МВД России
Санкт-Петербургский университет

И. Н. Васильева, А. И. Локнов, А. И. Примакин

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

Учебное пособие

Санкт-Петербург
2023

План-график 2023, поз. 46

УДК 004.056.55
ББК 32.973
В19

В19 Криптографическая защита информации: учебное пособие / И. Н. Васильева, А. И. Локнов, А. И. Примакин. — Санкт-Петербург: СПбУ МВД России, 2023. — 120 с.

Авторский коллектив:
Васильева И. Н. — гл. 2, 4, закл.; *Локнов А. И.* — введ., гл. 1, 5;
Примакин А. И. — гл. 3.

ISBN 978-5-91837-672-0

Учебное пособие соответствует программе учебной дисциплины «Криптографическая защита информации».

В учебном пособии рассмотрены основные понятия и задачи криптографии, идеально и практически стойкие криптосистемы, блочные и потоковые шифры, симметричные и асимметричные криптосистемы, криптографические хэш-функции, криптографические протоколы аутентификации и распределения криптографических ключей, основы организационно-правового обеспечения криптографической защиты информации; описаны основные понятия криптоанализа.

Предназначено для курсантов и слушателей образовательных организация высшего образования МВД России, обучающихся по специальности 10.05.05 Безопасность информационных технологий в правоохранительной сфере.

УДК 004.056.55
ББК 32.973

Рецензенты:
Бецков А. В., доктор технических наук, доцент
(Академия управления МВД России);
Нестеровский О. И., кандидат технических наук
(Воронежский институт МВД России);
Еськов А. В., доктор технических наук, профессор
(Краснодарский университет МВД России);
Симаков А. А., кандидат технических наук, доцент
(Омская академия МВД России)

ISBN 978-5-91837-672-0

© Санкт-Петербургский университет
МВД России, 2023

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Криптографическая защита информации — один из основных предметов, изучаемых в образовательных организациях системы МВД России, относящийся к дисциплинам (модулям) специализации для специальности 10.05.05 Безопасность информационных технологий в правоохранительной сфере. Для того чтобы органы внутренних дел могли эффективно выполнять свои задачи, сотрудники подразделений должны хорошо знать и правильно понимать вопросы обеспечения защиты информации.

Вопросам изучения криптографических методов защиты информации посвящены исследования многих учёных, таких, как О. С. Авсентьев, В. В. Солдатов, В. Н. Думачев, С. В. Зарубин, Е. А. Кучмасов и др.[1] Данные исследователи внесли значительный вклад в разработку теоретических проблем криптографии. Вместе с тем ряд аспектов нуждается в уточнениях и альтернативном рассмотрении с точки зрения математического описания протекающих процессов.

Настоящее учебное пособие посвящено вопросам криптографической защиты информации в компьютерных системах и сетях передачи данных. В содержании, наряду с освещением исторических аспектов криптографии, отражаются современные криптографические стандарты. В частности, рассмотрены российские стандарты блочного шифрования, проанализированы американские аналоги. Приведены различные виды потокового шифрования, сведения о хэш-функциях и цифровых подписях, рассмотрены соответствующие российские стандарты. Подробно освещены современные подходы к защите информации. В рамках криптографических протоколов рассмотрены вопросы электронных денег (электронной наличности), создание скрытого канала, особенности доказательства с нулевым разглашением информации. Уделено внимание вопросам организационно-правового обеспечения криптографической защиты информации.

Учебное пособие разработано в соответствии с Федеральным государственным образовательным стандартом высшего образования, основными образовательными и рабочими учебными программами по специальности 10.05.05 Безопасность информационных технологий в правоохранительной сфере. Глубина изложения материала соответствует требованиям к знаниям, умениям и навыкам обучающихся, вытекающим из компетенций, представленных в рабочих программах дисциплины «Криптографическая защита информации» данной специальности. Пособие может также быть использовано курсантами направлений подготовки (специальностей), не связанных с обучением специалистов в области информационной безопасности, эксплуатации средств связи, радиоэлектронных средств и радиоэлектронных систем различного назначения, для углубленного изучения криптографии.

[1] Авсентьев О. С., Солдатов В. В., Думачев В. Н., Зарубин С. В., Кучмасов Е. А. Криптографические методы защиты информации: учебное пособие. Воронеж: ВИ МВД России, 2018. 204 с.

Глава 1
ОБЩИЕ СВЕДЕНИЯ О КРИПТОГРАФИЧЕСКИХ МЕТОДАХ ЗАЩИТЫ ИНФОРМАЦИИ

§ 1.1. Основные понятия и задачи криптографии

1.1.1. Основные понятия и определения криптографии

Во все времена для значительной части военной, правоохранительной, экономической, дипломатической и другой информации требовалось обеспечение конфиденциальности. Доступ к такой информации ограничен кругом законных (легальных) пользователей. С другой стороны, должна обеспечиваться защита от незаконных (нелегальных) пользователей (противников, нарушителей, злоумышленников), которые стремятся завладеть этой информацией.

В переводе с греческого языка слово «криптография» означает тайнопись. Способы и методы тайного письма путем преобразования информации издавна называют шифром, а процесс применения шифра к защищаемой информации — шифрованием.

Наука о методах и алгоритмах шифрования называется *криптографией*. Людей, занимающихся криптографией, называют *криптографами*. *Криптоаналитики* являются специалистами в области *криптоанализа* — науки о вскрытии шифров, которая отвечает на вопрос о том, как прочесть открытый текст, скрывающийся под шифрованным. Раздел науки, объединяющий криптографию и криптоанализ, именуется *криптологией*.

Впервые формальное математическое описание криптографии (симметричных криптографических систем) появляется в 1945 году в секретных работах К. Шеннона. В 1949 году эти работы были рассекречены и опубликованы в Bell System Technical Journal в статье «The Communication Theory of Secrecy systems» (Теория связи в секретных системах). В настоящее время криптография представляет собой сложную научную дисциплину, основанную на теории чисел, теории вероятностей, теории групп и ряде других разделов математики и использующую для решения своих задач вычислительные мощности современных компьютерных систем и сетей.

Основные направлениями криптографической защиты информации являются осуществление поиска и исследования методов шифрования информации ограниченного доступа с целью сокрытия ее содержания; методов обмена зашифрованной информацией между законными пользователями; методов расшифрования информации законными пользователями; методов выявления попыток незаконных пользователей получить доступ к этой информации, исказить или имитировать ее.

Шифрование представляет собой преобразование исходного текста в кажущуюся случайной последовательность символов, называемую *криптограммой* или *шифртекстом* (англ. ciphertext). Каждое преобразование однозначно определяется ключом и описывается некоторым криптографическим алгоритмом. Один и тот же криптографический алгоритм может применяться для шифрования в различных режимах. Каждый режим шифрования имеет как свои преимущества, так и недостатки. Поэтому выбор режима шифрования зависит от конкретной ситуации и решаемых практических задач.

Шифртекст становится понятным только после его обратного преобразования — *расшифрования*. Но это преобразование может сделать только тот, кто знает алгоритм расшифрования и ключ. Алгоритм расшифрования в общем случае может отличаться от алгоритма шифрования. Соответственно могут различаться ключи шифрования и расшифрования. Процесс извлечения открытого текста из шифртекста в случае, если ключ (ключ и алгоритм шифрования) неизвестен, принято называть *дешифрованием*.

Согласно К. Шеннону[2], *криптосистема*, *шифр* (рис. 1.1) определяется как семейство взаимно однозначных отображений множества возможных сообщений X во множество криптограмм Y, проиндексированное элементами k из множества ключей: $\{F_k : X \to Y, k \in K\}$. K — пространство ключей — набор всех возможных значений ключа k. $K = \{k\}$.

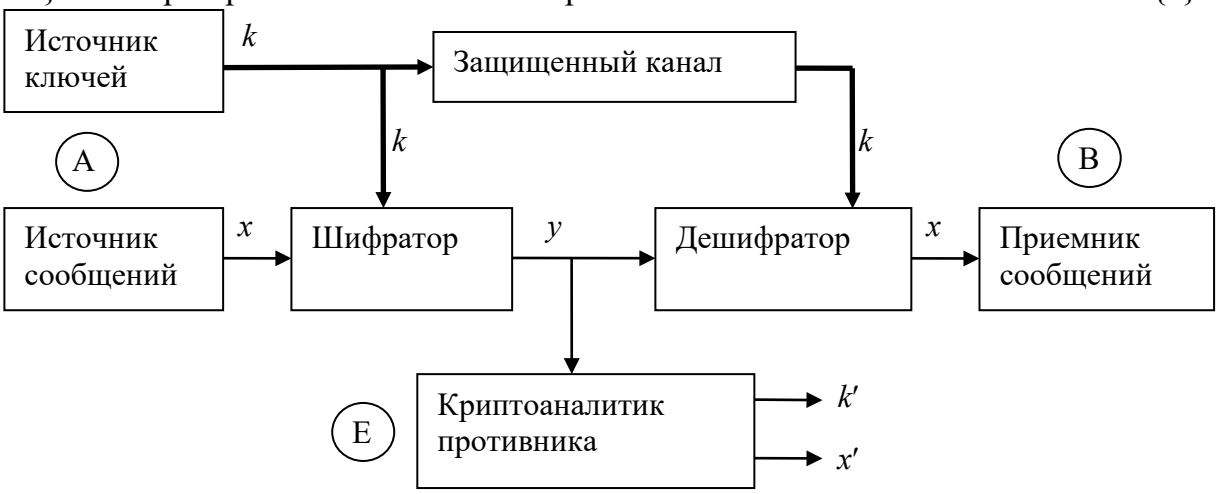

Рис. 1.1. Схема симметричной криптосистемы

Следует отметить, что К. Шенноном рассматривался только один определенный тип криптосистем, а именно симметричные криптосистемы, в которых и шифрование, и расшифрование производится с помощью одного и того же *секретного ключа*. В схеме К. Шеннона на стороне отправителя (А) имеются два источника информации — источник сообщений $\{x\}$ и источник ключей $\{k\}$. Источник ключей выбирает из множества всех возможных ключей один ключ k, который будет использоваться при отправке информации. Ключ передается отправителю и получателю (В) сообщения таким образом, что его невозможно перехватить. Канал передачи ключа является закрытым (защищенным). Этот канал абсолютно надежен и недоступен для посторонних. В то же время основной канал обмена сообщениями является общедоступным. Противник (Е) может перехватывать передаваемые сообщения, то есть канал передачи криптограмм является открытым. Целью противника является определение открытого сообщения, соответствующего переданной криптограмме, или ключа шифрования. Под противником понимается любой субъект, не имеющий права ознакомления с содержанием передаваемой информации. В качестве противника может выступать криптоаналитик, владеющий методами раскрытия шифров.

В общем случае *криптографической системой* назовем совокупность алгоритмов и ключей шифрования/расшифрования, а реализующие их устройства — *шифротехникой*. Математическая модель криптосистемы в наиболее общем виде может быть описана системой двух уравнений:

$$y = f(x, k_{\text{Ш}}), x^* = q(y, k_{\text{Р}}),$$

где: x — шифруемое сообщение, y — результат шифрования (криптограмма), x^* — результат расшифрования, $k_{\text{Ш}}$ — ключ шифрования, $k_{\text{Р}}$ — ключ расшифрования, f, q — функции, определяющие алгоритмы шифрования и расшифрования. Эти функции зависят от ключей шифрования/расшифрования. В симметричных криптосистемах ключи шифрования и расшифрования совпадают.

[2] Шеннон К. Теория связи в секретных системах // Работы по теории информации и кибернетике / пер. с англ. М.: Изд-во иностранной литературы, 1963. С. 333–369.

Так как смена алгоритма шифрования/расшифрования требует гораздо больше времени и затрат, чем смена ключа, высокая стойкость шифра от незаконного расшифрования должна обеспечиваться и в случае, когда злоумышленнику известен алгоритм расшифрования, но неизвестен ключ. Этот принцип известен, как *принцип Керкгоффа*. К. Шеннон несколько позже сформулировал этот принцип, видимо, независимо от О. Керкгоффа, фразой «Враг знает систему». Таким образом, при оценке надежности шифрования необходимо предполагать, что противник знает об используемой системе шифрования все, кроме применяемых ключей.

Соблюдение принципа Керкгоффа позволяет рассматривать шифр как параметризованный алгоритм, состоящий из процедурной части (описания того, какие именно операции и в какой последовательности выполняются над шифруемыми данными) и сменных параметров (изменяемых данных, используемых в преобразованиях, — ключей). При этом, согласно принципу Керкгоффа, *защищенность системы не должна зависеть от секретности долговременных элементов шифра* (то есть таких элементов, которые невозможно было бы быстро изменить в случае утечки секретной информации). *Защищенность криптографической системы определяется только секретностью кратковременных, сменных параметров — ключей*.

Такая организация криптографической системы имеет ряд преимуществ: разглашение шифра (алгоритма и ключа) не приводит к необходимости полной замены реализации всего алгоритма, достаточно заменить только скомпрометированный ключ; ключи можно отчуждать от остальных компонентов системы шифрования — хранить или генерировать отдельно от реализации конкретного алгоритма; появляется возможность для точной оценки «степени неопределенности» алгоритма шифрования — она равна неопределенности используемого ключа.

Под *криптостойкостью* (*стойкостью* алгоритма шифрования или стойкостью шифра) понимается способность криптографического алгоритма противостоять возможным атакам на него.

Существуют криптосистемы, для которых любой объем перехваченной информации недостаточен для того, чтобы найти шифрующее отображение F_k. Шифры такого типа называются *безусловно стойкими*. Доказано, что безусловно стойкие шифры существуют. Однако большинство криптосистем допускает теоретическую возможность определения ключа (или дешифрования сообщения без знания ключа) при достаточном объеме перехваченных криптограмм. Для криптоаналитика, обладающего ограниченными вычислительными ресурсами, вероятность найти это решение за приемлемое время может оказаться чрезвычайно малой. Шифры такого типа называются *вычислительно стойкими*. Их стойкость основана на высокой вычислительной сложности противостоять криптоаналитическим атакам. Вычислительная сложность последних определяется временной и емкостной составляющими (сложность по времени, сложность по данным).

Криптоаналитические атаки могут быть пассивными и активными. *Пассивной* называется атака, при которой противник не имеет возможности изменять передаваемые сообщения. При пассивной атаке возможно лишь прослушивание передаваемых сообщений, их дешифрование и анализ трафика. При *активной* атаке противник имеет возможность добавлять свои сообщения в канал связи и модифицировать сообщения, передаваемые легальными пользователями криптосистемы.

Криптоаналитические атаки можно классифицировать по количеству и типу информации, доступной противнику для криптоанализа. По данной классификации выделяют следующие виды атак.

1. *Криптоатака на основе известного шифртекста.* В распоряжении противника есть криптограммы различных неизвестных открытых текстов, зашифрованные на одном и том же ключе. Задача криптоаналитика состоит в получении открытого текста как можно большего числа сообщений или в получении ключа, использованного при шифровании. Полученный ключ будет затем использован для дешифрования других сообщений.

2. *Криптоатака на основе известного открытого текста*. Противник знает криптограмму и исходный текст, или хотя бы часть исходного текста. Сопоставляя пары «текст — криптограмма», противник пытается узнать секретный ключ, чтобы с его помощью дешифровать все последующие сообщения. На первый взгляд, казалось бы, противнику достаточно сложно заполучить в свое распоряжение некоторое количество пар «текст — шифртекст», однако на практике криптоаналитик может иметь информацию о формате перехваченного зашифрованного файла: например, знать, что это файл с изображением в формате *BMP*, *JPEG*, *TIF*, документ *Word*, *Excel,* файл базы данных *ORACLE* и пр. Все эти и многие другие приложения имеют определенные стандартные заголовки или фрагменты. Проанализировав криптограмму в предположении, что открытый текст имеет тот или иной формат, специфичный для определенного приложения, криптоаналитик сможет сформировать необходимые исходные данные для проведения атаки на основе известного открытого текста.

3. *Криптоатака на основе выбранного открытого текста*. Противник может ввести специально подобранный текст в шифратор и получить криптограмму на основе секретного ключа, который он хочет узнать. Во время второй мировой войны американцы, подкупив охрану, выкрали шифровальную машину в японском посольстве на два дня и имели возможность формировать и подавать ей на вход различные тексты и получать соответствующие шифровки. Таким образом, они узнали ключ. Взламывать саму машину для определения заложенного в нее ключа было нельзя, так как после ее возвращения это было бы замечено и повлекло бы за собой смену ключа.

4. *Криптоатака на основе выбранного шифртекста*. Противник имеет возможность подставить для расшифровки фиктивную криптограмму и получить текст, в который она преобразуется.

Существуют также *адаптивные криптоатаки* на основе выбранных открытых текстов или криптограмм, когда выбор может осуществляться противником неоднократно с учетом результатов предыдущего криптоанализа. Это наиболее мощные, но и наиболее сложные с точки зрения получения исходного материала атаки.

Между криптографией и криптоанализом существует тесная взаимосвязь. Криптография — это защита, разработка шифров, а криптоанализ — нападение, атака на шифры. Не бывает хороших специалистов по криптографии, не владеющих методами криптоанализа. В иностранной литературе для обозначения области знаний, объединяющей криптографию и криптоанализ, часто используется термин криптология.

1.1.2. Основные задачи криптографии

Первоначально под областью применения криптографии понимали исключительно шифрование конфиденциальной информации с целью ее защиты от несанкционированного доступа[3]. В настоящее время такое определение не вполне корректно. По мере своего развития криптография оказалась способной, наряду с обеспечением конфиденциальности, решать ряд новых важных задач обеспечения информационной

[3] Рябко Б. Я., Фионов А. Н. Криптографические методы защиты информации. М.: Горячая линия — Телеком, 2012. 230 с.

безопасности: обеспечение целостности информации, обеспечение аутентификации сообщений и абонентов, обеспечение невозможности отказа сторон от авторства. Вышеперечисленные задачи и являются основными задачами криптографии.

1. *Обеспечение конфиденциальности*. Решение проблемы защиты информации от ознакомления с ее содержанием лицами, не имеющими права доступа к ней. В зависимости от контекста вместо термина «конфиденциальная» информация могут выступать термины «секретная», «частная», «ограниченного доступа» и др. информация. Перечень информации с ограниченным доступом, подлежащей защите, определен в законах о государственной и коммерческой тайне, о персональных данных и пр.

2. *Обеспечение целостности*. Целостность информации — обеспечение неизменности информации незаконными пользователями в процессе ее передачи или хранения. Для гарантии целостности необходим простой и надежный критерий обнаружения любых манипуляций с данными. Манипуляции с данными включают вставку, удаление и подмену. Для обеспечения целостности в информацию вносится избыточность. Как правило, это достигается добавлением к сообщению некоторой проверочной комбинации, вычисляемой с помощью специального алгоритма, входными данными для которого является открытый текст, и играющей роль контрольной суммы для проверки целостности полученного сообщения. В криптографических приложениях используемая контрольная сумма должна быть криптографически сильной, то есть стойкой к возможным атакам противника.

3. *Обеспечение невозможности отказа от авторства* (неотрекаемости). В некоторых ситуациях, например, в силу изменившихся обстоятельств, отдельные лица могут отказаться от ранее принятых обязательств. В связи с этим необходим некоторый механизм, препятствующий подобным попыткам. Основным механизмом решения этой проблемы в криптографии является применение цифровой подписи.

4. *Обеспечение аутентификации источника информации*. Аутентификация — это установление подлинности сторон и самой информации в процессе информационного взаимодействия. Информация, передаваемая по каналу связи, должна быть аутентифицирована по источнику, времени создания, содержанию данных, времени пересылки и т. д. Задачи аутентификации решаются различными методами, в том числе и криптографическими. Основным механизмом решения задач аутентификации в криптографии также является цифровая подпись.

Некоторые авторы к числу основных задач криптографии относят задачу обеспечения неотслеживаемости (анонимности), под которой понимается возможность легального пользователя выполнять законные действия (например, оплату услуг через Интернет) не будучи опознанным. Такие задачи могут решаться в рамках протоколов различных цифровых сервисов, например, электронных денег, с помощью механизмов слепой цифровой подписи.

Кроме вышеперечисленных основных задач защиты информации, с помощью криптографических методов возможно решение ряда прикладных задач, к числу которых относятся электронное голосование, жеребьевка, разделение секрета (распределение секретной информации между несколькими субъектами так, чтобы воспользоваться ей они могли только все вместе), обслуживание банковских карт, хранение и обработка эталонов с системах аутентификации, защищенная передача бухгалтерских и иных отчетов по открытым каналам связи (например, по сети «Интернет»), дистанционное банковское обслуживание предприятий и др.

Сами по себе криптографические алгоритмы, как правило, решают только какую-то одну определенную задачу, однако для получения практического результата

в рамках решения прикладных задач важно определить порядок действий по применению тех или иных криптографических алгоритмов. Такой порядок определяется *криптографическими протоколами*, задающими алгоритмы взаимодействия двух или более абонентов информационной системы с использованием криптографических методов защиты информации. В основе протокола лежит набор правил, регламентирующих использование криптографических преобразований и алгоритмов в информационных процессах.

Участники протокола могут доверять или не доверять друг другу, поэтому криптографические протоколы должны предусматривать защиту не только от внешнего противника, но и от нечестных действий партнеров. Атаки противника при этом могут быть направлены не на вскрытие алгоритма шифрования, а на сам протокол. Поэтому криптографические протоколы являются предметом тщательного анализа со стороны специалистов.

Вопросы для самоконтроля:

1. Что такое криптограмма? Что такое криптоанализ?

2. Что понимается под криптосистемой?

3. В чем заключается основной принцип криптографии?

4. Что понимается под криптостойкостью? Как криптографические системы подразделяются по стойкости?

5. Каковы основные виды криптоаналитических атак?

§ 1.2. Исторические методы шифрования

1.2.1. Развитие методов шифрования

Точная историческая дата и даже исторический период зарождения криптографии неизвестны. Известно только, что криптографии не менее четырех тысяч лет. Слово шифр арабского происхождения (от слова «цифра»). Арабы первыми стали заменять буквы цифрами с целью сокрытия содержания текста. Сведения о шифровании сообщений известны из исторических документов древних цивилизаций: Месопотамии, Египта, Индии, Китая и т. д. Широко применялась криптография в Древней Греции. Например, жрецы хранили свои прорицания в зашифрованном виде. Для шифрования текстов в древности использовались системы перестановки и замены (подстановки). Позднее, в XIX веке, появились шифры гаммирования.

В шифрах *перестановки* для получения зашифрованного текста символы открытого текста переставляются с одних позиций на другие, а в шифрах *замены* — заменяются другими буквами (символами). В шифрах *гаммирования* для получения зашифрованного текста на открытый текст накладывается ключевая гамма. Фактически шифры гаммирования являются частным случаем шифров замены. Операции перестановки, замены и гаммирования используются и в настоящее время, но уже в составе более сложных композиционных шифров. Шифры, в которых как для шифрования сообщений, так и для их расшифровывания используется один и тот же ключ, называются *симметричными*.

Симметричные шифры можно разделить на блочные и потоковые. *Блочные* шифры применимы к некоторому блоку подряд идущих символов сообщения, результат шифрования в этом случае зависит, вообще говоря, от всех символов блока. Например, перестановку символов можно сделать только в пределах некоторого блока текста (понятие перестановки применительно к одиночному символу бессмысленно), поэтому шифры перестановки являются блочными.

В то же время замену символов можно производить для каждого символа по отдельности, в этом случае шифр будет считаться потоковым. *Потоковые* шифры выполняют преобразование каждого символа сообщения независимо от остальных.

1.2.2. Шифры перестановки

Сцитала. Сцитала — одно из первых шифровальных приспособлений. Это приспособление использовалась в Спарте в V–IV вв. до н. э. Сцитала — это жезл цилиндрической формы, на который наматывалась лента пергамента (рис. 1.2). Кроме жезла могли использоваться рукоятки мечей, кинжалов копий и пр. Лента была столь узкой, что поперек нее мог быть записан только один символ. Вдоль оси цилиндра на пергамент, намотанный на сциталу, построчно записывался текст. После записи текста лента сматывалась с жезла и передавалась адресату, который имел точно такую же сциталу. После снятия с цилиндра буквы на ленте пергамента оказывались расположенными вразнобой, и сообщение нельзя было прочесть. Очевидно, что сцитала осуществляла перестановку букв сообщения, и это был один из первых шифров перестановки. Ключом шифра служит диаметр сциталы.

Шифр сциталы является реализацией наиболее простого случая шифрующей таблицы — когда символы записываются в таблицу в одном направлении (например, по строкам), а считываются в другом (например, по столбцам).

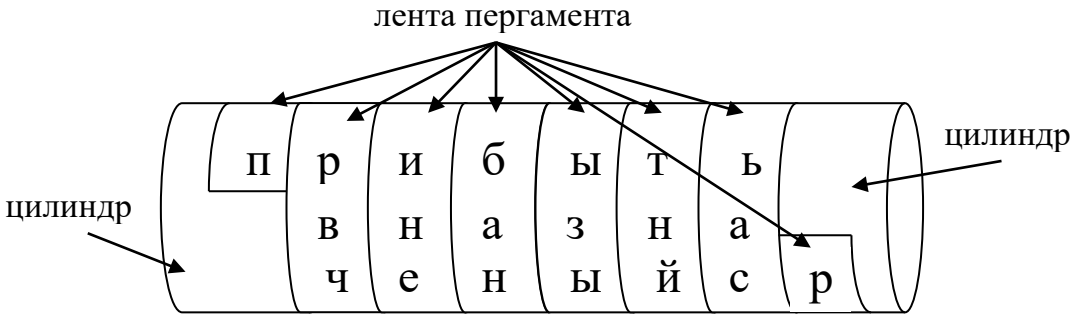

Рис. 1.2. Шифр сциталы

Метод вскрытия этого шифра принадлежит Аристотелю. Аристотель предлагал заточить на конус длинный брус и, обернув вокруг него ленту, начать сдвигать ее по конусу, постепенно сжимая, от большего диаметра до самого малого. В том месте, где диаметр конуса совпадал с диаметром сциталы, начинали просматриваться слоги и слова текста. После этого оставалось лишь замерить диаметр и изготовить цилиндр нужного размера.

Шифрующие таблицы. Одним из самых примитивных табличных шифров перестановки является простая перестановка, для которой ключом служит размер таблицы. Этот метод шифрования в простейшем варианте сходен с шифром сциталы. Например, текст сообщение записывается в таблицу определенного размера в столбик, а считывается по строкам.

Запишем фразу «система защиты информации» в таблицу размером 5×5 по строкам (рис. 1.3). Последние две клетки таблицы заполним произвольным символом, например, буквой «ж». Выписав текст из таблицы по столбцам, получим «смифциатоисзыритаимжещнаж».

Отправитель и получатель сообщения должны заранее условиться о ключе (в данном случае ключ — это размер таблицы). При расшифровании действия выполняют в обратном порядке (построчная запись, чтение по столбцам).

с	и	с	т	е
м	а	з	а	щ
и	т	ы	и	н
ф	о	р	м	а
ц	и	и	ж	ж

Рис. 1.3. Пример шифрующей таблицы

Отправитель и получатель сообщения должны заранее условиться о ключе (в данном случае ключ — это размер таблицы). При расшифровании действия выполняют в обратном порядке (построчная запись, чтение по столбцам).

На практике обычно размер таблицы задается заранее. Пусть таблица содержит n столбцов и m строк. Тогда в нее может вместиться не более $n \times m$ символов. Если длина сообщения $L > n \times m$, то исходный текст делится на несколько блоков, каждый из которых шифруется одинаковым образом.

Для повышения стойкости шифра он может быть несколько усложнен, например, столбцы могут быть переставлены в некоторой последовательности, определяемой ключом. Возможна и двойная перестановка — столбцов и строк.

1.2.3. Шифры простой замены

Шифр простой замены заключается в замене символов исходного сообщения на новые символы по определенному правилу, неизменному на протяжении всего процесса шифрования. Таким образом, для шифров простой замены может быть введено понятие *нормативного* алфавита — алфавита исходных сообщений и *шифралфавита* — алфавита, с помощью символов которого записаны зашифрованные сообщения[4]. В качестве символов шифралфавита могут выступать переставленные символы исходного алфавита, алфавит другого языка, числа или даже пиктограммы. Важно, что в процессе шифрования сообщения заданное соответствие между символами нормативного алфавита и символами шифралфавита остается постоянным (не меняется). Это соответствие и является ключом шифра простой замены.

Квадрат Полибия. Полибий — древнегреческий государственный деятель, полководец и историк, в III веке до н. э. предложил простой код замены, названный позже квадратом (таблицей, шахматной доской) Полибия. Этот шифр использовали древние греки и римляне для передачи сообщений по семафору. Квадрат Полибия пытались использовать находящиеся в тюрьме декабристы для связи с находящимся на воле князем А.И. Одоевским. Попытка не удалась из-за того, что князь не смог запомнить расположение букв в таблице.

Квадрат Полибия представлял собой таблицу, клетки которой заполнялись буквами. Каждая буква открытого сообщения при шифровании заменялась на два индекса: индекс строки и индекс столбца. В качестве индексов строки и столбца могли использоваться буквы (рис. 1.4 а) или цифры (рис. 1.4 б). Так, слово «полиция», зашифрованное с помощью табл. 1.4 б, преобразуется к виду 33, 23, 46, 26, 22, 26, 15.

В исходном варианте символы алфавита заносились в таблицу по порядку (как на рис. 1.4 а, таким образом, квадрат Полибия определял не шифр, а код, принятый для сокращения множества символов алфавита. В этом коде вместо 26 символов

[4] Васильева И. Н., Куватов В. И., Потехин В. С. Криптографическая защита информации: учебное пособие. СПб.: Изд-во СПб ун-та МВД России, 2016. 152 с.

исходного алфавита (символы I и J отождествляются) используются всего 5. Например, символ P будет закодирован парой символов CE.

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

а) для английского языка

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

б) для русского языка

Рис. 1.4. Варианты таблиц «Квадрат Полибия»

Если же буквы алфавита занести в таблицу $n \times m$ в произвольном порядке, как это сделано на рис. 1.4 б, то может быть получено $(n \times m)!$ вариантов расположения букв. В современной терминологии расположение символов в таблице представляет собой ключ шифрования. Этот ключ должен быть известен передающей и принимающей стороне и держаться в секрете от всех остальных. Для таблицы на рис. 1.5 получим $(5 \cdot 6)! = 30! \approx 2{,}6 \cdot 10^{32}$ вариантов ключа, что исключает возможность подбора ключа вручную. Вместе с тем, для шифров простой замены возможно применение методов частотного анализа, что позволяет однозначно дешифровать достаточно длинный текст, не прибегая к полному перебору значений ключа.

Шифр Цезаря. Одним из наиболее известных шифров древнего Рима (I век до н. э.) был шифр Цезаря. В этом шифре каждая буква открытого текста заменяется третьей после нее буквой в алфавите, который считается написанным циклически, т. е. после буквы «Z» — последней буквы латинского алфавита следует буква «A». Результат применения шифра Цезаря к слову RIM имеет вид ULP. Цезарь заменял букву третьей после неё буквой, но можно заменять и какой-либо другой. Главное, чтобы тот, кому посылается шифрованное сообщение, знал величину сдвига. Очевидно, что шифр Цезаря также является шифром простой замены. Ключом здесь является коэффициент сдвига. Поскольку число возможных ключей здесь равно числу букв алфавита минус единица, стойкость такого шифра невелика.

С математической точки зрения шифр Цезаря описывается операциями сложения/вычитания по модулю с величиной фиксированного сдвига S исходного алфавита, при этом считается, что символы алфавита нумеруются числами от 0 до N–1, где N — мощность (число символов) алфавита. Тогда

$y_i = (x_i + S) \bmod N$ — для шифрования,

$x_i = (y_i - S) \bmod N$ — для расшифрования,

где: $\{x_n\}$ — последовательность символов исходного текста в числовой кодировке, $0 \le x_i \le N-1$, $\{y_n\}$ — последовательность символов шифртекста в числовой кодировке, $0 \le y_i \le N-1$, $i = 0,\ldots,n$.

Существуют и другие варианты шифров *простой замены*, задаваемых соответствием между символами алфавита исходного сообщения и алфавита шифртекста. Основная особенность шифров этого типа заключается в том, что это соответствие не изменяется на протяжении всего процесса шифрования.

1.2.4. Частотный криптоанализ простых шифров

В криптоанализе исходят из того, что известен язык сообщения, тип шифра и метод кодирования символов. Обычно эта информация становится известной из агентурных или иных источников. Тем не менее в ряде случаев для того, чтобы определить язык сообщения, тип шифра и метод кодирования символов, необходимо проделать определенную работу.

Криптоанализ учитывает статистические особенности текстов зашифрованных сообщений[5]. Наиболее простыми характеристиками текстов, используемыми в криптоанализе, являются такие характеристики, как повторяемость букв, пар букв (биграмм) и вообще *n*-грамм, сочетаемость букв друг с другом, чередование гласных и согласных и др. Большую помощь в криптоанализе имеют слова или выражения, появление которых можно ожидать в перехваченном сообщении (например, для английского текста это «the», «and» и др., для русского — «что», «как» и др.).

Важнейшим свойством шифров *простой замены* является то, частоты появления отдельных символов и *k*-буквенных сочетаний символов в исходном сообщении и в его криптограмме совпадают (это называется наследованием статистических характеристик открытых текстов шифрованными текстами). Например, в английском языке наиболее часто встречающейся буквой является «e», затем идут «t», «a». Значит, при использовании шифра Цезаря для текстов на английском языке в шифртекстах наиболее часто встречающимися будут буквы «d», «h» и «w». Если, например, зашифровать текст «**в**с**т**р**е**тимся в п**о**лн**о**чь» с помощью шифра простой замены, заданного случайной перестановкой исходного алфавита, представленной на рис. 1.5, то будет получена криптограмма «с**ч**ц**я**у**ц**оачж с г**з**р**щ**з**ю**ш».

нормативный алфавит	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
алфавит шифрования	ф	к	с	э	и	у	х	п	о	б	ь	р	а	щ	з	г

нормативный алфавит	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	э	ю	я
алфавит шифрования	я	ч	ц	д	ы	й	н	ю	л	в	ш	е	т	м	ж

Рис. 1.5. Пример шифра простой замены

Как видно из этого примера, повторяющимся символам исходного сообщения соответствуют другие, но также повторяющиеся символы криптограммы. Поэтому криптоанализ шифров простой замены может быть проведен с помощью частотного анализа. В частотном анализе наиболее часто встречающиеся символы криптограммы заменяются наиболее часто используемыми символами естественного языка. Частотные характеристики естественного языка изучаются на основе анализа текстов достаточно большой длины. Таблицы частот символов языка можно встретить в орфографических словарях. В табл. 1.1 приведены частоты букв русского языка, в котором отождествлены буквы Е и Ё, Ь и Ъ, а также добавлен знак пробела (–) между словами.

Устойчивыми являются и частотные характеристики биграмм, триграмм, четырехграмм осмысленных текстов. Для русского языка наиболее частыми являются биграммы и триграммы: СТ, НО, ЕН, ТО, НА, ОВ, НИ, РА, ВО, КО, СТО, ЕНО, НОВ, ТОВ, ВО, ОВА. Полезной является информация о сочетаемости букв, то есть о предпочтительных связях букв друг с другом. Эту информацию легко извлечь из таблиц частот биграмм. Еще одна устойчивая закономерность открытых текстов связана с чередованием гласных и согласных букв. Частоты встречаемости биграмм вида гласная-гласная, гласная-согласная,

[5] Бабаш А. В., Шанкин Г. П. Криптография. М.: Солон-Р, 2002. 512 с.

согласная-гласная, согласная-согласная в русском тексте отражены в табл. 1.2. В строках этой таблицы указаны первые, а в столбцах — вторые буквы биграмм.

Таблица 1.1

Вероятности частот букв русского языка
(32-буквенный русский алфавит со знаком пробела)

Символ	*Относительная частота*	*Символ*	*Относительная частота*	*Символ*	*Относительная частота*	*Символ*	*Относительная частота*
–	0,175	З	0,016	П	0,023	Ч	0,012
А	0,062	И	0,01	Р	0,04	Ш	0,006
Б	0,014	И	0,062	С	0,045	Щ	0,003
В	0,038	К	0,028	Т	0,053	Ы	0,016
Г	0,013	Л	0,035	У	0,021	Ь, Ъ	0,014
Д	0,025	М	0,026	Ф	0,002	Э	0,003
Е, Ё	0,072	Н	0,053	Х	0,009	Ю	0,006
Ж	0,007	О	0,09	Ц	0,004	Я	0,018

Таблица 1.2

Чередование гласных и согласных в русском языке

Буква	*Гласная*	*Согласная*	*Всего*
Гласная	6 588	38 310	44 898
Согласная	38 296	16 806	55 102
Всего	44 884	55 116	100 000

Из этой таблицы следует, что для русского языка характерно чередование гласных и согласных[6], причем относительные частоты могут служить приближениями соответствующих условных и безусловных вероятностей:

$$P(\text{с/г}) \approx 0{,}853;\ P(\text{г/с}) \approx 0{,}695;\ P(\text{г}) \approx 0{,}449;\ P(\text{с}) \approx 0{,}551.$$

Приведенные выше закономерности имеют место для обычных открытых текстов на естественном языке. Эти закономерности, в частности, используются при построении формализованных критериев, позволяющих применять методы математической статистики в задаче распознавания открытого текста в потоке сообщений.

Частоты появления символов в конкретном тексте могут существенно отличаться от стандартных (усредненных). Эти отличия могут проявиться тем сильнее, чем короче сообщение. Поэтому частотный анализ коротких сообщений иногда бывает весьма

[6] Васильева И. Н., Куватов В. И. Криптографическая защита информации: практикум. СПб.: СПбУ МВД России, 2017. 260 с.

затруднителен. С другой стороны, перехватив достаточно длинное сообщение или несколько сообщений, зашифрованных одним и тем же шифром простой замены, криптоаналитик может с незначительными трудозатратами дешифровать их и определить используемый шифр. Криптоанализ *простых перестановочных* шифров, например, шифров табличной перестановки, может быть осуществлен как с использованием метода грубой силы (полным перебором возможных перестановок), так и с помощью анализа на основе частот буквосочетаний (биграмм и, в более общем случае, *k*-грамм).

1.2.5. Пропорциональный (омофонический) шифр замены

Разработанные в древности шифры простой замены сохраняют статистические свойства исходного открытого текста и, по этой причине, могут быть легко вскрыты с помощью частотного анализа. Поэтому в Средние века и более позднее время предпринимались попытки усиления шифров замены за счет введения пропорциональности (омофонии) и многоалфавитности.

В *пропорциональных* шифрах (*омофонах*) чем больше частота встречаемости символа в открытом тексте, тем большее число возможных замен используется для его представления в криптограмме. Пример соответствия между символами исходного русскоязычного текста и трехзначными символами криптограммы представлен в табл. 1.3. В процессе шифрования буквы исходного сообщения замену для нее выбирают из списка замен случайным образом.

Таблица 1.3

Таблица замен для пропорционального шифра

Исходный символ	*Варианты замены*	*Исходный символ*	*Варианты замены*
ПРОБЕЛ	751, 769, 758, 801, 849, 104	Р	134, 532
А	760, 128, 350, 201	С	800, 767, 195
Б	101	Т	759, 135, 214
В	210, 106	У	544
Г	351	Ф	560
Д	129	Х	768
Е	761, 130, 802, 352	Ц	545
Ж	102	Ч	215
З	753	Ш	103
И	762, 211, 131	Щ	752
К	754, 764	Ъ	561
Л	132, 354	Ы	136
М	755, 742	Ь	562
Н	763, 756, 212	Э	750
О	757, 213, 765, 133, 353	Ю	570
П	743, 766	Я	216, 104

Применение омофонов является одноалфавитной (простой) заменой, хотя для букв исходного сообщения определено по несколько вариантов замены. Буква открытого текста может быть представлена различными символами, но каждый из этих символов всегда представляет одну и ту же букву. После того как таблица замен определена, она не меняется на протяжении всего процесса шифрования. Пропорциональный (омофонический) шифр устойчив к анализу на основе частот отдельных

символов, но подвержен частотному анализу на основе сочетаний букв (частот биграмм и триграмм) и вероятных слов.

1.2.6. Шифры многоалфавитной замены

В *многоалфавитных* шифрах для замены символов исходного текста используется не один, а несколько шифралфавитов. В процессе шифрования постоянно осуществляется переход от одного шифралфавита к другому, то есть применяются различные простые шифры замены. Поэтому буква открытого текста может быть представлена различными символами, кроме того, один и тот же символ криптограммы может обозначать разные буквы.

Шифр Виженера. Примером многоалфавитной замены служит схема, основанная на использовании таблицы Виженера — шифр Виженера. Этот шифр был описан французом Б. Виженером в «Трактате о шифрах», вышедшем в 1585 году. В шифре Виженера для шифрования используется таблица, представляющая собой квадратную матрицу с числом элементов $N \times N$, где N — мощность (число символов) алфавита (рис. 1.6). В первой строке таблицы записывают буквы в порядке их очередности в исходном алфавите, во второй — ту же последовательность букв, но с циклическим сдвигом влево на одну позицию, в третьей — со сдвигом на две позиции и т. д. Фактически, каждая строка таблицы Виженера представляет собой простую замену — шифр Цезаря с определенным значением сдвига S.

*	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
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
5	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. 1.6. Таблица Виженера для англоязычных текстов

Для шифрования текста выбирают ключ, представляющий собой некоторое слово или набор символов исходного алфавита. Далее из таблицы Виженера получают матрицу шифрования, включающую первую строку и строки матрицы, начальными буквами которых являются последовательно буквы ключа. Так, если выбрать ключ «black», то матрица шифрования примет вид табл. 1.4.

Таблица 1.4

Матрица шифрования для ключевого слова «black»

*	**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
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
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
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
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

В процессе шифрования, отображенном в табл. 1.5, под каждой буквой шифруемого текста записывают буквы ключа. Ключ повторяется циклически, пока не будет достигнут конец шифруемого текста. Затем шифруемый текст по матрице шифрования заменяют буквами, расположенными в ячейке на пересечении столбца, соответствующего букве текста и строки, соответствующей букве ключа. Таким образом, в зависимости от символа ключа к символам открытого текста будут по очереди применяться разные шифры простой замены (шифр Цезаря с разной величиной циклического сдвига алфавита), задаваемые соответствующими строками таблицы Виженера.

Таблица 1.5

Пример шифрования с ключом «black»

Исходный текст	m	a	s	s	i	v	e		a	t	t	a	c	k
Ключ	b	l	a	c	k	b	l		a	c	k	b	l	a
Шифр	n	l	s	u	s	w	p		a	v	d	b	n	k

При шифровании текста «ma**ss**ive a**tt**ack» получили криптограмму «nl**su**swp a**vd**bnk». Как видно, одинаковым символам открытого текста соответствуют различные символы криптограммы (ss — su, tt — vd, символу a соответствуют l, a, b). Кроме того, встречающемуся в криптограмме повторяющемуся символу s соответствуют разные символы открытого текста — s и i. То же можно сказать и о символе криптограммы n, ему соответствуют m и c в открытом тексте. Расшифрование криптограммы, зашифрованной шифром Виженера, выполняется в следующей последовательности: над буквами шифрованного текста сверху последовательно записывают буквы ключа, повторяя ключ требуемое число раз, затем в строке матрицы шифрования для каждой буквы ключа отыскивается буква, соответствующая знаку шифрованного текста. Находящаяся над ней буква первой строки и будет знаком расшифрованного текста.

С математической точки зрения шифр Виженера описывается операциями сложения/вычитания по модулю (символы алфавита закодированы числами от 0 до N–1):

$$y_i = (x_i + k_i) \bmod N \text{ — для шифрования,}$$
$$x_i = (y_i - k_i) \bmod N \text{ — для расшифрования,}$$

где $\{x_n\}$ — последовательность символов исходного текста, $\{y_n\}$ — последовательность символов криптограммы, а $\{k_n\}$ — ключевая последовательность, $0 \le x_i, k_i, y_i \le N-1$, $i = 0, \ldots, n$, n — длина шифруемого текста.

Эти формулы повторяют формулы для шифра Цезаря за тем исключением, что постоянный сдвиг S шифра Цезаря заменен на переменное значение сдвига k_i, зависящее от текущего символа ключевой последовательности. Таким образом, к каждому символу открытого текста фактически применяется свой шифр Цезаря, величина циклического сдвига в котором определяется символом ключевой последовательности.

Шифр Виженера, как и другие многоалфавитные замены, достаточно хорошо маскирует естественные частоты появления символов в тексте, и, как следствие, значительно труднее поддается «ручному» криптоанализу. Однако при неслучайных ключах и ключах, длина которых существенно короче сообщения, он также поддается частотному анализу.

Криптоанализ шифра Виженера проводится в два этапа. На первом этапе определяют *период M* ключевой последовательности. Поскольку короткий ключ многократно повторяется при шифровании, определенные буквы открытого текста (порядковый номер которых кратен длине ключевого слова $M < n$) будут зашифрованы с помощью одного и того же символа ключа, то есть одним и тем же шифром Цезаря. Соответствующие символы криптограммы называются *группой периода* и сохраняют статистику открытого текста. После определения периода ключевой последовательности становится возможным определить зависимость между отдельными символами ключевого слова, а затем восстановить и сам ключ. Для этого перебирают все возможные варианты первого символа ключевого слова (всего будет N таких вариантов), восстанавливают все ключевое слово, а затем отсеивают бессмысленные слова. В случае если ключевое слово представляло собой случайный набор символов, придется опробовать в качестве ключа все N вариантов (расшифровав сообщение и проверив, получается ли ожидаемый осмысленный текст).

Такой метод криптоанализа дает хорошие результаты в том случае, если длина ключевого слова существенно меньше шифруемого текста, либо можно набрать несколько сообщений, зашифрованных одним и тем же ключом. Если же ключ используется однократно, он случаен и имеет ту же длину, что и шифруемый текст, то такой метод криптоанализа не применим. Также если в шифре Виженера заменить шифралфавит с шифра Цезаря на случайные перестановки исходного алфавита, рассмотренный метод криптоанализа не приведет к успешному дешифрованию. Такой вариант шифра многоалфавитной замены активно использовался в первой половине XX века при разработке электромеханических шифрующих устройств — *роторных шифровальных машин* (шифр колонной замены или шифр «Энигмы»).

