, спецификации на используемое оборудование и другие, так и финансовые аспекты обслуживания карт – процедуры расчетов с предприятиями торговли и сервиса, входящими в состав приемной сети, правила взаиморасчетов между банками и т. д.

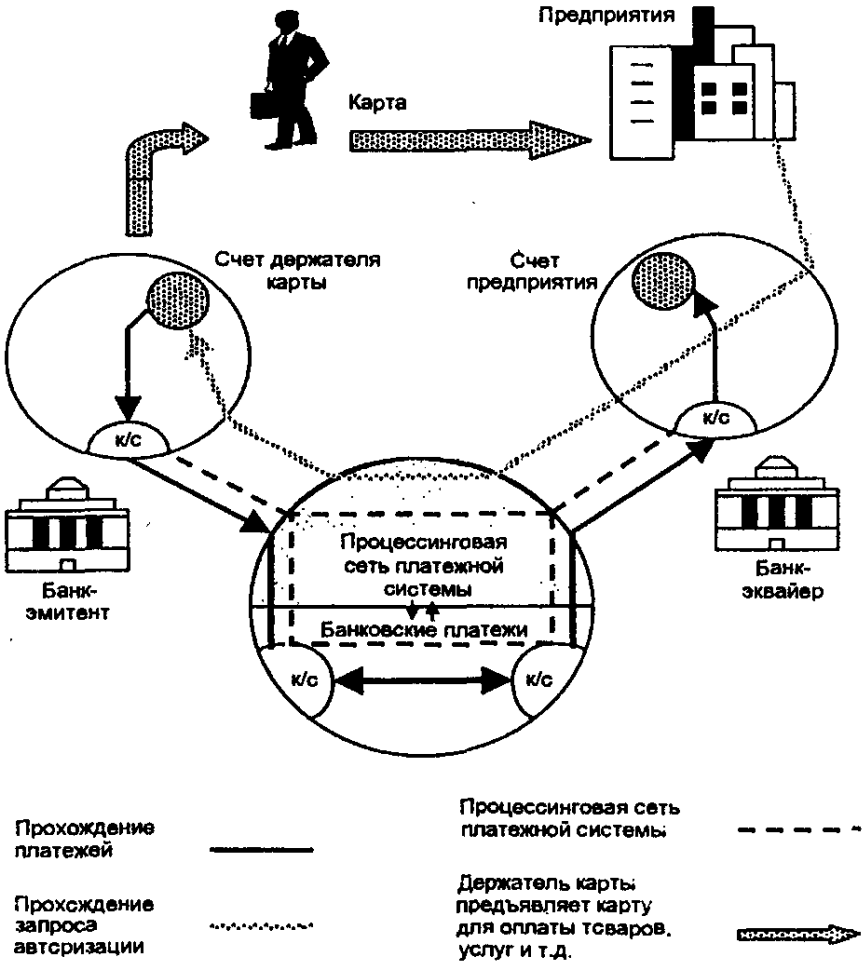


Рис. 9.1. Обобщенная схема функционирования электронных платежных систем

С организационной точки зрения ядром платежной системы является ассоциация банков, объединенная договорными обязательствами. Кроме того, в состав электронной платежной системы входят предприятия торговли и сервиса, образующие сеть точек обслуживания. Для успешного функционирования платежной системы необходимы и специализированные организации, осуществляющие техническую поддержку обслуживания карт: процессинговые и коммуникационные центры, центры технического обслуживания и т. п.

Обобщенная схема функционирования электронной платежной системы представлена на рис. 9.1. Банк, заключивший соглашение с платежной системой и получивший соответствующую лицензию, может выступать в двух качествах – как банк-эмитент и как банк-эквайер. *Банк-эмитент* выпускает пластиковые карты и гарантирует выполнение финансовых обязательств, связанных с использованием этих карт как платежных средств. *Банк-эквайер* обслуживает предприятия торговли и сервиса, принимающие к оплате карты как платежные средства, а также принимает эти платежные

средства к обналичиванию в своих отделениях и через принадлежащие ему банкоматы. Основными неотъемлемыми функциями банка-эквайера являются финансовые операции, связанные с выполнением расчетов и платежей точками обслуживания. Технические атрибуты деятельности банка-эквайера (обработка запросов на авторизацию; перечисление на расчетные счета точек средств за товары и услуги, предоставленные по картам; прием, сортировка и пересылка документов, фиксирующих совершение сделок с использованием карт и т. п.) могут быть делегированы эквайером процессинговым центрам.

Неавтоматизированная процедура приема платежа с помощью карты сравнительно проста [61]. В первую очередь кассир предприятия должен убедиться в подлинности пластиковой карты по ряду признаков, указанных в § 9.2. При оплате предприятие должно перенести реквизиты пластиковой карты клиента на специальный чек с помощью копировальной машины-импринтера, занести в чек сумму, на которую была совершена покупка или оказана услуга, и получить подпись клиента. Оформленный подобным образом чек называют *слипом*.

В целях обеспечения безопасности операций платежной системы рекомендуется не превышать нижние лимиты сумм для различных регионов и видов бизнеса, по которым можно проводить расчеты без авторизации. При превышении лимитной суммы или в случае возникновения сомнения в личности клиента предприятие должно проводить *процедуру авторизации*. При авторизации предприятие фактически получает доступ к информации о состоянии счета клиента и может установить принадлежность карты клиенту и его платежную способность в размере суммы сделки. Одна копия слипа остается на предприятии, вторая передается клиенту, третья доставляется в банк-эквайер и служит основанием для возмещения суммы платежа предприятию со счета клиента.

В последние годы широкую популярность приобрели *автоматизированные торговые POS-терминалы* (Point-Of-Sale – оплата в точке продажи) и банкоматы. При использовании POS-терминалов нет необходимости в заполнении слипов. Реквизиты пластиковой карты считываются с ее магнитной полосы на встроенном в POS-терминал считывателе. Клиент вводит в терминал свой PIN-код (Personal Identification Number – персональный идентификационный номер), известный только ему. Элементы PIN-кода включаются в общий алгоритм шифрования записи на магнитной полосе и служат электронной подписью владельца карты. На клавиатуре POS-терминала набирается сумма сделки.

Если сделка осуществляется в отделении банка и в ее процессе происходит выдача клиенту наличных денег, помимо банковских POS-терминалов может быть использован *электронный кассир-банкомат.* Конструктивно он представляет автоматизированный сейф со встроенным POS-терминалом.

Терминал через встроенный модем обращается за авторизацией в соответствующую платежную систему. При этом используются мощности процессингового центра, услуги которого предоставляются торговцу банком-эквайером.

Процессинговый центр [33] представляет собой специализированную сервисную организацию, которая обеспечивает обработку поступающих от банков-эквайеров или непосредственно из точек обслуживания запросов на авторизацию и протоколов транзакций – фиксируемых данных о произведенных посредством пластиковых карт платежах и выдачах наличными. Для этого процессинговый центр ведет базу данных, которая, в частности, содержит данные о банках-членах платежной системы и держателях пластиковых карт. Процессинговый центр хранит сведения о лимитах держателей карт и выполняет запросы на авторизацию в том случае, если банк-эмитент не ведет собственной базы данных (off-line банк). В противном случае (on-line банк) процессинговый центр пересылает полученный запрос в банк-эмитент авторизируемой карты. Очевидно, что процессинговый центр обеспечивает и пересылку ответа банку-эквайеру.

Выполнение банком-эквайером своих функций влечет за собой расчеты с банками-эмитентами. Каждый банк-эквайер осуществляет перечисление средств точкам обслуживания по платежам держателей карт банков-эмитентов, входящих в данную платежную систему. Поэтому соответствующие средства должны быть затем перечислены банку-эквайеру банками-эмитентами. Оперативное проведение взаиморасчетов между эквайерами и эмитентами обеспечивается наличием в платежной системе *расчетного банка* (одного или нескольких), в котором банки-члены системы открывают корреспондентские счета. На основании накопленных за операционный день протоколов транзакций процессинговый центр готовит и рассылает итоговые данные для проведения взаиморасчетов между банками-участниками платежной системы, а также формирует и рассылает банкам-эквайерам и непосредственно в точки обслуживания стоп-листы (перечни карточек, операции по которым по разным причинам приостановлены).

Процессинговый центр может также обеспечивать потребности банков-эмитентов в новых картах, осуществляя их заказ на заводах и последующую персонализацию.

Особенностью продаж и выдач наличных по пластиковым картам является то, что эти операции осуществляются магазинами и

банками "в долг", т.е. товары и наличные предоставляются клиентам сразу, а средства на их возмещение поступают на счета обслуживающих предприятий через некоторое время (не более нескольких дней). Гарантом выполнения платежных обязательств, возникающих в процессе обслуживания пластиковых карт, является выпустивший их банк-эмитент. Характер гарантий банка-эмитента зависит от платежных полномочий, предоставляемых клиенту и фиксируемых видом карточки.

По виду расчетов, выполняемых с помощью пластиковых карт, различают кредитные и дебетовые карты.

Кредитные карты являются наиболее распространенным видом пластиковых карт. К ним относятся карты общенациональных систем США Visa и MasterCard, American Express и ряда других. Эти карты предъявляют на предприятиях торговли и сервиса для оплаты товаров и услуг. При оплате с помощью кредитных карт банк покупателя открывает ему кредит на сумму покупки, а затем через некоторое время (обычно 25 дней) присылает счет по почте. Покупатель должен вернуть оплаченный чек (счет) обратно в банк. Естественно, подобную схему банк может предложить только наиболее состоятельным и проверенным из своих клиентов, которые имеют хорошую кредитную историю перед банком или солидные вложения в банк в виде депозитов, ценностей или недвижимости.

Держатель *дебетовой карты* должен заранее внести на свой счет в банке-эмитенте определенную сумму. Размер этой суммы определяет лимит доступных средств. При осуществлении расчетов с использованием этой карты соответственно уменьшается и лимит. Контроль лимита выполняется при проведении авторизации, которая при использовании дебетовой карты является обязательной. Для возобновления или увеличения лимита держателю карты необходимо вновь внести средства на свой счет. Для страхования временного разрыва между моментом осуществления платежа и моментом получения банком соответствующей информации на счете клиента должен поддерживаться неснижаемый остаток.

Как кредитная, так и дебетовая карты могут быть не только персональными, но и корпоративными. *Корпоративные карты* предоставляются компанией своим сотрудникам для оплаты командировочных или других служебных расходов. Корпоративные карты компании связаны с каким-либо одним ее счетом. Эти карты могут иметь разделенный или неразделенный лимит. В первом случае каждому из держателей корпоративных карт устанавливается индивидуальный лимит. Второй вариант больше подходит небольшим компаниям и не предполагает разграничения лимита.

В последние годы все большее внимание привлекают к себе электронные платежные системы с использованием микропроцессорных карт. Принципиальным отличием микропроцессорных карт

от всех перечисленных выше является то, что они непосредственно несут информацию о состоянии счета клиента, поскольку являются в сущности транзитным счетом. Все транзакции совершаются в режиме off-line в процессе диалога карта – терминал или карта клиента – карта торговца.

Такая система является почти полностью безопасной благодаря высокой степени защищенности кристалла с микропроцессором и полной дебетовой схеме расчетов. Кроме того, хотя карта с микропроцессором дороже обычной, платежная система оказывается дешевле в эксплуатации за счет того, что в режиме off-line нет нагрузки на телекоммуникации.

Для обеспечения надежной работы электронная платежная система должна быть надежно защищена.

С точки зрения информационной безопасности в системах электронных платежей существуют следующие уязвимые места:

- пересылка платежных и других сообщений между банком и клиентом и между банками;
- обработка информации внутри организаций отправителя и получателя сообщений;
- доступ клиентов к средствам, аккумулированным на счетах.

Одним из наиболее уязвимых мест в системе электронных платежей является пересылка платежных и других сообщений между банками, между банком и банкоматом, между банком и клиентом. Пересылка платежных и других сообщений связана со следующими особенностями [22]:

- внутренние системы организаций отправителя и получателя должны быть приспособлены для отправки и получения электронных документов и обеспечивать необходимую защиту при их обработке внутри организации (защита оконечных систем);
- взаимодействие отправителя и получателя электронного документа осуществляется опосредовано – через канал связи.

Эти особенности порождают следующие проблемы:

- взаимное опознавание абонентов (проблема установления взаимной подлинности при установлении соединения);
- защита электронных документов, передаваемых по каналам связи (проблемы обеспечения конфиденциальности и целостности документов);
- защита процесса обмена электронными документами (проблема доказательства отправления и доставки документа);
- обеспечение исполнения документа (проблема взаимного недоверия между отправителем и получателем из-за их принадлежности к разным организациям и взаимной независимости).

Для обеспечения функций защиты информации на отдельных узлах системы электронных платежей должны быть реализованы следующие механизмы защиты:

- управление доступом на оконечных системах;
- контроль целостности сообщения;
- обеспечение конфиденциальности сообщения;
- взаимная аутентификация абонентов;
- невозможность отказа от авторства сообщения;
- гарантии доставки сообщения;
- невозможность отказа от принятия мер по сообщению;
- регистрация последовательности сообщений;
- контроль целостности последовательности сообщений.

Качество решения указанных выше проблем в значительной мере определяется рациональным выбором криптографических средств при реализации механизмов защиты.

9.2. Электронные пластиковые карты

Применение POS-терминалов и банкоматов возможно при использовании некоторого носителя информации, который мог бы идентифицировать пользователя и хранить определенные учетные данные. В качестве такого носителя информации выступают пластиковые карты.

Пластиковая карта представляет собой пластину стандартных размеров (85,6×53,9×0,76 мм), изготовленную из специальной, устойчивой к механическим и термическим воздействиям пластмассы. Одна из основных функций пластиковой карты – обеспечение идентификации использующего ее лица как субъекта платежной системы. Для этого на пластиковую карту наносят логотипы банка-эмитента и платежной системы, обслуживающей эту карту, имя держателя карты, номер его счета, срок действия карты и т. п. Кроме того, на карте может присутствовать фотография держателя и его подпись. Алфавитно-цифровые данные – имя, номер счета и др. – могут быть эмбоссированы, т. е. нанесены рельефным шрифтом. Это дает возможность при ручной обработке принимаемых к оплате карт быстро перенести данные на чек с помощью специального устройства – импринтера, осуществляющего "прокатывание" карты (аналогично получению второго экземпляра при использовании копировальной бумаги).

По принципу действия различают пассивные и активные пластиковые карты. Пассивные пластиковые карты всего лишь хранят информацию на том или ином носителе. К ним относятся пластиковые карты с магнитной полосой.

Карты с магнитной полосой являются на сегодняшний день наиболее распространенными – в обращении находится свыше двух миллиардов карт подобного типа. Магнитная полоса располагается на обратной стороне карты и, в соответствии со стандартом ISO 7811, состоит из трех дорожек. Из них первые две предназначены для хранения идентификационных данных, а на третью дорожку можно записывать информацию (например, текущее значение лимита дебетовой карты). Однако из-за невысокой надежности многократно повторяемого процесса записи и считывания запись на магнитную полосу обычно не практикуется, и такие карты используются только в режиме считывания информации.

Карты с магнитной полосой относительно уязвимы для мошенничества. Например, в США в 1992 г. общий ущерб от махинаций с кредитными картами с магнитной полосой (без учета потерь с банкоматами) превысил один миллиард долларов. Тем не менее развитая инфраструктура существующих платежных систем и, в частности, мировых лидеров в области "карточного" бизнеса – компаний Visa и MasterCard/Europay является причиной интенсивного использования карт с магнитной полосой и сегодня.

Для повышения защищенности своих карт системы Visa и MasterCard/Europay используют дополнительные графические средства защиты: голограммы и нестандартные шрифты для эмбоссирования.

Платежные системы с подобными картами требуют on-line авторизации в торговых точках и, как следствие, наличия разветвленных, высококачественных средств коммуникации (телефонных линий). Поэтому с технической точки зрения подобные системы имеют серьезные ограничения по их применению в странах с плохо развитыми системами связи.

Отличительная особенность активных пластиковых карт – наличие встроенной в нее электронной микросхемы. Принцип пластиковой карты с электронной микросхемой запатентовал в 1974 г. француз Ролан Морено. Стандарт ISO 7816 определяет основные требования к картам на интегральных микросхемах или чиповым картам. В недалеком будущем карты с микросхемой вытеснят карты с магнитной полосой. Поэтому остановимся более подробно на основных типах карт с микросхемой.

Карты с микросхемой можно классифицировать по нескольким признакам [61].

Первый признак – функциональные возможности карты. Здесь можно выделить следующие основные типы карт:

- карты-счетчики;
- карты с памятью;
- карты с микропроцессором.

Второй признак – тип обмена со считывающим устройством:

- карты с контактным считыванием;
- карты с индукционным считыванием.

Карты-счетчики применяются, как правило, в тех случаях, когда та или иная платежная операция требует уменьшения остатка на счете держателя карты на некоторую фиксированную сумму. Подобные карты используются в специализированных приложениях с предоплатой (плата за использование телефона-автомата, оплата автостоянки и т.д.). Очевидно, что применение карт со счетчиком ограничено и не имеет большой перспективы.

Карты с памятью являются переходными между картами со счетчиком и картами с процессором. Карта с памятью – это в сущности перезаписываемая карта со счетчиком, в которой приняты меры, повышающие ее защищенность от атак злоумышленников. У простейших из существующих карт с памятью объем памяти может составлять от 32 байт до 16 килобайт. Эта память может быть реализована или в виде программируемого постоянного запоминающего устройства ППЗУ (EPROM), которое допускает однократную запись и многократное считывание, или в виде электрически стираемого программируемого постоянного запоминающего устройства ЭСППЗУ (EEPROM), допускающего многократную запись и многократное считывание.

Карты с памятью можно подразделить на два типа: с незащищенной (полнодоступной) и защищенной памятью.

В картах первого типа нет никаких ограничений на чтение и запись данных. Их нельзя использовать в качестве платежных, так как специалист средней квалификации может их достаточно просто "взломать".

Карты второго типа имеют область идентификационных данных и одну или несколько прикладных областей. Идентификационная область карт допускает лишь однократную запись при персонализации и в дальнейшем доступна лишь для считывания. Доступ к прикладным областям регламентируется и осуществляется только при выполнении определенных операций, в частности при вводе секретного PIN-кода.

Уровень защиты карт с памятью выше, чем у магнитных карт, и они могут быть использованы в прикладных системах, в которых финансовые риски, связанные с мошенничеством, относительно невелики. В качестве платежного средства карты с памятью используются для оплаты таксофонов общего пользования, проезда в транспорте, в локальных платежных системах (клубные карты). Карты с памятью применяются также в системах допуска в помещения и доступа к ресурсам компьютерных сетей (идентификационные карты). Карты с памятью имеют более низкую стоимость по сравнению с картами с микропроцессором.

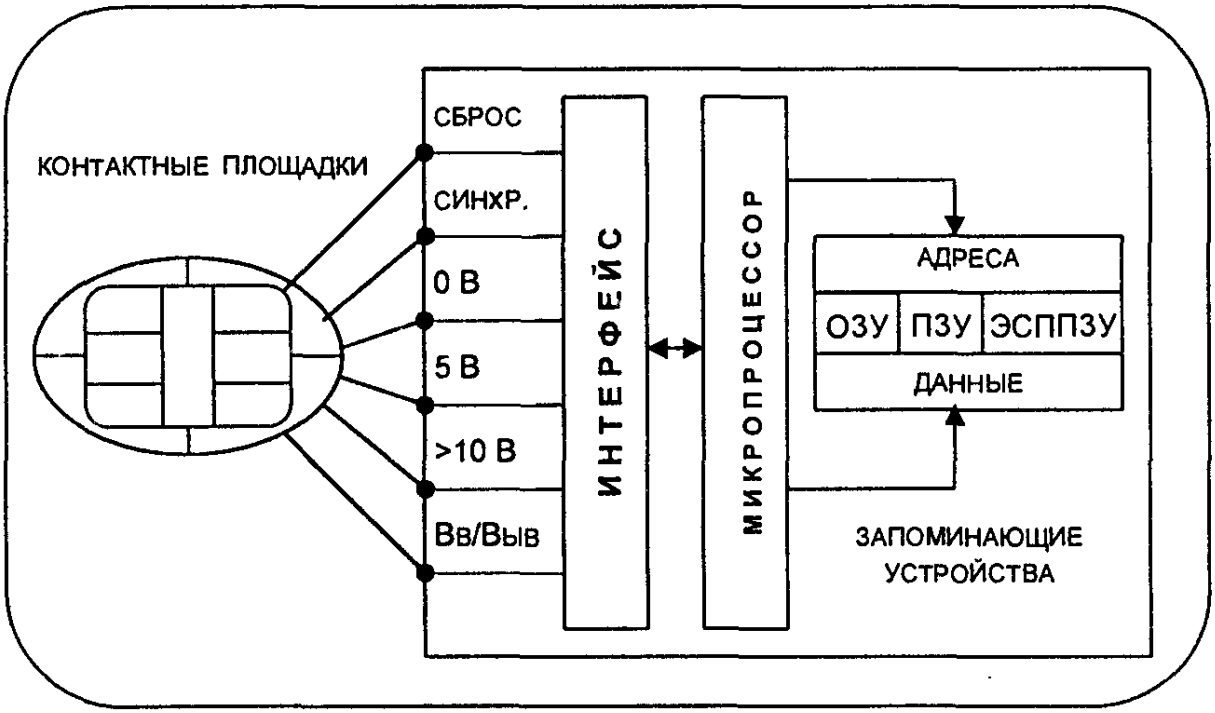


Рис. 9.2. Архитектура смарт-карты

Карты с микропроцессором называют также интеллектуальными картами или смарт-картами (smart cards). Карты с микропроцессором представляют собой по сути микрокомпьютеры и содержат все соответствующие основные аппаратные компоненты: центральный процессор (ЦП), оперативное запоминающее устройство (ОЗУ), постоянное запоминающее устройство (ПЗУ) и электрически стираемое программируемое ПЗУ (ЭСППЗУ) (рис. 9.2).

В настоящее время в смарт-карты устанавливают:

- микропроцессоры с текстовой частотой 5 Мгц;
- оперативное ЗУ емкостью до 256 байт;
- постоянное ЗУ емкостью до 10 Кбайт;
- энергонезависимое ЗУ емкостью до 8 Кбайт.

В ПЗУ записан специальный набор программ, называемый операционной системой карты COS (Card Operation System). Операционная система поддерживает файловую систему, базирующуюся в ЭСППЗУ (емкость которого обычно находится в диапазоне 1...8 Кбайт, но может достигать и 64 Кбайт) и обеспечивающую регламентацию доступа к данным. При этом часть данных может быть доступна только внутренним программам карточки.

Смарт-карта обеспечивает обширный набор функций:

- разграничение полномочий доступа к внутренним ресурсам (благодаря работе с защищенной файловой системой);
- шифрование данных с применением различных алгоритмов;
- формирование электронной цифровой подписи;
- ведение ключевой системы;
- выполнение всех операций взаимодействия владельца карты, банка и торговца.

Некоторые карты обеспечивают режим "самоблокировки" (невозможность дальнейшей работы с ней) при попытке несанкционированного доступа. Смарт-карты позволяют существенно упростить процедуру идентификации клиента. Для проверки PIN-кода применяется алгоритм, реализуемый микропроцессором на карте. Это позволяет отказаться от работы POS-терминала и банкомата в режиме реального времени и централизованной проверки PIN. Отмеченные выше особенности делают смарт-карту высокозащищенным платежным инструментом, который может быть использован в финансовых приложениях, предъявляющих повышенные требования к защите информации. Именно поэтому микропроцессорные смарт-карты рассматриваются в настоящее время как наиболее перспективный вид пластиковых карт.

По принципу взаимодействия со считывающим устройством различают карты двух типов:

- карты с контактным считыванием;
- карты с бесконтактным считыванием.

Карта с контактным считыванием имеет на своей поверхности 8...10 контактных пластин. Размещение контактных пластин, их количество и назначение выводов различны у разных производителей и естественно, что считыватели для карт данного типа различаются между собой.

В последние годы начали широко применяться *карты с бесконтактным считыванием.* В них обмен данными между картой и считывающим устройством производится индукционным способом. Очевидно, что такие карты надежнее и долговечнее.

Персонализация и авторизация карт являются важными этапами подготовки и применения пластиковых карт.

Персонализация карты осуществляется при выдаче карты клиенту. При этом на карту заносятся данные, позволяющие идентифицировать карту и ее держателя, а также осуществить проверку платежеспособности карты при приеме ее к оплате или выдаче наличных денег.

Под *авторизацией* понимают процесс утверждения продажи или выдачи наличных по карте. Для проведения авторизации точка обслуживания делает запрос платежной системе о подтверждении полномочий предъявителя карты и его финансовых возможностей. Технология авторизации зависит от типа карты, схемы платежной системы и технической оснащенности точки обслуживания.

Исторически сложилось так, что первоначальным способом персонализации карт было эмбоссирование.

Эмбоссирование – это процесс рельефного тиснения данных на пластиковой основе карты. На картах банков-эмитентов эмбоссируются, как правило, следующие данные: номер карты; даты начала и окончания срока ее действия; фамилия и имя владельца.

Некоторые платежные системы, например Visa, требуют тиснения двух специальных символов, однозначно идентифицирующих принадлежность банка-эмитента к платежной системе. Эмбоссеры (устройства для тиснения рельефа на карте) выпускает ограниченный круг изготовителей. В ряде стран Запада законодательно запрещена свободная продажа эмбоссеров. Специальные символы, подтверждающие принадлежность карты к той или иной платежной системе, поставляются владельцу эмбоссера только с разрешения руководящего органа платежной системы. Эмбоссированная карта может служить средством платежа при использовании импринтера – устройства для прокатки слипа (чека), подтверждающего совершенную платежную операцию.

К персонализации карт относится также кодирование магнитной полосы либо программирование микросхемы.

Кодирование магнитной полосы производится, как правило, на том же оборудовании, что и эмбоссирование. При этом часть информации о карте, содержащая номер карты и период ее действия, одинаковая как на магнитной полосе, так и на рельефе. Однако бывают ситуации, когда после первичного кодирования требуется дополнительно занести информацию на магнитную дорожку. В этом случае применяются специальные устройства с функцией "чтение-запись". Это возможно, в частности, когда PIN-код для пользования картой не формируется специальной программой, а может быть выбран клиентом по своему усмотрению.

Программирование микросхемы не требует особых технологических приемов, но зато оно имеет некоторые организационные особенности. В частности, для повышения безопасности и исключения возможных злоупотреблений операции по программированию различных областей микросхемы разнесены территориально и разграничены по правам различных сотрудников, участвующих в этом процессе.

Обычно эта процедура разбивается на три этапа:

- на первом рабочем месте выполняется активация карты (ввод ее в действие);
- на втором рабочем месте выполняются операции, связанные с обеспечением безопасности;
- на третьем рабочем месте производится собственно персонализация карты.

Традиционно *процесс авторизации* проводится либо "вручную", когда продавец или кассир передает запрос по телефону оператору (голосовая авторизация), либо автоматически, когда карта помещается в POS-терминал, данные считываются с карты, кассиром вводится сумма платежа, а владельцем карты со специальной клавиатуры – секретный PIN-код. После этого терминал осуществляет авторизацию, либо устанавливая связь с базой дан-

ных платежной системы (on-line режим), либо реализуя дополнительный обмен данными с самой картой (off-line авторизация). В случае выдачи наличных денег процесс носит аналогичный характер, с той лишь особенностью, что деньги в автоматическом режиме выдаются специальным устройством – банкоматом, который и проводит авторизацию.

Для защиты карт от подделки и последующего несанкционированного применения используются различные методы и способы. Например, для персонализации карт может применяться нанесение на пластиковую основу черно-белой или цветной фотографии владельца карты методом термопечати. На любой карте всегда существует специальная полоска с образцом подписи владельца карты. Для защиты карты, как таковой, различные платежные сообщества применяют специальные объемные изображения на лицевой и оборотной стороне карты (голограммы).

9.3. Персональный идентификационный номер

Испытанным способом идентификации держателя банковской карты является использование секретного персонального идентификационного номера PIN. Значение PIN должно быть известно только держателю карты. Длина PIN должна быть достаточно большой, чтобы вероятность угадывания злоумышленником правильного значения с помощью атаки полного перебора значений была приемлемо малой. С другой стороны, длина PIN должна быть достаточно короткой, чтобы дать возможность держателям карт запомнить его значение. Рекомендуемая длина PIN составляет 4...8 десятичных цифр, но может достигать 12.

Предположим, что PIN имеет длину четыре цифры, тогда противник, пытающийся подобрать значение PIN к банковской карте, стоит перед проблемой выбора одной из десяти тысяч возможностей. Если число попыток ввода некорректного значения PIN ограничивается пятью на карту в день, этот противник имеет шансы на успех менее чем 1:2000. Но на следующий день противник может попытаться снова, и его шансы увеличиваются до 1:1000. Каждый следующий день увеличивает вероятность успеха противника. Поэтому многие банки вводят абсолютный предел на число неверных попыток ввода PIN на карту, чтобы исключить атаку такого рода. Если установленный предел превышен, считается, что данная карта неправильная, и ее отбирают.

Значение PIN однозначно связано с соответствующими атрибутами банковской карты, поэтому PIN можно трактовать как подпись держателя карточки. Чтобы инициировать транзакцию, держатель карты, который использует POS-терминал, вставляет свою

карту в специальную щель считывателя и вводит свой PIN, используя специальную клавиатуру терминала. Если введенное значение PIN и номер счета клиента, записанный на магнитной полосе карты, согласуются между собой, тогда инициируется транзакция.

Защита персонального идентификационного номера PIN для банковской карты является критичной для безопасности всей платежной системы. Банковские карты могут быть потеряны, украдены или подделаны. В таких случаях единственной контрмерой против несанкционированного доступа остается секретное значение PIN. Вот почему открытая форма PIN должна быть известна только законному владельцу карты. Она никогда не хранится и не передается в рамках системы электронных платежей. Очевидно, значение PIN нужно держать в секрете в течение всего срока действия карты.

Метод генерации значения PIN оказывает существенное влияние на безопасность электронной платежной системы. Вообще, персональные идентификационные номера могут формироваться либо банком, либо держателями карт. В частности, клиент различает два типа PIN [22]:

- PIN, назначенный ему банком, выдавшим карту;
- PIN, выбираемый держателем карты самостоятельно.

Если PIN назначается банком, банк обычно использует один из двух вариантов процедур генерации PIN.

При первом варианте PIN генерируется криптографически из номера счета держателя карточки. Процесс генерации назначаемого PIN из номера счета показан на рис.9.3. Сначала номер счета клиента дополняется нулями или другой константой до 16 шестнадцатеричных цифр (8 байт). Затем получившиеся 8 байт шифруются по алгоритму DES с использованием секретного ключа. Из полученного шифртекста длиной 8 байт поочередно выделяют 4-битовые блоки, начиная с младшего байта. Если число, образуемое этими битами, меньше 10, то полученная цифра включается в PIN, иначе это значение не используется. Таким путем обрабатывают все 64 бита (8 байт). Если в результате обработки не удалось получить сразу требуемое количество десятичных цифр, то обращаются к неиспользованным 4-битовым блокам, из которых вычитают 10.

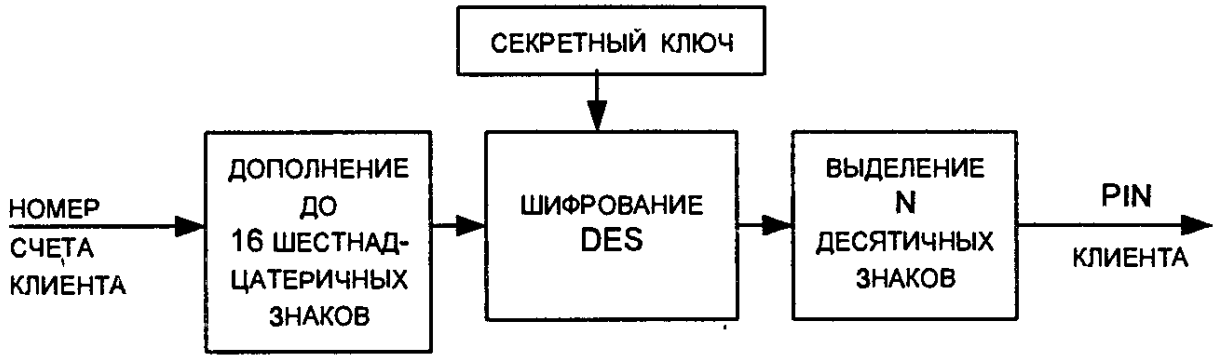


Рис. 9.3. Схема выведения PIN из номера счетв клиентв

Очевидное достоинство этой процедуры заключается в том, что значение PIN не нужно хранить внутри электронной платежной системы. Недостатком этого подхода является то, что при необходимости изменения PIN требуется выбор либо нового счета клиента, либо нового криптографического ключа. Банки предпочитают, чтобы номер счета клиента оставался фиксированным. С другой стороны, поскольку все PIN вычисляют, используя одинаковый криптографический ключ, изменение одного PIN при сохранении счета клиента неизбежно влечет за собой изменение всех персональных идентификационных номеров.

При втором варианте банк выбирает значение PIN случайным образом, сохраняя значение этого PIN в виде соответствующей криптограммы. Выбранные значения PIN банк передает держателям банковских карт, пользуясь защищенным каналом.

Использование PIN, назначенного банком, неудобно для клиента даже при небольшой его длине. Такой PIN трудно удержать в памяти, и поэтому держатель карты может записать его куда-нибудь. Главное – это не записать PIN непосредственно на карту или какое-нибудь другое видное место. Иначе задача злоумышленника будет сильно облегчена.

Для большего удобства клиента используют значение PIN, выбираемое самим клиентом. Такой способ определения значения PIN позволяет клиенту:

- использовать один и тот же PIN для различных целей;
- задавать PIN как совокупность букв и цифр (для удобства запоминания).

Когда PIN выбран клиентом, он должен быть доведен до сведения банка. PIN может быть передан в банк заказной почтой или отправлен через защищенный терминал, размещенный в банковском офисе, который немедленно его шифрует. Если банку необходимо использовать выбранный клиентом PIN, тогда поступают следующим образом. Каждую цифру выбранного клиентом PIN складывают по модулю 10 (без учета переносов) с соответствующей цифрой PIN, выводимого банком из счета клиента. Получаемое десятичное число называется "смещением". Это смещение запоминается на карте клиента. Поскольку выводимый PIN имеет случайный характер, то выбранный клиентом PIN невозможно определить по его "смещению".

Главное требование безопасности состоит в том, что значение PIN должно запоминаться владельцем карты и никогда не должно храниться в любой читабельной форме. Но люди несовершенны и очень часто забывают свои значения PIN. Поэтому банки должны заранее заготовить специальные процедуры для таких случаев. Банк может реализовать один из следующих подходов. Первый ос-

нован на восстановлении забытого клиентом значения PIN и отправке его обратно владельцу карты. При втором подходе просто генерируется новое значение PIN.

При *идентификации клиента по значению PIN* и предъявленной карте используются два основных способа проверки PIN: неалгоритмический и алгоритмический [22].

Неалгоритмический способ проверки PIN не требует применения специальных алгоритмов. Проверка PIN осуществляется путем непосредственного сравнения введенного клиентом PIN со значениями, хранимыми в базе данных. Обычно база данных со значениями PIN клиентов шифруется методом прозрачного шифрования, чтобы повысить ее защищенность, не усложняя процесса сравнения.

Алгоритмический способ проверки PIN заключается в том, что введенный клиентом PIN преобразуют по определенному алгоритму с использованием секретного ключа и затем сравнивают со значением PIN, хранящимся в определенной форме на карте. Достоинства этого метода проверки:

- отсутствие копии PIN на главном компьютере исключает его раскрытие персоналом банка;
- отсутствие передачи PIN между банкоматом или POS-терминалом и главным компьютером банка исключает его перехват злоумышленником или навязывание результатов сравнения;
- упрощение работы по созданию программного обеспечения системы, так как уже нет необходимости действий в реальном масштабе времени.

9.4. Обеспечение безопасности систем POS

Системы POS (Point-Of-Sale), обеспечивающие расчеты продавца и покупателя *в точке продажи*, получили широкое распространение в развитых странах и, в частности, в США. Системы POS осуществляют проверку и обслуживание дебетовых и кредитных карт покупателя непосредственно в местах продажи товаров и услуг в рамках системы электронных платежей. POS-терминалы, входящие в эти системы, размещаются на различных предприятиях торговли – в супермаркетах, на автозаправочных станциях и т. п.

POS-терминалы предназначены для обработки транзакций при финансовых расчетах с использованием пластиковых карт с магнитной полосой и смарт-карт. Использование POS-терминалов позволяет автоматизировать операции по обслуживанию этих карт и существенно уменьшить время обслуживания. Возможности и комплектация POS-терминалов варьируются в широких пределах, однако типичный современный POS-терминал снабжен устройствами

считывания как с карт с магнитной полосой, так и со смарт-карт; энергонезависимой памятью; портами для подключения PIN-клавиатуры (клавиатуры для набора клиентом PIN-кода); принтера; соединения с персональным компьютером или электронным кассовым аппаратом.

Обычно POS-терминал бывает также оснащен модемом с возможностью автодозвона. POS-терминал обладает "интеллектуальными" возможностями – его можно программировать. В качестве языков программирования используются язык ассемблера, а также диалекты языков Си и Бейсик. Все это позволяет проводить авторизацию карт с магнитной полосой в режиме реального времени (on-line) и использовать при работе со смарт-картами автономный режим (off-line) с накоплением протоколов транзакций. Эти протоколы транзакций передаются в процессинговый центр во время сеансов связи. Во время этих сеансов POS-терминал может также принимать и запоминать информацию, передаваемую ЭВМ процессингового центра. В основном это бывают стоп-листы.

Стоимость POS-терминалов в зависимости от комплектации и возможностей может меняться от нескольких сотен до нескольких тысяч долларов, хотя обычно не превышает полутора–двух тысяч долларов. Размеры и вес POS-терминала сопоставимы с аналогичными параметрами телефонного аппарата.

Схема системы POS приведена на рис. 9.4. Покупатель для оплаты покупки предъявляет свою дебетовую или кредитную карту и вводит значение PIN для подтверждения личности. Продавец, в свою очередь, вводит сумму денег, которую необходимо уплатить

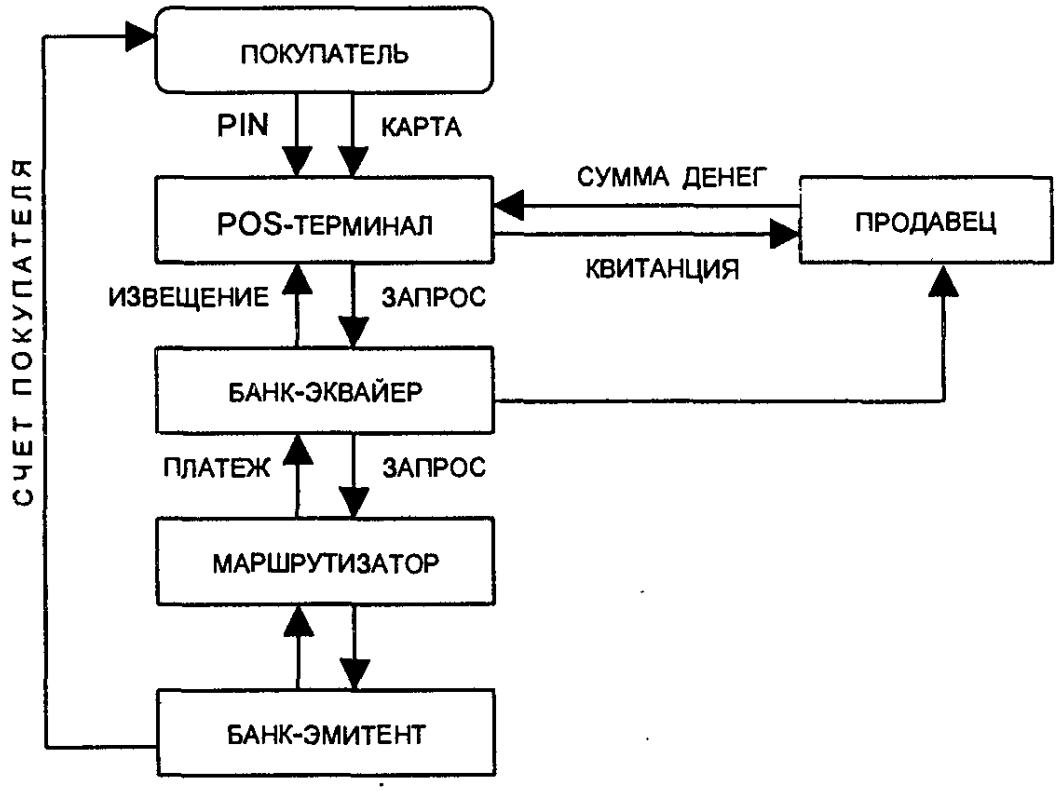


Рис. 9.4. Схема функционирования системы POS

за покупку или услуги. Затем в банк-эквайер (банк продавца) направляется запрос на перевод денег. Банк-эквайер переадресует этот запрос в банк-эмитент для проверки подлинности карты, предъявленной покупателем. Если эта карта подлинная и покупатель имеет право применять ее для оплаты продуктов и услуг, банк-эмитент переводит деньги в банк-эквайер на счет продавца. После перевода денег на счет продавца банк-эквайер посылает на POS-терминал извещение, в котором сообщает о завершении транзакции. После этого продавец выдает покупателю товар и извещение.

Следует обратить внимание на тот сложный путь, который должна проделать информация о покупке, прежде чем будет осуществлена транзакция. Во время прохождения этого пути возможны искажения и потеря сообщений.

Для защиты системы POS должны выполняться следующие требования.

- Проверка PIN, введенного покупателем, должна производиться системой банка-эмитента. При пересылке по каналам связи значение PIN должно быть зашифровано.
- Сообщения, содержащие запрос на перевод денег (или подтверждение о переводе), должны проверяться на подлинность для защиты от замены и внесения изменений при прохождении по линиям связи и обрабатывающим процессорам [22].

Самым уязвимым местом системы POS являются ее POS-терминалы. В отличие от банкоматов в этом случае изначально предполагается, что POS-терминал не защищен от внешних воздействий. Угрозы для POS-терминала связаны с возможностью раскрытия секретного ключа, который находится в POS-терминале и служит для шифрования информации, передаваемой этим терминалом в банк-эквайер. Угроза раскрытия ключа терминала достаточно реальна, так как эти терминалы устанавливаются в таких неохраняемых местах, как магазины, автозаправочные станции и пр.

Потенциальные угрозы из-за раскрытия ключа получили такие названия.

- *"Обратное трассирование"*. Сущность этой угрозы состоит в том, что если злоумышленник получит ключ шифрования, то он может пытаться восстановить значения PIN, использованные в предыдущих транзакциях.
- *"Прямое трассирование"*. Сущность этой угрозы состоит в том, что если злоумышленник получит ключ шифрования, то он попытается восстановить значения PIN, которые будут использоваться в последующих транзакциях.

Для защиты от угроз обратного и прямого трассирования предложены три метода:

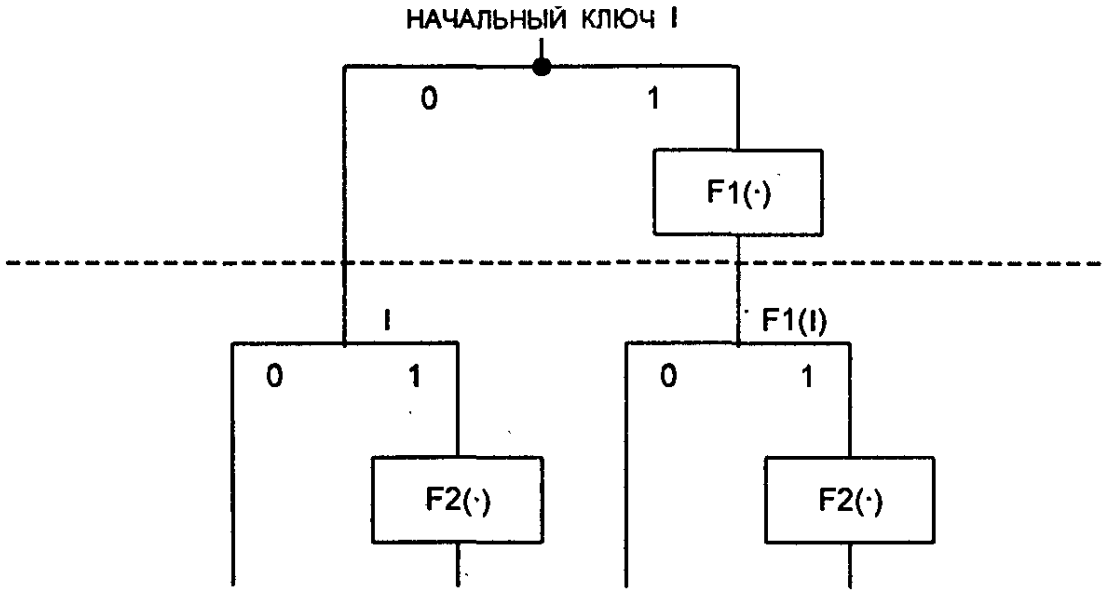


Рис. 9.5. Схема вывода ключа с учетом двоичного представления номера S ключа

- метод выведенного ключа;
- метод ключа транзакции;
- метод открытых ключей [22].

Сущность первых двух методов состоит в том, что они обеспечивают модификацию ключа шифрования передаваемых данных для каждой транзакции.

Метод выведенного ключа обеспечивает смену ключа при каждой транзакции независимо от ее содержания. Для генерации ключа шифрования используют однонаправленную функцию от текущего значения ключа и некоторой случайной величины. Процесс получения (вывода) ключа для шифрования очередной транзакции представляет собой известное "блуждание" по дереву (рис. 9.5).

