Министерство образования и науки Российской Федерации
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

П.П. Бескид, Т.М. Татарникова

КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

Часть 1
ОСНОВЫ КРИПТОГРАФИИ

Учебное пособие

РГГМУ
Санкт-Петербург
2010

УДК 621.391.037.372

Бескид П.П., Татарникова Т.М. Криптографические методы защиты информации. Часть 1. Основы криптографии. Учебное пособие. СПб., изд. РГГМУ, 2010. – 95 с.

Рецензент: Молдовян Н.А., д-р техн. наук, проф., зав. лабораторией криптологии СПИИРАН.

В учебном пособии представлены основные разделы теоретической части дисциплины «Криптографические методы защиты информации»: описание основных типов шифров, их свойств и криптографических алгоритмов, некоторые методы криптоанализа простейших шифров, основы математического моделирования в криптографии.

Предназначено для подготовки специалистов по специальности 090106 – информационная безопасность телекоммуникационных систем.

Beskid P.P., Tatarnikova T.M. Cryptographic methods of information protection. Part 1. Basics of cryptography. A manual. St. Petersburg, RSHU Publishers, 2010. – 95 pp.

The manual presents the main sections of the theoretical part of the discipline "Cryptographic Methods of Information Protection": description of the main types of fonts and their properties and cryptographic algorithms, certain techniques of cryptanalysis of simple fonts, the foundations of mathematical modelling in cryptography.

Intended for training experts in Speciality 090106 'Information Security of Telecommunication Systems'.

ВВЕДЕНИЕ

В переводе с греческого *криптография* – это тайнопись, понимаемая теперь в широком смысле. Криптография использует преобразование одних знаков в другие, взятые из того же или другого алфавита, при этом применяются ключи, с помощью которых сообщение может быть обнаружено и прочитано, зашифровано и расшифровано.

Базовая модель криптографии, представленная на рис. 1, предполагает существование противника, имеющего доступ к открытому каналу связи и перехватывающего путем прослушивания все сообщения, передаваемые от отправителя к получателю. Прослушивание со стороны противника называется пассивным перехватом сообщений. Кроме того, противник может активно вмешиваться в процесс передачи информации – модифицировать передаваемые сообщения, навязывать получателю свои собственные сообщения из канала. Такие действия называются активным перехватом сообщений.

Рис. 1. Базовая модель криптографии

Способы противодействия перехватам основаны на применении криптографических методов. Классическая одноключевая (или симметричная) криптосистема представлена на рис. 2.

Прежде чем передать сообщение в открытый канал, отправитель с целью сокрытия истинного содержания подвергает исходную информацию специальному преобразованию. Для получения исходной информации на приемном конце необходимо выполнить обратное преобразование. Процедура прямого преобразования на передающем конце называется *шифрованием*, процедура обратно-

го преобразования на приемном конце называется *дешифрованием*. В одноключевой криптосистеме для выполнения процедуры шифрования и дешифрования необходимо знать общий для отправителя и получателя секретный компонент – секретный ключ. Информация на входе шифратора называется открытым текстом. Информация на входе дешифратора называется шифртекстом.

Все криптоалгоритмы с ключом делятся на *симметричные* и *асимметричные*. В симметричных криптоалгоритмах ключи, используемые на передающей и приемной сторонах, полностью идентичны. Такой ключ несет в себе всю информацию о засекреченном сообщении и поэтому не должен быть известен никому, кроме двух участвующих в разговоре сторон.

Противник, наблюдая шифротекст, не может прочитать открытый текст. Отсюда возникает задача для криптоаналитика: получение открытого текста по наблюдаемому шифротексту, а именно задача сводится к раскрытию секретного ключа шифрования/дешифрования. Действия, предпринимаемые криптоаналитиком с целью раскрытия секретного ключа, называются атакой.

Секретный канал, используемый для передачи секретного ключа от отправителя к получателю, должен исключать возможность утечки информации. Общая схема шифрования на секретном ключе приведена на рис. 2.

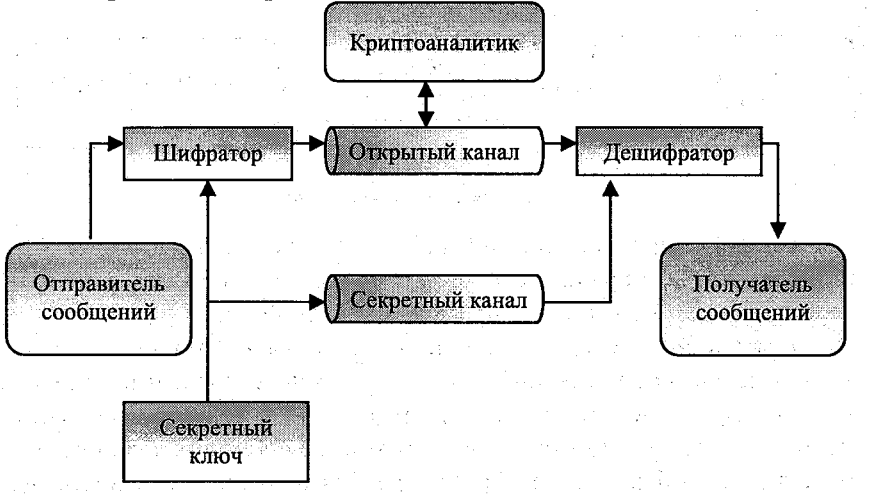


Рис. 2. Одноключевая криптосистема

Для решения задачи распределения ключей была выдвинута концепция двухключевой (или асимметричной) криптографии (рис. 3).