Шифрмашины. Главными элементами в таких устройствах являлись роторы — механические колеса, реализующие выполнение замены. Роторная шифровальная машина содержала обычно клавиатуру и набор роторов. Каждый ротор содержал набор символов (по количеству в алфавите), размещенных в произвольном порядке, и выполнял простую одноалфавитную замену. После выполнения первой замены ротор смещался, что обеспечивало переход к новому шифру простой замены. Начальные позиции роторов задавали ключ шифрования. Самым известным устройством подобного класса являлась немецкая шифровальная роторная машина «Энигма», использовавшаяся во время Второй мировой войны. Выпускалось несколько моделей «Энигмы» с разным числом роторов. В шифрмашине «Энигма» с тремя роторами можно было использовать более 17,5 тыс. разных алфавитов, и все они представляли собой различные перестановки символов.

Перебор такого числа вариантов «вручную» немыслим, однако появление в послевоенные годы программируемой компьютерной техники привело к существенному увеличению как скорости, так и гибкости подбора ключей, что потребовало разработки новых подходов к шифрованию.

Гаммирование. Гаммирование — метод шифрования, основанный на наложении ключевой последовательности – *гаммы* на открытый текст. По сути, шифр гаммирования представляет собой вариант шифра Виженера, когда алфавит исходных текстов представлен всего двумя символами — 0 и 1. Шифруемый текст и ключ в этом случае представляются в двоичном виде, а криптограмма представляет собой наложение гаммы на открытый текст с помощью логической функции исключающего ИЛИ — XOR (табл. 1.6). При этом обычно предполагается, что гамма является непредсказуемой (случайной или псевдослучайной равномерно распределенной) последовательностью бит.

Таблица 1.6

Таблица истинности операции исключающего ИЛИ (XOR)

Двоичный разряд исходного текста	0	0	1	1
Двоичный разряд ключевой гаммы	0	1	0	1
Двоичный разряд шифртекста	0	1	1	0

Функция побитового XOR эквивалентна сложению по модулю 2 для элементов множества {0, 1} и обладает интересным свойством — она обратна сама себе, то есть сложение и вычитание по модулю 2 — это одна и та же операция. Поэтому для расшифрования криптограммы, полученной с помощью гаммирования, на нее необходимо наложить ту же самую ключевую гамму. Математическая модель шифра гаммирования имеет вид $y_i = x_i \oplus f_i(k)$, где: x_i, y_i — i-й символ открытого текста и криптограммы соответственно, $f_i(k)$ − функция получения символов ключевой гаммы, в общем случае зависящая от номера шифруемого символа и ключа k. При этом $x_i = y_i \oplus f_i(k) = x_i \oplus f_i(k) \oplus f_i(k)$ в силу симметричности операции побитового XOR. Благодаря удобству реализации и формального описания шифры гаммирования обычно выделяются в отдельный класс.

Пусть исходный текст имеет вид 10110, а ключ — вид 11010. Наложим ключ на исходный текст с помощью операции исключающего ИЛИ. Получим зашифрованный текст: 011000. Для расшифрования наложим тот же самый ключ 11010 на зашифрованный текст. В результате получим расшифрованный текст идентичный исходному тексту. Преобразование исходного текста в шифртекст и обратно отображено в табл. 1.7.

Таблица 1.7

Пример шифрования и расшифрования методом гаммирования

Исходный текст	1	0	1	1	0
Ключ	1	1	0	1	0
Зашифрованный текст	0	1	1	0	0
Расшифрованный текст	1	0	1	1	0

Стойкость гаммирования определяется непредсказуемостью ключевой последовательности[7]. Если выполнен ряд условий, а именно: гамма является случайной равномерно распределенной последовательностью бит, она используется однократно, а ее длина не короче длины шифруемого сообщения, то такой шифр (одноразовый шифровальный блокнот, шифр Вернама), несмотря на его кажущуюся простоту, является теоретически стойким. На практике использование теоретически стойкого шифра Виженера оказалось неудобным. Ограничения, накладываемые на ключевую последовательность в шифре Вернама, означают, что между сторонами должен существовать защищенный канал связи для передачи ключей, имеющий ту же пропускную способность, что и канал передачи зашифрованных сообщений. В таком случае отпадает сама необходимость шифрования. С другой стороны, на практике, как правило, такого канала не существует. Поэтому современная симметричная криптография рассматривает разные возможности использования короткого ключа для шифрования длинных сообщений (гамма может быть получена из короткого ключа с помощью псевдослучайной функции $f(k)$).

Таким образом, в первой главе были рассмотрены общие сведения о криптографических методах защиты информации; проанализированы основные понятия, задачи криптографии и исторические методы шифрования. Далее необходимо рассмотреть современные симметричные криптосистемы.

Вопросы для самоконтроля:

1. Какие исторические шифры относятся к перестановкам? Какие исторические шифры относятся к шифрам простой замены?

2. Каковы основные принципы криптоанализа простых шифров?

3. Какая идея лежит в основе пропорционального шифра?

4. В чем суть метода шифрования «Энигмы»?

5. Что понимается под шифром гаммирования? Что такое ключевая гамма?

[7] Коржик В. И., Просихин В. П., Яковлев В. А. Основы криптографии: учебное пособие. СПб.: СПбГУТ, 2014. 276 с.

Глава 2
СОВРЕМЕННЫЕ СИММЕТРИЧНЫЕ КРИПТОСИСТЕМЫ

§ 2.1. Свойства современных криптосистем

2.1.1. Классификация и виды шифров

Наиболее важными признаками классификации шифров являются тип используемых ключей и особенности алгоритма шифрования; количество символов сообщения, шифруемых или расшифровываемых по однотипной процедуре преобразования; стойкость шифра (степень доказуемости безопасности шифра)[8].

Классификация по типу ключа и особенностям алгоритма. По типу ключа криптографические системы делятся на симметричные и асимметричные (с открытым ключом). *В симметричных криптосистемах* для шифрования и дешифрования используется один и тот же секретный ключ. Предполагается, что абоненты симметрической криптосистемы должны знать значение секретного ключа до начала обмена шифрованными сообщениями. Поэтому одной из наиболее острых проблем симметричных криптосистем является проблема распространения секретных ключей. С одной стороны, чтобы снизить вероятность компрометации ключей, их необходимо менять как можно чаще. С другой стороны, задача безопасного распределения секретных ключей требует для своего решения дополнительных времени и средств. Этого недостатка лишены криптосистемы с открытым ключом. В то же время, симметричные криптосистемы обладают большой производительностью.

Асимметричные криптосистемы используют пару ключей, один из которых является *открытым*, доступным любому желающему, а другой — *секретным* (*личным*), известным только его владельцу. Сообщение, зашифрованное одним из ключей, может быть расшифровано только парным к нему ключом. Открытые ключи абонентов могут распространяться свободно, функцию управления ключами обычно выполняет *центр сертификации*, которому доверяют все абоненты криптосистемы.

Вместе с тем шифрование/расшифрование в криптосистемах с открытым ключом связано с большими объемами вычислений и характеризуется низкой скоростью, не достаточной для обмена шифрованными сообщениями в режиме реального времени. Поэтому в настоящее время асимметричные криптосистемы используются в основном для решения задачи распределения секретных ключей симметричных криптосистем, защиты ключевой информации и цифровой подписи. Криптосистемы, в которых задачи аутентификации и распространения ключей решаются с помощью алгоритмов с открытым ключом, а шифрование выполняется симметричным шифром, называют *гибридными*.

Еще одна классификация указывает на число используемых криптографическим алгоритмом ключей. К *бесключевым* алгоритмам относятся бесключевые хэш-функции, к *одноключевым* — симметричные криптосистемы и коды аутентификации (MAC, HMAC), к *двухключевым* — криптосистемы с открытым ключом. Иногда упоминаются *многоключевые* криптосистемы, под которыми обычно понимаются схемы разделения секрета.

По особенностям алгоритмов шифрования симметричные криптосистемы могут быть подразделены на *перестановки*, *простые замены* и *гаммирование*. Большинство современных симметричных криптосистем являются *композиционными*, то есть используют преобразования всех этих типов.

---

[8] Васильева И. Н. Криптографические методы защиты информации: учебник и практикум для академического бакалавриата. М.: Юрайт, 2016. 349 с.

Алгоритмы асимметричных криптосистем базируются на математических преобразованиях и могут быть детерминированными или вероятностными (рандомизированными). При использовании *детерминированного* алгоритма шифрование и расшифрование с определенной парой ключей возможно единственным способом. *Вероятностный* алгоритм наряду обычно использует некоторый случайный параметр и при шифровании одного и того же исходного сообщения с одним и тем же ключом может давать разные шифртексты, которые при расшифровании дают один и тот же результат.

Отдельным типом являются *квантовые* криптосистемы (системы квантового распределения ключа), базирующиеся на принципах квантовой физики. Процесс отправки и приема информации осуществляется посредством объектов квантовой механики, например, при помощи фотонов в линиях волоконно-оптической связи. Самым ценным свойством криптосистем этого вида является то, что при посылке сообщения отправляющая и принимающая сторона с достаточно большой вероятностью могут установить факт его перехвата.

Классификация по способу формирования криптограммы из сообщения. По способу формирования криптограммы из сообщения криптосистемы системы делятся на два основных типа: блочные и потоковые. Основной критерий такого разделения — независимость преобразования символов входного текста. В *блочных криптосистемах* результат шифрования зависит от всех символов в рамках определенного блока входного текста. Примером могут служить исторические шифры перестановки. Современные блочные симметричные криптосистемы разбивают шифруемое сообщение на блоки определенной длины. Каждый блок шифруется отдельно, по одному и тому же правилу.

Потоковые криптосистемы обрабатывают информацию последовательно порциями единичной длины (обычно посимвольно, если речь идет об исторических шифрах, и побитно — в современных криптосистемах). Частным случаем потокового шифра является шифр гаммирования.

Потоковые криптосистемы, как правило, обеспечивают высокую скорость преобразования информации, что важно, например, в телефонных сетях. Блочные более часто применяются в сетях передачи данных, так как они обеспечивают прозрачный для пользователя режим шифрования информации, хранящейся в памяти компьютера. Прозрачность означает, что пользователь работает с данными, будто они не зашифрованы. Процедура шифрования/дешифрования данных от него скрыта. Асимметричные криптосистемы, как правило, осуществляют преобразование информации блоками, в то время как симметричные могут быть как блочными, так и потоковыми.

Следует отметить, что разделение современных симметричных криптосистем на блочные и потоковые весьма условно, поскольку способ применения криптографического алгоритма определяется определенным режимом работы. Так для симметричных блочных шифров определены режимы гаммирования, когда шифр фактически используется для формирования псевдослучайной гаммы, которая затем накладывается на открытый текст с помощью операции побитового XOR. В то же время потоковые шифры (например, Cha-Cha20) могут включать на этапе формирования гаммы элементы вычислений, характерные для блочных криптосистем.

Классификация по стойкости шифров. По стойкости шифра криптосистемы делятся на две группы: совершенные (по К. Шеннону) и вычислительно стойкие. *Совершенные* (совершенно стойкие, теоретически стойкие — по К. Шеннону) — шифры, заведомо неподдающиеся вскрытию, то есть не допускающие однозначного дешифрования ни при каком объеме перехваченных данных. Безопасность таких шифров может быть доказана математически. Дешифрование криптограммы приводит к множеству осмысленных открытых текстов, каждый из которых с одинаковой вероятностью может оказаться истин-

ным. Примерами совершенных криптосистем является система с одноразовым использованием случайных ключей (одноразовый шифровальный блокнот, шифр Вернама), а также система квантового распределения ключей. Применение совершенных криптосистем на практике имеет ряд ограничений, поэтому наиболее распространены криптосистемы другого типа, стойкость которых базируется на вычислительной сложности процедур их криптоанализа. Это *вычислительно стойкие* криптосистемы, взлом которых на практике в условиях ограниченности ресурсов, которыми обладает криптоаналитик, не представляется возможным.

При построении вычислительно стойких криптосистем с открытым ключом используется принцип *доказуемой стойкости*, поскольку их криптоанализ сводят к решению некоторой известной заведомо трудной вычислительной задачи (например, задачи дискретного логарифмирования или факторизации больших чисел).

2.1.2. Формальные модели шифров

Несмотря на то, что различные шифры известны с древних времен, первое формальное описание криптосистем было сделано лишь в 1949 году в работе К. Шеннона «Теория связи в секретных системах» (Communication Theory of Secrecy Systems). Для того чтобы иметь возможность получать новые теоретические результаты в криптографии, нужны математические модели основных шифров. Формальные модели введены К. Шенноном для симметричных криптосистем и позволяют формализовать понятие теоретической стойкости шифра. Существует две основные модели шифров: алгебраическая и вероятностная. Предложенные К. Шенноном модели могут быть обобщены и на случай криптосистем с открытым ключом. Обычно предполагается, что открытый текст, шифртекст и ключ представлены в некоторой числовой кодировке (целыми числами, битовыми последовательностями, компьютерными кодировками).

Алгебраическая модель симметричного шифра. Пусть X, Y, K — конечные множества открытых текстов, шифртекстов и ключей шифрования соответственно. На прямом (декартовом) произведении множеств $X \times K$ задана функция — криптографическое отображение f: $X \times K \to Y$ ($f(x, k) = y$, $x \in X$, $\in K$, $y \in Y$). Функции f соответствует множество отображений $f_k(x) = f(x, k)$. Таким образом, отображение: f_k: $X \to Y$, $k \in K$, — это ограничение функции f на множестве $X \times \{k\}$. Здесь $\{k\}$ — множество из одного элемента — конкретного значения ключа шифрования.

Тройка множеств X, K, Y с функцией f: $X \times K \to Y$ называется *шифром* $A = (X, K, Y, f)$, если выполнены два условия:

— функция f — сюръективна (осуществляет отображение «на» Y);

— для любого $k \in K$ функция f_k инъективна.

Инъективность функции f_k означает, что образы разных элементов различны: для любых различных открытых текстов x_1 и x_2: $x_1 \neq x_2$, соответствующие криптограммы, полученные на одном и том же ключе k, различны: $y_1 = f_k(x_1)$, $y_2 = f_k(x_2)$: $y_1 \neq y_2$. Требование инъективности f_k обеспечивает возможность однозначного расшифрования криптограммы. Инъективные функции определяют взаимно однозначные соответствия между множествами X и Y, поэтому $|X| \leq |Y|$.

Шифрующее преобразование f часто обозначают через E, а обратное к нему преобразование расшифрования f^{-1} — через D. Запись $f_k(x) = y$ или $E_k(x) = y$ называется *уравнением шифрования*, а $f_k^{-1}(x) = y$ или $D_k(x) = y$ — *уравнением расшифрования* на ключе k.

Шифр $A = (X, K, Y, f)$ называют *транзитивным*, если при любых $x \in X$ и $y \in Y$ найдется значение $k \in K$, при котором $f(x,k) = y$. Для транзитивных шифров уравнение

$f(x,k) = y$ разрешимо относительно $k \in K$ для любых пар $(x, y) \in X \times Y$, и выполняется неравенство $|X| \le |Y| \le |K|$.

Алгебраическая модель шифра отражает лишь функциональные свойства шифрования и расшифрования одним и тем же ключом в классических симметричных криптосистемах. В этой модели открытый текст и шифртекст — лишь элементы абстрактных множеств X и Y соответственно, особенности языка сообщений, его статистические свойства не учитываются.

Алгебраическая обобщенная модель шифра позволяет описать не только симметричные криптосистемы, но и криптосистемы с открытым ключом, использующие различные, но взаимосвязанные, ключи в процедурах шифрования и расшифрования. Фактически вводится две модели: для шифра шифрования и для шифра расшифрования, совокупность которых и составляет обобщенную модель шифра.

Шифр зашифрования может быть определен как $A_ш = (X, K_ш, Y, f)$,
где f: $X \times K \to Y$ — сюръективное отображение, причем для каждого $k \in K_ш$ отображение f_k: $X \to Y$, $f_k(x) = f(x, k)$ — инъективно.

Шифр расшифрования определяется как $A_р = (Y', K_р, X', g)$, где $Y \subseteq Y'$, $X \subseteq X'$, g: $Y' \times K_р \to X'$ — сюръективное отображение, для которого выполнены следующие условия:

— существует биекция (сюръективное и инъективное отображение) φ: $K_ш \to K_р$;

— для любых $x \in X$ и $k \in K_ш$ из условия $f(x, k) = y$ вытекает $g(y, \varphi(k)) = x$.

Алгебраической обобщенной моделью шифра называется тройка $(A_ш, A_р, \varphi)$.

Вероятностной моделью шифра (по К. Шеннону) называется его алгебраическая модель $A = (X, K, Y, f)$ с заданными дискретными независимыми вероятностными распределениями $P(X) = \{P(x), x \in X\}$ и $P(K) = \{P(k), k \in K\}$ на множествах открытых текстов X и ключей K соответственно.

Эти вероятностные распределения (*априорные* вероятности различных возможных сообщений и различных ключей) характеризуют исходные знания криптоаналитика относительно используемого шифра. При этом предполагается, что сам шифр известен противнику. После того как противник перехватил криптограмму, он может вычислить, по крайней мере теоретически, *апостериорные* вероятности различных открытых текстов и ключей. Если криптосистема допускает однозначное дешифрование, то при увеличении длины перехваченного шифртекста апостериорные вероятности некоторых из них будут возрастать. Для всех других апостериорные вероятности убывают. В конце концов, останется только одно сообщение, имеющее вероятность, близкую к единице, в то время как полная вероятность всех других будет близка к нулю. Однако эти вычисления могут быть эффективно выполнены лишь для простых криптосистем.

2.1.3. Криптостойкость и имитостойкость шифра

Теоретическая стойкость шифра. По К. Шеннону, основным свойством теоретически стойкого (совершенного) шифра является то, что положение криптоаналитика не облегчается в результате перехвата любого количества зашифрованных сообщений.

К. Шеннон назвал шифр *совершенным*, если знания, которые могут быть получены при анализе шифртекста, не раскрывают никакой информации о соответствующем ему открытом тексте, за исключением, возможно, длины последнего. Таким образом, для совершенных шифров апостериорные вероятности открытых текстов, вычисленные после перехвата криптограммы, совпадают с их априорными вероятностями, $P(x / y) = P(x)$ для любых $x \in X$ и $y \in Y$. Здесь $P(x)$ — априорная вероятность выбора сообщения x (обусловленная, например, избыточностью и статистической неравномерностью языка сооб-

щений), $P(x / y)$ — апостериорная вероятность того, что было передано именно сообщение x при условии, что перехвачена криптограмма y.

Эквивалентное (основанное на теории информации К. Шеннона) определение теоретически стойкого шифра: информация, содержащаяся в криптограмме y о сообщении x равна нулю, $I(y, x) = 0$.

Учитывая формулу Байеса для вычисления апостериорных вероятностей $P(x / y) = \frac{P(y / x)P(x)}{P(y)}$, можно получить необходимое и достаточное условие совершенной секретности: $P(y / x) = P(y)$ для любых $x \in X$ и $y \in Y$, где $P(y)$ — безусловная вероятность получения криптограммы y, $P(y / x)$ — условная вероятность получения криптограммы y при условии, что выбрано сообщение x, то есть суммарная вероятность всех тех ключей, которые переводят сообщение x в криптограмму y. Таким образом, вероятность $P(y / x)$ не должна зависеть от x.

Совершенные криптосистемы обладают рядом свойств: шифр с ограниченным ключом не является совершенным; у совершенного шифра может быть ограниченный ключ, но только если длина ключа равна длине сообщения; число возможных ключей должно быть не меньше числа сообщений, которые можно зашифровать на этих ключах; совершенный шифр является транзитивным; для совершенного шифра выполняются неравенства $|X| \le |Y| \le |K|$.

Примером совершенной криптосистемы является шифр Вернама (одноразовый шифровальный блокнот). В этом шифре шифрование и расшифрование производятся аналогично шифру Виженера, однако должны быть выполнены условия, связанные с применением ключа: ключ является равномерно распределенной случайной последовательностью; длина ключа не короче подлежащего шифрованию текста; ключ используется однократно.

Рассмотрим шифр гаммирования (криптограмма получается путем побитного сложения открытого текста с ключом по модулю 2): $y = x \oplus k$, $X, Y, K \in (0,1)^N$, открытое сообщение, ключ и криптограмма отображаются двоичными цепочками длины N. При соблюдении условий, накладываемых шифром Вернама на ключ, шифр гаммирования будет совершенным.

Пусть сообщение x генерируется источником без памяти, обозначим вероятность того, что очередным символом сообщения будет 1 через q: $P(x = 1) = q$, тогда $P(x = 0) = 1 - q$. В предположении, что ключ k случаен и равномерно распределен $P(k = 0) = P(k = 1) = 1/2$.

Найдем вероятность того, что очередной символ криптограммы будет равен 1. Это произойдет, если в исходном сообщении соответствующий символ равен 0, а в ключе — 1, или в сообщении — 1, а в ключе — 0. Эти пары событий несовместные, кроме того, значения битов ключа не зависят от значений битов сообщения, поэтому: $P(y = 1) = (1 - q) * 1/2 + q * 1/2 = 1/2$, тогда и $P(y = 0) = 1/2$.

Таким образом, вероятность появления в криптограмме единицы или нуля не зависит от статистических свойств исходного сообщения, а значит выполнено необходимое и достаточное условия совершенной стойкости криптосистемы. Данное свойство обусловлено случайностью и равномерным распределением гаммы. Если бы это было не так, вероятность q невозможно было бы исключить.

Отметим, что даже при полном переборе всех вариантов ключа (не учитывая трудоемкость такого перебора) однозначное дешифрование криптограммы совершенного шифра невозможно. В результате такого перебора будут получены все возможные осмысленные сообщения определенной длины, при этом ни одному из них нельзя отдать предпочтение и принять за истинное, поскольку все ключи — равновероятны.

Несмотря на абсолютную стойкость, на практике применение совершенных криптосистем весьма ограничено. Это связано, прежде всего, с необходимостью распределения большого объема ключевой информации, не меньшей, чем длина защищаемого текста.

Вычислительная стойкость шифра. Помимо совершенно стойких (по К. Шеннону) шифров, существуют криптосистемы, которые дают криптоаналитику некоторую информацию при перехвате криптограммы. При этом они могут допускать или не допускать однозначное дешифрование криптограммы. Криптосистемы, допускающие однозначное дешифрование, требуют определенных затрат вычислительных ресурсов, а также определенного объема перехваченных сообщений. К. Шеннон показал, что существует некоторая длина перехваченного шифртекста y, после которой ключ шифрования k может быть найден с вероятностью, близкой к единице.

Под *расстоянием единственности* понимается минимальное число символов перехваченного шифртекста, при котором криптосистема допускает однозначное дешифрование. Ниже приводится формальное определение расстояния единственности.

Определим *функцию надежности* ключа через условную информационную энтропию ключа k и символов перехваченного шифртекста $y_1, y_2,\ldots,y_n$:

$$f(0) = H(k), f(1) = H(k/y_1), f(2) = H(k/y_1,y_2), \ldots, f(n) = H(k/y_1,y_2,\ldots,y_n).$$

Информационная энтропия ключа определяется по формуле:

$$H(k) = -\sum_{i=1}^{N} P_i log_2(P_i),$$

где P_i — вероятность использования i-го ключа, N — число возможных ключей. Если выбор ключей из ключевого пространства K осуществляется равновероятно, то энтропия ключа достигает своего максимального значения $H(k) = \log |K|$.

Очевидно, что чем больше объем перехваченного шифртекста (чем больше n), тем меньше энтропия:

$$f(0) \geq f(1) \ldots \geq f(n-1) \geq f(n).$$

Расстоянием единственности U называется такая длина n, при котором $f(n)=0$, то есть неопределенность ключа шифрования равна нулю.

К. Шенноном получены значения расстояния единственности для некоторых классических криптосистем (для английского языка), например, для шифра простой замены теоретическое значение $U = 27$, а для шифра Виженера со случайным ключом $U \approx 2M$, где M — период ключевой последовательности. К сожалению, большинство криптосистем слишком сложны для точного определения значения расстояния единственности. Кроме того, значение расстояния единственности никак не характеризует вычислительную сложность процедуры дешифрования, то есть даже при перехвате шифртекста длиной $n > U$ дешифрование может оказаться практически не осуществимым.

Средний объем работы $W(n)$, необходимый для определения ключа по криптограмме, состоящей из n букв, измеренный в элементарных операциях, Шеннон назвал *рабочей характеристикой шифра*. Средний объем вычисляется по всем ключам с учетом их вероятностей. То есть функция $W(n)$ характеризует средние временные и емкостные затраты, требующиеся для расшифрования криптограммы.

К. Шеннон исследовал рабочую характеристику для некоторого модельного «случайного» шифра, сопоставив ее с неопределенностью шифра по открытому сообщению (как функцией длины сообщения), рис. 2.1. Пунктирная линия функции $W(n)$ на рис. 2.1 относится к области, где имеется более одного возможного решения

(расстояние единственности не достигнуто). Как только расстояние единственности достигнуто, на рисунке это точка $f(n) = 0$, решение оказывается единственным, но объем работы для определения ключа остается достаточно большим. По мере увеличения объема перехвата количество необходимой работы быстро уменьшается, стремясь к некоторому асимптотическому значению, которое достигается, когда дополнительные данные уже не уменьшают работы.

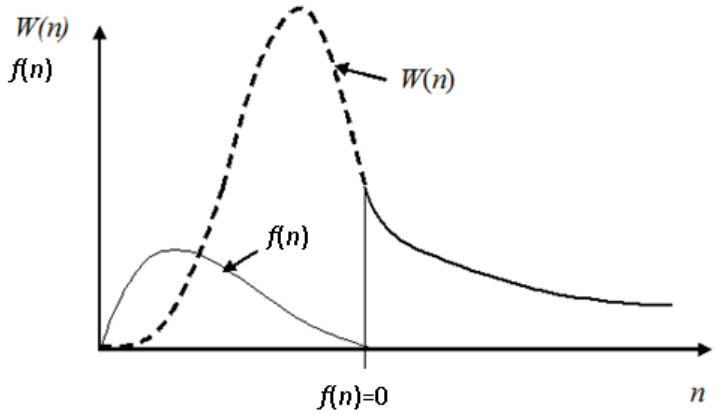

Рис. 2.1. Зависимость величины неопределенности шифра от длины открытого сообщения

Наибольший интерес представляет предельное значение $W(n)$ при $n \to \infty$, которое можно рассматривать как среднюю работу по дешифрованию при неограниченном объеме шифртекста. Теоретическое вычисление $W(\infty)$ представляет собой весьма сложную задачу. В связи с этим практическую (вычислительную) стойкость шифра обычно оценивают с помощью величины $W_Д(\infty)$, которую можно назвать *достигнутой оценкой* рабочей характеристики. Она определяется средней трудоемкостью наилучшего из известных методов вскрытая данного шифра.

Обычно практическая стойкость конкретного шифра оценивается только путем всевозможных попыток его вскрытия и зависит от квалификации криптоаналитиков, атакующих шифр. Такую процедуру иногда называют *проверкой стойкости*. Лучшая из возможных атак (требующая минимальных ресурсов), характеризует вычислительную криптостойкость алгоритма. Если значение $W_Д(\infty)$ вычисляет криптоаналитик, которому известны все имеющиеся в настоящее время методы анализа шифров, то полученная оценка практической стойкости шифра заслуживает доверия, особенно если вычисления сделаны с достаточным «запасом».

Криптосистема считается вычислительно стойкой, если не существует каких-либо иных методов его вскрытия, более эффективных (то есть менее трудоемких), чем *метод грубой силы* (*brute force*) — перебор всех возможных вариантов ключа; размер ключа криптоалгоритма достаточно велик для того, чтобы использование метода «грубой силы» было практически неосуществимо при текущем уровне развития вычислительной техники.

Поскольку вычислительная мощность компьютерных систем постоянно увеличивается, требования к размеру ключей со временем возрастают. В настоящее время существуют рекомендации по длинам ключей для обеспечения достаточной вычислительной стойкости криптосистем на практике: не менее 128 бит для симметричных криптосистем; не менее 2048 бит для криптосистемы RSA с открытым ключом (для обеспечения стойкости, эквивалентной стойкости симметричной криптосистемы со 128-битным ключом, RSA должна иметь ключ длиной более 2 300 бит); не менее 256 бит для криптосистем на эллиптических кривых.

Имитостойкость шифра. В общем случае, помимо пассивных действий противника, состоящих в перехвате передаваемых криптограмм с последующими попытками их криптоанализа, возможны также его активные действия, состоящие в попытках подмены или имитации сообщения. Если передается криптограмма $y \in Y$, то противник может подменить ее криптограммой $y' \neq y$. При этом противник будет рассчитывать, что криптограмма y' при расшифровании будет воспринята как некий осмысленный текст $x' \neq x$. Текст x' может быть с некоторой вероятностью воспринят получателем, как текст, переданный легальным отправителем. Чем выше эта вероятность, тем успешнее попытка подмены. Формально событие, состоящее в попытке *подмены криптограммы*, запишется в виде $(D_k(y') \in X) \cap (y' \neq y)$.

Попытка имитации может быть предпринята противником в том случае, если линия связи готова к работе (на передаче и на приеме установлены действующие ключи), но в рассматриваемый момент сообщения легальными абонентами не передаются. В таком случае противник может выбрать некоторую криптограмму $y \in Y$ и послать ее от имени легального отправителя. При этом он будет рассчитывать на то, что на действующем ключе его криптограмма при расшифровании будет воспринята как некий осмысленный текст. Чем больше вероятность этого события, тем успешнее попытка имитации. Формально событие, состоящее в *имитации криптограммы* противником, запишется в виде $D_k(y) \in X$

Имитостойкость шифра (стойкость к подмене и имитации) есть его способность противостоять попыткам противника по подмене или имитации криптограммы. Мерой имитостойкости является вероятность навязывания, состоящего в подмене или в имитации криптограммы:

$$P_{\text{подм}} = \max_{y,y' \in Y} P(D_k(y') \in X),$$

$$P_{\text{им}} = \max_{y \in Y} P(D_k(y) \in X).$$

Будем исходить из того, что противник выбирает подмену или имитацию, в зависимости от того, что из этих действий приведет к успеху с большей вероятностью. Тогда вероятность навязывания рассчитывается по формуле:

$$P_{\text{н}} = max\left(P_{\text{подм}}, P_{\text{им}}\right)$$

Для шифров с равновероятными ключами можно получить общие оценки введенных вероятностей:

— достижимая оценка вероятности успешной имитации $P_{\text{им}} \geq |X|/|Y|$;

— достижимая оценка вероятности успешной подмены $P_{\text{подм}} = (|X| - 1)/(|Y| - 1)$.

Для эндоморфного шифра с равновероятными ключами (например, для шифра гаммирования с равновероятной гаммой) $P_{\text{им}} = 1$. Это означает, что такой шифр максимально уязвим к угрозе имитации сообщения и нуждается в имитозащите. Имитостойкость шифра растет пропорционально отношению |Y|/|X|. Это объясняет широко используемый для имитозащиты способ *введения избыточности* в передаваемое

сообщение, например, «добавок» к передаваемому сообщению типа *кодов аутентификации* (*имитовставок*).

В общем случае неизвестно, при каких условиях существуют шифры, обеспечивающие совершенную имитостойкость, хотя и имеются соответствующие примеры. Эти примеры свидетельствуют о том, что криптостойкость и имитостойкость являются независимыми свойствами шифра.

В настоящее время для повышения имитостойкости используют системы аутентифицированного шифрования, позволяющие обеспечить как защиту конфиденциальности, так и контроль целостности передаваемых сообщений.

Вопросы для самоконтроля:

1. Каковы основные классификационные признаки криптосистем?

2. Каковы основные отличия симметричных и асимметричных криптосистем?

3. Приведите определение алгебраической модели шифра.

4. Что понимается под практической стойкостью шифров? Каковы актуальные рекомендации к длинам криптографических ключей?

5. Что понимается под имитацией, подменой криптограммы, имитостойкостью шифра?

§ 2.2. Блочные шифры

2.2.1. Структура блочных шифров

Принципы построения блочных шифров. Большинство современных симметричных криптосистем преобразуют исходную информацию блоками одинаковой длины (64, 128, 256 бит и т. д.) с использованием одного и того же ключа. При этом шифры должны обладать свойством вычислительной стойкости[9]. К. Шеннон сформулировал общие принципы, которые определили генеральный путь синтеза блочных шифров. В алгоритме вычислительно стойкого блочного шифра необходимо использовать подстановки (замены) одних блоков на место других и перестановки символов в блоках, причем делать это многократно и с разными ключами.

Такие шифры являются *композиционными*, то есть заключаются в многократном последовательном выполнении разнотипных элементарных криптографических преобразований. Многократно повторяющийся набор элементарных криптографических операций называется *раундом шифрования*, а используемый при этом ключ k_i — раундовым ключом, $i = 1, \ldots, r$. При этом выходные данные раунда считаются входными для следующего. Обычно раундовые ключи получаются из исходного ключа k системы с помощью алгоритма выработки раундовых ключей (при этом $|k|$ существенно меньше суммарного размера всех раундовых ключей). Процедура генерации раундовых ключей из исходного часто называется *процедурой расширения ключа*. Порядок генерации и использования раундовых ключей называется *расписанием использования ключа шифрования*. Как правило, чем больше раундов шифрования, тем выше вычислительная стойкость, но ниже эффективность реализации (скорость работы) композиционного шифра.

Многократное использование преобразований замены и перестановки позволяет обеспечить два свойства: рассеивание (diffusion) и перемешивание (confusion), совместное проявление которых обеспечивает практическую стойкость композиционного шифра.

[9] Панасенко С. П. Алгоритмы шифрования. Специальный справочник. СПб.: БХВ-Петербург, 2009. 564 с.

Рассеивание предполагает распространение влияния одного знака открытого текста, и также одного знака ключа на значительное количество знаков шифртекста и достигается использованием перестановок. Наличие у шифра этого свойства позволяет скрыть статистическую зависимость между знаками открытого текста, иначе говоря, перераспределить избыточность исходного языка посредством распространения ее на весь текст; не позволяет восстанавливать неизвестный ключ по частям. Например, обычная перестановка символов позволяет скрыть частоты появления биграмм, триграмм и т. д.

Цель *перемешивания* — сделать как можно более сложной зависимость между ключом и шифртекстом. Криптоаналитик на основе статистического анализа перемешанного текста не должен получить значительного количества информации об использованном ключе. Обычно перемешивание осуществляется при помощи подстановок (замен). Подстановки (замены) представляют собой нелинейные преобразования блоков в блоки такой же длины (рис. 2.2). Преобразования часто задаются фиксированной двумерной или одномерной таблицей (массивом). Часто таблицы замен применяются не к входному блоку в целом, а к его подблокам гораздо меньшей длины.

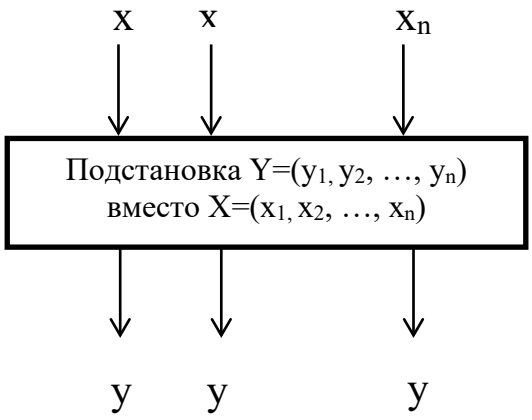

Рис. 2.2. Схематическое изображение блока замен

Криптографическая стойкость шифра, в частности, его устойчивость к методам линейного криптоанализа, во многом определяется именно нелинейностью S-блоков. Вместе с тем, чем больше размер таблицы замен, тем большая степень нелинейности преобразования может быть обеспечена. Раньше таблицы замен формировались вручную, в настоящее время для этих целей используется специальный математический аппарат (например, бент-функции), обеспечивающий максимально возможную нелинейность S-блока.

Для устранения зависимости между символами разных подблоков, которая может присутствовать в исходном сообщении, в раунд алгоритма шифрования включают перестановки. Перестановка — это преобразование, которое показывает, на какую позицию в выходном блоке должен попасть стоящий на определенной позиции символ входной последовательности (рис. 2.3). Перестановки могут задаваться в виде таблицы (P-блок), содержащей номера бит входного текста, например, перестановка, приведенная на рис. 2.3, может быть задана последовательностью {n, 3, 1, … , 2}.

В качестве перестановок могут также использоваться операции циклического сдвига битов блока, линейные регистры сдвига с обратной связью и т. п. Для наложения материала раундового ключа обычно используется операция исключающего ИЛИ (побитового XOR), однако могут использоваться и другие приемы, например, сложение по модулю 2^n, где n — длина раундового ключа.

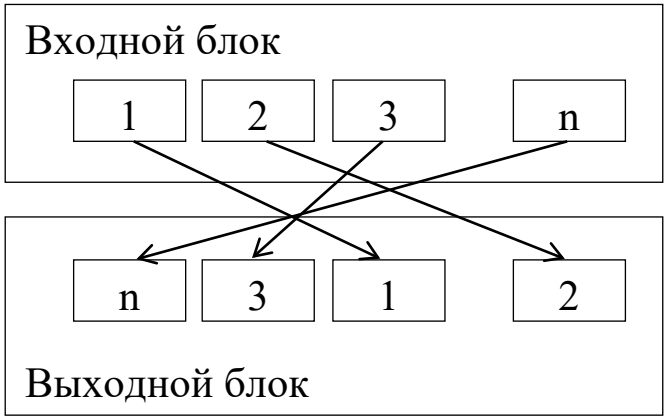

Рис. 2.3. Пример перестановки

Основным ограничением при построении композиционного шифра является запрет на использование подряд идущих однотипных операций как в рамках одного раунда, так и на границах раундов, так как любое количество подряд выполняющихся однотипных операций можно заменить одной эквивалентной. Если на границе соседних раундов оказались два однотипных преобразования, между ними вставляют другое простое преобразование (*буфер*), разрушающее эту однотипность.

На практике чаще всего используется два вида блочных шифров: шифры на основе схемы Фейстеля и подстановочно-перестановочные сети. В качестве примеров блочных шифров со структурой Фейстеля можно указать старый американский стандарт DES и российский шифр «Магма» ГОСТ 34.12–2018 (бывший ГОСТ 28147–89). Выделяют алгоритмы, построенные на расширенной сети Фейстеля (RC6, CAST-256), входной блок которых делится на кратное число подблоков, и в каждом раунде преобразуется лишь один подблок. К подстановочно-перестановочным сетям относят, например, современный российский шифр «Кузнечик» ГОСТ 34.12–2018, а также шифры SAFER+, Serpent. В последнее время в отдельную группу относят шифры со структурой «квадрат», в которых входной блок данных представлен в виде двумерной матрицы байтов. Такую структуру, в частности, имеет шифр Rijndael, на котором основан современный американский стандарт блочного шифра AES. Шифры со структурой «квадрат» можно рассматривать как частный случай подстановочно-перестановочной сети.

Существуют и шифры с нестандартной структурой (такие как FROG). Строго классифицировать все возможные варианты алгоритмов шифрования не представляется возможным, границы между различными структурами также строго не определены, поэтому зачастую один и тот же алгоритм относят к разным типам.

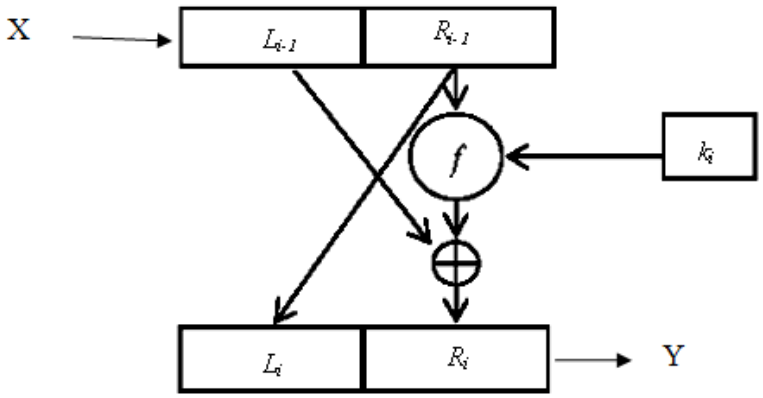

Рис. 2.4. Схема Фейстеля

Блочные шифры на основе схемы Фейстеля. Для упрощения процедур шифрования и расшифрования во многих блочных шифрах используют схему Фейстеля, основанную на следующих принципах: каждый входной блок делится на две равные части — левую L_i и правую R_i, где i — номер итерации (раунда); способ формирования следующих половинок выходного блока раунда выполняется по схеме, отображенной на рис. 2.4.

Преобразование, реализуемое сетью Фейстеля в i-м раунде, имеет вид

$$\begin{cases} L_i = R_{i-1}, \\ R_i = L_{i-1} \oplus f(R_{i-1}, k_i), \end{cases} \quad (2.1)$$

где $\oplus$ — операция побитового XOR (сложение по модулю два), f — нелинейная функция двух аргументов: выхода предыдущего раунда R_{i-1} и раундового ключа k_i.

Алгоритм шифрования реализуется несколькими итерациями (раундами) преобразований сети Фейстеля. В очередном (i-м) раунде в качестве входного блока используется результат, вычисленный в предыдущем раунде (L_{i-1}, R_{i-1}) и раундовый ключ (k_i), полученный из основного ключа с помощью детерминированного алгоритма.

Алгоритмы на основе схемы Фейстеля являются наиболее изученными. Основной особенностью схемы Фейстеля является то, что даже если раундовая функция f необратима, преобразование сети Фейстеля обратимо. Следовательно, для расшифрования в схеме Фейстеля можно использовать ту же функцию f, но с обратным порядком следования раундовых ключей.

Для доказательства представим i-й раунд шифрования в виде $y_i = f_i(x_i)$, $f_i = T \cdot V_i$, где T — перестановка половинок блока $T(L, R) = (R, L)$, $V_i(L, R) = (L \oplus f(R, k), R)$. Тогда для полного шифра $y = f_r, f_{r-1}, \dots, f_2, f_1(x)$, где r — число раундов. Расшифрование выполняется обратными преобразованиями

$$x = \left(f_r, f_{r-1}, \dots, f_2, f_1\right)^{-1}(y) = f_1^{-1}, f_2^{-1}, \dots, f_n^{-1}(y) =$$

$$= V_1 T V_2 T \dots V_n T(y) = V_1 f_2 f_3 \dots f_n T(y),$$

поскольку $f_i^{-1} = V_i T$. Таким образом, свойство обратимости доказано.

Обе половины блока постоянно меняются местами, поэтому, несмотря на кажущуюся несимметричность, они шифруются с одинаковой стойкостью.

Существует достаточно много вариантов блочных шифров на основе схемы Фейстеля, различающихся видом нелинейной функции f, числом раундов, размерами блока шифрования и способом выработки раундовых ключей из исходного ключа. В частности, на схеме Фейстеля построены алгоритм старого американского стандарта шифрования DES, шифра «Магма» отечественного стандарта ГОСТ 34.12–2018 «Информационная технология. Криптографическая защита информации. Блочные шифры», шифры RC5, Blowfish, TEA, CAST-128 и др.

Основным достоинством схемы Фейстеля является возможность использования широкого спектра преобразований, в том числе и необратимых, в качестве раундовой функции f, а также одинаковая структура раунда алгоритма шифрования и расшифрования. Вместе с тем в каждом раунде производится преобразование лишь половины входного блока, что в общем случае требует достаточно большого числа раундов для построения стойкого шифра. Так, например, шифры DES, Blowfish имеют 16 раундов, «Магма», TEA, — по 32 раунда (по сравнению с 10 раундами шифров AES-128, «Кузнечик»).

Шифры, базирующие на подстановочно-перестановочной сети, преобразуют за один раунд все биты входного блока данных, однако раундовые функции алгоритмов шифрования и расшифрования различны.

Подстановочно-перестановочные шифры (**SP-сети**). Шифры, построенные на принципах Шеннона и преобразующие входной блок целиком (рис. 2.5), получили название *подстановочно-перестановочных* сетей.

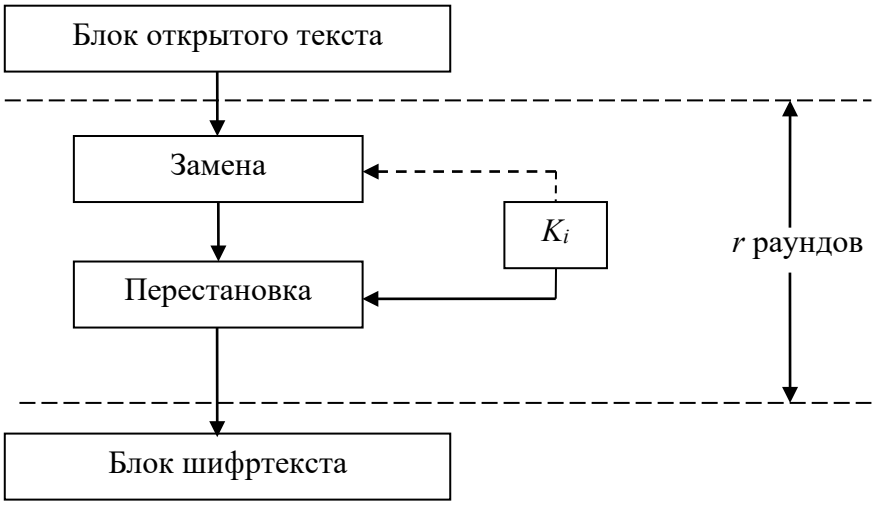

Рис. 2.5. Обобщенная схема подстановочно-перестановочной сети

За счет преобразования всего блока данных на каждом шаге обеспечивается гораздо более быстрое перемешивание входных данных по сравнению со схемой Фейстеля, поэтому современные криптосистемы на SP-сетях обеспечивают необходимую вычислительную стойкость за меньшее, по сравнению со схемой Фейстеля, число раундов. С другой стороны, для SP-сети функция расшифрования отличается от функции шифрования.

2.2.2. Американские стандарты блочного шифрования

Наиболее изученным блочным алгоритмом симметричного шифрования является старый американский стандарт DES (*Data Encryption Standard*), принятый в 1977 году и действовавший более 20 лет. В 2000 году ввиду недостаточной длины ключа он был заменен новым алгоритмом, построенным на принципиально другой схеме, — стандартом AES (*Advanced Encryption Standard*). В настоящее время в большинстве спецификаций защищенных сетевых протоколов определяют в качестве стандартного метода криптографической защиты тройное шифрование DES (3DES), а в США принят соответствующий стандарт TDEA (*Triple Data Encryption Algorithm*, NIST SP 800-67 Rev. 2). Вместе с тем на практике при возможности предпочтение отдается более эффективным алгоритмам, в том числе новому американскому стандарту AES.