Вершиной дерева рис. 9.5 является некоторое начальное значение ключа I. Чтобы получить ключ с номером S, число S представляют в двоичной форме. Затем при вычислении значения ключа учитывается структура двоичного представления числа S, начиная со старшего разряда. Если L-й двоичный разряд числа S равен 1, то к текущему значению ключа K применяется однонаправленная функция FL(K), где L – номер рассматриваемого двоичного разряда. В противном случае переходят к рассмотрению следующего разряда числа S, не применяя однонаправленной функции. Последняя реализована на основе алгоритма DES. Для получения достаточного быстродействия количество единиц в двоичном представлении числа S обычно ограничивается – их должно быть не более 10. Этот метод обеспечивает защиту только от угрозы "обратного трассирования".

Метод ключа транзакции позволяет шифровать информацию, передаваемую между POS-терминалами и банком-эквайером,

на уникальном ключе, который может меняться от транзакции к транзакции. Для генерации нового ключа транзакции используются следующие составляющие:

- однонаправленная функция от значения предыдущего ключа;
- содержание транзакции;
- информация, полученная от карты.

При этом предполагается, что предыдущая транзакция завершилась успешно. Метод ключа транзакции обеспечивает защиту как от "обратного трассирования", так и от "прямого трассирования". Раскрытие одного ключа не дает возможности злоумышленнику вскрыть все предыдущие и все последующие транзакции. Недостатком данной схемы является сложность ее реализации.

Метод открытых ключей позволяет надежно защититься от любых видов трассирования и обеспечить надежное шифрование передаваемой информации. В этом случае POS-терминал снабжается секретным ключом для расшифровки сообщений банка-эквайера. Этот ключ генерируется при инициализации терминала. После генерации секретного ключа терминал посылает связанный с ним открытый ключ на компьютер банка-эквайера. Обмен между участниками взаимодействия выполняется с помощью открытого ключа каждого из них. Подтверждение подлинности участников осуществляется специальным центром регистрации ключей с использованием своей пары открытого и закрытого ключей. Недостатком этого метода является его сравнительно малое быстродействие.

9.5. Обеспечение безопасности банкоматов

Банкоматом называют банковский автомат для выдачи и инкассирования наличных денег при операциях с пластиковыми картами. Кроме того, банкомат позволяет держателю карты получать информацию о текущем состоянии счета (в том числе и выписку на бумаге), а также проводить операции по перечислению средств с одного счета на другой.

Банкомат снабжен устройством для чтения карты, а также дисплеем и клавиатурой для интерактивного взаимодействия с держателем карточки. Банкомат оснащен персональной ЭВМ, которая обеспечивает управление банкоматом и контроль его состояния. Последнее весьма важно, поскольку банкомат является хранилищем наличных денег. Для обеспечения коммуникационных функций банкоматы оснащаются платами X.25, а иногда и модемами.

Денежные купюры в банкомате размещаются в кассетах, которые находятся в специальном сейфе. Число кассет определяет

количество номиналов купюр, выдаваемых банкоматом. Размеры кассет регулируются, что позволяет заряжать банкомат практически любыми купюрами.

Банкоматы – это стационарные устройства больших габаритных размеров и веса. Примерные размеры: высота – 1,5...1,8 м, ширина и глубина – около 1 м, вес – около тонны. Более того, с целью пресечения возможных хищений их монтируют капитально. Банкоматы размещают как в охраняемых помещениях, так и непосредственно на улице.

На сегодняшний день большинство моделей банкоматов рассчитано на работу в режиме реального времени (on-line) с картами с магнитной полосой, однако появились банкоматы, способные работать со смарт-картами в автономном режиме (off-line).

Автономный режим (off-line) работы банкомата характерен тем, что банкомат функционирует независимо от компьютеров банка. Запись информации о транзакции производится на внутренний магнитный диск и выводится на встроенный принтер. Достоинствами автономного режима банкомата являются его относительная дешевизна и независимость от качества линий связи. Это весьма важно для стран с плохой телефонной связью. В то же время низкая стоимость установки напрямую обусловливает высокую стоимость эксплуатации таких банкоматов [22, 52]. Чтобы обновлять "черные списки" (стоп-списки) утраченных карточек, необходимо хотя бы раз в день специально выделенному человеку обходить и обслуживать такие банкоматы. При большом числе таких устройств подобное обслуживание затруднительно. Отказ же от ежедневного обновления списков может привести к значительным потерям для банка в случае подделки карты или при пользовании краденой картой.

Сложности возникают также при идентификации (аутентификации) клиента. Для защиты информации, хранящейся на карте с магнитной полосой, применяется ее шифрование. Для того чтобы банкоматы одного и того же банка воспринимали пластиковые карты с магнитной полосой, в них должен быть использован один ключ для шифрования (расшифрования). Компрометация его хотя бы на одном из банкоматов приведет к нарушению защиты на всех банкоматах.

Режим реального времени (on-line) характерен тем, что банкомат должен быть подсоединен непосредственно или через телефонную сеть к главному компьютеру банка. В этом случае регистрация транзакций осуществляется непосредственно на главном компьютере банка, хотя подтверждение о транзакции выдается на принтер банкомата. При реализации транзакции банкомат обменивается с главным компьютером банка тремя сообщениями (рис. 9.6):

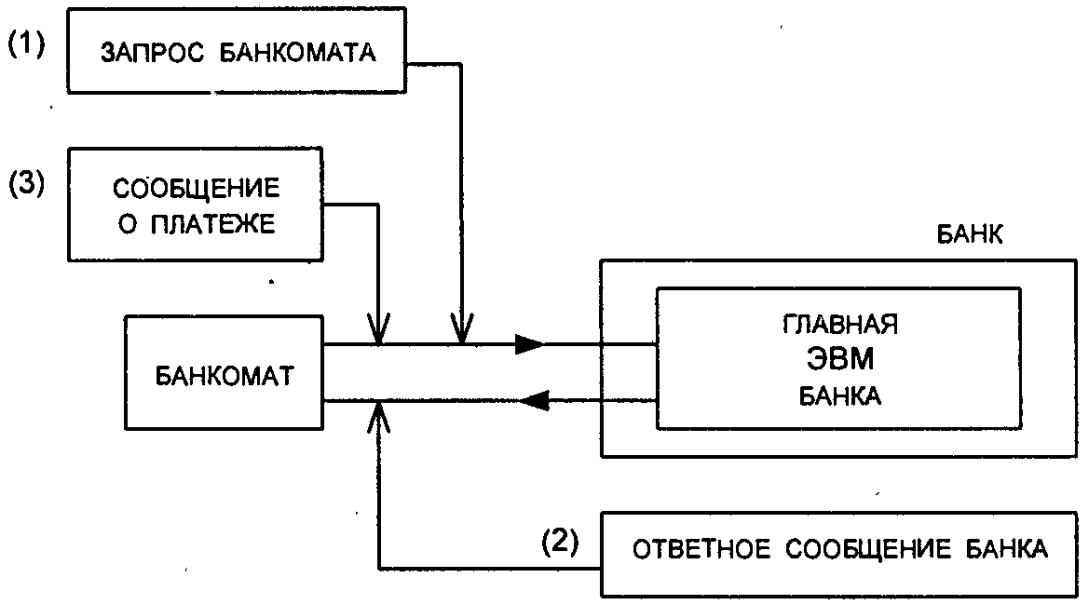


Рис. 9.6. Схема обмена сообщениями между банкоматом и главной ЭВМ банка при идентификации и платеже

1) запрос банкомата;
2) ответное сообщение банка;
3) сообщение банкомата о платеже.

Запрос банкомата включает следующие данные:

- идентификатор банкомата;
- номер счета и другая учетная информация клиента;
- серийный номер карты;
- защитный символ;
- зашифрованный PIN клиента;
- количество требуемых денег;
- номер транзакции;
- проверочный код для всех данных сообщения.

Ответное сообщение банка включает следующие данные:

- идентификатор банкомата;
- код операции, разрешающий (запрещающий) платеж;
- номер транзакции;
- проверочный код для всех данных сообщения.

В этом обмене сообщениями для проверки целостности данных используется код аутентификации сообщения MAC (Message Authentication Code).

Режим реального времени имеет ряд преимуществ по сравнению с автономным режимом. Он дает возможность клиенту не только получить наличные деньги, но и осуществлять манипуляции со своим счетом. Централизованная идентификация/аутентификация позволяет существенно повысить устойчивость системы к компрометации ключей шифрования. Централизованная проверка идентификатора пользователя делает возможным оперативное об-

новление списков запрещенных к использованию карт, а также введение ограничений на количество наличных денег, которые может получить клиент в течение одного дня (для защиты от использования украденных карт).

Однако этот режим возможен лишь при наличии надежных каналов связи между банкоматами и банком, что делает его довольно дорогим. Кроме того, наличие канала связи порождает и другие угрозы безопасности по сравнению с автономным режимом работы. Это – анализ трафика между банкоматом и главным компьютером и имитация работы главного компьютера компьютером злоумышленника. При анализе трафика можно получить информацию о счетах, суммах, условиях платежей и т.п. При имитации работы главного компьютера банка компьютер злоумышленника может выдавать положительный ответ на запрос банкомата о результатах идентификации/аутентификации.

Сети банкоматов являются в настоящее время распространенной формой эксплуатации банкоматов, в которой участвуют несколько банков [22, 123]. Банки-участники такой сети преследуют следующие цели:

- уменьшение стоимости операций для участников;
- разделение затрат и риска при внедрении новых видов услуг между участниками;
- преодоление географических ограничений и соответственно повышение субъективной ценности услуг для потребителей.

При совместном использовании несколькими банками сети банкоматов возникает серьезная проблема – защита конфиденциальной информации банков друг от друга (ключи шифрования и т.п.). Для разрешения этой проблемы предложена схема централизованной проверки PIN каждым банком в своем центре связи с банкоматами. Усложняется также система распределения ключей между всеми участниками сети.

Рассмотрим схему прохождения информации о PIN клиента между банкоматом, банком-эквайером (которому принадлежит банкомат) и банком-эмитентом (который выпустил карту клиента) (рис. 9.7).

Пусть клиент Банка 2 (Эмитента) обратился к банкомату Банка 1 (Эквайера). При этом в сети банкоматов происходят следующие действия.

1. Считывающее устройство банкомата считывает информацию, записанную на банковской карте, предъявленной клиентом, и затем банкомат определяет, имеет ли этот клиент счет в Банке 1 – Эквайере.

2. Если клиент не имеет счета в Банке 1, транзакция направляется в сетевой маршрутизатор, который, используя идентификационный номер Банка 2 – Эмитента BIN (Bank Identification Number),

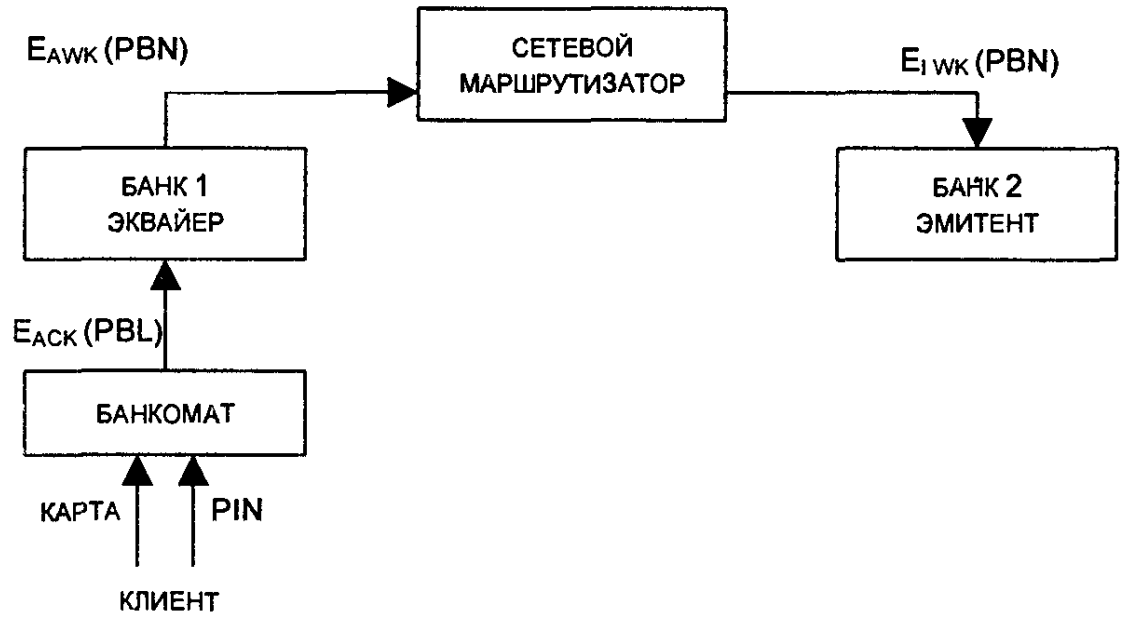


Рис. 9.7. Схема прохождения информации о PIN клиента между банкоматом, банком-эквайером и банком-эмитентом

направляет эту транзакцию на главный компьютер Банка 2 или производит проверку PIN для Банка 2.

3. Если проверка PIN производится на главном компьютере Банка 2, то этот компьютер получает полную информацию о транзакции и проверяет достоверность PIN.

4. Независимо от результата проверки компьютер Банка 2 пересылает сообщение с этим результатом через сетевой маршрутизатор компьютеру Банка 1.

Как следует из примера, к банку-эмитенту предъявляются следующие требования:

- выпускаемые им карты должны восприниматься всеми банкоматами сети;
- банк-эмитент должен обладать технологией проверки PIN собственных клиентов.

К банку-эквайеру предъявляются другие требования:

- в банкомате или главном компьютере банка должна быть реализована проверка принадлежности транзакции;
- если нет возможности проверить правильность чужого PIN, банк-эквайер должен передать данные о транзакции на сетевой маршрутизатор.

Для защиты взаимодействия компьютеров банков друг с другом и с банкоматами должно применяться оконечное (абонентское) шифрование информации, передаваемой по линиям связи. Обычно используется следующий подход: вся сеть банкоматов разбивается на зоны, и в каждой из них используется свой *главный зональный управляющий ключ ZCMK (Zone Control Master Key)*. Ключ ZCMK предназначен для шифрования ключей при обмене между сетевым

маршрутизатором и главным компьютером банка. Ключ ZCMK индивидуален для всех участников сети. Обычно он генерируется случайным образом маршрутизатором и передается неэлектронным способом в банк. Раскрытие ключа ZCMK приведет к раскрытию всех PIN, которые передаются между маршрутизатором и главным компьютером банка.

Для шифрования информации, поступающей от главного компьютера банка-эмитента на маршрутизатор используется *рабочий ключ эмитента IWK (Issuer Working Key)*. Его сообщает главному компьютеру банка-эмитента маршрутизатор в зашифрованном на уникальном ZCMK виде. Ключ IWK может меняться по запросу пользователя в процессе работы.

Аналогичный по назначению ключ для обмена между банком-эквайером и маршрутизатором называется *рабочим ключом эквайера AWK (Acquirer Working Key)*. Для шифрования информации при передаче от банкомата к главному компьютеру банка-эквайера используется *связной ключ эквайера ACK (Acquirer Communication Key)*.

При рассмотрении функционирования системы защиты введены следующие обозначения:

$E_Y(X)$ – шифрование сообщения X по алгоритму DES с использованием ключа Y;

$D_Y(X)$ – расшифрование сообщения X по алгоритму DES с использованием ключа Y;

PBL(PIN Block Local) – локальный блок PIN, полученный из введенного клиентом PIN, дополненного до восьми символов, и представленный во внутреннем формате банкомата;

PBN(PIN Block Network) – сетевой блок PIN, полученный из введенного клиентом PIN, дополненного до восьми символов, и представленный в виде, готовом для передачи в сети.

Вернемся к рассмотрению схемы на рис. 9.7.

1. Клиент предъявил банкомату Банка 1 банковскую карту и ввел с клавиатуры свой PIN. Банкомат формирует PBL, шифрует его с использованием ACK, т.е. вычисляет криптограмму $E_{ACK}(PBL)$, и отправляет ее на главный компьютер Банка 1.

2. На главном компьютере Банка 1 блок PBL расшифровывается и преобразуется в блок PBN, затем блок PBN шифруется с использованием AWK и отсылается в Сетевой маршрутизатор. Процесс преобразования

$$E_{ACK}(PBL) \rightarrow E_{AWK}(PBN)$$

называют трансляцией блока PIN с ключа ACK на ключ AWK. Основное назначение этого процесса – смена ключа шифрования.

3. Если PIN проверяется на Сетевом маршрутизаторе, после получения криптограммы $E_{AWK}(PBN)$ производится ее расшифрование, а затем выделение PIN с помощью преобразований

$$D_{AWK}(E_{AWK}(PBN)) = PBN \rightarrow PIN.$$

Если PIN проверяется Банком 2, принятая криптограмма транслируется с ключа AWK на ключ IWK (оба ключа хранятся на Сетевом маршрутизаторе):

$$E_{AWK}(PBN) \rightarrow E_{IWK}(PBN).$$

Затем криптограмма E_{IWK}(PBN) отправляется в Банк 2.

4. Поступившая в Банк 2 криптограмма E_{IWK}(PBN) преобразуется в зависимости от используемого способа проверки либо в открытый PIN:

$$D_{IWK}(E_{IWK}(PBN)) = PBN \rightarrow PIN,$$

либо в PIN в форме блока PBL, зашифрованного на ключе базы данных DBK:

$$E_{IWK}(PBN) \rightarrow E_{DBK}(PBL).$$

5. После любого из этих преобразований осуществляется поиск принятого PIN в базе данных существующих PIN.

6. В результате выполненной проверки введенный клиентом PIN либо принимается, либо отвергается. Вне зависимости от результата проверки главный компьютер Банка 2 пересылает сообщение с результатом через Сетевой маршрутизатор на компьютер Банка 1, а тот оповещает банкомат о результатах решения.

Рассмотренная схема обеспечения безопасности взаимодействия компьютеров в сети базируется на симметричном алгоритме шифрования DES. Поэтому на распространение ключа ZCMK налагаются жесткие ограничения. Применение асимметричной системы шифрования с открытым ключом позволяет несколько упростить ключевую систему и соответственно взаимодействие между банкоматами и главными компьютерами банков.

В неразделяемой сети банкоматов достаточно использовать на всех банкоматах одинаковый открытый ключ, а на главном компьютере банка – закрытый ключ. Это позволяет шифровать запрос и подтверждающее сообщение из банка, так как обеспечение конфиденциальности ответного сообщения необязательно.

Проблема защиты запроса от активных атак (изменения или введения ложного запроса) может быть решена в случае неразделяемой сети использованием пароля для идентификации банкоматов.

9.6. Универсальная электронная платежная система UEPS

Ряд социальных и экономических проблем, присущих России после распада СССР: наличие в стране высококвалифицированных специалистов, низкий уровень оплаты труда технической ин-

теллигенции, высокий уровень криминальности в стране – дают основание предположить, что проблемы мошенничества в электронных системах безналичных расчетов с использованием пластиковых карт могут стоять в России более остро по сравнению с Западом, где ежегодные потери составляют миллиарды долларов. Поэтому вопрос обеспечения безопасности функционирования электронной платежной системы и контроля доступа к финансовой информации приобретает особое значение. Ввиду недостаточного развития линий связи в России наиболее перспективны платежные системы, основанные на автономном принципе (off-line) обслуживания владельцев карточек в торговой точке или банкомате. Универсальная электронная платежная система UEPS (Universal Electronic Payment System) отвечает указанным требованиям и отличается высоким уровнем защищенности, что подтверждено результатами авторитетных международных экспертиз. Именно поэтому построение электронной платежной системы "Сберкарт" с использованием микропроцессорных карт в Сбербанке Российской Федерации базируется на технологии UEPS. Концепция и технология платежной системы UEPS разработана французской компанией NET 1 International [85].

Основным технологическим принципом UEPS является осуществление всех финансовых транзакций в режиме off-line при непосредственном взаимодействии двух интеллектуальных пластиковых карт. Базовым алгоритмом шифрования информации служит алгоритм DES. Высокая криптостойкость обеспечивается использованием двойного шифрования на ключах длиной 8 байт.

В платежных системах, работающих в режиме off-line, большая часть функций по обеспечению контроля действий, по защите от мошенничества ложится на микропроцессорную карту – базовый элемент UEPS. В UEPS используются три основных типа микропроцессорных карт:

- служебные карты персонала банка;
- торговые карты;
- карты клиента.

Все карты содержат 8-битовый микропроцессор.

Приведем технические характеристики карты клиента системы UEPS.

- Процессор: SGS-Thompson, 8 бит, система команд Motorola 6805.
- Операционная система: Многозадачная операционная система чипа MCOS (Multitasking Chip Operation System).
- ОЗУ: 160 байт.
- ПЗУ: 6 Кбайт.
- ЭСППЗУ: 2 Кбайт (16 Кбит).

Конструкция и архитектура микропроцессора не позволяют осуществить механическое считывание информации путем спиливания кристалла по слоям, сканирования электронным микроскопом, воздействия ультрафиолетом и т.д. При попытках совершить подобные операции микропроцессор полностью выходит из строя. Архитектура самой микропроцессорной карты такова, что процессор контролирует доступ к защищенным областям памяти, передавая управление специальной прикладной программе UEPS. Вся информация поступает извне на карту в зашифрованном виде и расшифровывается прикладной программой внутри самой карты с использованием ключей, хранящихся в защищенных областях памяти. Аналогичным образом шифруется информация, покидающая карту.

Банковские ключи никогда не покидают карту в открытом виде.

Состав и архитектура платежной системы. Системообразующим уровнем единой платежной системы является центр эмиссии (рис. 9.8), который выполняет следующие функции:

- генерацию генерального (системообразующего) ключа платежной системы;
- первичную эмиссию микропроцессорных карт – присвоение картам уникальных серийных номеров USN, занесение на карты общесистемной идентифицирующей и контрольной информации, занесение на карты генерального ключа системы;
- ведение справочников участников расчетов, регистрацию новых участников (банков-эмитентов и эквайеров) в системе;
- ведение справочников типов карт и кодов валют, используемых в системе;
- ведение единой базы данных по заводским номерам и USN-номерам карт, имеющих хождение в системе.

Вторым уровнем платежной системы являются банки-участники. Банк-участник платежной системы – финансовый институт, участвующий в расчетах по микропроцессорным картам и несущий

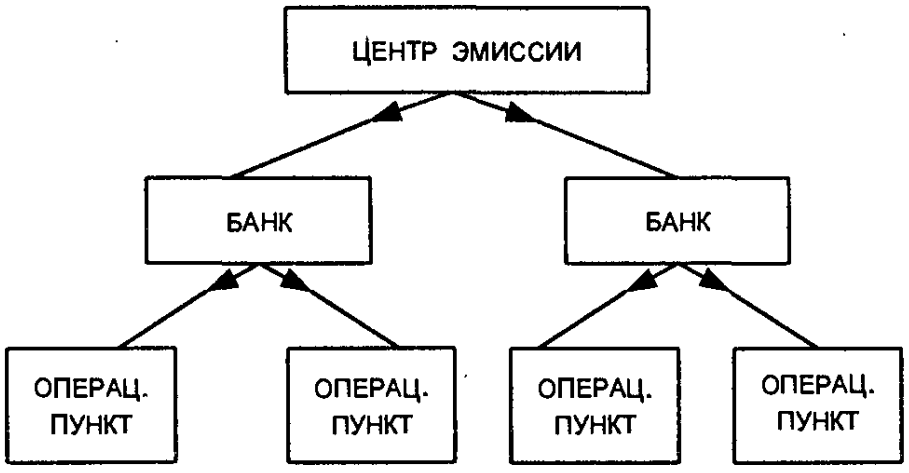


Рис. 9.8. Архитектура платежной системы

полную финансовую ответственность по транзакциям, совершенным эмитированными им картами. Каждый из банков-участников перед началом выпуска своих карт (клиентских и торговых) создает собственный набор ключей эмитента или эквайера, которые заносятся на карты в процессе эмиссии и используются при формировании и обработке финансовых транзакций. В составе технических средств банка-участника действует ряд автоматизированных рабочих мест (АРМ) исполнителей: администратора, безопасности, бухгалтера.

Третьим уровнем иерархии в платежной системе являются операционные пункты. Операционными пунктами называют структурные подразделения банка-участника, в которых производится обслуживание клиентов банка – открытие/закрытие карточных счетов, выдача карточек, выполнение приходных и расходных операций. Карточная система банка-участника должна включать как минимум один операционный пункт.

Распределение ключей и паролей. В основе безопасности платежной системы UEPS лежит тщательно проработанная схема распределения и использования ключей и индивидуальных паролей субъектов системы UEPS. Распределение ключей и паролей по картам банка, торговца и клиента приведено в табл. 9.1.

Таблица 9.1

Распределение ключей и паролей по картам банка, торговца и клиента

Карта банка	Карта торгоаца	Карта клиента	Наименование
P_0	P_0	P_0	Мастер-ключ
P1-PIN B	P1-PIN M	P1-PIN 1	Пароли P1
P2-RFU	P2-RFU	P2-PIN 2	Пароли P2
P3	P3	P3	Пароль P3
P4	P4	P4	Пароль P4
P5	P5	P5	Пароль P5
P6	P6	P6	Пароль P6
P7	P7	P7	Системообразующий ключ P7
KI1, KI2	–	KI1, KI2	Ключи клиентских карточек
–	KA1, KA2	–	Ключи торговых карточек
SK	SK	SK	Сессионный (сеансовый) ключ обмена

Дадим пояснения к табл. 9.1.

Мастер-ключ P_0 обеспечивает генеральный доступ к карте. Назначается и известен только центру эмиссии.

Группа паролей P1:

PIN B – пароль операциониста банка.

PIN M – пароль кассира магазина.

PIN 1 – пароль на зачисление средств на карту. Назначается и известен только владельцу карты. Изменяется владельцем в off-line терминале.

Группа паролей P2:

RFU – резервный пароль.

PIN 2 – пароль на списание средств с карты. Назначается и известен только владельцу карты. Изменяется владельцем в off-line терминале. (Пароли PIN 1 и PIN 2 могут быть одинаковыми по желанию владельца карты.)

Группы паролей P3 и P4 являются резервными.

Пароль P5 участвует совместно с P7 в образовании сессионных (сеансовых) ключей. Общий для всех банков-участников единой расчетной системы. Назначается центром эмиссии.

Пароль P6 предоставляет доступ на запись ключей KIx, KAx. Назначается банком-участником.

P6–RFU – резервный пароль.

Системообразующий ключ P7 участвует в образовании сессионных ключей. Является общим для всех банков-участников единой платежной системы. Назначается центром эмиссии.

Ключи клиентских карточек KI1, KI2 предъявляются при зачислении средств на карту. Участвуют в шифровании записи о транзакции. Назначаются банком-участником.

Ключи торговых карточек KA1, KA2 предъявляются при инкассации карты торговца. Участвуют в шифровании записи о транзакции. Назначаются банком-эмитентом.

Сессионный (сеансовый) ключ обмена SK формируется в памяти карт в результате диалога карты с картой и служит для шифрования всех информационных потоков между картами на протяжении сеанса связи. Уникален для каждого сеанса связи карта-карта.

Цикл платежной транзакции. В цикле платежной транзакции участвуют три стороны:

- финансовый институт (банк-участник);
- владелец карты;
- предприятие торговли или сферы услуг, банкомат.

Жизненный цикл платежной транзакции можно разбить на три этапа.

На первом этапе владелец карты имеет возможность получить по своей карте электронную наличность в размере, не превышающем остаток на его лицевом счете (или банк может кредитовать клиента). Эта операция может выполняться как оператором банка, так и в режиме самообслуживания. Она производится на банковском терминале самообслуживания или на рабочем месте оператора банка в режиме on-line с автоматизированной системой банка, так как нужен доступ к информации о состоянии карт-счета клиента, на основании которой и осуществляется финансовая операция. Поэтому подобные операции могут совершаться в любом месте, где есть on-line связь с базой данных карточных счетов клиентов банка.

Для выполнения этой операции клиент обязан предъявить пароль PIN1 на пополнение средств карты со своего счета в банке.

Далее клиент может совершать платежные операции на суммы, не превышающие остатка электронных средств на его карте, в любом месте, где установлено оборудование по обслуживанию микропроцессорных карт стандарта UEPS: off-line торговый терминал, банкомат и т.д. Следует заметить, что реальные деньги, полученные клиентом на карту, находятся на протяжении всего цикла платежной транзакции в банке на отдельном счете.

На втором этапе клиент осуществляет платежную операцию в торговой точке. Эта операция проходит в режиме off-line без запроса на авторизацию владельца карты, так как вся необходимая информация, включая и секретную часть, находится на карте клиента, а карта представляет собой электронный кошелек.

Технически эта операция выполняется следующим образом. В торговом терминале установлена микропроцессорная карта торговца, и клиент, вставив свою карту в считывающее устройство торгового терминала, производит списание суммы покупки со своей карты на карту торговца, при этом баланс карты клиента уменьшается на сумму транзакции, а баланс карты торговца возрастает на аналогичную сумму. Кроме того, на карту торговца и на карту покупателя заносится полная информация о совершенной транзакции: дата/время, сумма транзакции, идентификатор покупателя и магазина с информацией о банке и номере счета владельца.

Для совершения транзакции покупатель должен ввести свой пароль PIN 2 на расходование средств со своей карты. Клиент и торговец получают дополнительно твердые копии информации о совершенной транзакции (чек покупателя и журнальная лента магазина). Все транзакции также дублируются в памяти торгового терминала в зашифрованном виде. На бумажном чеке отображается название магазина, дата/время совершения операции, номер карты клиента, сумма операции, а также кодированная строка с информацией о совершенной транзакции (для обеспечения возможности восстановления информации о совершенной транзакции).

На третьем этапе торговец, собрав в течение дня на карту торговца список всех проведенных за торговую сессию транзакций с подробным описанием каждой, передает (инкассирует) данную информацию с карты торговца в систему расчетов банка. Эта операция может осуществляться автоматически, по модемной телефонной связи, или физически, по предъявлении карты торговца в любом ближайшем отделении банка или пункте инкассации, но в любом случае зашифрованный список транзакций передается именно с карты торговца, а не из памяти торгового терминала. После завершения сеанса "инкассации" карта торговца очищается

для работы в следующем сеансе, и на нее переносятся изменения списка "горячих карт" (hot-list), который карта торговца сообщает торговому терминалу в начале следующего рабочего дня (новой торговой сессии).

На следующем этапе банк, получив информацию о произведенных транзакциях, перечисляет сумму по всем совершенным транзакциям данного магазина на счет торговой организации.

Торговые терминалы. Торговые учреждения и банковские пункты выдачи наличности оснащаются терминалами типа EFT-10 с программным обеспечением UEPS. Терминал имеет два считывателя для микропроцессорных карт. В один считыватель в начале рабочего дня устанавливается карта торговца, в другой – карта покупателя при оплате покупки. В базовой поставке терминал EFT-10 имеет также считыватель для карт с магнитной полосой и встроенный модем, что позволяет организовать на одном устройстве обслуживание и пластиковых карт с магнитной полосой [29].

Торговый терминал, постоянно находящийся вне банковского контроля, является с точки зрения безопасности одним из самых уязвимых элементов платежной системы. Он может подвергаться попыткам взлома (несанкционированного доступа) со стороны криминальных структур. Поэтому недопустимо доверять торговому терминалу секретную, критичную с точки зрения функционирования платежной системы информацию, т.е. банковские ключи и пароли, алгоритмы шифрования, списки финансовых транзакций и т.д.

В платежной системе UEPS торговый терминал не хранит никакой секретной информации, а играет только роль элемента, обеспечивающего интерфейсное взаимодействие двух защищенных интеллектуальных устройств: карточки клиента и карточки торговца. Все платежные операции совершаются только в диалоге двух карт. При этом вне карт вся информация всегда зашифрована на базе сессионных ключей.

Формирование сессионных ключей. Диалог между картами клиента и торговца в торговом терминале осуществляется на базе сессионных ключей.

Карта клиента, используя внутренний датчик случайных чисел, вырабатывает случайное число в начале каждого нового сеанса взаимодействия с картой торговца, шифрует это число на системных ключах P7, P5 и сообщает карте торговца.

Карта торговца, располагая теми же самыми системными ключами P7, P5, расшифровывает принятую информацию и получает то же самое число в расшифрованном виде. Используя данное число в комбинации с другими ключами и общими для обеих карт данными, карты клиента и торговца одновременно вырабатывают сессионный ключ, который идентичен для обеих карт и уникален для каждого сеанса связи карточек клиента и торговца.

Сессионный ключ находится только в памяти обеих карт и никогда их не покидает. На базе этого сессионного ключа зашифровываются все информационные потоки между картами, что делает бесполезными попытки перехвата сообщений в торговом терминале.

Эмиссия карточек. Все банки-участники единой платежной системы по картам стандарта UEPS получают карты, оснащенные индивидуальным логотипом заказчика (банка-эмитента) и стандартизованным программным обеспечением.

Процедура эмиссии карт состоит их трех этапов:

- назначение центром эмиссии системных ключей;
- назначение банком-участником банковских ключей и паролей;
- персонализация карты клиента банком-участником.

Из них первые два этапа являются секретными и выполняются с соблюдением соответствующих мер безопасности в специально оборудованных помещениях. Третий этап, связанный с непосредственной персонализацией карты, является несекретным и выполняется рядовым оператором банка в операционном зале в присутствии клиента.

Система эмиссии карт, распределения и назначения ключей организована таким образом, чтобы сохранить за каждым банком уникальные права и ответственность за владение секретной информацией о своих банковских финансовых ключах.

Процесс эмиссии карт реализуется следующим образом. Центр эмиссии получает тираж карточек трех видов – банковские, торговые и клиентские. Все карточки изначально отформатированы и загружены соответствующим программным обеспечением UEPS. Доступ ко всем картам закрыт транспортным ключом P0-транспортный (уникальный для каждого тиража), который сообщается поставщиком уполномоченному сотруднику банка.

Первый этап эмиссии (секретная фаза) выполняется в центре эмиссии при получении каждого нового тиража карточек с обеспечением специальных мер безопасности администратором системы безопасности. Предъявляя карточкам P0-транспортный, центр эмиссии записывает на все карточки свой секретный мастер-ключ P0, системные ключи P7, P5 и устанавливает для каждой карты уникальный порядковый номер USN в системе банка.

Второй этап эмиссии (секретная фаза) выполняется в банке-участнике при получении каждого нового тиража карточек с обеспечением специальных мер безопасности администратором системы безопасности. Для банковской и торговой карт устанавливаются соответствующие значения паролей P1 и P6. Презентуя пароли P6 на карты банка и торговца, устанавливаются пароли KI1 и KI2 для банковских карт и KA1 и KA2 – для торговых. Банк заносит на карты также дополнительную информацию (коды валют, информация о магазине и т.д.).

Третий этап эмиссии – персонализация карты является несекретной операцией, выполняемой в присутствии клиента оператором банка, и не требует дополнительных мер безопасности.

Процесс персонализации карты клиента возможен только в диалоге с картой оператора банка. Оператор, презентуя банковской карте свой пароль PIN B, заносит на карту клиента информацию о владельце (Ф.И.О., банковские реквизиты, срок действия карты и др.). Банковская карта переносит в зашифрованном виде на карту клиента банковские ключи KI1 и KI2 и записывает на карту клиента номер карты оператора, которая участвовала в персонализации. Банковские ключи KI1 и KI2, переносимые на карту клиента с банковской карты, зашифрованы на базе сессионных ключей.

Клиент заносит на карту пароли PIN 1 и PIN 2 со своей отдельной клавиатуры.

Карта оператора банка контролирует доступ оператора в систему, проверяя его личный пароль PIN B. Кроме того, независимо от желания оператора при каждой процедуре персонализации новой карты в память микропроцессора этой карты всегда заносится номер банковской карты оператора, выдавшего карту клиенту. Поэтому всегда можно установить, какой оператор и когда выдавал эту карту.

Следует отметить, что оператор банка не получает информацию о клиентских паролях PIN 1 и PIN 2 на зачисление и списание. Эти клиентские пароли не хранятся в системе, они назначаются клиентом, известны только карте и ее владельцу и могут быть изменены клиентом самостоятельно в любой торговой точке в режиме off-line.

Таким образом, без санкции владельца карты, выраженной в сообщении этой карте правильного пароля, никто другой, в том числе и оператор банка, не может провести финансовые операции с картой клиента.

Разграничение ответственности между банками-участниками общей платежной системы. В системе UEPS только банк-участник имеет право и техническую возможность доступа к информации на эмитируемых банком картах. Даже производители и поставщики, обладая всеми техническими средствами, знаниями форматов данных и сообщений в системе, исходных текстов программ, местонахождения и назначения всех ключей и паролей, не в состоянии получить доступ к секретной финансовой информации на карточках без знания банковских ключей и паролей [3].

В системе UEPS предусмотрено четкое разделение ключей и разграничение ответственности между банками-участниками единой платежной системы. Каждый банк-участник платежной системы имеет собственные банковские ключи и пароли, участвующие в шифровании финансовой информации и известные только ему.

Эти ключи и пароли уникальны для каждого банка. Таким образом, обеспечение мер безопасности сводится к обеспечению надежного хранения ключей каждым банком-участником системы.

Утрата ключей каким-либо банком-участником может привести к возможности несанкционированного доступа только к финансовой информации, касающейся этого банка, и не создаст угрозы финансовых потерь для остальных банков-эмитентов, участников единой платежной системы.

Только одна пара ключей является общей для всех банков-участников единой платежной системы – это системные ключи P7, P5, которые определяют принадлежность конкретной карты к данной платежной системе. Эти системные ключи участвуют лишь в выработке сессионного ключа в картах при операциях в торговой точке и не отвечают за шифрование какой-либо другой информации на карточках клиента или торговца.

Двойное шифрование записи о транзакции на ключах банка-эквайера и банка-эмитента. Запись о каждой платежной транзакции заносится на карту торговца и имеет сложную структуру. Часть информации остается незашифрованной (дата транзакции, банковские реквизиты покупателя), часть информации шифруется на ключах банка-эквайера KA1 и KA2 (сумма, номер USN карты покупателя, номер транзакции на карте торговца и др.), а часть информации – на ключах банка-эмитента KI1 и KI2 (сумма, USN, PAN, номер транзакции в списке на карте клиента и др.).

Торговец в конце торговой сессии инкассирует список платежных транзакций в свой банк-эквайер. Этот банк-эквайер, предъявляя свои ключи KA1 и KA2, расшифровывает свою часть платежной транзакции и определяет, клиент какого банка, когда и на какую сумму совершил покупку в его магазине. Получив из записи о транзакции информацию о банковских реквизитах покупателя, банк-эквайер формирует электронное платежное уведомление для банка-эмитента, частью которого является зашифрованный сертификат банка-эмитента.

Банк-эмитент, получив платежное уведомление, расшифровывает вторую часть транзакции, предъявляя свои банковские ключи KI1 и KI2. Если расшифрованная информация полностью соответствует содержащейся в платежном уведомлении (в первую очередь сумма транзакции и реквизиты владельца карточки, совершившего покупку), то это платежное уведомление признается подлинным и оплачивается, в противном случае оно отвергается. Таким образом исключается возможность фальсификации платежных уведомлений в межбанковских расчетах.

При учреждении банками общего процессингового центра банки могут сохранить право контроля над межбанковскими взаиморасчетными операциями. При этом каждый банк оставляет за со-

бой исключительное право владения, назначения и ротации банковских ключей KI1, KI2, KA1, KA2.

Контроль прохождения транзакций в платежной системе. Для обеспечения контроля безопасности и решения спорных ситуаций в платежной системе необходима эффективная схема организации сквозной уникальной нумерации и учета платежных транзакций. В системе каждая платежная транзакция идентифицируется композицией следующих элементов:

- уникальный серийный номер карты клиента в системе;
- порядковый номер транзакции по списку транзакций на карте клиента;
- уникальный серийный номер карты магазина в системе;
- порядковый номер транзакции по списку транзакций на карте магазина;
- порядковый номер инкассации карты магазина.

Реализованная схема позволяет однозначно проследить прохождение транзакции по всем элементам системы:

Банк – Клиент – Магазин – Банк.

9.7. Обеспечение безопасности электронных платежей через сеть Internet

Еще несколько лет назад сеть Internet использовалась в основном только для обмена почтовыми сообщениями и пересылки файлов. Однако в последние годы современные информационные технологии превратили Internet в развитую инфраструктуру, которая охватывает все основные информационные центры, мировые библиотеки, базы данных научной и правовой информации, многие государственные и коммерческие организации, биржи и банки. Любая организация может распространять информацию по всему миру, создав информационный абонентский пункт в WWW Internet.

Все большее значение приобретает электронная торговля. Число покупок по банковским картам будет расти по мере создания систем заказов в оперативном режиме Internet. Сегодня Internet может рассматриваться как огромный рынок, способный охватить практически все население планеты Земля. Пользование открытой компьютерной сетью Internet меняет способ доступа к информации о приобретении, предложении и оплате услуг, покупке товаров и расчетах. Места совершения сделок постепенно перемещаются от традиционных рынков к более комфортным для потребителя – в дом или офис. Именно поэтому производители программных и аппаратных средств, торговые и финансовые организации активно развивают различные виды и методы ведения коммерческой деятельности в Internet – электронной торговли, проявляя надлежащую заботу об обеспечении ее безопасности [1, 95].

Основные виды электронной торговли

Под термином "электронная торговля" понимают предоставление товаров и платных услуг через глобальные информационные сети. Рассмотрим наиболее распространенные на сегодня виды электронной коммерции [17].

- Традиционной услугой в области электронной торговли является продажа информации, например подписка на базы данных, функционирующие в режиме on-line. Этот вид услуг уже получил распространение в России (базы данных "Россия-он-Лайн", "Гарант-Парк" и др.).
- За рубежом в последнее время становится все более популярной концепция "электронных магазинов". Обычно электронный магазин представляет собой Web-site, в котором имеется оперативный каталог товаров, виртуальная "тележка" покупателя, на которую "собираются" товары, а также средства оплаты – по предоставлению номера кредитной карточки по сети Internet или по телефону. Оперативные каталоги товаров могут обновляться по мере изменения предложений продукции либо для отражения сезонных мер стимулирования спроса. Отправка товаров покупателям осуществляется по почте или, в случае покупки электронных товаров (например, программного обеспечения), по каналам электронной почты, или непосредственно через Web-site по сети Internet.
- Начинает развиваться новый вид электронной коммерции – *электронные банки*. Среди основных достоинств электронных банков можно выделить относительно низкую себестоимость организации такого банка (не нужно арендовать престижные здания, не нужны хранилища ценностей и т.д.) и широкий охват клиентов (потенциальным клиентом электронного банка может стать практически любой пользователь Internet). Поэтому электронный банк может предоставлять клиентам более выгодные, чем у обычного банка, проценты, а также больший спектр банковских услуг за более низкую плату. Естественно, что электронный банк имеет собственные системы безопасности и защиты электронной информации, например специальные карты – генераторы случайных паролей, синхронизируемых с паролем на банковском сервере (это позволяет создавать уникальный пароль при каждом обращении клиента к банковскому серверу). Другой, менее дорогостоящий подход связан с использованием персональных смарт-карт, также позволяющих генерировать сессионные (сеансовые) ключи.

Некоторая задержка в развитии электронной торговли была обусловлена отсутствием надежной системы защиты. Пока платежная информация передается по открытым сетям с минимальными предосторожностями или вовсе без них. Это является благо-

приятной почвой для автоматизированного мошенничества (например, использование фильтров для всех сообщений, проходящих через какую-либо сеть, с целью извлечения номеров счетов кредитных карточек из потока данных), а также мошенничества "ради озорства", характерного для некоторых хакеров.

Основные методы защиты информации

Традиционный и проверенный способ электронной торговли, который ведет свое начало от обычной торговли по каталогам, представляет собой оплату товаров и услуг кредитной карточкой по телефону. В этом случае покупатель заказывает на Web-сервере список товаров, которые он хотел бы купить, и потом сообщает по телефону номер своей кредитной карточки продавцу коммерческой фирмы. Затем происходит обычная авторизация карты, а списание денег со счета покупателя производится лишь в момент отправки товара по почте или с курьером.

Для того чтобы покупатель – владелец кредитной карточки мог без опасений расплатиться за покупку через сеть, необходимо иметь более надежный, отработанный механизм защиты передачи электронных платежей. Такой принципиально новый подход заключается в немедленной авторизации и шифровании финансовой информации в сети Internet с использованием схем SSL и SET.

Протокол SSL (Secure Socket Layer) предполагает шифрование информации на канальном уровне (см. § 8.4).

Протокол "Безопасные электронные транзакции" SET (Secure Electronic Transactions), разработанный компаниями Visa и Master Card, предполагает шифрование исключительно финансовой информации. В течение полугода протокол SET обсуждался учеными всего мира. Главное требование, которое к нему предъявлялось, – обеспечить полную безопасность и конфиденциальность совершения сделок. На сегодняшний день технические условия протокола, обеспечивающие безопасность, признаны оптимальными. Ввод этого протокола в действие даст владельцам пластиковых карт возможность использовать компьютерные сети при проведении финансовых операций, не опасаясь за дальнейшую судьбу своих платежных средств.

Стандарт SET обещает существенно увеличить объем продаж по кредитным карточкам через Internet. Совокупное количество потенциальных покупателей – держателей карточек Visa и MasterCard по всему миру – превышает 700 миллионов человек. Обеспечение безопасности электронных транзакций для такого пула покупателей может привести к заметным изменениям, выражающимся в уменьшении себестоимости транзакции для банков и процессинговых компаний.

Особенности функционирования протокола SET

Для того чтобы обеспечить полную безопасность и конфиденциальность совершения сделок, протокол SET должен гарантировать непременное соблюдение следующих условий [8].

1. Абсолютная конфиденциальность информации. Владельцы карточек должны быть уверены в том, что их платежная информация надежно защищена и доступна только указанному адресату. Это является непременным условием развития электронной торговли.