В такой схеме для шифрования и дешифрования применяются различные ключи. Для шифрования информации, предназначенной конкретному получателю, используют уникальный открытый ключ получателя-адресата.

Соответственно, для дешифрования получатель использует парный секретный ключ. Для передачи открытого ключа от получателя к отправителю секретный канал не нужен. Вместо секретного канала используется аутентичный канал, гарантирующий подлинность источника передаваемой информации (открытого ключа отправителя). Подчеркнем, что аутентичный канал является открытым и доступен криптоаналитику противника.

В настоящее время криптографические методы нашли широкое применение не только для защиты информации от несанкционированного доступа, но и в качестве основы многих новых электронных информационных технологий: электронного документооборота, электронных денег, тайного электронного голосования и др.

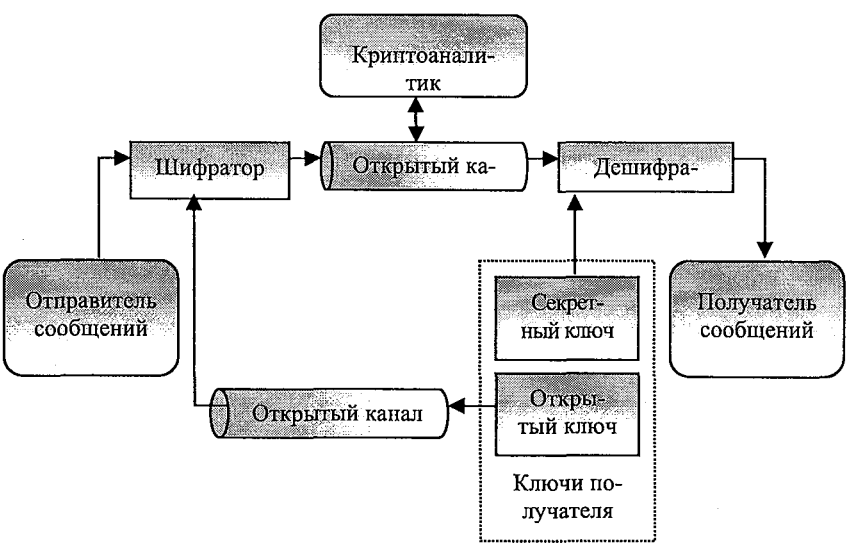


Рис. 3. Двухключевая криптосистема (шифрование/дешифрование)

Защита конфиденциальных сообщений в открытых сетях требует применения криптографических методов. Шифрование позволяет предотвратить доступ к информации со стороны третьих лиц в результате перехвата сообщения. Криптографические методы позволяют также устанавливать подлинность отправителя (аутентификация) и контролировать целостность (неизменность) передаваемых сообщений.

Асимметричные шифры не имеет смысла использовать для защищенного хранения документов. Их прерогатива – защита сообщений, электронной почты, прикрепленных к посланиям файлов и т.п.

1. КРАТКАЯ ИСТОРИЯ РАЗВИТИЯ КРИПТОГРАФИИ

История криптографии, как, впрочем, и нынешнее ее применение, неразрывно связана с военными и дипломатическими делами. Государственная и военная переписка возникли в глубокой древности и сопровождались изобретением различных криптографических методов для зашиты этой переписки от чтения противником.

Первым документально зафиксированным в письменности шифром является шифр Цезаря, хотя существуют свидетельства о наличии простейших устройств шифрования у древних греков в V–VI вв. до н.э.

Так, за 400 лет до н.э. в Спарте использовалось шифрование на круговом цилиндре. На его поверхность, как объясняет Плутарх, виток к витку, без промежутков наматывалась полоса папируса (скитала), по которой параллельно оси цилиндра записывался текст строка за строкой (рис. 4). В результате на развернутой полосе буквы располагались без видимого порядка. Для прочтения послания получатель должен был намотать скиталу на точно такой же цилиндр.

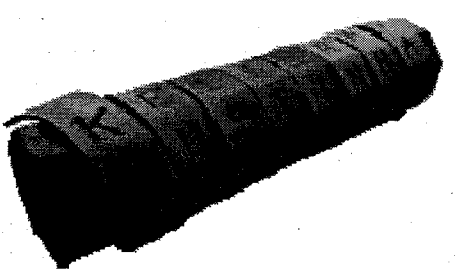


Рис. 4. Скитала

За 300 лет до н.э. в Греции был написан труд «Тактикус» о скрытых сообщениях. За 200 лет до н.э. изобретен полибианский квадрат, содержавший 5×5=25 клеток для двадцати четырех букв греческого алфавита и пробела, вписанных в произвольном порядке. При шифровании текста нужная буква отыскивалась в квадрате и заменялась на другую букву из того же столбца, но вписанную строкою ниже. Буква, которая находилась в нижней строке квадра-

та заменялась на букву из верхней строки того же столбца. Получатель, имевший точно такой же квадрат, производил расшифровку сообщения, выполняя указанные операции в обратном порядке.