Алгоритм DES. Алгоритм DES, используя комбинацию ряда подстановок и перестановок, осуществляет шифрование 64-битовых блоков данных с помощью 56-битового ключа *k.* Схема алгоритма DES изображена на рис. 2.6. Процесс шифрования состоит в начальной перестановке битов входного блока, шестнадцати раундов шифрования и, наконец, конечной перестановке битов. Начальная перестановка и конечная перестановки не влияют на безопасность DES. Однако все таблицы перестановок и замен жестко закреплены стандартом и должны использоваться при реализации алгоритма DES в неизменном виде.

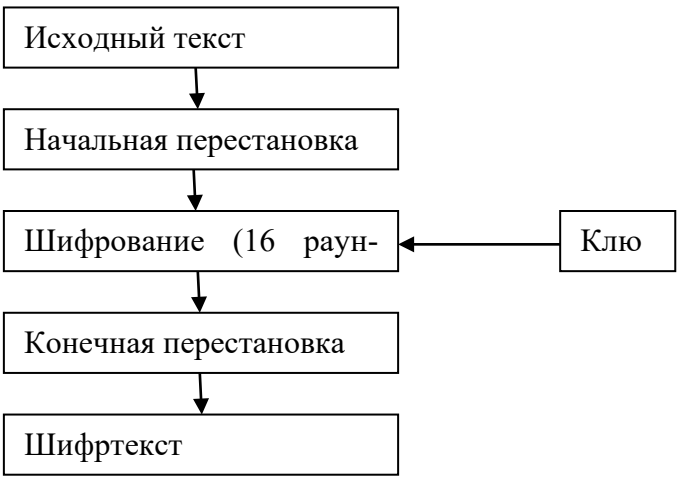

Рис. 2.6. Общая схема алгоритма DES

В приводимом ниже описании алгоритма DES использованы следующие обозначения:

L_i и R_i — левая и правая половины 64-битного блока L_iR_i;

$\oplus$ — операция побитового сложения векторов-блоков по модулю 2 (побитовое исключающее ИЛИ);

k_i — 48-битовые раундовые ключи;

f — функция шифрования;

IP — начальная перестановка степени 64;

IP^{-1} — конечная перестановка.

В ходе шифрования очередного блока T его биты подвергаются *начальной перестановке IP* согласно табл. 2.1. При этом бит 58 блока T становится битом 1, бит 50 — битом 2 и т. д.

Таблица 2.1

Начальная перестановка 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

Полученный после перестановки блок $IP(T)$ разделяется на две половины: L_0, состоящую из 32 старших бит, и R_0, состоящую из 32 младших бит. Затем выполняется итеративный процесс шифрования, состоящий из 16 раундов преобразования Фейстеля по формулам (2.1). Аргументами раундовой функции шифрования f являются 32-битовый вектор R_{i-1} и 48-битовый ключ k_i, который является результатом преобразования 56-битового ключа шифра k. По окончании шифрования осуществляется восстановление позиций битов применением обратной перестановки — IP^{-1}.

При дешифровании данных на вход алгоритма DES подается блок криптограммы, а раундовые ключи используются в обратном порядке, при этом вместо соотношений (2.1) следует пользоваться соотношениями

$$\begin{cases} R_i = L_{i-1}, \\ R_i = L_{i-1} \oplus f(R_{i-1}, k_{16+1-i}), i = 1,\ldots,16. \end{cases}$$

Схема вычисления значения функции шифрования *f* изображена на рис 2.7.

Для вычисления значения функции *f* используются: *расширение E*; нелинейная замена *S*, составленная из восьми преобразований *S-блоков*: S_1, S_2, …, S_8; перестановка *P*.

Перестановка с расширением *E* позволяет получить из входного 32-битного подблока данных R_{i-1} 48-битный вектор $E(R_{i-1})$ путем дублирования некоторых битов R_{i-1}, при этом порядок следования битов в $E(R_{i-1})$ указан в табл. 2.2. Такое преобразование необходимо, чтобы сравнять длины преобразуемого подблока входного текста (изначально 32 бита) и раундового ключа (48 бит) для последующего их сложения с помощью операции побитового XOR. Дублирование отдельных битов входного блока позволяет ускорить эффекты рассеивания.

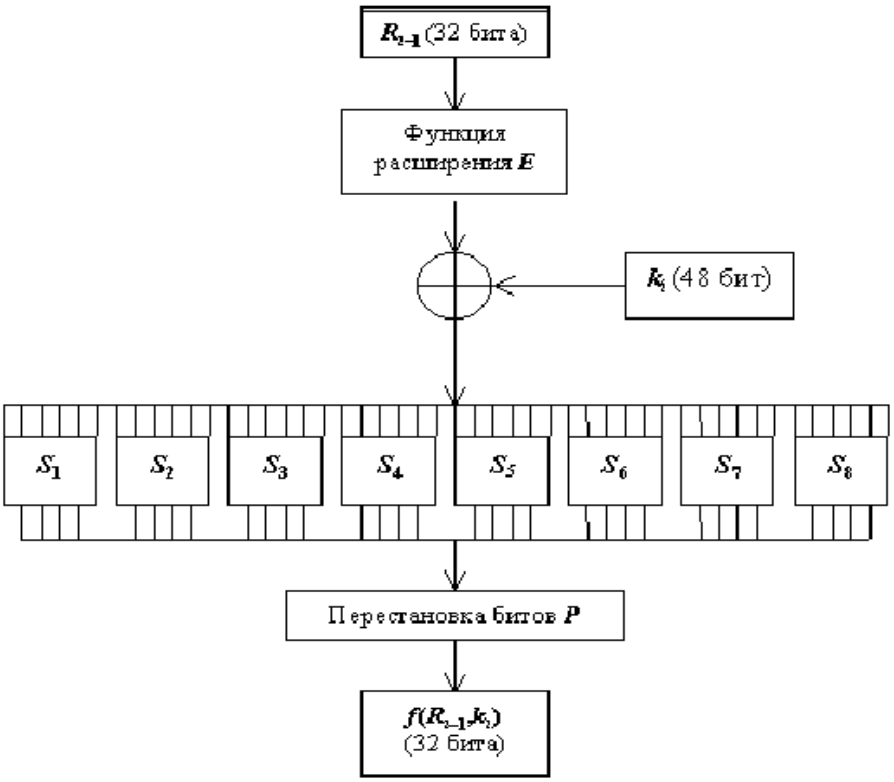

Рис. 2.7. Схема раундовой функции шифрования f алгоритма DES

Таблица 2.2

Таблица перестановки с расширением $E(R_{i-1})$

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

Полученный результат складывается побитно по модулю 2 с текущим значением ключа k_i и затем представляется в виде восьми последовательных 6-битовых блоков B_1, ..., B_8. Далее каждый из блоков B_j трансформируется в 4-битовый блок B'_j с помощью соответствующей таблицы нелинейной замены *S* — блока S_j. Таблицы замен определены стандартом и представляют собой прямоугольные матрицы размером 4 × 16 элементов. Элементы этой матрицы — целые числа от 0 до 15 (табл. 2.3).

Таблица 2.3

Пример S-блока алгоритма DES — таблица замен S_1

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

Преобразование 6-битового блока B_j в 4-битовый блок B'_j осуществляется следующим образом. Пусть, например, B_1 равен 101011. Преобразование B_1 производится с помощью таблицы замен S_1 (табл. 2.3). Первый и последний разряды B_1 являются двоичной записью числа a, $0 \leq a \leq 3$. В рассматриваемом примере двоичное представление a состоит из битов 11, $a = 3$. Аналогично средние 4 разряда представляют число b, $0 \leq b \leq 15$. В рассматриваемом примере *двоичное представление b* состоит из битов 0101, $b = 5$. Строки и столбцы ы S_1 пронумерованы числами a и b. Таким образом, пара (a,b) однозначно определяет выход S-блока — число, находящееся на пересечении строки с номером a и столбца с номером b. В данном случае это число равно 9. Записывая его в двоичной форме, получаем выход B'_1, равный 1001. Выходы S-блоков объединяются в 32-битовый блок $B'_1, B'_2, \ldots, B'_8$, к которому применяется перестановка битов P, заданная табл. 2.4.

Таблица 2.4

Таблица перестановки P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

48-битовые раундовые ключи k_i получаются из исходного ключа k следующей процедурой расширения ключа. Каждый восьмой бит исходного 64-битного ключа k алгоритма DES являются контрольными битами четности, используемые для обнаружения ошибок при обмене и хранении ключей. Начальная перестановка (табл. 2.5) процедуры расширения ключа позволяет выделить 56 случайных значащих битов ключа и разделить их на два блока C_0 и D_0 по 28 бит в каждый (верхняя и нижняя половины таблицы).

Таблица 2.5

Таблица перестановки 56 значащих бит ключа

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

Затем последовательно определяются блоки C_i и D_i, $i = 1, \ldots, 16$ следующим образом: если уже определены C_{i-1} и D_{i-1}, то C_i, и D_i получаются их циклическим сдвигом влево на число позиций, заданных табл. 2.6.

Таблица 2.6

Таблица циклических сдвигов процедуры расширения ключа

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

Ключ k_i, $1 \leq i \leq 16$, выбирается из битов блока $C_i D_i$ согласно табл. 2.7.

Таблица 2.7

Таблица формирования раундового ключа k_i

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

В каждом раундовом ключе используется уникальный набор битов исходного ключа шифрования, при этом каждый бит ключа k используется в среднем в 14 раундовых ключей k_i. Важной особенностью процедура расширения ключа алгоритма DES является то, что раундовые ключи могут формироваться «на лету» то есть параллельно с выполнением раундов шифрования DES. Это позволяет сократить время и объемы хранения ключевой информации в памяти шифрующего устройства. При расшифровании процедуру расширения ключа также можно использовать «на лету», выполняя вместо циклического сдвига влево сдвиг вправо на то же число позиций (кроме 1 раунда, в котором сдвиг не производится).

Известные аналитические методы вскрытия DES не дают существенного выигрыша по сравнению с полным перебором всего множества из 2^{56} ключей. Однако по сегодняшним меркам ключевое пространство DES невелико, что делает полный перебор ключей практически осуществимым. Кроме того, DES имеет слабые ключи, использование которых существенно облегчают криптоанализ алгоритма (с помощью слайдовой атаки). Число слабых ключей алгоритма DES невелико — всего 64, они известны и могут быть отбракованы на этапе выбора ключа шифрования.

Попытка увеличить мощность ключевого пространства алгоритма DES без смены самого алгоритма привела к возникновению схем двойного и тройного шифрования. Как оказалось, для двойных алгоритмов с независимыми подключами можно реализовать криптографическую атаку методом «встреча посередине», в результате вычислительная стойкость двойного шифра DES ненамного превосходит стойкость исходного алгоритма. В настоящее время специфицирован алгоритм тройного DES (3DES, TDEA) (рис. 2.7).

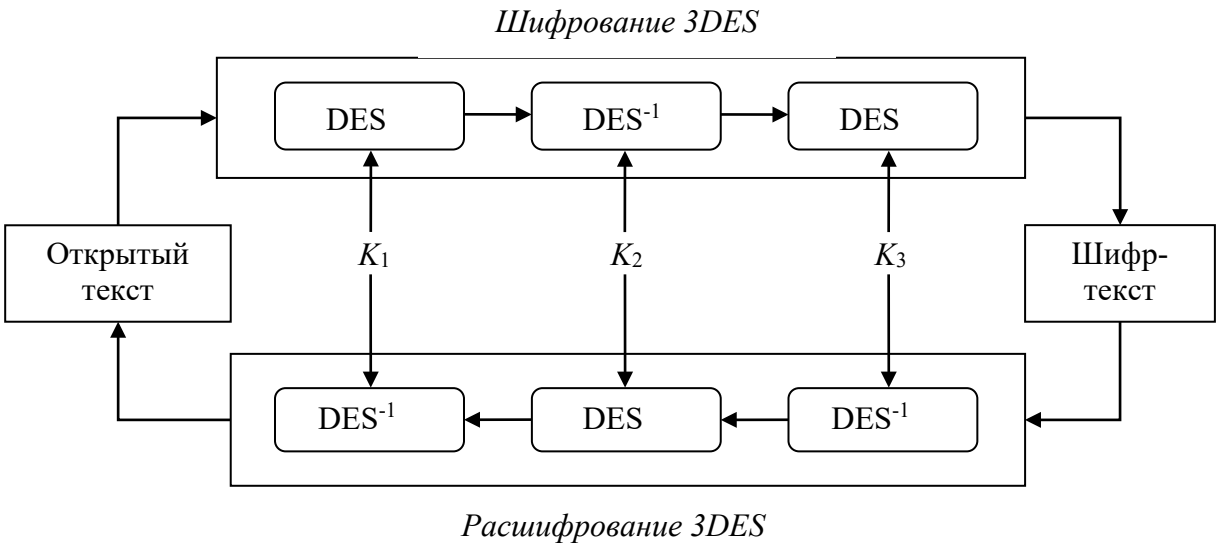

Рис. 2.7. Алгоритм тройного DES с тремя независимыми подключами

Алгоритм 3DES также подвержен атаке «встреча посередине», однако его стойкость сопоставима с алгоритмами с длиной ключа 112 бит. При этом число раундов 3DES составляет 48, что делает этот шифр достаточно медленным. С принятием новых более эффективных стандартов шифрования в большинстве практических реализаций предпочтение отдается именно им.

Стандарт шифрования AES. AES — новый американский стандарт блочного шифра (*Advanced Encryption Standard*), принятый в 2000 году (NIST FIPS — 197). В основу стандарта AES лег алгоритм Rijndael, выбранный на конкурсной основе. Алгоритм AES относят к SP-сетям или алгоритма со структурой «квадрат», поскольку на вход алгоритма подается 128-битовый блок открытого текста, представляемый в виде таблицы байтов. Алгоритм может иметь размер ключа 128, 196 или 256 бит, используемый размер ключа определяет число раундов алгоритма AES: 10, 12 или 14 соответственно. Для указания того, какой именно вариант алгоритма использован при шифровании, размер ключа записывается следом за названием шифрующего алгоритма, например, AES-128, AES-256. Входной 128-битовый блок алгоритма AES представляется в виде двумерного массива байтов размером 4 × 4. Операции раундовой функции шифрования, за исключением операции побитового XOR, используемого для наложения материала раундового ключа, оперируют отдельными байтами, либо строками или столбцами таблицы байтов.

Перед началом шифрования производится наложение исходного ключа шифрования *K* на открытый текст. Структура раунда AES представлена на рис. 2.8. Последний раунд алгоритма AES — сокращенный, в нем опущена операция MixColumns.

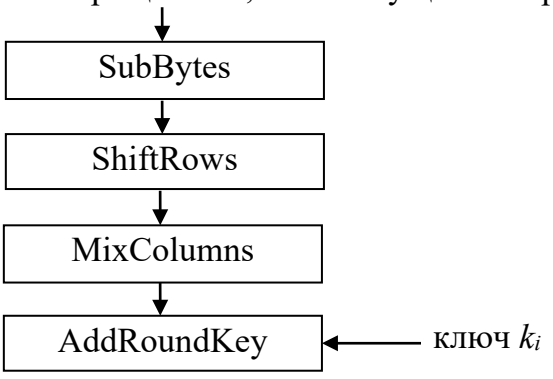

Рис. 2.8. Раунд шифрования алгоритма AES

Операция *SubBytes* представляет собой побайтную нелинейную табличную замену (табл. 2.8). Каждый байт входного блока преобразовывается с помощью одной и той же таблицы замен следующим образом. Байты представляются двухразрядными шестнадцатеричными числами. Первый шестнадцатеричный разряд определяет строку, а второй — столбец таблицы замен, ячейка на пересечении определенных таким образом строки и столба содержит выходное значение байта. Затем следует слой линейного перемешивания, реализуемый операциями *ShiftRows* и *MixColumns*. Операция *ShiftRows* выполняет циклический сдвиг влево последних трех строк таблицы байтов на 1, 2 и 3 байта соответственно. Операция *MixColumns* использует вычисления над элементами поля Галуа $GF(2^8)$, опирающееся на соответствующее представление байтов данных в виде полиномов 7 степени либо 8-разрядных двоичных чисел. Операция *MixColumns* производится над каждым из столбцов массива данных.

Таблица 2.8

Таблица нелинейной замены AES

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Элементы a_k каждого столбца таблицы входных данных рассматриваются как коэффициенты полинома третьей степени, новые значения b_k столбца байтов могут быть вычислены по формуле:

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix},$$

или

$$b_1 = 02 \bullet a_1 + 03 \bullet a_2 + a_3 + a_4$$
$$b_2 = a_1 + 02 \bullet a_2 + 03 \bullet a_3 + a_4$$
$$b_3 = a_1 + a_2 + 02 \bullet a_3 + 03 \bullet a_4$$
$$b_4 = 03 \bullet a_1 + a_2 + a_3 + 02 \bullet a_4,$$

где «02», «03» — шестнадцатеричные числа, a_k — шестнадцатеричные числа, представленные как элементы поля $GF(2^8)$, операция • — умножение над полем Галуа $GF(2^8)$ с неприводимым полиномом $m(x) = x^8 + x^4 + x^3 + x + 1$.

Каждый байт может быть представлен восьмиразрядным двоичным числом, которому в поле Галуа $GF(2^8)$ соответствует полином 7 степени, чьи коэффициенты (0 или 1) определяются значениями соответствующих битов. Операция сложения в этом случае понимается как побитовый XOR коэффициентов при одинаковых степенях полинома, а умножение — как умножение полиномов с последующим сложением по правилам поля Галуа $GF(2^8)$. В случае, если в результате операции получен полином степени, превышающей 7, берется остаток от деления этого результата нацело на $m(x)$. Таким образом, результат операции всегда имеет эквивалентное представление в виде байта.

Завершает раунд операция *AddRoundKey*, выполняющая наложение на массив данных материала раундового ключа с помощью побитового исключающего ИЛИ.

Как и DES, AES генерирует раундовые ключи на основе исходного ключа шифрования, однако механизм генерации иной — с использованием побитового сложения с константами раунда, циклического сдвига и определенной в AES нелинейной замены. Расширение ключа может быть выполнено «на лету».

Как и для всех SP-сетей, раунд расшифрования AES существенно отличается от раунда шифрования. При расшифровании используются обратные операции *InvShiftRows*, *InvSubBytes*, *AddRoundKey*, *InvMixColumns*. Операция *AddRoundKey* в силу свойств побитового XOR обратна сама себе и заключается в сложении с раундовыми ключами, указанными в обратном порядке. Остальные обратные операции могут быть представлены как исходные с точностью до параметров преобразования (например, в *InvSubBytes* используется другая таблица замен, а в *InvShiftRows* — другое направление сдвига). Существенным, однако, является тот факт, что в раунде алгоритма расшифрования изменен порядок следования этих операций. Однако возможна определенная модификация — процедура «прямого расшифрования», когда используется тот же порядок и набор операций раунда с точностью до параметров преобразований (таблицы замен, величины сдвига и т. д.), а раундовые ключи (в обратном порядке), предварительно пропущены через операцию *MixColumns*.

Алгоритм AES не имеет слабых ключей, достаточно эффективен на разных платформах и отвечает современным требованиям к вычислительной стойкости шифров. Использование 128-битовых алгоритмов против 64-битовых (как DES) позволяет повысить производительность шифрования при обработке данных большого объема, а также избежать частой смены ключей с целью предотвращения возможных комбинаторных атак.

2.2.3. Российский стандарт блочного шифрования ГОСТ 34.12–2018

Отечественные криптографические стандарты обязательны для реализации в сертифицированных средствах криптографической защиты информации (СКЗИ), применяемых в государственных информационных системах и, в некоторых случаях, в коммерческих системах (например, при защите персональных данных). Сертификация СКЗИ требуется для защиты сведений, составляющих государственную тайну Российской Федерации, и иной информации, конфиденциальность которой требуется обеспечить согласно действующему законодательству. Применение отечественных криптографических стандартов рекомендовано и в банковских системах Российской Федерации.

Более 25 лет в России действовал стандарт симметричного шифрования ГОСТ 28147–89 «Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования». Стандарт определял построенный на схеме Фейстеля

симметричный блочный шифр с длиной блока 64 бита и 256-битный ключом, а также режимы его работы. Однако в 2015 году были приняты новые криптографические стандарты ГОСТ Р 34.12–2015[10] и ГОСТ Р 34.13–2015[11], вступившие в действие с января 2016 года. Первый из этих стандартов определял два блочных шифра: 64-битовый, основанный на сети Фейстеля («Магма») и 128-битовый, сходный по своему строению с шифром AES («Кузнечик»). Шифр «Магма» по сути является старым алгоритмом ГОСТ 28147–89, для которого определены таблицы замен, что позволило учесть ряд критических замечаний в адрес старого стандарта. Кроме того, режимы работы блочных шифров вынесены в отдельный стандарт (ГОСТ Р 34.13–2015) и не привязаны к особенностям строения какого-либо шифра, то есть являются универсальными.

В 2018 году российские криптографические стандарты, включая ГОСТ Р 34.12–2015 и ГОСТ Р 34.13–2015, были приняты группой государств, в которую вошли Россия, Армения, Киргизия, Таджикистан и Туркмения, в качестве межгосударственных стандартов и получили обозначение ГОСТ 34.12–2018 и ГОСТ 34.13–2018 соответственно и действуют с июня 2019 года. Рассмотрим подробнее алгоритмы межгосударственного стандарта ГОСТ 34.12–2018 «Информационная технология. Криптографическая защита информации. Блочные шифры».

Алгоритм 34.12–2018 «Магма» (рис. 2.9) построен на схеме Фейстеля, имеет размер блока 64 бит, 256-битный ключ и выполняет 32 раунда шифрования.

В каждом раунде алгоритма выполняются следующие преобразования:

1. *Наложение раундового ключа*. Содержание подблока R_i складывается по модулю 2^{32} с ключом раунда k_j. k_j — это 32-битовая часть исходного ключа, используемая в качестве раундового.

Алгоритм ГОСТ не использует процедуру расширения ключа, вместо этого исходный 256-битный ключ шифрования на восемь 32-битовых подключей: k_0, k_1, k_2, k_3, k_4, k_5, k_6, k_7. В процессе шифрования подключи k_j используются в соответствии со следующим расписанием:

— с 1 по 24 раунд — в прямой последовательности: k_0, k_1, k_2, k_3, k_4, k_5, k_6, k_7, ...

— с 25 по 32 раунд — в обратной последовательности: k_7, k_6, k_5, k_4, k_3, k_2, k_1, k_0.

2. *Нелинейная замена*. После наложения ключа 32-битовый блок разбивается на 8 частей по 4 бита, значение каждой из которых по отдельности заменяется в соответствии с одной из 8 таблиц замен (S-блоком). Каждый S-блок алгоритма «Магма» представляет собой вектор (одномерный массив) с 16 элементами, пронумерованными числами от 0 до 15. Из таблицы S-блока извлекается элемент, порядковый номер которого совпадает с входным значением подстановки. Значениями S-блока также являются 4-битовые значения (целые числа от 0 до 15).

---

[10] Национальный стандарт Российской Федерации ГОСТ Р 34.12-2015 «Информационная технология. Криптографическая защита информации. Блочные шифры» (утв. приказом Федерального агентства по техническому регулированию и метрологии от 19.06.2015 № 749-ст) // Электронный фонд правовых и нормативно-технических документов. URL: https://docs.cntd.ru/document/ 1200161708 (дата обращения: 12.01.2023).

[11] Национальный стандарт Российской Федерации ГОСТ Р 34.13–2015 «Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров» (утв. приказом Федерального агентства по техническому регулированию и метрологии от 19.06.2015 № 750-ст) // Электронный фонд правовых и нормативно-технических документов. URL: https://docs.cntd.ru/document/ 1200121984 (дата обращения: 12.01.2023).

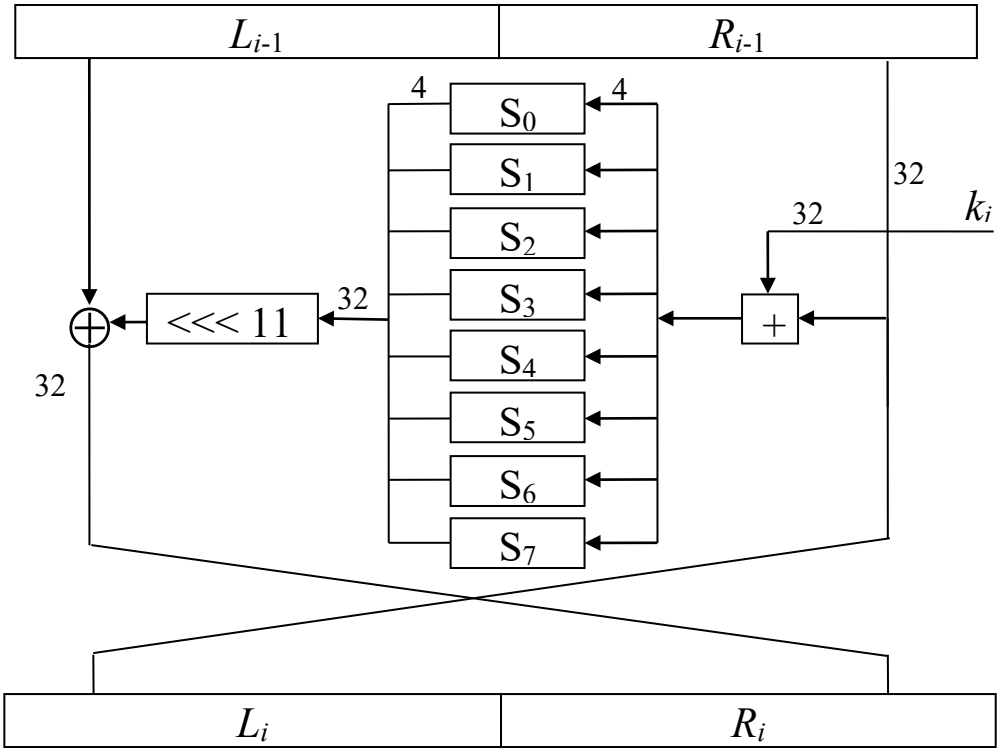

Рис. 2.9. Схема раунда алгоритма «Магма»

Старым стандартом ГОСТ 28147–89 значения S-блоков не закреплялись. Предполагалось, что таблицы замен являются общими для всех узлов шифрования в рамках одной системы криптографической защиты. При этом стойкость алгоритма существенно зависит от правильного выбора S-блоков, очевидно имеются «слабые» таблицы замен, существенно облегчающие криптоанализ алгоритма (например, тривиальная таблица, в которой вход равен выходу). Для алгоритма «Магма» блоки замен были закреплены стандартом.

S_0 = (12, 4, 6, 2, 10, 5, 11, 9, 14, 8, 13, 7, 0, 3, 15, 1);
S_1 = (6, 8, 2, 3, 9, 10, 5, 12, 1, 14, 4, 7, 11, 13, 0, 15);
S_2 = (11, 3, 5, 8, 2, 15, 10, 13, 14, 1, 7, 4, 12, 9, 6, 0);
S_3 = (12, 8, 2, 1, 13, 4, 15, 6, 7, 0, 10, 5, 3, 14, 9, 11);
S_4 = (7, 15, 5, 10, 8, 1, 6, 13, 0, 9, 3, 14, 11, 4, 2, 12);
S_5 = (5, 13, 15, 6, 9, 2, 12, 10, 11, 7, 8, 1, 4, 3, 14, 0);
S_6 = (8, 14, 2, 5, 6, 9, 1, 12, 15, 4, 11, 0, 13, 10, 3, 7);
S_7 = (1, 7, 14, 13, 0, 5, 8, 3, 4, 15, 10, 6, 9, 12, 11, 2).

Приведенный набор S-блоков обладает хорошей степенью нелинейности и сформирован с помощью математического аппарата. Это позволяет обеспечить стойкость алгоритма «Магма» к дифференциальному и линейному криптоанализу.

3. *Циклический сдвиг* битов обрабатываемого подблока влево на 11 позиций выполняет роль перестановки.

Расшифрование осуществляется по этой же схеме, но с другим расписанием использования ключей:

— с 1 по 8 раунд — в прямой последовательности: k_0, k_1, k_2, k_3, k_4, k_5, k_6, k_7.

— с 9 по 32 раунд — в обратном порядке: k_7, k_6, k_5, k_4, k_3, k_2, k_1, k_0, …

Как видно, операции раунда алгоритма достаточно просты. Это позволяет алгоритму «Магма» обеспечивать приемлемую скорость работы даже при достаточно большом числе раундов (32 раунда).

За время существования стандарта ГОСТ 28147–89 не было предложено практически осуществимых атак на алгоритм, что определяется большой длиной ключа и большим числом раундов шифрования. Хотя и были предложены криптоаналитические атаки на полнораундовый алгоритм ГОСТ, снижающие оценки его стойкости (лучшая — до 2^{192}) по сравнению с полным перебором ключей, они далеки от возможности практической реализации.

Наряду с неопределенностью таблиц замен, стандарт ГОСТ 28147–89 подвергался критике из-за слабых ключей. Очевидно, алгоритм имеет слабые ключи, например, те, при которых все восемь чередующихся раундовых ключей окажутся одинаковыми. Однако механизмы отсева слабых ключей в стандарте не приведены.

Несмотря на определенную критику, алгоритм «Магма» был включен в новый криптографический стандарт на блочные шифры, что обусловлено как необходимостью иметь шифры с различной длиной блока, так и приемлемыми эксплуатационными характеристиками (криптографической стойкостью и скоростью работы) шифра. Следует также отметить, что простая структура раунда делает эффективными реализации алгоритма «Магма» для низкоресурсных устройств (таких как криптографические смарт-карты и токены).

Алгоритм 34.12–2018 «Кузнечик» — это 128-битный шифр со структурой «подстановочно-перестановочная сеть», имеющий ключ длиной 256 бит и выполняющий 10 раундов шифрования. Каждый раунд состоит из трех слоев: наложения раундового ключа с помощью операции побитового XOR (X), нелинейной замены (S) и линейного перемешивания L (рис. 2.10). Последний раунд является усеченным — в нем производится только наложение ключа.

Таким образом, шифрование 128-битного входного блока a описывается формулой $E(a) = X[K_{10}]LSX[K_9] \ldots LSX[K_2]LSX[K_1](a)$, где K_i — раундовые ключи, а $LSX[k]$ — последовательное применение операций побитового XOR с ключом k, нелинейной замены S и линейного перемешивания L.

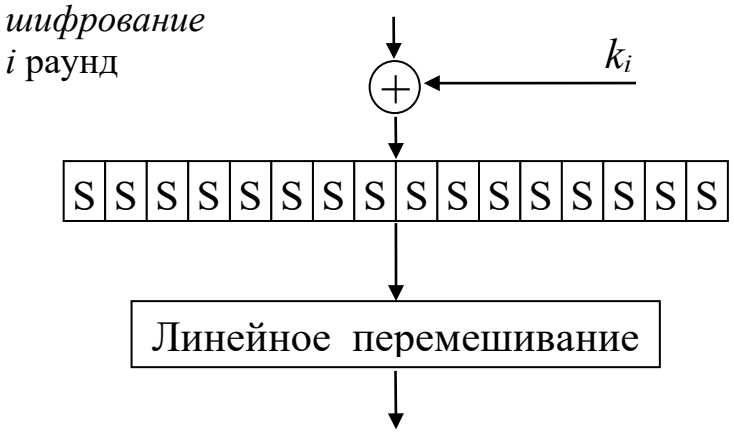

Рис. 2.10. Раунд шифрования алгоритма «Кузнечик»

Первой выполняется операция наложения ключа раунда с помощью побитового XOR: $X[k](a) = k \oplus a$.

Затем входной 128-битовый блок a алгоритма шифрования представляется в виде последовательности 16 байтов, которые нумеруются справа налево, начиная с 0: $a = a_{15} \| a_{14} \| \ldots \| a_1 \| a_0$, где значком || обозначена операция конкатенации строк.

К каждому байту применяется нелинейная подстановка, задаваемая 256-значным массивом S = (252, 238, 221, 17, 207, 110, 49, 22, 251, 196, 250, 218, 35, 197, 4, 77, 233, 119, 240, 219, 147, 46, 153, 186, 23, 54, 241. 187, 20, 205, 95, 193, 249, 24, 101, 90, 226,

92, 239, 33, 129, 28, 60, 66, 139, 1, 142, 79, 5, 132, 2, 174, 227, 106, 143, 160, 6, 11, 237, 152, 127, 212, 211, 31, 235, 52, 44, 81, 234, 200, 72, 171, 242, 42, 104, 162, 253, 58, 206, 204, 181, 112, 14, 86, 8, 12, 118, 18, 191, 114, 19, 71, 156, 183, 93, 135, 21, 161, 150, 41, 16, 123, 154, 199, 243, 145, 120, 111, 157, 158, 178, 177, 50, 117, 25, 61, 255, 53, 138, 126, 109, 84, 198, 128, 195, 189, 13, 87, 223, 245, 36, 169, 62, 168, 67, 201, 215, 121, 214, 246, 124, 34, 185, 3, 224, 15, 236, 222, 122, 148, 176, 188, 220, 232, 40, 80, 78, 51, 10, 74, 167, 151, 96, 115, 30, 0, 98, 68, 26, 184, 56, 130, 100, 159, 38, 65, 173, 69, 70, 146, 39, 94, 85, 47, 140, 163, 165, 125, 105, 213, 149, 59, 7, 88, 179, 64, 134, 172, 29, 247, 48, 55, 107, 228, 136, 217, 231, 137, 225, 27, 131, 73, 76, 63, 248, 254, 141, 83, 170, 144, 202, 216, 133, 97, 32, 113, 103, 164, 45, 43, 9, 91, 203, 155, 37, 208, 190, 229, 108, 82, 89, 166, 116, 210, 230, 244, 180, 192, 209, 102, 175, 194, 57, 75, 99, 182).

Поскольку каждый байт a_i может принимать значение от 0 до 255, массив S имеет 256 элементов. Каждый байт блока данных заменяется на элемент массива, порядковый номер которого совпадает с входным значением. Это значит, что при $a_i = 0$ значение a_i будет заменено на $S(0) = 252$, при $a_i = 1$ — на $S(1) = 238$ и т. д. (все числа записаны в десятеричном виде). Для входного блока в целом нелинейная подстановка может быть определена как

$$S(a) = S(a_{15} \,||\, \ldots \,||\, a_0) = S(a_{15}) \,||\, \ldots \,||\, S(a_0).$$

Линейное преобразование алгоритма «Кузнечик» использует операции над полем Галуа $GF(2^8)$. Операция линейного перемешивания L может быть описана с помощью линейного регистра сдвига R:

1. Байты a_i представляются в виде полиномов поля Галуа $GF(2^8)$. Для простоты обозначим их также a_i.

2. Вычисляется $\ell(a_{15}, \ldots, a_0) = 148 \bullet a_{15} + 32 \bullet a_{14} + 133 \bullet a_{13} + 16 \bullet a_{12} + 194 \bullet a_{11} + 192 \bullet a_{10} + 1 \bullet a_9 + 251 \bullet a_8 + 1 \bullet a_7 + 192 \bullet a_6 + 194 \bullet a_5 + 16 \bullet a_4 + 133 \bullet a_3 + 32 \bullet a_2 + 148 \bullet a_1 + 1 \bullet a_0$, где + и • — операции сложения и умножения над полем Галуа $GF(2^8)$ с неприводимым многочленом $m(x) = x^8 + x^7 + x^6 + x + 1$ (вычисления проводятся по модулю $m(x)$). Байты $a_i \in GF(2^8)$, а также числовые коэффициенты выражения ℓ, записанные в десятеричном виде, могут быть представлены восьмиразрядными двоичными числами, которым соответствуют полиномы 7 степени над $GF(2^8)$.

Результат вычисления ℓ — полином 7 степени, который может быть переведен обратно в 8 двоичных разрядов и представлен в виде байта.

3. Полученный на предыдущем шаге результат записывается первым в строку байтов, затем записываются все байты входной строки a, кроме младшего (последнего). Таким образом, фактически производится (нециклический) сдвиг строки байтов a вправо на 1 позицию:

$R(a) = R(a_{15} \,||\, \ldots \,||\, a_0) = \ell(a_{15}, \ldots, a_0) \,||\, a_{15} \,||\, \ldots \,||\, a_1$,

4. Шаги 1–3 повторяются 16 раз, при этом входом каждого следующего шага является выход предыдущего:

$$L(a) = R^{16}(a).$$

В результате на выходе не останется ни одного не преобразованного байта исходного блока.

В ходе шифрования используются 10 128-битовых раундовых ключей. При этом первые два раундовых ключа получаются простым делением исходного ключа на две части. Таким образом, процедура расширения ключа должна сформировать 8 раундовых ключей, которые формируются парами, то есть 4 пары 128-битовых ключей. В качестве процедуры расширения ключа используется 8-раундовый алгоритм на сети

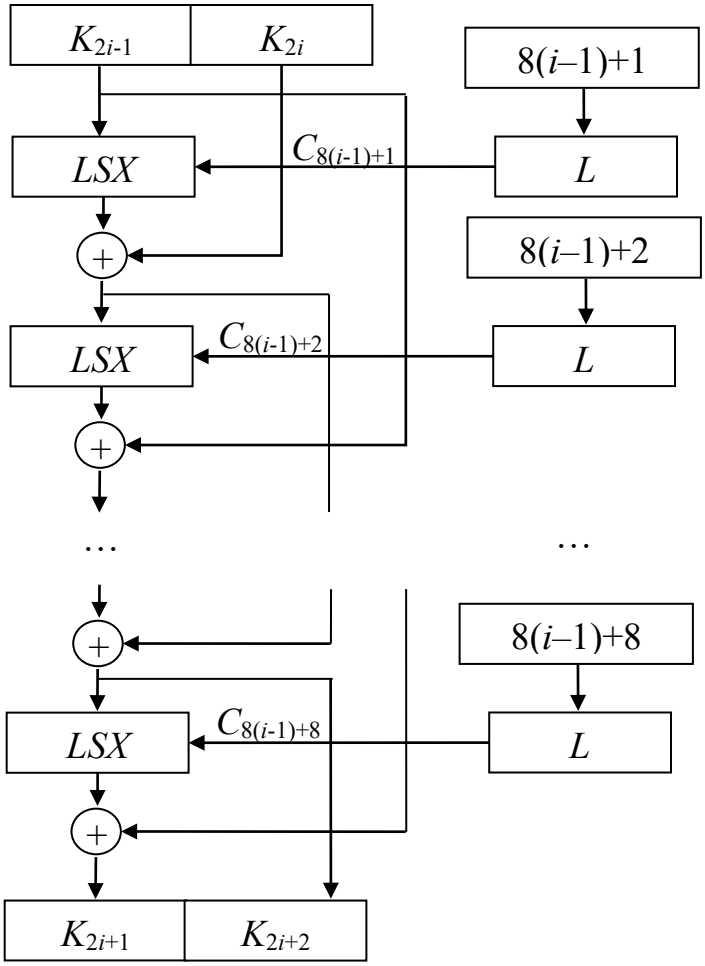

Рис. 2.11. Процедура расширения ключа алгоритма «Кузнечик» (получение пары раундовых ключей на основе предыдущей)

Фейстеля, раундовая функция которого совпадает с раундовой функцией основного алгоритма шифра «Кузнечик», а раундовыми ключами C_i служат константы, пропущенные через линейное преобразование L (рис. 2.11):

$$(K_{2i+1}, K_{2i+2}) = F[C_{8(i-1)+8}] \dots F[C_{8(i-1)+1}](K_{2i-1}, K_{2i}),\ i = 1, 2, 3, 4,$$

где $F[k](A_1, A_0) = (LSX[k](A_1) \oplus A_0, A_1)$;

$C_i = L(i)$, $i = 1, 2, \dots, 32$, значение i представлено в виде 128-битовой строки.

Алгоритм расшифрования записывается как

$$D(a) = X[K_1]S^{-1}L^{-1}X[K_2]\dots S^{-1}L^{-1}X[K_9]S^{-1}L^{-1}X[K_{10}](a).$$

Как видно, раундовые ключи используются в обратном порядке, а порядок операций раунда отличен от раунда шифрования. Здесь S^{-1} — нелинейная замена по матрице, обратной блоку замен шифрования, матрица S^{-1} легко может быть вычислена на основе S. Операция L^{-1} может быть задана как $L^{-1}(a) = (R^{-1})^{16}(a)$. В свою очередь, $R^{-1}(a) = R^{-1}(a_{15}\| \dots \|a_0) = a_{14} \| a_{13} \| \dots \| a_0 \| \ell(a_{14}, a_{13}, \dots, a_0, a_{15})$. В настоящее время неизвестны какие-либо атаки на полнораундовый алгоритм «Кузнечик». Появились работы относительно атак на алгоритм с усеченным числом раундов (3, 5), однако их эффективность невысока.

2.2.4. Режимы работы блочных шифров ГОСТ 34.13–2018

Режимы работы блочных шифров выведены в отдельный стандарт ГОСТ Р 34.13–2018 «Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров», что соответствует принятой международной практике. Эти режимы универсальны, то есть не привязаны к конкретным шифрам и применимы как к шифру «Магма», так и к шифру «Кузнечик», а также к любому другому блочному симметричному шифру.

Режим работы блочного шифра показывает, каким образом обрабатывать с помощью алгоритма шифрования различные блоки одного и того же открытого текста. Как правило, при этом предполагается, что для обработки всех блоков открытого текста используется один и тот же криптографических ключ (за исключением режимов со сменой ключа).

Стандартом ГОСТ 34.13–2018 определено 6 режимов работы алгоритмов блочного шифрования: режим простой замены (Electronic Codebook, ECB); режим гаммирования (Counter, CTR); режим гаммирования с обратной связью по выходу (Output Feedback, OFB); режим простой замены с зацеплением (Cipher Block Chaining, CBC); режим гаммирования с обратной связью по шифртексту (Cipher Feedback, CFB); режим выработки имитовставки (Message Authentication Code algorithm, MAC). Каждый из этих режимов имеет свое назначение.

Режим простой замены наиболее прост. В нем сообщение делится на равные блоки (длина каждого блока совпадает с длиной n входного блока шифра), каждый блок шифруется и расшифровывается независимо от других с использованием одного и того же ключа: $Y_i = E_k(X_i)$, E_k — шифрующее преобразование с ключом k, X_i и Y_i — блоки открытого текста и соответствующие им блоки зашифрованного текста. При этом очевидно, что блоки с одинаковым содержимым будут и зашифрованы одинаково, то есть соответствующие им криптограммы совпадут. Это может дать криптоаналитику противника дополнительную информацию об открытом тексте. В этом заключается основной недостаток данного режима, поэтому с помощью режима простой замены рекомендуется шифровать лишь короткие сообщения, длина которых не превышает размера блока алгоритма шифрования.

Вторая проблема — шифрование блоками одинакового размера, что может потребовать дополнения последнего блока открытого текста до полного размера n.

Стандарт 34.13–2018 определяет три *процедуры дополнения*, последняя из которых рекомендована к использованию только совместно с режимом выработки имитовставки (MAC).

1. Если последний блок открытого текста имеет недостаточную длину (то есть короче входного блока используемого алгоритма шифрования), то он дополняется до полной длины нулями. Если же последний блок открытого текста имеет полную длину, то никаких преобразований для него не производится. Эта процедура не всегда позволяет однозначно расшифровать открытый текст, если его длина неизвестна (это произойдет, например, если двоичное представление открытого текста оканчивается нулями).

2. Последний блок открытого текста всегда дополняется единицей, а затем нулями до полной длины. При использовании такой процедуры дополнения расшифрование сообщения всегда однозначно, однако в некоторых случаях придет зашифровать лишний блок неинформативных данных (например, если последний блок открытого текста имеет полную длину).

3. В случае использования режима выработки имитовставки можно дополнять последний неполный блок текста единицей и последующими нулями, а полный блок оставить без изменений.

Кроме режима простой замены, дополнение последнего блока сообщения необходимо в режимах простой замены с зацеплением и выработки имитовставки. Эти режимы работы блочных шифров, как и режим простой замены, обрабатывают сообщение блоками фиксированной длины.

В *режиме простой замены с зацеплением* перед шифрованием блока открытого текста на него накладывается полученный ранее шифртекст (рис. 2.12). Для начальных блоков для этих целей используется непредсказуемая (случайная или псевдослучайная) последовательность бит — синхропосылка (вектор инициализации в западной терминологии).

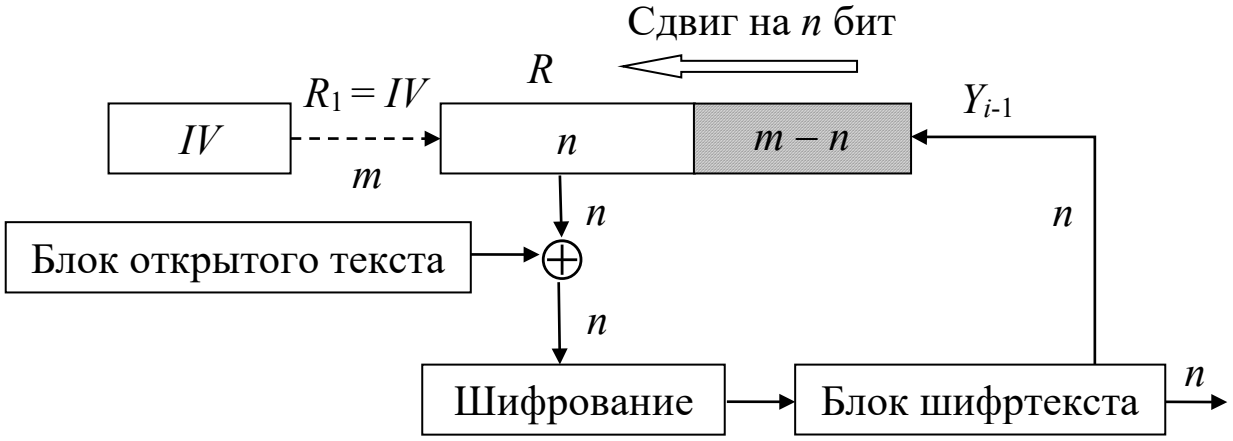

Рис. 2.12. Шифрование в режиме простой замены с зацеплением

Синхропосылка (*IV*) — комбинация знаков, передаваемая по каналу связи и предназначенная для инициализации алгоритма шифрования. Она используется при шифровании нескольких сообщений на одном и том же ключе (например, в рамках сеанса связи). Синхропосылка применяется во всех рассматриваемых в стандарте ГОСТ 34.13–2018 режимах работы блочных шифров, кроме режимов простой замены и выработки имитовставки. Обеспечение конфиденциальности синхропосылки не требуется.