2. Полная сохранность данных. Участники электронной торговли должны быть уверены в том, что при передаче от отправителя к адресату содержание сообщения останется неизменным. Сообщения, отправляемые владельцами карточек коммерсантам, содержат информацию о заказах, персональные данные и платежные инструкции. Если в процессе передачи изменится хотя бы один из компонентов, то данная транзакция не будет обработана надлежащим образом. Поэтому во избежание ошибок протокол SET должен обеспечить средства, гарантирующие сохранность и неизменность отправляемых сообщений. Одним из таких средств является использование цифровых подписей.

3. Аутентификация (установление подлинности) счета владельца карточки. Использование цифровых подписей и сертификатов владельца карточки гарантирует аутентификацию счета владельца карточки и подтверждение того, что владелец карточки является законным пользователем данного номера счета.

4. Владелец карточки должен быть уверен, что коммерсант действительно имеет право проводить финансовые операции с финансовым учреждением. Использование цифровых подписей и сертификатов коммерсанта гарантирует владельцу карточки, что можно безопасно вести электронную торговлю.

Участники системы расчетов и криптографические средства защиты транзакций. Протокол SET изменяет способ взаимодействия участников системы расчетов. В данном случае электронная транзакция начинается с владельца карточки, а не с коммерсанта или эквайера.

Коммерсант предлагает товар для продажи или предоставляет услуги за плату. Протокол SET позволяет коммерсанту предлагать электронные взаимодействия, которые могут безопасно использовать владельцы карточек.

Эквайером (получателем) является финансовое учреждение, которое открывает счет коммерсанту и обрабатывает авторизации и платежи по кредитным карточкам. Эквайер обрабатывает сообщения о платежах, переведенных коммерсанту посредством платежного межсетевого интерфейса. При этом протокол SET гарантирует, что при взаимодействиях, которые осуществляет владелец

карточки с коммерсантом, информация о счете кредитной карточки будет оставаться конфиденциальной.

Финансовые учреждения создают ассоциации банковских кредитных карточек, которые защищают и рекламируют данный тип карточки, создают и вводят в действие правила использования кредитных карточек, а также организуют сети для связи финансовых учреждений друг с другом.

Системы кредитных карт утвердились в значительной степени в качестве платежного средства для приобретения товаров непосредственно у продавца. Основное отличие использования кредитных карт в сети Internet заключается в том, что в соответствии со стандартом SET для защиты транзакций электронной торговли используются процедуры шифрования и цифровой подписи.

Сеть Internet рассчитана на одновременную работу миллионов пользователей, поэтому в коммерческих Internet-приложениях невозможно использовать только симметричные криптосистемы с секретными ключами (DES, ГОСТ28147-89). В связи с этим применяются также асимметричные криптосистемы с открытыми ключами. Шифрование с использованием открытых ключей предполагает, что у коммерсанта и покупателя имеются по два ключа – один открытый, который может быть известен третьим лицам, а другой – частный (секретный), известный только получателю информации.

Правила SET предусматривают первоначальное шифрование сообщения с использованием случайным образом сгенерированного симметричного ключа, который, в свою очередь, шифруется открытым ключом получателя сообщения. В результате образуется так называемый *электронный конверт*. Получатель сообщения расшифровывает электронный конверт с помощью своего частного (секретного) ключа, чтобы получить симметричный ключ отправителя. Далее симметричный ключ отправителя используется для расшифрования присланного сообщения.

Целостность информации и аутентификации участников транзакции гарантируется использованием электронной цифровой подписи.

Для защиты сделок от мошенничества и злоупотреблений организованы специальные центры (агентства) сертификации в Internet, которые следят за тем, чтобы каждый участник электронной коммерции получал бы уникальный электронный сертификат. В этом сертификате с помощью секретного ключа сертификации зашифрован открытый ключ данного участника коммерческой сделки. Сертификат генерируется на определенное время, и для его получения необходимо представить в центр сертификации документ, подтверждающий личность участника (для юридических лиц – их легальную регистрацию), и затем, имея "на руках" открытый ключ центра сертификации, участвовать в сделках.

Рассмотрим пример шифрования. Коммерсант Алиса хочет направить зашифрованное сообщение о товаре покупателю Бобу в ответ на его запрос. Алиса пропускает описание товара через однонаправленный алгоритм, чтобы получить уникальное значение, известное как дайджест сообщения. Это своего рода цифровой слепок с описания товара, который впоследствии будет использован для проверки целостности сообщения. Затем Алиса шифрует этот дайджест сообщения личным (секретным) ключом для подписи, чтобы создать цифровую подпись.

После этого Алиса создает произвольный симметричный ключ и использует его для шифрования описания товара, своей подписи и копии своего сертификата, который содержит ее открытый ключ для подписи. Для того чтобы расшифровать описание товара, Бобу потребуется защищенная копия этого произвольного симметричного ключа.

Сертификат Боба, который Алиса должна была получить до инициации безопасной связи с ним, содержит копию его открытого ключа для обмена ключами. Чтобы обеспечить безопасную передачу симметричного ключа, Алиса шифрует его, пользуясь открытым ключом Боба для обмена ключами. Зашифрованный ключ, который называется цифровым конвертом, направляется Бобу вместе с зашифрованным сообщением.

Наконец, она отправляет сообщение Бобу, состоящее из следующих компонентов:

- симметрично зашифрованного описания товара, подписи и своего сертификата;
- асимметрично зашифрованного симметричного ключа (цифровой конверт).

Продолжим предыдущий пример и рассмотрим процедуру расшифрования.

Боб получает зашифрованное сообщение от Алисы и прежде всего расшифровывает цифровой конверт личным (секретным) ключом для обмена ключами с целью извлечения симметричного ключа. Затем Боб использует этот симметричный ключ для расшифрования описания товара, подписи Алисы и ее сертификата. Далее Боб расшифровывает цифровую подпись Алисы с помощью ее открытого ключа для подписи, который получает из ее сертификата. Тем самым он восстанавливает оригинальный дайджест сообщения с описанием товара. Затем Боб пропускает описание товара через тот же однонаправленный алгоритм, который использовался Алисой, и получает новый дайджест сообщения с расшифрованным описанием товара.

Потом Боб сравнивает свой дайджест сообщения с тем дайджестом, который получен из цифровой подписи Алисы. Если они в точности совпадают, Боб получает подтверждение, что содержание

сообщения не изменилось во время передачи и что оно подписано с использованием личного (секретного) ключа для подписи Алисы. Если же дайджесты не совпадают, это означает, что сообщение либо было отправлено из другого места, либо было изменено после того, как было подписано. В этом случае Боб предпринимает определенные действия, например уведомляет Алису или отвергает полученное сообщение.

Протокол SET вводит новое применение цифровых подписей, а именно использование двойных цифровых подписей. В рамках протокола SET *двойные цифровые подписи* используются для связи заказа, отправленного коммерсанту, с платежными инструкциями, содержащими информацию о счете и отправленными банку [8].

Например, покупатель Боб хочет направить коммерсанту Алисе предложение купить единицу товара и авторизацию своему банку на перечисление денег, если Алиса примет его предложение. В то же время Боб не хочет, чтобы в банке прочитали условия его предложения, равно как и не хочет, чтобы Алиса прочитала его информацию о счете. Кроме того, Боб хочет связать свое предложение с перечислением так, чтобы деньги были перечислены только в том случае, если Алиса примет его предложение.

Все вышесказанное Боб может выполнить посредством цифровой подписи под обоими сообщениями с помощью одной операции подписывания, которая создает двойную цифровую подпись. Двойная цифровая подпись создается путем формирования дайджеста обоих сообщений, связывания двух сообщений вместе, вычисления дайджеста итога предыдущих операций и шифрования этого дайджеста личным ключом для подписи автора. Автор обязан включить также дайджест другого сообщения, с тем чтобы получатель проверил двойную подпись.

Получатель любого из этих сообщений может проверить его подлинность, генерируя дайджест из своей копии сообщения, связывая его с дайджестом другого сообщения (в порядке, предусмотренном отправителем) и вычисляя дайджест для полученного итога. Если вновь образованный дайджест соответствует расшифрованной двойной подписи, то получатель может доверять подлинности сообщения.

Если Алиса принимает предложение Боба, она может отправить сообщение банку, указав на свое согласие и включив дайджест сообщения с предложением Боба. Банк может проверить подлинность авторизации Боба на перечисление и дайджеста сообщения с предложением Боба, предоставленного Алисой, чтобы подтвердить двойную подпись. Таким образом, банк может проверить подлинность предложения на основании двойной подписи, но банк не сможет прочитать условия предложения.

Использование сертификатов. Альтернативой безопасной передаче ключа служит использование доверенной третьей стороны – центра сертификации (агентства по сертификатам) – для подтверждения того, что открытый ключ принадлежит именно владельцу карточки [8, 42].

Центр сертификации создает сообщение, содержащее имя владельца карточки и его открытый ключ, после предъявления владельцем карточки доказательств идентификации личности (водительские права или паспорт). Такое сообщение называется *сертификатом.* Сертификат снабжается подписью центра сертификации и содержит информацию об идентификации владельца, а также копию одного из открытых ключей владельца.

Участники протокола SET имеют две пары ключей и располагают двумя сертификатами. Оба сертификата создаются и подписываются одновременно центром сертификации.

Сертификаты владельцев карточек функционируют как электронный эквивалент кредитных карточек. Они снабжаются цифровой подписью финансового учреждения и поэтому не могут быть изменены третьей стороной. Эти сертификаты содержат номер счета и срок действия, которые шифруются с использованием однонаправленного алгоритма хэширования. Если номер счета и дата окончания действия известны, то связь с сертификатом можно подтвердить, однако эту информацию невозможно получить путем изучения данного сертификата. В рамках протокола SET владелец карточки представляет информацию о счете в тот платежный межсетевой интерфейс, где проводится данная связь.

Сертификат выдается владельцу карточки только с разрешения финансового учреждения – эмитента карточки. Запрашивая сертификат, владелец карточки указывает свое намерение использовать торговлю электронными средствами. Эти сертификаты передаются коммерсантам вместе с запросами о покупке и зашифрованными платежными инструкциями. Когда коммерсант получает сертификат владельца карточки, он может не сомневаться в том, что номер счета подтвержден финансовым учреждением.

Сертификаты коммерсантов являются электронным аналогом фирменной картинки, которая выставляется в витрине электронного магазина. Эти сертификаты снабжены цифровой подписью финансового учреждения коммерсанта и, следовательно, не могут быть изменены третьей стороной. Сертификаты служат гарантией того, что коммерсант имеет действующее соглашение с эквайером.

Коммерсант должен иметь по меньшей мере одну пару сертификатов для того, чтобы участвовать в операционной среде SET, но у одного коммерсанта может быть множество пар сертификатов – для каждого типа кредитных карточек, которые он принимает к оплате.

Сертификаты платежных межсетевых интерфейсов выдаются эквайерам или их обработчикам для систем, которые обрабатывают авторизации и получают сообщения. Ключ шифрования конкретного интерфейса, который владелец карточки получает из этого сертификата, используется для защиты информации о счете владельца карточки. Сертификаты платежного интерфейса выдаются эквайеру оператором карточек определенного типа.

Сертификаты эквайеров выдаются эквайерам для того, чтобы они могли принимать и обрабатывать запросы о сертификатах, инициированных коммерсантами. Эквайеры получают сертификаты от каждой ассоциации кредитных карточек.

Сертификаты эмитентов нужны эмитентам для того, чтобы пользоваться услугами центра сертификации, который может принимать и обрабатывать запросы о сертификатах непосредственно от владельцев карточек по открытым и частным сетям. Эмитенты получают сертификаты от ассоциации кредитных карточек.

Сертификаты SET проверяются в *иерархии доверия* (рис. 9.9). Каждый сертификат связан с сертификатом подписи того объекта, который снабдил его цифровой подписью. Следуя по "дереву доверия" до известной доверенной стороны, можно быть уверенным в том, что сертификат является действительным. Например, сертификат владельца карточки связан с сертификатом эмитента (или ассоциации по поручению эмитента), который, в свою очередь, связан с корневым ключом через сертификат ассоциации.

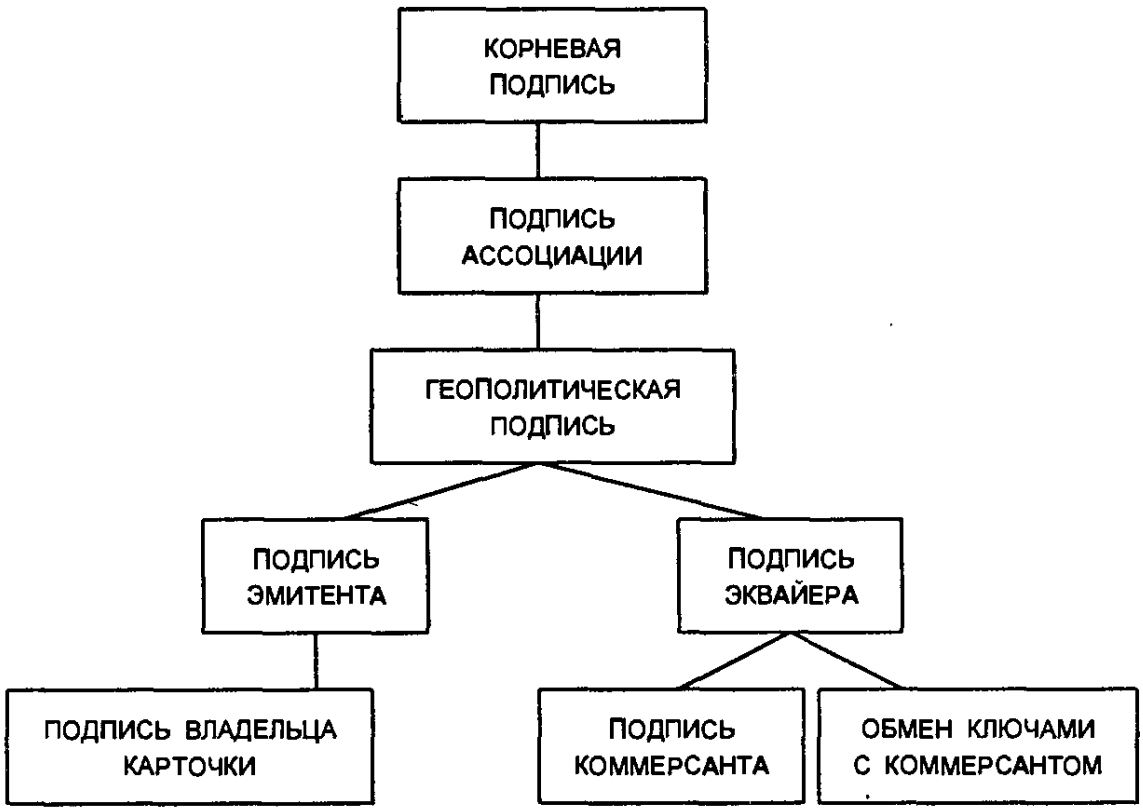


Рис. 9.9. Иерархическое дерево доверия

Открытый ключ для корневой подписи известен всем программным средствам SET и может быть использован для проверки каждого из сертификатов. Корневой ключ будет распространяться в сертификате с автоподписью. Этот сертификат корневого ключа будет доступен поставщикам программного обеспечения для включения в их программные средства.

Протокол SET определяет множество протоколов транзакций, которые используют криптографические средства для безопасного ведения электронной коммерции. Среди этих протоколов транзакций – регистрация владельца карточки, регистрация коммерсанта, запрос о покупке, авторизация платежа, получение платежа. В [8] подробно рассмотрены две транзакции – регистрация владельца карточки и авторизация платежа.

Новые достижения в области безопасности использования кредитных карточек, реализованные в стандарте SET, способны удовлетворить самых недоверчивых клиентов электронных платежных систем, поскольку устраняются все их опасения путем внедрения средств шифрования для скремблирования кредитной карточки в таком порядке, чтобы ее могли читать только продавец и покупатель.

Системы такого типа имеют ряд преимуществ.

- Деньги клиента находятся под надежным присмотром банка. Если клиент потеряет карточку, то его счет все равно связан с его именем. В отличие от систем с использованием наличности у банка есть возможность проверить остаток на счете клиента, поэтому деньги клиента не теряются.
- Отпадает необходимость в открытии нового счета. В банке для обработки транзакций данного типа клиент может продолжать пользоваться действующим счетом и кредитной карточкой. Этот фактор имеет большое значение на начальных стадиях электронной торговли в WWW сети Internet.

Однако имеется и недостаток, причем существенный – отсутствие конфиденциальности. В отличие от транзакций с электронной наличностью, которые являются анонимными, в транзакциях с кредитными картами имя клиента жестко связано со счетом.

Технологические решения для электронной торговли

В настоящее время наибольшее распространение получили два программно-аппаратных решения, предложенные компаниями Microsoft, VeriFone и Netscape.

Оба этих решения предполагают использование следующего набора компонентов [17]:

- клиентский компьютер, имеющий доступ к Internet и Web-brouser;
- сервер электронной торговли, на котором ведется каталог товаров и принимаются зашифрованные запросы клиентов на покупку тех или иных товаров;
- средство для обеспечения взаимной конвертации протоколов Internet и стандартных протоколов авторизации (ISO 8583 и др.).

Рассмотрим реализацию данной схемы на примере продуктов Microsoft (Merchant Server) и VeriFone (vPOS и vGate). Программное обеспечение vPOS устанавливается на рабочей станции клиента и осуществляет поддержку протокола SET, шифрование и аутентификацию информации, получение необходимых сертификатов и др.

Microsoft Merchant Server помимо указанных выше функций ведения каталога и приема запросов клиентов осуществляет связь с другим продуктом VeriFone–vGate. Программное обеспечение vGate, получая запросы в формате SET, расшифровывает их и конвертирует в формат ISO 8583. Таким образом, становится возможным осуществлять платежи в сети Internet с использованием обычных кредитных карт.

Следует отметить, что описанные выше решения являются по существу адаптацией технологий кредитных карт, существующих еще с 60-х годов, к современным электронным технологиям.

Альтернативный путь – внедрение концепции "чисто" электронных денег, концепции DigiCash и CyberCash. *Электронные деньги* представляют собой специальную последовательность электронных деноминаций и электронных подписей, подготовленных банками. Системы, подобные DigiCash, CyberCash и NetCash, позволяют клиентам вносить реальные деньги на банковский счет, после чего использовать эту наличность в электронной форме для приобретения различных товаров через Internet. Клиент банка заводит виртуальный электронный "кошелек", поместив в него определенную сумму денег. Клиенты системы DigiCash в качестве эквивалента любой мелкой монеты получают 64-битовый номер, который затем переводится на жесткий диск конкретного пользователя. Дальнейшая оплата товаров и услуг осуществляется перечислением соответствующей битовой информации. Клиент может перечислять эту электронную наличность продавцам в Internet (если данный продавец согласен с такой формой оплаты). Затем продавец возвращает электронную наличность банку в обмен на настоящие деньги [49, 67].

К достоинствам систем такого типа относятся:

- конфиденциальность (движение электронной наличности нельзя проследить; банк не связывает номера с каким-либо конкретным лицом, поэтому не может раскрыть инкогнито плательщика);
- гарантированная безопасность для банков (любой покупатель может потратить только ту сумму, которую он имеет на счете).

Недостатком транзакций описанного типа является то, что электронные деньги ничем не гарантированы. Например, если жесткий диск компьютера выходит из строя, или разоряется электронный банк, или хакеры расшифровывают номера электронной наличности, во всех этих случаях нет никакого способа вернуть утраченную клиентом наличность. Поскольку банк не связывает деньги с именем клиента, он не может компенсировать потери клиента.

Другим технологическим решением является система платежей с использованием смарт-карт Mondex, которую недавно приобрела компания MasterCard. В отличие от традиционных платежных систем система на основе смарт-карт Mondex предполагает эмиссию электронных денег, которые помещаются на смарт-карту и могут переписываться на другие смарт-карты, сниматься с карты в пунктах продажи и т.д. Еще одним отличием системы Mondex от других платежных систем типа "электронный кошелек" является анонимность платежей. Однако следует иметь в виду, что во многих странах законодательно запрещены анонимные платежи на крупные суммы.

В системе Mondex решены и проблемы конвертации валюты. В каждой из стран, присоединившихся к этому проекту, планируется организовать специальный банк, который будет эмитировать электронную наличность. При переводе средств из одной валюты в другую в системе организуется специальная транзакция между электронными банками двух стран. Перерасчет осуществляется по официальному курсу, а затем на карту клиента помещается действующая сумма в другой валюте.

ГЛАВА 10. ОТЕЧЕСТВЕННЫЕ АППАРАТНО-ПРОГРАММНЫЕ СРЕДСТВА КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ СЕРИИ КРИПТОН

10.1. Концептуальный подход фирмы АНКАД к защите информации в компьютерных системах и сетях

Полностью контролируемые компьютерные системы

Любая компьютерная система (КС) использует стандартное и специализированное оборудование и программное обеспечение, выполняющее определенный набор функций: аутентификацию пользователя, разграничение доступа к информации, обеспечение целостности информации и ее защиты от уничтожения, шифрование и электронную цифровую подпись и др.

Целостность и ограничение доступа к информации обеспечиваются специализированными компонентами системы, использующими криптографические методы защиты. Для того чтобы компьютерной системе можно было полностью доверять, ее необходимо аттестовать, а именно:

- определить множество выполняемых функций;
- доказать конечность этого множества;
- определить свойства всех функций.

Отметим, что в процессе функционирования системы невозможно появление в ней новой функции, в том числе и в результате выполнения любой комбинации функций, заданных при разработке. Здесь мы не будем останавливаться на конкретном составе функций, поскольку они перечислены в соответствующих руководящих документах Федерального агентства правительственной связи и информации (ФАПСИ) и Государственной технической комиссии (ГТК) России.

При использовании системы ее функциональность не должна нарушаться, иными словами, необходимо обеспечить целостность системы в момент ее запуска и в процессе функционирования.

Надежность защиты информации в компьютерной системе определяется:

- конкретным перечнем и свойствами функций КС;
- используемыми в функциях КС методами;
- способом реализации функций КС.

Перечень используемых функций соответствует классу защищенности, присвоенному КС в процессе сертификации, и в принципе одинаков для систем одного класса. Поэтому при рассмотрении конкретной КС следует обратить внимание на используемые методы и способ реализации наиболее важных функций: аутентификацию и проверку целостности системы. Здесь следует отдать предпочтение криптографическим методам: шифрования (ГОСТ 28147-89), электронной цифровой подписи (ГОСТ Р 34.10-94) и функции хэширования (ГОСТ Р 34.11-94), надежность которых подтверждена соответствующими государственными организациями.

Программная реализация функций КС. Большинство функций современных КС реализованы в виде программ, поддержание целостности которых при запуске системы и особенно в процессе функционирования является трудной задачей. Значительное число пользователей в той или иной степени обладают познаниями в программировании, осведомлены об ошибках в построении операционных систем. Поэтому существует достаточно высокая вероятность применения ими имеющихся знаний для "атак" на программное обеспечение.

Проверка целостности одних программ при помощи других не является надежной. Необходимо четко представлять, каким образом обеспечивается целостность собственно программы проверки целостности. Если обе программы находятся на одних и тех же носителях, доверять результатам такой проверки нельзя. В связи с этим к программным системам защиты от несанкционированного доступа (НСД) следует относиться с особой осторожностью.

Аппаратная реализация функций КС. Использование аппаратных средств снимает проблему обеспечения целостности системы. В большинстве современных систем защиты от НСД применяется зашивка программного обеспечения в ПЗУ или в аналогичную микросхему. Таким образом, для внесения изменений в ПО необходимо получить доступ к соответствующей плате и заменить микросхему.

В случае использования универсального процессора реализация подобных действий потребует применения специального оборудования, что еще более затруднит проведение атаки. Использование специализированного процессора с реализацией алгоритма работы в виде интегральной микросхемы полностью снимает проблему нарушения целостности этого алгоритма.

На практике для повышения класса защищенности КС функции аутентификации пользователя, проверки целостности (платы

Таблица 10.1

Аппаратные устройства криптографической защиты данных серии КРИПТОН

Наименование	Описание
КРИПТОН-4	Шифрование по ГОСТ 28147-89 (специализированным шифропроцессором "Блюминг-1"). Генерация случайных чисел. Хранение 3 ключей и 1 узла замены в шифраторе. Загрузка ключей в устройство до загрузки ОС с дискеты или со смарт-карты, минуя оперативную память ПК. Защита от НСД. Скорость шифрования до 350 Кбайт/с. Интерфейс шины ISA-8
КРИПТОН-4К/8	Функции устройства КРИПТОН-4. Более современная, чем в КРИПТОН-4, отечественная элементная база (шифропроцессор "Блюминг-1К"). Аппаратный журнал работы с устройством. Загрузка ключей с Touch-Memory. Скорость шифрования до 610 Кбайт/с. Интерфейс шины ISA-8
КРИПТОН-4К/16	Функции устройства КРИПТОН-4К/8. Функции электронного замка персонального компьютера – разграничение доступа, проверка целостности ОС. Скорость шифрования до 950 Кбайт/с. Интерфейс шины ISA-16
КРИПТОН-4/PCI	Функции устройства КРИПТОН-4К/16. Скорость шифрования до 1100 Кбайт/с. Интерфейс шины PCI Target. Возможность параллельной работы нескольких плат
КРИПТОН-7/PCI	Функции устройства КРИПТОН-4/PCI. Хранение до 1000 ключей (таблиц сетевых ключей) в защищенном ОЗУ. Управление доступом к ключам. Скорость шифрования до 1300 Кбайт/с. Интерфейс шины PCI Master/Target. Возможность параллельной работы нескольких плат
КРИПТОН-8/PCI	Функции устройства КРИПТОН-7/PCI. Хранение 32 ключей и 2 узлов замены в шифраторе, до 4000 ключей в защищенном ОЗУ. Аппаратная реализация быстрой смены ключей. Скорость шифрования до 8800 Кбайт/с. Интерфейс шины PCI Master/Target. Возможность параллельной работы нескольких плат
КРИПТОН-НСД	Шифрование по ГОСТ 28147-89 (программой из ПЗУ). Генерация случайных чисел. Защита от НСД. Загрузка ключей с дискет, смарт-карт и Touch-Memory
Специализированная сетевая плата	Размещение коммуникационных модулей внутри платы для исключения их обхода (стадия разработки)

КРИПТОН-НСД, АККОРД и др.), криптографические функции (платы КРИПТОН-4, КРИПТОН-4К/8, КРИПТОН-4К/16, КРИПТОН-4/PCI, КРИПТОН-7/PCI, КРИПТОН-8/PCI), образующие ядро системы безопасности, реализуются аппаратно (табл.10.1), все остальные функции – программно.

Для построения надежной системы защиты КС ее разработчик должен владеть возможно более полными знаниями о конкретной операционной системе (ОС), под управлением которой будет работать система. В настоящее время отечественные разработчики располагают относительно полной информацией только об одной операционной системе – DOS. Таким образом, к целиком контролируемым можно отнести КС, работающие в операционной системе DOS, или КС собственной разработки.

Частично контролируемые компьютерные системы

Именно к таким системам можно отнести современные КС, использующие ОС Windows 95/98, Windows NT, различные версии UNIX, поскольку аттестовать их программное обеспечение полностью не представляется возможным. Сегодня вряд ли кто-нибудь возьмется достоверно утверждать, что в нем отсутствуют ошибки, программные закладки недобросовестных разработчиков или соответствующих служб.

Безопасность в таких КС может быть обеспечена:

- использованием специальных аттестованных (полностью контролируемых) аппаратно-программных средств, выполняющих ряд защищенных операций и играющих роль специализированных модулей безопасности;
- изоляцией от злоумышленника ненадежной компьютерной среды, отдельного ее компонента или отдельного процесса с помощью полностью контролируемых средств.

В частично контролируемых КС использование каких-либо программно реализованных функций, отвечающих за шифрование, электронную цифровую подпись, доступ к информации, доступ к сети и т.д., становится показателем наивности администратора безопасности. Основную опасность представляет при этом возможность перехвата ключей пользователя, используемых при шифровании и предоставлении полномочий доступа к информации.

Одним из наиболее известных и надежных аппаратных модулей безопасности являются платы серии КРИПТОН, обеспечивающие как защиту ключей шифрования и электронной цифровой подписи (ЭЦП), так и неизменность их алгоритмов. Все используемые в системе ключи могут шифроваться на мастер-ключе (загружаемом в плату минуя шину компьютера) и храниться на внешнем носителе в зашифрованном виде. Они расшифровываются только

внутри платы, в которой применяются специальные методы фильтрации и зашумления для предотвращения возможности считывания ключей с помощью специальной аппаратуры. В качестве ключевых носителей используются дискеты, микропроцессорные электронные карточки (смарт-карты) и "таблетки" Touch-Memory.

В современных аппаратно-программных средствах защиты от НСД для частично контролируемых КС можно серьезно рассматривать только функции доступа к ПК, выполняемые до загрузки операционной системы, и аппаратные функции блокировки портов ПК. Таким образом, существуют широкие возможности для разработки модулей безопасности для защиты выбранных процессов в частично контролируемых системах.

Основная проблема защиты отечественных корпоративных и ведомственных сетей состоит в том, что их программное и аппаратное обеспечение в значительной степени является заимствованным, приспособленным к ведомственным нуждам и производится за рубежом. Сертификация и аттестация компонентов этих сетей очень трудоемкий процесс. За время аттестации одной системы в продажу поступает, как правило, не одна, а несколько новых версий системы или отдельных ее элементов.

Для построения защищенной сети необходимо прежде всего обеспечить защиту ее компонентов. К основным компонентам сети относятся:

- абонентские места, персональные компьютеры или терминалы клиента;
- центры коммутации пакетов, маршрутизаторы, шлюзы и сетевые экраны;
- корпоративный сервер, локальные серверы и серверы приложений;
- отдельные сегменты сетей.

Защита каждого из компонентов (как правило, компьютера) складывается из:

- исключения несанкционированного доступа к компьютеру со стороны консоли;
- разграничения доступа к ресурсам компьютера со стороны консоли;
- исключения несанкционированного доступа к компьютеру со стороны сети;
- разграничения доступа к ресурсам компьютера со стороны сети;
- обеспечения секретности используемых для защиты криптографических ключей.

Кроме того, необходимо также защитить сеть целиком от проникновения извне и каналы обмена с другими сетями.

10.2. Основные элементы и средства защиты от несанкционированного доступа

Фирма АНКАД известна на отечественном рынке как разработчик, производитель и поставщик аппаратно-программных криптографических средств защиты информации серии КРИПТОН. Традиционно они выпускались в виде устройств с минимальным программным обеспечением. Встраивание их в конечные системы осуществлялось пользователем. В настоящий момент наряду с производством и поставкой устройств фирма предлагает готовые решения: от программ абонентского шифрования и электронной подписи до защиты отдельных рабочих мест и систем в целом.

В состав средств криптографической защиты информации (СКЗИ) фирмы АНКАД включены (рис.10.1):

- устройства криптографической защиты данных (УКЗД) и их программные эмуляторы;
- контроллеры смарт-карт;
- системы защиты информации от несанкционированного доступа (СЗИ НСД);

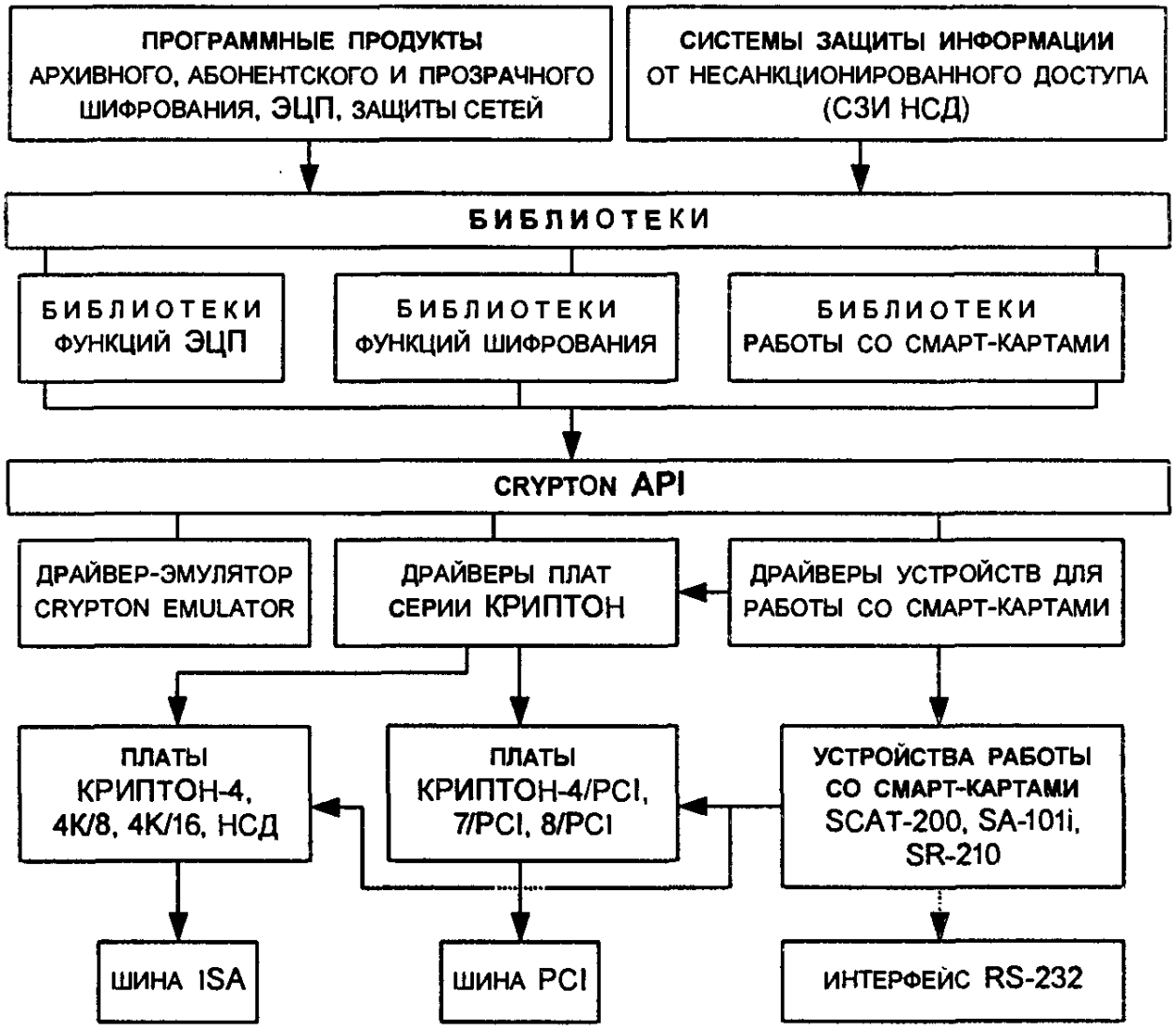


Рис.10.1. Структура средств криптографической защиты информации

- программы абонентского шифрования, электронной подписи и защиты электронной почты;
- коммуникационные программы прозрачного шифрования IP-пакетов и ограничения доступа к компьютеру по сети;
- криптомаршрутизаторы;
- библиотеки поддержки различных типов смарт-карт;
- библиотеки функций шифрования и электронной цифровой подписи для различных операционных систем.

Отдельным рядом (семейством) устройств с использованием криптографических методов защиты являются специализированные модули безопасности для терминального оборудования, контрольно-кассовых машин, банкоматов и другого оборудования, используемого в палтежных и расчетных системах.

Устройства криптографической защиты данных серии КРИПТОН

Отличительной особенностью и в этом смысле уникальностью семейства УКЗД фирмы АНКАД является разработанная ею в рамках научно-технического сотрудничества с ФАПСИ отечественная специализированная микропроцессорная элементная база для наиболее полной и достоверной аппаратной реализации российского стандарта шифрования (см. табл.10.1). В настоящее время серийно выпускаются УКЗД КРИПТОН-4, 4К/8 и 4К/16, предназначенные для шифрования по ГОСТ28147-89 и генерации случайных чисел при формировании ключей. Началось производство устройств серии КРИПТОН с интерфейсом шины PCI.

В качестве ключевых носителей применяются дискеты, смарт-карты и Touch-Memory. Все ключи, используемые в системе, могут шифроваться на мастер-ключе и храниться на внешнем носителе в зашифрованном виде. Они расшифровываются только внутри платы. Устройство может выполнять проверку целостности программного обеспечения до загрузки операционной системы, а также играть роль электронного замка персонального компьютера, обеспечивая контроль и разграничение доступа к нему.

УКЗД семейства КРИПТОН аттестованы в ФАПСИ, широко применяются в разнообразных защищенных системах и сетях передачи данных и имеют сертификаты соответствия ФАПСИ в составе ряда АРМ абонентских пунктов при организации шифровальной связи I класса для защиты информации, содержащей сведения, составляющие государственную тайну.

Для систем защиты информации от несанкционированного доступа разработана специальная плата КРИПТОН-НСД, выполняющая программное шифрование по ГОСТ28147-89, аппаратную генерацию случайных чисел, загрузку ключей с дискет, смарт-карт или Touch Memory.

Для встраивания в конечные системы пользователя УКЗД имеют два уровня интерфейса в виде набора команд устройства и библиотеки функций. Команды выполняются драйверами устройств для операционных систем DOS, Windows 95/98 и NT4.0, UNIX. Функции реализованы на основе команд.

Наиболее важными особенностями рассматриваемых плат являются:

- наличие загружаемого до загрузки операционной системы мастер-ключа, что исключает его перехват;
- выполнение криптографических функций внутри платы, что исключает их подмену или искажение;
- наличие аппаратного датчика случайных чисел;
- реализация функций проверки целостности файлов операционной системы и разграничения доступа к компьютеру;
- высокая скорость шифрования: от 350 Кбайт/с (КРИПТОН-4) до 8800 Кбайт/с (КРИПТОН 8/PCI).

Допустимо параллельное подключение нескольких устройств одновременно в одном персональном компьютере, что может значительно повысить интегральную скорость шифрования и расширить другие возможности при обработке информации.

Средства серии КРИПТОН независимо от операционной среды обеспечивают:

- защиту ключей шифрования и электронной цифровой подписи (ЭЦП);
- неизменность алгоритма шифрования и ЭЦП.

Все ключи, используемые в системе, могут шифроваться на мастер-ключе и храниться на внешнем носителе в зашифрованном виде. Они расшифровываются только внутри платы. В качестве ключевых носителей используются дискеты, микропроцессорные электронные карточки (смарт-карты) и "таблетки" Touch-Memory.

Устройства для работы со смарт-картами

Для ввода ключей, записанных на смарт-карты, предлагаются разработанные фирмой АНКАД устройства для работы со смарт-картами, функции которых приведены в табл.10.2.

Адаптер смарт-карт SA-101i предназначен для чтения и записи информации на смарт-картах. Адаптер подключается к УКЗД КРИПТОН и позволяет вводить в него ключи шифрования, хранящиеся на смарт-карте пользователя.

На одной смарт-карте могут быть размещены:

- таблица заполнения блока подстановок УЗ (ГОСТ28147-89);
- главный ключ шифрования;
- секретный и открытый ключи электронной цифровой подписи (ЭЦП) пользователя;

Таблица 10.2

Устройства для работы со смарт-картами

Наименование	Описание
SA-101i (Адаптер смарт-карт)	Запись/чтение информации на/с смарт-карты EEPROM (протокол I2C). Интерфейс с УКЗД серии КРИПТОН, обеспечивающий прямую загрузку ключей в устройство
SCAT-200 (Контроллер смарт-карт)	Шифрование по ГОСТ 28147-89, DES. Память для хранения одного мастер-ключа. Генерация случайных чисел. Запись/чтение информации на/с смарт-карты. Протоколы карт: I2C, GPM, ISO 7816 T= 0. Интерфейс RS-232 с компьютером и специализированный интерфейс с УКЗД серии КРИПТОН
SR-210 (Контроллер смарт-карт)	Запись/чтение информации на/с смарт-карты. Протоколы карт: I2C, GPM, ISO 7816 T= 0, T= 1. Интерфейс RS-232 с компьютером и специализированный интерфейс с УКЗД серии КРИПТОН

- открытый ключ ЭЦП сертификационного центра;
- идентификатор пользователя системы защиты от несанкционированного доступа КРИПТОН-ВЕТО.

Адаптер SA-101i выпускается во внутреннем исполнении и легко встраивается в персональный компьютер на свободное место, предназначенное для дисковода.

Универсальный контроллер смарт-карт SCAT-200 предназначен для работы со смарт-картами. Контроллер SCAT-200 может подключаться как к УКЗД, так и к интерфейсу RS-232. Наиболее важными представляются следующие функции контроллера:

- запись информации на смарт-карту;
- чтение информации со смарт-карты;
- шифрование по ГОСТ28147-89 и DES;
- хранение секретных ключей (так же, как в плате КРИПТОН-4);
- генерация случайной последовательности;
- набор на клавиатуре PIN-кода.

В контроллере могут применяться электронные карточки:

- открытая память (протокол I2C);
- защищенная память (серия GPM);
- микропроцессорные карты (PCOS).

Универсальный контроллер SCAT-200 позволяет строить информационные системы на базе смарт-карт, что делает его полезным для систем:

- безналичных расчетов (дебетно/кредитные карты);
- контроля доступа (хранения прав доступа);
- хранения конфиденциальной информации (медицина, страхование, финансы);
- защиты информации (хранения идентификаторов, паролей и ключей шифрования).

Контроллер может использоваться в компьютерах, электронных кассовых аппаратах, электронных замках, торговых автоматах, бензоколонках, платежных терминалах на базе IBM-совместимых компьютеров. Контроллер SCAT-200 – совместный продукт фирмы АНКАД и АО "Скантек".

Универсальный контроллер смарт-карт SR-210 имеет те же возможности, что и SCAT-200, за исключением функций шифрования и генерации случайных последовательностей. Контроллер совместим с российскими интеллектуальными микропроцессорными карточками.

Программные эмуляторы функций шифрования устройств КРИПТОН

Для программной эмуляции функций шифрования УКЗД серии КРИПТОН разработаны и применяются:

- программа шифрования Crypton LITE для работы в среде MS-DOS;
- эмулятор Crypton Emulator для ОС Windows 95/98/NT.

Программа шифрования Crypton LITE предназначена для криптографической защиты (шифрования) информации, обрабатываемой ПЭВМ типа IBM PC/XT/AT 286, 384,486, Pentium в среде MS-DOS 3.0 и выше по алгоритму ГОСТ 28147-89.

Программа Crypton LITE полностью совместима с устройствами серии КРИПТОН, обеспечивающими гарантированную защиту информации. Crypton LITE и устройства серии КРИПТОН используют общее программное обеспечение.

Программа Crypton LITE рекомендуется для применений в компьютерах, где использование устройств КРИПТОН затруднено из-за конструктивных особенностей (например, в notebook). Crypton LITE применяется не только для защиты информации в компьютерах различного конструктивного исполнения, но и как средство поддержки при написании и отладке специализированного программного обеспечения к устройствам серии КРИПТОН.

Основные характеристики программы Crypton LITE:

Алгоритм шифрования	ГОСТ 28147-89
Скорость шифрования "память-память"	до 3 Мбайт/с (для Pentium-2)
Необходимая оперативная память	2,5...8 Кбайт
Длина ключа	256 бит
Ключевая система	3-уровневая

Программа Crypton LITE реализует все режимы алгоритма ГОСТ 28147-89:

- режим простой замены;
- режим гаммирования;
- режим гаммирования с обратной связью;
- режим вычисления имитовставки (имитоприставки).

Crypton LITE имеет встроенный датчик случайных чисел, используемый для генерации ключей. В программе Crypton LITE используются следующие ключевые элементы:

К1 – первичный или файловый ключ (ключ данных), применяемый непосредственно для шифрования данных;

К2 – вторичный ключ, применяемый для шифрования первичного ключа (в зависимости от используемой ключевой системы в качестве К2 выступают пользовательский ключ или сетевой ключ);

ГК (или К3) – главный ключ (мастер-ключ), применяемый для шифрования других ключей;

УЗ – узел замены, представляющий собой несекретный элемент, определяющий заполнение блока подстановки в алгоритме шифрования ГОСТ 28147-89.

Главный ключ и узел замены называют базовыми ключами. Базовые ключи загружаются при запуске программы Crypton LITE.

Дискета пользователя, на которой записаны базовые ключи ГК и УЗ, является ключом ко всей шифруемой информации. Для ключевой дискеты должен быть обеспечен специальный режим хранения и доступа. Следует отметить, что ГК может быть защищен от злоумышленников паролем (на случай потери ключевой дискеты).

Ключи К1 и К2 могут вводиться в программу Crypton LITE в любое время. В зашифрованном виде ключи К1 и К2 могут свободно храниться на внешних носителях и передаваться по каналам связи.

Открытый программный интерфейс программы Crypton LITE позволяет внедрять ее в любые системы без затруднений, а также разрабатывать дополнительное программное обеспечение специального назначения для защиты информационных и финансовых, биржевых и банковских коммуникаций, баз данных и других массивов компьютерной информации.

Программные продукты фирмы АНКАД, совместимые с Crypton LITE, позволяют:

- прозрачно шифровать логические диски;
- разграничить доступ к компьютеру;
- осуществлять цифровую подпись электронных документов;
- передавать зашифрованную информацию по открытым каналам связи.

Программный эмулятор Crypton Emulator обеспечивает криптографическое преобразование данных по алгоритму шифрования ГОСТ 28147-89 в компьютере, работающем под управлением ОС Windows 95/98/NT. Основная задача данной программы заключается в эмуляции шифровальных функций устройств криптографической защиты данных серии КРИПТОН.

Для работы программы необходима операционная система Windows 95/98/NT 4.0. Перед установкой драйвера-эмулятора на компьютер необходимо установить программный интерфейс Crypton API версии 2.1 и выше. Никаких особых требований к компьютеру не предъявляется – драйвер-эмулятор будет работать на любом компьютере, где установлены вышеназванные ОС.