История криптографии насчитывает около 4 тысяч лет. В качестве основного критерия периодизации криптографии возможно использовать технологические характеристики используемых методов шифрования.

Первый период (приблизительно с 3-го тысячелетия до н.э.) характеризуется господством моноалфавитных шифров (основной принцип – замена алфавита исходного текста другим алфавитом через замену букв другими буквами или символами).

Второй период (хронологические рамки – с IX в. на Ближнем Востоке и с XV в. в Европе и до начала XX в.) ознаменовался введением в обиход полиалфавитных шифров.

Третий период (с начала и до середины XX в.) характеризуется внедрением электромеханических устройств в работу шифровальщиков, при этом продолжалось использование полиалфавитных шифров.

Четвертый период – с середины до 70-х гг. XX в. – период перехода к математической криптографии. В работе Шеннона появляются строгие математические определения количества информации, передачи данных, энтропии, функций шифрования. Обязательным этапом создания шифра считается изучение его уязвимости к различным известным атакам: линейному и дифференциальному криптоанализу. Однако до 1975 г. криптография оставалась «классической», или же, более корректно, криптографией с секретным ключом.

Самым древним и самым простым из известных подстановочных шифров является шифр, использовавшийся Юлием Цезарем в переписке с Цицероном.

Метод Цезаря состоит в следующем. Сначала каждой букве алфавита сопоставляется ее порядковый номер l. Затем при шифровании записывается не сама буква, а та, чей номер больше на целое число K, называемое ключом (у Цезаря $K = 3$).

Для алфавита, содержащего m букв, правило шифрования задается соотношением, в котором использовано сложение по модулю m:

$$n = (K + l) \bmod m,$$

где n – номер буквы, полученной в результате шифрования; l – номер буквы исходного текста; m – длина алфавита.

Вычисления по модулю производятся над целыми числами. При этом числа заменяются остатками от их деления на модуль (например, 13 = 1 mod 12). Таким образом, зашифрованный номер буквы равен остатку от деления суммы (K+l) на m.

Обобщение шифра Цезаря называется шифром простой замены. Его суть заключается в том, что все буквы алфавита заменяются другими неповторяющимися буквами того же алфавита по правилу, которое является ключом, например, буква *а* заменяется на букву *в*, *б* – на *с*, *в* – на *а*,..., *я* – на *г*. Количество замен, возможных при таком шифровании, равно числу всех возможных перестановок из символов алфавита с объемом m. Оно составляет $m!$ При m = 32 общее число перестановок $m!$ = 32! = $2{,}63 \cdot 10^{35}$. Если в секунду при дешифровании методом простого перебора проверять миллион ключей, общее время на дешифрование без знания ключа составит $8{,}3 \cdot 10^{21}$ лет.

Несмотря на устрашающие числа, нетрудно указать способ криптоанализа, т.е. вскрытия этих шифров без знания ключа.

Первой уязвимостью одноалфавитных замен, замеченной нашими предками, была неизменность частот появления определенных букв в среднестатистическом тексте, т. е., если буква «О» является бесспорным лидером по частоте появления в русском языке, то с очень большой вероятностью буква, появляющаяся в зашифрованном тексте чаще всего, скажем «Ф», заменяет в нем «О», далее аналогично по убыванию частот. Зависимость частот появления распространенных букв можно довольно отчетливо проследить уже, скажем, в 100-буквенном послании. Редкие буквы обычно отслеживают подобным способом – их просто подбирают по смыслу в тех словах, где большая часть распространенных букв уже раскрыта. Данный способ получил название метод частотного анализа, позднее он применялся по той же самой схеме для пар рядом стоящих букв – биграмм и даже для буквенных троек – триграмм. Конечно, в этом случае для анализа требовался больший объем шифрованного текста, однако сам метод зачастую оказывался намного быстрее и эффективнее.

Вторая уязвимость схем, подобных шифру Цезаря, связана с короткими часто встречающимися словами: предлогами, союзами и местоимениями. Их роль в тексте часто довольно важна, чтобы допускать их пропуск, а вот человеку пытающемуся разгадать шифр, они дают очень многое. Например, не каждая буква алфавита может встречаться в тексте в полном одиночестве (союзы «и», «а», предлоги «в», «о» и другие). А чего стоит специфичный русский союз «или»!

Пусть зафиксировано большое число шифровок и известно, что открытые сообщения составлены на русском языке. В типичных текстах буквы русского алфавита, включая пробел, встречаются с разной частотой в указанном порядке: пробел, о, е, и, а, т, н, с, р, в, л, к, м, д, п, у, я, ь, з, б, ы, г, ч, й, х, ж, ю, д ц, э, щ, ф» ъ, как показано в табл. 1.

Таблица 1

Частоты появления букв в среднестатистическом тексте для русского языка

Буква	Частота, %	Буква	Частота, %	Буква	Частота, %
А	6,2	Л	3,5	Ц	0,4
Б	1,4	М	2,6	Ч	1,2
В	3,8	Н	5,3	Ш	0,6
Г	1,3	О	9,0	Щ	0,3
Д	2,5	П	2,3	Ъ	0,1
Е	7,2	Р	4,0	Ы	1,3
Ж	0,7	С	4,5	Ь	1,6
З	1,6	Т	5,3	Э	0,3
И	6,2	У	2,1	Ю	0,6
Й	1	Ф	0,2	Я	1,8
К	2,8	Х	0,9	пробел	17,5

Из табл. 1 можно видеть, что практически все буквы имеют разные частоты появления. Известны статистики биграмм, триграмм и других сочетаний букв для многих языков. Следовательно, анализируя шифртекст, полученный прямой заменой, по частоте использования символов и их сочетаний, можно определить, какую букву скрывает данный символ.