Кроме того, большинство режимов работы блочных шифров используют сдвиговый регистр *R*, значение которого изменяется после каждого шага шифрования. Использованные на предыдущем шаге значения удаляются из регистра, на их место сдвигаются еще не использованные значения, а в конец дописываются новые.

При расшифровании содержимое сдвигового регистра накладывается на расшифрованный блок данных для получения исходного блока открытого текста (рис. 2.13).

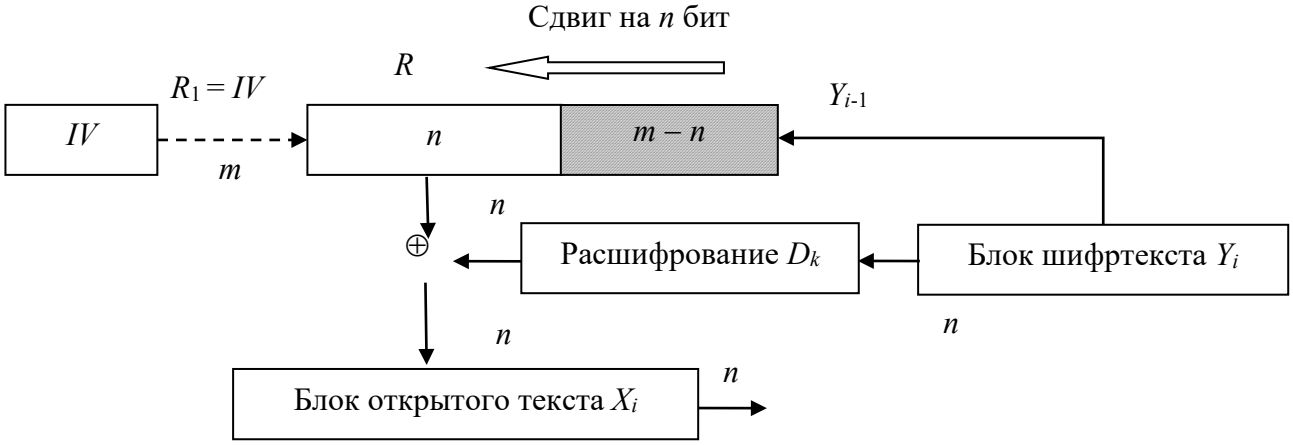

Рис. 2.13. Расшифрование в режиме простой замены с зацеплением

Режим простой замены с зацеплением (сцепления блоков шифртекста) более надежен по сравнению с режимом простой замены, поскольку здесь даже для одинаковых блоков открытого текста будут получены разные значения шифртекста. Этот режим может использоваться при шифровании сообщений большого объема.

Режимы гаммирования позволяют использовать блочный шифр в качестве генератора псевдослучайной гаммы, которая затем накладывается на открытый текст порциями произвольной длины s с помощью операции побитового XOR. Значение s может быть любым, не превосходящим длины n блока шифрующего алгоритма. Разница заключается лишь в том, какие данные заносятся в сдвиговый регистр. Например, в режиме гаммирования с обратной связью по выходу (OFB) в сдвиговый регистр заносится блок r_i преобразованных значений этого регистра (рис. 2.14). В режиме гаммирования с обратной связью по шифртексту (CFB) после каждого шага шифрования регистр сдвигается уже не на n, а на s битов, а в его конец заносится полученный на предыдущем шаге шифртекст Y_i. В режиме гаммирования (счетчика CTR) сдвиг регистра вообще не производится, его значение просто увеличивается на 1 после каждого шага шифрования.

Различаются и требования к синхропосылке. Значения синхропосылки для режима гаммирования с обратной связью по шифртексту выбирают случайно, равновероятно и независимо друг от друга из множества всех допустимых значений. В этом случае значение каждой используемой синхропосылки IV должно быть непредсказуемым (случайным или псевдослучайным). Для режима гаммирования (счетчика) значения синхропосылок должны быть уникальными (то есть не повторяться для сообщений, зашифрованных на одном ключе). А для режима гаммирования с обратной связью по выходу значение синхропосылки должно быть либо непредсказуемым (случайным или псевдослучайным), либо уникальным.

Режимы гаммирования фактически позволяют использовать блочный шифр для потоковой обработки данных. При расшифровании требуется сгенерировать с помощью блочного алгоритма ту же самую гамму и наложить ее на криптограмму.

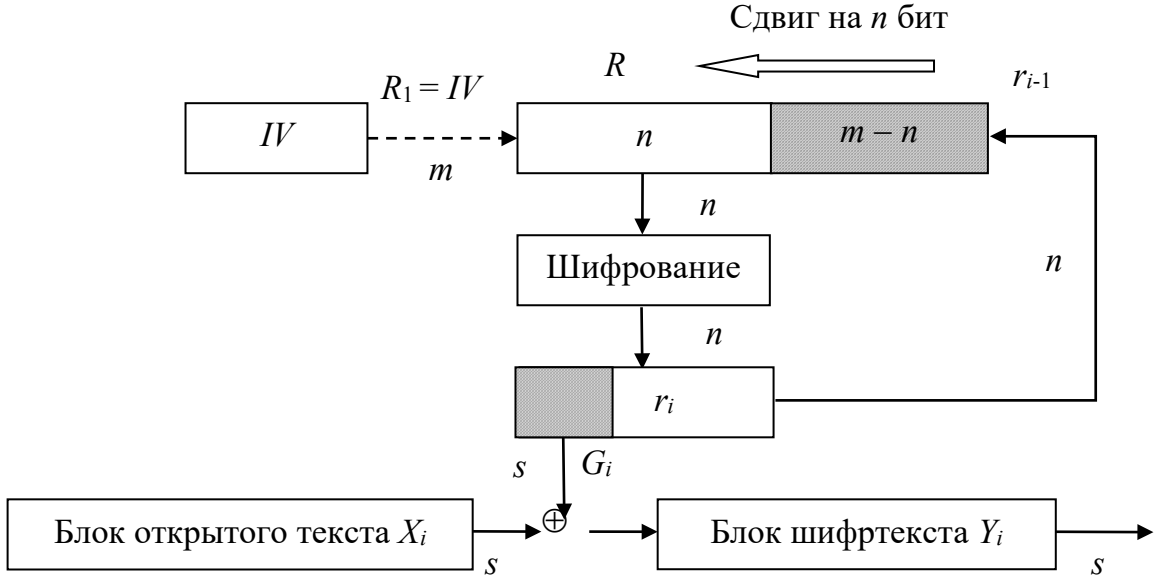

Рис. 2.14. Схема шифрования в режиме гаммирования с обратной связью по выходу (OFB)

Режим выработки имитовставки позволяет рассчитать зависящую от ключа криптографическую контрольную сумму (код аутентификации MAC — Message Authentication Code), которая используется для контроля целостности и аутентификации источника данных.

Режим выработки имитовставки преобразует данные полными блоками, выполняет сцепление блоков шифртекста, однако, в отличие от режима простой замены с зацеплением, здесь не предусматривается использование синхропосылки *IV* и сдвигового регистра *R*. На последнем шаге используется дополнительный ключ *K**, получаемый из основного ключа *K* (рис. 2.15). Значение *K** зависит от того, выполнялось ли дополнение последнего блока текста X_q до полной длины или нет.

В качестве значения кода аутентификации выбирается *s* (s ≤ *n*) старших битов последнего блока шифртекста.

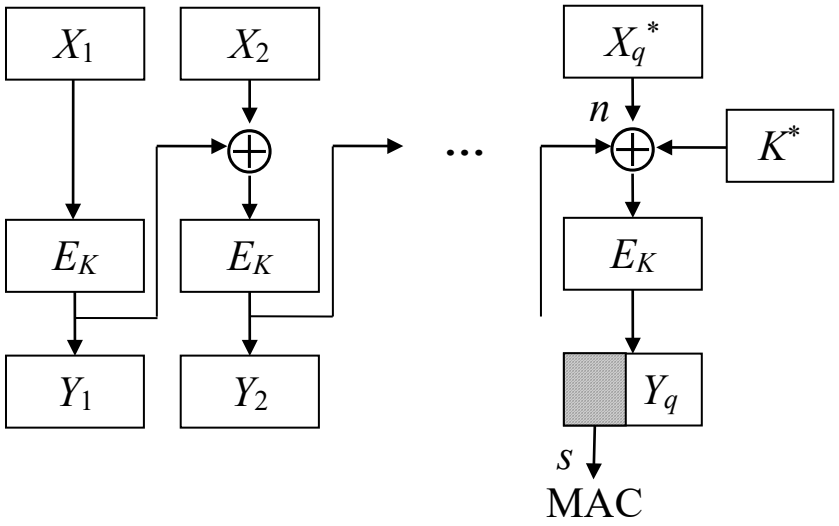

Рис. 2.15. Режим выработка имитовставки (MAC)

Наряду со стандартами ГОСТ, в России принят ряд Рекомендаций по стандартизации, регламентирующих различные аспекты применения криптографических систем. Рекомендации по стандартизации Р 1323565.1.005–2017 «Информационная технология. Криптографическая защита информации. Допустимые объёмы материала для обработки на одном ключе при использовании некоторых вариантов режимов работы блочных шифров в соответствии с ГОСТ Р 34.13-2015» определяет максимально допустимое количество блоков открытого текста, которое может быть зашифровано с использованием одного и того же ключа (табл. 2.9).

Здесь используются следующие обозначения:

— $N_{\max}$ — максимально допустимое количество блоков (объём материала), которые могут быть обработаны с использованием выбранного режима работы алгоритмом блочного шифрования без изменения значения ключа;

— *n* — двоичная длина блока блочного шифра;

— π_{enc} — максимально допустимое значение вероятности эффективного применения методов криптоанализа;

— π_{mac} — максимально допустимое значение вероятности однократного навязывания сообщения;

— *s* — двоичная длина блоков открытого текста для режимов гаммирования (CTR), гаммирования с обратной связью по выходу (OFB), гаммирования с обратной связью по шифртексту (CFB) и выработки имитовставки (MAC);

— *m* — двоичная длина синхропосылки для режимов гаммирования с обратной связью по выходу (OFB), гаммирования с обратной связью по шифртексту (CFB) и простой замены с зацеплением (CBC).

Таблица 2.9

Ограничения на длину текста, шифруемого на одном и том же ключе

Название режима в соответствии с ГОСТ Р 34.13-2015	Вариант реализации режима	Максимально допустимое количество блоков открытого текста
Простой замены	–	$N_{max} = 1$
Гаммирования	$s = n$	$N_{max} = 2^{\frac{n}{2}}\sqrt{\pi_{enc}}$
Гаммирования с обратной связью по выходу	$s = m = n$	$N_{max} = 2^{\frac{n-1}{2}}\sqrt{\pi_{enc}}$
Простой замены с зацеплением	$m = n$	$N_{max} = 2^{\frac{n-1}{2}}\sqrt{\pi_{enc}}$
Гаммирования с обратной связью по шифртексту	$s = m = n$	$N_{max} = \sqrt{\frac{2}{3}}2^{\frac{n}{2}}\sqrt{\pi_{enc}}$
Выработки имитовставки	$s = n$	$N_{max} = \frac{2^{\frac{n}{2}-1}}{n}\sqrt{\pi_{mac} - \frac{1}{2^n}}$

Превышение указанных лимитов позволяет криптоаналитику противника получить достаточный объем материала для проведения комбинаторных атак на шифр, поэтому по достижении указанных объемов используемый ключ шифрования должен быть заменен. Для защищенных коммуникационных протоколов это может означать выполнение относительно трудоемкой процедуры пересогласования ключа, заключающейся в обмене служебными сообщениями и вычислениях по генерации случайных величин и согласованию ключа. Использование специальных режимов работы блочных шифров с преобразованием ключа, когда новые ключи получаются н основе ранее согласованных, позволяет продолжать обмен зашифрованными сообщениями без пересогласования ключей.

Два таких режима — гаммирования с преобразованием ключа (CTR-ACPKM) и вычисления имитовставки с преобразованием ключа (OMAC-ACPKM) — определены Рекомендациями по стандартизации Р 1323565.1.017–2018 «Информационная технология. Криптографическая защита информации. Криптографические алгоритмы, сопутствующие применению алгоритмов блочного шифрования» (рис. 2.16).

Защищенные коммуникационные протоколы предполагают, как правило, не только обеспечение конфиденциальности, но и контроль целостности передаваемых данных. Традиционно для этих задач использовались разные механизмы защиты — шифрование данных и расчет кода аутентификации MAC/HMAC, каждый — с использованием своего ключа в соответствии с принципом функционального разделения ключей.

В настоящее время отдается предпочтение *аутентифицированному шифрованию*, которое позволяет решить задачи обеспечения конфиденциальности и целостности сообщений одновременно с использованием одного и того же ключа. Режим аутентифицированного шифрования MGM (*Multilinear Galois Mode*) определен Рекомендациями по стандартизации Р1323565.1.026–2019 «Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров, реализующие аутентифицированное шифрование». Режим специфицирован для применения в протоколах TLS 1.3 и IPSec с российской криптографией.

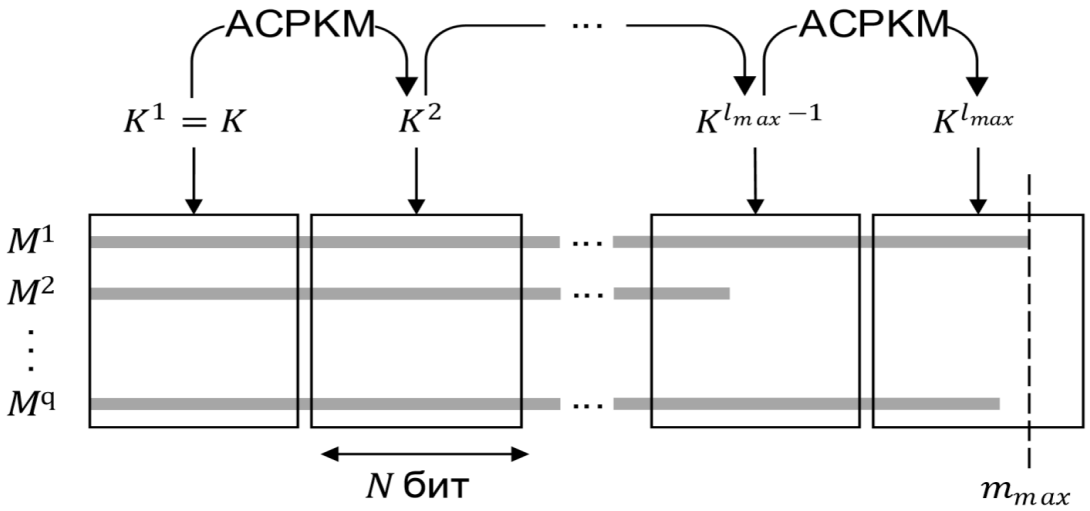

Рис. 2.16. Пример режима с преобразованием ключа — CTR-ACPKM

2.2.5. Основные методы криптоанализа блочных шифров

Метод грубой силы (brute-force attack) предполагает перебор всех возможных вариантов ключа шифрования до нахождения искомого. Если ключ алгоритма шифрования имеет длину n битов, мощность ключевого пространства $|K| = 2^n$, поэтому поиск истинного ключа потребует опробования в среднем половины всех вариантов ключа $2^n/2$, то есть выполнения 2^{n-1} тестовых операций шифрования. Метод грубой силы неэффективен при достаточно большом значении n. При этом увеличение длины ключа на 1 бит увеличивает среднее время атаки в 2 раза.

Критерием корректности найденного ключа является совпадение результата расшифрования известной криптограммы с соответствующим известным открытым текстом. Этот критерий позволяет однозначно идентифицировать ключ и требует наличия хотя бы одной пары «открытый текст — шифртекст». Несколько сложнее определить истинный ключ, если атака осуществляется при наличии только шифртекста. В этом случае требуется какая-либо дополнительная информация об открытом тексте (сообщение на естественном языке, формат компьютерного файла, наличие контрольной суммы для проверки целостности).

С развитием вычислительной техники скорость перебора возрастает, а его стоимость снижается. Поэтому требования к размерам ключа постоянно возрастают. В настоящее время для алгоритмов симметричного блочного шифрования рекомендовано использование 128-битных ключей.

Метод «встреча посередине». Любые методы, способные вскрыть алгоритм шифрования быстрее, чем полный перебор ключей, эксплуатируют те или иные конструктивные недостатки алгоритма или его реализации. Атака «встреча посередине» позволяет вскрывать любой алгоритм шифрования, представляющий собой двойное шифрование с помощью какого-либо «одинарного» алгоритма. Атака «встреча посередине» разработана в применении к алгоритму Double DES (двойной DES) (рис. 2.17). Ключи K_1 и K_2 алгоритма Double DES представляют собой 56-битные половины 112-битного ключа.

Атака методом «встреча посередине» является атакой на основе известных открытых текстов.

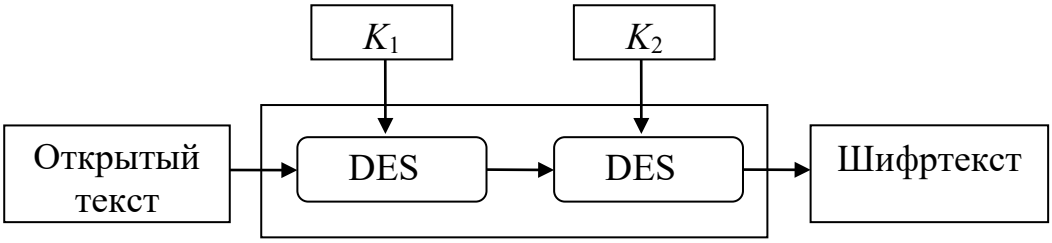

Рис. 2.17. Алгоритм двойного DES

Пусть известна пара «открытый текст — шифртекст»: $X_1 – Y_1$. Криптоаналитик должен выполнить следующую последовательность действий (рис. 2.18).

1. Выполняется шифрование DES $E_{kx}(X_1)$ на всем ключевом пространстве $\{K\}$ ($kx = 0, 1, \ldots, 2^{56} – 1$), результаты записываются в таблицу.

2. Производится расшифровывание DES $E^{-1}_{ky}(Y_1)$ также на всем ключевом пространстве $\{K\}$ ($ky = 0, 1, \ldots, 2^{56} – 1$). Результаты расшифровки сравниваются с таблицей шифрования.

3. Если какой-либо результат расшифровки Y_1 совпал с одним из результатов шифрования X_1, то можно предположить, что нужный ключ найден, то есть соответствующие значения ключей $kx = K_1$, $ky = K_2$.

4. Следует иметь в виду, что таких совпадений может быть достаточно много — порядка 2^{48} для Double DES. Для уточнения истинного значения ключа требуется наличие еще одной пары «открытый текст — шифртекст»: $X_2 – Y_2$.

5. Эта пара используется так же, как и первая, но теперь перебор вариантов ключей kx и ky осуществляется только по уже отобранным значениям ключей, ранее давшим совпадения.

6. В результате будет найден верный ключ. Вероятность повторного совпадения крайне мала — порядка 2^{-16} (Double DES), в этом случае может быть использована третья пара $X_3 – Y_3$.

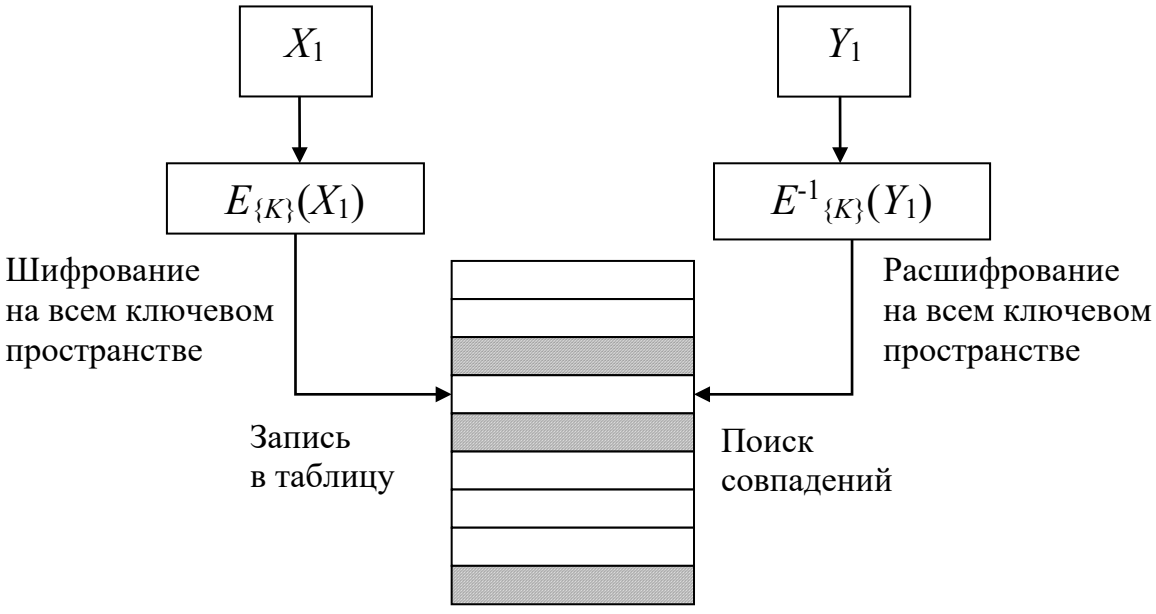

Рис. 2.18. Схема проведения атаки «встреча посередине»

Применение атаки «встреча посередине» к Double DES позволяет вычислить ключ за 2^{57} операций, что всего в два раза выше сложности полного перебора ключей обычного DES, и несравнимо ниже сложности полного перебора вариантов двойного

ключа (2^{112}). Атака «встреча посередине» применима, но малоэффективна для других более современных алгоритмов блочного шифрования.

Атаки методами «грубой силы» и «встреча посередине» нередко используются в комбинации с другими атаками, а также для оценки запаса криптостойкости модифицированных версий алгоритмов и алгоритмов с усеченным количеством раундов.

Дифференциальный криптоанализ. Классический дифференциальный криптоанализ предложен в 1990 году и требует выбора открытых текстов с определенными различиями, называемыми разностью (difference). При анализе разных алгоритмов разность текстов Δ может быть определена по-разному. Для DES разность открытых текстов $x1$ и $x2$ определяется как операция побитового исключающего ИЛИ этих текстов: $\Delta x = x1 \oplus x2$.

Задачей дифференциального криптоанализа является нахождение таких разностей исходных текстов Δx и последовательностей раундовых ключей k_1, k_2,…, k_r, чтобы разность выходных текстов (шифртекстов) Δy имела как можно большую вероятность. На основе различий в получившихся шифртекстах различным ключам присваиваются различные вероятности. В процессе дальнейшего анализа следующих пар шифртекстов один из ключей станет наиболее вероятным. Это и есть правильный ключ.

Анализ начинается с исследования однораундового алгоритма, затем переходят к алгоритмам с большим числом раундов. Для этого используются *характеристики* — разности открытых текстов, которые с высокой вероятностью вызывают определенные разности в получаемых шифртекстах. На сегодняшний день не известно универсальных алгоритмов поиска характеристик с высокой вероятностью для разных алгоритмов шифрования. Поэтому поиск характеристик является в сильной степени творческой задачей.

Существуют различные методы (бумеранга, невозможных дифференциалов и др.), являющиеся развитием классического дифференциального криптоанализа.

Линейный криптоанализ впервые был применен в 1993 году, этот метод вскрытия с известным открытым текстом основан на поиске линейной аппроксимации между исходным и зашифрованным текстом, то есть соотношений вида $x_{i1} \oplus x_{i2} \oplus \ldots \oplus x_{ia} \oplus y_{j1} \oplus y_{j2} \oplus \ldots \oplus y_{jb} = k_{k1} \oplus k_{k2} \oplus \ldots \oplus k_{kc}$, где x_n, y_n, k_n — n-е биты открытого текста, шифртекста и ключа соответственно.

Для произвольно выбранных битов открытого текста, шифртекста и ключа вероятность того, что такое соотношение будет выполняться, составляет около 1/2 ($P \approx 1/2$). Если криптоаналитику удастся найти такие биты, при которых вероятность существенно отличается от 1/2, это может быть использовано для вскрытия алгоритма.

Сначала производится поиск однораундового соотношения, а затем делается попытка распространить на большее количество раундов его. Существуют и алгоритмы поиска полезных соотношений.

При определенных обстоятельствах линейное соотношение может быть преобразовано к виду $y_{j1} \oplus y_{j2} \oplus \ldots \oplus y_{jb} = k_{k1} \oplus k_{k2} \oplus \ldots \oplus k_{kc}$, что позволяет исключить необходимость знания открытого текста и проводить атаку только на основании известного шифртекста. Например, описана подобная атака на полнораундовый DES, которая работает при допущении, что исходное сообщение — осмысленный текст на английском языке в ASCII-кодировке. Для этой атаки требуется порядка 2^{54} шифртекстов, что ненамного эффективнее полного перебора ключей.

Применяется также объединение методов линейного и дифференциального криптоанализа (линейно-дифференциальный криптоанализ). Эффективность всех этих методов резко падает при увеличении числа раундов шифрования, поэтому представляют интерес методы криптоанализа, не зависящие от числа раундов.

Слайдовая атака. К методам криптоанализа, трудоемкость которых не зависит от числа раундов в алгоритме шифрования, относятся слайдовая атака (скользящая, сдвиговая атака, *Slide Attacks*) и атака на связанных ключах. *Слайдовая атака* эксплуатирует степень самоподобия шифрующей функции f, зависящей от одного и того же подключа k в каждом раунде шифрования (значения всех раундовых ключей одинаковы). В зависимости от структуры алгоритма шифрования, слайдовая атака может использовать как слабость процедуры формирования раундовых ключей, так и более общие структурные свойства шифра. Идея слайдовой атаки заключается в том, что можно сопоставить один процесс зашифрования с другим таким образом, что один из процессов будет «отставать» от другого на один раунд (рис. 2.19).

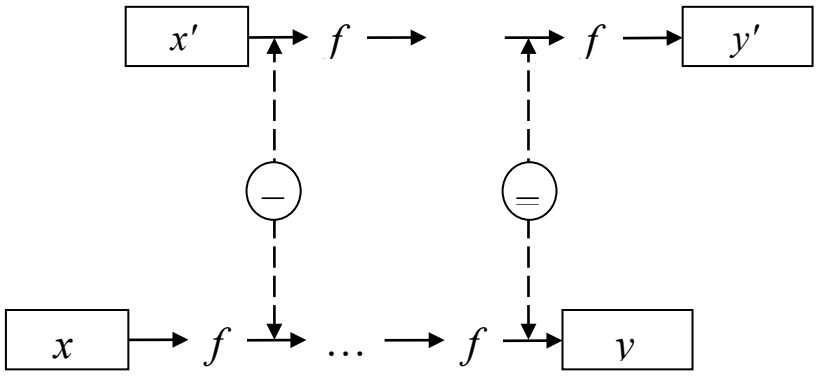

Рис. 2.19. Схема обычной слайдовой атаки

Пусть x_0 и x'_0 — некоторые открытые тексты, в процессе их шифрования $x_j = f(x_{j-1})$, $x'_j = f(x'_{j-1})$, $j = 1, 2, \ldots, r$. Если имеется некоторая пара значений, такая что, $x'_0 = x_1$, то и для любого j выполняется $x'_{j-1} = x_j$, поскольку $x'_j = f(x'_{j-1}) = f(x_j) = x_{j+1}$. А значит, и $x'_{r-1} = x_r$. Обозначим входы шифрования как x и x', выходы — как y и y'. Тогда для слайдового процесса $x' = f(x)$, будет выполнено и условие $y' = f(y)$. Пара открытых текстов и соответствующих им шифртекстов (x,y) (x',y') называется слайдовой парой в том случае, если выполнены условия $f(x) = x'$ и $f(y) = y'$.

Слайдовая атака проходит следующим образом. Получают пары «открытый текст — шифртекст» (x_i, y_i) с помощью одного и того же ключа, среди которых ожидают найти хотя бы одну слайдовую пару. После того, как слайдовая пара найдена, могут быть определены некоторые биты ключа. Для выяснения оставшихся битов секретного ключа следует определить следующую слайдовую пару, и с ее помощью провести анализ. Таким образом, для полного определения секретного ключа достаточно найти всего несколько слайдовых пар.

В случае, когда речь идет об алгоритмах шифрования, построенных на схеме Фейстеля, раундовая функция преобразует только половину входного сообщения. Поэтому условие $f(x) = x'$ можно легко проверить, сравнив правую часть сообщения x и левую часть сообщения x'. Таким образом, сложность атаки может быть снижена до $2^{n/2}$ открытых текстов, где n — длина входного блока алгоритма шифрования.

Для сети Фейстеля условие нахождения потенциальной слайдовой пары может быть сформулировано следующим образом: (x, y) образует слайдовую пару совместно с (x', y'), если правые половины текстов x и y' равны соответственно левым половинам x' и y: $x\_R = x'\_L$, $y'\_R = y\_L$.

Если существует возможность выбора открытых текстов, то для сети Фейстеля сложность анализа может быть снижена до $2^{n/4}$ открытых текстов. В этом случае выбирается произвольное $n/2$-битовое значение x, затем подбирается массив открытых текстов

$x'_i = (x, g_i)$, которые будут различаться только случайно выбранной правой частью g_i, и массив $x_j = (g_j, x)$, которые будут различаться только случайной левой частью g_j.

Слайдовые атаки применимы также для алгоритмов, в которых функция формирования подключей периодична, то есть один и тот же подключ повторяется через равное количество раундов. Известны эффективные слайдовые атаки для сети Фейстеля с двухраундовым самоподобием, комбинация которых позволяет проводить атаки на сети Фейстеля и с четырехраундовым самоподобием.

Атака на связанных ключах. Классическая атака на связанных ключах во многом похожа на слайдовую атаку. Основное отличие состоит в том, что если в слайдовой атаке используют два открытых текста x и x', связанные между собой соотношением $x' = f(x, k_1)$, где f — функция раундового преобразования, k_1 — ключ первого раунда, то атака на связанных ключах использует весьма похожее соотношение, только между ключами.

Пусть искомый ключ шифрования k в результате его обработки процедурой расширения ключа $g(k)$ дает следующую последовательность подключей:

$$g(k)=\{k_1, k_2, \dots, k_{r-1}, k_r\},$$

где r — количество раундов алгоритма шифрования.

Предположим также, что есть ключ k', расширение которого дает последовательность раундовых ключей, аналогичную предыдущей, но с циклическим сдвигом на одну позицию (рис. 2.20): $g(k') =\{k_2, k_3, \dots, k_r, k_1\}$.

Тогда возможна атака, позволяющая отыскать раундовый ключ k_1 при наличии $2^{n/2}$ пар «открытый текст — шифртекст», зашифрованных на первом ключе k и $2^{n/2}$ пар «открытый текст — шифртекст», зашифрованных на втором ключе k'. При этом потребуется анализ в среднем не более $2^{n/2+1}$ текстов. В зависимости от структуры самого алгоритма трудоемкость может быть значительно ниже (в частности, для алгоритмов, построенных на сбалансированной сети Фейстеля). Атака на связанных ключах долгое время считалась мощной, но сугубо теоретической. В любом случае она должна учитываться при разработке и реализации алгоритмов шифрования.

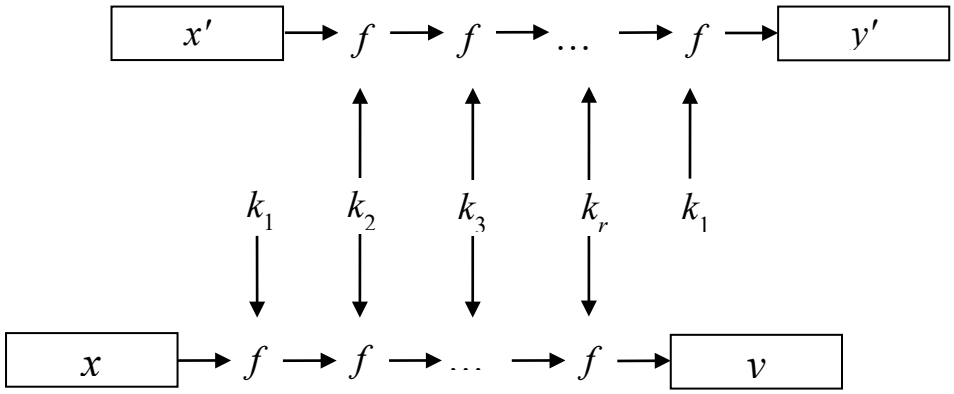

Рис. 2.20. Соотношение раундовых ключей в атаке на связанных ключах

Атаки, использующие утечки по побочным каналам. Существует и принципиально иной вид атак, которые эксплуатирует не уязвимости структуры шифра, а особенности его реализации, как правило, они нацелены на низкоресурсные аппаратные реализации с прошитым ключом шифрования (например, смарт-карты). Атаки этого вида заключаются либо в высокоточном замере характеристик шифратора, таких как: времени выполнения, потребляемой мощности, электромагнитного излучения (пассивные атаки), либо в анализе реакции на активные воздействия на шифратор (изменение напряжения питания, тактовой частоты, высокочастотное облучение или наведение электромагнитного поля, локальный нагрев шифратора, а также внесение изменений в его конструкцию).

Вопросы для самоконтроля:

1. Каковы основные принципы построения композиционных шифров?

2. Каковы основные параметры и особенности шифров «Магма» ГОСТ 34.12, «Кузнечик» ГОСТ 34.12?

3. Каковы основные методы анализа симметричных блочных шифров?

4. К каким криптосистемам применим криптоаналитический метод «встреча посередине»? В чем заключается суть этого метода?

5. В чем сущность методов линейного, дифференциального криптоанализа?

§ 2.3. Потоковые шифры

2.3.1. Общие свойства потоковых шифров

Потоковые шифры преобразуют открытый текст в шифртекст последовательно, бит за битом. Фактически они являются шифром гаммирования и различаются только методами генерации ключевой гаммы $\gamma_1(k)$, $\gamma_2(k)$, ..., $\gamma_i(k)$, ..., зависящими от ключа k. Шифрование производится наложением гаммы на открытый текст: $y_i = x_i \oplus \gamma_i(k)$,

расшифрование — повторным наложением гаммы:

$$x_i = y_i \oplus \gamma_i(k) = x_i \oplus \gamma_i(k) \oplus \gamma_i(k).$$

Общие свойства потокового шифра:

1. Ошибки, возникшие в определенных позициях криптограммы, полностью сохраняются в тех же позициях после ее расшифрования:

$$\tilde{y} = y \oplus e \Rightarrow \tilde{x} = x \oplus e,$$

где e — двоичная последовательность с единицами в позициях, где возникли ошибки и с нулями в остальных позициях.

2. Для правильного расшифрования потокового шифра необходима синхронизация источников гаммы при шифровании и расшифровании с точностью до одного бита. По способу синхронизации поточные шифрсистемы делят на *синхронные* и системы *с самосинхронизацией.* При использовании синхронной системы потеря знака шифртекста приводит к невозможности расшифрования оставшейся части сообщения. Поэтому должны быть предусмотрены специальные способы восстановления синхронности работы. Поточные системы с самосинхронизацией правильно расшифровывают криптограмму и при потере какого-либо ее знака.

3. Криптостойкость потокового шифра целиком определяется непредсказуемостью ключевой гаммы. При известном алгоритме формирования ключевой гаммы и неизвестном ключе знание никакой части ключевой гаммы не должно позволять за приемлемое время вычислений определить продолжение ключевой последовательности. Данное свойство достигается использованием генераторов истинно случайных равномерно распределенных последовательностей либо криптографически сильных генераторов псевдослучайных чисел.

4. Если в разных сеансах связи ключевая последовательность повторяется, то взлом шифра достаточно прост. Перехватив два шифртекста, противник может сложить их по модулю 2 и получить два исходных текста, сложенных также по модулю 2. Такую систему раскрыть очень просто. Если же в руках противника окажется пара «исходный текст — шифртекст», задача становится тривиальной.

На практике ключевая последовательность обычно вырабатывается режимами гаммирования блочных симметричных шифров, другой вариант – использование специально построенных криптографически сильных генераторов псевдослучайных чисел (например, BBS-генератор). Такой генератор может быть построен и с использованием линейных рекуррентных регистров сдвига (ЛРР). Потоковые шифры на базе сдвиговых регистров активно использовались в годы войны, задолго до появления электроники. Они просты в проектировании и реализации. Следует отметить, что сам по себе линейный реккурентный регистр сдвига не может обеспечить достаточной стойкости, что требует комбинации с нелинейным преобразованием. Следует также иметь в виду, что любая псевдослучайная последовательность периодична, то есть повторяется через некоторое, возможно достаточно большое, число *M* элементов. Использования каскадов ЛРР с управлением тактированием позволяет увеличить период формируемой гаммы.

2.3.2. Линейный рекуррентный регистр сдвига

Основные определения конечных полей. Для описания свойств линейного реккурентного регистра сдвига необходимо ввести основные определения конечных полей.

Конечным полем GF(*q*) или полем Галуа порядка *q*, называют конечное произвольное множество элементов с заданными между ними операциями сложения, умножения и деления. Конечным полем *GF*(2) или полем Галуа порядка 2, называют множество, состоящее из двух элементов (0 и 1) с заданными между ними операциями сложения, умножения и деления. Все операции выполняются по модулю два (0+0=0, 0+1=1+0=1, 1+1=0).

Характеристикой p конечного поля *GF*(*q*) называется минимальное число, такое что $e \cdot p = \underbrace{e + e + \cdots + e}_{p} = 0$, где $e = a \cdot a^{-1}$. В поле *GF*(2) существует всего два элемента: 0 и 1. Пусть $a = 0$. Тогда для $e = a \cdot a^{-1} = 0 \cdot 1 = 0$. Аналогично, если $a = 1$, то $e = a \cdot a^{-1} = 1 \cdot 0 = 0$. Следовательно, характеристика поля *GF*(2) равна единице.

Многочлен *p*(*x*) называется *неприводимым* над полем *GF*(*p*), где $p^n=q$ (*n* — длина последовательности), если он не может быть представлен как произведение двух и более многочленов с коэффициентами из поля *GF*(*p*).

Порядком e элемента a конечного поля *GF*(*q*) называется наименьшее целое положительное число такое, что $a^e = 1$. Очевидно, что порядок каждого из двух элементов поля *GF*(2) равен единице: $0^1=1$, $1^1=1$.

Элемент a конечного поля *GF*(*q*) называется *примитивным*, если его порядок равен *q*-1. Поскольку для поля *GF*(2) *q*-1= 1, оба его элемента (0 и 1) являются примитивными.

Неприводимый многочлен p(*x*) над полем *GF*(*p*) называется *примитивным*, если его корень, принадлежащий *GF*(p^n), является примитивным элементом этого поля.

Линейный рекуррентный регистр сдвига (ЛРР) задается рекуррентным уравнением:

$$b_j = \begin{cases} a_j, \text{если } j = 1, 2, \ldots, n-1; \\ \sum_{i=0}^{n-1} h_i b_{i+j-n}, \text{если } j \geq n. \end{cases} \qquad (2.2)$$

Здесь действия с коэффициентами и сложение производятся над полем Галуа *GF*(2), *n* — длина регистра, a_0, a_1, …,a_{n-1} — двоичное начальное заполнение регистра (выступает в качестве ключа *k* при использовании регистра в качестве генератора

гаммы), b_j — двоичная последовательность на выходе регистра, h_i — двоичные коэффициенты, определяющие структуру регистра. Все параметры в уравнении (2.2), за исключением длины регистра — n, представлены в двоичном виде, поэтому регистр является двоичным.

На рис. 2.21 приведена схема аппаратной реализации двоичного линейного рекуррентного регистра сдвига. Работа регистра при формировании гаммы происходит следующим образом. Вначале во все ячейки регистра поступает двоичный начальный код. Затем в регистр начинают поступать тактовые импульсы. При поступлении каждого тактового импульса содержимое регистра сдвигается на одну ячейку вправо. Одновременно содержимое ячеек, помноженное на двоичные коэффициенты h_i, поступает на входы сумматоров по модулю два, обозначенных на схеме значком ⊕. Содержимое крайней правой ячейки поступает на выход схемы и на вход первой ячейки регистра.

Таким образом, при ненулевом начальном заполнении схемы и ненулевом векторе h_0, h_1, …, h_{n-1} на выходе схемы будет формироваться потенциально неограниченная по длине двоичная последовательность. Нулевое состояние недопустимо, так как в этом случае на выходе регистра всегда будет формироваться нулевая последовательность. Регистр длиной n-разрядов может находиться в одном из 2^n-1 внутренних состояний, следовательно, формируемая на его выходе последовательность может иметь максимальный период $M = 2^n-1$ бит.

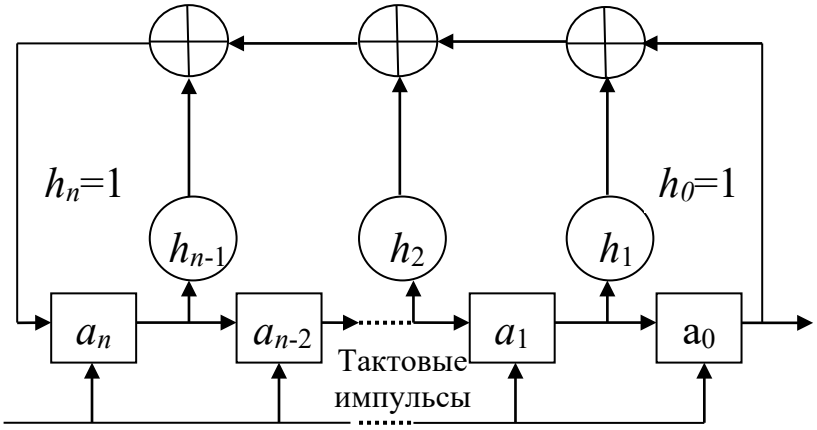

Рис. 2.21. Схема линейного рекуррентного двоичного регистра

Последовательность с максимально возможным периодом $M = 2^n-1$ бит будет формироваться не при любых начальных заполнениях. Выходная последовательность имеет максимально возможный период, если многочлен $g(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x^1 + 1$, образованный из начальной последовательности a_0, a_1, …, a_{n-1}, является примитивным по модулю два. *Примитивный многочлен* степени n – это неприводимый многочлен, который является делителем $x^{2n-1}+1$, но не является делителем x^d+1 для всех d, являющихся делителями 2^n-1. Не существует простого способа получения примитивных многочленов степени n по модулю 2. Проще всего выбирать многочлен случайным образом и проверять, не является ли он примитивным. Алгоритмы проверки многочленов на примитивность достаточно сложны и требуют большого объема вычислений. Ситуация упрощается тем, что во многих математических пакетах есть средства для решения этой задачи.

Линейный рекуррентный двоичный регистр сдвига обладает следующими основными свойствами.

Свойство 1. Минимально возможный период выходной последовательности равен такому минимальному целому числу n_0, для которого многочлен $x^{n_0}+1$ делится на многочлен обратных связей $h(x)=h_n x^n+h_{n-1}x^{n-1}+\dots h_1 x+h_0$, где h_i — двоичный коэффициент, определяющий структуру регистра (при этом всегда $h_0=h_1=1$). Этот минимальный период будет достигаться для начального заполнения, состоящего из коэффициентов многочлена $g(x)=(x^{n_0}+1)/h(x)$ и следующих за ними n_0-n-1 нулей. Для других начальных заполнений период может оказаться и меньше n_0. Если многочлен $h(x)$ окажется неприводимым над полем $GF(2)$, то при любом начальном заполнении период выходной последовательности в точности равен n_0.

Для линейного рекуррентного регистра, реализованного на *примитивном полиноме*, период выходной последовательности при любом начальном заполнении будет в точности равен 2^n-1. Последовательности, генерируемые такими регистрами, называются *последовательностями максимальной длины*.

Пример. Пусть регистр длины 4 имеет полином обратных связей $h(x)=x^4+x^3+x+1=(x+1)(x^2+x+1)$. Тогда получим

$$h(x)/x^6+1 \Rightarrow M=6,\ g(x)=\frac{x^6+1}{h(x)}=x^2+x+1.$$

Поэтому для начального заполнения 0111 выходная последовательность оказывается равной 111000 и имеет период равный 6. Если же выбрать $h(x)$ примитивным, например $h(x)=x^4+x+1$, то такой регистр будет иметь максимальный период выходной последовательности равный 15 для любого начального заполнения.

Выбор линейного рекуррентного регистра на примитивном полиноме является необходимым условием при его использовании в качестве датчика гаммы в потоковом шифре. В противном случае гамма будет повторяться и криптоаналитик может расшифровать криптограмму. Далее будем рассматривать только регистры, построенные на примитивных полиномах.

Свойство 2. Свойство «окна» для выходной последовательности регистра. Пусть b' — выходная последовательность регистра, а b'' — подпоследовательность последовательности b' длины $2^n+k-2,\ k\le n$. Тогда каждая ненулевая подпоследовательность b''' длины k будет в последовательности b'' присутствовать в точности 2^{n-k} раз. Это свойство называется свойством окна, поскольку подпоследовательности b''' получаются как будто при перемещении окна шириной k вдоль подпоследовательности b''. Отсюда следует, что, если $k=n$, то все ненулевые подпоследовательности будут появляться в окне точно по одному разу.

Свойство 3. Свойство баланса для выходной последовательности. Количество нулей на периоде выходной последовательности линейного регистра сдвига в точности равно $2^{n-1}-1$, а количество единиц — 2^{n-1}. Действительно, если состояния ячеек памяти регистра представить целыми числами, то нечетным числам будет соответствовать появление единиц в выходной последовательности, а четным — появление нулей. Но количество нечетных чисел на множестве 2^n-1 равно 2^{n-1}, а четных — $2^{n-1}-1$.

Свойство 4. Свойство серий для выходной последовательности. Различают *серии блоков* (единиц) длиной k, $\ldots 0\underbrace{11\ldots1}_{k}0\ldots$ и *серии пробелов* (нулей) длиной k, $\ldots 1\underbrace{00\ldots0}_{k}1\ldots$ Свойство серий состоит в том, что половина всех серий на периоде выходной последовательности имеет длину 1, четверть — длину 2, восьмая часть — длину 3 и т. д. При этом половина серий любой длины — блоки, а половина — пробелы.