Win32-программы могут обращаться к функциям драйвера-эмулятора с помощью программного интерфейса Crypton API. Драйвер-эмулятор обеспечивает также возможность использования прерывания 0x4C в DOS-сессии Windows 95/98 или Windows NT 4.0. Драйвер-эмулятор находится на уровне ядра операционной системы, и все запросы на шифрование или расшифрование проходят через него при отсутствии в компьютере платы шифрования.

Входными данными для драйвера-эмулятора являются главный ключ (мастер-ключ) и узел замены (секретный элемент, определяющий заполнение блока подстановки в алгоритме ГОСТ 28147-89). Для инициализации драйвера-эмулятора необходимо загрузить базовые ключи ГК и УЗ с защищенной ключевой дискеты. Эта загрузка выполняется с помощью специальной утилиты, поставляемой вместе с драйвером-эмулятором. В зависимости от применяемой операционной системы обмен данными между приложением Win32 или DOS и драйвером-эмулятором ведется двумя разными способами.

Рассмотрим, в частности, особенности обмена данными в Windows NT. При обращении приложения Win32 к драйверу-эмулятору запрос от приложения Win32 проходит три уровня:

1) уровень приложений;

2) уровень, обеспечивающий интерфейс приложений с драйвером;

3) уровень ядра ОС.

Драйвер эмулирует работу платы шифрования, т.е. каждое Win32-приложение имеет собственную виртуальную плату шифрования со своими ключами К1 и К2, однако ГК и УЗ являются общими для всех приложений.

Программные продукты фирмы АНКАД, совместимые с Crypton Emulator, позволяют эффективно решать разнообразные задачи защиты информации в компьютерных системах и сетях.

10.3. Системы защиты информации от несанкционированного доступа

Система криптографической защиты информации от НСД КРИПТОН-ВЕТО

Система предназначена для защиты ПК с процессором не ниже 386, работающего под управлением MS DOS 5.0 и выше, Windows 3.1 [90]. Персональный компьютер при этом может использоваться в качестве:

- абонентского пункта;
- центра коммутации пакетов;
- центра выработки ключей.

Система ограничивает круг лиц и их права доступа к информации на персональном компьютере. Ее реализация основана на технологиях "прозрачного" шифрования логических дисков по алгоритму ГОСТ 28147-89 и электронной цифровой подписи по ГОСТ 34.10/11-94. Согласно требованиям ГТК России ее можно отнести к СЗ НСД класса 1В-1Б. (Сертификат №178 от 29 апреля 1998 г. на соответствие классу 1В, выдан ГТК при президенте Российской Федерации. Система также передана на сертификацию в ФАПСИ.)

В состав основных функций системы КРИПТОН-ВЕТО включены следующие (рис.10.2):

- обеспечение секретности информации в случае кражи "винчестера" или ПК;
- обеспечение защиты от несанкционированного включения компьютера;
- разграничение полномочий пользователей по доступу к ресурсам компьютера;
- проверка целостности используемых программных средств системы в момент включения системы;
- проверка целостности программы в момент ее запуска на выполнение;
- запрещение запуска на выполнение посторонних программ;
- ведение системного журнала, регистрирующего события, возникающие в системе;
- обеспечение "прозрачного" шифрования информации при обращении к защищенному диску;
- обнаружение искажений, вызванных вирусами, ошибками пользователей, техническими сбоями или действиями злоумышленника.

Основным аппаратным элементом системы являются серийно выпускаемые аттестованные ФАПСИ платы серии КРИПТОН, с помощью которых проверяется целостность системы и выполняется шифрование по ГОСТ 28147-89. Система предполагает наличие администратора безопасности, который определяет взаимодействие между управляемыми ресурсами:

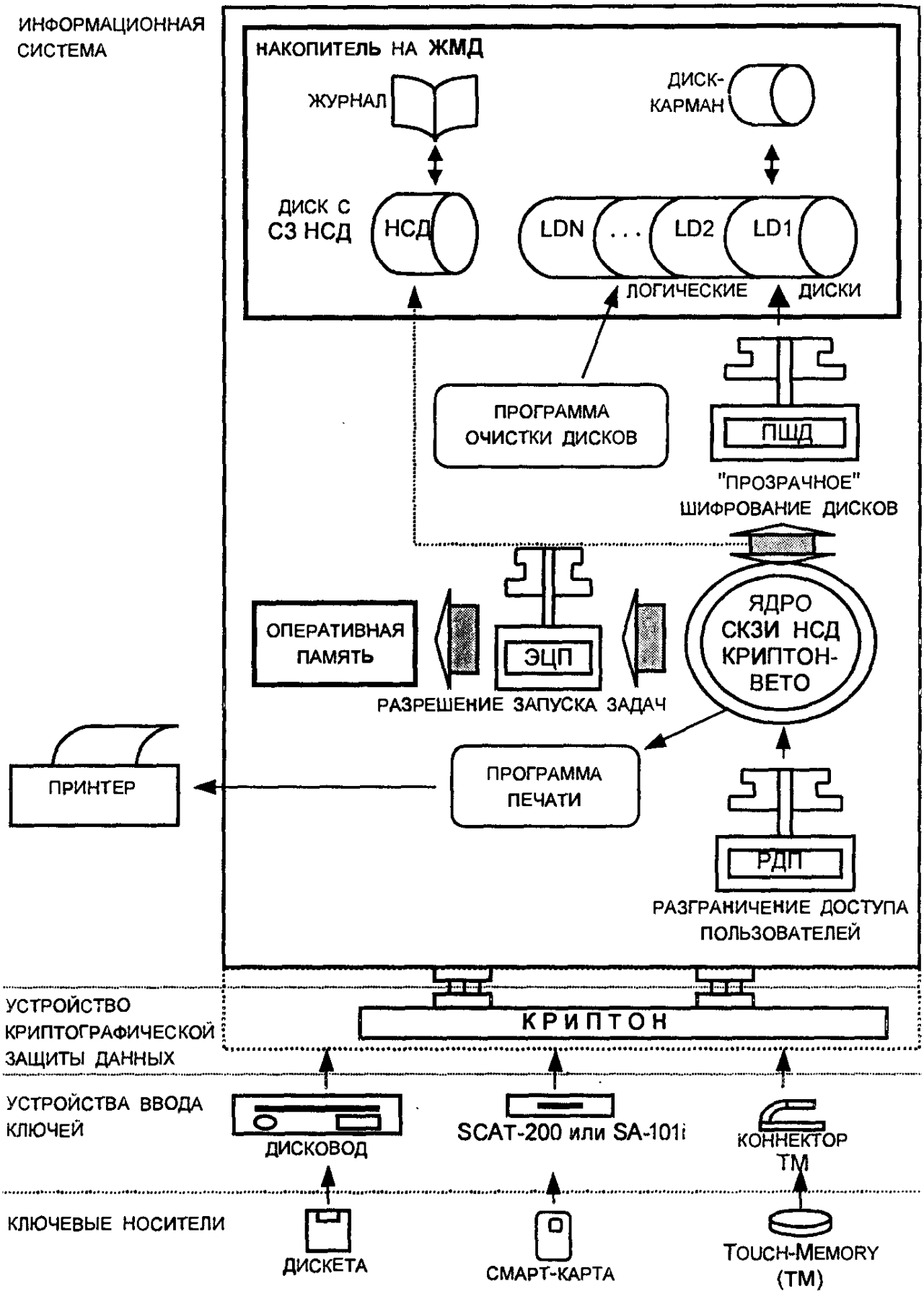


Рис.10.2. Структура системы КРИПТОН-ВЕТО

- пользователями;
- программами;
- логическими дисками;
- файлами (дискреционный и мандатный доступ);
- принтером;
- дисководами.

Система обеспечивает защиту следующим образом. Жесткий диск разбивается на логические диски. Первый логический диск (C:) отводится для размещения системных программ и данных; последний логический диск – для размещения СЗИ НСД и доступен только администратору. Остальные логические диски предназначены для хранения информации и программ пользователей. Эти диски можно разделить по пользователям и/или по уровню секретности размещаемой на них информации. Можно выделить отдельные диски с информацией различного уровня секретности (доступ к таким дискам осуществляется с помощью специальной программы, проверяющей допуск пользователя к документам-файлам). Сначала администратор устанавливает уровень секретности диска, а затем определяет круг лиц, имеющих доступ к этому диску. По форме хранения информации диски подразделяются на открытые и шифруемые; по уровню доступа – на доступные для чтения и записи, доступные только для чтения, недоступные (заблокированные).

Недоступный диск делается невидимым в DOS и, следовательно, не провоцирует пользователя на несанкционированный доступ к нему. Доступный только для чтения диск можно использовать для защиты не только от целенаправленного, но также от непреднамеренного (случайного) искажения (удаления) информации. Открытый диск ничем не отличается от обычного логического диска DOS. Очевидно, что системный диск должен быть открыт. Для шифруемых дисков используется шифрование информации в прозрачном режиме. При записи информации на диск она автоматически шифруется, при чтении с диска автоматически расшифровывается. Каждый шифруемый диск имеет для этого соответствующий ключ. Последнее делает бесполезными попытки улучшения своих полномочий пользователями, допущенными на ПК, поскольку они не имеют ключей доступа к закрытым для них дискам. Наличие шифрования обеспечивает секретность информации даже в случае кражи жесткого диска.

Для допуска к работе на ПК администратором формируется список пользователей, в котором:

- указывается идентификатор и пароль пользователя;
- определяется уровень допуска к секретной информации;
- определяются права доступа к логическим дискам.

В дальнейшем только администратор может изменить список пользователей и их полномочия.

Для исключения возможности установки на ПК посторонних программ с целью взлома защиты администратор определяет перечень программ, разрешенных к запуску на данном компьютере. Разрешенные программы подписываются администратором электронно-цифровой подписью (ЭЦП). Только эти программы могут быть запущены в системе. Использование ЭЦП одновременно с наличием разрешения позволяет отслеживать целостность запускаемых программ. Последнее исключает возможность запуска измененной программы, в том числе и произошедшего в результате непредвиденного воздействия "вируса".

Для входа в компьютер используются ключи, записанные на ключевой дискете, смарт-карте или на устройстве Touch-Memory. Ключи изготавливаются администратором системы и раздаются пользователям под расписку.

Для исключения загрузки компьютера в обход СЗ НСД загрузка осуществляется только с жесткого диска. При включении ПК (до загрузки операционной системы) с "винчестера" аппаратно проверяется целостность ядра системы безопасности КРИПТОН-ВЕТО, системных областей "винчестера", таблицы полномочий пользователей. Затем управление передается проверенному ядру системы безопасности, которая проверяет целостность операционной системы. Расшифрование полномочий пользователя, ключей зашифрованных дисков и дальнейшая загрузка операционной системы производятся лишь после заключения о ее целостности. В процессе работы в ПК загружены ключи только тех дисков, к которым пользователю разрешен доступ.

Для протоколирования процесса работы ведется журнал. В нем регистрируются следующие события:

- установка системы КРИПТОН-ВЕТО;
- вход пользователя в систему (имя, дата, время);
- попытка доступа к запрещенному диску (дата, время, диск);
- зашифрование диска;
- расшифрование диска;
- перешифрование диска;
- добавление нового пользователя;
- смена полномочий пользователя;
- удаление пользователя из списка;
- сброс причины останова системы;
- попытка запуска запрещенной задачи;
- нарушение целостности разрешенной задачи и т.д.

Журнал может просматриваться только администратором. Для проверки работоспособности системы используются программы тестирования. При необходимости пользователь может закрыть информацию на своем диске и от администратора, зашифровав последнюю средствами абонентского шифрования.

Комплекс КРИПТОН-ЗАМОК для ограничения доступа к компьютеру

Комплекс КРИПТОН-ЗАМОК предназначен для построения аппаратно-программных средств ограничения доступа к компьютеру с использованием УКЗД серии КРИПТОН. Комплекс позволяет организовать на базе персонального компьютера рабочее место с ограничением круга лиц, имеющих доступ к содержащейся в нем информации.

Для работы комплекса КРИПТОН-ЗАМОК необходим персональный компьютер IBM PC с процессором не ниже i386 и операционной системой – MS DOS, Windows 95/98/NT, UNIX и другими, для которых имеется соответствующий драйвер, позволяющий под управлением MS DOS понимать формат установленной на компьютере файловой системы.

Комплекс служит для защиты компьютеров с жесткими дисками, с файловыми системами в форматах FAT 12, FAT 16, FAT 32, NTFS, UNIX и т. д.

Работа с дисками с файловыми системами FAT 12, FAT 16 и FAT 32 обеспечивается средствами комплекса без дополнительных драйверов. Работа с дисками с нестандартными файловыми системами NTFS, HTFS, UNIX и т. д., не поддерживаемыми операционной системой MS-DOS, может производиться только при наличии на компьютере соответствующих DOS-драйверов. Комплекс КРИПТОН-ЗАМОК выпускается в двух исполнениях:

- для жестких дисков объемом менее 8 Гбайт,
- для жестких дисков объемом более 8 Гбайт.

В базовый состав аппаратно-программных средств ограничения доступа к компьютеру входят:

- УКЗД серии КРИПТОН, поддерживающие режим работы комплекса ЗАМОК;
- комплект драйверов и библиотек УКЗД;
- комплекс ЗАМОК, включающий:
  микросхему с программным обеспечением комплекса, устанавливаемую в УКЗД серии КРИПТОН;
  инсталляционный дистрибутивный носитель с программным обеспечением комплекса.

Установленный в персональный компьютер комплекс ограничения доступа КРИПТОН-ЗАМОК выполняет следующие функции:

- ограничивает доступ пользователей к компьютеру путем их идентификации и аутентификации;
- разделяет доступ пользователей к ресурсам компьютера в соответствии с их полномочиями;
- контролирует целостность ядра комплекса, программ операционной среды, прикладных программ и областей памяти в момент включения компьютера до загрузки его операционной системы;
- регистрирует события в защищенном электронном журнале;
- передает управление и параметры пользователя программному обеспечению (RUN-файлам), указанному администратором (например, ПО защиты от несанкционированного доступа).

В соответствии с выполняемыми функциями комплекс КРИПТОН-ЗАМОК содержит следующие основные подсистемы:

- подсистему управления доступом, состоящую из устройства КРИПТОН и программы обслуживания CRLOCK.EXE;
- подсистему регистрации и учета, включающую два журнала (аппаратный – на устройстве КРИПТОН, фиксирующий попытки входа в компьютер до запуска его операционной системы, и полный – на жестком диске, в котором после удачного входа в комплекс отображаются все события, в том числе и содержимое аппаратного журнала), управление которыми осуществляется программой обслуживания комплекса CRLOCK.EXE;
- подсистему обеспечения целостности, состоящую из устройства КРИПТОН и программы CHECKOS.EXE, проверяющей целостность главной ОС при работе комплекса.

При этом комплекс КРИПТОН-ЗАМОК обеспечивает выполнение следующих задач:

- в компьютер может войти только санкционированный пользователь;
- загружается достоверное ядро комплекса;
- загружается достоверная операционная система;
- проверяется целостность прикладного ПО, указанного администратором;
- производится запуск программ, указанных администратором.

Рассмотрим штатную работу комплекса КРИПТОН-ЗАМОК. В начале работы с комплексом устройство КРИПТОН при инициализации его ключами с ключевого носителя (дискеты, смарт-карты или Touch Memory) загружает три файла: UZ.DB3 (УЗ, он один для всех пользователей данного компьютера); GK.DB3 (ГК, он уникален для каждого и может быть зашифрован на пароле пользователя) и файл-паспорт пользователя INIT.NSD.

Первые два файла обеспечивают выполнение устройством КРИПТОН криптографических процедур в соответствии с ГОСТ 28147-89 и формируются при помощи любой из программ генера-

ции криптографических ключей, выпускаемых фирмой АНКАД для средств серии КРИПТОН (например, Crypton Soft, Crypton Tools или Cr Mng). Файл INIT.NSD уникален для каждого пользователя и используется при входе в комплекс для загрузки и проверки его ядра, поиска пользователя в файле полномочий, его аутентификации и расшифровки его записи. Файл INIT.NSD формируется на ключевом носителе пользователя: для администратора – автоматически программой INSTAL.EXE при установке комплекса на компьютер, а для всех остальных пользователей – администратором при помощи программ CRLOCK.EXE.

Алгоритм работы комплекса КРИПТОН-ЗАМОК включает следующие шаги:

- УКЗД КРИПТОН инициализируется файлами UZ.DB3 и GK.DB3.
- КРИПТОН загружает файл INIT.NSD и проверяет его целостность по имитовставке. В случае нарушения целостности этого файла или при его отсутствии дальнейшая загрузка компьютера не производится.
- КРИПТОН производит поиск имени вошедшего пользователя в списке пользователей. В случае отсутствия пользователя в списке дальнейшая загрузка компьютера не производится.
- КРИПТОН производит аутентификацию пользователя – проверяет имитовставку его ключа. В случае несовпадения имитовставки пользователь считается несанкционированным и дальнейшая загрузка компьютера не производится.
- КРИПТОН производит загрузку ОС комплекса ЗАМОК с Flash-диска. При загрузке автоматически стартует программа проверки целостности защищаемой ОС компьютера (далее "главной ОС") – CHECKOS.EXE.
- CHECKOS.EXE получает параметры вошедшего пользователя от устройства КРИПТОН и:
  разблокирует клавиатуру;
  проверяет целостность файл-списка;
  проверяет целостность системных областей и файлов главной ОС;
  при наличии RUN-файлов проверяет их целостность и запускает на выполнение;
  по запросу пользователя меняет пароль ключей на его носителе;
  по запросу администратора запускает программу обслуживания комплекса CRLOCK.EXE;
  при успешном завершении всех проверок CHECKOS.EXE запускает главную ОС.

После загрузки главной ОС компьютера комплекс ограничения доступа к компьютеру прекращает свою деятельность и не вмешивается в дальнейшую работу компьютера (до следующей загрузки).

Далее устройство КРИПТОН может использоваться как обычный шифратор.

Механизм RUN-файлов позволяет в процессе работы комплекса КРИПТОН-ЗАМОК запускать любые программы с предварительной проверкой их целостности. В частности, механизм RUN-файлов может быть использован при проверке файлов, находящихся на логических дисках с нестандартными файловыми системами (NTFS, HPFS, UNIX и т.д.). Другой вариант использования – запуск из под комплекса КРИПТОН-ЗАМОК любого другого программного обеспечения: системы ЗНСД, криптомаршрутизатора, операционной системы и т.д. На этой основе может быть построена система защиты персонального компьютера с требуемыми свойствами.

Система защиты конфиденциальной информации Secret Disk

Система защиты конфиденциальной информации Secret Disk разработана компанией Aladdin при участии фирмы АНКАД и предназначена для широкого круга пользователей компьютеров: руководителей, менеджеров, бухгалтеров, аудиторов, адвокатов, т.е. всех тех, кто должен заботиться о защите личной или профессиональной информации.

При установке системы Secret Disk на компьютере создаются новые логические диски, при записи на которые информация автоматически шифруется, а при чтении – расшифровывается. Работа с секретными дисками совершенно незаметна и равносильна встраиванию шифрования во все запускаемые приложения (например, бухгалтерскую программу, Word, Excel и т.п.).

В системе Secret Disk используется смешанная программно-аппаратная схема защиты с возможностью выбора, соответствующего российским нормативным требованиям криптографического алгоритма ГОСТ 28147-89 с длиной ключа 256 бит (программный эмулятор платы КРИПТОН или криптоплата КРИПТОН фирмы АНКАД).

Следует отметить, что применяемая в этой версии системы Secret Disk плата КРИПТОН сертифицирована ФАПСИ для защиты государственной тайны и поставляется по отдельному запросу фирмой АНКАД.

Система Secret Disk допускает также подключение внешнего криптомодуля того стандарта и с той длиной ключа, которую пользователь считает возможной для своих приложений.

Важная особенность системы Secret Disk заключается в том, что для доступа к защищенной информации необходим не только вводимый пользователем пароль, но и электронный идентификатор. В качестве такого идентификатора может использоваться обычный электронный ключ для параллельного порта, карточка

PCMCIA для ноутбуков или смарт-карта (в этом случае необходимо установить в компьютер специальный считыватель смарт-карт).

Система Secret Disk подключается только после того, как пользователь введет пароль и система обнаружит соответствующий идентификатор. Поэтому, если пользователь вытащит из компьютера электронный ключ, злоумышленникам не поможет даже знание пароля.

При работе в критических ситуациях (например, под принуждением) система предоставляет пользователю специальный режим входа с помощью отдельного пароля и ряд блокировок, позволяющих не раскрывать информацию (т.е. доступ к диску будет открыт, но при попытке считать с него данные или переписать их на другой диск будут генерироваться системные ошибки Windows и будет разрушено содержимое памяти электронного идентификатора, без чего невозможно расшифровать содержимое секретного диска).

Система защиты данных Crypton Sigma

Система Crypton Sigma – это программный комплекс, предназначенный для защиты данных на персональном компьютере. По своим возможностям он во многом аналогичен системе Secret Disk. Будучи установленной на компьютере, система Crypton Sigma хранит конфиденциальные данные в зашифрованном виде, не допуская несанкционированный доступ и утечку данных. Для шифрования данных в системе Crypton Sigma используется алгоритм шифрования ГОСТ 28147-89.

Система защиты конфиденциальных данных Crypton Sigma ориентирована на широкий круг пользователей компьютеров – бизнесменов, менеджеров, бухгалтеров, адвокатов и др., т.е. всех тех, кто нуждается в защите профессиональной и личной информации.

Система Crypton Sigma легко устанавливается, проста и надежна в использовании, а также полностью "прозрачна"для всех программ и системных утилит операционной системы. При установке системы Crypton Sigma на компьютере создаются новые логические диски. При записи на эти диски информация автоматически шифруется, а при считывании – расшифровывается. Этот метод прозрачного шифрования позволяет полностью снять с пользователя заботу о защите данных. Работа с защищенными дисками незаметна для пользователя и равносильна встраиванию процедур шифрования/расшифрования в запускаемые приложения. Защищенные системой диски на вид ничем не отличаются от обычных и могут использоваться в локальной или глобальной сети.

Поддерживаемые файловые системы – FAT 16, FAT 32 и NTFS. Система Crypton Sigma может работать как с УКЗД КРИПТОН, так и с его программным эмулятором. Криптографические ключи для

защиты диска хранятся на съемном носителе (дискете), а при использовании УКЗД КРИПТОН возможно хранение ключевой информации на устройстве Touch Memory или смарт-карте. Кроме того, можно использовать устройство eToken (ключевой носитель для USB-порта). Применение УКЗД КРИПТОН не позволит злоумышленнику перехватить ключи пользователя с помощью внедренных программ.

Для работы системы Crypton Sigma требуется следующая минимальная конфигурация.

Компьютер:

- IBM PC/AT, PS/2 (с процессором X486 или выше) или полностью совместимый;
- минимум 8 Мбайт оперативной памяти;
- минимум 3 Мбайт свободного дискового пространства для установки и запуска системы Crypton Sigma.

Программно-аппаратное обеспечение:

- операционная система Windows 95/98 или Windows NT 4.0;
- интерфейс Crypton API v.2.2 или выше;
- УКЗД КРИПТОН с соответствующим драйвером или его программный эмулятор.

Система Crypton Sigma специально разрабатывалась так, чтобы сделать все процедуры управления максимально простыми и ясными. Все, что должен уметь пользователь, – это создать специальный файл (контейнер) для хранения зашифрованных данных и открыть его для доступа через логический диск системы Crypton Sigma.

Контейнер – это специальный файл, создаваемый при помощи Панели Управления системы Crypton Sigma. Контейнер можно открыть для доступа через логический диск, обслуживаемый драйвером системы Crypton Sigma. Все файлы, находящиеся на этом логическом диске, хранятся в зашифрованном виде. Пользователь может создать любое количество контейнеров. Каждый контейнер имеет собственный пароль. Пользователь должен ввести этот пароль при создании контейнера и использовать его для получения доступа к тем данным, которые будут храниться в данном контейнере. Используя Панель Управления Crypton Sigma, пользователь может сменить пароль для выбранного контейнера при условии, что ему известен прежний пароль.

Схема работы системы Crypton Sigma показана на рис.10.3. Логический диск системы Crypton Sigma создается и управляется драйвером этой системы. Этот логический диск используется для записи (чтения) данных в контейнер. Работа пользователя с таким логическим диском не отличается от работы с любыми другими дисками компьютера.

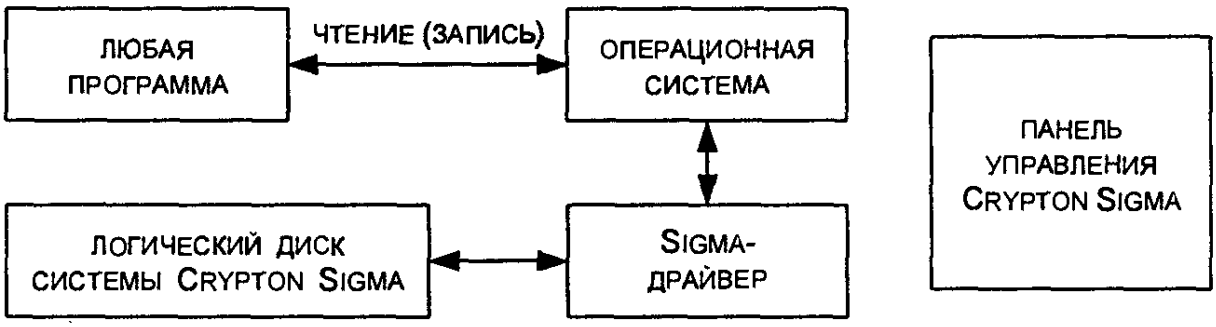


Рис.10.3. Схема выполнения операций чтения (записи) с логических дисков системой Crypton Sigma

Драйвер системы Crypton Sigma обрабатывает запросы операционной системы на чтение (запись) с логических дисков, при этом драйвер автоматически производит шифрование/расшифрование данных. Следует отметить, что драйвер системы Crypton Sigma обрабатывает не все запросы на чтение/запись. Как уже упоминалось, система Crypton Sigma создает и обслуживает собственные логические диски. Драйвер системы обслуживает операции чтения (записи) только с этих логических дисков.

Эти диски доступны точно так же, как и остальные диски на компьютере, и могут обозначаться любыми незанятыми на данный момент буквами, например D:, E:, K:, Z:.

Данные, записываемые на логический диск, фактически записываются драйвером системы в контейнер системы. Естественно, размер доступной памяти логического диска равен размеру соответствующего контейнера. Максимальный размер контейнера, создаваемого

- на жестком диске с файловой системой FAT16, равен 2 Гбайта;
- на жестком диске с файловой системой FAT 32, равен 4 Гбайта;
- на жестком диске с файловой системой NTFS, равен 512 Гбайт;
- на сетевом диске, равен 2 Гбайта.

Минимальный размер контейнера системы равен 5 Кбайт.

Для доступа к зашифрованным данным, хранящимся в контейнере, следует присоединить этот контейнер к выбранному логическому диску, например E:, и открыть его для доступа, введя соответствующий пароль. После завершения работы с данными следует закрыть этот логический диск для доступа. При этом данные, сохраненные в контейнере, станут недоступными.

Следует заметить, что если пользователь забудет пароль для доступа к данным, хранящимся в контейнере системы Crypton Sigma, то он полностью теряет возможность доступа к этим данным. Высокостойкие алгоритмы шифрования, используемые в системе Crypton Sigma, не позволяют восстановить информацию без знания пароля. Если существует опасность того, что пароль может быть забыт или утрачен, пользователь должен записать его и спрятать в надежном месте.

Отметим основные преимущества системы Crypton Sigma.

Надежная защита. Практически ни одна из существующих универсальных программ со встроенной защитой документов не имеет такой надежной защиты как Crypton Sigma. Компания Access Data (США) продает программный пакет, который вскрывает защиту данных в WordPerfect, Lotus 1-2-3, MS Excel, Symphony, Quattro Pro, Paradox, MS Word. Эта программа не просто перебирает все возможные комбинации паролей – она проводит математически обоснованный криптографический анализ – и тратит на вскрытие защищенных данных всего лишь несколько секунд. Система Crypton Sigma выгодна отличается использованием стойких и надежных алгоритмов шифрования и не содержит встроенных программных блоков, позволяющих злоумышленнику совершить несанкционированный доступ к зашифрованным данным.

Высокая степень секретности. После того как данные записываются на логический диск системы Crypton Sigma, они уже никогда не хранятся на компьютере в открытом (расшифрованном) виде. Расшифрование данных происходит только в момент доступа к ним пользователей, знающих пароль. Система Sigma нигде не хранит паролей, необходимых для доступа к данным. Она лишь проверяет, подходит ли пароль для расшифрования данных, на которые претендует пользователь, точно так же, как замок нигде не хранит дубликат ключа, а только "проверяет", может ли данный ключ открыть его или нет.

Использование системы в локальных сетях. Программное обеспечение Crypton Sigma для Windows 95/98/NT позволяет использовать любой сетевой диск для создания на нем контейнеров и доступа к хранящимся на них данным. Эти сетевые диски могут быть выделены для доступа компьютерами с любой другой, отличной от Windows, операционной системой, например ОС семейства UNIX (OSF/1, LINUX, BSD, Sun OS, AIX и др.), а также Novell, Windows 3.xx и др.

Логические диски Crypton Sigma с точки зрения операционной системы или любого другого программного обеспечения выглядят точно так же, как обыкновенные локальные диски компьютера. Поэтому логические диски Crypton Sigma могут быть открыты для доступа через локальную компьютерную сеть. Таким образом, зашифрованная информация при необходимости может быть доступна для коллективного использования.

Удобство в использовании. Система Crypton Sigma проста в использовании и, следовательно, практически не позволяет совершать случайных действий, приводящих к появлению секретной информации на компьютере в открытом виде. Пользователю необходимо только ввести правильный пароль – об остальном позаботится система. После верификации пароля доступ к зашифрованной информации становится прозрачным для всех запускаемых пользователем программ. Все зашифрованные данные автоматически расшифровываются перед тем, как появиться перед пользователем, и автоматически зашифровываются перед записью их на диски, обслуживаемые системой Crypton Sigma.

ГЛАВА 11. ЗАЩИТА ОТ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА СО СТОРОНЫ СЕТИ С ПОМОЩЬЮ СРЕДСТВ СЕРИИ КРИПТОН/CRYPTON

К основным методам защиты от несанкционированного доступа (НСД) со стороны сети относятся следующие криптографические методы:

- абонентское шифрование (АШ);
- электронная цифровая подпись (ЭЦП);
- пакетное шифрование (ПШ) (шифрование IP-пакетов или им подобных);
- криптографическая аутентификация абонентов.

Шифрование может проводиться как с открытым распределением ключей (асимметричная криптография), так и с закрытым (симметричная криптография). В любом случае используется матрица ключей для связи абонентов сети. Однако в первом случае она вычисляется на основе собственного секретного ключа абонента и базы открытых сертификатов других абонентов, во втором – генерируется заранее.

Защита информации в каналах связи производится с помощью абонентского шифрования (с применением программ Crypton Tools, Crypton Sign, Crypton Soft, Crypton ArcMail для MS-DOS и пакетов программ КРИПТОН®Шифрование, КРИПТОН®Подпись, Crypton ArcMail для Windows 95/98/NT) и прозрачного шифрования передаваемых пакетов с помощью коммуникационных программ (табл.11.1 и рис.11.1). Коммуникационные программы осуществляют также изоляцию компьютера от злоумышленника со стороны сети.

Таблица 11.1

Программные средства СКЗИ серии Crypton

Наименование	Выполнявмая функция
CryptonSoft Интегрированная криптографическая оболочка	Шифрование по ГОСТ 28147-89. Электронная цифровая подпись ГОСТ Р 34.10-94 и функция хеширования ГОСТ Р 34.11-94. Генерация ключей шифрования и ЭЦП. Генерация главных ключей и узлов замены, ключей парной связи для сети. Чтение/запись ключей с/на дискеты, смарт-карты ОС MS-DOS

Продолжение табл.11.1

Наименование	Выполняемая функция
КРИПТОН®Шифрование Шифрование данных	Шифрование по ГОСТ 28147-89. Функции генерации и хранения ключей аналогичны программе Crypton Soft. ОС Windows'95(98)/NT 4.0
CryptonSign Электронная цифровая подпись (ЭЦП)	Электронная цифровая подпись ГОСТ Р 34.10-94 и функция хеширования ГОСТ Р 34.11-94. Генерация ключей ЭЦП. Автоматическая проверка сертификатов ключей. Создание и обслуживание баз данных с открытыми ключами. Фиксеция проведенных действий в журнале. Чтение/запись ключей с/на дискеты, смарт-карты ОС MS-DOS
КРИПТОН®Подпись ЭЦП	Функции программы Crypton Sign. ОС Windows'95(98)/NT 4.0
CryptonArcMail Защита электронных документов	Функции программ CryptonSoft и CryptonSign. Сжатие информации. Контроль входа в программу. Открытое распределение ключей. Имеются версии для MS-DOS и Windows'95(98)/NT 4.0
CryptonSilent "Прозрачное" шифрование дисков	Прозрачное шифрование логических дисков по ГОСТ 28147-89. ОС MS-DOS
Crypton Sigma "Прозрачное" шифрование дисков	Прозрачное шифрование логических дисков. ОС Windows'95(98)/NT 4.0
Secret Disk "Прозрачное" шифрование дисков	Прозрачное шифрование логических дисков (совместная разработка ООО АНКАД и "Аладдин"). Для шифрования по ГОСТ 28147-89 используется криптографический модуль КРИПТОН. ОС Windows'95(98)/NT 4.0
Crypton LITE Программный шифратор	Шифрование по ГОСТ 28147-89. Генерация случайных чисел. Загрузка ключей с дискеты/жесткого диска. ОС MS-DOS
Crypton Emulator Программный шифратор	Функции программы Crypton LITE. ОС Windows'95(98)/NT 4.0
Драйверы устройств серии КРИПТОН	Поддержка функционирования плат в ОС MS-DOS, Windows 95(98)/NT, Unix
Библиотека функций Crypton API	Библиотека представляет универсальный интерфейс доступа к функциям УКЗД серии КРИПТОН. ОС MS-DOS и Windows'95(98)/NT 4.0
КРИПТОН-IP Криптографический маршрутизатор	Прозрачное шифрование IP-пакетов для передачи их в открытых сетях связи (Интернет). Защита локальных сетей от вторжения извне. Возможности построения виртуальных защищенных сетей (VPN, Virtual Privat Network). ОС MS-DOS.
Коммуникационные программы семейства ЗАСТАВА	Защита рабочих мест клиента, серверов, локальных сетей. "Прозрачное" шифрование передаваемых IP-пакетов для различных операционных систем (совместная разработка ООО АНКАД и ОАО "ЭЛВИС+"). ОС MS-DOS. Windows'95(98)/NT 4.0 и Unix
DiCrypt	Абонентское место в телекоммуникационной технологии "Дионис" (совместная разработка ООО АНКАД и НПП "Фактор"). Поддерживает функции Crypton ArcMail без выработки ключей плюс функции по работе и пересылке почты. ОС MS-DOS

Окончание табл. 11.1

Наименование	Выполняемая функция
Библиотеки функций асимметричного шифрования Crypton ArcMail	Функции шифрования и ЭЦП для различных ОС. Имеются версии для MS-DOS и Windows'95(98)/NT 4.0
Библиотека Crypton Sign DK	Библиотеки для разрабочиков приложений, использующих ЭЦП. Имеются версии для MS-DOS и Windows'95(98)/NT 4.0
Библиотека Crypton Development Kit (DK)	Библиотека функций шифрования для устройств серии КРИПТОН. ОС Windows'95(98)/NT 4.0, Unix (Solaris)
Библиотеки функций контроллеров SA-101i, SCAT-200, SR-210	От функций чтения записи ключей до функций блока безопасности в плвтежных системах с использованием электронных карточек

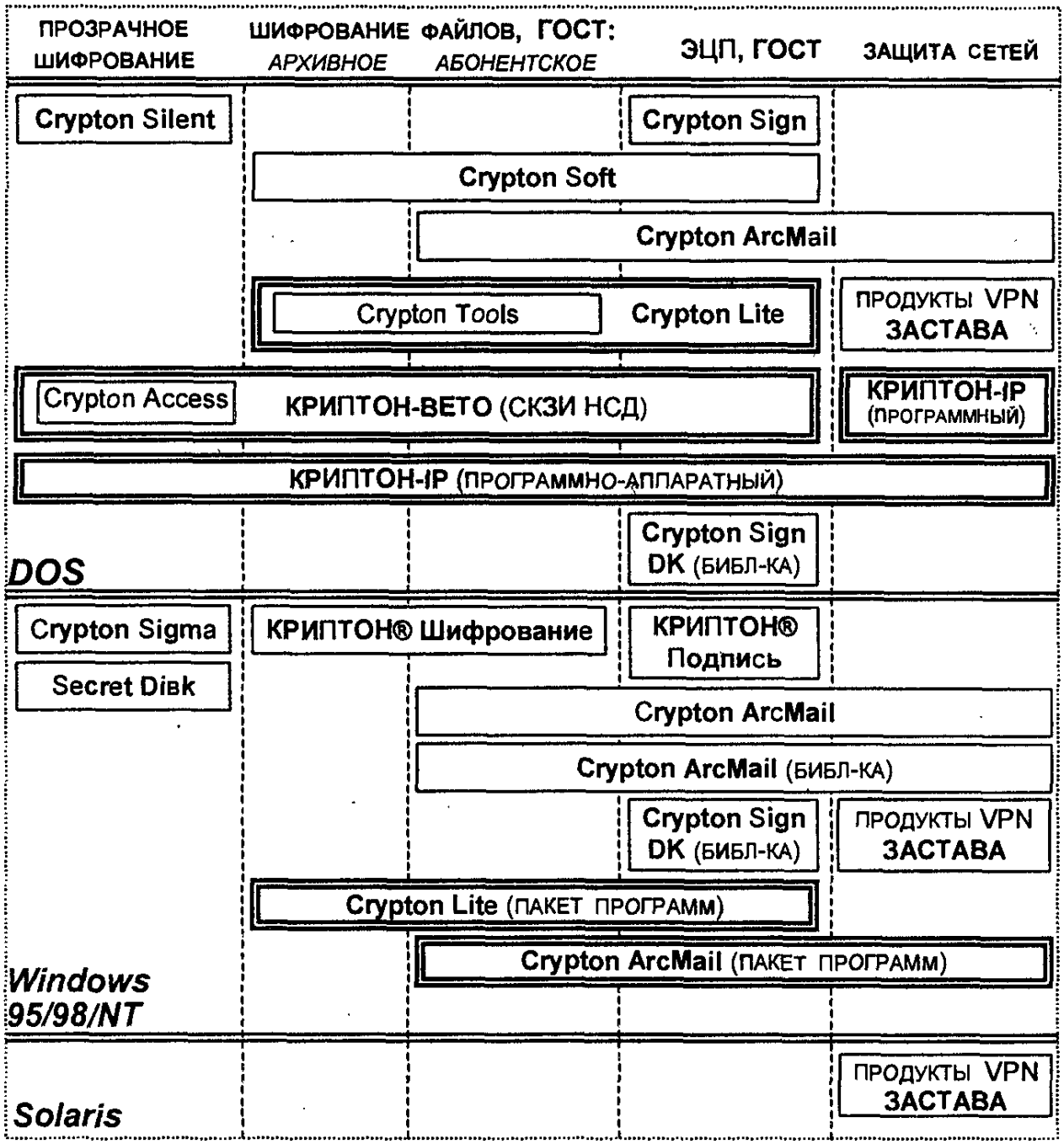


Рис.11.1. Программные средства шифрования и ЭЦП, соответствующие библиотеки и программно-аппаратные продукты фирмы АНКАД

11.1. Абонентское шифрование и электронная цифровая подпись

Для реализации абонентского шифрования (АШ) и электронной цифровой подписи (ЭЦП) может применяться отдельная программа или программно-аппаратная система, запускаемая непосредственно перед подготовкой документов к передаче или после их приема (автономный вариант). Второй вариант использования АШ и ЭЦП предусматривает включение соответствующих модулей в коммуникационные программы. В обоих вариантах система выполняет примерно одни и те же функции. Рассмотрим их на примерах реализации АШ и ЭЦП в виде отдельных программ для MS-DOS и пакетов программ для Windows 95/98/NT.

Программы АШ и ЭЦП для MS-DOS

К программным средствам абонентского шифрования и электронной цифровой подписи серии Crypton относят:

- программу симметричного шифрования и работы с ключами Crypton Tools;
- программу электронной цифровой подписи Crypton Sign;
- программу Crypton Soft для защиты файлов-документов с помощью симметричного шифрования и ЭЦП;
- программу Crypton ArcMail для защиты файлов-документов с помощью асимметричного шифрования и ЭЦП.

Для успешного функционирования каждой из этих программ компьютер должен удовлетворять следующим требованиям:

- микропроцессор 386 или выше;
- операционная система MS-DOS версии 4.0 и выше;
- не менее 350 Кбайт оперативной памяти;
- плата шифрования КРИПТОН или программа Crypton LITE.

Программа шифрования и работы с ключами Crypton Tools. Программа Crypton Tools предназначена для выполнения операций шифрования и генерации ключей. Она поставляется фирмой АНКАД в качестве базового программного обеспечения. Программа совместима "сверху вниз" с ранее поставлявшимся базовым программным обеспечением – программами CRTOOLS, CRMNG, CRBAT. Функции шифрования реализованы в соответствии со стандартом ГОСТ 28147-87. Для управления программой пользователю предоставляется интерфейс, похожий на интерфейс Norton Commander.

Шифрование файлов. В данной системе в качестве ключей могут использоваться:

- главный ключ;
- пароль;
- ключ пользователя;
- сетевой ключ.

Долговременным элементом ключевой системы алгоритма ГОСТ 28147-89 является узел замены (УЗ), который обычно хранится в файле на ключевой дискете и является первым ключевым элементом, вводимым в устройство шифрования при инициализации. Все компьютеры, между которыми предполагается обмен зашифрованной информацией (например, локальная сеть), должны использовать один и тот же УЗ, так как несоответствие узлов замены приведет к невозможности расшифрования файлов с другой машины. УЗ создается администратором.

Главный ключ (ГК) представляет собой секретный ключ, используемый для шифрования других ключей. ГК может быть зашифрован на пароле. ГК создается администратором.

Пароль – последовательность символов, вводимых с клавиатуры. Пароль защищает ключи от несанкционированного использования в случае их хищения или потери. Максимальная длина пароля для ключей шифрования – 37 символов, минимальная длина – 4 символа. Длина пароля определяет стойкость системы. Поэтому рекомендуется использовать длинные пароли с неповторяющимися символами.

Ключ пользователя (ПК) – секретный ключ, используемый для шифрования файлов и других ключей. Создается пользователем и защищает его данные от посторонних лиц, включая администратора.

Сетевой ключ – секретный ключ, используемый для шифрования файлов с целью передачи их между узлами "криптографической" сети.

Все узлы сети нумеруются. Для каждого узла, с которым планируется обмен информацией, необходимо иметь свой сетевой ключ.

Для обмена зашифрованной информацией между N узлами необходимо N*(N – 1) ключей (каждый узел с каждым). Эти ключи можно разместить в *сетевой таблице,* которая представляет собой таблицу-матрицу. В заголовках строк и столбцов этой таблицы-матрицы представлены номера узлов, а в ячейках таблицы хранятся ключи. Эта таблица-матрица симметрична, т.е. ключ для передачи от узла А к узлу Б (сетевой ключ А-Б) равен сетевому ключу Б-А.

Из полной сетевой таблицы можно для каждого из узлов сформировать *сетевой набор* ключей для связи с другими узлами. Такой сетевой набор представляет собой одну из строк таблицы. Сетевой набор хранится в файле NNNNN.SYS в каталоге сетевых ключей, где NNNNN – номер данного узла. Он всегда зашифрован на ключе сетевого набора (КСН), хранящемся в файле NNNNN.KEY в каталоге сетевых ключей. КСН получают вместе с сетевым набором от администратора криптографической сети.

Для обеспечения защиты системы шифрования ГОСТ 28147-89 от навязывания ложных данных применяется имитовставка (имитоприставка). *Имитовставка* представляет собой отрезок информации фиксированной длины, получаемый из открытых данных и ключа. Имитовставка создается при зашифровании данных и добавляется к ним. При расшифровании данных также вычисляется имитовставка и сравнивается с хранимой. В случае несовпадения можно выделить следующие причины:

- изменен УЗ;
- изменен ключ, на котором были зашифрованы данные;
- изменены зашифрованные данные;
- если при зашифровании использовался пароль, то при расшифровании он был неверно введен;
- неисправно устройство шифрования.

Шифрование файлов может проходить по двум схемам:

- архивное шифрование файлов (обмен которыми не предполагается);
- шифрование файлов для передачи в криптографической сети.

Архивное шифрование файлов. При архивном шифровании файлов сначала генерируется так называемый файловый (или сеансовый) ключ – последовательность из 256 бит, получаемая с датчика случайных чисел устройства шифрования. Вся информация, содержащаяся в файле, шифруется на данном файловом ключе. Поскольку расшифрование файла без этого файлового ключа невозможно, то он записывается в зашифрованный файл. При этом файловый ключ шифруется на ключах, указанных пользователем, с вычислением имитоприставки. Применение для шифрования файлового ключа позволяет увеличить криптографическую устойчивость реализованного механизма шифрования, а также существенно ускорить операцию перешифрования, поскольку исчезает необходимость осуществлять перешифрование всего файла, достаточно лишь перешифровать файловый ключ.