Отметим, что анализ и прочтение текстов, зашифрованных таким образом, встречаются в художественной литературе (например, Эдгар Алан По «Золотой жук», Артур Конан Дойль «Пляшущие человечки»).

Значительной модификацией одноалфавитной замены является шифр, именуемый квадратом Полибия или тюремной азбукой. В нем определенные символы алфавита заменялись уже парой чисел согласно какому-либо простому правилу. Так, исходный квадрат Полибия имел следующий вид, представленный на рис. 5.

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

Рис. 5. Квадрат Полибия

Вместо буквы в письме записывалась пара чисел: номер строки и номер столбца ячейки, в котором она располагалась. Так же, как и у шифра Цезаря, у данной схемы не счесть модификаций. В связи с тем, что в латинском алфавите 26 букв (а в современном русском – 33) и оба эти числа очень «неудачно» раскладываются на сомножители, модификации либо выбрасывали редко встречающиеся буквы, либо добавляли знаки препинания и пустые элементы, добиваясь прямоугольников и квадратов: 5×5, 5×6, 4×7, 6×6, 5×7.

Подобное представление двумя небольшими числами было очень удобно для передачи на большие расстояния визуально (костры, факелы, различные сигнальные флаги, солнечные зайчики) либо звуками (выстрелы, громкие удары, перестукивания).

В XVI в. Джероламо Кардано, итальянский математик, врач и философ, изобрел совершенно новый тип шифра, основанный на очень простой и в то же время надежной перестановке букв послания.

Для шифрования Кардано предложил использовать квадратные таблицы, где четверть ячеек прорезана так, что при четырех поворотах они покрывают весь квадрат. Вписывание в прорезанные ячейки текста и повороты решетки продолжаются до тех пор, пока весь квадрат не будет заполнен.

«Решетка Кардано» – прямоугольная карточка с отверстиями, чаще всего квадратная, которая при наложении на лист бумаги оставляет открытыми лишь некоторые его части. Карточка сделана так, что при ее последовательном использовании (поворачивании) каждая клетка лежащего под ней листа окажется занятой. Карточку сначала поворачивают вдоль вертикальной оси симметрии на 180°, а затем вдоль горизонтальной оси также на 180°.

И вновь повторяют ту же процедуру, как показано на рис. 6:

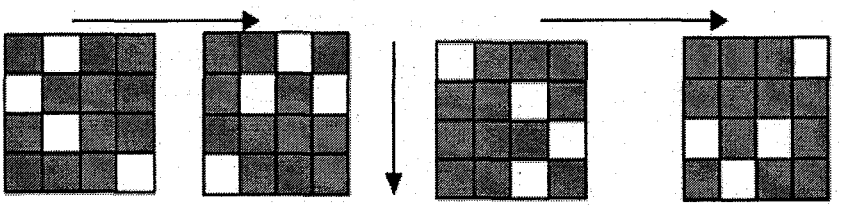


Рис. 6. Один из вариантов поворота «Решетки Кардано»

Также возможен второй вариант самосовмещений фигуры – последовательные повороты вокруг центра квадрата на 90° (рис. 7).

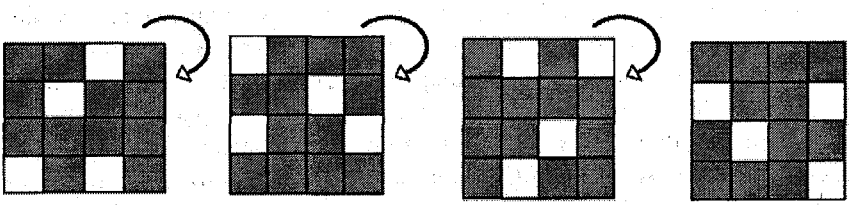


Рис. 7. Другой вариант поворота «Решетки Кардано»

Число подобных решеток быстро растет с их размером. Так, решетка 2×2 единственна, решеток 4×4 уже 256, а решеток размером 6×6 свыше ста тысяч. Несмотря на кажущуюся сложность, шифры типа решеток довольно просто вскрываются и не могут использоваться в виде самостоятельного шифра. Однако они очень удобны и еще долго использовались в практике для усиления шифров замены.

Подобным методом маскировки сообщения пользовались многие известные исторические лица, например, кардинал Ришелье во Франции и русский дипломат и писатель А. Грибоедов.

Примерно в одно время с Кардано французский дипломат Блез де Виженер предлагает модификацию шифра замены, получившего название таблица Виженера, основанное на использовании ключевого слова для шифрования.

Составим шифровальную таблицу для шифра Виженера, содержащую m+1 строку, где m – число букв в ключевом слове, и n столбцов, где n – число букв примерного алфавита. В первую строку таблицы впишем все буквы в алфавитном порядке. Свободные m ячеек первого (левого) столбца таблицы заполним сверху вниз буквами ключевого слова. Наконец, ячейки, оставшиеся пустыми в каждой строке заполним циклическими перестановками алфавита, каждая из которых начинается с соответствующей буквы ключевого слова.