Свойство 5. Автокорреляционная функция выходной последовательности. Эта функция определяется следующим образом:

$$R(k) = \frac{A(k)-B(k)}{M} = \frac{2A(k)-M}{M},$$

где $A(k)$ и $B(k)$ — число совпадений и несовпадений соответственно между b_i и b_{i+k} на периоде M. При этом сумма $i+k$ вычисляется по модулю M. Для автокорреляционной функции выходной последовательности имеет место следующее соотношение:

$$R(k) = \begin{cases} 1, \text{если } k = 0, \\ -1/(2^n - 1), \text{если } 1 \le k \le (2^n - 1) \end{cases}.$$

Рассмотренные пять свойств линейного рекурсивного регистра сдвига свидетельствуют о том, что он формирует на выходе последовательность, которая удовлетворяет нескольким постулатам для псевдослучайных последовательностей. Однако следующее свойство показывает невозможность использования этого регистра непосредственно в качестве датчика гаммы в потоковом шифре.

Свойство 6. Предсказуемость выходной последовательности. По известной последовательности длины $2n$: $\underbrace{b_k, b_{k+1} \ldots b_{k+2n-1}}_{2n}$ можно однозначно найти все обратные связи $h_0, h_1, \ldots, h_{n-1}$ линейного реккурентного регистра сдвига, а затем и предсказать все последующие элементы на выходе регистра. Причем сложность решения этой задачи не выше $O(n^2)$, где n — длина регистра.

Следовательно, зная $2n$ смежных элементов сообщения и криптограммы, можно найти $2n$ элементов гаммы и вычислить ее продолжение на любую длину. Поэтому использование регистра в качестве генератора гаммы шифрования нецелесообразно. На практике его используют в качестве датчика первичной гаммы, которая затем преобразуется при помощи нелинейных операций. Такие преобразователи называются *нелинейными узлами усложнения*. Простейший нелинейный узел – узел перемножения, работающий в соответствии с правилами: 0·0=0, 0·1=0, 1·0=0, 1·1=1. Тогда устройство потокового шифрования приобретает вид, как показано на рис. 2.22.

В качестве ключа здесь можно использовать начальное состояние регистра a_0, a_1, …, a_{n-1} или коэффициенты h_0, h_2, …, h_{n-1}. При выборе в качестве ключа начального заполнения примитивный полином обратных связей является открытым и может быть выбран заранее. В этом случае полное число нетривиальных ключей будет равно 2^n–1 и при выборе $n \ge 128$ криптоанализ путем полного перебора становится практически неосуществимым.

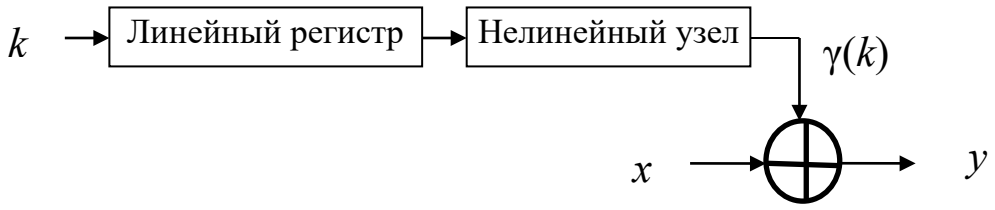

Рис. 2.22. Структура устройства потокового шифрования

Если в качестве ключа выбран полином обратных связей, то он должен быть примитивным и секретным, поскольку полное число таких ключей оказывается меньше 2^n–1. Однако при больших значениях n число ключей достаточно велико и подобрать ключ также не удастся.

2.3.3. Основные методы анализа потоковых криптосистем

Основными видами атак на потоковые шифры являются: метод грубой силы — тотальный перебор ключей; использование алгоритма Берлекэмпа — Месси; корреляционные атаки; быстрые корреляционные атаки; побочные атаки.

Тотальный перебор ключей. Ключом чаще всего является начальное заполнение регистра. В этом случае общее число возможных ключей равно 2^{n-1}. Если ключом являются коэффициенты обратных связей, то число возможных ключей равно $N_2(n) = \varphi(2^n - 1)/n$, где φ — функция Эйлера, n — длина регистра, или $N_2(n) = (2^n - 2)/n$, если 2^n–1 — простое число. В качестве ключа могут быть выбраны начальное заполнение регистра и коэффициенты обратных связей вместе. Тогда число возможных ключей возрастет соответствующим образом до 2^{2n}–1. Для исключения атаки перебором величина n должна быть достаточно большой.

Атака на основе алгоритма Берлекэмпа — Месси. Суть атаки данного вида заключается в следующем. Не зная структуры шифра, но зная, что он основан на использовании одного или нескольких линейных рекуррентных регистров сдвига, и зная, что линейная эквивалентная сложность шифрующей гаммы является обозримой величиной, находится ее отрезок $\gamma = x \oplus y$ большой длины N. Затем к этому отрезку применяется алгоритм Берлекэмпа — Месси, который позволяет найти длину эквивалентного регистра, а также его отводы, и начальное заполнение, а затем — вычислить произвольное продолжение гаммы.

Для предотвращения атаки данного вида требуется использовать потоковые шифры с нелинейными узлами управления такой линейной эквивалентной сложности L гаммы, что определение $N = 2L$ элементов этой гаммы или выполнение $O(L^2)$ операций оказывается практически неосуществимым. Пример такого шифра — генератор гаммы с управляемым тактированием (рис. 2.23) с параметрами $n_1 \approx n_2 \approx n_3 \approx 128$.

В схеме на рис. 2.23: $\otimes$ — умножитель, $\oplus$ — сумматор по модулю два. ЛРР-1 регулярно тактируется импульсами ТИ, а ЛРР-2 и ЛРР-3 — нерегулярно, в зависимости от наличия или отсутствия импульсов ТИ-1, ТИ-2, вырабатываемых псевдослучайно при помощи выходной последовательности ЛРР-1. Ключи k_1, k_2, k_3 — начальные состояния регистров, длиной n_1, n_2, n_3 соответственно. Если выход ЛРР-1 равен 1, то ЛРР-2 продвигается, а ЛРР-3 — нет, и на его выходе остается предыдущий символ. Если выход ЛРР-1 равен 0, то все происходит в точности наоборот.

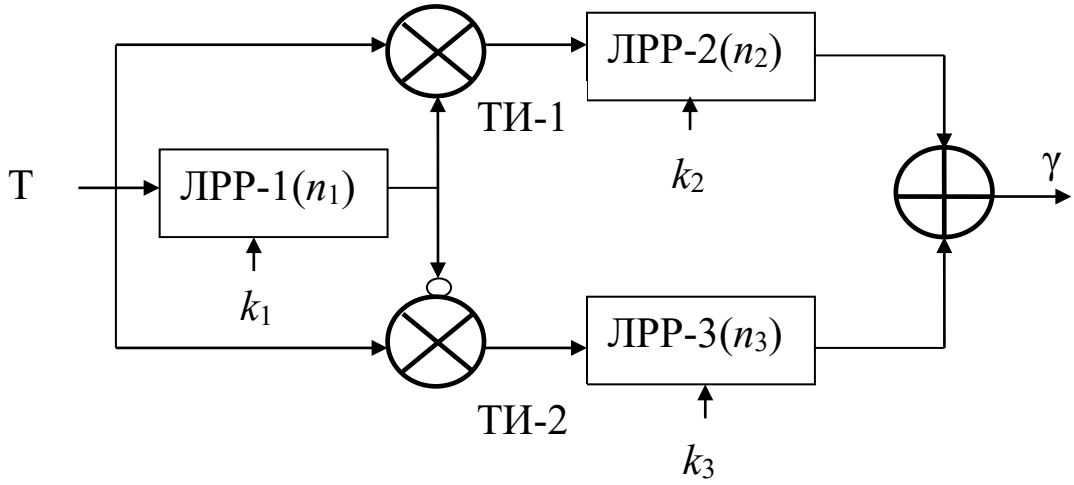

Рис. 2.23. Датчик гаммы на ЛРР с управляемым тактированием

Доказано, что период M выходной последовательности данной схемы будет равен $M = 2^{n_1}(2^{n_2} - 1)(2^{n_3} - 1)$ при условии, что вместо ЛРР-1 используется так называемый генератор последовательности де Брейна, и при выполнении дополнительного условия: n_2 и n_3 — взаимно просты: НОД(n_2, n_3) = 1. Последовательность де Брейна получается из ЛРР-1, если в конце каждой генерируемой подпоследовательности, состоящей из n_1–1 смежных нулей, добавляется еще один ноль. Эта схема с параметрами $n_1 \approx n_2 \approx n_3 \approx 128$ устойчива к любым известным атакам на потоковые шифры.

Корреляционные атаки. Допустим, что датчик шифрующей последовательности построен по схеме (рис. 2.24).

Зная вид булевой функции $f(x_1, x_2, \ldots, x_m)$, обеспечивающей нелинейное преобразование выходов ЛРР, можно рассчитать теоретическую корреляцию между двоичным выходом x_{ik} каждого из ЛРР и двоичным элементом гаммы γ_k.

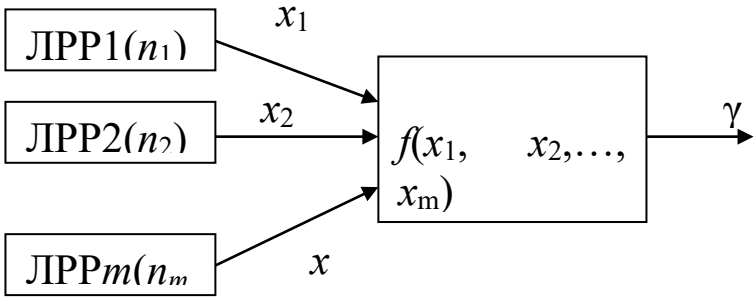

Рис. 2.24. Схема формирования гаммы с использованием нескольких ЛРР

2.3.4. Потоковые шифры сетей GSM

Стандарты шифрования в сетях GSM. В сетях GSM используется несколько стандартов шифрования, а именно: A3 — для аутентификации пользователей, A8 — для генерации ключа шифрования, A5 — для шифрования передаваемых сообщений. Место и назначение каждого из этих шифров показано на рис. 2.25.

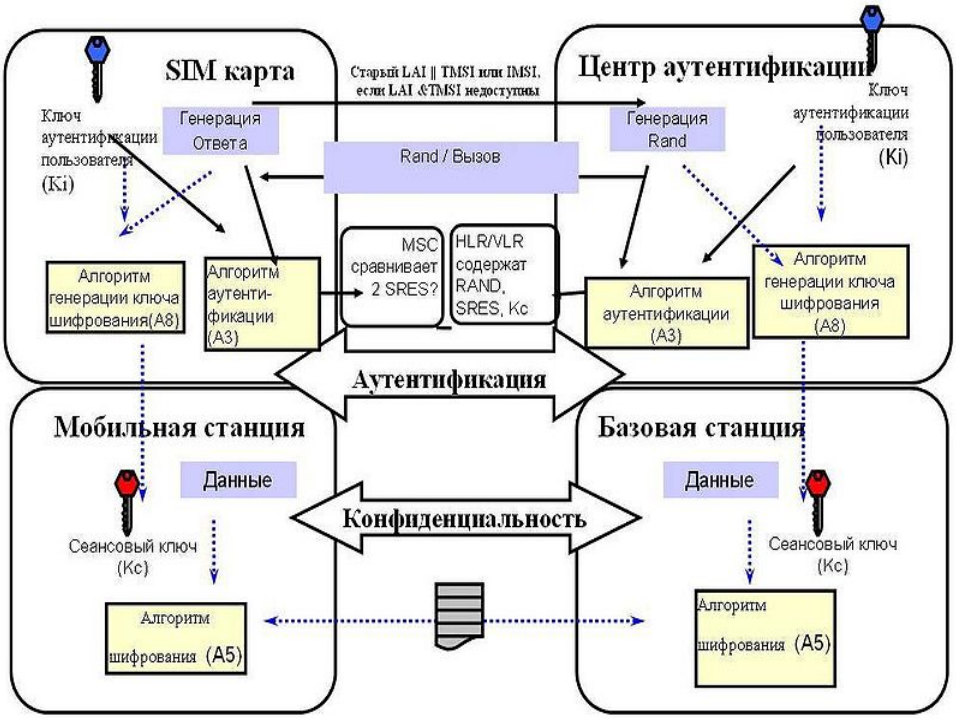

Рис. 2.25. Стандарты шифрования в сетях GSM

Алгоритм организации обмена информацией в сети состоит из двух этапов.

1. *Аутентификация пользователей*. Телефон абонента подключается к сети. Для подтверждения своей подлинности телефон посылает в центр аутентификации специальный идентификационный код TMSI (Temporary Mobile Subscriber Identity). В ответ на это центр аутентификации генерирует случайное число RAND длиной 128 бит и посылает его на мобильную станцию. Станция зашифровывает это число с помощью алгоритма аутентификации A3 и своего секретного ключа K_i. Затем она берет первые 32 бита из последовательности, полученной на предыдущем шаге, и отправляет их обратно в центр аутентификации. Центр, проделав аналогичную операцию, получает 32-битное число и сравнивает его с числом, полученным от мобильной станции. Если числа равны, телефон считается аутентифицированным.

2. *Обеспечение конфиденциальности*. Мобильная станция и центр аутентификации вычисляют сессионный ключ шифрования, используя секретный ключ K_i и алгоритм формирования ключа A8, $Kc = A8(K_i)$. После этого телефон абонента готов к работе. Шифрование переговоров осуществляется с помощью этого сессионного ключа и шифра стандарта A5.

Семейство алгоритмов стандарта шифрования A5. Стандарт A5 представляет собой семейство, состоящее на 2015 год из трех алгоритмов: потоковый шифр A5/1, потоковый шифр A5/2 и блочный шифр A5/3. Остановимся более детально на шифре A5/1. Генератор гаммы этого шифра отображен на рис. 2.26.

Генератор гаммы состоит из трех ЛРР длины 19, 22 и 23 с отводами обратных связей, соответствующих примитивным полиномам. Это обеспечивает периоды выходных последовательностей $2^{19}–1$, $2^{22}–1$, $2^{23}–1$ соответственно. Все три регистра используют псевдослучайное тактирование: биты с отводами 8 первого ЛРР, 10 — второго и третьего ЛРР подаются на так называемый мажоритарный элемент. Последний (на рис. 2.26 он входит в состав схемы управления тактированием) формирует на выходе значение ноль или единица в зависимости от того, чего больше на его входах: нулей или единиц. Затем выход мажоритарного элемента сравнивается с выходами 8, 10 и 10 ЛРР 1, 2 и 3 соответ-

ственно, сигналы с которых подавались ранее на его входы. Каждый ЛРР продвигается на один такт тогда и только тогда, когда сравниваемые биты оказываются одинаковыми. Шифрующая гамма формируется как сумма по модулю два выходов всех трех ЛРР. Ключом является начальное заполнение регистров. Таким образом, длина ключа оказывается равной 19 + 22 + 23 = 64 бита.

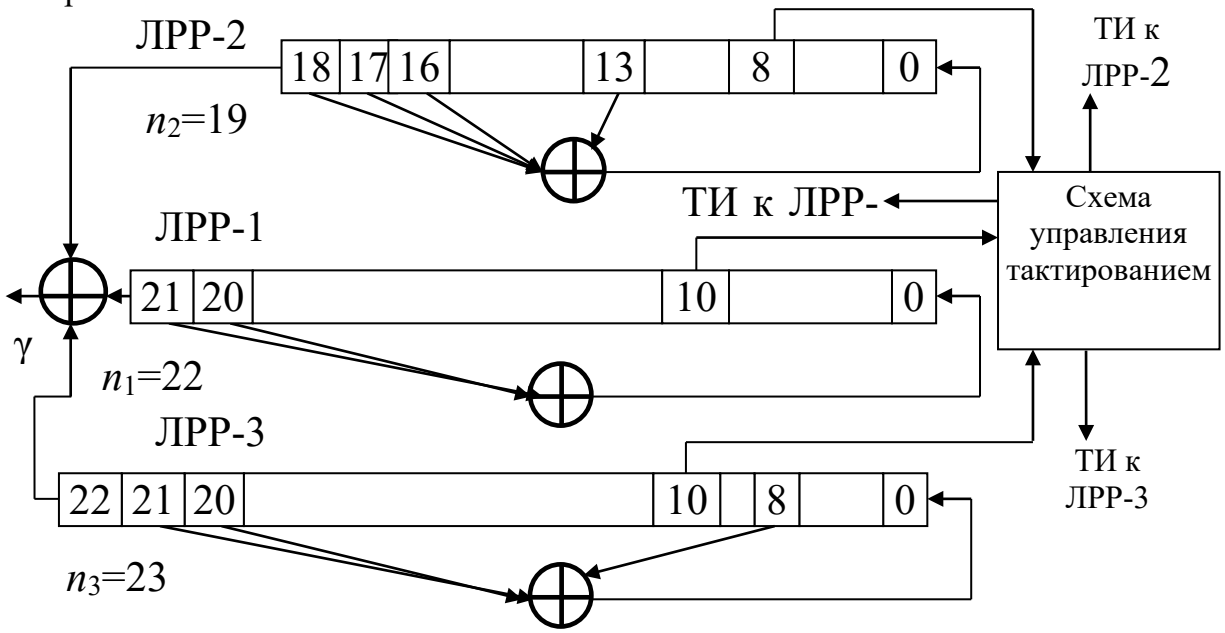

Рис. 2.26. Алгоритм шифрования A 5/1 стандарта GSM

Стойкость шифра A5/1. При разработке шифра предполагалось, что он будет иметь высокую стойкость. Однако затем независимые криптографы обнаружили у него слабые стороны. Одна слабость заключается в том, что ЛРР, входящие в состав шифратора, имеют малые длины, поэтому они подвержены некоторым модификациям статистических атак, а также атакам на основе соотношений между требуемыми объемами памяти и временем анализа. Данный шифр может быть взломан даже с помощью обычного ПК в реальном времени.

Применение и модификации шифра A5/1. Шифр A5 используется для шифрования сообщений в системе *GSM-R* на участке между мобильной и базовой станциями. На мобильную станцию по каналам служебной связи передается сеансовый ключ шифрования из состава триплета, используемого центром коммутации в ходе аутентификации. В *GSM*-системах, поставляемых на восточные рынки, используется модификация A5/2 алгоритма A5/1. В этой схеме используется четыре ЛРР, три из которых имеют те же характеристики, что и A5/1, а четвертый используется в качестве генератора нерегулярных тактовых импульсов для первых трех ЛРР. Для усложнения шифрующей гаммы в схеме используется дополнительная нелинейная последовательность, образуемая суммой по модулю два выходов трех мажоритарных элементов, на вход каждого из которых подается последовательность с некоторых промежуточных разрядов каждого ЛРР.

В поздних версиях GSM (GSM/GPRS) используется более совершенный алгоритм A5/3, называемый также GEA3. Этот алгоритм реализован как блочный шифр Касуми в режиме работы OFB. Шифр Касуми построен по схеме Фейстеля с восемью раундами. Сокращенная версия данного шифра с использованием шести раундов может быть взломана с использованием атаки на связанных ключах. Полная версия пока требует 2^{55} выбранных сообщений и 2^{74} шифрований, то есть обладает приемлемой вычислительной стойкостью. У шифра найден ряд уязвимостей, но о практически реализованных атаках пока не известно.

Вопросы для самоконтроля:

1. Каковы основные свойства потоковых шифров?

2. Каков минимальный и максимальный период гаммы, формируемой линейным рекуррентным регистром сдвига?

3. Какие схемы на линейных рекуррентных регистрах сдвига используются в качестве криптографически сильного генератора гаммы?

5. Какие криптографические алгоритмы используются в мобильной связи стандарта GSM?

§ 2.4. Хэш-функции

2.4.1. Общие сведения о хэш-функциях

Шифрование сообщений позволяет решить одну из задач криптографии — обеспечение конфиденциальности. Для обеспечения другой задачи — контроля целостности используются криптографически сильные контрольные суммы — хэш-функции и коды аутентификации.

Хэш-функцией называется любая функция $y = h(x_1x_2\ldots x_n)$, которая строке символов (сообщению) $x = x_1x_2\ldots x_n$ произвольной длины n ставит в соответствие целое число (битовую строку) фиксированной длины. Результат y вычисления хэш-функции называется *хэш-значением* или *хэш-кодом*. В криптографии хэш-функции используются для контроля целостности в системах аутентификации сообщений и в системах электронной подписи. Хэш-функция должна быть односторонней. Это значит, что для любого сообщения его хэш-значение может быть легко вычислено, однако по известному хэш-значению невозможно определить исходное сообщение. Кроме того даже незначительное изменение исходного документа должно приводить к значительному изменению его хэш-значения. К криптографическим хэш-функциям предъявляются следующие основные требования:

1) для любого заданного x вычисление $h(x)$ должно выполняться относительно быстро;

2) для известного значения y практически невозможно найти x, такое, что $y = h(x)$;

3) для известного значения x практически невозможно найти другое сообщение x', $x' \neq x$, такое, что $h(x') = h(x)$;

4) практически невозможно найти пару каких-либо различных сообщений x и x', $x' \neq x$, для которых $h(x') = h(x)$.

Существование сообщений с одним и тем же хэш-значением называется *коллизией* хэш-функции. Так, нарушение третьего из приведенных выше требований называется *коллизией первого рода*, четвертого — *коллизией второго рода*. Возможность нахождения коллизий влечет ненадежность использующих хэш-функции криптосистем. Существование необратимых хэш-функций, для которых невозможно вычисление прообраза заданного хэш-значения теоретически не доказано. На практике используются конструкции, нахождение обратного значения для которых является сложной вычислительно задачей.

В криптографии используются хэш-функции двух классов: *бесключевые* хэш-функции и *одноключевые* хэш-функции. В иностранной литературе предназначенные для криптографии бесключевые хэш-функции принято называть кодом, обнаруживающим модификации (*MDC*-кодом), а ключевые — кодом аутентификации сообщений (*MAC*-кодом).

2.4.2. Бесключевые хэш-функции

Бесключевые хэш-функции могут быть слабыми и сильными. *Слабой* хэш-функцией называется односторонняя функция $h(x)$, удовлетворяющая первым трем вышеприведенным требованиям (то есть допускающая возможность коллизий второго рода). Сильной хэш-функцией называется односторонняя функция $h(x)$, удовлетворяющая всем четырем требованиям.

Использующиеся в настоящее время хэш-функции считаются криптографически стильными, хотя это строго не доказано ни для одной из них. Наиболее известными бесключевыми хэш-функциями являются SHA-1 (*Secure Hash Algorithm*) и MD5 (*Message Digest*), однако они содержат уязвимости, а для MD5 опубликован практически применимый алгоритм поиска коллизий и показана возможность генерирования поддельных цифровых сертификатов. В настоящее время в США принят стандарт SHA-2 (FIPS PUB 180), который фактически представляет семейство родственных SHA-1 хэш-функций (SHA-224, SHA-256, SHA-384, SHA-512) с различной длиной (в битах) хэш-значения. Несмотря на то, что SHA-2 до сих пор остается достаточно надежным, в 2015 году принят стандарт SHA-3 (FIPS PUB 202) на основе криптопримитива Keccak («кетчэк»). В SHA-3 определены 4 хэш-функции с разной длиной выхода (SHA3-224, SHA3-256, SHA3-384 и SHA3-512), наряду с SHA-3продолжает использоваться и SHA-2. Алгоритм Keccak, в зависимости от параметров, может быть использован не только как бесключевая хэш-функция, но и как код аутентификации MAC, потоковый шифр или генератор ключевой информации.

Традиционное построение хэш-функций основано на *итерационной конструкции*, последовательно обрабатывающей блоки сообщения *функцией сжатия*, входом которой является результат обработки предыдущего блока. В качестве самого сжатия обычно используется симметричный блочный шифр, имеющий нескольких раундов.

В России действует межнациональный стандарт ГОСТ 34.11–2018 «Информационная технология. Криптографическая защита информации. Функция хэширования». Стандарт определяет две хэш-функции с выходными значениями длиной 256 и 512 бит, обе хэш-функции обрабатывают входную строку M блоками m_i по 512 бит, блоки последовательно выделяются из исходной строки слева: $M = M' \parallel m$, $M = M'$. В случае если длина последнего блока меньше 512 бит, он дополняется до полного размера добавлением вектора (0 … 01) в начало блока.

Для каждого блока m_i выполняется функции сжатия g, результатом которой является обновление внутреннего состояния хэш-функции. Также вычисляется контрольная сумма блоков Σ и число обработанных бит N. На завершающем этапе функцией сжатия обрабатываются значения общей длины сообщения $|M|$ и контрольная сумма. Графически хэш-функция ГОСТ Р 34.11–2012 для сообщения M, состоящего из t блоков после дополнения (m_i, $i = 1, 2, \ldots, t$) может быть представлена в виде итерационной процедуры (рис. 2.27).

На вход хэш-функции подается подлежащее хэшированию сообщение M блоками по 512 бит и вектор инициализации IV, значение которого определено стандартом. Длина вектора инициализации $|IV| = 512$, для хэш-функции с выходом 512 бит $IV = 0^{512}$, с выходом 256 – $IV = (00000001)^{64}$.

Знаком $\boxplus$ обозначается операция сложения в кольце вычетов по модулю 2^{512}. Функция сжатия g основана на 512-битовом блочном симметричном шифре:

$$g(h, m) = E(LPS(h \oplus N), m) \oplus h \oplus m.$$

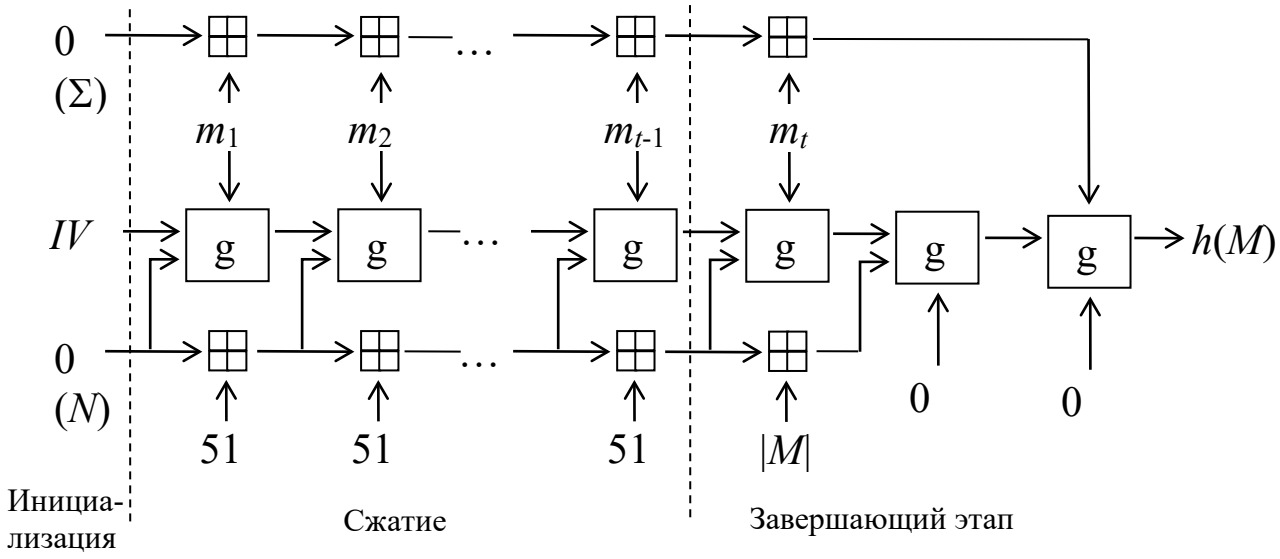

Рис. 2.27. Структура хэш-функции ГОСТ 34.11–2018

В случае если требуется получить хэш-значение длиной 256 бит, в качестве $h(M)$ берутся старшие биты выхода функции g.

E — AES-подобный шифр, состоящий из 12 раундов, в каждом из которых выполняется LPS-преобразование, а затем выполняется наложение раундового ключа K_{i+1}, вычисляемого на основе заданных стандартом констант Ci и внутреннего состояния хэш-функции (результатов обработки хэш-функцией предыдущего блока m_{i-1}). В последнем раунде производится только наложение ключа. $E(K, m) = X[K13]LPSX[K12] \ldots LPSX[K2]LPSX[K1](m)$; $K_i = (K_{i-1} \oplus C_{i-1})$, $i = 2, \ldots, 13$, $K_1 = K$.

X — наложение материала ключа с помощью операции побитового исключающего ИЛИ. Преобразование LPS — композиция следующих трех функций:

S — нелинейная замена, определена так же, как и в шифре «Кузнечик» ГОСТ Р 34.13-2015 (замена производится побайтно с помощью одной таблицы замен);

P — перестановка байт по определенной стандартом таблице, фактически осуществляет транспонирование матрицы байт 8 × 8. Таблица перестановки P = {0, 8, 16, 24, 32, 40, 48, 56, 1, 9, 17, 25, 33, 41, 49, 57, 2, 10, 18, 26, 34, 42, 50, 58, 3, 11, 19, 27, 35, 43, 51, 59, 4, 12, 20, 28, 36, 44, 52, 60, 5, 13, 21, 29, 37, 45, 53, 61, 6, 14, 22, 30, 38, 46, 54, 62, 7, 15, 23, 31, 39, 47, 55, 63};

L — линейное преобразование множества двоичных векторов. 512-битный вход преобразования делится на 8 частей, каждая их которых как 64-битовый вектор b. Преобразование L заменяет каждый вектор b результатом умножения над $GF(2)$ на определенную стандартом матрицу A размером 64×16, содержащую 4-битовые значения. Результатом этой операции будет 64-битовый вектор $c = c_{63} \oplus \ldots \oplus c_1 \oplus c_0$, где c_i , $i = 63, \ldots, 1, 0$ — также 64-битовые вектора,

$$c_i = \begin{cases} 0^{64}, b_i = 0 \\ a_{63-i,15} \| \ldots \| a_{63-i,0}, b_i = 1 \end{cases}, \; a_{ij} \text{ — элементы матрицы } A.$$

После опубликования российского стандарта на хэш-функцию был объявлен конкурс на ее криптоанализ. Исследования показали, что существуют атаки, позволяющие снизить стойкость хэш-функции, однако они далеки от практической реализации.

2.4.3. Код аутентификации HMAC

Конструкция Keyed-Hash Message Authentication Code (HMAC) позволяет совмещать использование односторонних хэш-функций и секретных ключей для подтверждения целостности сообщения и аутентичности его отправителя. Как и при использовании MAC аутентификация обеспечивается знанием секретного ключа.

Первоначально код HMAC был определен в RFC 2104 (1997 г.), позже он был описан американским стандартом FIPS PUB 198-1, первая версия которого вышла в 2002 г. HMAC определен следующим образом:

$$\mathrm{HMAC}(k, M) = h((k_0 \oplus opad) \,||\, h((k_0 \oplus ipad) \,||\, M)),$$

где: || — операция конкатенации (сцепления строк), ⊕ – операция побитового исключающего ИЛИ;

M — сообщение, для которого вычисляется HMAC;

k_0 — секретный ключ, приведенный к длине B входного блока хэш-функции h (в байтах):

— если исходный ключ k длиннее, то он хэшируется $h(k)$, а затем при необходимости дополняется нулевыми байтами справа до длины B: $k_0 = h(k) \,||\, 00\ldots0$,

— если же ключ k — короче, то он дополняется нулевыми байтами справа до длины B: $k_0 = k \,||\, 00\ldots0$;

opad — константа, состоящая из B повторений байта 5C;

ipad — константа, состоящая из B повторений байта 36.

Значения байтов заданы в шестнадцатеричном представлении. Константы *opad* и *ipad* зависят от величины B входного блока используемой хэш-функции h. Стойкость HMAC определяется криптостойкостью используемой хэш-функции h.

HMAC используется в защищенных сетевых протоколах передачи данных, таких как IPSec, TLS и SET.

Конструкция HMAC с хэш-функцией ГОСТ 34.11 определена в Рекомендациях по стандартизации Р 50.1.113–2016 «Информационная технология. Криптографическая защита информации. Криптографические алгоритмы, сопутствующие применению алгоритмов электронной цифровой подписи и функции хэширования».

Итак, во второй главе были рассмотрены современные симметричные криптосистемы, проанализированы свойства современных криптосистем, блочных и потоковых шифров, хэш-функций. В следующей главе рассмотрим криптографические шифры с открытым ключом.

Вопросы для самоконтроля:

1. Что понимается под хэш-функцией? Какие требования предъявляются к криптографически сильным хэш-функциям?

2. Какие бесключевые хэш-функции используются в настоящее время?

3. Какие функции хэширования определены ГОСТ 34.11–2018?

4. Каково назначение кода аутентификации HMAC?

Глава 3
КРИПТОГРАФИЧЕСКИЕ ШИФРЫ С ОТКРЫТЫМ КЛЮЧОМ

§ 3.1. Общие сведения об асимметричных криптосистемах

Наиболее острой проблемой симметричных криптосистем является проблема распространения секретных ключей. Если требуется организовать обмен сообщениями между абонентами, то предварительно стороны должны быть снабжены общими ключами. Причем в процессе передачи ключей должна обеспечиваться их конфиденциальность. Радикальное решение этой проблемы было найдено в середине 1970-х годов американцами У. Диффи и М. Хеллманом и опубликовано ими в ноябре 1976 года в статье «Новые направления в криптографии»[12]. Эта статья заложила основы асимметричной криптографии (криптографии с открытым ключом), использующей пару ключей.

Основополагающим отличием асимметричных криптосистем является использование пары ключей. Причем если один из этих ключей применяется для шифрования сообщения, то расшифрование производится с помощью парного ему ключа. Один из ключей асимметричной пары можно сделать общедоступным, то есть *открытым*, соблюдение его конфиденциальности не требуется. Второй ключ считается *личным ключом* абонента и на всем протяжении своего существования никому не передается и держится своим владельцем в секрете. Таким образом, считается, что криптография с открытым ключом решает проблему доверия между абонентами, поскольку абонент не должен ни с кем разделять свой секрет (личный ключ). Так, при обеспечении конфиденциальности передаваемого сообщения нужно потребовать, чтобы никто, кроме получателя, не мог расшифровать сообщения. Это значит, что сообщение должно быть расшифровано личным ключом получателя. Тогда зашифровать сообщение может любой с использованием известного открытого ключа получателя (рис. 3.1).

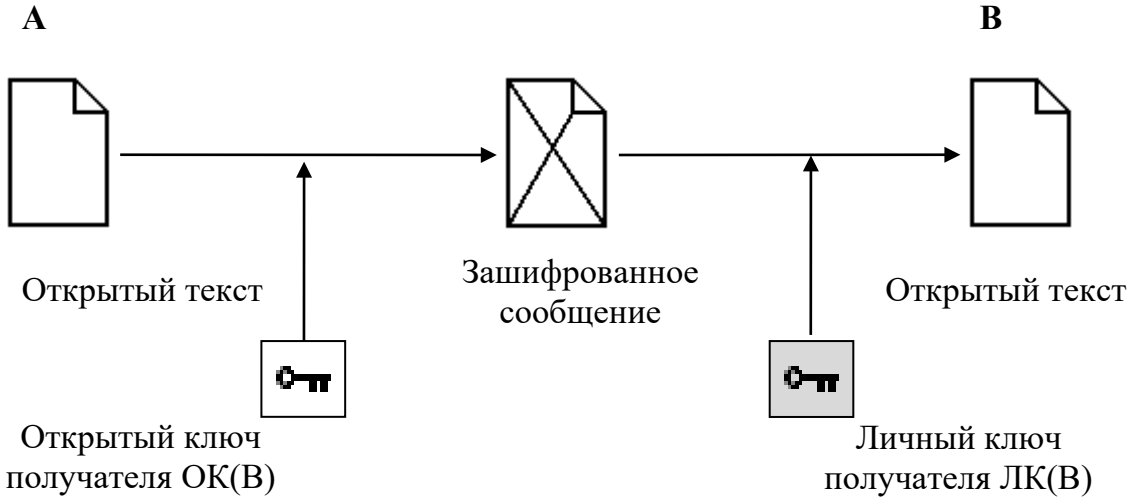

Рис. 3.1. Применение криптосистемы с открытым ключом для решения задачи обеспечения конфиденциальности передаваемых данных

Асимметричные криптосистемы позволяют решать и другие задачи, для которых применение симметричных криптосистем затруднено[13]. Например, для аутентификации

[12] Васильева И. Н., Куватов В. И., Потехин В. С. Криптографическая защита информации.
[13] Молдовян Н. А. Практикум по криптосистемам с открытым ключом. СПб.: БХВ-Петербург, 2007. 304 с.

источника сообщения отправитель должен продемонстрировать знание только одному ему доступного секрета — своего личного ключа. Тогда отправитель может зашифровать сообщение своим личным ключом, а любой другой — провести проверку источника с помощью открытого ключа отправителя (рис. 3.2).

Знание личного ключа позволяет его владельцу легко вычислить открытый ключ. Естественно потребовать, чтобы знание открытого ключа не давало возможности определить парный ему личный ключ.

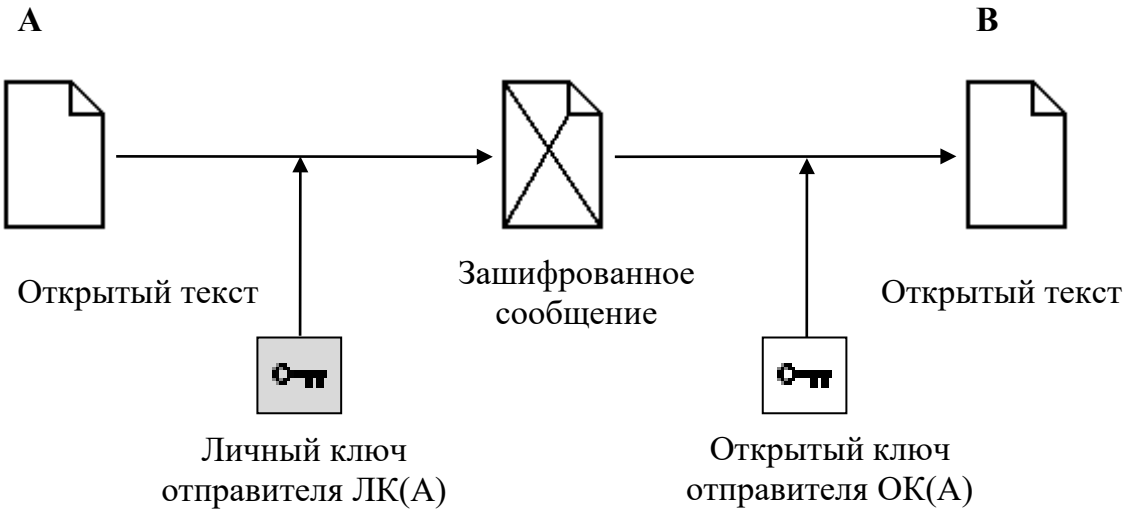

Рис. 3.2. Применение криптографии с открытым ключом для аутентификации источника сообщения

В основе асимметричных криптосистем лежат односторонние (однонаправленные) функции. *Односторонней* называется такая функция $y = f(x)$, которая при известном x легко вычислима, тогда как при известном y вычислить обратную по отношению к ней функцию $x = f^{-1}(y)$ практически невозможно.

Для шифрования информации обычная односторонняя функция не применима, так как никто не сможет расшифровать сообщение, даже зашифровавший его. Обычные односторонние функции нашли применение в парольной аутентификации пользователей ЭВМ, в генераторах псевдослучайных последовательностей и в хэш-функциях. Как было отмечено ранее, генераторы псевдослучайных последовательностей находят применение в поточных шифрах, хэш-функции используются для решения задач контроля целостности и аутентификации, в частности, в технологии цифровых подписей.

В асимметричной криптографии применяются односторонние функции с секретом. *Односторонние функции с секретом* f_Z это такие функции, которые обладают следующим свойством: при известном Z можно найти алгоритмы E_Z и D_Z, позволяющие легко вычислить $y = f_Z(x)$ для всех x и $x = f_Z^{-1}(y)$ для всех y. Если Z — неизвестно, вычислить $y = f_Z(x)$ также легко, однако вычислить $x = f_Z^{-1}(y)$ практически невозможно.

Существование однонаправленных функций до сих пор не доказано. Однако предложено несколько кандидатов, основанных на трудно решаемых задачах теории чисел.

Ниже приведены некоторые из таких задач и функций.

1. Задача дискретного логарифмирования. Первый кандидат в однонаправленные функции — функция дискретного возведения в степень:

$$y = a^x (mod\ p),$$

где p — большое простое число. Даже при очень больших x эта функция легко вычислима. Обратной по отношению к функции дискретного возведения в степень, является функция вычисления дискретного логарифма:

$$x = log_a(y).$$

Известно, что нахождение дискретных логарифмов требует исключительно большого количества вычислений.

2. *Задача факторизации целых чисел.* Второй кандидат в однонаправленные функции — функция $f(x, y) = x \cdot y$, где x и y — простые числа.

Однонаправленность следует из того, что нахождение больших простых чисел и их произведения в вычислительном отношении осуществляется относительно легко, но разложение на множители большего числа практически неосуществимо.

3. *Задача вычисления квадратичного вычета*. В качестве однонаправленной функции рассматривается $f(x) = x^2 \bmod n$, n — произведение двух простых больших чисел.

Эффективное вычисление значения $x = \sqrt{x^2}\ mod\ n$ возможно лишь при известном разложении числа n на сомножители. То есть эта задача сводится к трудной задаче факторизации составного числа n.

Все три вышеприведенных кандидата в однонаправленные функции являются функциями с секретом.

Асимметричные криптосистемы считают системами *с доказуемой стойкостью*, поскольку они изначально строятся таким образом, чтобы решение задачи дешифрования (криптоанализа) сводилось к известной сложной вычислительной задаче (например, факторизации большого числа, квадратичного вычета, либо вычисления дискретного логарифма). При этом строго не доказано, что такие задачи не могут иметь эффективного, то есть полиномиального по времени алгоритма решения. Однако на текущий момент эффективные алгоритмы решения таких задач неизвестны.

Наиболее важным аспектом асимметрических криптосистем является низкая скорость шифрования/расшифрования. Поэтому криптосистемы с открытым ключом, практически не применяются для шифрования сообщений, а используются в основном для шифрования секретных ключей симметричных криптосистем и формирования электронной подписи.

Несмотря на то, что для асимметричных криптосистем не требуется обеспечивать конфиденциальность открытых ключей, при распределении ключей следует обеспечивать их надежную связь с абонентами. В случае отсутствия достоверной связи с абонентом при доставке открытых ключей возможна реализация атаки «человек посередине» (рис. 3.3).

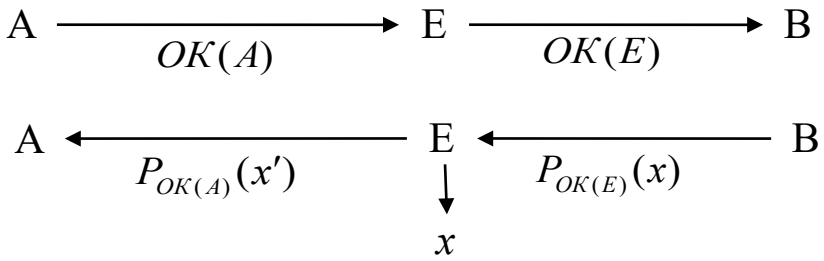

Рис 3.3. Атака «человек посередине» на криптосистему с открытым ключом

При отсутствии аутентифицированного канала взаимодействия между абонентами A и B противник E сможет навязать абоненту B свой открытый ключ вместо открытого ключа истинного получателя A сообщения, и шифрование $P(x)$ сообщений будет вестись на известном противнику ключе. Тогда противник сможет перехватывать и читать открытое содержимое всех сообщений, кроме того, он сможет выдавать себя за любую из сторон, навязывая свои сообщения. Задача обеспечения связности открытых ключей с конкретными абонентами решается использованием *цифровых сертификатов* и развертывания *инфраструктуры открытых ключей* (PKI), основу которой составляет осуществляющая управление ключами *служба сертификации*.

Вопросы для самоконтроля:

1. В чем заключается принципиальное различие симметричных и асимметричных криптографических систем?

2. Каким образом ключи асимметричной пары используются для обеспечения конфиденциальности сообщения, аутентификации источника сообщения?

3. Какие задачи лежат в основе криптографии с открытым ключом?

4. Почему криптосистемы с открытым ключом называют системами с доказуемой стойкостью?

§ 3.2. Система распределения ключей Диффи — Хеллмана (DH)

Для организации системы обмена ключами между абонентами сначала выбираются два больших целых числа p и g, причем p — простое число (порядка 2^{1024} или более), а g таково, что: $1 < g < p - 1$, и все числа из множества $\{1, 2, \ldots, p - 1\}$ могут быть представлены как различные степени g по модулю p. При произвольном p нахождение g может оказаться трудной задачей, связанной с разложением на множители числа $p - 1$. Стойкость криптосистемы Диффи — Хеллмана базируется на вычислительной сложности задачи дискретного логарифмирования. Для обеспечения стойкости системы число $p - 1$ должно содержать большой простой множитель. Поэтому p выбирается как $p = 2q + 1$, где q — также большое простое число. Тогда g — любое число, для которого справедливы неравенства: $1 < g < p - 1$, $g^q \bmod p \neq 1$. Пусть такие числа p и g найдены, это общие открытые параметры системы Диффи — Хеллмана, доступные всем абонентам.

Пусть абонент i хочет установить секретную связь с абонентом j. Абонент i случайным образом выбирает число x_i, $1 \leq x_i \leq p$ и держит его в секрете. Затем он вычисляет $y_i = a^{x_i}(mod\ p)$ и отсылает его абоненту j. То же самое делает и второй абонент.

Тогда абонент i имеет число: $Z_{ij} = y_j^{x_i}(mod\ p) = a^{x_i x_j}(mod\ p)$.

Абонент j располагает числом: $Z_{ji} = y_i^{x_j}(modp) = a^{x_j x_i}(modp)$.

Как видно, $Z_{ij} = Z_{ji}$, обозначим это число Z_{ij}. Поскольку x_i знает только абонент i, а x_j — только абонент j, то Z_{ij} знают только они оба. Следовательно, они могут использовать это число в качестве секретного ключа симметрической криптосистемы.

Алгоритм Диффи — Хеллмана в чистом виде уязвим для классической сетевой атаки «человек посередине» (man in the middle), поэтому схемы с его использованием должны обеспечивать предварительную аутентификацию сторон.