Шифрование файлов для передачи в криптографической сети. При шифровании файлов для передачи в криптографической сети файл данных, передаваемый узлом А узлу Б, зашифровывается на файловом (сеансовом) ключе. Файловый ключ создается автоматически при зашифровании файла данных и передается вместе с ним. Так как файловый ключ не может передаваться в открытом виде, то он зашифровывается на сетевом ключе А-Б. Этот ключ узел А берет из своего сетевого набора. Сетевой набор узла А зашифрован на ключе сетевого набора узла А, который, в свою очередь, тоже может быть зашифрован на каком-либо ключе узла А (как правило, ГК). Узел Б по информации, заключенной в зашифрованном файле, понимает, что файл пришел от узла А.

Используя свои ключи, узел Б сначала расшифровывает свой КСН. Затем, используя КСН, узел Б расшифровывает свой набор и достает из него сетевой ключ А-Б. Так как этот сетевой ключ совпадает с тем сетевым ключом, который был использован узлом А для зашифрования, узел Б может расшифровать файловый ключ, пришедший вместе с файлом. Наконец, с помощью файлового ключа расшифровывается пришедший файл.

Перешифрование информации выполняется следующим образом. Из зашифрованного файла извлекается зашифрованный файловый ключ и расшифровывается. Затем производится его зашифрование на новой ключевой информации, предоставляемой пользователем. При этом файловый ключ (в расшифрованном виде) остается неизменным, что позволяет оставить тело зашифрованного файла без изменений. В результате получается, что перешифрование файла – операция значительно более быстрая, чем шифрование или расшифрование файла.

Для обеспечения *целостности информации* при расшифровании файловых ключей производится проверка имитоприставки. Если она не совпала с хранимой в файле, то система выдает сообщение об ошибке. Следует отметить, что при расшифровании информации самих файлов проверка целостности данных не производится. Если зашифрованная информация была изменена, никаких диагностических сообщений выдаваться не будет, но получаемый после расшифрования файл не будет эквивалентен исходному.

При *зашифровании информации на пароле*, а также при расшифровании ключей и файлов, зашифрованных с использованием пароля, производится запрос пароля. Если пароль запрашивается для зашифрования объекта (файла или ключа), то пользователю предоставляется запрос на ввод пароля. При этом пароль необходимо ввести дважды, что уменьшает вероятность опечатки. Если пароль запрашивается для расшифрования объекта, то пользователю предоставляется диалог запроса пароля с одним полем ввода. При неправильном вводе пароля выдается сообщение о неверном пароле, и запрос пароля повторяется до тех пор, пока пользователь не введет верный пароль или откажется от ввода пароля.

В случае операций над несколькими файлами последний введенный пароль запоминается в оперативной памяти (ОП) до окончания операции, что избавляет от необходимости вводить один и тот же пароль для каждого файла. По окончании операции пароль стирается из ОП.

Зашифрование файлов производится системой в диалоговом режиме работы с пользователем. При этом пользователь должен выбрать (отметить) файлы, подлежащие шифрованию, затем вы-

брать ключевую систему шифрования, ввести пароль и ключ пользователя. При выполнении операции зашифрования для каждого выбранного файла будут последовательно выполняться следующие действия:

- генерируется файловый ключ;
- файл шифруется на данном файловом ключе;
- файловый ключ шифруется на указанной пользователем ключевой системе с вычислением имитоприставки;
- в файл записывается информация, необходимая для последующего расшифрования: старое имя файла, имена ключей и т. д.

Расшифрование файлов производится аналогичным образом в диалоговом режиме работы системы с пользователем. Система расшифровывает считанный файловый ключ. Если при этом необходим пароль, он запрашивается у пользователя. При помощи восстановленного файлового ключа зашифрованная информация расшифровывается и записывается в файл с тем же именем, что и до зашифрования.

Создание ключей шифрования. Программа Crypton Tools позволяет выполнить следующие операции:

- генерацию узла замены;
- генерацию главного ключа;
- смену пароля ключа;
- генерацию ключа пользователя;
- генерацию сетевой таблицы;
- генерацию сетевого набора;
- перешифрование ключей шифрования;
- создание ключевой дискеты.

Указанные действия с ключами осуществляются системой в диалоге с пользователем. Пользователь должен ввести в диалоговом окне информацию, описывающую ключ или набор ключей.

Следует отметить, что смену УЗ рекомендуется проводить только в самых экстренных случаях. Смена УЗ требует расшифрования всей зашифрованной на скомпрометированном УЗ информации и зашифрования с новым УЗ. Поскольку УЗ должен быть одинаков для всех абонентов, ведущих обмен зашифрованной информацией, эту работу придется проделать всем пользователям защищаемого контура. Поэтому рекомендуется проводить эту операцию только один раз при установке системы.

При создании главного ключа необходимо использовать пароль. Этот пароль применяется для закрытия главного ключа.

Для генерации ключей используется датчик случайных чисел устройства шифрования.

Программа электронной цифровой подписи Crypton Sign. Программа Crypton Sign предназначена для формирования и проверки электронной цифровой подписи электронных документов, которая обеспечивает установление авторства электронных документов и проверку целостности электронных документов. В программе Crypton Sign реализованы алгоритмы цифровой подписи и функции хэширования ГОСТ Р 34.10-94, ГОСТ Р 34.11-94.

Электронная цифровая подпись представляет собой последовательность байтов, помещаемую в конец подписываемого документа (файла) или в отдельный файл. ЭЦП формируется на основании содержимого документа, секретного ключа и пароля лица, подписывающего документ (файл). Для каждого секретного ключа создается открытый ключ для проверки подписи.

Подписывание документа-файла состоит в вычислении с помощью программы по содержимому файла некоторого большого числа (512 или 1024 бита), которое и является его электронной подписью. Важной особенностью электронной подписи является невозможность ее подделывания без секретного ключа.

Программа проверки на основании анализа содержимого документа-файла, электронной подписи и открытого ключа удостоверяет, что подпись вычислена именно из этого документа-файла конкретной программой подписывания.

В качестве подписываемого электронного документа в программе может использоваться любой файл. При необходимости несколько владельцев могут подтвердить достоверность документа, т.е. один документ-файл может содержать несколько подписей. При этом не изменяются ни имя файла, ни его расширение. Подписанный файл показан на рис.11.2.

Рис.11.2. Схема подписанного файла

В подпись записывается следующая информация:

- дата формирования подписи;
- срок окончания действия открытого и секретного ключей;
- информация о лице, сформировавшем подпись (Ф.И.О., должность, краткое наименование фирмы);
- идентификатор подписавшего (имя файла открытого ключа);
- собственно код ЭЦП.

Электронная цифровая подпись может быть записана также в отдельный файл. Это файл, имеющий имя, соответствующее подписанному файлу, и расширение sg*. В данном файле хранится вся вышеуказанная информация, а также имя файла, который был

подписан. При таком способе простановки ЭЦП исходный файл не изменяется, что может быть полезно в случаях, когда документы-файлы обрабатываются программами пользователя, не допускающими посторонней информации в конце документов.

Схема создания и проверки ЭЦП с помощью программы Crypton Sign показана на рис.11.3. Для формирования и последующей проверки подписи необходимо создать два ключа подписи: секретный и открытый. Ключи представляют собой обычные файлы на дискете или последовательность байтов на электронной карточке.

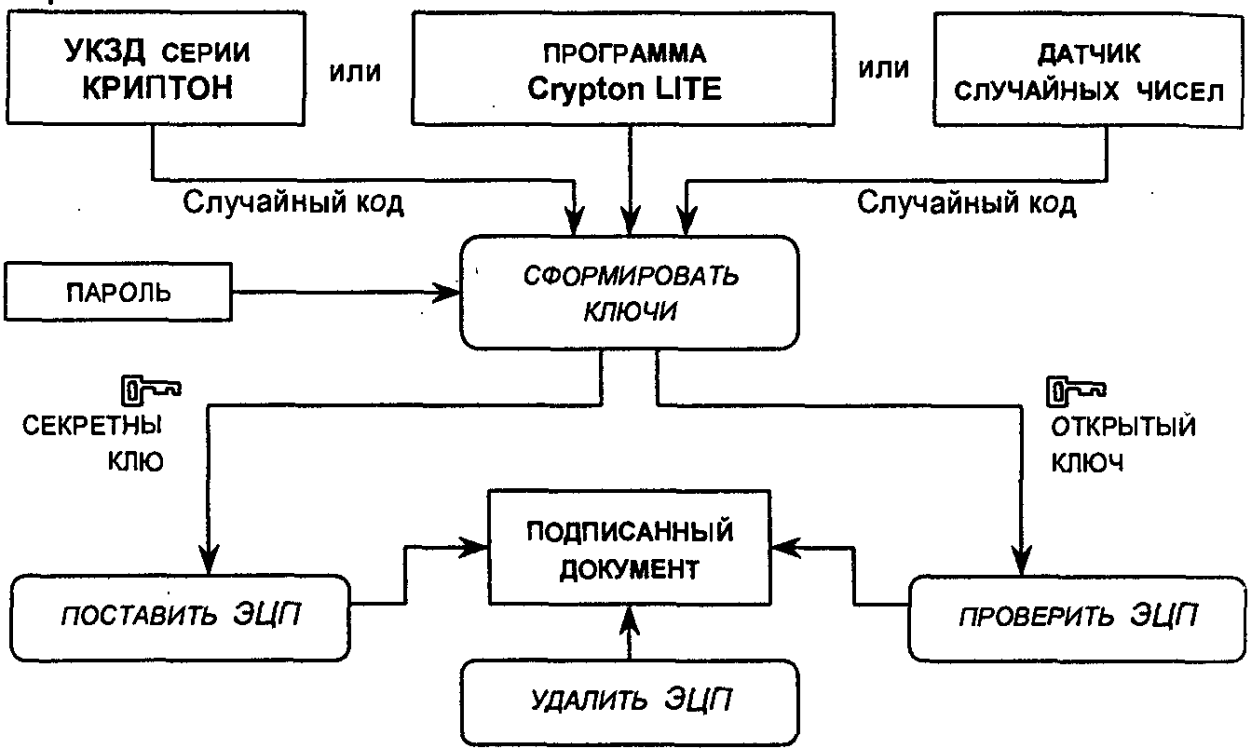


Рис.11.3. Схема создания и проверки электронной цифровой подписи

Генерация случайного кода для создания ключей выполняется аппаратно с помощью одного из УКЗД серии КРИПТОН. Если УКЗД в компьютере нет, случайный код можно получить программно с помощью программы Crypton LITE или генератора случайных чисел.

Для управления программой Crypton Sign пользователю предоставляется интерфейс, похожий на интерфейс Norton Commander. Основное меню программы Crypton Sign разделено на две части (панели). В левой части меню расположены наименования команд, выполняемых программой, в правой части – перечень файлов и раздел, в котором находятся эти файлы. Для выбора команд и файлов используется маркер.

Для генерации ключей достаточно выполнить команду "Создать ключи".

Для подписи файла необходимо выбрать сам подписываемый файл и секретный ключ, а затем выполнить команду "Поставить подпись".

Две команды –"Показать подпись" и "Проверить подпись" используются для проверки наличия и подлинности подписей у файла, а также получения дополнительной информации о подписи. Для выполнения данных команд следует выбрать проверяемые файлы и указать каталог с открытыми ключами.

При необходимости можно удалить последнюю подпись, группу последних подписей или все подписи у файла. Для этого в программе используется команда "Удалить подпись". Для ее выполнения должны быть указаны документы-файлы, у которых удаляются подписи.

Программа Crypton Soft. Интегрированная программная оболочка Crypton Soft представляет собой систему защиты файлов-документов на персональном компьютере. Программа Crypton Soft предназначена для выполнения операций шифрования, электронной цифровой подписи файлов и работы с ключами. Эта программа совместима "сверху-вниз" с программой шифрования и генерации ключей Crypton Tools 3.X и программами электронной подписи CR SIGN 1.X – CR SIGN 2.X.

Программа CryptonSoft v1.2 обеспечивает:

- шифрование файлов по ГОСТ 28147-89 (симметричные архивный и сетевой методы);
- электронную цифровую подпись файлов (существует возможность формировать подпись как внутри подписываемого файла, так и в отдельном файле);
- управление ключами (шифрования, подписи, главными ключами), заключающееся в создании ключей, их перешифровании, смене паролей;
- копирование, перенос, переименование, удаление файлов непосредственно из оболочки программы; при удалении файлов осуществляется полное их затирание с невозможностью дальнейшего восстановления;
- работу с устройствами чтения пластиковых карт типа SA-101, SCAT-200. Эти устройства представляются в виде логического устройства SC:, с которым можно производить необходимые операции: создание/чтение ключей, копирование файлов.

Программа имеет оконный пользовательский интерфейс и развернутую контекстную помощь. Управление может вестись как с клавиатуры, так и манипулятором "мышь". Реализован также режим командной строки, что позволяет использовать программу в пакетном режиме и в других системах.

Программа имеет широкий набор настраиваемых параметров: цвета интерфейса, язык сообщений, подключенные устройства, используемые внешние программы и правила отображения файлов, параметры шифрования/подписи. Кроме того, возможен запрет на использование ряда команд программы. Все настройки сохраняются в конфигурационном файле, который шифруется для предотвращения несанкционированной модификации.

Программа Crypton ArcMail. Предназначается для защиты файлов-документов, передаваемых в сети. Позволяет закрыть обмен информацией между:

- отдельными абонентами (кабинетами, пользователями и т.д.);
- различными подразделениями (отделами и т.д.);
- различными управлениями (департаментами) и т.д.

Программа работает на ПК с процессором 386 и выше под управлением DOS 5.0 и выше. Интерфейс пользователя подобен интерфейсу Norton Commander. Стандартная конфигурация системы включает:

- программу центрального пункта, обеспечивающую регистрацию абонентов и создание и сопровождение списков зарегистрированных абонентов;
- программу абонентского пункта;
- при необходимости систему защиты ПК от несанкционированного доступа КРИПТОН-ВЕТО;
- при необходимости устройство считывания информации со смарт-карт и Touch-Memory.

Программа обеспечивает сжатие документов, аутентификацию автора, целостность документов, конфиденциальность передаваемой информации. В минимальной конфигурации представляет собой программу обработки документов перед передачей и после приема. Передача и прием осуществляются стандартными программами электронной почты.

Для работы с системой каждый абонент снабжается секретным и открытым ключами. Секретный ключ абонента хранится на дискете или смарт-карте и запрашивается при запуске программы абонентского или центрального пунктов. Открытый ключ абонента направляется на регистрацию (сертификацию) на центральный пункт сети. В регистрационном центре открытые ключи всех абонентов с сертификатами помещаются в базу данных зарегистрированных абонентов. В процессе обработки полученного по почте документа автоматически проверяется наличие абонента в базе зарегистрированных абонентов. Целостность документов подтверждается электронной подписью, помещаемой в конец передаваемых документов. При формировании подписи используется текст документа и секретный ключ абонента.

При подготовке документов к передаче автоматически осуществляются:

- запрос из базы данных открытых ключей зарегистрированных абонентов, которым направляются документы;
- электронная подпись передаваемых документов;
- сжатие документов в один файл;
- генерация сеансового ключа;
- шифрование файла с документами на сеансовом ключе;
- вычисление парносвязных ключей (на основе секретного ключа отправителя и открытых ключей получателей) и шифрование на них сеансового ключа.

Таким образом, прочитать данный документ может только абонент, обладающий соответствующим секретным ключом. После приема автоматически выполняются обратные действия:

- вычисление парносвязного ключа (на основе секретного ключа получателя и открытого ключа отправителя с автоматической проверкой сертификата) и расшифрование сеансового ключа;
- расшифрование файла с помощью полученного сеансового ключа;
- разархивирование документов;
- проверка электронной подписи полученных документов.

Все действия программы-архиватора и результаты протоколируются в специальном журнале в зашифрованном виде.

Программа легко и надежно настраивается на любую комбинацию своих команд для различных абонентов. В системе можно реализовать уровень защиты, исключающий несанкционированное ознакомление с передаваемой информацией даже при получении злоумышленником секретных ключей и паролей. В этом случае приведенная выше схема защиты несколько изменяется.

Пакеты программ АШ и ЭЦП для Windows 95/98/NT

К программным средствам абонентского шифрования и электронной цифровой подписи серии КРИПТОН/Crypton для Windows 95/98/NT можно отнести следующие пакеты программ:

- пакет "КРИПТОН ® Шифрование";
- пакет "КРИПТОН ® Подпись";
- пакет программ Crypton ArcMail для Windows'95(98)/NT 4.0.

Для успешного функционирования этих пакетов программ компьютер должен иметь:

- операционную систему Windows 95/98 или Windows NT 4.0;
- УКЗД серии КРИПТОН с соответствующим драйвером для Windows 95/98/NT или его программный драйвер-эмулятор для Windows – Crypton Emulator версии 1.3 или выше;
- Crypton API для Windows 95/NT версии 2.2 или выше;
- манипулятор мышь.

Для осуществления более надежной защиты рекомендуется вместо программы Crypton Emulator использовать УКЗД серии КРИПТОН.

Пакет КРИПТОН®Шифрование. Пакет предназначен для защиты электронных документов (файлов) от несанкционированного доступа при хранении их на персональном компьютере или передаче по открытым каналам связи. Защита документов осуществляется путем их шифрования по ГОСТ 28147-89.

Для считывания ключевой информации могут применяться устройства чтения смарт-карт типа SA-101i и др.

В данной системе в качестве ключей шифрования могут использоваться: главный ключ; пароль; ключ пользователя; сетевой ключ.

Как и в программах Crypton Tools и Crypton Soft, в данной системе шифрование файлов может протекать по двум схемам:

- архивное шифрование файлов (обмен которыми не предполагается);
- шифрование файлов для передачи в криптографической сети.

Перешифрование информации, контроль целостности информации и ввод пароля осуществляются в данной системе аналогично программам Crypton Tools и Crypton Soft.

Пакет "КРИПТОН®Шифрование" состоит из трех компонентов.

1. Программа управления ключами шифрования "Мастер ключей шифрования" позволяет создавать все виды используемых ключей, менять их характеристики, а также изменять настройки всего комплекса.

2. Расширение Windows Explorer (Проводника Windows) для шифрования файлов обеспечивает добавление дополнительных команд в контекстное меню (а также в меню "Файл") программы Windows Explorer. Эти команды обеспечивают основные криптографические операции над файлами.

3. Утилита командной строки для пакетной обработки файлов позволяет автоматизировать процесс шифрования файлов, а также легко встраивать функции шифрования в клиентские системы путем вызова данной утилиты с параметрами, передаваемыми в командной строке.

Пользовательские программы содержат интерактивную справку с подробной информацией о всех выполняемых командах и выдаваемых диалоговых окнах.

Все программы пакета могут использовать смарт-карты в качестве носителей ключей. Поскольку они работают через SCApi (универсальный интерфейс смарт-карт), ограничения на тип устройств чтения смарт-карт и на тип используемых карт накладываются только текущим установленным набором драйверов карточных устройств.

Все программы данного пакета имеют ряд общих параметров, влияющих на их работоспособность. Эти параметры сохраняются в реестре Windows и являются персональными для каждого локального пользователя Windows. Параметры могут быть изменены при помощи программы "Мастер ключей шифрования".

Программы данного пакета многоязычны. Это значит, что пользователь может выбрать язык пользовательского интерфейса по своему предпочтению. На данный момент существуют два языковых варианта: русский и английский.

Для *управления ключевой информацией* используется программа "Мастер ключей шифрования". Программа предлагает пользователю диалоговое окно с двумя панелями. Левая панель содержит список доступных команд, иерархически оформленный в виде дерева. Правая панель отображает параметры выбранной команды и содержит элементы управления для ее выполнения. Для любой команды доступна интерактивная справка, вызываемая по нажатию кнопки "Справка". Программа "Мастер ключей шифрования" позволяет выполнить следующие операции:

- генерацию узла замены;
- генерацию главного ключа;
- генерацию ключа пользователя;
- смену пароля ключа;
- генерацию сетевой таблицы;
- генерацию сетевых наборов;
- перешифрование главного ключа;
- перешифрование ключа пользователя на новом главном ключе;
- перешифрование ключей пользователя.

Обработка файлов в интерактивном режиме ведется при помощи программы-расширения Windows Explorer (Проводника Windows). Эта программа позволяет выполнить:

- шифрование файлов;
- расшифрование файлов;
- перешифрование файлов;
- уничтожение файлов;
- получение информации о файлах.

Обработка файлов в пакетном режиме производится путем вызова утилиты командной строки. Обработка файлов в пакетном режиме необходима для автоматизации процесса шифрования файлов, а также для встраивания функций шифрования в другие системы. Данной программой выполняются следующие команды:

- зашифровать файлы;
- расшифровать файлы;
- перешифровать файлы;
- уничтожить файлы.

Пакет "КРИПТОН® Подпись". Этот пакет программ предназначен для формирования и использования электронной цифровой подписи электронных документов, которая обеспечивает установление авторства электронных документов и проверку целостности электронных документов. В пакете программ "КРИПТОН® Подпись" реализованы стандартные алгоритмы цифровой подписи ГОСТ Р 34.10-94 и функции хэширования ГОСТ Р 34.11-94.

Авторство и целостность электронных документов подтверждаются цифровой подписью, помещаемой в конец подписываемых документов-файлов. При формировании цифровой подписи используется текст документа и секретный ключ. Пакет КРИПТОН® Подпись, как и программа Crypton Sign, допускает два способа объединения ЭЦП с подписываемым документом-файлом:

1) размещение ЭЦП вместе с подписываемым файлом;

2) размещение ЭЦП на отдельном файле, с указанием имени подписанного файла.

Пакет "КРИПТОН® Подпись" и программа Crypton Sign имеют одинаковую структуру цифровой подписи.

Для штатной работы с программами пакета "КРИПТОН® Подпись" каждый пользователь, предполагающий использовать ЭЦП в электронном документообороте, должен иметь пару ключей – секретный и открытый.

Секретный ключ пользователя является именно тем ключевым элементом, с помощью которого формируется ЭЦП данного пользователя, поэтому ключевой носитель (дискета, смарт-карта и т.д.), содержащий данный ключ, должен храниться пользователем особо тщательно. Этим предотвращается подделка его подписи. Секретный ключ запрашивается программами пакета перед выполнением каких-либо действий, поэтому, не имея секретного ключа ЭЦП, войти в программы пакета невозможно.

Открытые ключи подписи используются для проверки ЭЦП получаемых документов-файлов. Владелец пары ключей подписи должен обеспечить наличие своего открытого ключа у всех, с кем он собирается обмениваться подписанными документами. При этом следует исключить возможность подмены открытых ключей как на этапе передачи, так и на этапе их использования.

При работе с пакетом "КРИПТОН® Подпись" каждый пользователь должен иметь как минимум один собственный секретный ключ для формирования своей подписи и множество открытых ключей для проверки чужих подписей. Понятно, что собственный секретный ключ должен быть недоступен для других. Открытые ключи должны быть сертифицированы для предотвращения опасности их подмены.

Рассмотрим совокупность действий по защите ключей. Схема работы с ключами представлена на рис.11.4.

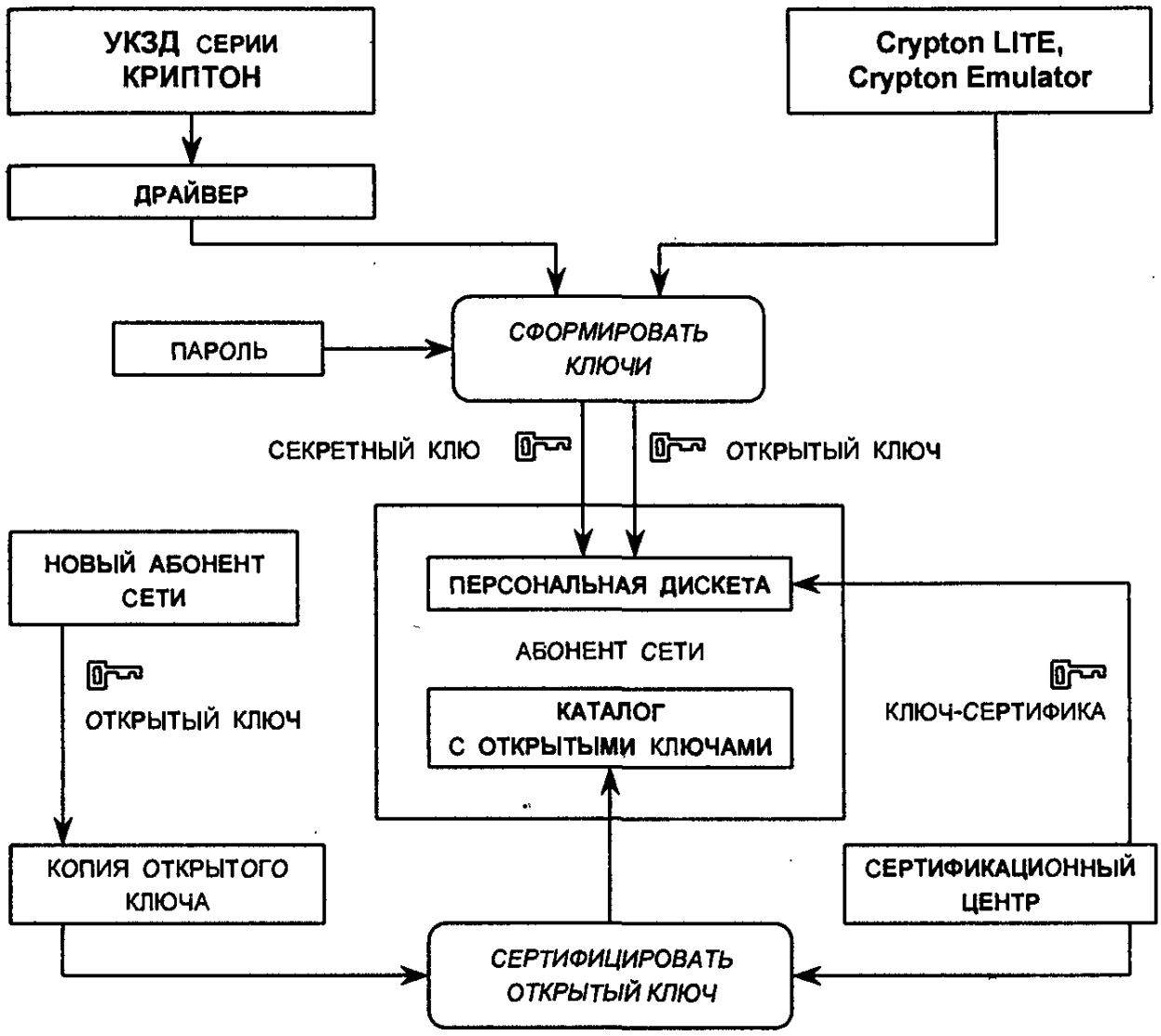


Рис.11.4. Схема работы с ключами

В общем случае необходимо выполнить следующую последовательность шагов.

1. Создание собственных ключей на персональной дискете. Генерация случайного кода для создания секретного ключа выполняется аппаратно устройством криптографической защиты данных серии КРИПТОН или программно драйвером-эмулятором УКЗД (Crypton Emulator). Секретный ключ необходимо закрыть паролем, который не позволит злоумышленнику воспользоваться им при похищении или копировании его.

2. Создание отдельного раздела (каталога) для размещения открытых ключей (например, PK DIR).

3. Создание резервных копий открытых ключей, полученных путем прямого обмена с другими пользователями. Эти ключи могут понадобиться при решении спорных вопросов, поэтому необходимо обеспечить их сохранность. С этой целью осуществляют запись открытых ключей в раздел PK DIR, удаляя у них подпись и формируя новую подпись собственным секретным ключом (для обеспечения целостности открытых ключей в процессе работы).

3'. Создание резервной копии открытых ключей, сертифицированных на ключе-сертификате. Эти открытые ключи записываются в соответствующий раздел PK DIR. Ключи-сертификаты записываются на персональную дискету. Данный вариант предпочтительнее предыдущего. В результате на персональной дискете будут располагаться:

- собственный секретный ключ (обязательно);
- собственный открытый ключ (обязательно, если он используется в качестве открытого ключа для проверки факта сертификации);
- ключи-сертификаты (их число определяется числом сертификационных центров, в которых пользователь сертифицирован).

Такая организация работы с ключами обеспечивает их относительную безопасность.

Пакет программ "КРИПТОН ® Подпись" состоит из следующих программных модулей:

- *расширение Windows Explorer "КРИПТОН ® Подпись"* выполняет основные действия пакета "КРИПТОН ® Подпись";
- *программа "КРИПТОН ® Подпись-Конфигурация"* служит для указания параметров работы программ, входящих в пакет программ "КРИПТОН ® Подпись";
- *программа "Мастер ключей подписи"* служит для работы с ключами и базами данных открытых ключей;
- *программа "Менеджер журналов операций"*служит для просмотра и редактирования файлов журналов операций.

Кроме того, пакет "КРИПТОН ® Подпись" предоставляет возможность автоматической обработки файлов-документов с помощью входящей в состав пакета утилиты командной строки Sgn Cmd.

Схема создания и проверки ЭЦП с помощью пакета "КРИПТОН® Подпись" аналогична схеме, используемой в Crypton Sign.

Как отмечалось, основные действия пакета "КРИПТОН ®Подпись" выполняются программным модулем "Расширение Windows Explorer "КРИПТОН ® Подпись". Этот модуль встраивается в контекстное меню Windows Explorer в виде дополнительного пункта меню с названием "КРИПТОН ® Подпись". При активизации данного пункта меню появляется подменю, содержащее основные команды модуля:

- подписать;
- проверить;
- удалить подпись;
- информация.

Все команды меню "КРИПТОН ® Подпись" выполняются над всеми выбранными файлами, а если выбран один или несколько каталогов – то над всеми файлами всех выбранных каталогов и их подкаталогов.

Непосредственно перед выполнением для выбранных файлов команды "Подписать" производится загрузка требуемого секретного ключа. Если загружаемый секретный ключ закрыт на пароле, происходит запрос пароля. При трехкратном вводе неверного пароля операция будет отменена. Аналогичным образом производится загрузка секретного ключа при выполнении других команд меню "КРИПТОН ® Подпись".

Соответствующие команды меню "КРИПТОН ® Подпись" позволяют проверить или удалить ЭЦП. Пункт меню "Информация" позволяет просматривать информацию о выбранных файлах.

Пакет программ защиты электронных документов Crypton ArcMail. Пакет программ Crypton ArcMail для Windows 95/98/NT 4.0 предназначен для защиты электронных документов от несанкционированного доступа и контроля их целостности при хранении в организации или передаче по открытым каналам связи. Пакет обеспечивает сжатие документов, проверку целостности документов, конфиденциальность документов, установление автора документа.

Передаваемые в электронном виде документы имеют различную степень конфиденциальности и могут содержать сведения от полностью открытых до составляющих коммерческую тайну самого предприятия или его партнеров. Кроме того, при введении электронного документооборота возникает вопрос обеспечения достоверности передаваемых документов.

Наиболее остро вопрос защиты документооборота стоит для предприятий, имеющих территориально-распределенную структуру. Такие предприятия могут иметь несколько локальных вычислительных сетей (ЛВС), расположенных в разных местах, в том числе в различных регионах России, и вынуждены использовать для передачи информации различные глобальные вычислительные сети (ГВС) общего пользования, например сеть Internet.

При электронном документообороте возникают различные угрозы со стороны пользователей ГВС, которые можно разделить на две основные категории:

- угрозы конфиденциальности информации;
- угрозы целостности информации.

Наиболее надежным средством для обеспечения конфиденциальности информации является шифрование. Авторство и целостность электронного документа позволяет установить электронная цифровая подпись. Схема использования ЭЦП в пакете программ Crypton ArcMail такая же, как и в предыдущем пакете (см. рис.11.3).

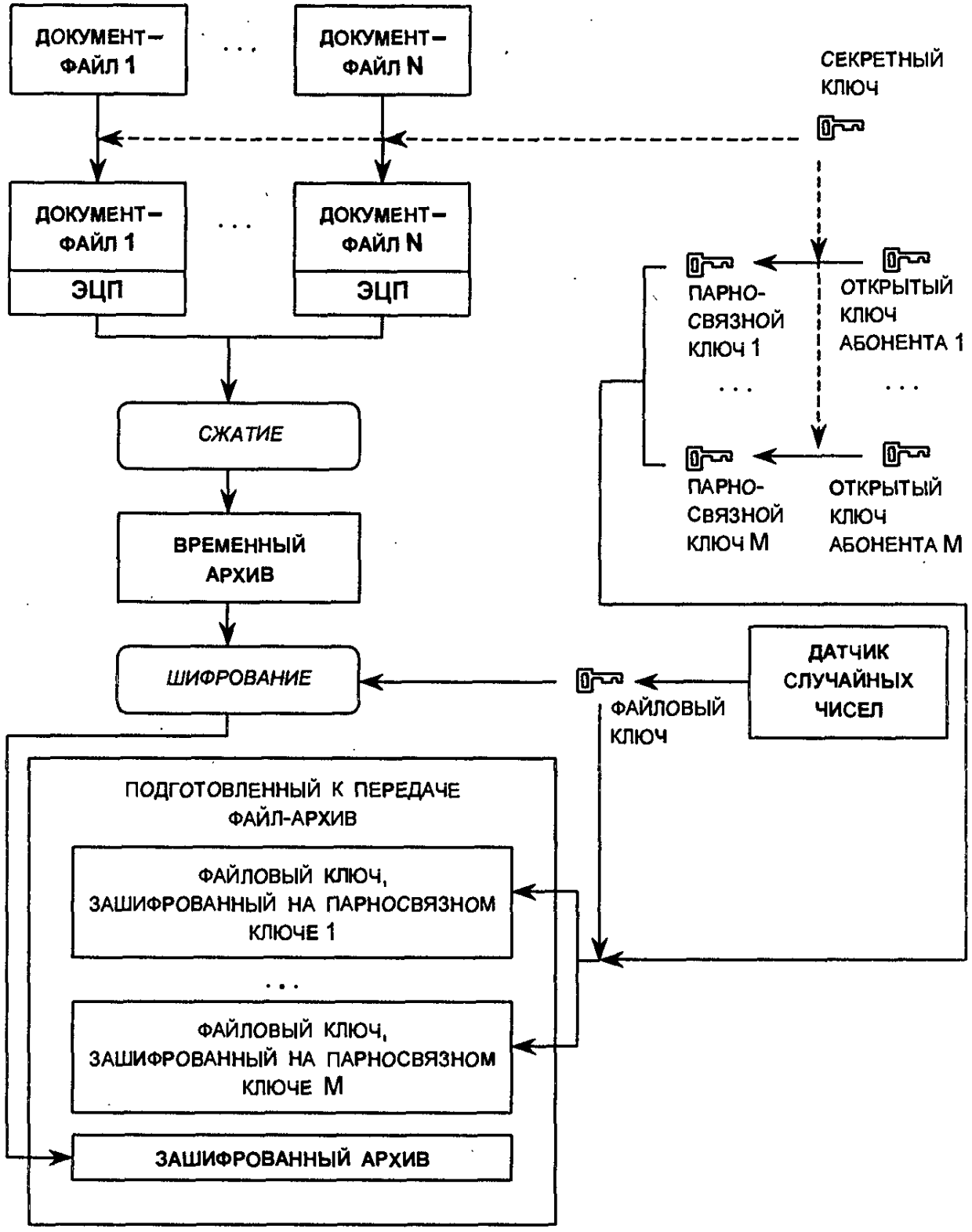


Рис.11.5. Алгоритм создания архива

Секретный ключ (СК) генерируется абонентом сети. ЭЦП формируется на основе СК и вычисленного с помощью хэш-функции значения хэша документа. Ключ СК может быть зашифрован на пароле абонента. Открытый ключ (ОК) вычисляется как значение некоторой функции от СК и используется для проверки ЭЦП. Ключ ОК должен быть передан всем абонентам сети, с которыми планируется обмен защищенной информацией.

При проверке ЭЦП принятого по сети подписанного документа вычисляется значение хэша этого документа. Любые изменения документа приведут к другому значению хэша, что явится сигналом нарушения целостности принятого документа.

Для защиты и конфиденциальности, и целостности информации необходимо использовать в комплексе шифрование и ЭЦП, что можно совместить с таким дополнительным сервисом, как сжатие (архивация) информации. Эти возможности обеспечивает специализированный архиватор электронных документов Crypton ArcMail.

Схема алгоритма создания архива для передачи по сети приведена на рис.11.5.

При создании архива исходные документы-файлы подписываются на секретном ключе абонента сети, после чего подписанные файлы сжимаются. Получаемый в результате сжатия временный архив шифруется на случайном (файловом) ключе.

Абоненты, которым предназначается архив, могут расшифровать его с помощью записанного в архив зашифрованного файлового ключа. Файловый ключ зашифровывается на парносвязном ключе, вычисляемом по алгоритму Диффи–Хеллмана из секретного ключа СК отправителя и открытого ключа абонента-адресата. Парносвязный ключ может также выбираться из симметричных ключей сетевого набора (КСН), структурно оформленных в виде базы данных открытых ключей. В этом случае КСН зашифрованы на главном ключе.

Таким образом, достигаются следующие цели:

- передаваемые электронные документы снабжаются кодом подтверждения достоверности – ЭЦП, который защищает их от нарушения целостности или подмены;
- документы передаются в защищенном виде, что обеспечивает их конфиденциальность;
- абоненты-адресаты могут расшифровывать документы, используя свой секретный ключ и открытый ключ отправителя (или КСН);
- абоненты сети, которым не предназначается данный архив, не могут прочитать его содержимое, поскольку не имеют файлового ключа и не могут его вычислить;
- дополнительный сервис – уменьшение объема передаваемой информации, обусловленное архивацией.

Согласно описанной выше схеме, ключи должны распределяться следующим образом: секретный ключ должен находиться у его владельца, парный ему открытый ключ должен быть передан владельцем всем абонентам сети, с которыми он хочет обмениваться защищенной информацией. Открытые ключи не являются

секретными, но существует возможность их подмены. Например, возможна ситуация, когда у злоумышленника есть доступ к компьютеру, на котором абонент А хранит открытые ключи. Злоумышленник считывает интересующие его сведения (Ф.И.О., должность, ...) из открытого ключа, например, абонента В, после чего генерирует где-либо СК и ОК с такими данными и заменяет на компьютере абонента А открытый ключ абонента В на фиктивный. После этого злоумышленник может подписать любой документ своим СК с данными абонента В и переслать его абоненту А. При проверке ЭЦП такого документа будет выдано сообщение типа "Подпись лица (Ф.И.О., должность, ...) верна", что введет в заблуждение абонента А.

Таким образом, очевидно, что необходима защита и открытых ключей. Такую защиту можно обеспечить несколькими способами:

- использование персональной дискеты;
- использование ключей-сертификатов.

При первом способе собственный секретный ключ и открытые ключи других абонентов могут быть записаны на персональную дискету, доступ к которой должен быть только у ее владельца. Однако при большом количестве абонентов сети и большом потоке документов такой вариант нецелесообразен, так как замедляется обработка документов. Более предпочтительным является второй способ хранения и защиты ключей. Схема использования ключей-сертификатов показана на рис.11.6. Данный способ предполагает наличие сертификационного центра (СЦ), в котором на специальном ключе (ключе-сертификате) подписывается открытый ключ абонента сети перед передачей его другим абонентам. Открытый ключ-сертификат должен храниться у всех абонентов сети для проверки целостности всех используемых в сети открытых ключей. При таком варианте рекомендуется при проверке ЭЦП какого-либо документа автоматически проверять подпись соответствующего ОК. В этом случае ОК могут храниться в открытом виде, а персональная дискета, помимо секретного ключа владельца, должна содержать еще и открытый ключ-сертификат.

Сертификационный центр можно объединить с центром распределения ключей (ЦРК). В этом случае выделяется специальное рабочее место, используемое как для генерации ключей абонентов, так и для их сертификации и рассылки абонентам. Даже в случае генерации ключей непосредственно абонентами на местах, СЦ можно использовать для рассылки абонентам заверенных открытых ключей, как показано на рис.11.6. Данная схема особенно целесообразна при организации электронного документооборота между несколькими юридическими лицами.

Распределение ключей согласно схеме обмена ключами (см. рис.11.6) происходит в следующем порядке.

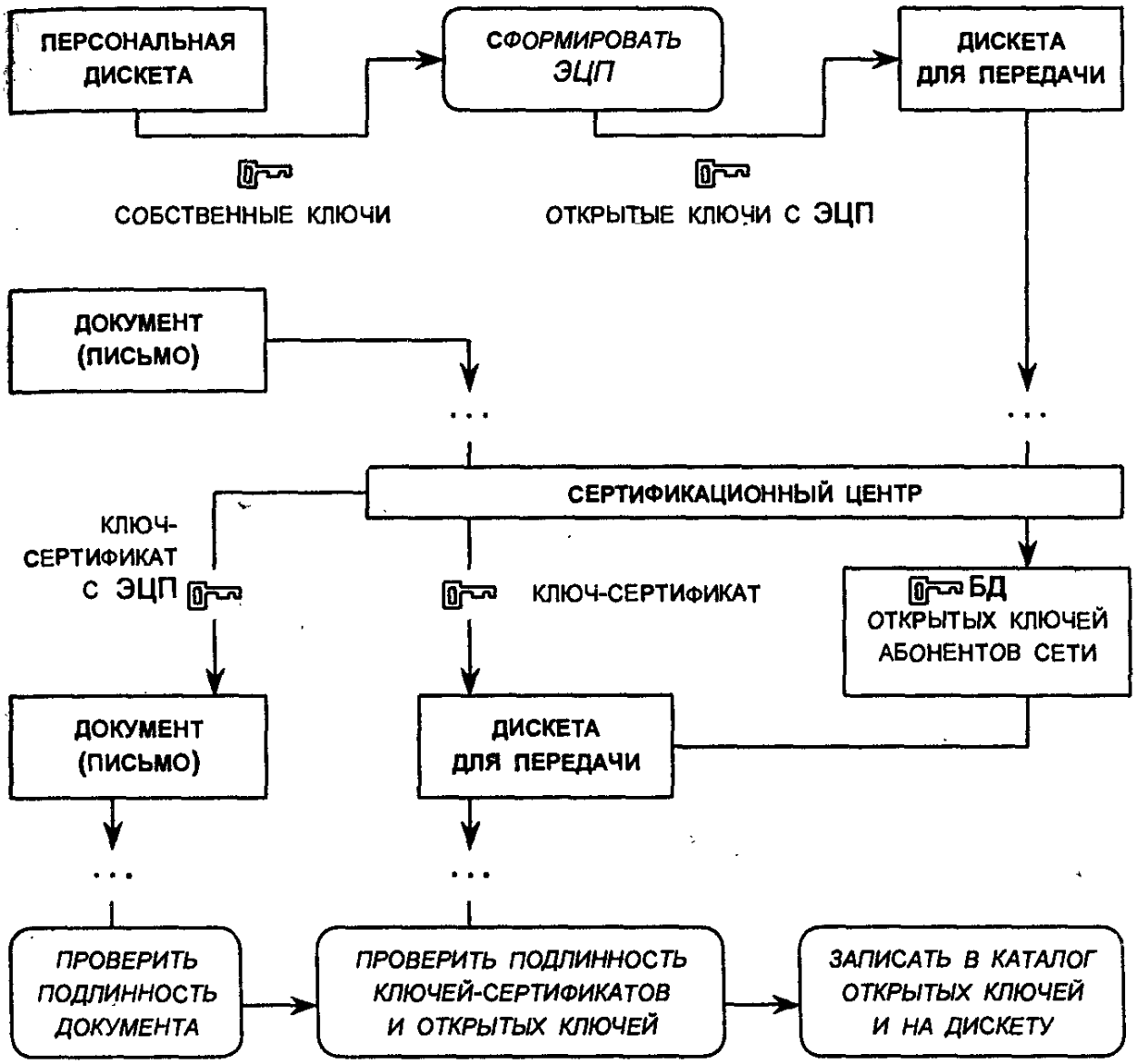


Рис.11.6. Схема обмена ключами через сертификационный центр

- Абонент создает персональную дискету с собственными ключами. Секретный ключ закрывается паролем.
- Для собственного открытого ключа формируется подпись на собственном секретном ключе, открытый ключ записывается на дискету для передачи.
- Создается юридический документ на бумаге (например, письмо), в котором указываются: данные о владельце (Ф.И.О., должность, место работы), открытый ключ (распечатка в шестнадцатеричном коде), полномочия владельца (перечень документов, которые уполномочен удостоверять владелец открытого ключа). Данный документ должен быть оформлен таким образом, чтобы иметь юридическую силу в случае возникновения спорных вопросов о принадлежности подписи и полномочиях владельца. Если в письме не установлены полномочия, то они определяются по должности и месту работы.

- Данный документ вместе с ОК пересылается в СЦ.
- СЦ проверяет юридическую силу полученного документа, а также идентичность ОК на дискете и в документе.
- В ответ абонент получает:
  сертифицированные ОК всех абонентов (в том числе, и свой);
  сертифицированные файлы с полномочиями владельцев ОК;
  ключ-сертификат центра как в виде файла, так и виде юридического документа.
- Владелец проверяет истинность ключа-сертификата, а также подписи всех полученных им открытых ключей и файлов. При успешной проверке открытые ключи записываются в соответствующий каталог, а ключ-сертификат – на персональную дискету.

При такой организации абонент формирует ЭЦП документов и может не заботиться об обмене открытыми ключами и полномочиями. Однако большая нагрузка по рассылке открытых ключей и полномочий ложится на СЦ. В том случае, если у абонента сети остаются какие-либо сомнения относительно конкретного ОК, он может запросить распечатку ОК и полномочия напрямую у его владельца.

Можно оставить за СЦ только сертификацию ключей и полномочий, освободив его от рассылки ОК. В этом случае, при первой посылке в любой адрес документов, абоненту необходимо послать по этому адресу также сертифицированные ОК и полномочия.