Ниже, для примера, показана шифровальная таблица (табл. 2), образованная ключевым словом «имя».

Таблица 2

Шифровальная таблица Виженера

А	Б	В	Г	Д	Е	Ж	З	…	Ю	Я
и	й	к	л	м	н	о	ц	…	ж	з
м	н	о	п	р	с	т	у	…	к	л
я	а	б	в	г	д	е	ж	…	э	ю

Пусть требуется зашифровать слово «багаж». Подписываем под ним, повторяя необходимое число раз, ключевое слово:

Б	А	Г	А	Ж	
и	м	я	и	м	я

Далее пользуемся шифровальной таблицей. На пересечении столбца, начинающегося с буквы «б» сообщения, и строки, начинающейся с буквы «и» ключевого слова, находим букву «й» – криптограмму буквы «б». Полученное «й» подписываем под рассмотренной парой «б», «и». Под буквой «а» сообщения в строке, начинающейся с буквы «м» ключевого слова, находим криптограмму «м» для буквы «а» и т.д. Таким образом, окончательно получаем криптограмму всего сообщения: «ймвит».

Одним из наиболее известных шифров, базирующихся на методе многобуквенного шифрования, является шифр Плейфейера (1854 г.), в котором биграммы открытого текста рассматриваются

как самостоятельные единицы, преобразуемые в заданные биграммы шифрованного текста.

Алгоритм Плейфейера основан на использовании матрицы букв размерности 5×5, создан на основе некоторого ключевого слова. Рассмотрим пример, в котором ключевым словом является monarchy. Матрица представлена в табл. 3.

Таблица 3

Матрица шифрования алгоритмом Плейфейера

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

Матрица создается путем размещения букв, использованных в ключевом слове, слева-направо и сверху-вниз (повторяющиеся буквы в процессе формирования матрицы отбрасываются). Затем оставшиеся буквы алфавита размещаются в естественном порядке в оставшихся строках и столбцах матрицы. Буквы I и J считаются одной и той же буквой.

Открытый текст шифруется порциями по две буквы (биграммами) по следующим правилам:

1. Если оказывается, что повторяющиеся буквы открытого текста образуют одну пару для шифрования, то между этими буквами вставляется специальная буква-заполнитель, например, x. В частности, такое слово как, balloon разобьется на следующие биграммы:

ba lx lo on.

2. Если буквы открытого текста попадают в одну и ту же строку матрицы, каждая из них заменяется буквой, следующей за ней в той же строке справа, с условием, что для замены последнего элемента строки матрицы служит первый элемент той же строки.

3. Если буквы открытого текста попадают в один и тот же столбец матрицы, каждая из них заменяется буквой, стоящей в том же столбце сразу под ней, с тем условием, что для замены самого нижнего элемента столбца матрицы берется самый верхний элемент того же столбца.

4. Если не выполняется ни одно из перечисленных условий, каждая буква из пары букв открытого текста заменяется буквой, находящейся на пересечении, содержащей эту букву строки матрицы и столбца, в котором находится вторая буква открытого текста.

Долгое время шифры Виженера и их модификации считались невзламываемыми, пока прусским офицером Ф.В. Касисским не был найден (1863) простой способ поиска ключа для шифра Виженера.

Кроме непосредственного увеличения стойкости шифр Виженера привнес в криптографию две качественно новых идеи.

Во-первых, процесс шифрования стал зависеть в первую очередь от небольшого неизвестного третьей стороне слова – пароля. Конечно, сокрытие от злоумышленника всей таблицы значительно усложняет процедуру взлома шифра, но теперь случайное ее раскрытие не несет такого критического для всей системы значения, как в шифре Цезаря.

Во-вторых, сама таблица, использованная Виженером, несет по своей сути первые намеки на идею цифрового шифрования. При том, если все буквы латинского алфавита пронумеровать по порядку от 0 до 25, то процедура шифрования по такой таблице превратится в обычную операцию сложения. Например, буква исходного текста *A* (код 1) + буква пароля *B* (код 2) = код 3 – буква шифртекста *D*. При дешифровании, наоборот, из кода каждой зашифрованной буквы вычитается код буквы пароля. Если при сложении получается число, большее 25, то из него вычитается 26, если при вычитании получается отрицательное число – к нему прибавляется 26. Подобная схема сложения/вычитания называется в математике сложением по модулю – в данном случае сложением по модулю 26.

Дальнейшей модификацией данного способа, применяемой на практике, пожалуй, до наших дней стало шифрование по книге. В этом методе отправитель и получатель сообщения договариваются об одинаковой книге одного и то же издания. При шифровании отправитель выбирает в ней произвольное место и выписывает вместо повторяющегося пароля под исходным сообщением во всю его длину текст из книги, начиная с этого места. Само шифрование ведется по той же самой схеме, что и в схеме Виженера. Где-либо в письме или по другому каналу получателю передается номер

страницы и слова в книге, с которого началось шифрование – без этой информации дешифрование будет довольно проблематичным, даже если знать все о самом издании.

Революционным шагом в криптографии стало предложение сотрудниками телефонной компании AT&T Мейджором Джозефом Моборном и Гильбертом Вернамом системы симметричного шифрования.