В самом деле, общие параметры системы Диффи — Хеллмана p и g являются открытыми. Противник может сгенерировать свое случайное число z, перехватить сформированное абонентом i значение $y_i = g^{xi} \bmod p$ при его передаче и отослать абоненту j свое значение $y_z = g^z \bmod p$. В ответ абонент j отправит значение $y_j = g^{xj} \bmod p$, которое противник также может подменить на свое значение $y_z = g^z \bmod p$. Тогда противник может использовать ключ $y_i^z \bmod p$ для дальнейшей связи с абонентом i от имени j и ключ $y_j^z \bmod p$ — для связи с j от имени i, читая всю их переписку и при желании подменяя сообщения.

Пример. Пусть двум абонентам (1 и 2) необходимо сформировать общий секретный ключ симметрической криптосистемы. Пусть для всех абонентов выбраны параметры системы Диффи — Хеллмана $g = 2,\ p = 11\ (q = 5, 2^5 mod\ 11 = 10 \neq 1)$. Пусть первый абонент выбрал себе $x_1 = 3$, второй абонент — $x_2 = 5$. Тогда первый абонент отсылает своему контрагенту число $y_1 = 2^3 (mod\ 11) = 8$, второй — число $y_2 = 2^5 (mod\ 11) = 10$. Первый абонент вычисляет $Z_{12} = 10^3 (mod\ 11) = 10$. Второй вычисляет $Z_{21} = 8^5 (mod\ 11) = 10$. Значения Z_{12} и Z_{21} совпадают и могут использоваться как ключ для шифрования сообщений в симметрической криптосистеме.

Вопросы для самоконтроля:

1. Какую задачу решает криптосистема Диффи — Хеллмана? Каким образом выбираются параметры криптосистемы Диффи — Хеллмана?

2. Каким образом абоненты могут сформировать общий секретный ключ?

3. Каковы особенности практического использования криптосистемы Диффи — Хеллмана?

§ 3.3. Системы шифрования с открытым ключом

3.3.1. Криптографическая система RSA

Общие сведения о RSA. Описание RSA было опубликовано Р. Райвестом, А. Шамиром и Л. Адлеманом в августе 1977 года в журнале Scientific American. В основу предложенной авторами криптографической системы легла функция

$$y = x^L (mod\ N),$$

где: x — открытый текст; y — закрытый текст; N — целое число, равное произведению двух различных больших простых чисел P и Q примерно одного порядка; L – целое число в пределах от 2 до N–1, взаимно простое с произведением $(P-1)(Q-1)$.

Напомним, что два числа называются взаимно простыми, если их наибольший общий делитель (НОД) равен 1. Чтобы не проверять значение L на взаимную простоту с произведением $(P-1)(Q-1)$, можно выбирать его среди простых чисел.

Абонент криптосистемы RSA случайным образом выбирает два больших простых числа P и Q, и получает значение $N = PQ$. Затем он также случайным образом выбирает значение L, проверяя условие НОД(L, (P–1)(Q–1)) = 1. Пара значений (N, L) публикуется в качестве открытого ключа абонента.

Любой другой пользователь, зная N и L получателя, может зашифровать свое сообщение x по формуле

$$y = x^L \pmod N.$$

Блоки открытого текста, как и криптограммы, в системе RSA представляются целыми числами от 0 до N–1.

Расшифровать криптограмму y может только тот, кто знает разложение числа N на множители P и Q. Это позволяет вычислить значение функции Эйлера $\varphi(N) = (P-1)(Q-1)$, которая показывает количество целых положительных чисел от 1 до N, взаимно простых с N.

В соответствии с теоремой, если P и Q — простые числа, $P \neq Q$ и k — произвольное целое число, то $x^{k\varphi(pq)+1} \bmod pq = x$. Обозначив $PQ = N$, имеем $x^{k\varphi(N)+1} \bmod N = x$.

Зная значение $\varphi(N)$, с помощью расширенного алгоритма Евклида можно вычислить целое число d, такое, что $L \cdot d = 1 \pmod{\varphi(N)}$ или, что то же самое, $L \cdot d = K\varphi(N) + 1$. При выполнении условия НОД(L, φ(N)) = 1 число d существует и единственно. Число d является личным (закрытым) ключом абонента. Для того чтобы расшифровать присланную ему криптограмму y, он должен вычислить значение

$$y^d \pmod N = x.$$

Действительно, $y = x^L \pmod N$, $y^d = (x^L)^d = x^{Ld} = x^{K\varphi(N)+1} = x \pmod N$.

Из изложенного выше материала следует, что личный ключ абонента легко получить при известных P и Q. Поэтому на практике после генерации открытого им личного ключа абонента значения P, Q, φ(N) уничтожаются для предотвращения попыток их эксплуатации для взлома криптосистемы. Задача нахождения d по открытому ключу (N, L) без знания φ(N) (P, Q) практически неосуществима.

Пример. Пусть абонент системы RSA выбрал в качестве секретных параметров два простых числа $P = 3$, $Q = 5$, тогда $N = 3 \cdot 5 = 15$. Он находит значение функции Эйлера φ(15) = (3–1)(5–1) = 8, затем выбирает случайным образом значение $L = 3$ и проверяет выполнение равенства НОД(L, N) = НОД(3, 8) =1. В качестве своего открытого ключа абонент публикует пару (N, L), то есть значения (8, 3). Абонент вычисляет свой личный ключ d из соотношения Ld (mod φ(N)) = 1 или $d = (K\varphi(N) + 1)/L = (K \cdot 8 + 1)/3$, при $K = 1$ получаем $d = (8 + 1)/3 = 9/3 = 3$. В общем случае значение d находится с помощью расширенного алгоритма Евклида. Проверим, Ld (mod φ(N)) = 1. Действительно, $3 \cdot 3 \bmod 8 = 9 \bmod 8 = 1$.

Пусть теперь абоненту хотят передать число $x = 792$. Оно разбивается на блоки $x_i \leq N$–1, то есть $x_i \leq 14$. Таким образом, отправитель должен зашифровать числа открытым ключом $L = 3$ числа $x_1 = 7$, $x_2 = 9$, $x_3 = 2$:

$$y_1 = 7^3 \bmod 15 = 13,\ y_2 = 9^3 \bmod 15 = 9,\ y_3 = 2^3 \bmod 15 = 8,$$

таким образом, абоненту будут переданы криптограммы 13, 9 и 8.

Абонент расшифровывает их своим личным ключом $d = 3$:

$$y_1 = 13^3 \bmod 15 = 7,\ y_2 = 9^3 \bmod 15 = 9,\ y_3 = 8^3 \bmod 15 = 2,$$

получая блоки исходного текста 7, 9, 2.

Стойкость криптосистемы RSA определяется вычислительной сложностью задачи разложения на сомножители составного числа *N*, полученного в результате перемножения больших простых чисел *P* и *Q*.

Вопросы стойкости криптосистемы RSA. Система RSA может быть вскрыта, если удастся найти значения *P* и *Q*, то есть решить *задачу факторизации* числа *N*. Поэтому простые числа *P* и *Q* следует выбирать так, чтобы факторизация *N* потребовала несоразмерно много времени. В настоящее время задача факторизации не имеет полиномиального решения, разработаны лишь некоторые алгоритмы, ее упрощающие. Сложность решения задачи факторизации числа *N* можно оценить по формуле $O\left(e^{\sqrt{\ln(N)-\ln\ln(N)}}\right)$.

К началу 2010 года с помощью *метода решета числового поля* удалось факторизовать 768-битное число, а в 2020 году длина таких чисел достигла 829 битов. Следовательно, безопасность разрядность *N* должна быть больше. В настоящее время рекомендовано применение 2048-битовых значений модуля криптосистемы RSA. Заметим, что для обеспечения эквивалентной вычислительной стойкости 64-битному ключу симметричного шифрования примерно соответствует 512-битный ключ RSA, а 128-битному — ключ RSA длиной более 2300 бит.

Известен ряд атак на RSA, связанных с неверным выбором параметров этой криптосистемы. Рассмотрим две подобные криптоатаки.

Выбор малой длины открытого ключа (экспоненты) *L*. Ключ малой длины позволяет ускорить процедуру шифрования, однако делает потенциально возможной реализацию атаки на основе китайской теоремы об остатках (при условии, что пользователи используют одно и то же значение *L*). Для реализации атаки потребуется *L* отправленных разным пользователям криптограмм, соответствующих одному и тому же сообщению *x*. Пусть *L* = 3. И пусть одно и то же сообщение посылается трем разным пользователям, имеющим разные значения *N*. В этом случае можно восстановить это сообщение не зная ключа. Покажем это. Рассмотрим передаваемые криптограммы как систему уравнений:

$$y_1 = x^3 \ mod \ N_1,$$

$$y_2 = x^3 \ mod \ N_2,$$

$$y_3 = x^3 \ mod \ N_3.$$

Пусть НОД (N_i, N_j)=1, для $i \neq j$.

В соответствии с китайской теоремой об остатках эта система уравнений имеет решение $z = x^3 \ mod \ (N_1 \cdot N_2 \cdot N_3)$, которое может быть найдено за полиномиальное время. Напомним, что полиномиальная сложность (полиномиальное время) решения задачи означает, что для решения задачи требуется $O(n^C)$ операций, где *n* — длина входного слова, а *C* — константа. А так как $z < N_1 \cdot N_2 \cdot N_3$, то расшифрование выполняется по формуле $x = \sqrt[3]{z}$.

Для защиты от этой атаки необходимо использовать большие значения *L*, либо исключить отправку одних и те же сообщений разным пользователям. Если последнее все же необходимо, то одни и те же сообщения, посылаемые разным пользователям, требуется «подсаливать», например, добавляя в их конец разные (случайные) числа.

Кроме того, при использовании малых (даже различных) значений L достаточно большое число открытых сообщений, удовлетворяющих неравенству $x < \sqrt[L]{N}$, будут зашифровываться простым возведением в степень $y = x^e < N$ и поэтому их можно найти путем извлечения корня степени L из y.

Атака при малом числе возможных сообщений. Пусть число сообщений ограничено значениями $x_1, x_2, \ldots, x_r$, где r — обозримо по величине. Это могут быть, например, различные команды: вперед, назад, вправо, влево. Пусть криптограмма y перехвачена. В этом случае необходимо все команды зашифровать известным открытым ключом и определить ту команду, которая совпадает с принятой y.

$$\left.\begin{array}{l} y_1 = x_1^L \bmod N \\ y_2 = x_2^L \bmod N \\ \ldots\ldots \\ y_r = x_r^L \bmod N \end{array}\right\} = y$$

Способ борьбы с такой атакой — «подсаливание» сообщений.

Использование малых значений личного ключа d ($d < N^{0,292}$) позволяет эффективно определять значение d с помощью атаки, предложенной М. Винером. Поэтому для 1024-битного RSA значение d должно иметь длину порядка 300 бит и более.

В случае если пользователи применяют общий модуль N и разные взаимно простые значения L, возможна атака методом *бесключевого чтения*. Другой случай — если выбраны относительно близкие значения P и Q, что делает возможным их определение *методом Ферма*.

Таким образом, вычислительная стойкость RSA во многом определяется правильным выбором параметров криптосистемы. Так, длина N должна быть достаточно большой (2048 бит и более), абоненты криптосистемы не должны использовать одинаковые значения N или L (особенно если L мало), а также малые значения L и d. Чтобы исключить возможность применения методов факторизации, выбираемые значения P и Q должны удовлетворять следующим условиям: числа P–1, P+1, Q–1, Q+1 не должны разлагаться в произведение малых простых множителей, и должны содержать в качестве сомножителя хотя бы одно большое простое число; числа $P_1 = \frac{P-1}{2}, P_2 = \frac{P+1}{2}, Q_1 = \frac{Q-1}{2}, Q_2 = \frac{Q+1}{2}$ должны быть простыми (при выполнении этого условия выполнено и первое условие), причем P_1–1 и Q_1–1 не должны разлагаться в произведение малых простых множителей; P и Q не должны содержаться в списках известных больших простых чисел; значения P и Q не должны быть близкими, в то же время порядок обоих чисел должен быть достаточно велик для исключения подбора значений.

3.3.2. Криптографическая система Эль-Гамаля

Криптосистема Эль-Гамаля предложена в 1985 году и фактически базируется на алгоритме Диффи-Хеллмана. Криптографическая стойкость данной системы основана на сложности проблемы дискретного логарифмирования (логарифмирования в мультипликативной группе конечного простого поля). Общие открытые параметры криптосистемы p и g выбираются так же, как и в системе Диффи-Хеллмана: p — достаточно большое простое число, g – порождающий элемент мультипликативной группы Z_p^* целых чисел по модулю p, $1 < g < p$–1. Рекомендуется также, чтобы число p–1 содержало бы большой простой делитель.

Каждый абонент выбирает свой личный ключ — случайное число a_i, $1 < a_i < p-1$, которое держится в секрете. Затем он вычисляет и публикует свой открытый ключ $b_i = g^{x_i} \bmod p$ в справочнике. Если кто-либо желает послать абоненту i зашифрованное сообщение, он разбивает его на блоки, которые представляются целыми числами x от 0 до $p-1$.

Для каждого блока x_i выбирается свое случайное значение k, $0 < k << p-1$. Затем отправитель извлекает из справочника открытый ключ получателя b_i и вычисляет части криптограммы $y_1 = g^k \pmod p$ и $y_2 = x \cdot b_i^k \pmod p$. Криптограмма, представленная парой (y_1, y_2) пересылается получателю — абоненту i.

Получатель расшифровывает криптограмму с использованием своего личного ключа a_i: $x = y_2 \cdot y_1^{p-1-a_i} (\bmod p)$.

Действительно, p — простое число, тогда согласно *малой теореме Ферма*, $g^{p-1} \pmod p = 1$ для любого целого g, $0< g < p$, тогда и $g^{k(p-1)} \pmod p = 1$ и $y_2 \cdot y_1^{p-1-a_i} = (x \cdot b_i^k)(g^k)^{p-1-a_i} = x \cdot g^{a_i k + k(p-1) - k a_i} = x \cdot g^{k(p-1)} = x (\bmod p)$.

Введение в правило шифрование случайного значения k позволяет получать различные криптограммы даже для одинаковых блоков открытого текста. Таким образом, криптосистема Эль-Гамаля является *рандомизированной* или *вероятностной*. Необходимо использовать различные значения k для шифрования различных блоков открытого текста. Вероятностный характер шифрования можно отнести к достоинствам системы Эль-Гамаля (схемы вероятностного шифрования обладают, как правило, большей стойкостью по сравнению со схемами с детерминированным процессом шифрования). Недостаток системы – удвоение длины открытого текста при шифровании.

Как уже отмечалось, стойкость системы Эль-Гамаля определяется сложностью решения задачи дискретного логарифмирования в Z_p^*. Методы криптоанализа имеют субэкспоненциальную сложность при правильном выборе параметров криптосистемы. В настоящее время эта задача практически нереализуема для значений p, содержащих не менее 150 десятичных знаков (или порядка 500 битов и более). Рекомендуемой длиной модуля p в криптосистемах, стойкость которых основана на сложности задачи дискретного логарифмирования, — не менее 1024 бит.

Система Эль-Гамаля может быть обобщена для применения в любой конечной циклической группе G. Криптографическая стойкость такой обобщенной схемы определяется сложностью задачи логарифмирования в группе G. При этом групповая операция в G должна быть легко реализуемой. В качестве G чаще всего выбираются следующие три группы: мультипликативная группа Z_p^* целых чисел по модулю простого числа p; мультипликативная группа $GF(2^m)^*$ конечного поля $GF(2^m)$ характеристики 2; аддитивная группа точек эллиптической кривой над конечным полем.

Возможное появление в ближайшем будущем производительных квантовых компьютеров позволит за счет использования квантового алгоритма Шора снизить вычислительную стойкость криптосистем с открытым ключом до квадратичной оценки. Это фактически означает катастрофическую потерю стойкости. Поэтому в настоящее время ведется активная работа по поиску новых алгоритмов *постквантовой криптографии* с открытым ключом, устойчивых к атакам с применением квантовых компьютеров.

Вопросы для самоконтроля:

1. *Что является личным и открытым ключами абонентов системы RSA?*
2. *Как производится шифрование и расшифрование в системе RSA?*

3. Как реализуется атака на RSA при малом числе возможных сообщений? Каковы меры противодействия ей?

4. Как производится шифрование и расшифрование в системе Эль-Гамаля?

§ 3.4. Технология электронной подписи

3.4.1. Общие сведения о цифровой подписи

Появление криптографии с открытым ключом позволило решать задачи, которые ранее считались неразрешимыми. Задачи обеспечения целостности информации, обеспечения аутентификации сообщений и абонентов, обеспечения невозможности отказа сторон от авторства легко решаются в асимметричных криптосистемах с помощью технологии электронной подписи.

Под *электронной подписью* (*цифровой подписью*, *электронно-цифровой подписью*) понимается цифровой аналог собственноручной подписи абонента. Любая, в том числе и цифровая, подпись должна удовлетворять следующим требованиям:

— подлинность подписи можно проверить;

— подпись нельзя подделать (данную подпись может поставить только ее обладатель и никто другой);

— подпись является неотъемлемой частью документа и не может быть перенесена в другой документ;

— после подписания документ не подлежит никаким изменениям;

— автор подписи не может от нее отказаться.

Электронная подпись является числом (парой чисел), зависящим от подписываемого сообщения и от секретного ключа, известного только подписывающему субъекту. При этом подпись должна легко проверяться без знания секретного ключа. При возникновении спорной ситуации, связанной с отказом подписавшего от факта подписи им некоторого сообщения либо с попыткой подделки подписи, третья независимая сторона (судья или арбитр) должна иметь возможность разрешить спор.

Как показано на рис. 3.2, цифровые подписи могут быть реализованы с помощью асимметричных криптосистем путем шифрования сообщения личным ключом отправителя. Однако в силу низкой производительности асимметричных криптосистем и возможной большой длины подписи такой подход неэффективен. Поэтому криптографическому преобразованию подвергается не само сообщение, а лишь его контрольная сумма, однозначно характеризующая содержимое этого сообщения. В качестве такой контрольной суммы выступает значение бесключевой хэш-функции. Причем требование на отсутствие коллизий хэш-функции лежит в основе обеспечения безопасности технологии цифровой подписи.

Применяемые на практике технологии электронной подписи базируются на совместном использовании бесключевых хэш-функций и криптосистем с открытым ключом (цифровые подписи RSA, Эль-Гамаля, Шнорра). Современный отечественный стандарт электронной подписи ГОСТ 34.10–2018 «Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи» использует хэш-функцию ГОСТ 34.11–2018 совместно с асимметричным криптоалгоритмом, фактические являющимся вариантом цифровой подписи Эль-Гамаля, на эллиптических кривых.

На практике возможно совместное использование шифрования и электронной подписи сообщений. При этом порядок операций определяется реализацией защищенного сетевого протокола (например, S/MIME) или выбором пользователя. Если

первым выполняется шифрование, подпись может быть проверена без вскрытия защищенной оболочки («конверта») сообщения. Такой порядок предпочтителен в системах, использующих автоматическую проверку подписей. В этом случае получатель сможет удостовериться, что блоки шифртекста не были изменены при передаче, однако явная связь между открытым текстом присланного сообщения и отправителем отсутствует. В случае использования обратного порядка операций получатель сможет удостовериться в подлинности открытого текста сообщения, но не сможет проверить источник и целостность полученной криптограммы (рис. 3.4).

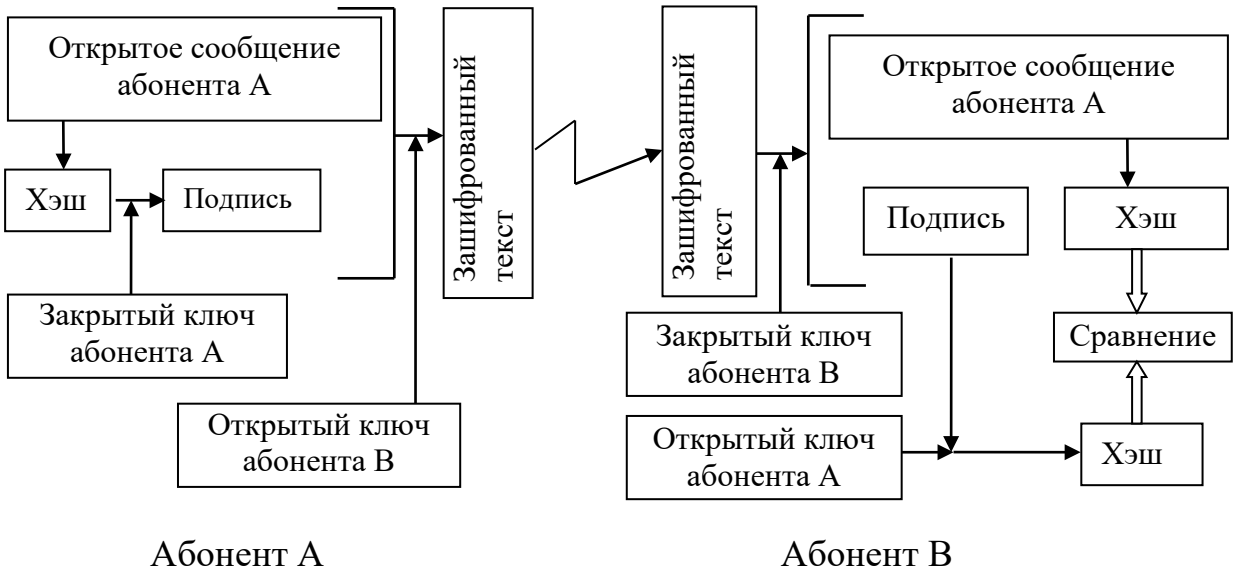

Рис. 3.4. Защищенное сообщение с цифровой подписью

На практике, как правило, используется второй вариант, то есть сообщение сначала подписывается, а затем зашифровывается для определенного получателя.

3.4.2. Электронная подпись на основе RSA

Алгоритм обмена двух абонентов подписанными сообщениями с помощью алгоритма RSA состоит из следующих шагов (рис. 3.5):

— отправитель А составляет документ M и находит его хэш-значение $h=h(M)$ при помощи однонаправленной хэш-функции;

— отправитель А шифрует вычисленное хэш-значение своим личным ключом (ЛК(А)), тем самым ставя под документом свою подпись $E_h = P_{ЛК(А)}(h)$;

— зашифрованное хэш-значение E_h вместе с документом отправляется получателю В;

— получатель В, используя ту же однонаправленную функцию, вычисляет хэш-значение принятого документа $h=h(M^*)$, где M^* — принятое сообщение;

— получатель В, используя открытый ключ отправителя А (ОК(А)), расшифровывает принятое хэш-значение, $h^*=P_{ОК(А)}(E^*_h)$, где E^*_h — принятое зашифрованное хэш-значение.

Если вычисленное и расшифрованное хэш-значения совпадают $h^*=h$, то подпись отправителя под документом верна.

Рис. 3.5 иллюстрирует алгоритм обмена открытым сообщением M и его электронной подписью S. Опишем формально систему электронной подписи на основе RSA. Пусть абонент А хочет отправить подписанное сообщение M получателю В.

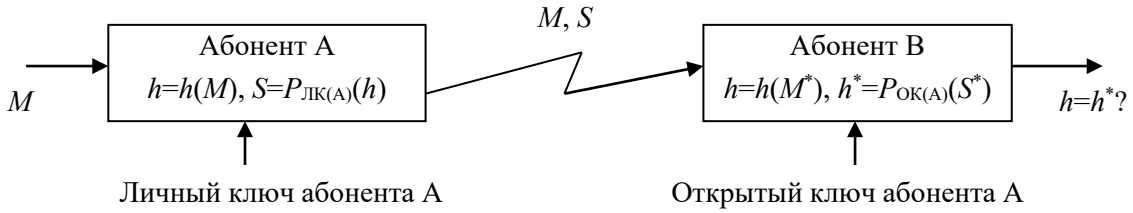

Рис. 3.5. Цифровая подпись открытого сообщения в криптосистеме RSA

Так же, как и в шифре RSA, абонент выбирает параметры: два больших простых числа P, Q, вычисляет $N = PQ$ и $\varphi(N)$, затем выбирает число L, взаимно простое с $\varphi(N)$, и вычисляет $d = L^{-1} \bmod \varphi(N)$. Абонент публикует свой открытый ключ — пару чисел (N, L), — для того, чтобы любой смог проверить его подпись.

1. Подписываемое сообщение делится на блоки $M = m_1, \ldots, m_n$ (в соответствии с используемым алгоритмом хэширования), и вычисляется значение хэш-функции $h = h(M) = h(m_1, \ldots, m_n)$. Значение h должно удовлетворять неравенству $0 < h < N$. Алгоритм вычисления хэш-значения считается общеизвестным.

2. Отправитель А вычисляет $s = h^d \bmod N$, то есть шифрует хэш-значение своим личным (закрытым) ключом. Значение s и есть подпись сообщения M.

3. Подпись s и сообщение M передаются получателю.

Теперь каждый, кто знает открытый ключ отправителя (N, L) может проверить подлинность подписи. Для этого необходимо:

1. Взять подписанное сообщение (M, s) и вычислить $h(M)$;

2. Вычислить число $h^* = s^L \bmod N$ и проверить выполнение равенства $h^* = h(M)$. Если подпись подлинная, то $h^* = h(M)$.

Действительно, из свойств схемы RSA следует: $h^* = s^L \bmod N = h^{dL} \bmod N = h$.

Пример. Пусть абонент выбрал параметры RSA: $P = 5$, $Q = 11$, тогда $N = 55$, $\varphi(N) = 4 \cdot 10 = 40$. Пусть выбрано $L = 3$, НОД(3, 40) = 1. Абонент публикует свой открытый ключ (55, 3). С помощью расширенного алгоритма Евклида он вычисляет свой личный ключ $d = 3^{-1} \bmod 40 = 27$.

Пусть хэш-значение сообщения $h(M) = 13$. Отправитель вычисляет $s = h^d \bmod N = 13^{27} \bmod 55 = 7$ и формирует сообщение $(M, 7)$.

Теперь любой, кто знает открытый ключ отправителя (55, 3), может проверить подлинность подписи под сообщением M. Получатель подписанного сообщения заново вычисляет значение хэш-функции $h(M) = 13$ и число $h^* = s^L \bmod N = 7^3 \bmod 55 = 13$. Поскольку значения h^* и $h(M)$ совпали, подпись верна.

Цифровая подпись на основе системы RSA является *детерминированной*, это значит, что если абонент дважды подпишет одно и то же сообщение, будут получены одинаковые подписи. Это свойство может использоваться в атаках повторной передачи сообщения. Кроме того, поскольку подпись фактически представляет собой зашифрованное на личном ключе абонента известное значение (значение бесключевой хэш-функции легко может быть вычислено любым желающим), возможны атаки на личный ключ на основе известного открытого текста.

3.4.3. Электронная подпись на основе схемы Эль-Гамаля

Криптосистема Эль-Гамаля позволяет рандомизировать значение подписи за счет использования случайного значения (разового личного ключа) k. Кроме того, хэш-значение не передается в явном виде. Однако для проверки подписи потребуется подсказка r (связанный с k разовый открытый ключ). Поэтому длина подписи увеличивается вдвое — подпись представляет собой уже не одно число, как в RSA, а пару чисел r, s.

Пусть отправитель А собирается подписывать документы. А выбирает большое простое число p и число g, $1 < g < p-1$, все числа из множества $\{1, 2,\ldots, p-1\}$ могут быть представлены как различные степени g по модулю p. Выбор значений p и g осуществляется так же, как и в системе Диффи — Хеллмана. Затем А выбирает личный ключ — случайное число x, $1 < x < p-1$, которое держит в секрете.

Отправитель вычисляет значение $y = g^x \bmod p$, которое публикует как свой открытый ключ.

1. Отправитель вычисляет хэш-значение сообщения $h = h(M)$. Значение h должно удовлетворять неравенству $0 < h < p$.

2. Далее отправитель выбирает случайное число k, $0 < k < p-1$, взаимно простое с $p-1$, и вычисляет числа:

$r = g^k \bmod p$,

$u = (h - xr) \bmod (p-1)$,

$s = k^{-1}u \bmod (p-1)$,

k^{-1} — число, обратное k по модулю $p-1$, так как k и $p-1$ — взаимно просты k^{-1} существует и единственно.

3. Подпись (r, s) добавляется к сообщению, и тройка (M, r, s) передается получателю.

4. Получатель заново вычисляет хэш-значение присланного сообщения $h = h(M)$ и проверяет подпись, используя равенство:

$y^r r^s \pmod p = g^h \pmod p$.

Если подпись верна, то это равенство выполнено. Действительно:

$$y^r r^s = (g^x)^r (g^k)^s = g^{xr} g^{kk^{-1}(h-xr)} = g^{xr} g^h g^{-xr} = g^h \pmod p .$$

Существуют различные варианты алгоритма электронной подписи Эль-Гамаля, различающиеся, в частности, выбором начальных параметров, формулой для расчета значения s и, соответственно, видом проверочного равенства. Целью этих модификаций является повышение эффективности вычислений за счет использования «коротких» показателей степени. Именно они являются основой стандартов электронной подписи: DSA (ECDSA) и версии отечественного ГОСТ Р 34.10.

Зная открытые ключи абонентов, практически невозможно вычислить соответствующие им секретные ключи в силу вычислительной сложности задачи дискретного логарифмирования при больших значениях p.

Пример. Пусть имеются общие параметры группы абонентов системы Эль-Гамаля $p = 23$, $g = 5$. Отправитель А выбирает свой личный (секретный) ключ $x = 7$, вычисляет и публикует открытый ключ $y = g^x \bmod p = 5^7 \bmod 23 = 17$.

А хочет подписать документ M. Пусть А вычислил значение хэш-функции этого документа $h(M) = 3$. А генерирует случайное число k, пусть выбрано $k = 5$. Затем А вычисляет:

$r = g^k \bmod p = 5^5 \bmod 23 = 20$,

$u = (h - xr) \bmod (p - 1) = (3 - 7\cdot 20) \bmod 22 = 17$.

С помощью расширенного алгоритма Евклида А получает: $k^{-1} \bmod 22 = 5^{-1} \bmod 22 = 9$, $s = k^{-1}u \bmod (p - 1) = 9\cdot 17 \bmod 22 = 21$. А формирует подписанное сообщение в виде $(M, 20, 21)$ и передает его получателю В.

Получатель В заново вычисляет значение хэш-функции $h(M) = 3$, вычисляет:

— левую часть проверочного равенства: $y^r r^s \bmod p = (17^{20}\cdot 20^{21}) \bmod 23 = 16\cdot 15 \bmod 23 = 10$;

— правую часть проверочного равенства: $g^h \bmod p = 5^3 \bmod 23 = 10$;

— затем В проверяет: 10=10, значит, подпись верна.

Рассмотренный метод электронной подписи сложнее, чем RSA, а его стойкость базируется на другой функции. Цифровая подпись Эль-Гамаля является *рандомизированной* (*вероятностной*). Это значит, что если отправитель дважды подпишет одно и то же сообщение, подписи окажутся различными. Такое свойство подписи Эль-Гамаля обеспечивается за счет участия в процессе подписания случайного значения k, которое каждый раз генерируется заново.

3.4.4. Атаки на схемы электронной цифровой подписи

Стойкость схемы электронной подписи зависит от стойкости используемых криптоалгоритмов и хэш-функций и определяется относительно пары угроза-атака. Возможна следующая классификация атак на схемы электронной подписи:

— атака на основе известного открытого ключа (key-only attack) — самая слабая из атак, практически всегда доступная противнику;

— атака на основе известных подписанных сообщений (known-message attack) — в распоряжении противника имеется некоторое число пар сообщение — допустимая подпись на него, при этом противник не может влиять на выбор сообщений;

— простая атака с выбором подписанных сообщений (generic chosen-message attack) — противник имеет возможность выбрать некоторое количество подписанных сообщений, при этом открытый ключ он получает после этого выбора;

— направленная атака с выбором сообщений (directed chosen-message attack) — выбирая подписанные сообщения, противник знает открытый ключ;

— адаптивная атака с выбором сообщений (adaptive chosen-message attack) — противник знает открытый ключ, выбор каждого следующего подписанного сообщения он может делать на основе знания допустимой подписи предыдущего выбранного сообщения.

Каждая атака направлена на достижение определенной цели. Можно выделить следующие виды угроз для схем электронной подписи (в порядке возрастания силы):

— экзистенциальная подделка (existential forgery) — создание противником подписи для какого-нибудь, возможно бессмысленного, сообщения M', отличного от перехваченного: $M' \neq M$;

— селективная подделка (selective forgery) — создание подписи для заранее выбранного сообщения;

— универсальная подделка (universal forgery) — нахождение эффективного алгоритма формирования подписи, функционально эквивалентного исходному;

— полное раскрытие (total break) — вычисление ключа, возможно, отличного от личного ключа абонента (ЛК(А)), но соответствующего открытому ключу абонента (ОК(А)), что дает возможность формировать подписи для любых сообщений.

Наиболее надежными являются схемы цифровой подписи, стойкие против самой слабой из угроз на основе самой сильной из атак, то есть против экзистенциальной подделки на основе атаки с выбором подписанных сообщений.

Справедливо следующее утверждение. Схемы электронной подписи, стойкие против экзистенциальной подделки на основе атаки с ювыбором подписанных сообщений, существуют тогда и только тогда, когда существуют односторонние функции.

3.4.5. Цифровые сертификаты

Для создания надежной линии передачи сообщений с помощью асимметричной схемы шифрования между отправителем и получателем необходимо обеспечить доверенную передачу открытых ключей, поскольку в противном случае возможна реализация классиче-

ской сетевой атаки «человек посередине» (man in the middle). Подменив передаваемые ключи, злоумышленник-посредник может представить себя как отправителем подписанных данных, так и получателем зашифрованных сообщений. Поэтому одной из наиболее серьезных проблем при использовании методов криптографии с открытым ключом является проверка, действительно ли открытый ключ принадлежит абоненту, с которым планируется вести обмен зашифрованными сообщениями. Эта проблема решается с помощью методов цифровой сертификации.

Цифровой сертификат — электронный документ, который связывает открытый ключ с определенным пользователем или приложением. Информация сертификата подтверждает истинность открытого ключа и владельца соответствующего личного ключа.

Наиболее распространенным стандартом цифровых сертификатов является X.509, хотя используются сертификаты и других форматов. Некоторые из них связаны с определенными приложениями (например, PGP), другие — с определенными сетевыми службами и протоколами (такие как SET и IPSec). Физическое представление ключей зависит от конкретной реализации системы, поддерживающей использование криптографии с открытым ключом (рис. 3.6).

Функции выдачи, отзыва и управления цифровыми сертификатами берет на себя *служба сертификации* (удостоверяющий центр, центр сертификации, Certification Authority — CA). Наряду с удостоверяющими центрами государственных структур существуют и коммерческие службы сертификации.

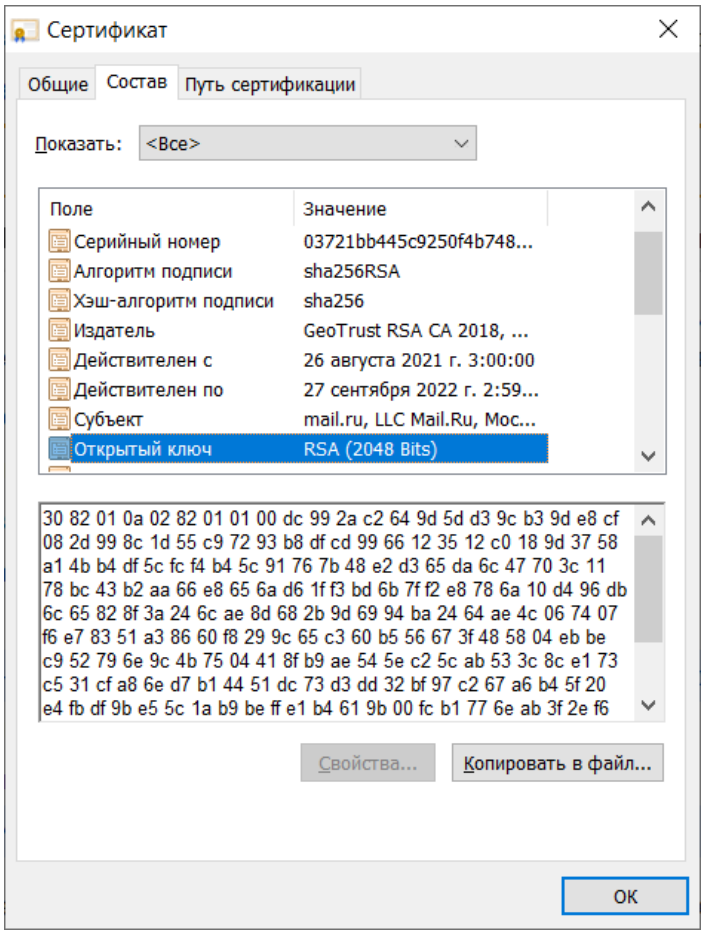

Рис. 3.6. Пример вывода информации сертификата в формате X.509

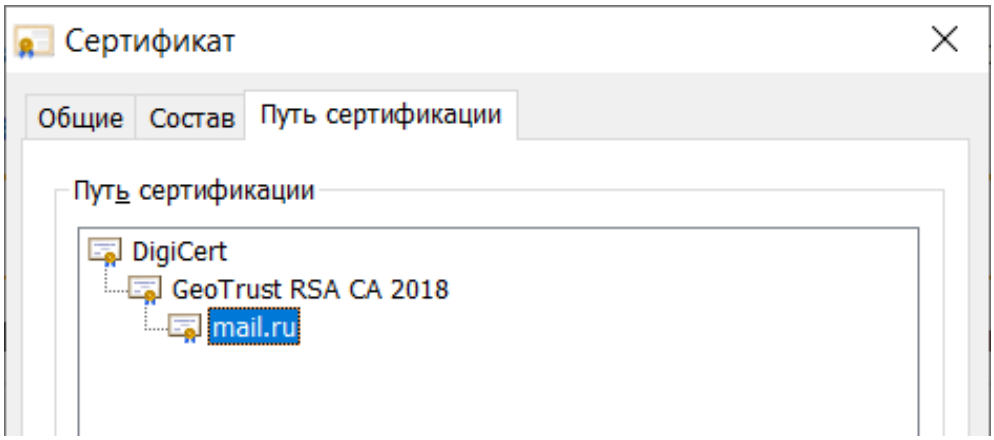

Рис. 3.7. Пример вывода информации о пути сертификации

Служба сертификации упаковывает передаваемый ключ абонента вместе с именем владельца ключа и набором специальных удостоверений, подтверждающих его истинность, а затем скрепляет весь подготовленный таким образом пакет своей цифровой подписью.

Центры сертификации получают сертификаты своих ключей у центров более высокого уровня (рис. 3.7). В пакет включается и удостоверение, подтверждающее полномочия самой службы сертификации, а также ее идентификатор. Таким образом, служба сертификации выступает в качестве гаранта истинности связи между открытым ключом субъекта и идентифицирующей этот субъект информацией, то есть позволяет соотнести открытые ключи с их владельцами.

Служба сертификации должна пользоваться доверием сторон, обменивающимися сообщениями, которые добровольно передают ей полномочия на проверку аутентичности отправителей сообщений. Если абоненты сети пользуются разными службами сертификации, нужно последовательно пройти через несколько служб, образующих иерархическую «цепь доверия», пока в ней не встретится узел, общий для ветвей обеих сторон. Каждая служба сертификации получает полномочия и «сертификат доверия» от службы сертификации более высокого уровня. Между корневыми службами сертификации устанавливаются взаимные отношения доверия за счет использования сертификатов специального вида. Как правило, компьютерное программное обеспечение имеет встроенный список доверенных корневых сертификатов, что позволяет проводить проверку подлинности сертификатов более низких иерархических уровней (рис. 3.8).

Структура, позволяющая предоставлять доверенные действительные открытые ключи участников и управлять всем жизненным циклом цифровых сертификатов получила название инфраструктуры открытых ключей (PKI, Public Key Infrastructure). *Инфраструктура открытых ключей* (PKI) — полный комплекс программно-аппаратных средств, а также организационно-технических мероприятий, необходимых для использования технологий шифрования с открытыми ключами и электронной подписи.

PKI обеспечивает взаимодействие между службой сертификации и конечными пользователями. Она включает: центр сертификации (удостоверяющий центр), осуществляющий генерацию ключей и выдачу сертификатов, централизованную базу данных действительных и отозванных сертификатов, регистрирующий центр, непосредственно обрабатывающий запросы пользователей, соответствующие данные и протоколы.

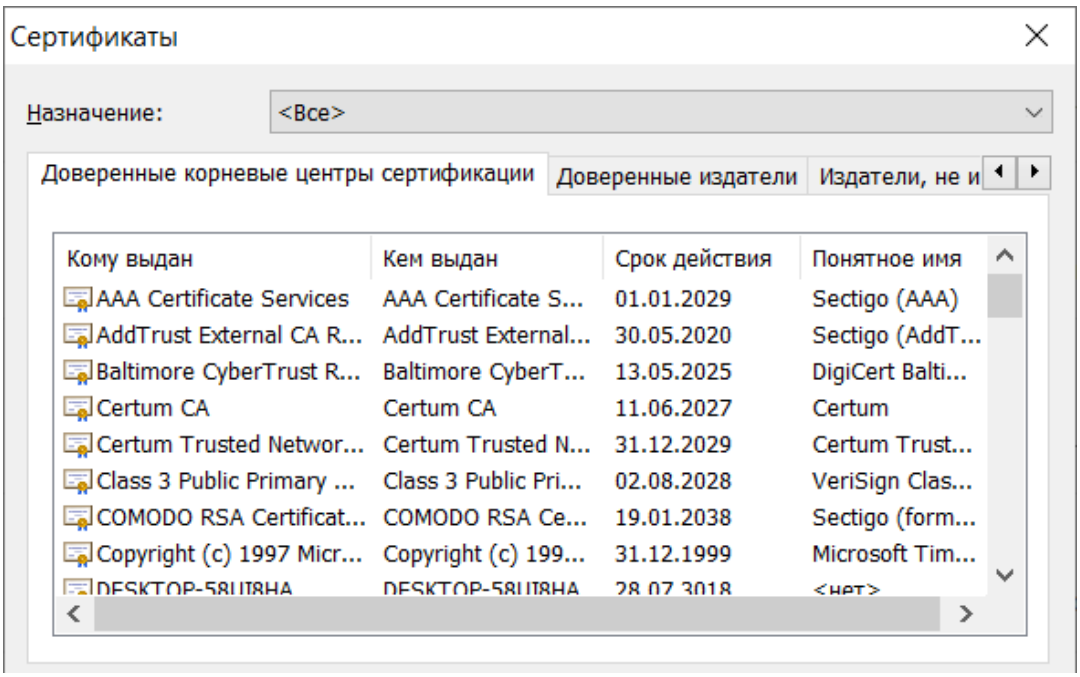

Рис. 3.8. Пример вывода информации о корневых доверенных центрах сертификации

Вопросы для самоконтроля:

1. Какими свойствами обладают собственноручная и цифровая подпись? Что понимается под защищенной цифровой подписью?

2. Как производится формирование и проверка цифровой подписи в криптосистеме RSA? Как производится формирование и проверка цифровой подписи в криптосистеме Эль-Гамаля?

3. В чем заключается принципиальная разница между детерминированными и рандомизированными системами электронной подписи? Приведите примеры таких систем.

4. Какие криптоалгоритмы использует отечественный стандарт цифровой подписи?

§ 3.5. Электронная подпись ГОСТ 34.10–2018

3.5.1. Применение эллиптических кривых в криптографии

Математический аппарат эллиптических кривых исследован еще в XIX веке, однако способы его применения в криптографии были независимо предложены Н. Коблицом (Neal Koblitz) и В. Миллером (Victor Miller) в 1985 году. В криптографии используются эллиптические кривые над конечными полями двух типов:

1. Поля F_p большой простой характеристики p — кривая представима каноническим уравнением вида $y^2 = x^3 + ax + b$, при этом считается, что кривая не должна иметь особых точек, что равносильно требованию неравенства нулю ее дискриминанта: $D = (a / 3)^3 + (b / 2)^2 \neq 0$ или $4a^3 + 27b^2 \neq 0$ (все операции выполняются по модулю p).

2. Поля Галуа $GF(2^m)$, m — большое простое число, кривые вида $y^2 + xy = x^3 + ax + b$ (несуперсингулярные эллиптические кривые) или $y^2 + ay = x^3 + bx + c$ (суперсингулярные эллиптические кривые), при этом допускается равенство дискриминанта нулю.

Российская криптография рассматривает только кривые над конечным полем F_p большой простой характеристики. Такая кривая представима большим, но конечным

множеством точек, удовлетворяющих уравнению кривой, с дополнительным элементом — точкой в бесконечности O. На множестве точек эллиптической кривой E с добавлением точки O вводят *операцию сложения*, по отношению к которой O играет роль нулевого элемента. Операция сложения точек эллиптической кривой подчиняется всем обычным правилам сложения, в частности коммутативному и ассоциативному законам.

Перевод криптосистем с открытым ключом на формализм эллиптических кривых позволяет сократить разрядность используемых величин при эквивалентном уровне криптографической стойкости. Так, в настоящее время минимально допустимый размер модуля криптосистемы RSA составляет 2048 бит (более 600 десятеричных разрядов), в то время как криптосистемы на эллиптических кривых используют минимальные модули порядка 256 бит, то есть в 8 раз меньшей длины. Это безусловно положительно сказывается на производительности вычислений.

Однако не для всех асимметричных схем переход на эллиптические кривые может дать такой выигрыш. Например, вариант алгоритма RSA на эллиптических кривых не имеет преимуществ перед своим оригиналом. *Выигрыш при переходе на эллиптические кривые получают лишь те криптографические схемы, вычислительная стойкость которых базируется на сложности задачи дискретного логарифмирования*. Это обусловлено тем, что при надлежащем выборе параметров эллиптической кривой задача логарифмирования в группе точек существенно сложнее задачи логарифмирования в мультипликативной группе исходного конечного поля.

При переходе на формализм эллиптических кривых умножению двух чисел a и b по модулю p ставится в соответствие сложение двух точек A и B эллиптической кривой E над конечным полем F_p. Операции возведения числа a в степень x соответствует вычисление кратной точки xA, $A \in E$.

Задачей дискретного логарифмирования на эллиптической кривой E, определенной над полем F_p, с порождающей точкой $G \in E$ называется задача нахождения для некоторой заданной точки $P \in E$ такого целого числа $x \in Z$ (если оно существует), что $xG = P$.

Доказано, что сложность задачи дискретного логарифмирования в группе точек эллиптической кривой не ниже сложности дискретного логарифмирования группы точек поля, над которым определена кривая.