В общем случае сертификационных центров может быть несколько. Пользователь может сертифицировать свой ОК в разных, не связанных друг с другом СЦ. Кроме того, СЦ могут быть связаны в сеть с любой необходимой иерархической организацией для обмена либо только ключами-сертификатами, либо дополнительно еще и открытыми ключами. Тогда пользователю достаточно сертифицировать ОК только в одном из таких СЦ для обмена информацией с абонентами всех охватываемых этим центром сетей.

При большом количестве абонентов сети рекомендуется использовать базы данных (БД) открытых ключей. В этом случае СЦ пересылает абоненту не отдельные ОК, а одинаковый для всех абонентов файл БД, содержащий все используемые ОК.

Согласно изложенному выше, персональная дискета должна содержать следующее:

- СК владельца;
- открытые ключи-сертификаты по числу сертификационных центров.

В качестве ключа-сертификата может быть использован собственный секретный ключ абонента; в этом случае при получении ОК другого абонента, этот ОК необходимо подписать. При этом на персональную дискету следует записать свой ОК для проверки целостности ОК других абонентов.

Вместо персональной дискеты может быть использован другой ключевой носитель, например устройства Touch Memory или смарт-карта, что иногда предпочтительнее, поскольку ключи шифрования могут быть непосредственно загружены со смарт-карты в шифроустройство, минуя оперативную память компьютера.

Следует отметить, что если используются персональные дискеты с централизованной генерацией ключей абонентов, для защиты ЦРК необходимо применять системы защиты от несанкционированного доступа (ЗНСД).

Следует также учесть, что любые системы защиты документооборота будут недостаточны без определенных организационных мер:

- Протоколирование всех операций, совершенных с помощью системы защиты. Протоколирование (ведение журналов операций) должно быть обязательным на особо важных рабочих местах, например ЦРК, СЦ, рабочем месте администратора безопасности.
- Предотвращение получения злоумышленниками ключевых дискет и их тиражирования владельцами. Для этого помимо простого введения паролей можно использовать в качестве ключевых носителей микропроцессорные смарт-карты.
- Разграничение доступа на рабочие места как административными мерами (например, ограничением доступа в помещения), так и с помощью различных ЗСНД, что особенно актуально для тех же ЦРК и СЦ.

Пакет программ Crypton ArcMail состоит из четырех программных модулей:

- спецархиватора Crypton ArcMail;
- программы "Конфигурация Crypton ArcMail";
- программы "Менеджер баз данных открытых ключей";
- программы "Менеджер журнала операций".

Программный модуль "Спецархиватор Crypton ArcMail" выполняет основные действия пакета Crypton ArcMail. Главное окно спецархиватора по внешнему виду очень похоже на главное окно Проводника Windows (программы Explorer). Кроме стандартного набора команд Проводника Windows, в меню и панель инструментов главного окна спецархиватора Crypton ArcMail добавлены функции, реализующие возможности спецархиватора. Правила работы с файлами и папками в главном окне спецархиватора Crypton ArcMail практически полностью аналогичны правилам работы в окне Проводника Windows.

Программа "Конфигурация Crypton ArcMail" служит для установки параметров работы программ, входящих в пакет Crypton ArcMail.

Программа "Менеджер баз данных открытых ключей" предназначена для работы с ключами владельцев и базами данных открытых ключей. Главное окно менеджера баз данных открытых ключей, как и главное окно спецархиватора Crypton ArcMail, имеет меню, панель инструментов и строку статуса. Кроме того, данное окно может содержать несколько окон редактируемых баз данных открытых ключей.

Программа "Менеджер журнала операций" выполняет просмотр и редактирование файлов журнала операций. Главное окно менеджера журнала операций может содержать несколько окон просмотра и редактирования журналов операций.

11.2. Пакетное шифрование

Шифрование пакетов осуществляется коммуникационными программами на сетевом уровне (IP-протокол) семиуровневой модели OSI/ISO непосредственно перед передачей пакетов сетевому интерфейсу (канальному уровню). Коммуникационные программы могут располагаться как на абонентском месте клиента, так и на сервере, в центре коммутации пакетов и т.д. Ключевым моментом является защита целостности коммуникационных программ от возможного их обхода. Эта защита определяется надежностью применяемой системы защиты от НСД. Вторым не менее важным моментом является недосягаемость ключей шифрования и ЭЦП. Это можно обеспечить только с помощью аппаратных средств (например, платы серии КРИПТОН).

Шифрование пакетов может быть реализовано в виде отдельного устройства – так называемого шифратора IP-пакетов (криптомаршрутизатора). В простейшем виде последний представляет собой ПК с двумя сетевыми платами. В этом случае надежность криптомаршрутизатора определяется системой защиты от НСД со стороны консоли. В настоящее время надежный криптомаршрутизатор может быть только под управлением DOS или "операционной системы" собственной разработки.

Шифрование пакетов может осуществляться и с помощью коммуникационных программ совместной разработки ОАО "Элвис+" и ООО АНКАД:

- Crypton Fort E+ Personal Client – средство защиты персональной рабочей станции;
- Crypton Fort E+ Corporate Client – средство защиты рабочей станции корпоративной сети;
- Crypton Fort E+ Server – средство защиты сервера;
- Crypton Fort E+ Branch – защита сегмента локальной сети от несанкционированного доступа из внешней сети.

Два компьютера, имеющих такие коммуникационные модули, могут осуществлять защищенную связь посредством шифрования пакетов по схеме использования открытых ключей в протоколе SKIP (см. § 7.3 и 8.4). Непосредственно шифрование и генерация случайных последовательностей осуществляются описанными выше УКЗД и их программными эмуляторами.

Протокол SKIP реализуется следующим образом. Каждый узел сети снабжается секретным ключом K_c и открытым ключом K_o. Открытые ключи могут свободно распространяться среди пользователей, заинтересованных в организации защищенного обмена информацией. Узел I, адресующий свой трафик к узлу J, на основе логики открытых ключей Диффи–Хеллмана вычисляет разделяемый секрет K_{ij}:

$$K_{ij} = (K_{oj})^{Kci} \bmod n = (g^{Kcj})^{Kci} \bmod n = g^{Kci * Kcj} \bmod n,$$

где g и n – некоторые заранее выбранные многоразрядные простые целые числа. Ключ K_{ij} является долговременным разделяемым секретом для любой пары абонентов I и J и не может быть вычислен третьей стороной. В то же время легко видеть, что и отправитель, и получатель пакета могут вычислить разделяемый секрет на основании собственного секретного ключа и открытого ключа партнера:

$$K_{ij} = (K_{oj})^{Kci} \bmod n = (K_{oi})^{Kcj} \bmod n = K_{ji}.$$

Ключ K_{ij} ответственен за обмены с узлами I и J, однако он не используется в протоколе SKIP непосредственно для защиты трафика. Вместо этого для каждого конкретного IP-пакета (или некоторой группы пакетов) узел I вырабатывает специальный пакетный ключ $K_п$, при помощи которого защищает исходный IP-пакет и укладывает его в блок данных SKIP-пакета. Далее собственно пакетный ключ $K_п$ защищается при помощи разделяемого секрета K_{ij}, причем, возможно, при помощи другого, более сложного алгоритма защиты данных, и тоже записывается в SKIP-пакет.

Использование пакетного ключа является дополнительной мерой защиты по двум причинам:

- во-первых, долговременный секрет не должен быть скомпрометирован и не следует давать вероятному противнику материал для статистического криптоанализа в виде большого количества информации, защищенной соответствующим ключом;
- во-вторых, частая смена пакетных ключей повышает защищенность обмена, так как если пакетный ключ и будет скомпрометирован, то ущерб будет нанесен лишь небольшой группе пакетов, защищенных при помощи данного пакетного ключа.

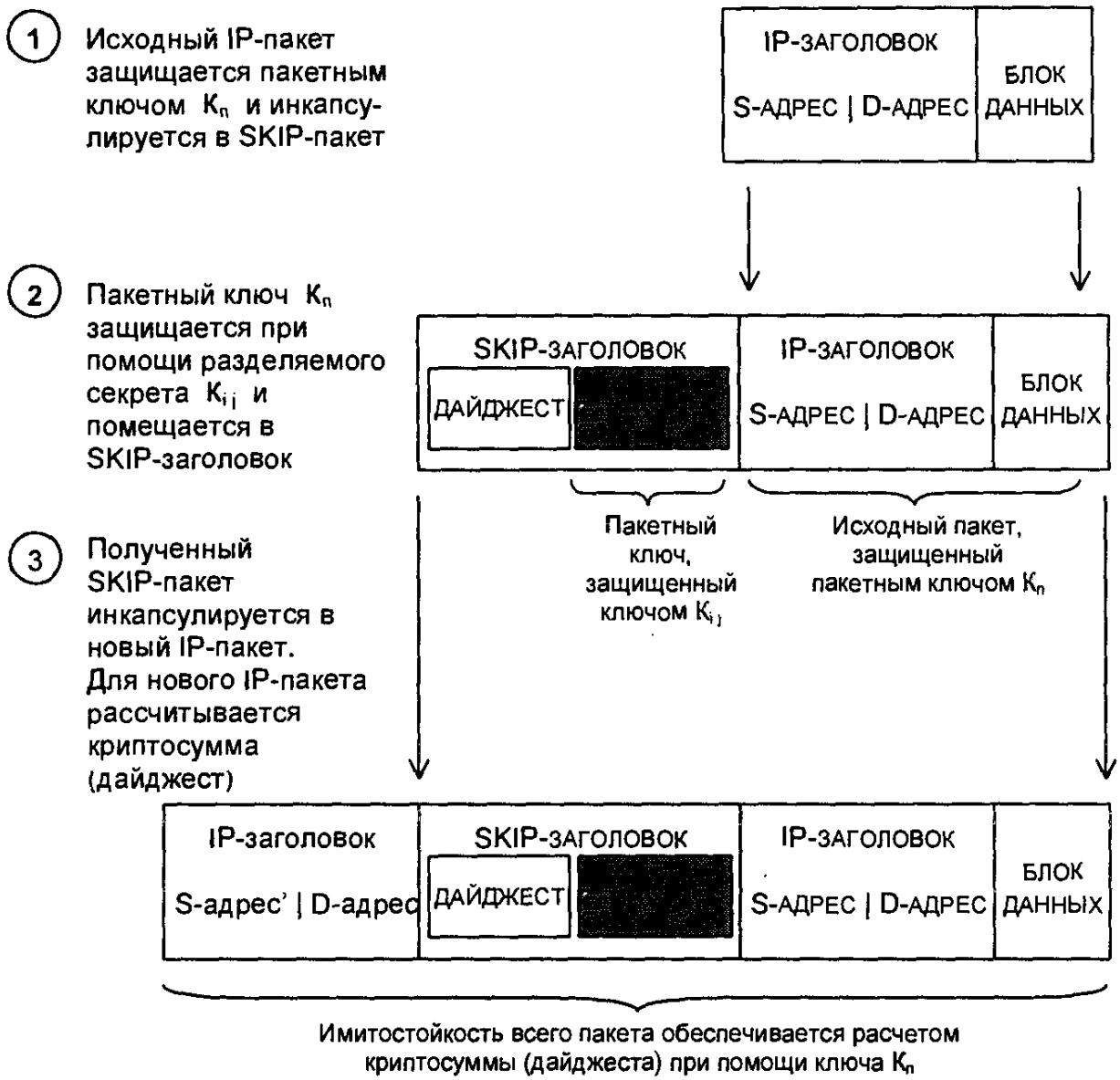


Рис.11.7. Последовательность операций формирования пакета в протоколе SKIP

Последовательность операций по обработке исходного IP-пакета показана на рис.11.7. Исходный пакет защищается пакетным ключом и помещается внутрь нового пакета, снабженного SKIP-заголовком. Пакетный ключ включается в заголовок полученного SKIP-пакета. Затем этот SKIP-пакет включается (инкапсулируется) в новый IP-пакет. Отметим, что заголовок нового IP-пакета может не совпадать с заголовком исходного IP-пакета. В частности, может быть изменена адресная информация, содержащаяся в исходном пакете. Такая подмена адресов является самостоятельным аспектом защиты информационных систем и называется адресной векторизацией.

По всему новому пакету (за исключением динамически меняющихся полей заголовка IP-пакета) с использованием пакетного ключа рассчитывается контрольная криптосумма. Поскольку пакетный ключ защищен при помощи разделяемого секрета, восста-

новить его и корректно рассчитать криптосумму могут только два участника защищенного обмена. Тем самым обеспечивается аутентификация информации как на уровне узла сети (достоверно известен IP-адрес отправителя), так и, возможно, на уровне пользователя, идентификатор которого (однозначно соответствующий секретному ключу) включается в SKIP-заголовок. Сформированный в результате перечисленных операций новый IP-пакет отправляется получателю, который в обратном порядке производит его обработку: вычисляет разделяемый секрет, восстанавливает пакетный ключ, проверяет контрольную криптосумму, восстанавливает и извлекает из SKIP-пакета исходный IP-пакет.

Поскольку SKIP-защищенный пакет является стандартным IP-пакетом, все промежуточное сетевое оборудование между отправителем и получателем стандартным образом маршрутизует этот пакет до его доставки узлу-получателю.

Аутентификация

Реализуется в коммуникационном модуле. Один из вариантов аутентификации, использующий симметричную криптографию, выглядит следующим образом.

- Шифрование и генерация случайных чисел могут осуществляться с помощью платы серии КРИПТОН.
- Оба абонента владеют секретным ключом Sk.
- Абонент А генерирует случайное число Da и посылает его абоненту В.
- Абонент В шифрует принятое число Da на ключе Sk (результат обозначим через Da(Sk)), генерирует свое случайное число Db и посылает абоненту А числа Db и Da(Sk).
- Абонент А шифрует Da и сравнивает результат шифрования Da(Sk) с полученным числом. Если они совпадают, значит, это абонент В. Далее абонент А шифрует число Db и посылает абоненту В число Db(Sk) совместно с информацией.
- Абонент В шифрует Db и сравнивает Db(Sk) с полученным числом. Если числа совпадают, значит, абонент В связан с абонентом А и может передавать ему информацию.

11.3. Защита компонентов ЛВС от НСД

Защита абонентских пунктов

Рассмотрим несколько вариантов реализации абонентских пунктов (АП). Предположим, что уровень защищенности абонентского пункта должен обеспечивать нахождение в нем секретной информации. Помещения, в которых будут находиться такие або-

нентские пункты, должны иметь соответствующую категорию. Передача информации между абонентскими пунктами и абонентским пунктом и сервером в защищенном режиме производится в зашифрованном виде.

Абонентский пункт для DOS, Windows 3.11. В состав пункта входят (рис.11.8):

- система защиты от несанкционированного доступа КРИПТОН-ВЕТО;
- коммуникационный модуль с аутентификацией, абонентским шифрованием или с шифрованием пакетов.

Коммуникационный модуль обеспечивает защиту ПК со стороны сети. Он пропускает на ПК только пакеты, зашифрованные определенными ключами. Запись ключевой информации на ПК осуществляется администратором сегмента сети, в котором находится абонентский пункт. В функции коммуникационных программ Cryptoп Fort E+ Client входят:

- реализация заданной дисциплины работы и разрешение доступа (на абонентский пункт могут прислать пакеты только абонентские пункты, включенные в список разрешенных соединений);
- аутентификация трафика (производится строгая аутентификация абонентского пункта, который прислал пакет по сети);

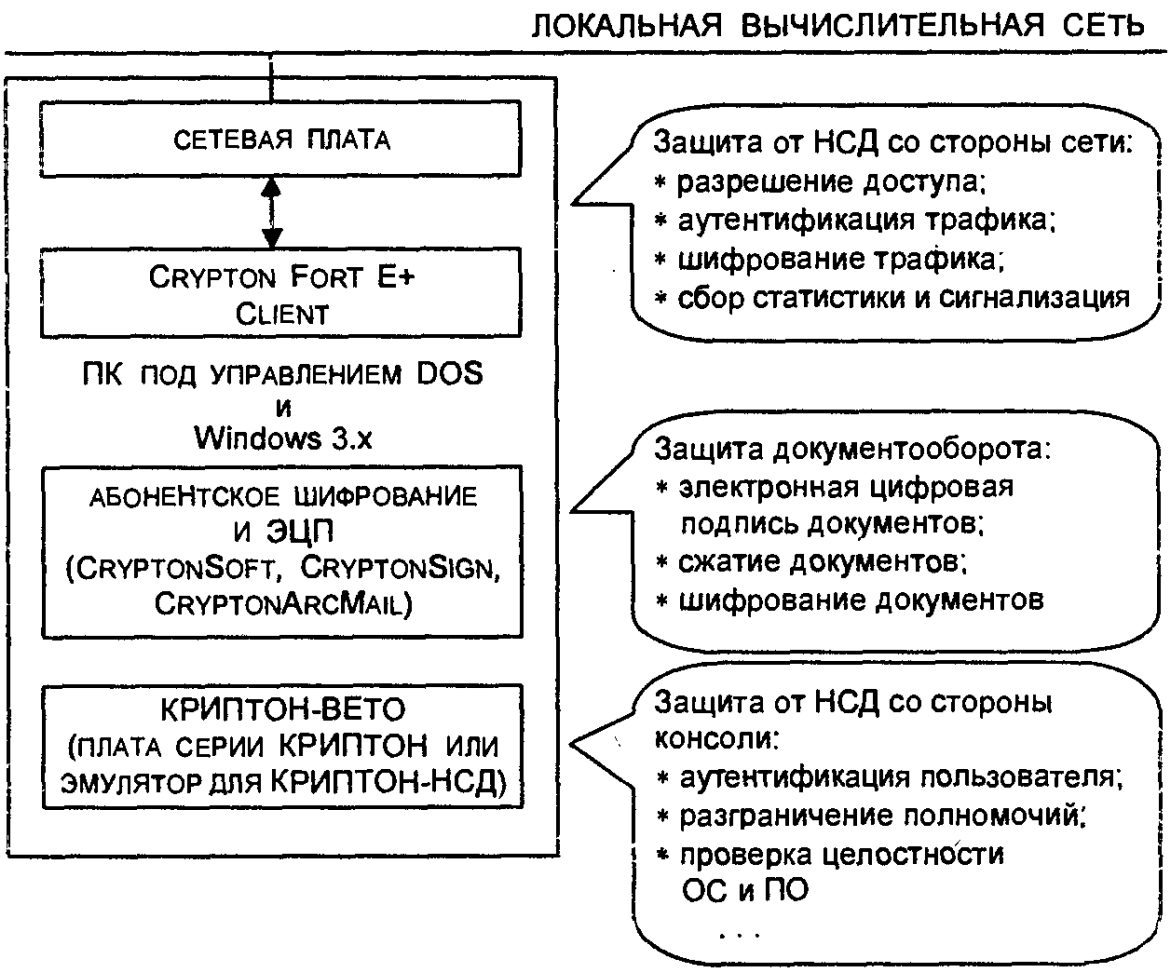


Рис.11.8. Структура средств защиты АП под управлением DOS и Windows 3.x

- шифрование IP-пакетов;
- сбор статистики и сигнализация;
- настройка параметров защиты с помощью удобного графического интерфейса.

Абонентский пункт, работающий под управлением DOS, можно отнести к полностью контролируемой системе. Для абонентского пункта с Windows 3.x при необходимости возможно усиление защиты с помощью криптомаршрутизатора под управлением DOS или специализированной сетевой платы.

Абонентский пункт для Windows NT и UNIX. Поскольку в этом случае нельзя полностью доверять операционной системе, то такой абонентский пункт относится к частично контролируемым системам. Поэтому для подобных абонентских пунктов внутри сегмента сети необходимо использовать полностью контролируемое устройство для анализа функционирования программных средств защиты. В целом защита строится аналогично описанной выше для абонентского пункта под управлением DOS (рис.11.9).

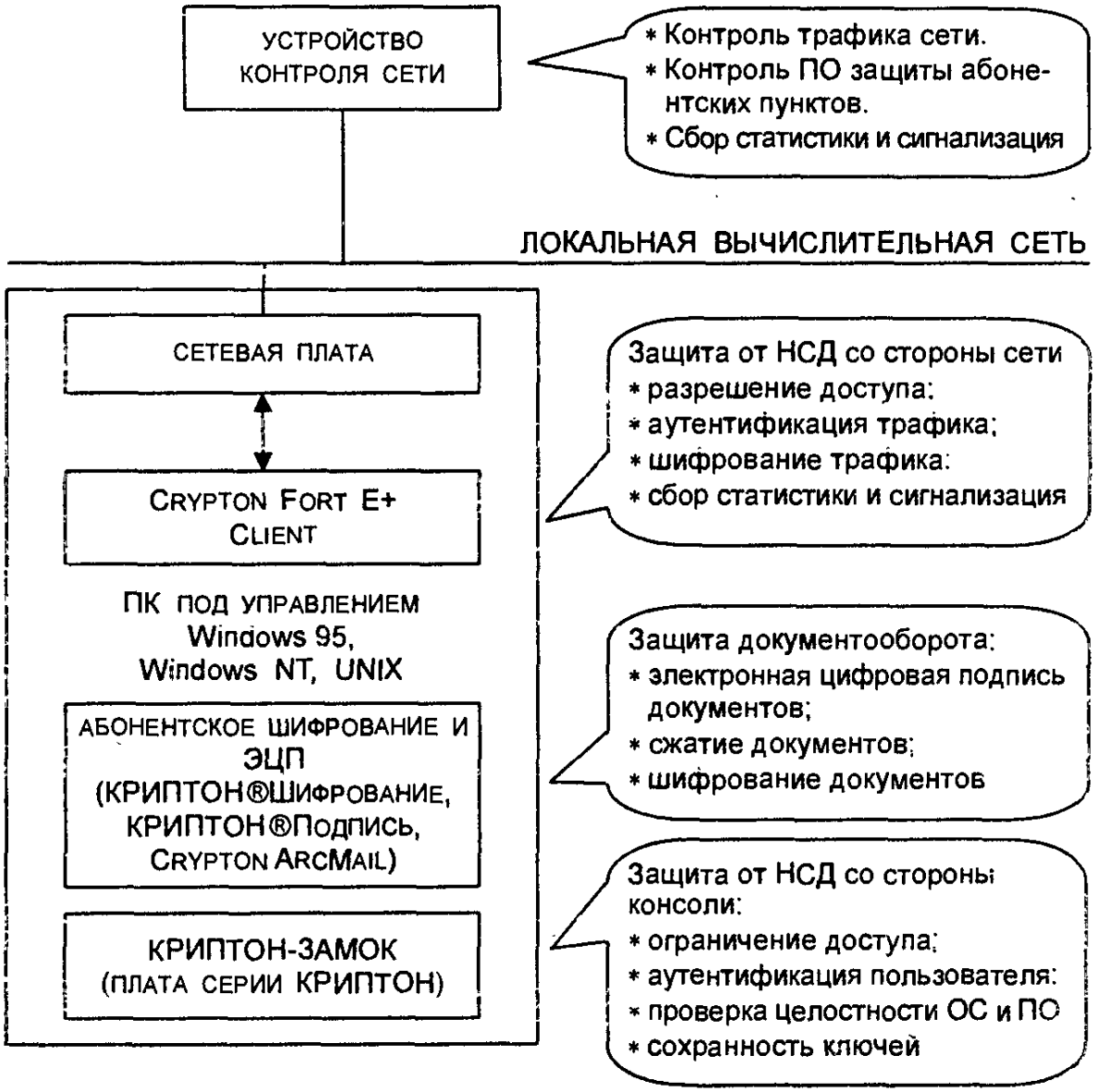


Рис.11.9. Структура средств защиты АП под управлением Windows 95/98/NT и UNIX

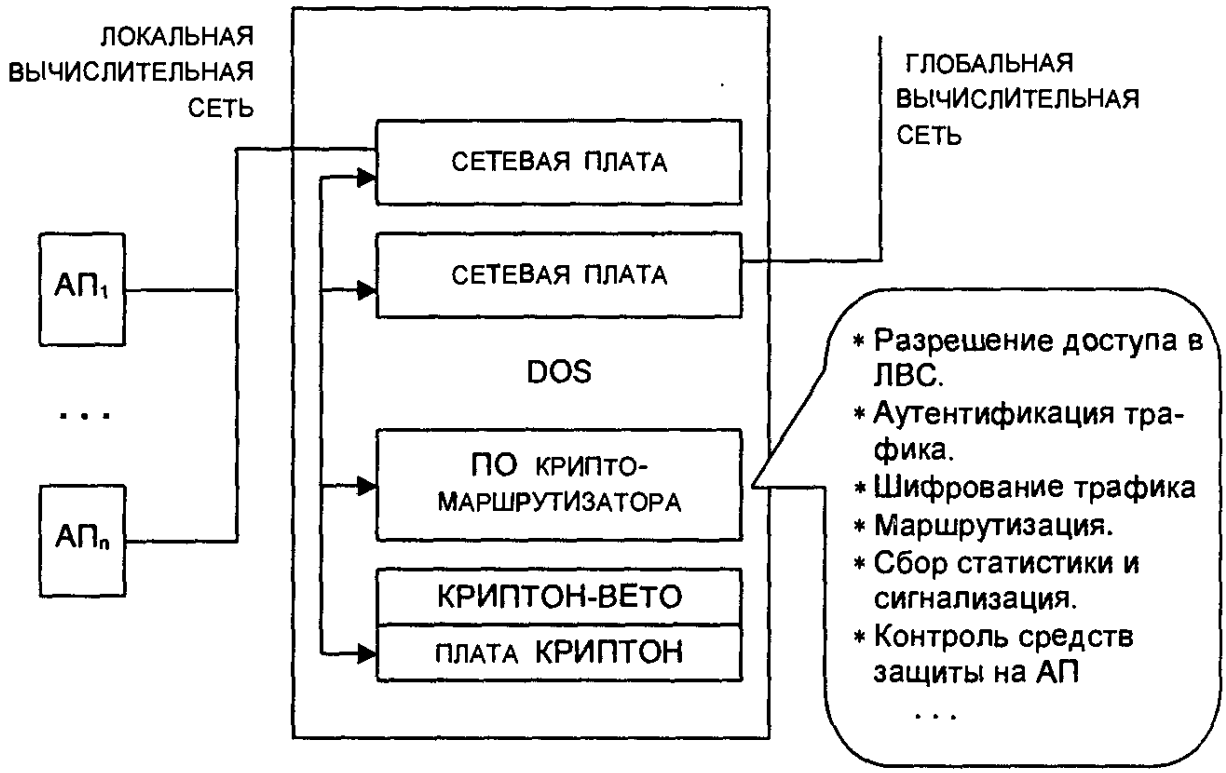


Рис.11.10. Структура криптомаршрутизатора

Пункт состоит из:

- системы защиты от несанкционированного доступа КРИПТОН-ЗАМОК (обеспечивает ограничение доступа к ПК и проверку целостности программного обеспечения перед загрузкой операционной системы);
- коммуникационного модуля с аутентификацией, абонентским шифрованием или с шифрованием пакетов.

Для выхода во внешний сегмент или при хранении и обработке важной информации возможна реализация шифрующего модуля в виде криптомаршрутизатора (рис.11.10), который реализуется как полностью контролируемая система. Такой криптомаршрутизатор может выполнять также функции контроля за программным обеспечением абонентских пунктов с частично контролируемым ПО.

Защита терминалов. Терминальные устройства должны подключаться к хост-компьютерам, которые защищаются так же, как абонентский пункт, описанный выше.

Поскольку канал связи между хост-компьютером и терминалом не шифруется, последние должны находиться в одной комнате.

Защита маршрутизаторов. Криптомаршрутизатор

Подобная защита реализуется только под управлением сертифицированной DOS. Схема защиты похожа на расширенный абонентский пункт, дополненный сетевыми платами и программным обеспечением для выполнения функций маршрутизации пакетов. Такой защищенный компьютер рассматривается как барьер между открытой и закрытой средой.

Структурная схема криптографического IP-маршрутизатора КРИПТОН-IP показана на рис.11.10.

Криптомаршрутизатор КРИПТОН-IP предназначен для применения в качестве маршрутизатора пакетов данных (с IP-форматом) между глобальной и локальными компьютерными сетями с обеспечением защиты от несанкционированного доступа к данным пакета. Как в комплексе КРИПТОН-IP, так и в пакетах данных при обмене или в открытой сети осуществляется криптографическая защита данных (их шифрование происходит согласно ГОСТ 28147-89). Для контроля целостности и истинности файлов данных введено формирование (при необходимости) их электронной цифровой подписи согласно ГОСТ Р 34.10-94.

Для защиты от НСД подключенных к комплексу локальных компьютерных сетей используются также методы фильтрации IP-пакетов по определенным правилам с аутентификацией их источников.

Кроме криптографической защиты данных, в комплексе КРИПТОН-IP реализовано разграничение доступа к размещенным в его памяти программным средствам и данным с регистрацией в электронном журнале процесса доступа к ресурсам комплекса.

В комплексе применено сертифицированное аппаратно-программное средство криптографической защиты информации Crypton ArcMail, имеющее сертификат ФАПСИ (регистрационный номер СФ/120-0278 от 30.06.99 г.), в состав которого входят следующие аппаратные и программные средства.

- *Устройство криптографической защиты данных* КРИПТОН-4К/16, имеющее свой локальный программный BIOS УКЗД, который выполняет:
  загрузку ключей шифрования данных до загрузки операционной среды компьютера;
  контроль целостности операционной среды компьютера до загрузки MS DOS под управлением программного обеспечения комплекса;
  шифрование данных под управлением программного обеспечения комплекса.
- *Адаптер смарт-карт SA-101i*, обеспечивающий ввод ключевой информации в УКЗД со смарт-карт с открытой памятью, минуя шину данных компьютера.
- *Программы системы криптографической защиты информации от несанкционированного доступа (СКЗИ НСД) КРИПТОН-ВЕТО 2.0*, которые управляют:
  процессом контроля целостности операционной среды компьютера;
  шифрованием данных;
  контролем разграничения доступа к ресурсам компьютера комплекса;
  регистрацией в электронном журнале процесса работы комплекса.

- *Программы комплекса Crypton Router v.2.0,* реализующие методы криптографической защиты и автоматическую маршрутизацию пакетов при приеме/передаче по сети обмена данными.

Любой абонент защищенной сети, подсоединенной к криптомаршрутизатору, может обмениваться данными с любым другим абонентом сети, причем шифрование передаваемых данных для абонентов является прозрачным. Кроме того, применение криптомаршрутизатора позволяет скрыть трафик между абонентами защищенных локальных сетей. Это определяется тем, что обмен данными происходит между криптомаршрутизаторами, имеющими собственные сетевые адреса, а адреса абонентов передаются по каналам связи только в зашифрованном виде.

Для систем с конфиденциальной информацией можно использовать маршрутизатор под управлением UNIX Crypton Fort E+ Branch, отдавая себе отчет, что это все-таки частично контролируемая система.

Для контроля абонентского пункта локальной сети, не выходящей в глобальную вычислительную сеть (ГВС), можно использовать облегченный вариант устройства контроля сети, представляющего собой урезанный криптомаршрутизатор – без второй сетевой платы и ПО маршрутизации.

Защита центра генерации ключей. Центр генерации ключей располагается на отдельном компьютере (по структуре аналогичен абонентскому пункту под управлением DOS), доступ к которому имеет только администратор сети. В состав центра включаются:

- система защиты от несанкционированного доступа КРИПТОН-ВЕТО;
- программа CryptonSoft;
- коммуникационный модуль с аутентификацией и шифрованием пакетов.

Защита локальных серверов, серверов приложений и корпоративного сервера. Защита баз данных и файл-серверов может производиться так же, как защита абонентского места (см. рис.11.9). Однако доступ к серверам с секретной информацией должен ограничиваться с помощью устройств защиты от НСД с разграничением доступа, представляющих собой расширенный функциями проверки полномочий и обеспечения доступа к разрешенным приложениям и документам криптомаршрутизатор (см. рис.11.10).

Защита сегментов сетей. Защита осуществляется на основе перечисленных выше компонентов. Внутри сегмента сети используются программные коммуникационные модули на абонентских пунктах $АП_1$–$АП_n$ (рис.11.11). В частично контролируемой среде есть вероятность их подмены. Поэтому для выхода в сеть из сег-

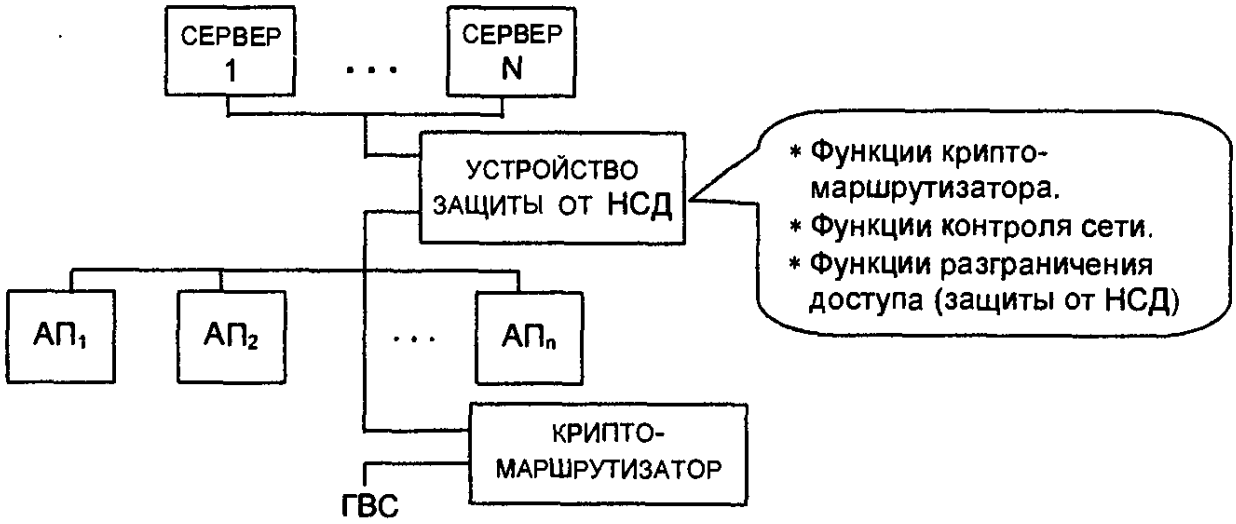


Рис.11.11. Схема защиты локальной вычислительной сети

мента и входа из сети в сегмент применяются средства с операционной системой DOS и шифрованием – маршрутизаторы (см. рис.11.10) или специализированные сетевые платы.

Предложенная выше структура не является единственной. Допускается введение мостов, маршрутизаторов, межсетевых экранов как в защищенной части сети, так и в открытой (рис.11.12).

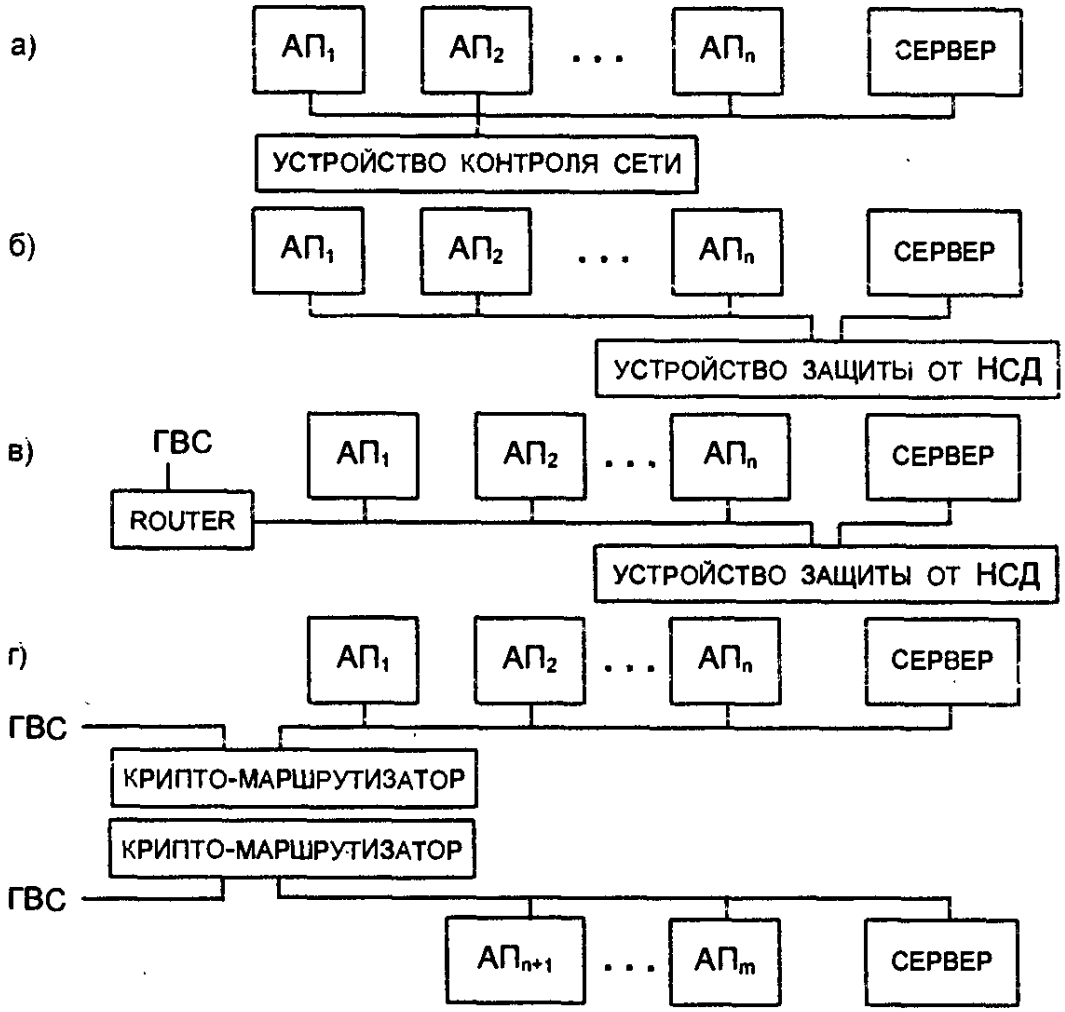


Рис.11.12. Варианты структуры защищенных ЛВС:

а) ЛВС с программной защитой сервера; б) ЛВС с программно-аппаратной защитой сервера;
в) ЛВС с выходом в глобальную вычислительную систему;
г) ЛВС с защищенным обменом через открытую сеть

Поскольку все они, как правило, реализованы под управлением неаттестованных операционных систем, их можно рассматривать не как основные средства защиты, а лишь как вспомогательные.

На базе криптомаршрутизатора КРИПТОН-IP можно строить защищенные виртуальные корпоративные (частные) сети (Virtual Private Network – VPN), в которых сетевые соединения устанавливаются в интересах и по требованию определенных пользователей.

Типичная схема подключения локальной сети к высшей глобальной сети через криптомаршрутизаторы (КМ) представлена на рис.11.13. КМ обеспечивает контроль всего IP-трафика между внешней и внутренней сетями. К прохождению допускаются только пакеты, исходящие из узлов или предназначенные для узлов, связь с которыми разрешена. Кроме того, КМ обеспечивает прозрачное шифрование/дешифрование трафика, что позволяет использовать открытые каналы связи (в том числе и сеть Internet) для организации VPN. При использовании сети Internet КМ должен

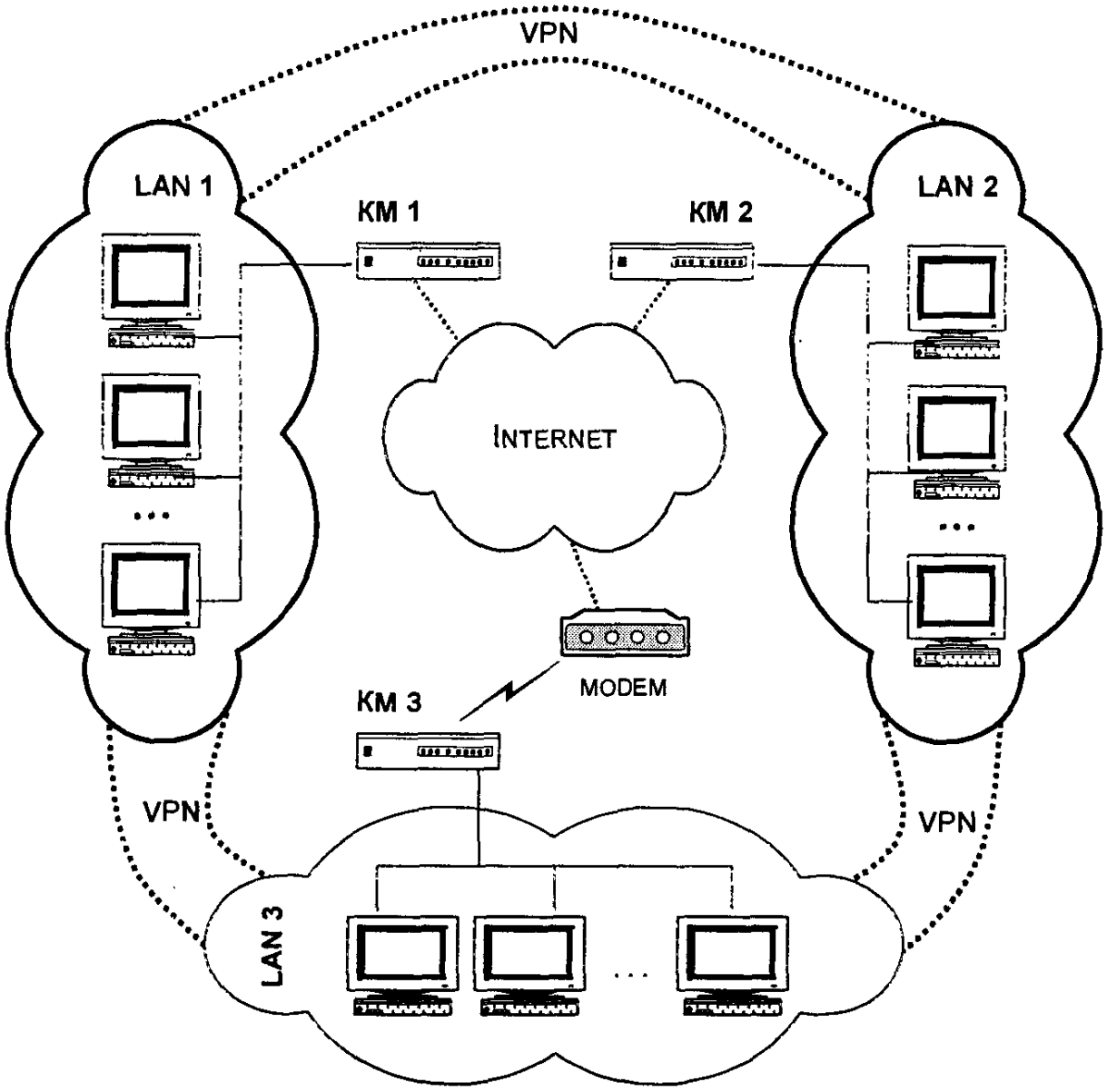


Рис.11.13. Типичная сетевая топология с использованием КМ

иметь корректно выделенный провайдером внешний IP-адрес, необходимый для правильной промежуточной маршрутизации. Организация работы КМ позволяет полностью скрывать топологию ЛС. В настоящий момент КМ может включать несколько сетевых интерфейсов (в том числе модем и мультипортовую плату) и поддерживать связь через последовательный порт (нуль-модем). КМ поддерживает сетевые адаптеры, соответствующие широко распространенному стандарту Ethernet.

Криптомаршрутизатор КРИПТОН-IP является средством защиты компьютерных сетей и позволяет создавать защищенные VPN на базе любых открытых сетей или на базе ЛС, использующих незащищенные линии связи. Кроме того, он может быть использован для разграничения передаваемой информации с разным уровнем доступа. Фильтрация трафика на IP-уровне и выбор правил политики безопасности позволяет применять КМ для организации как полностью защищенной сети, так и сети с выборочной передачей и приемом пакетов в открытом виде (с разрешением работы некоторым узлам сети без защиты трафика).

11.4. Технология работы с ключами

При использовании шифрования с открытым распределением ключей каждый пользователь должен иметь как минимум один секретный ключ и множество открытых ключей других абонентов. Понятно, что секретный ключ должен быть недоступен для других. Открытые ключи не являются секретными, но существует опасность их подмены.

Использование секретной дискеты или электронной карточки с ключами. Секретный ключ и все открытые ключи могут быть записаны на определенный ключевой носитель, в качестве которого может использоваться дискета, смарт-карта или Touch-Memory. Доступ к этим носителям должен быть только у их владельца. Однако при большом количестве открытых ключей такой вариант нецелесообразен, поскольку генерация ключей замедляется.

Предлагается следующий вариант работы с ключами. На ключевой дискете (или другом носителе) находятся:

- собственный секретный ключ (ключи);
- собственный открытый ключ (ключи);
- открытый ключ для проверки сертификата.

Напомним, что под сертификатом понимается открытый ключ с подписью ключом сертификационного центра (ключом администратора сети). Таким образом, минимально на ключевой дискете

могут находиться только два собственных ключа. В этом случае в качестве ключей для формирования и проверки сертификата могут использоваться собственные ключи.

Регистрация (сертификация) открытых ключей у администратора. Организуется сертификационный центр для регистрации пользователей сети. В сертификационный центр поступают открытые ключи и сопровождающие их документы. В ответ пользователь получает:

- открытые ключи с сертификатом всех зарегистрированных пользователей (в том числе и свой);
- файл или базу данных с полномочиями этих пользователей (также с сертификатом);
- открытый ключ для проверки сертификата как в виде файла, так и в распечатанном виде.

Пользователь при получении должен сначала проверить истинность открытого ключа для проверки сертификата, а затем – сертификаты всех полученных ключей и файлов. Также необходимо проверить свой открытый ключ. При положительных результатах проверки такие ключи можно использовать для шифрования.

Распределение открытых ключей по компьютерам и разрешение использования их для доступа к конкретному компьютеру или серверу осуществляет только администратор сети или сегмента сети.