Для получения шифртекста исходное сообщение объединяется операцией «исключающее ИЛИ» с ключом, называемым одноразовым блокнотом или шифроблокнотом. При этом ключ должен обладать тремя критически важными свойствами:

– быть истинно случайным;

– совпадать по размеру с заданным исходным сообщением (открытым текстом);

– применяться только один раз.

Шифр назван в честь телеграфиста Гильберта Вернама, который в 1917 г. построил телеграфный аппарат, выполняющий операцию шифрования телетайпных посланий автоматически.

Работа шифровального аппарата осуществляется по следующей схеме. Информация на телетайпной ленте представляла собой последовательность отверстий и непробитых участков, соответствующих нулям и единицам. Вернам предложил на передающей стороне разместить замкнутую в кольцо ленту с довольно длинной последовательностью подобных же нулей и единиц – она играла роль пароля. В момент передачи очередного импульса в канал связи лента-пароль сдвигалась на одну позицию, и с нее считывалось текущее двоичное значение, если был считан 0, то исходный двоичный сигнал не изменялся, если на ленте-пароле в этом месте оказывалась 1, то исходный сигнал инвертировался. На приемной стороне должна была присутствовать точно такая же лента и причем в том же начальном положении, как у отправителя. Процедура дешифрования точно соответствует процедуре шифрования.

Помимо того, что шифровальный аппарат Вернама был, по-видимому, первым полностью автоматическим электрическим шифратором, данная схема содержала две очень интересные идеи. Во-первых, в ней была применена операция сложения по модулю 2, ставшая базовой в цифровой криптографии. Заметим, что если

примененную Вернамом операцию изменения одного бита в замости от значения другого представить как сложение двух двоичных чисел с отбрасыванием старшего бита, то правило выполнится для всех четырех возможных комбинаций:

$$0 \oplus 0 = 0,$$
$$0 \oplus 1 = 1,$$
$$1 \oplus 0 = 1,$$
$$1 \oplus 1 = 0.$$

Это операция – исключающее ИЛИ – записывается двумя обозначениями: «⊕» и XOR – английским сокращением от исключающее ИЛИ (англ. eXclusive OR). В отношении сложения по модулю 2 интересен еще один факт – обратный к нему, т.е. восстанавливающей операцией является кроме вычитания по модулю 2 еще и сама эта операция, т. е. для любых X и Y выполняется

$$(X \oplus Y) \oplus Y = X$$

Это свойство нередко используется в различных современных шифрах и схемах.

Вторым интересным моментом оказалось то, что если ленту, склеенную в кольцо, сделать потенциально бесконечно длинной, а на самом деле равной длине сообщения, то данный шифр станет абсолютно стойким. Не имея информации о ленте-пароле, третья сторона не сможет сделать никаких предположений о содержании исходного сообщения. Ведь сам факт взлома шифра может быть установлен только в том случае, когда дешифровщик получит из перехваченного послания какое-либо осмысленное сообщение. А если пароль по длине равен самому сообщению, то существует огромное множество потенциальных паролей, которые при наложении на шифровку будут давать осмысленный текст – и все время разный.

Шифр Вернама (другое название: англ. One-time pad – схема одноразовых блокнотов) является единственной системой шифрования, для которой доказана абсолютная криптографическая стойкость.

Основные недостатки шифра Вернама заключаются в следующем:

Для работы шифра Вернама необходима истинно случайная последовательность нулей и единиц (ключ). По определению, последовательность, полученная с использованием любого алгоритма, является не истинно случайной, а псевдослучайной. То есть, нужно получить случайную последовательность неалгоритмически (например, используя распад ядер, создаваемый электронным генератором белый шум или другие случайные события).

Проблемой также является тайная передача последовательности и сохранение её в секрете. Если существует надёжно защищённый от перехвата канал передачи сообщений, шифры вообще не нужны: секретные сообщения можно передавать по этому каналу. Если же передавать ключ системы Вернама с помощью другого шифра (например, DES), то полученный шифр окажется защищённым ровно настолько, насколько защищён DES. При этом, поскольку длина ключа та же, что и длина сообщения, передать его не проще, чем сообщение. Шифроблокнот, сохраненный на физическом носителе, можно украсть или скопировать.

Возможны проблемы с надёжным уничтожением использованной страницы. Этому подвержены как бумажные страницы блокнота, так и современные электронные реализации с использованием компакт-дисков или флэш-памяти. Если третья сторона каким-нибудь образом узнает сообщение, она легко восстановит ключ и сможет подменить послание на другое такой же длины.

Шифр Вернама чувствителен к любому нарушению процедуры шифрования. Например, контрразведка США часто расшифровывала советские и немецкие послания из-за неточностей генератора случайных чисел (программный генератор псевдослучайных чисел у немцев и машинистка, бьющая по клавишам, в СССР).

Тем не менее, схема шифроблокнотов достаточно надёжна при ручной шифровке. Вышеперечисленные недостатки можно также устранить, если применить квантовую криптографию, в частности, протокол BB84 для генерации и передачи одноразовых блокнотов. В случае использования квантовой криптографии шифр Вернама также будет достаточно надёжным.