Таким образом, задача дискретного логарифмирования на эллиптической кривой является трудной вычислительной задачей. Для кривых над полем большой простой характеристики не известны полиномиальные алгоритмы решения данной проблемы.

Перевод на эллиптические кривые позволяет существенно сократить размер модуля схемы ключевого обмена Диффи — Хеллмана (ECDH), а также криптосистемы Эль-Гамаля и цифровой подписи DSA (ECDSA).

Процедура согласования ключа с использованием Алгоритма Диффи — Хеллмана на эллиптических кривых ECDH специфицирована для ряда защищенных сетевых протоколов в Рекомендациях по стандартизации Р 1323565.1.030–2020 «Информационная технология. Криптографическая защита информации. Использование криптографических алгоритмов в протоколе безопасности транспортного уровня (TLS 1.3)», Р 1323565.1.004–2017 «Информационная технология. Криптографическая защита информации. Схемы выработки общего ключа с аутентификацией на основе открытого ключа» (схема «Лимонник-3»).

В США система ECDH специфицирована стандартом NIST SP 800-56A.

Пусть абоненты A и B хотят сформировать общий секретный ключ. Пусть выбрана эллиптическая кривая E над конечным полем F_p и порождающая точка $G \in E$. Эти параметры используются всеми абонентами. Каждый из абонентов выбирает

случайное целое число, которое держится в секрете. Секретные числа должны иметь порядок, сравнимый с p.

1. А выбирает случайное число k_A и пересылает абоненту В координаты точки $k_A G \in E$.

2. В выбирает случайное число k_B и пересылает абоненту А координаты точки $k_B G \in E$.

Теперь абонент А может вычислить точку $k_A k_B G \in E$, а абонент В — точку $k_B k_A G \in E$. Результат не зависит от порядка вычисления (виду коммутативности группы точек эллиптической кривой E), и абоненты получат общую точку $k_A k_B G = k_B k_A G = (x, y)$. Если полученная точка отлична от точки в бесконечности O, то в качестве общего секрета для вычисления ключа можно использовать одну из ее координат, например, число x.

Выбор эллиптической кривой для обеспечения криптографической стойкости криптосистем является достаточно сложной задачей. После выбора простого большого значения p нужной разрядности и генерации параметров эллиптической кривой E (например, случайным образом) необходимо:

— определить порядок (число точек) N эллиптической кривой;

— удостовериться, что N имеет большой простой делитель n, но не совпадает со значением p или p+1 (например, ГОСТ 34.10–2018 рассматривает значения n порядка 256 и 512 бит);

— найти какую-либо точку эллиптической кривой $P \in E$ и сформировать порождающую точку $G = N/n\ P \neq O$;

— удостовериться, что $p^t \neq 1 \pmod{n}$ для всех $t = 1, 2, \ldots, C$, где C достаточно велико (например, ГОСТ 34.10–2018 устанавливает значение $C = 31$ для значений n порядка 256 бит и $C = 131$ для значений n порядка 512 бит).

Трудоемкость операции выбора эллиптической кривой компенсируется наличием стандартных рекомендаций. Так, NIST SP 800-56A определяет 10 конечных полей (5 конечных полей F_p большой простой характеристики p порядка 192, 224, 256, 384 или 521 битов и 5 полей Галуа $GF(2^m)$ с m равным 163, 233, 283, 409 или 571) и 15 эллиптических кривых (по одной — для каждого поля F_p и по две — для полей Галуа $GF(2^m)$).

Для российских криптоалгоритмов рекомендовано использование двух эллиптических кривых над конечными полями F_p со значением p порядка 256 и 512 битов.

3.5.2. Алгоритм электронной подписи ГОСТ 34.10–2018

Межгосударственный стандарт ГОСТ 34.10–2018 «Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи» описывает алгоритм электронной подписи, реализуемой с использованием операций в группе точек эллиптической кривой, определенной над конечным простым полем F_p. Алгоритм подписи использует варианты хэш-функции, определенные ГОСТ 34.11, и генерирует значения порядка $2^{254} < n < 2^{256}$ либо $2^{508} < n < 2^{512}$. Стандарт не определяет процесс генерации параметров схемы цифровой подписи.

Два варианта параметров эллиптической кривой E над конечным полем F_p, включая координаты генерирующей точки $G \in E$ и ее порядок n, определены в Рекомендациях по стандартизации Р 50.1.114–2016 «Информационная технология. Криптографическая защита информации. Параметры эллиптических кривых для криптографических алгоритмов и протоколов».

После выбора эллиптической кривой каждый абонент системы генерирует личный ключ (ключ подписи) – случайное число d, $0 < d < n$, n — порядок генерирующей точки G (простое число). Затем вычисляется открытый ключ абонента (ключ проверки подписи), представляющий собой точку эллиптической кривой $Q = dG \in E$.

Процедура генерации подписи заключается в следующем:

1. Отправитель выбирает случайное число (разовый ключ) k: $0 < k < n$.

2. Отправитель вычисляет точку $kG = (x, y) \in E$ и число $r = x \bmod n$.

3. Если оказалось, что $r = 0$, то необходимо вернуться к шагу 1, поскольку в этом случае подпись не зависит от секретного ключа d.

4. Отправитель вычисляет хэш-значение значение $h = h(M)$.

Значение h должно быть меньше значения n. Если хэш-функция не удовлетворяет этому условию, вычисляют $h = h(M) \bmod n$.

5. Вычисляется значение $s = (kh + rd) \bmod n$.

6. Если оказалось, что $s = 0$, то необходимо вернуться к шагу 1, так как в этом случае подпись не зависит от самого сообщения.

Пара чисел (r, s) формирует подпись сообщения M и добавляется к нему при передаче. Дополнительно вместе с сообщением может передаваться метка времени или какая-либо другая информация.

Для проверки электронной подписи необходимо знать само сообщение M, значение подписи (r, s) к нему, параметры эллиптической кривой и открытый ключ отправителя сообщения $Q \in E$.

Процедура проверки подписи ГОСТ 34.10 заключается в следующем.

1. Получатель проверяет отличие значений подписи от нуля: $0 < r < n$, $0 < s < n$. Если хотя бы одно из условий нарушается, то делается вывод, что подпись недействительна.

2. Получатель вычисляет хэш-значение для сообщения: $h = h(M) \bmod n$.

3. Получатель вычисляет значения:

$v = h^{-1} \bmod n$ (с помощью расширенного алгоритма Евклида, значение v существует и единственно, так как n — простое число, а $h < n$, значит h и n — взаимно простые числа);

$u_1 = sv \bmod n$,

$u_2 = -rv \bmod n$

4. Получатель вычисляет точку $X = (x, y) \in E$: $X = u_1G + u_2Q$.

5. Получатель проверяет выполнение равенства $r = x \bmod n$. Если равенство выполнено, то делается заключение о действительности подписи подлинная, иначе подпись недействительна.

Отметим, что подпись ГОСТ 34.10 как вариант криптосистемы Эль-Гамаля является рандомизированной. Несмотря на то что алгоритм формирования и проверки подписи использует операции с точками эллиптической кривой, сама подпись представляет собой пару чисел.

Итак, в третьей главе были рассмотрены криптографические шифры с открытым ключом, проанализированы общие сведения об асимметричных криптосистемах, система распределения ключей Диффи — Хеллмана (DH), системы шифрования с открытым ключом, технология электронной подписи, электронная подпись по ГОСТ 34.10–2018. Далее рассмотрим криптографические протоколы.

Вопросы для самоконтроля:

1. Какой тип криптографической системы определяет стандарт ГОСТ 34.10?

2. Что понимается под эллиптической кривой? Эллиптические кривые какого вида используются в криптографии? Какое преимущество дает перевод криптосистем на формализм эллиптических кривых?

3. Какие требования предъявляются к эллиптическим кривым для использования в алгоритме ГОСТ 34.10?

4. Каков алгоритм формирования цифровой подписи по стандарту ГОСТ 34.10?

Глава 4
КРИПТОГРАФИЧЕСКИЕ ПРОТОКОЛЫ

§ 4.1. Определение и типы криптографических протоколов

С широким распространением компьютерных технологий проблематика криптографии пополнилась еще одним разделом — криптографическими протоколами. *Криптографический протокол* — это набор правил и соглашений, регламентирующих процедуру решения некоторой задачи криптографическими методами.

Протокол определяет последовательность шагов, которые предпринимают две или большее количество сторон для совместного решения задачи[14]. Все шаги следуют в порядке строгой очередности, и ни один из них не может быть сделан прежде, чем закончится предыдущий. Кроме того, любой протокол подразумевает участие, по крайней мере, двух сторон. Протоколы имеют и другие отличительные черты:

— каждый участник протокола должен быть заранее оповещен о шагах, которые ему предстоит предпринять;

— все участники протокола должны следовать его правилам добровольно, без принуждения;

— необходимо, чтобы протокол допускал только однозначное толкование, а его шаги были совершенно четко определены и не допускали возможности их неправильного понимания;

— протокол должен содержать описание реакции его участников на любые ситуации, возникающие в ходе реализации этого протокола — иными словами, недопустимым является положение, когда для возникшей ситуации протоколом не определено соответствующее действие.

Описание протокола, исключающее неоднозначность восприятия, называют *спецификацией протокола*.

Основные отличия криптографических протоколов от криптосистем:

— протоколы могут быть интерактивными, то есть подразумевать многораундовый обмен сообщениями между участниками;

— в протоколе может быть задействовано более двух участников;

— участники протокола могут не доверять друг другу. Поэтому криптографические протоколы должны защищать их участников не только от внешнего противника, но и от нечестных действий партнеров.

В зависимости от участия третьей стороны, которой безразличен результат исполнения протокола и его участники, протоколы делят на:

1. *Протоколы с арбитражем*. Арбитр является посредником, участвующим в протоколе и помогающим его исполнить другим сторонам. Стороны могут не доверять друг другу, но они полностью доверяют арбитру.

2. *Протоколы с судейством*. Посредник вступает в протокол только в исключительных случаях – когда между сторонами возникают разногласия, то есть играет роль судьи.

3. *Самодостаточные* (*самоутверждающиеся*) *протоколы*. Честность сторон гарантируется самим протоколом. Самоутверждающийся протокол устроен так, что если один из его участников мошенничает, другие смогут моментально распознать не-

[14] Васильева И. Н. Криптографические протоколы: учебное пособие. СПб.: Изд-во СПбГЭУ, 2021. 119 с.

честность, проявленную этим участником, и прекратить выполнение дальнейших шагов протокола. К сожалению, универсальных самоутверждающихся протоколов не существует. На практике в каждом конкретном случае приходится конструировать свой специфический самоутверждающийся протокол.

Участники протокола могут доверять или не доверять друг другу, поэтому криптографические протоколы должны защищать их участников не только от внешнего противника, но и от нечестных действий партнеров. Криптоаналитические атаки могут быть направлены не на алгоритм шифрования, а на протокол. Поэтому даже наличие абсолютно надежного алгоритма шифрования не гарантирует полной безопасности абонентам системы связи. Известны случаи, когда применяемые на практике криптографические протоколы содержали изъяны, допускающие мошенничество участвующих сторон или реализацию активных атак. Поэтому криптографические протоколы являются предметом тщательного анализа со стороны специалистов.

Криптографические протоколы — сравнительно молодая (первые протоколы появились во второй половине XX века) и бурно развивающаяся область криптографии. В настоящее время имеется несколько десятков различных типов криптографических протоколов. Все криптографические протоколы можно условно разделить на две группы: прикладные протоколы и примитивные протоколы. *Прикладные протоколы* решают конкретные практические задачи. *Примитивные протоколы* — это элементы, из некоторого подмножества которых формируются прикладные протоколы. Перечислим основные практически важные протоколы.

Протоколы конфиденциальной передачи сообщений. Пусть имеются два участника протокола, являющиеся легальными абонентами сети связи. Участники соединены линией связи, по которой можно пересылать сообщения в обе стороны, и которую может контролировать противник. У одного из абонентов имеется конфиденциальное сообщение M, и задача состоит в том, чтобы это сообщение конфиденциальным же образом передать второму абоненту. Протоколы этого типа появились раньше других криптографических протоколов, так как задача конфиденциальной передачи сообщений — исторически первая задача, которая решалась криптографией.

Протоколы аутентификации и идентификации предназначены для предотвращения несанкционированного доступа нелегальным пользователям к информации ограниченного доступа.

Протоколы распределения ключей необходимы для обеспечения секретными ключами легальных участников обмена криптограммами.

Протоколы электронной подписи позволяют ставить под электронными документами подпись, аналогичную обыкновенной подписи на бумажных документах. В результате выполнения протокола электронной подписи к передаваемой информации добавляется уникальное числовое дополнение, позволяющее проверить ее авторство.

Протоколы обеспечения неотслеживаемости («электронные деньги»). Под электронными деньгами в криптографии понимают электронные платежные средства, обеспечивающие неотслеживаемость, то есть невозможность проследить источник пересылки информации.

Теоретические протоколы, решающие отдельные задачи, объединяются в защищенные протоколы реального мира, используемые на практике в компьютерных и технических системах (примерами являются протоколы SSL/TLS, IPSec, Kerberos и др.). Эти протоколы, как правило, имеют достаточно сложную структуру, поскольку позволяют решать одновременно несколько задач, например, выполнять аутентификацию сторон сетевого взаимодействия, формировать общие криптографические ключи, выполнять

защищенную передачу данных с обеспечением конфиденциальности и контроли целостности передаваемых сообщений. Кроме того, один протокол может предусматривать разные условия применения, например, наличие общего предраспределенного секрета, либо наличие цифрового сертификата, аутентификацию одной, обеих сторон, либо отсутствие аутентификации.

Вопросы для самоконтроля:

1. Что понимается под криптографическим протоколом?
2. Каковы основные отличительные черты криптографических протоколов?
3. Какие особенные роли могут выполнять участники протокола?
4. Каковы особенности самоутверждающегося протокола?
5. Каковы основные типы криптографических протоколов?

§ 4.2. Протоколы, связанные с электронной подписью

4.2.1. Разрешение споров по электронной подписи (протокол с судейством)

Для практического применения схем электронной подписи, помимо алгоритмов формирования подписи и ее верификации, требуется процедура судейства. Судейство необходимо, когда один из абонентов, например, В, предъявляет сообщение и подпись, утверждая, что эта пара сообщение-подпись была получена от А, в то время как абонент А отказывается признавать эту подпись своей.

При разборе дела в суде судья выступает в качестве эксперта, дающего заключение о подлинности электронной подписи. Протокол судейства заключается в следующем:

Абонент В предъявляет судье электронный документ и подпись.

Судья требует от абонента А предъявления своего секретного ключа. Если А отказывается, судья дает заключение, что подпись ложная.

Судья выбирает из сертифицированного справочника открытый ключ абонента А и проверяет его соответствие секретному ключу, предъявленному А. Если они совпадают, судья переходит к шагу 5.

1. При обнаружении факта несоответствия ключей судья обращается в центр сертификации и требует предоставления заверенного абонентом А документа, содержащего его открытый ключ. Если выясняется, что открытый ключ, взятый из справочника, не совпадает с указанным в документе, судья признает подпись, предъявленную В, подлинной, при этом все издержки такого решения компенсируются за счет центра сертификации. Если открытые ключи в справочнике и документе совпадают, то есть абонент А предъявил некорректный секретный ключ, судья признает электронную подпись ложной.

Судья проверяет соответствие друг другу подписи и документа. При положительном результате проверки подпись признается подлинной, в противном случае отвергается.

Задача судейства значительно сложнее, чем, кажется на первый взгляд. Решение споров в суде невозможно в следующих случаях:

— секретный ключ сформирован не самим абонентом А, а специальным центром генерации ключей;

— аппаратура, на которой выполняется алгоритм генерации или проверки подписи, содержит какие-либо элементы, не контролируемые пользователем («черные ящики», защищенные участки памяти и т. п.).

Наконец, возможны безвыходные ситуации, в которых судья не может принять никакого обоснованного решения. Например, абонент В предъявляет s и утверждает, что это подпись под документом M. Абонент А признает, что это его подпись, но под

документом $M' \neq M$, при этом выясняется, что хэш-значения этих документов совпадают, то есть $h(M') = h(M)$. В этом случае невозможно доказать, кто из абонентов нашел коллизию, а значит, подделал подпись. Выход из положения возможен только в том случае, если заранее обговорен порядок разрешения спора в такой ситуации.

4.2.2. Особые схемы электронной подписи

В некоторых ситуациях могут потребоваться схемы электронной подписи, отличные от классических схем. Известны следующие специальные схемы электронной подписи:

— схема подписи «вслепую», когда абонент А подписывает документ, не зная его содержимого;

— схема групповой подписи, которая позволяет получателю убедиться в принадлежности полученного сообщения некоторой группе абонентов, но кто именно из членов группы подписал документ, получатель определить не в состоянии;

— схема разделяемой подписи, которая формируется только при участии определенного количества участников протокола, иначе говоря, данная схема является объединением классической схемы подписи и схемы разделения секрета;

— схема конфиденциальной (неотвергаемой) подписи, которая не может быть проверена без участия сформировавшего ее участника протокола;

— схема неоспоримой подписи, в которой подделка подписи может быть доказана.

Вопросы для самоконтроля:

1. Каковы основные шаги протокола разрешения споров о подписи документа?

2. В каких случаях невозможно вынесение судейского решения об электронной подписи?

3. Каковы особые, отличные от классических, схемы электронной подписи?

§ 4.3. Протоколы аутентификации и распределения ключей

4.3.1. Виды протоколов распределения секретных ключей

При использовании симметричной криптографии для защиты передаваемых сообщений основной проблемой является необходимость получения общего ключа обеими сторонами до начала шифрования. Этой проблемы лишена криптография с открытым ключом (при использовании цифровой сертификации), однако ее производительность существенно ниже. Поэтому встает вопрос разработки протоколов защищенного распределения симметричных ключей между абонентами системы.

Протокол распределения ключей (key establishment protocol) — это криптографический протокол, в процессе выполнения которого общий секрет становится доступен двум или более сторонам для последующего использования в криптографических целях.

Протоколы распределения ключей подразделяются на два класса.

Протоколы транспортировки ключей (key transport) — протоколы распределения ключей, в которых один участник создает или другим образом приобретает секрет и безопасным образом передает его другим участникам.

Протоколы согласования ключей (key agreement, key exchange) — протоколы распределения ключей, в которых общий секрет вырабатывается двумя или более сторонами как функция от информации, вносимой каждой из них таким образом, что никакая другая сторона не может предопределить их общий секрет.

При этом не все абоненты системы могут быть известны заранее, кроме того, абоненты, вступающие в связь, могут не иметь предварительно распределенных общих секретов. В соответствии с принципом иерархии ключей, даже если абоненты и обладают общим предварительно распределенным секретом (долговременным ключом), для шифрования передаваемых сообщений должны использоваться, и соответственно до этого каким-то образом транспортироваться или согласовываться, другие — кратковременные ключи, различные для каждого сеанса связи (сеансовые ключи).

Распределение ключей может производиться (рис. 4.1):

— прямым обменом сеансовыми ключами между пользователями сети (децентрализованное распределение ключей);

— централизованно — с участием третьей доверенной стороны, в роли которой может выступать один или несколько доверенных центров распределения ключей (key distribution center, KDC) либо центров транспортировки ключей (key translation center, KTC).

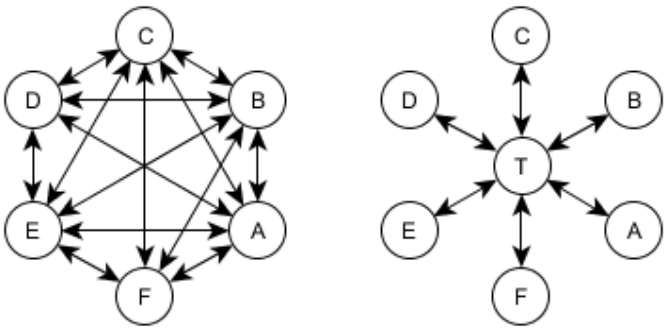

Рис. 4.1. Варианты распределения ключей без выделенного доверенного центра и с выделенным доверенным центром T

Децентрализованное распределение ключей симметричного шифрования требует наличия у каждого пользователя большого количества предраспределенных ключей (для связи с каждым из абонентов системы), которые необходимо сначала безопасно распределить, а потом обеспечивать их секретность в процессе хранения. Децентрализованное распределение симметричных ключей возможно при использовании гибридных схем, когда для аутентификации сторон и защиты передаваемого ключа используется криптография с открытым ключом, а для подтверждения связи самих асимметричных ключей с абонентами используется цифровая сертификация. Здесь надежность долговременных асимметричных ключей обеспечивается третьей доверенной стороной, выпустившей сертификат, — центром сертификации (certification authority, CA), удостоверяющим центром.

Если же асимметричные ключи по какой-то причине недоступны в системе, централизованное распределение ключей с участием третьей доверенной стороны может оказаться более рациональным.

Централизованное распределение ключей симметричного шифрования подразумевает, что у каждого пользователя есть только один основной ключ для взаимодействия с центром распределения ключей. Одной из самых известных систем централизованного распределения ключей является протокол Kerberos.

Когда в системе работает KDC, любые два участника A и B имеют общие долговременные ключи для общения с KDC, но не имеют ключей для связи друг с другом. Сеансовые ключи генерируются KDC централизованно, после чего они рассылаются обеим сторонам.

Когда в системе имеется КТС, участники A и B имеют общие с КТС ключи, но ключи для связи между собой генерируются ими самостоятельно, а КТС выполняет лишь функцию пересылки (трансляции) ключей от одного участника к другому.

Централизованное распределение ключевой информации позволяет сократить объем хранимого ключевого материала и более защищено в случае компрометации отдельных абонентов. Под компрометацией абонента понимается ситуация, когда вся информация об этом абоненте (включая его секретные ключи) становится известной противнику. Однако при компрометации сервера ключей оказывается скомпрометированной вся сеть. Кроме того, центр ключей является потенциальным узким местом системы, если он не сможет оперативно снабдить абонентов сеансовыми ключами (например, в силу нехватки вычислительных мощностей или отказов оборудования), абоненты не смогут обмениваться зашифрованными сообщениями.

Протоколы распределения ключей могут быть классифицированы и по методам защиты передаваемого или формируемого ключа — с использованием *симметричных криптосистем* или с использованием *криптосистем с открытым ключом*. Для ситуаций, когда пользователи доверяют друг другу, и ситуаций, когда пользователи не доверяют друг другу, используются разные протоколы. В случае доверия между пользователями могут использоваться средства как симметричной, так и асимметричной криптографии, а в случае недоверия — протоколы только на основе криптосистем с открытым ключом.

4.3.2. Распределение ключей с помощью симметричных криптосистем и арбитра

Распределение ключевой информации играет центральную роль в применении симметричных криптосистем. Решение этой проблемы средствами симметричной криптографии требует участия третьей доверенной стороны — арбитра.

Ключи для связи между центром и его абонентами обычно считаются *предраспределенными*, то есть распределяются, как правило, с помощью ручного процесса (при личном обращении, с использованием курьерской доставки, доставки официальной почтовой службой или с помощью иной доверенной службы распространения) и с использованием методов физической защиты, при котором либо центр, либо абонент генерирует ключевой материал и предоставляет его другой стороне.

Предположим, что абоненты A и B получили от арбитра D секретные ключи K_A и K_B. Считается, что эти ключи были переданы надежным способом до начала сеанса связи. Ключи K_A и K_B не используются абонентами для общения друг с другом, а только для взаимодействия с арбитром D. Тогда для обмена зашифрованными сообщениями абоненты A и B должны использовать сеансовый ключ, для получения которого они могут обратиться к арбитру D:

1. Абонент A обращается к арбитру D и запрашивает у него сеансовый ключ для связи с абонентом B.
2. Арбитр случайным образом генерирует сеансовый ключ и создает две копии этого ключа, одну из которых он шифрует с помощью секретного ключа K_A абонента A, а вторую — с помощью секретного ключа K_B абонента B. Затем арбитр отсылает обе копии обратившемуся к нему абоненту A.
3. A расшифровывает свою копию сеансового ключа.
4. A отправляет абоненту B его зашифрованную копию сеансового ключа.
5. B расшифровывает свою копию сеансового ключа.
6. Теперь абоненты могут обмениваться сообщениями, зашифрованными с помощью сеансового ключа, копия которого имеется у них обоих.

Данный протокол нашел практическое применение в протоколе Kerberos, который наряду с задачей распределения сеансовых ключей решает задачу аутентификации сторон и реализует концепцию «единого входа» в локальной сети. Протокол Kerberos используется для аутентификации сетевых пользователей в домене Windows. Кроме того, Kerberos поддерживается и большинством Linux/Unix систем. Протокол Kerberos использует дополнительно метки времени и ограничение времени действия выданных аутентификаторов. В настоящее время протокол Kerberos поддерживает расширение PKINIT, позволяющее использовать цифровые сертификаты асимметричных криптосистем на этапе аутентификации пользователей.

В рассмотренном протоколе абоненты, которые хотят обмениваться зашифрованными сообщениями, полностью полагаются на честность арбитра. При компрометации арбитра будет скомпрометирована вся сеть шифрованной связи.

В большинстве прикладных задач для распределения ключей отдается предпочтение использованию криптосистем с открытым ключом.

4.3.4. Распределение ключей с помощью асимметричных криптосистем

Для распределения ключей может быть использована рассмотренная ранее система Диффи — Хеллмана. Однако схема Диффи — Хеллмана в чистом виде уязвима для атаки «человек посередине», поэтому требует предварительной аутентификации сторон. Другой подход к распределению ключей состоит в использовании асимметричного шифра для защищенной транспортировки сеансового ключа, который будет затем использован абонентами для шифрования своих данных. В этом случае протокол обмена сеансовыми ключами будет следующим:

1. Абонент A находит открытый ключ абонента B в справочнике, в базе данных или получает его от арбитра D.

2. A генерирует случайный сеансовый ключ k, шифрует его при помощи открытого ключа абонента B и отправляет ему в зашифрованном виде.

3. B расшифровывает полученное от абонента A сообщение с использованием своего личного ключа. Теперь он имеет в своем распоряжении сеансовый ключ k.

4. Абоненты могут обмениваться сообщениями, зашифрованными на одном и том же сеансовом ключе k.

Однако и этот протокол без дополнительной аутентификации уязвим для активных атак типа «человек посередине». Таким образом, при кажущейся простоте, основной проблемой протоколов ключевого обмена с использованием асимметричных криптосистем является проблема доверия к источнику открытых ключей. Отправитель сеансового ключа должен быть уверен, что используемый им для шифрования открытый ключ получателя не подделан противником, в противном случае он фактически передаст свои секреты в его руки.

Использование цифровых подписей абонентов позволяет провести аутентификацию отправителя (то есть проверить его подлинность) и противодействовать активным атакам при обмене ключами, однако в этом случае требуется наличие доверенной стороны — арбитра.

В этом случае на втором шаге описанного выше протокола сеансовый ключ до шифрования подписывается личным ключом отправителя (абонента A). На следующем шаге после расшифрования абонент B должен проверить подлинность подписи отправителя, предварительно получив его открытый ключ от арбитра D.

Арбитр позволяет решить проблему аутентификации открытых ключей абонентов. Теперь абоненты A и B получают открытые ключи, подписанные арбитром D. Каждый ключ снабжается свидетельством о его принадлежности определенному лицу. Получая

открытый ключ, A и B могут проверить подпись арбитра и убедиться, что этот ключ принадлежит именно тому абоненту, которому они собираются послать сообщение. Открытый ключ арбитра D для проверки его подписи должен быть передан абонентам надежным способом заранее.

Противник E не сможет выдать себя за одну из сторон взаимодействия A или B, поскольку не знает личных ключей этих абонентов, соответствующих открытым ключам, переданным арбитром. E также не сможет подменить открытые ключи A и B на свои собственные, поскольку не сможет выдать себя за арбитра.

Если же противнику E удастся выдать себя за арбитра, возможность проведения активных атак сохранится. Таким образом, активные атаки на криптосистемы с открытым ключом становятся невозможными, если абоненты не передают открытые ключи по каналу связи, а получают их из надежного источника (от арбитра), в подлинности которого они могут удостовериться. Решение этой задачи предполагает наличие организационной структуры, занимающейся сертификацией открытых ключей (инфраструктуры открытых ключей PKI), использование цифровых сертификатов и их удостоверение центрами (службами) сертификации (CA).

4.4.4. Взаимная аутентификация с распределением ключей с помощью асимметричных криптосистем

Пусть два абонента A и B хотят провести взаимную аутентификацию и сформировать сеансовый ключ для последующего обмена зашифрованными сообщениями. Предполагается, что абоненты могут получить действующие открытые ключи надежным способом (например, от службы сертификации). При этом требуется обеспечить явную аутентификацию сторон, чтобы по окончании протокола стороны точно знали, с кем они взаимодействуют.

Криптографический протокол Нидхема — Шредера (Roger Needham, Michael Schroeder) позволяет производить явную аутентификацию сторон одновременно с формированием общего сеансового ключа:

1. Абонент A генерирует случайное значение k_A, объединяет его со своей открытой информацией, затем шифрует открытым ключом абонента B и отсылает B: A: $P_{OK(B)}(k_A$, абонент A$) \rightarrow$ B.

Абонент B расшифровывает своим личным ключом присланное ему сообщение и убеждается, что присланное сообщение содержит открытую информацию абонента A. Затем B выбирает случайное значение k_B, объединяет его с присланным значением k_A, шифрует открытым ключом абонента A и отсылает A: B: $P_{OK(A)}(k_A, k_B) \rightarrow$ A.

2. Абонент A расшифровывает своим личным ключом полученное от B сообщение и убеждается в том, что оно содержит верное значение k_A. Это является гарантией аутентификации B, так как никто, кроме абонента B, не смог бы извлечь это значение из зашифрованного сообщения. Удостоверившись в подлинности абонента B, A шифрует значение k_B открытым ключом B и пересылает его B: A: $P_{OK(B)}(k_B) \rightarrow$ B.

3. Абонент B расшифровывает своим личным ключом сообщение и извлекает k_B. Верное значение k_B является гарантией аутентификации абонента A, так как на предыдущем шаге никто, кроме него, не смог бы извлечь k_B из зашифрованного сообщения.

Теперь стороны A и B убедились в подлинности друг друга и имеют в своем распоряжении случайные значения k_A и k_B, известные только им. На основе этих значений абоненты A и B могут сформировать общий сеансовый ключ, например, $k = k_A \oplus k_B$ или $k = k_A \parallel k_B$, либо использовать ключи k_A и k_B по отдельности для шифрования входящих и исходящих сообщений. Поскольку числа k_A и k_B каждый раз генерируются заново, для нового сеанса будет получен новый ключ.

Важно, что на первом шаге абонент А добавляет к случайному числу k_A свою личную информацию. Если бы протокол не предусматривал этого, то абонент А мог бы использовать его для вскрытия криптосистемы абонента В. Пусть, например, А перехватил некоторое предназначенное для В и зашифрованное его открытым ключом сообщение y. А не может сам расшифровать сообщение y, однако он может притвориться, что хочет пройти процедуру аутентификации и на шаге 1 протокола послать y абоненту В в качестве зашифрованного значения k_A. В ответ на шаге 2 абонент В расшифрует y пришлет его А.

Возможна атака на рассмотренный протокол, предложенная Г. Лоу (Gavin Lowe). Пусть противник является одним из законных абонентов системы, обозначим его С. Атака становится возможной, если абонент А вступил в протокол с С, и заключается в проведении абонентом С параллельного протокола с абонентом В.

Противостоять атакам на протокол Нидхема — Шредера можно, выполняя дополнительное подписание абонентом (то есть заверение своим личным ключом) информации перед ее отправлением получателю:

1. А: $P_{ЛК(A)}(P_{ОК(B)}(k_A)$, абонент А) $\rightarrow$ В.
2. В: $P_{ЛК(B)}(P_{ОК(A)}(k_A, k_B)) \rightarrow$ А.
3. А: $P_{ЛК(A)}(P_{ОК(B)}(k_B)) \rightarrow$ В.

В этом случае атака Лоу не может быть реализована, так как атакующий С даже не сможет вступить в связь от имени другого абонента. Таким образом, пришли к необходимости использования не только шифрования, но и цифровой подписи пересылаемой информации.

Спецификация X.509 вводит трехступенчатый протокол, использующий цифровые подписи абонентов. Этот протокол может быть использован как однопроходный, двухпроходный и трехпроходный в зависимости от числа выполняемых шагов. Протоколы предполагают использование цифровых сертификатов, полученных от доверенного центра сертификации.

В протоколах используются следующие величины: t_A, t_B — метки времени, r_A, r_B — случайные числа, опционально: X_A, X_B — произвольные данные, для которых требуется обеспечить аутентичность и целостность, Y_A, Y_B — произвольные данные, для которых, кроме того, требуется обеспечить еще и секретность (такими данными могут быть, например, сеансовые ключи или данные для формирования общего секрета). Через P_X обозначено шифрование на открытом ключе абонента X, а через S_X — подписанное абонентом X сообщение.

Однопроходный протокол обеспечивает одностороннюю аутентификацию абонента А:

1. А $\rightarrow$ В: $A, S_A(t_A, r_A, B, X_A, P_B(Y_A))$.

Добавление к нему еще одной пересылки сообщения дает двухпроходный протокол, обеспечивающий взаимную аутентификацию сторон:

2. В $\rightarrow$ А: $B, S_B(t_B, r_B, A, X_B, P_A(Y_B))$.

Добавлением еще одной пересылки получается трехпроходный протокол, подтверждающий получение данных стороной А:

3. А $\rightarrow$ В: $A, S_A(B, r_B)$.

В трехпроходном протоколе метки времени не передаются, так как их использованием является избыточным.

Использование меток времени противодействует атакам с повторным использованием сообщений или с блокированием канала и отправлением искаженных сообщений. Принятию искаженных сообщений противодействует также наличие цифровой подписи.

Вопросы для самоконтроля:

1. Каков протокол распределения ключей на симметричных криптосистемах? Какой специфической роли он требует?

2. Каким образом могут быть распределены ключи с помощью асимметричного шифра? Каковы возможные атаки на этот протокол?

3. Каковы возможные атаки на протокол Нидхема — Шредера? Каковы способы противодействия этим атакам?

4. Какие протоколы определяет спецификация X.509? Какие задачи они решают?

§ 4.4. Специальные криптографические протоколы

4.4.1. Протоколы разделения секрета (распределения ответственности)

В некоторых ситуациях требуется обеспечить совместный доступ к секретному сообщению нескольких участников протокола, при этом никто из них не мог бы получить его по отдельности. Такая ситуация может возникнуть при отсутствии взаимного доверия между участниками[15].

Тогда секретное сообщение (например, пароль или секретный ключ) требуется разделить на части, каждая из которых не дает никакого представления об исходном сообщении, однако соединив их все вместе, можно получить исходный секрет. Таким образом, ответственность за хранение и восстановление секрета распределяется между участниками протокола. В криптографии под *разделением секрета* (secret sharing) понимают любой метод распределения секрета среди группы участников, каждому из которых достается доля секрета (shadow). Воссоздать секрет может только коалиция участников.

Простейший криптографический протокол позволяет арбитру D поровну распределить секрет M между двумя членами A и B. Этот протокол легко обобщается на произвольное число участников:

Арбитр D генерирует случайную битовую строку R той же длины, что и исходное сообщение M.

1. D вычисляет $S = M \oplus R$.

2. D вручает R участнику A, а S — участнику B.

3. Восстановление исходного сообщения M требует совместного участия A и B: $M = R \oplus S$.

Знание одной из частей (только S или только R) не позволяет реконструировать M.

Фактически арбитр шифрует сообщение M при помощи одноразового блокнота и отдает полученную криптограмму одному участнику, а сам блокнот — другому. Основной недостаток такого подхода к разделению секрета состоит в том, что если хотя бы один из его участников потеряет доверенную ему часть, будет также безвозвратно утрачено и само сообщение.

Может возникнуть необходимость в восстановлении секретного сообщения даже в отсутствие некоторых участников протокола. Для решения этой задачи используется *пороговый протокол*. Секретное сообщение делится на n долей так, что для его восстановления обязательно требуется хотя бы m из них. Такой протокол называется (m, n)-*пороговым протоколом разделения секрета*.

Пороговый протокол разделения секрета называется *совершенным*, если не менее m участников могут восстановить секрет, а любая группа из m — 1 или меньшего

[15] Шнайер Б. Прикладная криптография: Протоколы, алгоритмы, исходные тексты на языке Си. М.: Триумф, 2012. 815 с.

числа участников — не сможет этого сделать. Пороговый протокол разделения секрета называется *идеальным*, если доли любой группы участников численностью меньшей m, не дают информации о секрете. Показано, что в любом идеальном пороговом протоколе разделения секрета размер каждой из долей должен быть не меньше размера самого секрета.

Существуют несколько пороговых схем разделения секрета, основанных на разных математических задачах:

— схема Шамира (восстановление (m — 1)-мерного многочлена по m точкам);

— схема Блэкли (восстановление координат точки пересечения m-мерных гиперплоскостей);

— схемы, основанные на китайской теореме об остатках;

— схемы, основанные на решении систем уравнений или сравнений с m неизвестными.

В некоторых из них, например, в протоколе Шамира, полномочия участника по восстановлению секрета могут быть увеличены за счет выдачи ему дополнительных долей. Потребуем, например, чтобы секрет мог быть восстановлен не менее чем 4 рядовыми сотрудниками, либо начальником отдела в присутствии хотя бы одного рядового сотрудника. Тогда можно использовать пороговый протокол с $m = 4$, выдав рядовым сотрудникам по 1 доли, а начальникам отделов — по 3 доли.

Большинство протоколов разделения секрета предполагает наличие арбитра, распределяющего доли, поэтому зависит от его честности и надежности. Кроме того, не во всех протоколах владельцы долей могут удостовериться, что получили от арбитра правильную долю. Существуют протоколы *подтверждаемого совместного владения секретом*, в которых участники могут убедиться, что их доля правильна без необходимости восстанавливать сам секрет. Кроме того, могут быть сконструированы протоколы разделения секрета, в которых никто из участников не знает секрета до момента его восстановления, то есть, нет арбитра, владеющего секретом заранее и распределяющего его между другими участниками.

Другой особенностью рассмотренных протоколов является необходимость открытого оглашения своих долей участвующими в восстановлении секрета сторонами. Это позволяет противнику, притворившись участником протокола, дождаться оглашения нужного числа долей и самостоятельно восстановить секрет. Для предотвращения такого рода мошенничества используются протоколы разделения секрета без раскрытия долей (например, с использованием групповых подписей). Вместе с тем, это не защищает от саботажа, когда владелец одной из долей может помешать восстановлению общего секрета, отдав неверную долю.

Другой проблемой является возможный сговор m участников (например, владеющих долями уволенных сотрудников) для несанкционированного восстановления секрета. Для предотвращения таких ситуаций могут быть использованы более сложные протоколы с двумя порогами: m и k, которые позволяют восстановить секрет только в том случае, если не менее m участников высказались за его реконструкцию и при этом не более k участников были против.

4.4.2. Протокол доказательства с нулевым разглашением конфиденциальной информации

Протокол доказательства с нулевым разглашением (нулевым знанием) позволяет стороне А подтвердить знание некоторой конфиденциальной информации без передачи этой информации или каких-либо сведений о ней проверяющей стороне В.

Доказательство с нулевым разглашением должно обладать тремя свойствами:

— *полнота*: если утверждение действительно верно (А знает секрет), то доказывающий убедит в этом проверяющего;

— *корректность*: если утверждение неверно, то даже нечестный доказывающий не сможет убедить проверяющего за исключением пренебрежимо малой вероятности;

— *нулевое разглашение*: если утверждение верно, то любой даже нечестный проверяющий не получит никакой другой информации, кроме самого факта, что утверждение верно.

Доказательство обычно имеет форму *интерактивного протокола*. Знание секрета определяет множество вопросов, для которых владелец секрета знает правильные ответы. Проверяющий В задает А серию вопросов, предполагающих ответ в виде «да»/«нет». Если А знает секрет, то он ответит правильно на все заданные ему вопросы. Если же А не знает секрет, то вероятность угадывания правильного ответа на каждый из вопросов будет около 1/2. Тогда после n вопросов вероятность правильного угадывания всех ответов составит 2^{-n} и проверяющий В сможет понять, обманывает ли его А. При этом проверяющая сторона В не сможет получить какую-либо полезную информацию о сути самого секрета.

Использование доказательства с нулевым разглашением конфиденциальной информации можно пояснить на простом примере (рис. 4.2). Предположим, что имеется пещера. Вход в пещеру находится в точке (0), а в точке (1) пещера разветвляется на две половины — левую L и правую R. Между левой и правой половинами пещеры есть секретная дверь, однако воспользоваться ей сможет только тот, кто знает пароль.

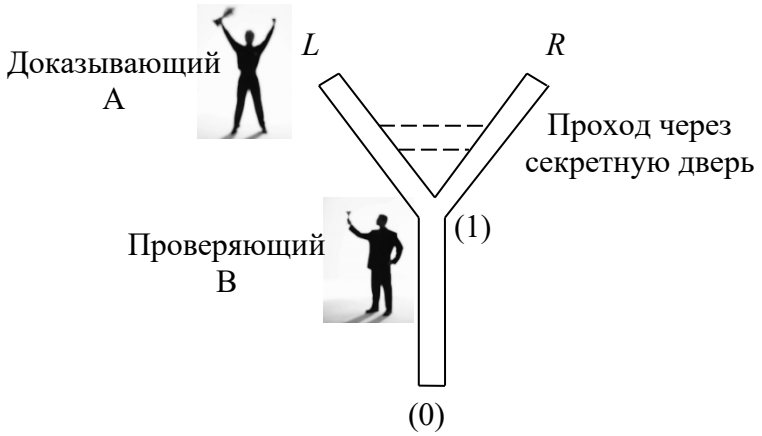

Рис. 4.2. Пример протокола с нулевым разглашением «Пещера с секретной дверью»

А секретный пароль известен, а В — нет. А хочет доказать В, что знает пароль, но так, чтобы проверяющий В его не узнал. Тогда А может воспользоваться следующим протоколом:

1. Проверяющий В стоит у входа в пещеру в точке (0).

2. По своему усмотрению А заходит в пещеру либо слева (через вход L), либо справа (через вход R) и подходит к секретной двери.

3. Проверяющий В перемещается в точку (1).

4. Проверяющий В приказывает доказывающему А появиться либо через левый проход, либо через правый.

5. А подчиняется приказу В, в случае необходимости используя пароль, чтобы пройти через секретную дверь.

6. Шаги 1–5 (раунд протокола) повторяются n раз, где n — параметр протокола.

Не зная пароля, А сможет выходить только с той стороны, с которой зашел в пещеру. Следовательно, он сумеет обмануть проверяющего В, догадавшись, с какой именно стороны тот попросит его выйти, лишь примерно в половине случаев. Если количество экспериментов равно n, то вероятность случайного угадывания во всех испытаниях составит 2^{-n}. Таким образом, если А успешно прошел все n испытаний, вероятность обмана, то есть того, что А при этом не знает пароль, составит 2^{-n}.

На практике при доказательстве с нулевым разглашением конфиденциальной информации используется сведение одной вычислительно трудной задачи (знание решения которой доказывает проходящая проверку сторона А) к другой задаче, изоморфной первой. Примером таких задач может служить *задача о раскраске графа* (вершины графа раскрашиваются тремя различными красками таким образом, чтобы любые две вершины, соединенные одним ребром, были окрашены в разные цвета), которая относится к классу NP-полных задач. Другой задачей является *отыскание в связном графе Гамильтонова цикла* для больших изоморфных графов.

Гамильтоновым циклом называется замкнутый путь, проходящий через все вершины графа только один раз (за исключением первой и последней вершин, которые совпадают).

Доказано, что задача нахождения гамильтонова цикла в графе является NP-полной, а значит, относится к классу вычислительно сложных задач.

Протокол доказательства с нулевым разглашением, основанный на задаче нахождения Гамильтонова цикла, использует изоморфные графы. Два графа называются *изоморфными*, если они отличаются только названиями своих вершин. Задача создания графа, изоморфного известному, достаточно проста, например, можно осуществить случайную перестановку названий или номеров вершин графа, сохраняя связи между вершинами. Однако задача доказательства изоморфизма двух заданных графов, то есть нахождения такой перестановки, является вычислительно сложной и относится к классу NP (то есть не решаема за полиномиальное время).

Пусть сторона А знает цикл Гамильтона в большом графе G. Стороне В известен граф G, но не известен гамильтонов цикл в нем. А хочет доказать В, что он знает гамильтонов цикл, не сообщая при этом ни самого цикла, ни какой-либо информации о нем.

Для этого А и В могут использовать следующий интерактивный протокол:

1. Вначале А создает граф H, изоморфный исходному графу G. Преобразование Гамильтонова цикла между изоморфными графами является тривиальной задачей, поэтому если А известен гамильтонов цикл в G, то он также знает гамильтонов цикл в H.

2. А передает второй граф H проверяющему В.

3. В выбирает случайное значение бита $b \leftarrow \{0, 1\}$.

Если $b = 0$, то В просит А доказать изоморфизм графов G и H, то есть предоставить соответствие вершин этих двух графов. В может проверить, действительно ли G и H изоморфны.

Если b = 1, то В просит А показать Гамильтонов цикл в H.

4. Шаги 1–3 (раунд протокола) повторяется n раз, где n — параметр протокола.

Важно, что доказывающий А не знает заранее, какой из двух вопросов задаст проверяющий В. Если А не знает Гамильтонова цикла, то он сможет заранее подготовиться к ответу только на один из вопросов, а значит, ему удастся обмануть проверяющую сторону В только в половине случаев.

В каждом раунде В выбирает новый случайный бит, который неизвестен А, поэтому чтобы А мог правильно ответить на оба вопроса, нужно чтобы H был в самом деле изоморфен G и А знал бы гамильтонов цикл в H (а значит также и в G). Поэтому после достаточного числа раундов n, проверяющая сторона может быть уверена

в том, что А действительно знает Гамильтонов цикл в G, вероятность обмана составит 2^{-n}. С другой стороны, А не раскрывает никакой информации об этом цикле.

Данный протокол имеет большое теоретическое значение, так как М. Блюмом показано, что любое математическое утверждение может быть представлено в виде графа, причем доказательство этого утверждения соответствует Гамильтонову циклу в графе. Поэтому в общем случае любая NP-сложная задача может быть соответствующим образом преобразована в доказательство с нулевым разглашением конфиденциальной информации. К сожалению, на практике протокол не слишком эффективен, поскольку требует передачи больших объемов информации.