Администрирование сети. Для обслуживания сегмента сети (и возможно, сети в целом) необходим администратор безопасности, который:

- генерирует (сертифицирует, регистрирует) ключи абонентов;
- определяет права доступа к абонентским пунктам и серверам;
- уточняет права доступа к отдельным фрагментам информации на серверах;
- устанавливает систему ограничения доступа на ПК;
- осуществляет текущий контроль за работой сети.

В качестве администратора безопасности можно использовать администратора сегмента сети при наличии у него необходимых прав допуска к информации. Однако, разграничение доступа между отдельными сегментами сети должен выполнять другой человек.

Разрешение на допуск к некоторой конкретной информации определяется при регистрации открытого ключа (и размещении его на соответствующем ПК). Для ограничения доступа к ПК достаточно убрать с него соответствующий ключ.

11.5. Программные продукты ЗАСТАВА фирмы "ЭЛВИС+" для защиты корпоративной сети

На основе опыта внедрения систем сетевой информационной безопасности ОАО "Элвис+" разработала общий концептуальный подход к решению задач построения защищенных корпоративных систем [92]. Суть решения заключается в следующем:

- построение жесткого периметра корпоративной части сети на основе технологий виртуальных защищенных сетей VPN (Virtual Private Network) с использованием протокола SKIP;
- обеспечение небольшого числа контролируемых точек открытого доступа в периметр корпоративной защищенной сети;
- построение эшелонированной системы защиты с контролем проникновения в защищенный периметр;
- обеспечение дистанционного администрирования и аудита всех компонентов системы защиты.

Эти решения обеспечивают построение виртуальных закрытых сетей (intranet), их безопасную эксплуатацию и интеграцию с открытыми коммуникационными системами.

Организация безопасного взаимодействия корпоративной сети с открытыми коммуникационными сетями

Защищенная корпоративная сеть, построенная на основе технологий VPN, не может быть полностью изолирована от внешних информационных систем, поскольку людям необходимо обмениваться почтой, новостями, получать данные из внешних информационных источников, что приводит к угрозе проведения атаки на информационные ресурсы корпоративной сети в рамках этого информационного обмена. Поэтому наряду с проблемой построения виртуальной защищенной сети предприятия существует другая важная проблема защиты корпоративной сети – организация безопасного взаимодействия с открытыми сетями.

Концепция защищенной корпоративной сети ОАО "ЭЛВИС+" состоит в том, чтобы закрыть трафик корпоративной сети средствами защиты информации сетевого уровня (построить виртуальную корпоративную сеть) и организовать фильтрацию информации в точках соединения с открытыми сетями. В качестве средств фильтрации информации на интерфейсах с открытыми сетями применяются традиционные решения – межсетевой экран (firewall), сервисы защиты типа proxy (посредник).

Важным элементом защиты от несанкционированного проникновения из открытой сети в корпоративную является последовательное (каскадное) включение нескольких фильтров-эшелонов защиты. Как правило, между открытой и корпоративной сетями устанавливается зона контролируемого доступа или "демилитари-

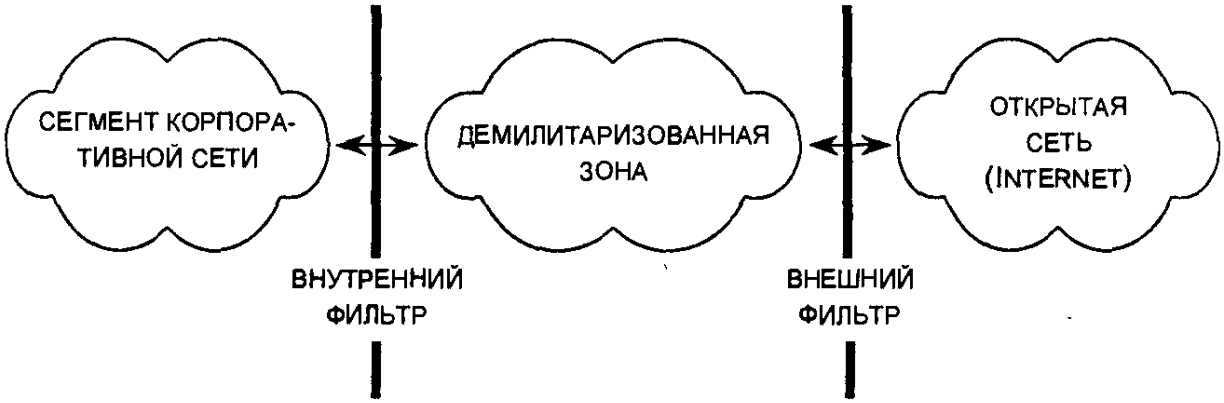


Рис.11.14. Демилитаризованная зона на интерфейсе между корпоративной и открытой сетями

зованная зона" (рис.11.14). В качестве внешнего и внутреннего фильтров применяются межсетевые экраны. Демилитаризованная зона представляет собой, как правило, сегмент сети, характеризующийся тем, что в нем представляются информационные ресурсы для доступа из открытой сети. При этом серверы, предоставляющие эти ресурсы для открытого доступа, конфигурируются специальным образом для того, чтобы на них не могли использоваться "опасные" сервисы (приложения), которые могут дать потенциальному нарушителю возможность реконфигурировать систему, компрометировать ее и, опираясь на скомпрометированные ресурсы, атаковать корпоративную сеть.

В решениях ОАО ЭЛВИС+ по организации взаимодействия с открытыми сетями обычно применяются межсетевые экраны, обеспечивающие так называемую расширенную пакетную фильтрацию. Такие пакетные экраны принимают "решение" о доступе каждого пакета на основе набора правил фильтрации, информации, содержащейся в пакете и некоторой предыстории, которую помнит фильтрационная машина, настраиваемая на обмены в рамках конкретных протоколов (протокольный автомат). Перепрограммируемые протокольные автоматы поставляются для большинства распространенных протоколов. Критерий фильтрации может быть основан на одном или нескольких правилах фильтрации.

Каждое правило формируется на основе применения операций отношения к таким элементам IP-пакета, как:

- IP-адрес источника/приемника пакета (эти правила позволяют разрешать или запрещать информационный обмен между некоторыми заданными узлами сети);
- поле "протокол" (TCP, UDP, ICMP и прочие) (правила фильтрации на основе этого поля регламентируют использование инкапсулируемых в IP-протоколов);

- поле "порт" для источника/приемника пакета (с понятием "порт" в стеке протоколов TCP/IP ассоциируется некоторое приложение и правила этой группы могут разрешать/запрещать доступ к заданному узлу по заданному прикладному протоколу (зависимость правил фильтрации по IP-адресам для пар источник/приемник позволяет контролировать направление доступа));
- бинарные данные с заданным смещением относительно заголовка IP.

На практике межсетевые экраны часто представляют собой программный продукт, который устанавливается на вычислительную платформу с несколькими сетевыми интерфейсами и обеспечивает сегментирование (рис.11.15) и независимую политику безопасности (набор правил фильтрации) для различных компьютеров в различных сегментах сети.

Централизованная архитектура системы, показанная на рис.11.15, не противоречит каскадной схеме построения защиты. Политика доступа между сегментами настраивается как независимый набор правил фильтрации для каждой пары интерфейсов (сегментов корпоративной сети). В примере на рис.11.15 можно предполагать следующую модельную настройку политики безопасности:

- внешние абоненты имеют доступ к открытому сегменту, например на корпоративный Web-сервер, по определенному набору коммуникационных протоколов (фильтр 1);
- пользователи корпоративной сети имеют доступ к информации высшей критичности, которая расположена в сегменте серверов (фильтр 3), а также могут, используя proxy-сервис в открытом сегменте, выходить в открытые сети (фильтры 2 и 1);

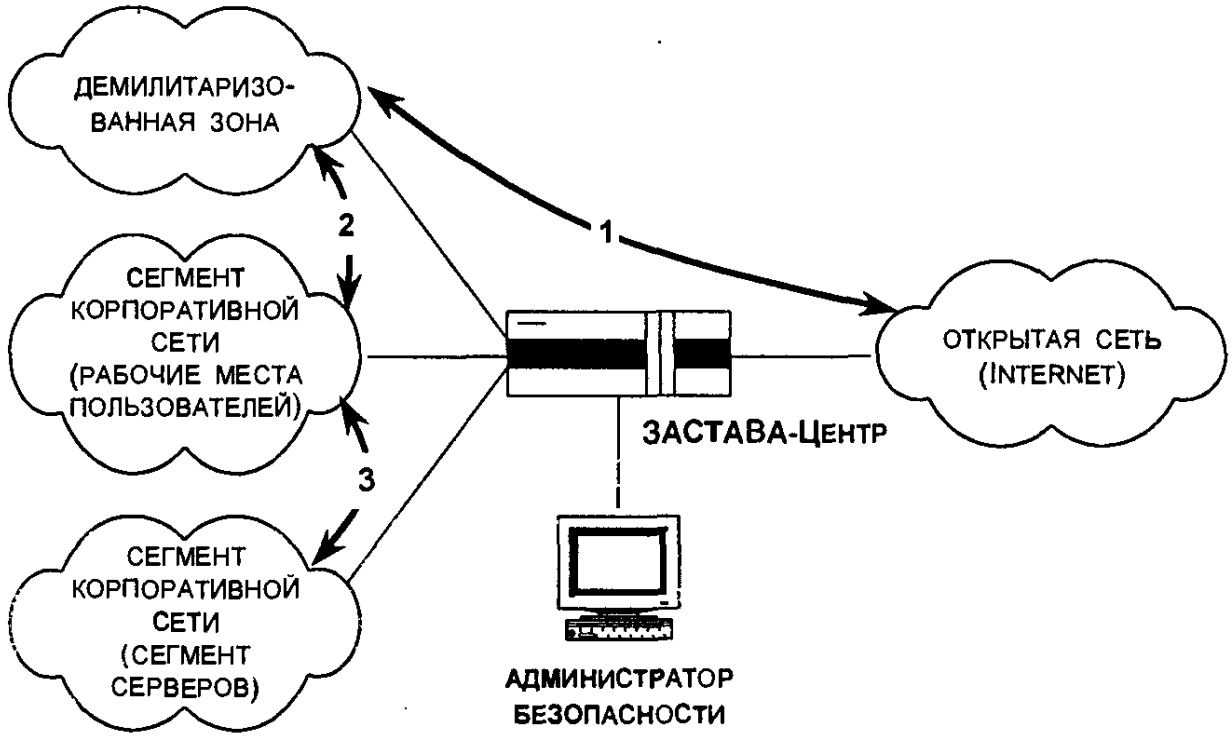


Рис.11.15. Сегментирование корпоративной сети

- администрирование безопасности производится дистанционно, обязательно с использованием средств защиты трафика.

В данном примере для атакующей стороны из внешней сети наиболее критичный ресурс (например, в сегменте серверов) может оказаться достижимым только при последовательной компрометации двух эшелонов защиты: демилитаризованной зоны и сегмента рабочих мест пользователей.

Программные продукты семейства ЗАСТАВА

Компания ЭЛВИС+ разработала ряд программных продуктов для построения VPN (защищенных корпоративных сетей) на основе стандарта IPSec. Назначение продуктов данного ряда заключается в обеспечении гибкого, масштабируемого решения для защиты и аутентификации трафика корпоративной сети и для защиты корпоративной сети от несанкционированного доступа.

В состав этого ряда продуктов входят:

- клиентские агенты для защиты отдельных рабочих мест (персональных компьютеров);
- программные агенты для защиты серверных платформ;
- шлюзы для защиты входящего и исходящего трафика сегмента корпоративной сети.

Продукты работают на операционных платформах Windows 95, NT, Solaris (SPARC и Intel), кроме того, обеспечена совместимость с платформами, соответствующими стандарту UNIX SVR 4.

В этих продуктах в качестве шифраторов используются (наряду с другими) программно-аппаратные средства серии КРИПТОН.

ЗАСТАВА-Персональный клиент. Продукт является средством защиты рабочей станции, находящейся в персональной эксплуатации. Программа содержит все средства администрирования и конфигурирования, необходимые для ее взаимодействия с любыми другими IPSec-совместимыми средствами защиты.

Функции продукта:

- защита и аутентификация трафика, реализация заданной дисциплины работы индивидуально для каждого защищенного соединения, разрешение доступа в заданном режиме только для санкционированных станций, контроль списка партнеров по взаимодействиям, защита от НСД из сети;
- настройка политики безопасности при помощи графического интерфейса продукта и/или при помощи внешне определенной конфигурации;
- сбор статистики и сигнализации.

ЗАСТАВА-Корпоративный клиент. Продукт является средством защиты рабочей станции корпоративной сети. От предыдущего продукта эта программа отличается тем, что пользователь за-

щищаемой рабочей станции лишен возможности единолично определять политику безопасности (и, следовательно, структуру сетевых соединений) для своей станции. Политика безопасности полностью контролируется администратором безопасности корпоративной сети и выдается пользователю как целостная структура данных на некотором носителе. Данные, определяющие политику безопасности, могут загружаться с внешнего носителя (дискеты, пластиковой карты) и существуют в защищаемом компьютере только в течение сеанса его работы, разрушаясь после прекращения работы компьютера.

ЗАСТАВА-Сервер. Продукт является функциональным аналогом продуктов семейства ЗАСТАВА-Клиент для серверных платформ. Отличается расширенными ресурсами для поддержания множественных соединений с клиентскими программными агентами. ЗАСТАВА-Сервер поддерживает защищенные соединения с мобильными пользователями, не имеющими фиксированных IP-адресов.

ЗАСТАВА-Офис – программный комплекс для коллективной защиты входящего и исходящего трафика сегмента локальной сети, защиты этого сегмента от несанкционированного доступа из внешней сети, а также для обеспечения защищенного взаимодействия с другими сегментами локальных сетей путем туннелирования трафика.

Межсетевой экран ЗАСТАВА-Центр представляет собой программный комплекс, предназначенный для контроля входящей и/или исходящей информации и защиты автоматизированной системы предприятия от несанкционированного доступа с использованием расширенной пакетной фильтрации на сетевом и транспортном уровнях. Межсетевой экран ЗАСТАВА предназначен для работы в операционных системах Solaris 2.5, 2.6 или более поздних версиях.

Функциональные возможности продукта:

- фильтрация на основе сетевых адресов отправителя и получателя;
- фильтрация с учетом входного и выходного сетевого интерфейса как средство проверки подлинности сетевых адресов;
- фильтрация с учетом любых значимых полей сетевых пакетов;
- фильтрация запросов на транспортном уровне на установление виртуальных соединений (при этом учитываются транспортные адреса отправителя и получателя);
- фильтрация запросов на прикладном уровне к прикладным сервисам (при этом учитываются прикладные адреса отправителя и получателя).

Межсетевой экран ЗАСТАВА обеспечивает возможность идентификации и аутентификации входящих и исходящих запросов методами, устойчивыми к пассивному и/или активному прослушиванию сети. Идентификация и аутентификация обеспечивается программными средствами с поддержкой протокола SKIP.

Межсетевой экран ЗАСТАВА обеспечивает:

- возможность регистрации и учета фильтруемых пакетов (в параметры регистрации включаются адрес, время и результат фильтрации);
- регистрацию и учет запросов на установление виртуальных соединений;
- локальную сигнализацию попыток нарушения правил фильтрации;
- идентификацию и аутентификацию администратора защиты при его локальных запросах на доступ.

Применение продуктов семейства ЗАСТАВА для защиты корпоративной сети

Ряд продуктов ОАО ЭЛВИС+ является функционально полным в том смысле, что решение проблем защиты может быть распространено на всю корпоративную сеть предприятия или организации любого масштаба. При этом может быть обеспечено решение следующих задач (рис.11.16):

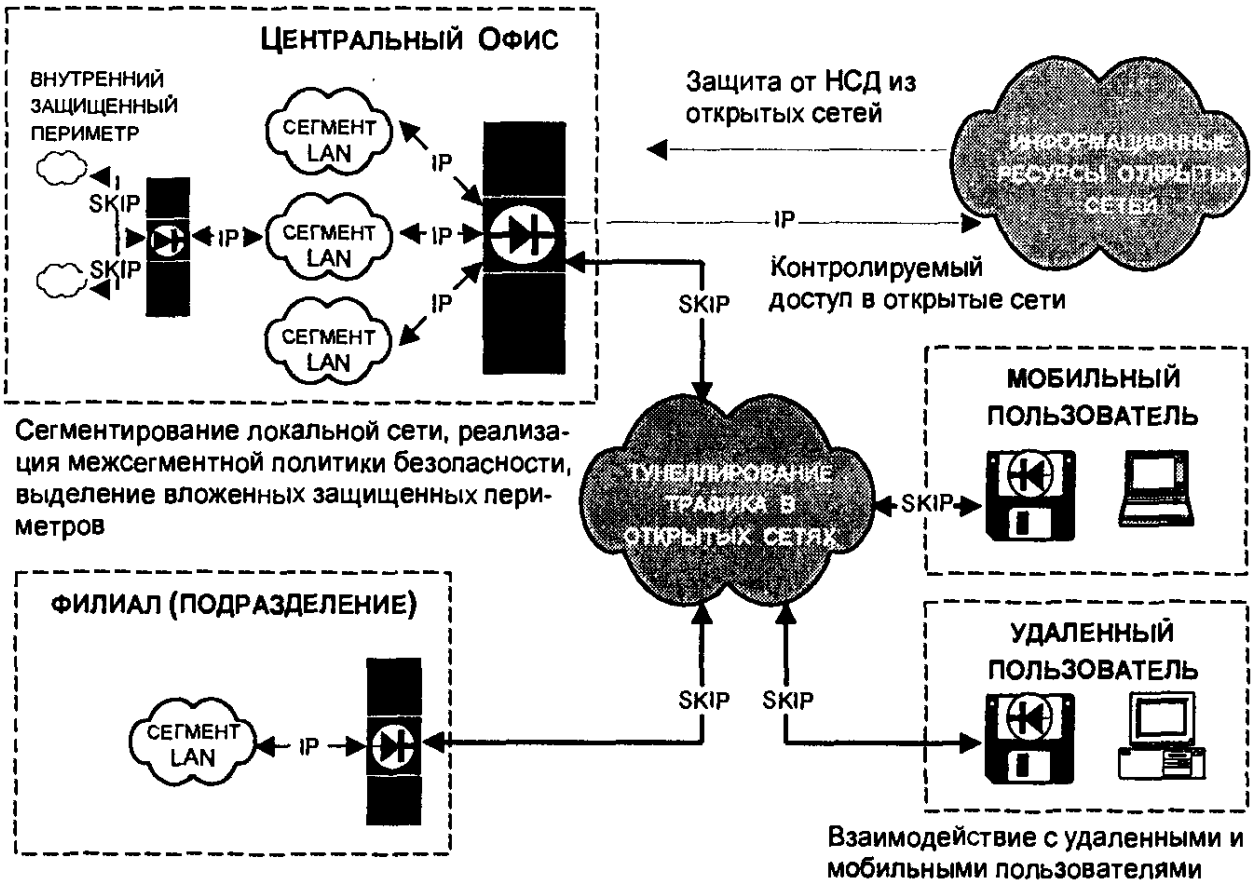


Рис.11.16. Пример схемы защищенной корпоративной сети

- построение надежной и прозрачной системы защиты трафика от перехвата и фальсификации как для связи между локальными сетями удаленных подразделений, так и для входа в корпоративную сеть уединенных удаленных (в том числе мобильных) пользователей, а также для построения абонентских сетей;
- обеспечение контроля сетевого доступа к информации (вплоть до обеспечения аутентифицированного доступа отдельных пользователей), построение эшелонированной системы защиты от атак, осуществляемых методами сетевого доступа;
- построение при необходимости системы вложенных защищенных периметров, ориентированных на работу с информацией различной степени конфиденциальности;
- построение системы событийного протоколирования и аудита с обеспечением возможности оперативного мониторинга безопасности в масштабах корпоративной сети;
- обеспечение централизованного дистанционного управления средствами сетевой защиты.

ГЛАВА 12. ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ ПЛАТЕЖНЫХ СИСТЕМ НА ОСНОВЕ СМАРТ-КАРТ И ПРОГРАММНО-АППАРАТНЫХ СРЕДСТВ ФИРМЫ АНКАД

Одна из наиболее важных сфер применения средств серии КРИПТОН – обеспечение безопасности платежных систем. Эта задача имеет комплексный характер. Рассмотрим один из ее аспектов – защиту информации, циркулирующей в аппаратно-программных объектах платежной системы.

12.1. Технические объекты платежной системы

К техническим объектам платежной системы относятся:

- платежная смарт-карта (карточка с однокристальным специализированным процессором, имеющим долговременную защищенную память, стандартизированные размеры и интерфейс);
- устройство чтения/записи (отдельное интерфейсное устройство или устройство в составе терминала, в которое вставляется смарт-карта, и взаимодействующие с смарт-картой механические и электрические устройства);
- терминальное оборудование (торговые терминалы, кассовые аппараты, банкоматы и другое оборудование), предназначенное для непосредственных операций со смарт-картой и передачи информации в центр обработки транзакций;
- автоматизированное рабочее место (АРМ) персонализации смарт-карт (персональный компьютер с устройством чтения/записи для записи на смарт-карты информации);
- АРМ операциониста банка (персональный компьютер с устройством чтения/записи);
- рабочая станция банка-эквайера, банка-эмитента, процессингового центра (персональный компьютер с устройством чтения/записи);
- сервер банка-эквайера, банка-эмитента, процессингового центра;
- каналы связи, объединяющие указанные выше компоненты.

На платежной смарт-карте обычно размещается следующая информация:

- ключи шифрования и подписи (может быть несколько наборов ключей: ключ аутентификации, ключи для связи с процессинговым центром, ключи для кредитования и дебетования и т.д);
- информация о наличии денег, о проведенных транзакциях и т.п.

12.2. Основные принципы обеспечения безопасности платежной системы

Идентификация и аутентификация субъектов платежной системы. До начала проведения любой операции устанавливаются: выпускающая смарт-карты организация, банк, торговая организация, владелец смарт-карты.

Данные выпускающей смарт-карты организации и номер смарт-карты записываются в однократно программируемую память и защищаются от изменения аппаратным способом.

Данные о банке и владельце смарт-карты могут быть изменены на смарт-карте в центре персонализации банка-эмитента при предъявлении соответствующих полномочий (ключей).

Данные о торговой организации вносятся непосредственно в терминал сервисной организации с соответствующих технологических смарт-карт вместе с ключами банков или со смарт-карт кассиров и продавцов.

Идентификация и аутентификация должна осуществляться на основе методов симметричной и асимметричной криптографии. Для решения этих задач фирма АНКАД предлагает библиотеки функций шифрования и вычисления имитовставки (ГОСТ 28147-89), хеширования (ГОСТ Р 34.11-94) и электронной цифровой подписи (ГОСТ Р 34.10-94), реализованные программным и аппаратным способами.

Проверка целостности данных. Система должна гарантировать, что данные в системе или на смарт-карте не будут изменены неправомочными пользователями неправомочным образом. Целостность данных в системе с шифрованием можно подтвердить имитовставкой, вычисленной на секретных ключах банков. Целостность информации о проведенных операциях (транзакциях) подтверждается имитовставками, вычисленными на секретных ключах участников операции (клиента, кассира и продавца). Для систем с электронной подписью вместо имитовставки можно использовать ЭЦП с проверкой ее достоверности сертифицированными ранее открытыми ключами. Можно комбинировать оба способа в зависимости от возможностей используемых устройств.

Проверка целостности данных может осуществляться с помощью библиотек (см. табл.11.1).

Определение происхождения информации. Все сделки и данные системы должны сопровождаться информацией, идентифицирующей их происхождение и назначение. При выполнении каждой операции фиксируются дата и время ее проведения, участвующие в ней субъекты и объекты. Информация об операции подтверждается сертификатами каждого участника (имитовставкой или электронной подписью). Действительность операции можно проверить на основании соответствующих ключей ее участников. Эти ключи знает участник и банк.

Для проверки происхождения информации предназначены те же самые библиотеки.

Обеспечение секретности данных. Система должна гарантировать конфиденциальность данных. Только допущенный к системе пользователь имеет возможность просмотра данных. Секретность гарантируется применением шифрования как в каналах связи, так и внутри устройств.

Для ограничения доступа к оборудованию на базе ПК предлагаются системы защиты от несанкционированного доступа КРИПТОН-ВЕТО или КРИПТОН-ЗАМОК. Защита ПК со стороны сети и каналов связи осуществляется с помощью "прозрачно" шифрующих коммуникационных программ или криптошлюзов и криптомаршрутизаторов.

Защита кредитных операций. Кредитные операции должны выполняться непосредственно в банке или в режиме on-line. Разрешается их проводить и в режиме off-line для определенных сумм кредита. Система должна защищать проведение кредитных операций (перевод денег со счета на смарт-карту) путем:

- проверки банком действительности смарт-карты;
- проверки действительности терминала банка;
- определения владельца карточки по PIN коду;
- создания подтверждающего электронного сертификата;
- подтверждения сертификатом проведенной операции;
- вывода смарт-карты из обращения при попытке мошенничества с возможностью восстановления.

Сумма кредита подписывается специальным ключом банка (создание сертификата), если терминалы, работающие в режиме off-line, умеют проверять электронную подпись. Иначе формируется имитовставка на все данные банка и клиента, включая и сумму кредита. Подтверждение сертификатом проведенной операции выполняется в виде электронной подписи работника банка.

Для решения приведенных выше задач предлагается использовать библиотеки криптографических функций фирмы АНКАД.

Защита дебитных операций. При приобретении товаров или услуг производятся:

- проверка терминалом торгового предприятия действительности смарт-карты;
- проверка картой действительности торгового терминала;
- определение владельца смарт-карты по PIN коду;
- проверка наличия средств на смарт-карте;
- создание подтверждающего электронного сертификата;
- подтверждение сертификатом проведенной операции;
- вывод смарт-карты из обращения при попытке мошенничества с возможностью восстановления.

Все проверки выполняются аналогично операции кредитования, только в качестве работника банка выступает кассир магазина. Проверка наличия средств на смарт-карте производится путем сравнения суммы услуг с остатком средств на смарт-карте. На смарт-карте записывается с нарастанием сумма истраченных средств. Она сравнивается с предоставленной суммой кредита. Последняя может быть изменена только банком-эмитентом.

Согласование проведенных операций. Все кредитные и дебетовые операции, проведенные в течение дня в каждом магазине, собираются, согласовываются и передаются в банк. При этом:

- удостоверяется действительность торгового терминала;
- проверяется подлинность сделки;
- проверяется целостность данных о сделке.

Проверки должны производиться на основе оценки сертификатов сделок с помощью библиотеки криптографических функций.

Обеспечение безопасности смарт-карты. Обеспечение безопасности смарт-карты должно осуществляться на всех этапах ее жизненного цикла – от производства кристалла, транспортировки, персонализации до защиты информации на ней от самого владельца карты. При производстве и транспортировке могут быть искажены или модифицированы характеристики карты с целью последующего получения доступа к ресурсам платежной системы или нарушения ее функционирования.

Для определения качества карточки, выходящей из производства, используется выборочный контроль на соответствие топологии кристалла и содержимого ОС эталонным образцам. При этом только применение отечественной смарт-карты в достаточной степени может гарантировать надежность контроля.

Основная опасность утечки конфиденциальной информации о владельце карты существует на этапе персонализации карты. Ра-

бочее место для персонализации карт должно иметь надежную систему защиты от НСД, в качестве которой может быть использована система КРИПТОН-ВЕТО или КРИПТОН-ЗАМОК.

Защита конфиденциальной информации на карточке от ее владельца или постороннего злоумышленника (при краже карты) обеспечивается использованием физико-технологических методов защиты от НСД к кристаллу и ОС, а также механизмом аутентификации владельца.

Защита оборудования и программного обеспечения. Все оборудование и программное обеспечение, используемое в системе, должно подвергаться проверке, с тем чтобы оно не могло быть использовано в недобросовестных целях. Все терминалы должны быть защищены от несанкционированного доступа аппаратными и программными средствами. Терминалы не должны работать без загрузки определенных ключей. Следует вести журналы работы с фиксацией попыток несанкционированного доступа или использования устройств.

Эта задача также решается с помощью системы КРИПТОН-ВЕТО или КРИПТОН-ЗАМОК.

Защита устройств чтения/записи. Основные угрозы заключаются в перехвате, модификации, вторичном использовании информации, передаваемой по каналу связи, и считывании ключевой информации из устройства. Для смарт-карт с симметричными методами криптографии ситуация усложняется тем, что компрометация ключа в одном устройстве приводит к необходимости замены ключей во всех устройствах.

Безопасность устройства строится на закрытии канала связи с применением методов симметричной и асимметричной криптографии. В настоящее время имеется единственный отечественный ридер SCAT-200 (разработки фирмы АНКАД) с шифрованием по ГОСТ 28147-89 и библиотеками к нему для различных смарт-карт под управлением операционных систем DOS и Windows 95/98. Сохранность ключевой информации обеспечивается модулем безопасности устройства, в составе которого работает ридер. Фирма разрабатывает защищенное устройство чтения/записи (с модулем безопасности) по требованию заказчика.

Защита терминального оборудования. Для терминального оборудования имеют место те же угрозы, что и для считывающих устройств. Кроме того, существуют угрозы нарушения целостности программного обеспечения и перехвата, модификации, уничтожения информации в канале терминал – рабочая станция банка.

Для защиты терминального оборудования, реализованного на основе ПК, предлагается использовать:

- систему защиты от НСД КРИПТОН-ВЕТО или КРИПТОН-ЗАМОК;
- коммуникационные программы прозрачного шифрования IP-пакетов и ограничения доступа к компьютеру по сети;
- библиотеки функций шифрования и электронной цифровой подписи для различных операционных систем.

Персональные компьютеры, рабочие станции, серверы процессинговых центров и банков и каналы связи. Подробное описание угроз для этого оборудования и соответствующих методов защиты было приведено ранее. В качестве средств защиты можно использовать:

- систему КРИПТОН-ВЕТО или КРИПТОН-ЗАМОК;
- коммуникационные программы прозрачного шифрования IP-пакетов и ограничения доступа к компьютеру по сети;
- криптомаршрутизаторы (для защиты серверов и сегментов сетей);
- библиотеки функций шифрования и электронной цифровой подписи для различных операционных систем.

ПРИЛОЖЕНИЕ

ЭЛЕМЕНТЫ ТЕОРИИ ЧИСЕЛ

Модулярная арифметика

Модулярная арифметика часто изучается в школе как "арифметика часов". Если отсчитать 14 часов от 3 часов после полудня, то получится 5 часов утра следующего дня:

$$3 + 14 \equiv 5 \pmod{12}$$

или

$$(3 + 14) \bmod 12 = 5.$$

Это арифметика по модулю 12.

Обычная запись в модулярной арифметике

$$a \equiv b \pmod{n}$$

читается так: "a сравнимо с b по модулю n". Это соотношение справедливо для целых значений a, b и $n \neq 0$, если, и только если

$$a = b + k * n$$

для некоторого целого k.

Отсюда, в частности, следует

$$n \mid (a - b).$$

Это читается как "n делит $(a - b)$".

Если $a \equiv b \pmod{n}$,

то b называют *вычетом* числа a по модулю n.

Операцию нахождения вычета числа a по модулю n

$$a \pmod{n}$$

называют приведением числа a по модулю n или *приведением по модулю.*

В нашем примере

$$(3 + 14) \bmod 12 = 17 \bmod 12 = 5$$

или

$$17 \equiv 5 \pmod{12},$$

число 5 является вычетом числа 17 по модулю 12.

Набор целых чисел от 0 до (n–1) называют *полным набором вычетов по модулю* n. Это означает, что для любого целого a(a>0) его вычет r по модулю n есть некоторое целое число в интервале от 0 до (n–1), определяемое из соотношения

$$r = a - k * n,$$

где k – целое число.

Например, для n=12 полный набор вычетов:

$$\{0, 1, 2, \ldots, 11\}.$$

Обычно предпочитают использовать вычеты

$$r \in \{0, 1, 2, \ldots, n-1\},$$

но иногда полезны вычеты в диапазоне целых:

$$r \in \left\{-\frac{1}{2}(n-1), \ldots, \frac{1}{2}(n-1)\right\}.$$

Заметим, что

$$-12 (\text{mod } 7) \equiv -5 (\text{mod } 7) \equiv 2 (\text{mod } 7) \equiv 9 (\text{mod } 7) \text{ и т.д.}$$

Модулярная арифметика аналогична во многом обычной арифметике: она коммутативна, ассоциативна и дистрибутивна. Точнее говоря, целые числа по модулю n с использованием операций сложения и умножения образуют коммутативное кольцо при соблюдении законов ассоциативности, коммутативности и дистрибутивности.

Фактически мы можем либо сначала приводить по модулю n, а затем выполнять операции, либо сначала выполнять операции, а затем приводить по модулю n, поскольку приведение по модулю n является *гомоморфным отображением* из кольца целых в кольцо целых по модулю n:

$$(a + b) \bmod n = [a(\text{mod } n) + b(\text{mod } n)] \bmod n,$$

$$(a - b) \bmod n = [a(\text{mod } n) - b(\text{mod } n)] \bmod n.$$

$$(a * b) \bmod n = [a(\text{mod } n) * b(\text{mod } n)] \bmod n.$$

$$[a * (b + c)] \bmod n = \{[a * b(\text{mod } n)] + [a * c(\text{mod } n)]\} \bmod n.$$

Криптография использует множество вычислений по модулю n, потому что задачи типа вычисления дискретных логарифмов и квадратных корней очень трудны. Кроме того, с вычислениями по модулю удобнее работать, потому что они ограничивают диапазон всех промежуточных величин и результата.

Для модуля n длиной k бит промежуточные результаты любого сложения, вычитания или умножения будут не длиннее 2k бит. Поэтому возведение в степень в модулярной арифметике можно выполнить без генерации очень больших промежуточных результатов.

Вычисление степени числа a по модулю n

$$a^x \bmod n$$

можно выполнить как ряд умножений и делений. Существуют способы сделать это быстрее. Поскольку эти операции дистрибутивны, быстрее произвести возведение в степень как ряд последовательных умножений, выполняя каждый раз приведение по модулю. Это особенно заметно, если работать с длинными числами (200 бит и более).

Например, если нужно вычислить

$$a^8 \bmod n,$$

не следует применять примитивный подход с выполнением семи перемножений и одного приведения по модулю громадного числа:

$$(a * a * a * a * a * a * a * a) \bmod n.$$

Вместо этого выполняют три малых умножения и три малых приведения по модулю:

$$((a^2 \bmod n)^2 \bmod n)^2 \bmod n.$$

Тем же способом вычисляют

$$a^{16} \bmod n = (((a^2 \bmod n)^2 \bmod n)^2 \bmod n)^2 \bmod n.$$

Вычисление

$$a^x \bmod n,$$

где x не является степенью 2, лишь немного сложнее. Двоичная запись числа x позволяет представить число x как сумму степеней 2:

$$x = 25_{(10)} \rightarrow 1\ 1\ 0\ 0\ 1_{(2)}, \quad \text{поэтому} \quad 25 = 2^4 + 2^3 + 2^0.$$

Тогда

$$a^{25} \bmod n = (a * a^{24}) \bmod n = (a * a^8 * a^{16}) \bmod n =$$

$$= a * ((a^2)^2)^2 * (((a^2)^2)^2)^2 \bmod n = ((((a^2 * a)^2)^2)^2 * a) \bmod n.$$

При разумном накоплении промежуточных результатов потребуется только шесть умножений:

$$(((((((a^2 \bmod n) * a) \bmod n)^2 \bmod n)^2 \bmod n)^2 \bmod n) * a) \bmod n.$$

Этот метод уменьшает трудоемкость вычислений до 1,5xk операций в среднем, где k – длина числа в битах [123].

Поскольку многие алгоритмы шифрования основаны на возведении в степень по модулю n, целесообразно использовать алгоритмы быстрого возведения в степень.

Алгоритм Евклида для нахождения наибольшего общего делителя

Целое число a делит без остатка другое целое число b, если, и только если

$$b = k * a$$

для некоторого целого числа k. В этом случае число a называют *делителем числа* b или *множителем в разложении* числа b на множители.

Пусть a – целое число, большее 1. Тогда a является *простым числом*, если его единственными положительными делителями будут 1 и само a, в противном случае a называется *составным*.

Любое целое n>1 может быть представлено единственным образом с точностью до порядка сомножителей как произведение простых [45].

Существенный с точки зрения криптографии факт состоит в том, что не известно никакого эффективного алгоритма разложения чисел на множители; не было получено и никакой нетривиальной нижней оценки временной сложности разложения. Никаких эффективных методов не известно даже в таком простом случае, когда необходимо восстановить два простых числа p и q из их произведения:

$$n = p * q.$$

Наибольший общий делитель чисел a и b, обозначаемый как НОД (a, b) или просто (a, b), – это наибольшее целое, делящее одновременно числа a и b. В эквивалентной форме (a, b) – это то единственное натуральное число, которое делит a и b и делится на любое целое, делящее и a и b. Если НОД (a, b)=1, то целые a и b – *взаимно простые*.

Наибольший общий делитель может быть вычислен с помощью *алгоритма Евклида*. Евклид описал этот алгоритм в своей книге "Начала", написанной около 300 лет до н.э. Он не изобрел его. Историки полагают, что этот алгоритм, возможно, старше еще на 200 лет. Это древнейший нетривиальный алгоритм, который просуществовал до настоящего времени и все еще хорош и сегодня.

Опишем алгоритм Евклида для нахождения НОД (a, b). Введем обозначения: q_i – частное; r_i – остаток. Тогда алгоритм можно представить в виде следующей цепочки равенств:

$$\begin{aligned}
a &= b * q_1 + r_1, & 0 < r_1 < b,\\
b &= r_1 * q_2 + r_2, & 0 < r_2 < r_1,\\
r_1 &= r_2 * q_3 + r_3, & 0 < r_3 < r_2,\\
&\vdots & \vdots\\
r_{k-2} &= r_{k-1} * q_k + r_k, & 0 < r_k < r_{k-1},\\
r_{k-1} &= r_k * q_{k+1}.
\end{aligned}$$

Остановка гарантируется, поскольку остатки r_i от делений образуют строго убывающую последовательность натуральных чисел. Из этой цепочки немедленно получаем, что r_k есть общий делитель чисел a и b и, более того, что любой общий делитель чисел a и b делит и r_k. Таким образом, r_k=НОД(a, b) или r_k=(a, b).

Алгоритм Евклида для вычисления наибольшего общего делителя

```
begin
        g0: = b;
        g1: = a;
         i : =1.
        while  gi! = 0   do
        begin
                gi+1 : = gi–1 mod gi;
                   i : = i + 1
        end
       gcd : = gi–1{gcd – результат}
end
```

Вычисление обратных величин

В арифметике действительных чисел нетрудно вычислить мультипликативную обратную величину a^{-1} для ненулевого a:

$$a^{-1}=1/a \quad \text{или} \quad a * a^{-1}=1.$$

Например, мультипликативная обратная величина от числа 4 равна 1/4, поскольку

$$4 * \frac{1}{4} = 1.$$

В модулярной арифметике вычисление обратной величины является более сложной задачей. Например, решение сравнения

$$4 * x \equiv 1 (\text{mod } 7)$$

эквивалентно нахождению таких значений x и k, что

$$4 * x \equiv 7 * k + 1,$$

где x и k – целые числа.

Общая формулировка этой задачи – нахождение такого целого числа x, что

$$a * x (\text{mod } n) = 1.$$

Можно также записать

$$a^{-1} \equiv x (\text{mod } n).$$

Решение этой задачи иногда существует, а иногда его нет. Например, обратная величина для числа 5 по модулю 14 равна 3, поскольку

$$5 * 3 = 15 \equiv 1 (\text{mod } 14).$$

С другой стороны, число 2 не имеет обратной величины по модулю 14.

Вообще сравнение

$$a^{-1} \equiv x(\text{mod } n)$$

имеет единственное решение, если a и n – взаимно простые числа.

Если числа a и n не являются взаимно простыми, тогда сравнение

$$a^{-1} \equiv x(\text{mod } n)$$

не имеет решения [45].

Сформулируем основные способы нахождения обратных величин. Пусть целое число $a \in \{0,1,2,\ldots,n-1\}$. Если НОД$(a,n)=1$, то $a*i(\text{mod}\, n)$ при $i=0,1,2,\ldots,n-1$ является перестановкой множества $\{0,1,2,\ldots,n-1\}$.

Например, если $a=3$ и $n=7$(НОД$(3,7)=1$), то

$$3 * i(\text{mod } 7) \quad \text{при} \quad i = 0, 1, 2, \ldots, 6$$

является последовательностью 0, 3, 6, 2, 5, 1, 4, т.е. перестановкой множества $\{0, 1, 2, \ldots, 6\}$.

Это становится неверным, когда НОД$(a,n) \neq 1$. Например, если $a=2$ и $n=6$, то

$$2 * i(\text{mod } 6) \equiv 0, 2, 4, 0, 2, 4 \quad \text{при} \quad i = 0, 1, 2, \ldots, 5.$$

Если НОД$(a,n)=1$, тогда существует обратное число a^{-1}, $0<a^{-1}<n$, такое, что

$$a * a^{-1} \equiv 1(\text{mod } n).$$

Действительно, $a*i(\text{mod}\, n)$ является перестановкой $0,1,\ldots,n-1$, поэтому существует i, такое, что

$$a * i \equiv 1(\text{mod } n).$$

Как уже отмечалось, набор целых чисел от 0 до $n-1$ называют *полным набором вычетов* по модулю n. Это означает, что для любого целого числа $a(a>0)$ его вычет $r=a(\text{mod}\, n)$ – это некоторое целое число в интервале от 0 до $n-1$.

Выделим из полного набора вычетов подмножество вычетов, взаимно простых с n. Такое подмножество *называют приведенным набором вычетов.*

Пример. Пусть модуль $n=11$ – простое число. Полный набор вычетов по модулю 11

$$\{0, 1, 2, \ldots, 10\}.$$

При формировании приведенного набора вычетов из них удаляется только один элемент – 0. Приведенный набор вычетов по модулю 11 имеет $11-1=10$ элементов.

Вообще приведенный набор вычетов по модулю простого числа n имеет $n-1$ элементов.

Пример. Пусть модуль n=10. Полный набор вычетов по модулю n=10

$$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}.$$

Из них только 1, 3, 7, 9 не имеют общего сомножителя с числом 10. Поэтому приведенный набор вычетов по модулю 10 равен {1, 3, 7, 9}. При формировании этого приведенного набора были исключены элементы:

	0	(1 элемент),
кратные	2	(4 элемента),
кратные	5	(1 элемент),

т.е. всего шесть элементов. Вычитая их из 10, получаем 10–1–4–1=4, т.е. четыре элемента в приведенном наборе.

Для произведения простых чисел p*q=n приведенный набор вычетов имеет (p–1)(q–1) элементов. При n=p*q=2*5=10 число элементов в приведенном наборе

$$(p-1)(q-1) = (2-1)(5-1) = 4.$$

Пример. Приведенный набор вычетов по модулю $27=3^3$ имеет 18 элементов:

$$\{1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 20, 22, 23, 25, 26\}.$$

Из полного набора вычетов исключены элементы, кратные 3 (всего девять элементов).

Для модуля в виде простой степени n^r приведенный набор вычетов имеет $n^{r-1}(n-1)$ элементов.

При n=3, r=3 получаем $3^{3-1}(3-1)=3^2*2=18$.

Функция Эйлера $\varphi(n)$ характеризует *число элементов в приведенном наборе вычетов* (табл. П.1).

Таблица П.1

Модуль n	Функция $\varphi(n)$
n – простое n^2 ... n^r	$n-1$ $n(n-1)$ ... $n^{r-1}(n-1)$
p * q (p, q – простые) ... $\prod_{i=1}^{t} p_i^{e_i}$ (p_i – простые)	$(p-1)(q-1)$ ... $\prod_{i=1}^{t} p_i^{e_i-1}(p_i-1)$

Иначе говоря, функция $\varphi(n)$ – это количество положительных целых, меньших n, которые взаимно просты с n [123].

Малая теорема Ферма: если n – простое и НОД(a, n)=1, то

$$a^{n-1} \equiv 1 (\text{mod } n).$$

Согласно обобщению Эйлером малой теоремы Ферма имеем: если НОД(a, n)=1, то

$$a^{\varphi(n)} \equiv 1 (\text{mod } n).$$

Если п – простое число, то предыдущий результат, учитывая, что $\varphi(п)=п-1$, приводится к виду (малой теоремы Ферма)

$$a^{п-1} \equiv 1 (\mathrm{mod}\ п).$$

Основные способы нахождения обратных величин

$$a^{-1} \equiv 1 (\mathrm{mod}\ п).$$

1. Проверить поочередно значения $1, 2, \ldots, п-1$, пока не будет найден $a^{-1} \equiv 1 (\mathrm{mod}\ п)$, такой, что $a * a^{-1} (\mathrm{mod}\ п) \equiv 1$.

2. Если известна функция Эйлера $\varphi(п)$, то можно вычислить

$$a^{-1} (\mathrm{mod}\ п) \equiv a^{\varphi(п)-1} (\mathrm{mod}\ п),$$

используя алгоритм быстрого возведения в степень.

3. Если функция Эйлера $\varphi(п)$ не известна, можно использовать расширенный алгоритм Евклида.

Проиллюстрируем эти способы на числовых примерах.

1. Поочередная проверка значений $1, 2, \ldots, п-1$, пока не будет найден $x = a^{-1} (\mathrm{mod}\ п)$, такой что $a * x \equiv 1 (\mathrm{mod}\ п)$.

Пусть $п=7$, $a=5$. Требуется найти $x = a^{-1} (\mathrm{mod}\ п)$.

$$a * x \equiv 1 (\mathrm{mod}\ п) \quad \text{или} \quad 5 * x \equiv 1 (\mathrm{mod}\ 7).$$

$$п - 1 = 7 - 1 = 6.$$

Получаем $x = 5^{-1} (\mathrm{mod}\ 7) = 3$.