В настоящее время роль ленты пароля, естественно играет, уже двоичная информация на цифровом носителе, а само преобразование выполняется вычислительной техникой. Но схема однора-

зового блокнота все еще имеет очень широкое применение именно из-за своей 100%-й стойкости к взлому при перехвате шифровок третьей стороной. Интересно, что в американском кинофильме 2001 г. «Пароль «Рыба-меч» хакер, которого играет Хью Джекмен взламывает именно шифр Вернама.

В конце XIX в. голландец А. Кергофф в своей статье, посвященной применению шифров, упомянул, как одну из настоятельных рекомендаций, идею о том, что при раскрытии самого алгоритма преобразований третья сторона не должна ни на шаг приблизиться к сокрытому тексту. Это должно было достигаться базированием всего алгоритма шифрования на небольшом объеме секретной информации. Данная идея легла в основу всей современной криптографии и получила название принцип Кергоффа. В противовес тайнописи криптографией с ключом называют сегодня алгоритмы шифрования, в которых сам алгоритм преобразований широко известен и доступен для исследований каждому желающему, но шифрование производится на основе небольшого объема информации – ключа, известного только отправителю и получателю сообщения. В современной криптографии в зависимости от методик размер ключа составляет от 56 до 4096 бит. Естественно, любой криптоалгоритм с ключом можно превратить в тайнопись, просто «зашив» в исходном коде программы некоторый фиксированный ключ и считая закрытым весь алгоритм. А вот обратное преобразование – попытка выделить из тайнописи некий небольшой аргумент-параметр – удастся далеко не всегда.

В некоторых (наиболее простых) случаях ключ формирует человек, отправляющий сообщение, в отдельных – ключ создается автоматически с помощью программного обеспечения или даже запрашивается у удаленной базы данных ключей.

Все криптоалгоритмы с ключом делятся на симметричные и асимметричные. В симметричных криптоалгоритмах ключи, используемые на передающей и приемной сторонах, полностью идентичны. Такой ключ несет в себе всю информацию о засекречивании сообщения и поэтому не должен быть известен никому, кроме двух участвующих в разговоре сторон. Поэтому в отношении ключа симметричных систем часто применяется термин секретный ключ, а сами подобные системы называются шифрами на секретном ключе.

Симметричное шифрование можно применять как при отправке сообщений между двумя пользователями, разделенными большим расстоянием, так и при отправке "посланий" одним и тем же человеком самому себе, но во времени. Примером подобных отправлений является шифр файлов на жестких дисках и сменных носителях с тем, чтобы другие пользователи тех же ЭВМ не могли считать информацию в отсутствие владельца.

В асимметричном шифровании для шифрования сообщения применяется один ключ, а для дешифрования другой. Процедура дешифрования в симметричных системах устроена таким образом, что ни одно постороннее лицо не может, зная зашифрованный таким способом текст и ключ шифрования, восстановить исходный текст. Прочитать зашифрованный текст можно, только зная ключ дешифрования. Поэтому ключ шифрования может быть известен всем пользователям сети – его раскрытие не несет никакого урона переписке. Ключ шифрования в асимметричных системах называется открытым ключом.

Современный период развития криптографии (с конца 1970-х г. по настоящее время) отличается зарождением и развитием нового направления – криптографии с открытым ключом. Её появление знаменуется не только новыми техническими возможностями, но и сравнительно широким распространением криптографии для использования частными лицами (в предыдущие эпохи использование криптографии было исключительной прерогативой государства).

Криптографией занимались многие известные математики, такие как Франсуа Виет, Джероламо Кардано, Готфрид Вильгельм Лейбниц и, наконец, Френсис Бэкон. Именно он предложил двоичное кодирование латинского алфавита.

В России самостоятельная криптографическая служба была впервые организована Петром I, который под влиянием общения с Лейбницом учредил цифирную палату для развития и использования криптографии.

Промышленная революция в развитых странах привела к созданию шифровальных машин. В конце XVIII века Джефферсоном (будущим третьим президентом США) были изобретены шифрующие колеса. Первую практически работающую шифровальную машину предложил Вернам (1917). В том же году была изобретена роторная

шифровальная машина, выпускавшаяся с 1926 г. фирмой Сименс под названием «Энигма» (загадка), – основной противник криптоаналитиков союзных держав в годы второй мировой войны.

Владимир Александрович Котельников сформулировал (1941) четкое положение о том, каким требованиям должна удовлетворять недешифруемая система связи. Эта работа и «теорема отсчетов» (теорема Котельшкова, 1932) явились основополагающими в развитии отечественной криптографии. Созданные под руководством Котельникова закрытые системы с успехом использовались для связи Москвы с фронтами (1942 – 1945), в том числе во время принятия капитуляции Германии.

Неоценимый вклад в криптографию внес К. Шеннон, особенно своей работой «Теория связи секретных систем» (1948). Диффи У. и Хеллман М.Э. предложили (1976) принципиально новую криптосистему с открытым ключом. В США (1977) введен открытый Федеральный стандарт шифрования для несекретных сообщений (DES), который с 2001 г. заменен новым стандартом AES.

Вводится (1989) открытая отечественная система шифрования ГОСТ 28147-89.

Одновременно с совершенствованием искусства шифрования (рис. 8) шло развитие и криптоанализа, предметом которого является вскрытие криптограмм без знания ключа.