Существуют и другие схемы интерактивных доказательств с нулевым разглашением, например, доказательство того, что *число является числом Блюма*, доказательство *знания дискретного логарифма*.

Важным свойством интерактивных доказательств является невозможность для поверяющей стороны В продемонстрировать доказательство третьей стороне. В самом деле, уверенность проверяющего В в истинности доказываемого утверждения базируется на случайности выбора вопросов проверяющим. Если В запишет весь ход доказательства и предоставит его третьей стороне (назовем ее С), С это не убедит. С может сказать, например, что В заранее предупреждал А о том, какой вопрос он будет задавать, чтобы А мог подготовить правильный ответ. Таким образом, доказать утверждение стороне С без непосредственного участия, доказывающего А, невозможно.

Существует, однако, возможность мошенничества в протоколах доказательства с нулевым разглашением, когда В фактически является посредником между А и С, оперативно транслируя А вопросы С и передавая ответы А в обратном направлении. Таким образом, В получает возможность доказать стороне С свое знание утверждения, которым на самом деле владеет А. Защита от такого рода злоупотреблений заключается в точном задании времени, в которое должен проходить каждый этап протокола, что не оставляет времени для дополнительного обмена информацией.

Практическое значение протоколов аутентификации с нулевым разглашением заключается в том, что любой такой протокол может быть преобразован в схему цифровой подписи. Таким образом, если для конкретной системы электронной подписи удастся сформировать соответствующий протокол доказательства с нулевым разглашением, можно получить теоретическое обоснование стойкости этой системы электронной подписи.

4.4.3. Электронные деньги Чаума (электронная наличность)

Электронные деньги — это бессрочные денежные обязательства банковской или иной структуры, представленные в электронной форме, сопровождаемые электронной подписью выдавшей их структуры и погашаемые в момент предъявления обычными денежными средствами. Термин «электронные деньги» является сравнительно новым и применяется к достаточно широкому спектру платежных инструментов (электронные кошельки, предоплаченные карты: подарочные, топливные, телефонные и т.п., игровые деньги, криптовалюты), поэтому в настоящее время нет устоявшегося в мировой практике определения электронных денег, их экономическая и правовая сущность однозначно не определена.

При разработке концепции электронной наличности ставилась задача обеспечения неотслеживаемости платежей, наподобие анонимных наличных денег. То есть необходимо реализовать такую систему доступа к ресурсам и услугам, в которой одновременно решены задачи идентификации и анонимности. Были сформулированы свойства *идеальной* системы электронной наличности:

— *анонимность покупателя* (*неотслеживаемость*) — оплата электронными деньгами не позволяет определить, кто именно произвел оплату (связь между пользователем и его покупками обнаружить невозможно);

— *независимость от местоположения* — безопасность электронных денег не должна зависеть от места их физического хранения на носителях информации, они могут быть переданы по компьютерным сетям;

— *защита от мошенничества* — электронную купюру нельзя скопировать и использовать для оплаты повторно;

— *автономность* — осуществление платежа (протокол между покупателем и продавцом) не требует привлечения третьих лиц;

— *перемещаемость* — владелец может передавать электронные деньги другим пользователям;

— *делимость* — сумма может быть поделена на более мелкие части, чтобы покупатель мог получить сдачу.

На практике многие системы электронных денег обеспечивают, по крайней мере, первые четыре свойства. Ряд диалоговых систем электронных денег удовлетворяют всем свойствам, кроме автономности. Задача обеспечения анонимности электронных купюр решается с помощью технологии слепой подписи (blind signature), предложенной Д. Чаумом (David Chaum) с использованием криптосистемы RSA. Алгоритм слепой подписи использует *маскирующие* (*затемняющие*) *множители*, что позволяет удостоверить (подписать) информацию, не зная содержания.

В зависимости от реализации, протоколы электронной наличности используют *электронные купюры* (*банкноты* или *монеты*) — номинал фиксирован, требуются купюры различных номиналов, и *электронные чеки* — может быть использована любая сумма до заданного максимума, остаток средств может быть возвращен на счет пользователя. Рассмотрим далее протоколы с электронными купюрами.

Безопасность банка основывается на невозможности подделать его подпись для создания фальшивой банкноты, или, в более общем случае, на невозможности, получив набор подлинных электронных банкнот, подделать подпись еще хотя бы для одной купюры. Для неотслеживаемости покупателя необходимо, чтобы банк, получив купюру в транзакции платежа, не мог установить, кому она была выдана.

Пусть банк C выбрал параметры системы RSA: большие простые числа P, Q, $N = PQ$, L: НОД(e, $\varphi(N)$) = 1, $d = L^{-1} \bmod \varphi(N)$ и опубликовал (N, L) в качестве своего открытого ключа. Пара ключей (L, d) используется банком для создания купюр какого-то одного фиксированного номинала. Для создания купюр другого номинала используется иная пара ключей.

Пусть также выбрана односторонняя функция f: $\{0, \ldots, N - 1\} \to \{0, \ldots, N - 1\}$. В качестве односторонней функции f обычно используются криптографические хэш-функции. Функция f публикуется открыто и известна всем участникам протокола (банку C, покупателю A, продавцу B).

В транзакции снятия со счета покупатель получает подписанную банком электронную банкноту на затребованную сумму. При этом счет покупателя уменьшается на эту сумму. Протокол заказа (снятия со счета) электронной купюры:

1. Покупатель (клиент банка) A выбирает случайное число n, $1 < n < N$, которое фактически является номером купюры и держится в секрете до момента платежа. Затем он вычисляет $f(n)$.

2. A генерирует маскирующий множитель — случайное число r, взаимно простое с N: $1 < r < N$, НОД(r, N) = 1. Затем A вычисляет $n' = f(n)r^L \bmod N$ и отсылает число n' банку C.

3. Банк C подписывает пришедшую купюру $s' = n'^d \bmod N$ и отправляет значение s' клиенту A, сняв сумму, равную номиналу купюры, с его счета.

$s' = n'^d \bmod N = (f(n)r^L)^d \bmod N = f(n)^d r^{Ld} \bmod N = f(n)^d r \bmod N$

4. Клиент A снимает маскирующий множитель, получив с помощью расширенного алгоритма Евклида число $r^{-1} \bmod N$ и вычислив $s = s'r^{-1} \bmod N = f(n)^d r r^{-1} \bmod N = f(n)^d \bmod N$. Кроме того, клиент A проверяет подпись банка на купюре: $s^L \bmod N = f(n)^{dL} \bmod N = f(n)$. Таким образом, клиент A получает подписанную банком купюру (n, s).

В полном протоколе дополнительно используются шаги формирования и проверки подписи клиента A при отправке n', что позволяет банку C идентифицировать клиента и убедиться, что купюра пришла от него в неизмененном виде.

Безопасность банка в этой системе основывается на вере в стойкость схемы электронной подписи RSA. Применение функции f необходимо ввиду известного свойства мультипликативности схемы RSA: если s_1 и s_2 – подписи n_1 и n_2 соответственно, то $s_1 s_2$ – подпись $n_1 n_2$. Поэтому, если бы в системе электронных платежей использовались банкноты вида $(n, n^d \bmod N)$ вместо $(n, f(n)^d \bmod N)$, то из двух подлинных банкнот всегда можно было бы изготовить третью $(n_1 n_2, s_1 s_2)$. В самом деле, $(n_1 n_2)^d \bmod N = (n_1^d \bmod N) \cdot (n_2^d \bmod N) = s_1 s_2$.

Использование слепой подписи гарантирует неотслеживаемость клиентов. Все, что остается у банка от транзакции снятия со счета, — это значение $f(n)^d r \bmod N$, которое благодаря маскирующему множителю r представляет собой случайное число в диапазоне от 0 до $N - 1$. Поэтому у банка нет информации о том, какую именно купюру он выдал данному клиенту (банк знает номинал купюры, но не ее номер).

В транзакции платежа покупатель передает банкноту продавцу и указывает сумму платежа. Продавец, в свою очередь, передает эту информацию банку, который проверяет подлинность купюры. Банк проверяет свою цифровую подпись, и если купюра подлинная, то определяет, не была ли она потрачена ранее. Если нет, то банк заносит номер купюры в специальный реестр, и зачисляет требуемую сумму на счет продавца, а затем уведомляет его о том, что платеж принят. Протокол платежа с использованием электронной купюры:

1. Покупатель A передает продавцу B электронную купюру (n, s).

2. Продавец B вычисляет $f(n)$ и проверяет подпись банка: $s^L \bmod N = f(n)$. Убедившись, что купюра подлинная, он посылает (n, s) банку C.

3. Банк C вычисляет $f(n)$ и проверяет свою подпись, убеждаясь в справедливости равенства $f(n)^d \bmod N = s$.

4. Банк C проверяет, не была ли купюра с данным номером потрачена ранее, и, если нет, перечисляет на счет клиента B сумму, равную номиналу купюры, после чего уведомляет его об этом.

В полном протоколе дополнительно используются шаги формирования и проверки подписи клиента B на документе (n, s).

Таким образом, гарантируется анонимность покупателя, но не продавца. Рассмотренная система электронных платежей позволяет определить, сколько денег получил (заработал) клиент, но как он их потратил — определить невозможно.

В рассмотренной системе электронных платежей проблема повторной оплаты решается *централизовано* банком, который ведет реестр номеров использованных купюр. Если продавец предъявил банку купюру, номер которой уже содержится в реестре, банк откажется принять платеж. Рассмотренный протокол защищает банк от повторного использования купюр, но не устанавливает личности мошенников — невозможно определить, кто повторно использовал купюру — покупатель A или продавец B.

С другой стороны, клиентам приходится полностью полагаться на честность банка, поскольку не существует возможности независимой проверки использованных банкнот. Недобросовестный банк может, например, присвоить купюру, заявив, что она уже была использована ранее. Возможны и другие злоупотребления, как со стороны покупателя, так и со стороны продавца, характерные для использования любых сертификатов на предъявителя.

Еще одной возможностью является использование купюры мошенником, перехватившим ее номер в процессе обмена информацией между покупателем и продавцом, и успевшим предъявить в банк раньше продавца.

Система электронной наличности Д. Чаума имеет и другие недостатки:

1. Два разных клиента (или один и тот же клиент, не ведущий записи ранее использованных банкнот), могут запросить у банка купюры с одинаковым номером. Банк вслепую подпишет обе купюры. Действительной же будет признана только одна из них — та, которая предъявлена к оплате первой. Правда, для практически используемых значений N вероятность такого совпадения достаточно мала.

2. Покупатель не может доказать факт оплаты конкретного товара (например, в случае возврата некачественного товара).

3. Размер реестра использованных купюр растет с каждым новым платежом, а поиск в нем — замедляется. Для ограничения размеров реестра может быть ограничен *период оперативной платежеспособности* купюр. По истечении этого периода номер купюры удаляется из реестра использованных, а значит, купюра может использоваться повторно. Поэтому период оперативной платежеспособности купюр не должен быть слишком коротким.

4. Электронные купюры неделимы, поэтому клиентам придется дополнительно обращаться в банк за их разменом, чтобы заплатить продавцу точную сумму. Рост числа купюр в платеже усложняет операцию платежа и увеличивает время его выполнения (например, за счет увеличения времени проверки купюр). Для решения этой проблемы Д. Чаумом был предложен *протокол слепого размена* электронной наличности.

Система электронной наличности Д. Чаума является *централизованной*, то есть требует участия банка во всех транзакциях. *Автономная* система платежей предполагает, что продавец сам, без обращения к банку, проверяет подлинность предъявленной покупателем электронных денег. В этом случае банк идет на определенный риск, так как используемые схемы не обеспечивают защиты от злоупотреблений со стороны покупателя, но позволяют их обнаружить постфактум. Поэтому протоколы автономных систем электронных денег должны однозначно идентифицировать нарушителя. Тогда за счет введения крупных штрафов можно сделать такие злоупотребления невыгодными.

Следует также отметить возможные законодательные ограничения на использование электронных денег и, в частности, криптовалют. Так, Министерство финансов Российской Федерации считает последние денежными суррогатами, которые запрещены федеральным законом «О Центральном банке Российской Федерации (Банке России)»[16], а операции с криптовалютой рассматривает как потенциально сомнительные, однако на практике вопрос законности криптовалют в России остается неурегулированным.

[16] Федеральный закон «О Центральном банке Российской Федерации (Банке России)» от 10.07.2002 № 86-ФЗ. Нормативно-правовые акты приведены в соответствии с данными официального интернет-портала правовой информации Pravo.gov.ru (дата обращения: 12.01.2023).

4.4.4. Создание скрытого канала в системе рандомизированной электронной подписи

С помощью алгоритма цифровой подписи, например, алгоритма Эль-Гамаля, может быть организован скрытый (подсознательный) канал передачи информации. При этом скрываемое сообщение содержится в цифровой подписи передаваемых открытых сообщений. Извлечь скрытую таким образом информацию может только человек, обладающий общим с отправителем секретным ключом.

Проверяющая сторона (цензор) С не только не сможет прочитать сообщение, передаваемое по скрытому каналу, но у него вообще нет ни малейшего представления о существовании такого сообщения.

Проверяющий С видит, как подписанные безобидные сообщения передаются между отправителем и получателем, но реальная передаваемая информация проходит незаметно для него по скрытому каналу.

В общем случае организация скрытого канала выглядит следующим образом:

1. Отправитель А создает безобидное сообщение.

2. Используя общий с получателем секретный ключ k_{AB}, А подписывает безобидное сообщение, пряча свое подсознательное сообщение в подписи.

3. А посылает подписанное сообщение по открытому каналу передачи данных.

4. Проверяющий С читает сообщение и проверяет подпись. Не обнаружив ничего подозрительного, он передает сообщение дальше.

5. Получатель В проверяет подпись под безобидным сообщением, убеждаясь, что сообщение получено от А.

6. В игнорирует безобидное сообщение и, используя общий с отправителем секретный ключ k_{AB}, извлекает подсознательное сообщение.

В действительности, алгоритм создания скрытого канала в подписях практически не отличим от нормального алгоритма создания подписи. Для системы цифровых подписей Эль-Гамаля единственным отличием является то, что в роли случайного числа k выступает подсознательное сообщение MS.

Сначала формируются общие (открытые) параметры системы Эль-Гамаля, то есть числа p и g, выбор которых осуществляется так же, как и в системе Диффи-Хеллмана. Затем отправитель выбирает случайное число x, $1 < x < p-1$ и вычисляет $y = g^x \bmod p$. В системе Эль-Гамаля число x является личным ключом отправителя, число y — его открытым ключом. Для организации скрытого канала число x должно быть общим секретным ключом абонентов, то есть отправитель должен заранее надежным способом передать число x получателю.

Отправитель А создает безобидное сообщение M и скрытое сообщение MS, такие, что: $0 < M < p$ и $0 < MS < p-1$, MS — взаимно простое с $p-1$.

Отправитель А формирует цифровую подпись сообщения, вычисляя:

$r = g^{MS} \bmod p$,

$u = (M - xr) \bmod (p-1)$,

$s = MS^{-1}u \bmod (p-1)$.

А формирует подпись и в то же время создает скрытый канал, пересылая сообщение M и подпись (r, s) получателю В.

Проверяющая сторона С может удостовериться в подлинности цифровой подписи сообщения M, проведя стандартные для этого случая вычисления: $y^r r^s \bmod p = g^M \bmod p$.

Получатель В также проверяет подлинность подписи, удостоверяясь, что сообщение получено именно от того отправителя, с которым организован скрытый канал. Затем получатель В, зная общий секретный ключ x, извлекает скрытое сообщение:

$u = (M - xr) \bmod (p-1)$,

$MS = s^{-1}u \bmod (p-1)$.

Как видно, для получения MS требуется вычислить число, обратное s по модулю p–1. Таким образом, s должно быть взаимно простым с p–1 (НОД(s, p–1)=1). Этого можно добиться, варьируя значение сообщения M. В приведенных формулах вместо M может использоваться его хэш-значение $h(M)$.

Пример. Пусть выбраны параметры системы Эль-Гамаля $p = 2*5 + 1 = 11$ ($q = 5$), $g = 2$, $2^5 \bmod 11 = 10 \neq 1$.

Пусть абонент выбрал личный ключ $x = 8$ и вычислил открытый ключ $y = 2^8 \bmod 11 = 3$. Пусть абонент надежным способом передал получателю секретный ключ $x = 8$.

Пусть теперь абонент хочет передать скрытое сообщение $MS = 9$, используя цифровую подпись Эль-Гамаля и открытое сообщение $M = 5$. $MS = 9$ взаимно просто с $p - 1 = 10$.

Абонент вычисляет: $r = g^{MS} \bmod p = 2^9 \bmod 11 = 6$, $u = (M - xr) \bmod (p - 1) = (5 - 8*6) \bmod 10 = 7$. Используя расширенный алгоритм Евклида, он получает $MS^{-1} \bmod (p - 1) = 9^{-1} \bmod 10 = 9$. Затем вычисляет $s = MS^{-1}u \bmod (p - 1) = 9 * 7 \bmod 10 = 3$.

Полученное число $s = 3$ взаимно просто с $p - 1 = 10$, значит, получатель сможет восстановить скрытое сообщение MS.

Абонент отсылает сообщение $M = 5$ и подпись (6, 3) получателю.

Любой проверяющий может убедиться в подлинности подписи, сравнив значения $y^r r^s \bmod p = 3^6 * 6^3 \bmod 11 = 10$ и $g^M \bmod p = 2^5 \bmod 11 = 10$, 10 = 10.

Получатель также проверяет цифровую подпись под сообщением M и удостоверяется, что получил сообщение от нужного абонента. Затем он вычисляет: $u = (M - xr) \bmod (p - 1) = (5 - 8*6) \bmod 10 = 7$. Используя расширенный алгоритм Евклида, он получает значение $s^{-1} \bmod (p - 1) = 3^{-1} \bmod 10 = 7$. Тогда $MS = s^{-1}u \bmod (p - 1) = 7 * 7 \bmod 10 = 9$. Скрытое сообщение восстановлено правильно.

Скрытый канал может быть организован и с использованием других рандомизированных схем цифровой подписи (Онга — Шнорра, Рабина). Создаваемые скрытые каналы имеют достаточно высокую пропускную способность, так, при использовании схемы Эль-Гамаля она составит около половины длины подписи, поскольку длина MS сопоставима с длиной p. Главным же недостатком такого подхода является необходимость передачи получателю личного ключа x другого абонента (в схеме Рабина передачи ключа не требуется).

Для защиты от организации скрытого канала на основе схемы Эль-Гамаля в протокол цифровой подписи вводится арбитр D, который участвует в выборе случайного элемента (числа k) и имеет возможность проверить, что при формировании подписи было использовано именно это число, не зная его значения. В обычный протокол формирования цифровой подписи добавляются следующие шаги:

1a. Отправитель A сообщения выбирает случайное число k, формирует значение $r = g^k \bmod p$ и отправляет его арбитру D, выступающему в роли проверяющей стороны.

1b. Арбитр D выбирает случайное число k' и направляет его отправителю.

1c. Отправитель использует kk' в качестве случайного числа для формирования подписи, так, он вычисляет первую часть подписи как $r' = g^{kk'} \bmod p$. Подпись направляется арбитру D.

Теперь D может удостовериться, что отправитель использовал именно число kk', проверяя выполнение равенства $r' = r^{k'} \bmod p$.

Кроме того, существуют рандомизированные схемы цифровой подписи (схемы на основе решения системы сравнений), не допускающие встраивания скрытого канала.

Таким образом, в четвёртой главе были рассмотрены криптографические протоколы, проанализированы определение и типы криптографических протоколов, протоколы, связанные с электронной подписью, протоколы аутентификации и распределения ключей, специальные криптографические протоколы. Далее необходимо рассмотреть организационно-правовые аспекты применения криптографических систем.

Вопросы для самоконтроля:

1. Что понимается под разделением секрета? Какова простейшая схема разделения секрета между двумя участниками?

2. Какими свойствами должен обладать протокол доказательства с нулевым разглашением?

3. Какие математические задачи лежат в основе протоколов доказательства с нулевым разглашением?

4. Каковы свойства идеальной системы электронной наличности?

5. Каковы меры защиты от организации скрытого канала в подписях?

Глава 5
ОРГАНИЗАЦИОННО-ПРАВОВЫЕ АСПЕКТЫ ПРИМЕНЕНИЯ КРИПТОГРАФИЧЕСКИХ СИСТЕМ

§ 5.1. Гражданско-правовое значение криптографии

Длительное время (примерно до середины 1970-х годов в мире, а в нашей стране — до начала 1990-х годов) криптография использовалась только военными, дипломатическими, разведывательными и другими государственными структурами и считалась сугубо секретной областью. Однако внедрение компьютерных систем и сетей во все сферы профессиональной деятельности и общественной жизни потребовало использования криптографии и для гражданских нужд. Сегодня стал привычным термин «гражданская криптография»[17], обозначающий общедоступную криптографию, в противоположность специальной государственной (здесь термин «гражданская» не несет юридического смысла).

На сегодняшний день криптография является одним из базовых методов комплексного обеспечения информационной безопасности автоматизированных систем, в том числе в таких областях, как мобильная телефонная связь стандарта GSM, интернет-мессенджеры и IP-телефония; международные платежные системы Visa, MasterCard и др., обеспечивающие операции по банковским картам; защитные функции массового программного обеспечения, используемого в домашних и офисных компьютерных системах (безопасная передача информации в интернет-браузерах; шифрование и электронная подпись в приложениях MS Office, документах Adobe pdf и др.).

Криптография стала неотъемлемым спутником бизнеса в банковской и финансовой сферах для решения таких задач, как обмен электронными платежными документами в системе расчетно-кассовых центров Центрального банка Российской Федерации с использованием электронной подписи и шифрования; управление расчетным счетом в системах «Клиент-Банк», построенных на основе электронной подписи; международные межбанковские расчеты (система SWIFT) с использованием программно-аппаратных средств криптографической защиты информации (СКЗИ); удаленная торговля ценными бумагами на фондовом рынке (РТС, фондовая секция ММВБ и др.) с использованием терминалов на основе технологии электронной подписи; предоставление по открытым каналам связи (по электронной почте) налоговой декларации и бухгалтерской отчетности в электронном виде и др.

Широкое использование криптографических технологий ставит вопрос об их соотношении с важнейшими институтами гражданского права. Гражданско-правовое значение криптографии характеризуется следующим образом.

1. Криптографические методы могут использоваться как *средство обособления* невещественных объектов гражданских прав, если они физически представлены в виде записей, электронных документов (файлов).

Обособление (индивидуализация) является ключевым вопросом для любых объектов абсолютных гражданских прав, будь то вещи или невещные объекты гражданских прав (результаты интеллектуальной деятельности и т. д.). Кратко, но емко, эта мысль можно сформулировать как: «Объектом всякого абсолютного права может быть только уникальная или индивидуализированная субстанция».

Существуют три основных способа обособления (индивидуализации) объектов исключительных прав:

[17] Васильева И. Н., Куватов В. И. Криптографическая защита информации: практикум. СПб.: СПбУ МВД России, 2017. 260 с.

— обособление в силу уникальной формы объекта, что характерно для объектов авторского права;

— обособление формализацией признаков объекта, в рамках регистрационной системы (это характерно для объектов патентного права);

— обособление посредством сохранения конфиденциальности объекта (такой способ используется применительно к ноу-хау).

Произведения, выраженные в цифровой форме, представляют собой легко копируемую информацию (файл), копии которой даже теоретически не имеют своей уникальности. Невозможно различить оригинал и копию, в том числе и ту копию, которая сделана незаконно. Данное обстоятельство затрудняет реализацию и защиту авторских прав их обладателями. Для дополнительного обособления объектов авторского права, представленных в электронно-цифровой форме, могут быть использованы средства криптографии и стеганографии.

Примером может служить встраивание цифровых водяных знаков с использованием электронных цифровых подписей сторон, участвующих в процессе передачи авторского права. Это позволяет правообладателю выявить нарушителя при появлении контрафактных копий его продукта, обнаружить и подтвердить факт его незаконного использования. Кроме того, лицензиат продукта в случае ложного обвинения может доказать факт законного использования.

Подобные методы реализуются в технических средствах защиты авторских прав DRM (Digital Rights Management) — программных или программно-аппаратных средства контроля и управления доступом, которые намеренно ограничивают либо затрудняют различные действия с данными в электронной форме (копирование, модификацию, просмотр и т.п.), либо позволяют отследить такие действия. Примером использования DRM являются «электронные книги» формата *.pdf компании Adobe. Большинство современных систем DRM используют криптостойкие алгоритмы защиты, однако эти методы не могут использоваться полноценно, поскольку основаны на предположении, что для получения доступа к зашифрованной информации требуется секретный ключ. Однако типичной является ситуация, когда ограничения обходятся правомерным обладателем копии, поэтому системы DRM пытаются скрыть от пользователя используемый ключ шифрования, что затруднено ввиду контроля пользователем устройств воспроизведения.

Поскольку DRM малоэффективны сами по себе, для них установлена правовая защита. Так, в Российской Федерации ст. 1299 IV части Гражданского кодекса Российской Федерации[18] запрещает обход средств DRM. За указанные нарушения предусмотрена гражданско-правовая (ст. 1301 ГК РФ) и административная (ст. 7.12 Кодекса Российской Федерации об административных правонарушениях[19]) ответственность.

Другим подходом является использование сервисов электронной регистрации (депонирования) произведений, обеспечивающих постановку электронной подписи со штампом времени. Подобные сервисы, по сути, обеспечивают предварительную защиту авторских прав путем предоставления свидетельств существования у автора определенных объектов авторских прав на конкретную дату.

Широко практикуется подписание программного обеспечения (отдельных драйверов, обновлений программ и др.), здесь электронная подпись подтверждает, что данной файл действительно исходит от правообладателя первоначальной программы, а не является подброшенным программой-вирусом. Возможность обособления с помощью электронной подписи произведений в электронной форме также ценна тогда,

[18] Гражданский кодекс Российской Федерации (далее — ГК РФ). Ч. IV от 18.12.2006 № 230-ФЗ.

[19] Кодекс Российской Федерации об административных правонарушениях от 30.12.2001 № 195-ФЗ (далее — КоАП РФ).

когда на одной вещи (компьютер-сервер) физически размещается несколько различных охраняемых произведений, принадлежащих разным лицам (т. н. «хостинг»).

Обособление объекта гражданских прав может быть осуществлено и путем фактического ограничения доступа к нему. Для объектов, физически представленных в виде записей (файлов, электронных документов), ограничение доступа чаще всего организуется путем изменения формы представления нематериального объекта, перевод его в нечитаемую форму, здесь речь идет о шифровании информации; собственно ограничением доступа средствами технической защиты информации (парольная защита, межсетевые экраны, экранированные ЭВМ и др.).

Например, для такого объекта гражданских прав, как информация, составляющая коммерческую тайну, ограничение доступа является одним из условий охраноспособности (ч. 1, ч. 2 и ч. 4 ст. 10 федерального закона от 29.07.2004 № 98-ФЗ «О коммерческой тайне»). И поскольку ограничение доступа может быть достигнуто путем шифрования, то шифрование такой информации следует квалифицировать как факт, свидетельствующий о принятии мер к охране ее конфиденциальности. Таким образом, шифрование может выступать как часть юридического состава, порождающего возникновение субъективного права на информацию, составляющую коммерческую тайну.

2. Шифрование может использоваться в качестве превентивной *меры самозащиты гражданских прав*. Речь идет, в основном, о самозащите абсолютных гражданских прав на конфиденциальную информацию (коммерческая тайна, личная тайна, семейная тайна). Очевидно, что нарушителю сложнее нарушить право на конфиденциальную информацию, если она зашифрована надежным СКЗИ. Суть самозащиты в этом случае заключается в том, что фактическими мерами затруднен неправомерный доступ к охраняемой информации, ее прочтение и понимание.

При этом возможна защита как хранимой информации (использование шифрованной файловой системы EFS в MS Windows, утилит по созданию секретных дисков, шифрование документов на уровне приложений), так и защита внутрикорпоративного обмена конфиденциальной информацией за счет шифрования трафика средствами VPN (Virtual Personal Network).

Самозащита гражданских прав путем использования шифрования может осуществляться как непосредственно действиями самого обладателя защищаемых прав, так и силами третьих лиц, оказывающих соответствующие услуги. В последнем случае следует учитывать лицензируемость услуг шифрования.

Можно также говорить о самозащите средствами криптографии обязательственных гражданских прав. Например, в отношении права распоряжаться денежными средствами, находящимися на банковском счете, шифрование транзакций и электронная подпись выступают средством, фактически затрудняющим злоумышленнику посягнуть на эти права, сфальсифицировав платежное поручение. В этом случае имеется тесная связь и с правом на имя.

3. Криптография может использоваться как *средство реализации личного неимущественного права на имя*. Принято считать, что подпись (право на подпись) представляет собой нематериальное благо, при этом тесно связанное с правом на имя (п. 1 ст. 19 ГК РФ) и выражающее это право. Право на подпись является личным неотчуждаемым правом, позволяющим человеку индивидуализировать себя среди прочих лиц, а также свою волю в сделках.

Принципиально важным является то, что законодатель в ст. 19 ГК РФ[20] (п. 1, 2, 4) обязывает гражданина приобретать и осуществлять гражданские права, а равно принимать

[20] Гражданский кодекс Российской Федерации. Ч. I от 30.11.1994 № 51-ФЗ.

на себя и исполнять обязанности только под собственным именем, а, значит, и под своей подписью. Следовательно, использование электронной подписи для заключения сделок необходимо признать способом реализации права и обязанности действовать под своим именем. Отметим, что право на имя корректнее увязывать не с самой цифровой подписью, а с закрытым ключом, так как. результат подписания уникален для каждого конкретного документа, поскольку зависит от содержания документа. Право на имя (в том числе, реализуемое в форме использования электронной подписи) непосредственно связано с вопросами волеобразования и волеизъявления юридического лица. Право и обязанность человека использовать свою уникальную подпись, отражающую его имя, не меняются в зависимости от приобретения/утраты им статуса органа юридического лица. Личность человека и его статус органа юридического лица нераздельны: поэтому подпись И.И. Иванова (как человека) не может существовать автономно от подписи того же И.И. Иванова, но уже как директора юридического лица. Подпись едина, как едина личность человека. Неимущественное право на подпись (в т. ч. электронную подпись) присуще исключительно человеку, но никак не юридическому лицу.

4. Средства криптографии могут использоваться как *средство обеспечения надлежащей формы совершаемой сделки*. В зависимости от техники заключения сделок, выделяют сделки между присутствующими контрагентами (в их непосредственном личном присутствии), а также сделки между отсутствующими контрагентами путем обмена волеизъявляющими сообщениями (оферта/акцепт) в письменном виде, в том числе и с использованием компьютерных средств связи.

Действующее российское законодательство в п. 2 ст. 160 ГК РФ допускает при совершении сделок использовать различные аналоги собственноручной подписи, в том числе электронную подпись. А в п. 2 ст. 434 ГК РФ законодатель разрешил заключать договор в письменной форме путем обмена документами посредством электронной или иной связи, позволяющей достоверно установить, что документ исходит от стороны по договору (договор между отсутствующими).

Юридическая значимость документов, заверенных электронной подписью, обеспечивается федеральным законом от 06.04.2011 № 63-ФЗ «Об электронной подписи». Федеральный закон «Об электронной подписи» вводит понятия *простой* и *усиленной* электронной подписи, последняя может быть *усиленной неквалифицированной* и *усиленной квалифицированной* электронной подписью.

Усиленная электронная подпись должна быть получена в результате криптографического преобразования информации с использованием ключа электронной подписи, обеспечивает целостность электронного документа и позволяет определить лицо его подписавшее. *Усиленная квалифицированная* электронная подпись (квалифицированная подпись) дополнительно предполагает использование квалифицированных сертификатов, а также средств генерации и проверке подписи, отвечающих специальным требованиям. На практике это означает необходимость поддержки отечественных криптоалгоритмов, сертификацию ФСБ России используемого программного обеспечения с криптографическими функциями и выдачу сертификата удостоверяющим центром, использующим сертифицированные ФСБ России СКЗИ.

Использование усиленной квалифицированной подписи приравнивает подписанный электронный документ к документу на бумажном носителе, заверенному собственноручно. Электронные документы, заверенные другими типами подписей, могут иметь юридическую силу по соглашению сторон (участниками электронного взаимодействия должны быть предварительно приняты соответствующие нормативные правовые акты и/или соглашения). В государственных информационных системах используется только усиленная квалифицированная электронная подпись.

5. Криптография может являться *средством надлежащего исполнения договорного обязательства*. Например, шифрование применимо для исполнения такого специфического обязательства, как обязательство по профессиональной тайне (к разновидностям которой относятся банковская тайна, аудиторская, тайна страхования, тайна связи и т. п.). В качестве распространенного примера можно назвать шифрование телефонного трафика в стандарте GSM — будучи способом оператора связи исполнить свое обязательство по обеспечению тайны связи.

Стоит отметить, что исполнение обязательства по сохранению профессиональной тайны (банковской тайны и др.) нельзя квалифицировать как услугу по шифрованию, поскольку здесь СКЗИ используется в собственных нуждах, в целях исполнения своей обязанности по конфиденциальности.

Электронная подпись, кроме того, может использоваться в качестве технического средства обеспечения процедуры исполнения обязательства надлежащему лицу (ст. 312 ГК РФ). Технический термин «аутентификация» может характеризовать соотнесение лица, которое фактически принимает исполнение, с личностью кредитора. Особое значение электронная подпись как средство аутентификации приобретает, когда речь идет об обязательствах, которые исполняются удаленно, с использованием средств связи (в частности, информационные услуги бюро кредитных историй).

Процедура исполнения обязательства предполагает право должника зафиксировать факт исполнения, потребовав от кредитора расписку в этом (п. 2 ст. 408 ГК РФ). Электронная подпись также может служить целям реализации такого права, если исполнение обязательства происходит с использованием средств связи.

6. Средства электронной подписи могут быть использованы как техническое *средство, обеспечивающее реализацию права на участие в управлении юридическим лицом* (неимущественное право, относимое к числу корпоративных прав).

Так, п. 1 ст. 38 федерального закона от 08.02.1998 № 14-ФЗ «Об обществах с ограниченной ответственностью» прямо предусмотрено, что голосование (опросным путем) на заочном общем собрании участников общества может быть проведено путем обмена документами посредством электронной или иной связи, обеспечивающей аутентичность передаваемых и принимаемых сообщений и их документальное подтверждение. Указанные требования законодателя легко выполнить, используя технологию электронной подписи.

7. Средства шифрования или электронной подписи не только обеспечивают реализацию субъективных гражданских прав и исполнение обязанностей, но и сами *могут выступать объектом гражданских прав*. Используемые в деловой практике СКЗИ представляют собой аппаратные модулю или программное обеспечение, реализующие различные криптографические алгоритмы и протоколы. Соответственно, СКЗИ могут быть квалифицированы:

— как охраняемое авторским правом программное обеспечение: программы для ЭВМ и базы данных (на практике, большинство СКЗИ представлено программным обеспечением);

— исходный код («исходники») криптографического программного обеспечения может охраняться в качестве коммерческой тайны;

— как движимая вещь, если речь идет об СКЗИ в виде аппаратных средств. К таковым относятся, например, аппаратные шифрующие модули; аппаратные средства VPN; устройства хранения криптографических ключей (eToken) и др. В некоторых случаях можно ставить вопрос о патентоспособности аппаратных СКЗИ.

Важно подчеркнуть, что, по общему правилу, СКЗИ не являются ограниченно оборотоспособным объектом (ст. 129 ГК РФ), даже если речь идет о сертифицированных

(аттестованных) СКЗИ. Разумеется, к оборотоспособным не относятся секретные СКЗИ, которые могут создаваться для специальных государственных нужд (вооруженные силы, разведка, дипломатия и т.п.). Такие средства будут являться объектами, изъятыми их гражданского оборота.

Вопросы для самоконтроля:

1. *Каково применение криптографии в банковской и финансовой сферах?*
2. *Каково гражданско-правовое значение криптографии?*
3. *Каким образом криптографические методы могут быть использованы для обособления (индивидуализации) объектов исключительных прав?*
4. *Какие виды электронной подписи определены федеральным законом «Об электронной подписи»?*
5. *В какой роли могут выступать СКЗИ как объекты гражданских прав?*

§ 5.2. НОРМАТИВНО-ПРАВОВЫЕ ОСНОВЫ ИСПОЛЬЗОВАНИЯ ГРАЖДАНСКОЙ КРИПТОГРАФИИ

В основе использования криптографических средств частными лицами лежат как общие гарантии экономической деятельности, закрепленные в ст. 18, 34, 55 Конституции Российской Федерации[21], так и ряд положений ГК РФ и других нормативно-правовых актов. Практический интерес представляет проблема соотношения публичного и гражданского права в этой сфере, нашедшее отражение, прежде всего, в вопросе: необходимы ли специальные разрешения, лицензии и прочие формальности публично-правового характера, чтобы приобретать и использовать средства электронной цифровой подписи, либо чтобы шифровать свою коммерческую или личную тайну?

Продолжает действовать указ Президента Российской Федерации от 03.04.1995 № 334 «О мерах по соблюдению законности в области разработки, производства, реализации и эксплуатации шифровальных средств, а также предоставления услуг в области шифрования информации», п. 4 которого гласит: «В интересах информационной безопасности Российской Федерации и усиления борьбы с организованной преступностью запретить деятельность юридических и физических лиц, связанную с разработкой, производством, реализацией и эксплуатацией шифровальных средств, а также защищенных технических средств хранения, обработки и передачи информации, предоставлением услуг в области шифрования информации, без лицензий, выданных Федеральным агентством правительственной связи и информации при Президенте Российской Федерации в соответствии с законом Российской Федерации “О федеральных органах правительственной связи и информации”».

Процитированный п. 4 указа противоречит положениям ст. 18, 34 и 55 Конституции Российской Федерации, поскольку незаконно ограничивает конституционные права физических и юридических лиц на свободное использование своих способностей и имущества для предпринимательской и иной не запрещенной законом экономической деятельности.

В федеральном законе от 04.05.2011 № 99-ФЗ «О лицензировании отдельных видов деятельности» и постановлении Правительства Российской Федерации от 16.04.2012 № 313 «Об утверждении Положения о лицензировании деятельности по разработке, производству, распространению шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифро-

[21] Конституция Российской Федерации (принята всенародным голосованием 12.12.1993; с изменениями, одобренными в ходе общероссийского голосования 01.07.2020).

вальных (криптографических) средств, выполнению работ, оказанию услуг в области шифрования информации, техническому обслуживанию шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств (за исключением случая, если техническое обслуживание шифровальных (криптографических) средств, информационных ...» деятельность по использованию юридическими и физическими лицами средств для шифрования своей информации (в частности, коммерческой или личной тайны) не включена в список лицензируемой. Таким образом, можно утверждать, что приобретение и использование СКЗИ в собственных интересах (без оказания услуг третьим лицам) свободно, и не требует получения лицензий или выполнения иных формальностей публично-правового характера.

Согласно п. 1 ст. 12 федерального закона от 04.05.2011 № 99-ФЗ лицензированию подлежат «разработка, производство, распространение шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств, выполнение работ, оказание услуг в области шифрования информации, техническое обслуживание шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств (за исключением случая, если техническое обслуживание шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств, осуществляется для обеспечения собственных нужд юридического лица или индивидуального предпринимателя)».

В настоящее время действует также приказ ФСБ России от 09.02.2005 № 66 «Об утверждении положения о разработке, производстве, реализации и эксплуатации шифровальных (криптографических) средств защиты информации (положение пкз-2005)», который определяет порядок разработки и эксплуатации криптографических средств. В частности, согласно приказу, средства криптографии реализуются «юридическим лицом или индивидуальным предпринимателем, имеющим право на осуществление данного вида деятельности, связанного с шифровальными (криптографическими) средствами… вместе с правилами пользования ими, согласованными с ФСБ России».

Согласно указу Президента Российской Федерации от 03.04.1995 № 334, для защиты государственных информационных систем требуется использование сертифицированных СКЗИ, п. 2 этого указа гласит: «Запретить использование государственными организациями и предприятиями в информационно-телекоммуникационных системах шифровальных средств, включая криптографические средства обеспечения подлинности информации (электронная подпись), и защищенных технических средств хранения, обработки и передачи информации, не имеющих сертификата Федерального агентства правительственной связи и информации при Президенте Российской Федерации, а также размещение государственных заказов на предприятиях, в организациях, использующих указанные технические и шифровальные средства, не имеющие сертификата Федерального агентства правительственной связи и информации при Президенте Российской Федерации».

Согласно постановлению Правительства Российской Федерации от 01.11.2012 № 1119 «Об утверждении требований к защите персональных данных при их обработке в информационных системах персональных данных» и «Методическим рекомендациям по обеспечению с помощью криптосредств безопасности персональных данных при их обработке в информационных системах персональных данных с использованием средств автоматизации», утвержденные руководством 8 Центра ФСБ России 21.02.2008 № 149/54-144, использования сертифицированных ФСБ России СКЗИ потребуется и для защиты персональных данных.

Использование сертифицированных СКЗИ рекомендуется и для защиты банковской информации.

Согласно постановлению Правительства Российской Федерации от 21.11.2011 № 957, регулирующим органом, осуществляющим лицензирование видов деятельности, связанных с шифровальными (криптографическими) средствами, является ФСБ России. ФСБ России также организует систему сертификации СКЗИ. Требования ФСБ России к СКЗИ не являются открытыми.

Итак, в пятой главе были рассмотрены организационные и правовые составляющие организации функционирования криптографических систем, в том числе гражданско-правовое регулирование в этой сфере, интеграция криптографии с гражданским обществом.

Вопросы для самоконтроля:

1. Каковы ограничения на использование СКЗИ в государственных информационных системах?

2. В каких случаях требуется использование сертифицированных СКЗИ?

3. Какая деятельность, связанная с шифровальными (криптографическими) средствами, подлежит обязательному лицензированию?

4. Можно ли считать свободным использование криптографических средств для собственных нужд?

5. Какой орган является регулятором в сфере деятельности, связанной с криптографическими средствами?

ЗАКЛЮЧЕНИЕ

В учебном пособии рассмотрены вопросы построения криптосистем и криптоанализа, начиная с исторических шифров и до современных криптографических методов, применимых в компьютерных системах и сетях; приведены основные понятия и определения, сформулированы цели и задачи современной криптографии; приведены сведения из истории криптографии, необходимые для понимания классификации криптосистем; даны основные теоретические положения современной криптографии; приведены формальные модели криптосистем; рассмотрены вопросы теоретической и практической (вычислительной) стойкости криптосистем и имитостойкости; приведено описание блочных и потоковых симметричных криптосистем, включая современные американский и отечественный стандарты шифрования, описаны методы формирования криптографических хэш-функций, включая отечественный стандарт на функцию хэширования; рассмотрены основные асимметричные криптосистемы, системы цифровой подписи сообщений, включая российский стандарт электронной подписи на эллиптических кривых; даны основы криптоанализа современных шифров, рассмотрены основные криптографические протоколы, включая методы аутентификации и распределения ключевой информации с помощью симметричных и асимметричных криптосистем; приведены основные сведения по организационно-правовому обеспечению использования методов и средств криптографической защиты информации.

Криптографическая защита информации, как и любая наука, не останавливается в своём развитии. В разных областях криптографии постоянно ведутся научные исследования. Вопросами проверки стойкости алгоритмов и поиском их взлома занимаются ведущие мировые криптографы. Не прекращаются усилия по созданию новых методов для защиты информации.

Изучив данное учебное пособие, читатель ознакомится с базовыми понятиями криптографии, концепциями и принципами организации современных криптосистем, а также основными подходами к их использованию для решения практических задач информационной безопасности.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

Основная литература:

1. Авсентьев О. С., Солдатов В. В., Думачев В. Н., Зарубин С. В., Кучмасов Е. А. Криптографические методы защиты информации: учебное пособие. — Воронеж: ВИ МВД России, 2018. — 204 с.

2. Васильева И. Н. Криптографические протоколы: учебное пособие. — Санкт-Петербург: Изд-во СПбГЭУ, 2021. — 119 с.

Дополнительная литература:

1. Бабаш А. В., Шанкин Г. П. Криптография. — Москва: Солон-Р, 2002. — 512 с.

2. Васильева И. Н. Криптографические методы защиты информации: учебник и практикум для академического бакалавриата. — Москва: Юрайт, 2016. — 349 с.

3. Васильева И. Н., Куватов В. И. Криптографическая защита информации практикум. — Санкт-Петербург: Изд-во СПб ун-та МВД России, 2017. — 260 с.

4. Васильева И. Н., Куватов В. И., Потехин В. С. Криптографическая защита информации: учебное пособие. — Санкт-Петербург: Изд-во СПб ун-та МВД России, 2016. — 152 с.

5. Коржик В. И., Просихин В. П., Яковлев В. А. Основы криптографии: учебное пособие. — Санкт-Петербург: СПбГУТ, 2014. — 276 с.

6. Молдовян Н. А. Практикум по криптосистемам с открытым ключом. — Санкт-Петербург: БХВ-Петербург, 2007. — 304 с.

7. Панасенко С. П. Алгоритмы шифрования: специальный справочник. — Санкт-Петербург: БХВ-Петербург, 2009. — 564 с.

8. Рябко Б. Я., Фионов А. Н. Криптографические методы защиты информации. — Москва: Горячая линия — Телеком, 2012. — 230 с.

9. Шеннон К. Теория связи в секретных системах // Работы по теории информации и кибернетике / пер. с англ. — Москва: Изд-во иностранной литературы, 1963. — С. 333–369.

10. Шнайер Б. Прикладная криптография: Протоколы, алгоритмы, исходные тексты на языке Си. — Москва: Триумф, 2012. — 815 с.

Учебное издание

Васильева Ирина Николаевна,
кандидат физико-математических наук, доцент;
Локнов Алексей Игоревич,
кандидат технических наук;
Примакин Алексей Иванович,
доктор технических наук, профессор

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

Учебное пособие

Редактор *Корчуганова И. А.*
Компьютерная вёрстка *Фролова А. В.*
Дизайн обложки *Шеряй А. Н.*

ISBN 978-5-91837-672-0

Подписано в печать 18.01.2023. Формат 60×84 $^{1}/_{16}$
Печать цифровая. Объем 7,5 п. л. Тираж 100 экз. Заказ № 1/23

Отпечатано в Санкт-Петербургском университете МВД России
198206, Санкт-Петербург, ул. Летчика Пилютова, д. 1