Результаты проверки сведены в табл. П.2.

Таблица П.2

x	5 * x	5 * x (mod 7)
1	5	5
2	10	3
3	15	1
4	20	6
5	25	4
6	30	2

2. Нахождение $a^{-1} (\mathrm{mod}\ п)$, если известна функция Эйлера $\varphi(п)$.

Пусть $п=7$, $a=5$. Найти $x = a^{-1} (\mathrm{mod}\ п) = 5^{-1} (\mathrm{mod}\ 7)$. Модуль $п=7$ – простое число. Поэтому функция Эйлера $\varphi(п) = \varphi(7) = п - 1 = 6$. Обратная величина от 5 по mod 7

$$a^{-1} (\mathrm{mod}\ п) = a^{\varphi(п)-1} (\mathrm{mod}\ п) =$$

$$= 5^{6-1} \bmod 7 = 5^5 \bmod 7 = (5^2 \bmod 7)(5^3 \bmod 7) \bmod 7 =$$

$$= (25 \bmod 7)(125 \bmod 7) \bmod 7 = (4 * 6) \bmod 7 = 24 \bmod 7 = 3.$$

Итак, $x = 5^{-1} (\mathrm{mod}\ 7) = 3$.

3. Нахождение обратной величины $a^{-1} (\mathrm{mod}\ п)$ с помощью расширенного алгоритма Евклида.

Алгоритм Евклида можно обобщить способом, который имеет большое практическое значение. При этом способе во время вычисления НОД(a, b) можно попутно вычислить такие целые числа u_1 и u_2, что

$$a * u_1 + b * u_2 = НОД\ (a, b).$$

Это обобщение (расширение) алгоритма Евклида удобно описать, используя векторные обозначения [45].

Расширенный алгоритм Евклида

При заданных неотрицательных целых числах a и b этот алгоритм определяет вектор

$$(u_1, u_2, u_3),$$

такой, что

$$a * u_1 + b * u_2 = u_3 = НОД\ (a, b).$$

В процессе вычисления используются вспомогательные векторы (v_1, v_2, v_3), (t_1, t_2, t_3). Действия с векторами производятся таким образом, что в течение всего процесса вычисления выполняются соотношения

$$a * t_1 + b * t_2 = t_3, \quad a * u_1 + b * u_2 = u_3, \quad a * v_1 + b * v_2 = v_3.$$

Для вычисления обратной величины $a^{-1}(\mathrm{mod}\ п)$ используется частный режим работы расширенного алгоритма Евклида, при котором b=п, НОД(a, п)=1, и этот алгоритм определяет вектор

$$(u_1, u_2, u_3),$$

такой, что

$$u_3 = 1, \quad a * u_1 + п * u_2 = НОД(a, п) = 1,$$

$$(a * u_1 + п * u_2)\ \mathrm{mod}\ п \equiv a * u_1 (\mathrm{mod}\ п) \equiv 1,$$

$$a^{-1}(\mathrm{mod}\ п) \equiv u_1(\mathrm{mod}\ п).$$

Шаги алгоритма:

1. Начальная установка.

Установить $(u_1, u_2, u_3) := (0, 1, п)$.

$$(v_1, v_2, v_3) := (1, 0, a).$$

2. $u_3 = 1$?. Если $u_3 = 1$, то алгоритм заканчивается.

3. Разделить, вычесть.

Установить $q := [u_3/v_3]$.

Затем установить

$$(t_1, t_2, t_3) := (u_1, u_2, u_3) - (v_1, v_2, v_3) * q,$$

$$(u_1, u_2, u_3) := (v_1, v_2, v_3),$$

$$(v_1, v_2, v_3) := (t_1, t_2, t_3).$$

Возвратиться к шагу 2.

Пример. Заданы модуль n=23 и число a=5. Найти обратное число $a^{-1}(\text{mod}23)$, т.е. $x=5^{-1}(\text{mod}23)$.

Используя расширенный алгоритм Евклида, выполним вычисления, записывая результаты отдельных шагов в табл. П.3.

Таблица П.3

q	u_1	u_2	u_3	v_1	v_2	v_3
–	0	1	n = 23	1	0	a = 5
4	1	0	5	–4	1	3
1	–4	1	3	5	–1	2
1	5	–1	2	–9	2	1
–	–9	2	1			

При $u_3 = 1$, $u_1 = -9$, $u_2 = 2$

$$(a * u_1 + n * u_2) \bmod n = (5 * (-9) + 23 * 2) \bmod 23 =$$

$$= 5 * (-9) \bmod 23 \equiv 1,$$

$$a^{-1} (\text{mod } n) = 5^{-1} (\text{mod } 23) = (-9) \bmod 23 = (-9 + 23) \bmod 23 = 14.$$

Итак, $x \equiv 5^{-1} (\text{mod } 23) \equiv 14 (\text{mod } 23) = 14$.

Для решения более сложных сравнений

$$a * x \equiv b \ (\text{mod } n), \quad \text{т.е.} \quad b \neq 1, \quad x = ?$$

используется следующий прием. Сначала решают сравнение

$$a * y \equiv 1 (\text{mod } n),$$

т.е. определяют

$$y = a^{-1} (\text{mod } n),$$

а затем находят

$$x = a^{-1} b \ (\text{mod } n) = y * b \ (\text{mod } n).$$

Пример. Найти x для сравнения

$$5 * x \equiv 9 \ (\text{mod } 23).$$

Сначала решаем сравнение

$$5 * y \equiv 1 (\text{mod } 23).$$

Получаем $y \equiv 5^{-1} (\text{mod } 23) = 14$. Затем находим

$$x = 5^{-1} * 9 \ (\text{mod } 23) = 14 * 9 \ (\text{mod } 23) = 126 \ (\text{mod } 23) \equiv 11 (\text{mod } 23),$$

$$x = 11.$$

Китайская теорема об остатках

Любое неотрицательное целое число, не превосходящее произведения модулей, можно однозначно восстановить, если известны его вычеты по этим модулям. Этот результат был известен еще в древнем Китае и носит название *китайской теоремы об остатках*. Теорема была предложена китайским математиком первого века Сун Це. Китайская теорема об остатках является мощным криптографическим инструментом.

Китайская теорема об остатках формулируется следующим образом.

Пусть $m_1, m_2, \ldots, m_t$ – модули (целые числа, большие 1), которые являются попарно взаимно простыми, т.е. НОД$(m_i, m_j)=1$ при $i \neq j$.

Пусть $a_1, a_2, \ldots, a_t$ – тоже целые числа, $0 \le a_i \le m_i$.

Пусть $M = m_1 * m_2 * \ldots * m_t$ – произведение всех m_i. Обозначим $M_i = M/m_i$.

И пусть N_i будет обратным к $M_i (\mathrm{mod}\, m_i)$, $i=1,2,\ldots,t$, т.е. $M_i * N_i \equiv 1 (\mathrm{mod}\, m_i)$.

Так как НОД$(M_i, m_i)=1$, то обратный элемент N_i существует и легко находится из алгоритма Евклида из соотношения

$$M_i * N_i + m_i * n_i = 1, \quad i = 1, 2, \ldots, t.$$

Сравнения $x \equiv a_i (\mathrm{mod}\, m_i)$, $i=1,2,\ldots,t$, имеют в интервале $[0, M-1]$ единственное общее решение

$$x = \sum_{i=1}^{t} a_i * N_i * M_i \ (\mathrm{mod}\ M).$$

Рассмотрим *частный случай.* Пусть $M = m_1 * m_2$, где m_1, m_2 – взаимно простые числа. Тогда для произвольных целых $a_1 < m_1$ и $a_2 < m_2$ существует единственное число x, $x<M$, такое, что

$$x \equiv a_1 (\mathrm{mod}\ m_1) \quad \text{и} \quad x \equiv a_2 (\mathrm{mod}\ m_2).$$

Чтобы найти значение решения x, сначала используют алгоритм Евклида для вычисления значений N_1 и N_2, таких, что

$$N_1 * M_1 \equiv 1 \ (\mathrm{mod}\ m_1) \quad \text{и} \quad N_2 * M_2 \equiv 1 \ (\mathrm{mod}\ m_2).$$

Здесь $M_1 = \dfrac{M}{m_1} = \dfrac{m_1 * m_2}{m_1} = m_2; \quad M_2 = \dfrac{M}{m_2} = m_1.$

Затем вычисляют значение

$$x = (a_1 * N_1 * M_1 + a_2 * N_2 * M_2) (\mathrm{mod}\ M).$$

Алгоритм нахождения решения системы сравнений, использующий Китайскую теорему об остатках

```
{возврат x∈[0, M – 1], такого что
        x mod mi = ai(1≤ i ≤ t)}
begin
      for i : =1 to t do
        Ni : = inv(Mi mod mi, mi);
      x : = 0;
      for i : =1 to t do
        x : = [x + Mi * Ni * ai] mod n;
      crt : = x {crt – результат}
end
```

Пример. Решить систему из двух сравнений

$$x \equiv 1 (\mathrm{mod}\ 5),$$
$$x \equiv 10 (\mathrm{mod}\ 11)$$

и нвйти *общее решение* x по модулю 55. Здесь $m_1=5$; $m_1=11$; $M=m_1*m_2=5*11=55$; $a_1=1$; $a_2=10$; $M_1=M/m_1=m_2=11$; $M_2=M/m_2=m_1=5$.

Найдем значения N_1 и N_2, обратные к M_1 и M_2 соответственно по mod m_1 и mod m_2:

$$M_1 * N_1 \equiv 1 (\mathrm{mod}\ m_1), \quad 11 * N_1 \equiv 1 (\mathrm{mod}\ 5) \Rightarrow N_1 = 1,$$
$$M_2 * N_2 \equiv 1 (\mathrm{mod}\ m_2), \quad 5 * N_2 \equiv 1 (\mathrm{mod}\ 11) \Rightarrow N_2 = 9.$$

Вычисляем общее значение

$$x = (a_1M_1N_1 + a_2M_2N_2)(\mathrm{mod}\ N) = (1*11*1+10*5*9)(\mathrm{mod}\ 55) =$$
$$= (11 + 450)(\mathrm{mod}\ 55) = 461(\mathrm{mod}\ 55) = 21(\mathrm{mod}\ 55).$$

Итак, $x=21(\mathrm{mod}\ 55)$.

Квадратичные вычеты

Рассмотрим некоторое простое $p>2$ и число $a<p$. Если число a сравнимо с квадратом некоторого числа x по модулю p, т.е. выполняется сравнение $x^2 \equiv a(\mathrm{mod}\ p)$, тогда a называют *квадратичным вычетом* по модулю p. В противном случае a называют *квадратичным невычетом* по модулю p.

Если a – квадратичный вычет, сравнение $x^2 \equiv a(\mathrm{mod}\ p)$ имеет два решения: +x и –x, т.е. a имеет два квадратных корня по модулю p.

Все квадратичные вычеты находят возведением в квадрат элементов $1, 2, 3, \ldots, (p-1)/2$.

Не все значения $a<p$ являются квадратичными вычетами. Например, при $p=7$ квадратичные вычеты это 1, 2. 4:

$$1^2 = 1 \equiv 1 (\mathrm{mod}\ 7),$$
$$2^2 = 4 \equiv 4 (\mathrm{mod}\ 7),$$
$$3^2 = 9 \equiv 2 (\mathrm{mod}\ 7),$$
$$4^2 = 16 \equiv 2 (\mathrm{mod}\ 7),$$
$$5^2 = 25 \equiv 4 (\mathrm{mod}\ 7),$$
$$6^2 = 36 \equiv 1 (\mathrm{mod}\ 7).$$

Заметим, что каждый квадратичный вычет появляется в этом списке дважды. Не существует никаких значений x, которые удовлетворяли бы любому из следующих уравнений:

$$x^2 \equiv 3 (\mathrm{mod}\ 7),$$
$$x^2 \equiv 5 (\mathrm{mod}\ 7),$$
$$x^2 \equiv 6 (\mathrm{mod}\ 7).$$

Числа 3, 5 и 6 – квадратичные невычеты по модулю 7. Можно доказать, что существует точно (p–1)/2 квадратичных вычетов по модулю p и (p–1)/2 квадратичных невычетов по модулю p.

Если a – квадратичный вычет по модулю p, то a имеет точно два квадратных корня: один корень между 0 и (p–1)/2, другой корень между (p–1)/2 и (p–1).

Один из этих квадратных корней также является квадратичным вычетом по модулю p; он называется *главным квадратным корнем.*

Вычисление квадратных корней при p=7 представлено в табл. П.4.

Таблица П.4

$x^2 \equiv a \pmod 7$	Корни	
	x_1	x_2
$1^2 \equiv 1 \pmod 7$	+1	–1 = –1 + 7 = 6
$2^2 \equiv 4 \pmod 7$	+2	–2 = –2 + 7 = 5
$3^2 \equiv 2 \pmod 7$	+3	–3 = –3 + 7 = 4

Если n – произведение двух простых p и q, т.е. n=p*q, то существуют точно

$$(p-1)(q-1)/4$$

квадратичных вычетов по модулю n, взаимно простых с n. Например, по модулю 35 (p=5, q=7, n=5*7=35) существуют

$$\frac{(5-7)(7-1)}{4} = \frac{4*6}{4} = 6$$

квадратичных вычетов: 1, 4, 9, 11, 16, 29, взаимно простых с 35.

Вычисления в конечных полях

Поле F есть множество, на котором определены операции сложения и умножения, удовлетворяющие требованиям: ассоциативности, коммутативности, дистрибутивности, существования аддитивного 0 и мультипликативной 1, аддитивных обратных и мультипликативных обратных для всех элементов за исключением 0.

Конечное поле F(p) с конечным числом p элементов играет важную роль в криптографии. В общем случае число элементов

$$p = q^n,$$

где q – некоторое простое число и n≥1. Такие конечные поля называют *полями Галуа* и обозначают $GF(q^n)$ или GF(q) при n=1. (Эварист Галуа – французский математик начала XIX века.) Многие криптосистемы базируются на полях Галуа GF(q), где q – большое простое число.

Пример. Поле Галуа GF(5) имеет элементы 0, 1, 2, 3, 4 и описывается таблицами сложения и умножения (табл. П.5):

Таблица П.5

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

x	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

Если q – простое число, то число $a \in [1, q-1]$ является взаимно простым с q, и поэтому обратный элемент a^{-1} имеет единственное значение. Тем самым однозначно определяется операция деления.

Обозначим через GF*(q) множество всех ненулевых элементов поля GF(q). Некоторый элемент g из GF*(q) называют *образующим* или *порождающим элементом* GF*(q), если для всех a из GF*(q) найдется такое целое x, что $g^x = a \bmod q$. Всего имеется $\varphi(q-1)$ образующих элементов g. Число x называют дискретным логарифмом элемента a по основанию g и модулю q. Вычисление дискретных логарифмов (когда заданы g, a и q) примерно такая же труднорешаемая задача, как и разложение на множители.

Еще один тип поля Галуа, используемый в криптографии, основывается на арифметике по модулю неприводимых многочленов степени n, чьи коэффициенты – целые числа по модулю q, где q – простое. Эти поля Галуа обозначают как $GF(q^n)$. Они имеют элементы, которые описываются многочленами степени не выше (n–1) в форме

$$a(x) = a_{n-1}X^{n-1} + \ldots + a_1X + a_0.$$

Каждый элемент a(X) является вычетом по модулю p(X), где p(X) – *неприводимый* многочлен степени n (т.е. p(X) нельзя разложить на сомножители – многочлены степени меньше n).

Арифметические действия над коэффициентами a_i выполняются по модулю q, а наивысшая степень X равна (n–1), так как выполняется приведение по модулю многочлена p(X), имеющего старшую степень n.

Особый интерес представляют поля $GF(2^n)$. Здесь коэффициентами a_i являются 0 и 1. Поэтому многочлен a(X) степени не выше (n–1) можно представить как вектор из n двоичных цифр:

$$a_{n-1}a_{n-2} \ldots a_1a_0.$$

Каждый из n-битовых векторов соответствует конкретному элементу поля $GF(2^n)$.

Например, поле Галуа $GF(2^3)$ имеет элементы:

Многочлены	Двоичная форма
0	000
1	001
x	010
x + 1	011
x^2	100
$x^2 + 1$	101
$x^2 + x$	110
$x^2 + x + 1$	111

Организация вычислений в полях Галуа предполагает знание некоторых свойств многочленов и их корней в двоичном поле GF(2). Кратко приведем некоторые из них:

Свойство 1. Ненулевые элементы поля $GF(2^n)$ являются корнями обобщенного многочлена $X^{2^n-1}+1$.

Свойство 2. Каждый многочлен p(X) степени n, неприводимый над полем GF(2), является делителем двучлена $X^{2^n-1}+1$, и каждый делитель двучлена $X^{2^n-1}+1$, неприводимый над полем GF(2), имеет степень, равную n и менее.

Свойство 3. Все элементы поля $GF(2^n)$ можно получить как совокупность остатков от деления 100...00 на неприводимый многочлен p(X), входящий в разложение двучлена $(X^{2^n-1}+1)$. Эти остатки – корни двучлена $(X^{2^n-1}+1)$, т.е. обращают его в нуль. Число остатков равно (2^n-1).

Свойство 4. В поле $GF(2^n)$ существует примитивный элемент α, такой, что каждый ненулевой элемент поля $GF(2^n)$ может быть представлен как некоторая степень α, т.е. мультипликативная группа $GF(2^n)$ является циклической.

Пример. Определение элементов α_i поля $GF(2^4)$. Согласно свойству 1 ненулевые элементы поля $GF(2^4)$ являются корнями обобщенного двучлена $(X^{2^4-1}+1)=(X^{15}+1)$. Двучлен $(X^{15}+1)$ можно представить в виде произведения неприводимых многочленов – сомножителей:

$$(X^{15}+1) = P(X^1) * P(X^2) * P_1(X^4) * P_2(X^4) * P_3(X^4),$$

где

$$P(X^1) = (X+1), \quad P(X^2) = X^2 + X + 1,$$
$$P_1(X^4) = X^4 + X + 1, \quad P_2(X^4) = X^4 + X^3 + 1,$$
$$P_3(X^4) = X^4 + X^3 + X^2 + X + 1.$$

В соответствии со свойством 3 вычислим элементы α_i поля $GF(2^4)$ как совокупность остатков от деления 100...00 на неприводимый многочлен $P_1(X^4)=X^4+X+1$.

Процедура определения остатков

Делят на $P_1(X^4)=X^4+X+1 \leftrightarrow 10011$ единицу с возрастающим числом нулей, т.е. делят одночлены X^j, где $j=0,1,2,3,\ldots$, на многочлен (X^4+X+1). Степени одночленов X^0, X^1, X^2, X^3 меньше степени многочлена $P_1(X^4)$, поэтому первые четыре остатка от деления на $P_1(X^4)$ равны делимым, т.е. одночленам X^0, X^1, X^2, X^3. Для одночлена $X^4 \leftrightarrow 10000$ получаем остаток

$$\oplus\begin{array}{l}10000\\10011\\\hline 0011\end{array}\Big|\begin{array}{l}10011\\\hline\end{array}\leftrightarrow X^4.$$

Для одночлена $X^5 \leftrightarrow 100000$ получаем остаток

$$\oplus\begin{array}{l}100000\\10011\\\hline 0110\end{array}\Big|\begin{array}{l}10011\\\hline\end{array}\leftrightarrow X^5.$$

Схема вычисления остатков:

$$\oplus\begin{array}{l}10000000000000000\\10011\\\hline\end{array}\Big|\begin{array}{l}10011\\\hline\end{array}$$

$$X^4 \leftrightarrow \oplus\begin{array}{l}0011000\\10011\\\hline\end{array}$$

$$X^7 \leftrightarrow \oplus\begin{array}{l}10110\\10011\\\hline\end{array}$$

$$X^8 \leftrightarrow \oplus\begin{array}{l}01010 0\\10011\\\hline\end{array}$$

$$X^{10} \leftrightarrow \oplus\begin{array}{l}011100\\10011\\\hline\end{array}$$

$$X^{12} \leftrightarrow \oplus\begin{array}{l}11110\\10011\\\hline\end{array}$$

$$X^{13} \leftrightarrow \oplus\begin{array}{l}11010\\10011\\\hline\end{array}$$

$$X^{14} \leftrightarrow \oplus\begin{array}{l}10010\\10011\\\hline\end{array}$$

$$X^{15} \leftrightarrow 0001$$

$(= X^0)$.

Вычисленные остатки и нулевые элементы α_0–α_{14} поля Галуа $GF(2^4)$ сведены в табл. П. 6.

Таблица П.6

X^i	Остаток	α_i
X^0	0 0 0 1	α_0
X^1	0 0 1 0	α_1
X^2	0 1 0 0	α_2
X^3	1 0 0 0	α_3
X^4	0 0 1 1	α_4
X^5	0 1 1 0	α_5
X^6	1 1 0 0	α_6
X^7	1 0 1 1	α_7
X^8	0 1 0 1	α_6
X^9	1 0 1 0	α_9
X^{10}	0 1 1 1	α_{10}
X^{11}	1 1 1 0	α_{11}
X^{12}	1 1 1 1	α_{12}
X^{13}	1 1 0 1	α_{13}
X^{14}	1 0 0 1	α_{14}

Поле Галуа $GF(2^4)$ построено как поле многочленов с коэффициентами 0 и 1 по модулю неприводимого многочлена:

$$P(X^4) = X^4 + X + 1 \leftrightarrow 10011.$$

В поле Галуа $GF(2^n)$ определены четыре алгебраические операции. Операции сложения и вычитания выполняются как операции поразрядного сложения по модулю 2; операция умножения

элементов поля выполняется как умножение соответствующих многочленов с приведением по модулю неприводимого многочлена P(X), т.е. многочлена, по модулю которого построены элементы поля $GF(2^n)$.

Пример. $\alpha_5=0110$, $\alpha_6=1100$, $\alpha_5+\alpha_6=1010$, так как

$$\begin{array}{r} 0\,1\,1\,0 \\ \oplus\ 1\,1\,0\,0 \\ \hline 1\,0\,1\,0. \end{array}$$

Пример. $\alpha_{14}=1001$,

$$\alpha_{14}\cdot\alpha_{14}=\alpha^2_{14}=\alpha_{13} \quad \text{по} \quad \text{mod } P_1(X^4) \leftrightarrow 1\,0\,0\,1\,1.$$

$$\begin{array}{r} 1\,0\,0\,1 \\ 1\,0\,0\,1 \\ \hline 1\,0\,0\,1 \\ \oplus\ 1\,0\,0\,1 \\ \hline 1\,0\,0\,0\,0\,0\,1. \end{array} \qquad \begin{array}{r} 1\,0\,0\,0\,0\,0\,1 \mid 1\,0\,0\,1\,1 \\ \oplus\ 1\,0\,0\,1\,1 \mid \\ \hline \end{array}$$

$$\alpha_{13} \leftrightarrow \quad 1\,1\,0\,1.$$

Чтобы выполнить деление элемента b на элемент a в поле $GF(2^n)$ по модулю P(X), сначала находят обратный элемент $a^{-1}(\text{mod } P(X))$, а затем вычисляют

$$b * a^{-1}(\text{mod } P(X)).$$

Каждый двоичный вектор длиной n, исключая 0, является взаимно простым с неприводимым многочленом P(X) независимо от значения P(X). Поэтому число вычетов, взаимно простых с P(X), равно $\varphi(P(X))=2^n-1$ (расширение функции Эйлера для многочленов). Поэтому

$$a^{-1} = a^{\varphi(P(X))-1} \bmod P(X) = a^{2^n-2} \bmod P(X).$$

Пример. Пусть a=100 и P(X)=1011 в поле $GF(2^3)$.

$$a^{-1}=100^{2^3-2}\ (\text{mod } 1011)=100^6\ (\text{mod } 1011)=100^2 * 100^4\ (\text{mod } 1011).$$

$$100^2\ (\text{mod } 1011)=10000 \oplus 10110 = 110$$

или

$$\begin{array}{r} 1\,0\,0\,0\,0 \mid 1\,0\,1\,1 \\ \oplus\ 1\,0\,1\,1 \mid \\ \hline 1\,1\,0 \end{array}$$

$$100^4\ (\text{mod } 1011)=110^2\ (\text{mod } 1011) = 010$$

или

$$\begin{array}{r} 1\,1\,0 \\ 1\,1\,0 \\ \hline 0\,0\,0 \\ \oplus\ 1\,1\,0 \\ \hline 1\,0\,0 \\ \hline 1\,0\,1\,0\,0 \end{array} \qquad \oplus \qquad \begin{array}{r} 1\,0\,1\,0\,0 \mid 1\,0\,1\,1 \\ 1\,0\,1\,1 \mid \\ \hline 0\,1\,0 \end{array}$$

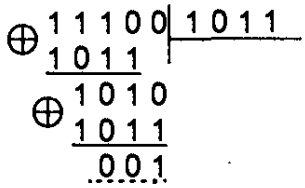
$$100^2 * 100^4\ (\text{mod } 1011) = 110 * 010\ (\text{mod } 1011) = 1100\ (\text{mod } 1011) = 111$$

или

$$\begin{array}{r} 1\,1\,0\,0 \mid 1\,0\,1\,1 \\ \oplus\ 1\,0\,1\,1 \mid \\ \hline 1\,1\,1 \end{array}$$

Итак, $a^{-1}=111$.

Проверка: a=100, $a^{-1}=111$, P(X)=1011. $a * a^{-1}=110 * 100 = 11100$.



т.е. $a * a^{-1} (\text{mod } 1011) = 1$.

Достоинства вычислений в поле $GF(2^n)$:

1. Все элементы поля Галуа имеют конечный размер, деление элементов не имеет каких-либо ошибок округления.

2. Сложение и вычитание элементов поля $GF(2^n)$ не требует деления на модуль.

3. Алгоритмы вычислений в поле $GF(2^n)$ допускают параллельную реализацию.

4. Для поля $GF(2^n)$ обычно применяют в качестве модуля трехчлен $P(X^n) = X^n + X + 1$.

Длинная строка нулей между коэффициентами при X^n и X обеспечивает более простую реализацию быстрого умножения (с приведением по модулю). Трехчлен $P(X^n)$ должен быть неприводимым и примитивным.

Трехчлен $P(X^n)=X^n+X+1$ является примитивным для следующих значений n ($n<1000$):

1, 3, 4, 6, 9, 15, 22, 28, 30, 46, 60, 63,

127, 153, 172, 303, 471, 532, 865, 900.

Список литературы

1. **Аксен Б.А.** Электронные системы расчетов в Internet: от реальной витрины к виртуальной // Конфидент.–1996.–№ 6.–С.43–48.
2. **Александрова Н., Пузырин В.** Системы защиты корпоративных сетей и аутентификации пользователей при помощи смарт-карт // Конфидент.–1998.–№ 4.–С.30–32.
3. **Андерсон Р.** UEPS – электронный бумажник второго поколения // Конфидент.–1997.–№ 1.–С.49–53.
4. **Андерсон Р., Нидхэм Р., Шамир А.** Стеганографическая система файлов // Конфидент.–1999.–№ 4–5.–С.93–99.
5. **Андрианов В.В., Калинский В.Г., Сапегин Л.Н.** Защита авторства, безотказности и целостности электронных документов // Конфидент.–1997.–№ 1.–С.80–84.
6. **Анин Б.** О шифровании и дешифровании // Конфидент.–1997.–№ 1.–С.71–79.
7. **Аснис И.Л., Федоренко С.В., Шабунов К.Б.** Краткий обзор криптосистем с открытым ключом // Защита информации.–1994.–№ 2.–С.35–43.
8. **Балакирский В.Б.** Безопасность электронных платежей // Конфидент.–1996.–№ 5.–С.47–53.
9. **Балакирский В.Б., Коржик В.И., Кушнир Д.В.** Принципы квантовой криптографии // Защита информации.–1995.–№ 3.–С.43–51.
10. **Биометрическая** аутентификация: Обзор // Защита информации.–1994.–№ 2.–С.29–33.
11. **Биркгоф Г., Барти Т.** Современная прикладная алгебра: Пер. с англ.–М.: Мир, 1976.–400 с.
12. **Бияшев Р.Г., Диев С.И., Размахнин М.К.** Основные направления развития и совершенствования криптографического закрытия информации // Зарубежная радиоэлектроника.–1989.–№ 12.–С.76–91.
13. **Борнин Д.Ю., Курочкин А.А., Мартынов А.П., Фомченко В.Н., Снанков В.А.** Метод защиты информации на гибких магнитных дисках от несанкционированного копирования // Конфидент.–1999.–№ 3.–С.92–93.
14. **Брикелл Э.Ф., Одлижко Э.М.** Криптоанализ: Обзор новейших результатов // ТИИЭР.–1988.–Т.76, № 5.–С.75–93.
15. **Буров В.** Данные под замком // Конфидент.–1999.–№ 4–5.–С.28–29.
16. **Вайнер П.С.** Применение машин с ассоциативным поиском для шифрования и атаки на DES // Конфидент.–1996.–№ 6.–С.80–85.

17. **Вайнштейн В.** Ведение личных финансов, покупки и управление банковским счетом через Internet. CIT Forum, 1997.
18. **Виноградов И.М.** Основы теории чисел. – М.: Наука, 1981.
19. **Водолазкий В.** Коммерческие системы шифрования: основные алгоритмы и их реализация // Монитор. – 1992. – № 6–7. – С. 14–19.
20. **Вэк Дж., Карнахан Л.** Содержание сети вашей организации в безопасности при работе с Интернетом (Введение в межсетевые экраны (брандмауэры)). Специальная публикация NIST 800–10. 1997. http://www.parkline.ru/Library/koi/SECURITY/kvn/800–10.txt.
21. **Гайкович В.** Компьютерная безопасность: заметки о текущем состоянии дел // Банковские технологии. – 1997. – Июнь. – С. 56–58.
22. **Гайкович В., Першин А.** Безопасность электронных банковских систем. – М.: Единая Европа, 1994. – 363 с.
23. **Галатенко В.А., Трифаленков И.А.** Комплексные межсетевые экраны обеспечивают безопасность систем intranet // Конфидент. – 1997. – № 2. – С. 29–34.
24. **Герасименко В.А.** Защита информации в автоматизированных системах обработки данных: развитие, итоги, перспективы // Зарубежная радиоэлектроника. – 1993. – № 3. – С. 3–21.
25. **Груздев С.Л., Раевский А.В.** Смарт-карты и персональные компьютеры // Банки и технологии. – 1997. – № 4. – С. 53-59.
26. **Груздев С.Л., Раевский А.В.** Электронная защита программ и данных // Системы безопасности связи и телекоммуникаций. – 1997. – № 3. – С. 52–54.
27. **Дергалин Н.Л.** Практика применения паролей // Защита информации. – 1995. – № 3. – С. 25–27.
28. **Диффи У.** Первые десять лет криптографии с открытым ключом // ТИИЭР. – 1988. – Т. 76, № 5. – С. 54–74.
29. **Диффи У., Хеллман М.Э.** Защищенность и имитостойкость: Введение в криптографию // ТИИЭР. – 1979. – Т. 67, № 3. – С. 71–109.
30. **Дшхунян В., Матюхин В., Наумов Ф. и др.** Российская интеллектуальная карта // Банки и технологии. – 1997. – № 4. – С. 60–61.
31. **Ефремов П.** Смарт-технологии в Интернете – ближайшая перспектива // Банковские технологии. – 1997. – Июнь. – С. 108–109.
32. **Жельников В.** Криптография от папируса до компьютера. – М.: ABF, 1997. – 336 с.
33. **Завалеев В.** Пластиковая карточка как платежный инструмент. Центр Информационных Технологий. Документ http://www.citforum.ru/marketing/articles/art8.shtml.
34. **Зайцева А.И.** Новая атака на DES // Конфидент. – 1996. – № 6. – С. 86–87.
35. **Защита информации в персональных ЭВМ** /А.В. Спесивцев, В.А. Вегнер, А.Ю. Крутяков и др. – М.: Радио и связь, МП "Веста", 1993. – 192 с.

36. **Защита программного обеспечения**: Пер. с англ. / Д. Гроувер, Р. Сатер, Дж. Фипс и др. / Под ред. Д. Гроувера.– М.: Мир, 1992.– 286 с.
37. **Зегжда Д.П., Ивашко А.М.** Как построить защищенную информационную систему: Под ред. Д.П. Зегжды и В.В. Платонова – СПб: Мир и семья – 95, 1997.– 312 с.
38. **Зубанов Ф.** Windows NT – броня крепка // Конфидент.– 1996.– № 2.– С. 31–38.
39. **Иванов И.Г., Кузнецов П.А., Попов В.И.** Методические основы защиты информации в банковских автоматизированных комплексах // Защита информации.– 1994.– № 1.– С.13–24.
40. **ГОСТ Р 34.10-94. Информационная технология**. Криптографическая защита информации. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма.
41. **ГОСТ Р 34.11-94. Информационная технология**. Криптографическая защита информации. Функция хэширования.
42. **Ипполитова К.В.** SET как двигатель электронной торговли // Конфидент.– 1996.– № 6.– С.66–69.
43. **Казарин О.В., Ухлинов Л.М.** Новые схемы цифровой подписи на основе отечественного стандарта // Защита информации.– 1995.– № 3.– С. 52–55.
44. **Ключевский Б.** Специальные криптографические протоколы // Конфидент.– 1999.– № 1–2.– С.71–79.
45. **Кнут Д.** Искусство программирования для ЭВМ: В 3-х т. Получисленные алгоритмы. Пер. с англ.– М.: Мир, 1977.–Т. 2.–724 с.
46. **Ковалерчик И.** Брандмауэры, или запирайте вашу дверь: Обзор // Сети.– 1997.– № 2.– С. 88–99.
47. **Кузьмич В.М.** Возможности злоумышленников по "взлому" систем защиты на персональных компьютерах // Защита информации.– 1995.– № 3.– С. 27–39.
48. **Лебедев А.** Нужны ли "шифровальные средства" // Банковские технологии.– 1997.– Январь.– С. 60–66.
49. **Лебедев А.** Платежные карточки. Новые возможности, проблемы и тенденции // Банки и технологии.– 1997.– № 6.– С. 36–41.
50. **Левин Е.М.** Электронные ключи как зеркало рынка программного обеспечения // Защита информации.– 1994.– № 1.– С.72–76.
51. **Левин Е.М.** Распространение и защита программ в Internet // Конфидент.– 1998.– № 5.– С. 30–33.
52. **Логинов А.А., Елхимов Н.С.** Общие принципы функционирования электронных платежных систем и осуществление мер безопасности при защите от злоупотребления и мошенничества // Конфидент.– 1995.– № 4.– С. 48–54.

53. **Лунин А.В., Сальников А.А.** Перспективы развития и использования асимметричных криптоалгоритмов в криптографии // Конфидент. – 1998. – № 6. – С. 15–23.
54. **Льюис К.** Как защитить сеть от "взлома"? // Сети и системы связи. – 1998. – № 2(24). – С. 136–141.
55. **Мафтик С.** Механизмы защиты в сетях ЭВМ: Пер. с англ. – М.: Мир, 1993. – 216 с.
56. **Медведовский И.Д., Семьянов П.В., Платонов В.В.** Атака через Internet. Под ред. П.Д. Зегжды – СПб.: Мир и семья – 95, 1997. – 296 с.
57. **Мельников В.В.** Защита информации в компьютерных системах. – М.: Финансы и статистика; Электроинформ, 1997. – 367 с.
58. **Мельников Ю.Н.** Электронная цифровая подпись. Возможности защиты // Конфидент. – 1995. – № 6. – С. 35–47.
59. **Месси Дж. Л.** Введение в современную криптологию // ТИИЭР. – 1988. – Т.76, № 5. – С. 24–42.
60. **Мещеряков В.А.** Криминалистическая классификация преступлений в сфере компьютерной информации // Конфидент. – 1999. – № 4–5. – С. 15–21.
61. **Михайлов А.Г.** Новые банковские технологии – пластиковые карты // Защита информации. – 1995. – № 3. – С. 62–68.
62. **Мошонкин А.Г.** Что такое шифрование с открытым ключом? // Защита информации. – 1994. – № 1. – С. 37–41.
63. **Мястковски С.** Найти и обезвредить (антивирусные программы) // Мир ПК. – 1997. – № 4. – С. 43–53.
64. **Никитин А.** Обеспечение защищенного обмена информацией в корпоративных сетях и Internet // Конфидент. – 1998. – № 5. – С. 34–37.
65. **Онучин С.В.** Устройства защиты информации. Критерии выбора // Connect! Мир связи. – 1998. – № 11. – С. 104–107.
66. **Осовецкий Л.** Построение средств межсетевой защиты информации. – НТЦ "Критические Информационные Технологии", 1997.
67. **Отставнов М.Е.** Электронная наличность в сетях Internet // Банковские технологии. – 1996. – Февраль – март. – С. 46–50.
68. **Отставнов М.Е.** От "средств защиты" – к финансовой криптографии // Конфидент. – 1999. – № 6. – С. 81–87.
69. **Питерсон У., Уэлдон Э.** Коды, исправляющие ошибки: Пер. с англ. – М.: Мир, 1976. – 594 с.
70. **Полевой Н.** Смарт-карта секретного доступа // Конфидент. – 1997. – № 5. – С. 91–93.
71. **Попов В.И.** Зарубежные средства канального шифрования // Защита информации. – 1994. – № 2. – С. 23–28.

72. **Правильный выбор криптографических средств**: Обзор современной криптографической техники (по материалам зарубежной печати) // Защита информации.–1994.–№ 1.–С.42–47.
73. **Программно-аппаратные средства обеспечения информационной безопасности**. Защита программ и данных: Учеб. пос. для вузов / П.Ю. Белкин, О.О. Михальский, А.С. Першаков и др.–М.:Радио и связь.–1999.–168 с.
74. **Проскурин Г.В.** Криптография. Методы защиты информации в телекоммуникационных сетях // Connect! Мир связи.–1999.–№ 6.–С.124–126.
75. **Раевский А., Груздев С.** Смарт-карты: завтрашние технологии сегодня! // Конфидент.–1997.–№ 2.–С.79–81.
76. **Райвест Р.Л.** Многоуровневая криптография // Конфидент.–1997.–№ 1.–С.65–70.
77. **Романец Ю.В., Тимофеев П.А., Шаньгин В.Ф.** Защита информации в компьютерных системах и сетях.–М.: Радио и связь.–1999, 328 с.
78. **Рябко С.Д.** Мир TCP/IP. Традиционные приложения // Сети и системы связи.–1996.–№ 4.–С.38–44.
79. **Саломаа А.** Криптография с открытым ключом: Пер. с англ.–М.: Мир, 1996.–304 с.
80. **Симмонс Г.Дж.** Обзор методов аутентификации информации // ТИИЭР.–1988.–Т.76.–№ 5.–С.105–125.
81. **ГОСТ 28147-89. Система обработки информации**. Защита криптографическая. Алгоритм криптографического преобразования.
82. **Смид М.Э., Бранстед Д.К.** Стандарт шифрования данных: Прошлое и будущее // ТИИЭР.–1988.–Т.76, № 5.–С.43–53.
83. **Смирнов В.А.** Средства обеспечения безопасности платежных систем на микропроцессорных смарт-картах // Конфидент.–1996.–№ 6.–С.50–51.
84. **Смит Г.С.** Программы шифрования данных // Мир ПК.–1997.–№ 3.–С.59–68.
85. **Соколов Ю.В.** Безопасность в платежной системе на основе карточек АС "Сберкарт" // Конфидент.–1997.–№ 1.–С.46–47.
86. **Стенг Д., Мун С.** Секреты безопасности сетей.–Киев: Диалектика, 1995.–544 с.
87. **Сяо Д., Керр Д., Мэдник С.** Защита ЭВМ: Пер. с англ.–М.: Мир, 1982.–264 с.
88. **Тайли Э.** Безопасность персонального компьютера: Пер. с англ.–Мн.: ООО "Попурри", 1997.–480 с.
89. **Теория и практика обеспечения информационной безопасности**. Под ред. П.Д. Зегжды.–М.: Изд-во Агентства "Яхтсмен", 1996.–192 с. Серия "Защита информации".

90. **Тимофеев П.А.** Принципы защиты информации в компьютерных системах // Конфидент. – 1998. – № 3. – С.72–76.
91. **Тимофеев П.А.** Защита информации от несанкционированного доступа в современных компьютерных системах // Конфидент. – 1998. – № 5. – С. 55–59.
92. **Турский А., Панов С.** Защита информации при взаимодействии корпоративных сетей в Internet // Конфидент. – 1998. – № 5. – С. 38–43.
93. **Фоменков Г.В.** Криптокарта Fortezza – правительственные технологии в коммерческих приложениях (обзор по материалам открытой печати) // Конфидент. – 1997. – № 1. – С. 23–29.
94. **Хоффман Л.** Современные методы защиты информации: Пер. с англ. / Под ред. В.А. Герасименко. – М.: Радио и связь, 1980. – 264 с.
95. **Хэйт Т.** Размышления об электронных платежах // Сети и системы связи. – 1996. – № 4. – С. 118–121.
96. **Чмора А.Л.** Практическая криптостойкость RSA: оценки и прогнозы // Мир связи. – 1997. – № 10. – С. 56–61.
97. **Чмора А.Л.** Криптосистема с депонированием ключа // Connect! – 1997. – № 3. – С. 34–39.
98. **Чмора А.Л.** Безопасность в W^3 // Конфидент. – 1996. – №4. – С.29–37.
99. **Шаньгин В.Ф.** Защита информации и информационная безопасность. Часть I. Основы информационной безопасности. Симметричные криптосистемы: Учеб. пос. для вузов. – М.: МИЭТ. – 1999. – 140 с.
100. **Шаньгин В.Ф.** Защита информации и информационная безопасность. Часть II. Асимметричные криптосистемы. Идентификация, аутентификация, электронная цифровая подпись и управление ключами: Учеб. пос. для вузов. – М.: МИЭТ. – 2000. – 124 с.
101. **Шеннон К.Э.** Теория связи в секретных системах. В кн. К.Э. Шеннона. "Работы по теории информации и кибернетике". – М.: ИЛ, 1963. – С. 243–332.
102. **Akl S.G.** Digital Signatures: A Tutorial Survey // Computer. – Feb. 1983. – P. 15–24.
103. **D'Angelo D.M., McNair B., Wilkes J.E.** Security in Electronic Massaging Systems // AT&T Technical Journal. – May/June 1994. – P. 7–13.
104. **Beckett B.** Introduction to Cryptology and PC Security. – The McGraw-Hill Companies, 1997. – 356 p.
105. **Boyar J., Chaum D., Damgard I.** Convertible Undeniable Signature // Advances in Cryptology – CRYPTO'90 Proceedings. Springer-Verlag. – 1991. – P. 189–205.
106. **Chapman D.B., Zwicky E.D.** Building Internet Firewalls. – O'Reily & Associates, Inc., 1995. – 517 p.

107. **Chaum D., van Antwerpen H.** Undeniable Signatures // Advances in Cryptology – CRYPTO'89 Proceedings. – Springer-Verlag. – 1990. – P. 212–216.
108. **Chaum D.** Blind Signature Systems // U.S.Patent # 4.759063, 19 Jul 1998.
109. **Hellman M.E.** The Mathematics of Public-Key Cryptography // Scientific American. – 1979. – № 8. – P. 146–157.
110. **Kahn D.** The Codebreakers: The Story of Secret Writing. – New York: Macmillan Publishing Co., 1983.
111. **Kent S.T.** Internet Privacy Enhanced Mail // Communications of the ACM. – 1993. – Vol. 36, № 8. – P. 48–60.
112. **Kent S.T.** Internet Security Standards: Past, Present and Future // StandardView. – 1994. – Vol. 2, № 2. – P. 78–85.
113. **Konheim A.G.** Cryptography. A Primer. – John Wiley & Sons, Inc., 1981. – 432 p.
114. **Krajewski M., Chipchak J.C., Chodorow D.A.,Trostle J.T.** Applicability of Smart Cards to Network User Authentication // Computing Systems. – Winter 1994. – P. 75–89.
115. **Lampson B., Abadi M., Burrows M., Wobber E.** Authentication in Distributed Systems: Theory and Practice // ACM Trans. on Computer Systems. – 1992. – Vol, 10. № 4. – P. 265–310.
116. **Massey J.L.** Feedback Shift Register Synthesis and BCH Decoding // IEEE Trans.Inform. Theory. – 1969. – Vol. IT–15. – P. 122–127.
117. **Menezes A.J., van Oorschot P.C., Vanstone S.A.** Handbook of Applied Cryptography. CRC Press, 1999. – 816 p.
118. **Meyer C.H., Matyas S.M.** Crytography: A New Dimension in Computer Data Security. – John Wiley & Sons, 1982. – 755 p.
119. **Neuman B.C.,** Ts'o T. Kerberos: An Authentication Service for Computer Networks // IEEE Comm. Magazine. – 1994. – Vol. 32, № 9. – P. 33–38.
120. **Odlyzko A.M.** Public Key Cryptography // AT&T Technical Journal. – Sept./Oct. 1994. – P. 17–23.
121. **Schneier B.** Applied Cryptography. – John Wiley & Sons, Inc., 1996. – 758 p.
122. **Schneier B.** One-Way Hash Functions // Dr. Dobb's J. – Sept. 1991. – P. 148–151.
123. **Seberry J., Pieprzyk J.** Cryptography. An Introduction to Computer Security. Advances in Computer Science Series. – Prentice Hall of Australia Pty Ltd., 1989. – 375 p.
124. **Sheman S.A., Skibo R., Murray R.S.** Secure Network Access Using Multiple Applications of AT&T's Smart Cards // AT&T Technical Journal. – Sept./Oct. 1994. – P. 61–72.
125. **Stallings W.** Practical Cryptography for Data Internetworks // IEEE Computer Society Press, 1996. – 356 p.
126. **Woo Y.C., Lam S.S.** Authetication for Distributed Systems // Computer. – 1992. – Vol. 25, № 1. – P. 39–51.

Оглавление