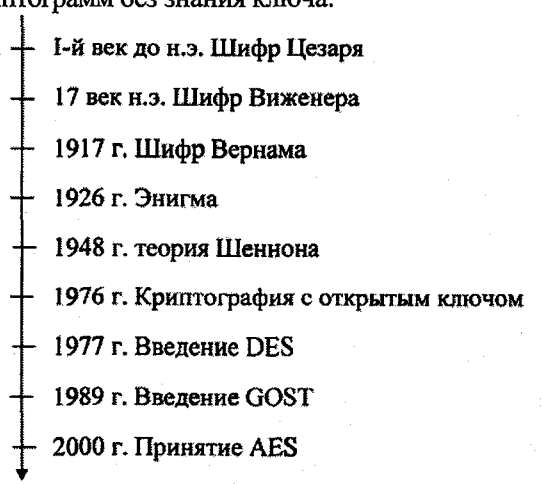


Рис. 8. Основные этапы развития криптографии

Хотя постоянное соревнование между шифрованием и криптоанализом продолжается и в настоящее время, однако имеется ряд существенных отличий современного этапа от предыдущих:

– широкое использование математических методов для доказательства стойкости шифров и для проведения криптоанализа;

– использование средств быстродействующей вычислительной техники;

– открытие нового вида криптографии с более «прозрачными» методам криптоанализа (криптография с общедоступным ключом);

– появление новых дополнительных функций обеспечения безопасности помимо шифрования;

– использование новейших физических методов в криптографии (динамический хаос, квантовая криптография, квантовый компьютер).

2. ИДЕИ И МЕТОДЫ КРИПТОГРАФИИ

2.1. Краткие сведения из теории чисел, применяемой в криптографии

Модульная арифметика

Действия в целочисленной модульной арифметике совершаются с целыми числами. Если с помощью $\mathrm{res}_n(x)$ обозначить остаток от деления x на n, то результаты сложения, вычитания и умножения по модулю n определяются соответственно равенствами:

$$(a+b) \bmod n = \mathrm{res}_n(a+b),\ (a-b) \bmod n = \mathrm{res}_n\,(a+(-b)),$$

$$-b \bmod n = \mathrm{res}_n(n-b),\ ab \bmod n = \mathrm{res}_n(b).$$

Легко проверить, что

$$(a+b) \bmod n = ((a \bmod n)+ (a \bmod n)) \bmod n,$$

$$ab \bmod n = ((a \bmod n)(b \bmod n)) \bmod n.$$

Для действий в модульной арифметике справедливы те же законы, что и в обычной:

- коммутативный (переместительный)

$$(a+b) = (b+a) \bmod n,\ ab = ba \bmod n,$$

- ассоциативный (сочетательный)

$$(a+(b+c)) = ((a+b)+c) \bmod n,$$

- дистрибутивный (распределительный)

$$a(b+c) = ab + ac \bmod n.$$

Функция Эйлера $\varphi(x)$ определяет число целых чисел из промежутка от 1 до $(x-1)$ взаимно простых с целым числом x.

Если p – простое число, то, очевидно, оно взаимно простое со всеми меньшими натуральными числами, поэтому для простого числа

$$\varphi(p) = p - 1.$$

Данная формула является частным случаем формулы для степени простого числа

$$\varphi(p^n) = p^n - p^{n-1}.$$

Для взаимно простых чисел x и y функция Эйлера мультипликативна:

$$\varphi(mn) = \varphi(m) \times \varphi(n).$$

В дальнейшем будет использована теорема Эйлера и ее частный случай малая теорема Ферма.

Теорема Эйлера. Если a и m взаимно простые целые числа, то $a^{\varphi(m)} = 1 \bmod m$.

Теорема Ферма. Если a и p взаимно простые целые числа, то $a^{p-1} = 1 \bmod p$, где p – простое число.

Действия целочисленной модульной арифметики должны выполняться абсолютно точно без округления. Если разрядность вычислительной машины не позволяет выполнить это условие, то используются специальные программы повышенной точности.

Действия сложения, вычитания и умножения являются *простым* в том смысле, что для их выполнения требуется число операций, которое описывается полиномиальной функцией от числа разрядов чисел, с которыми совершаются вычисления. *Сложные* операции – это такие, которые требуют для своего выполнения число операций, определяемое некоторой экспонентой. Показатель такой экспоненты зависит от числа разрядов чисел, участвующих в операции.

Рассмотрим далее некоторые операции и функции в поле целых чисел, где сложность операций определяется числом двоичных действий, выполняемых для получения результатов, и оценивается с помощью функции $O(n)$. Пусть даны две последовательности $f(n)$ и $g(n)$, тогда считают, что последовательность $f(n)$ растет как $O(g(n))$, если существует такая ненулевая константа C, что отношение $f(n)/g(n)$ стремится к C вместе с ростом n.

Возведение в степень

Прямой метод возведения в степень:

$$y = a^x \bmod n = \underbrace{a\ a\ a\ a\ \ldots\ a}_{\leftarrow x \text{ раз} \rightarrow} \bmod n,$$

где x – натуральное число, на первый взгляд, кажется сложной операцией, поскольку число перемножений пропорционально показателю степени x и экспоненциально зависит от числа разрядов x.

Однако существует *быстрый алгоритм возведения в степень*, число операций в котором пропорционально лишь числу разрядов показателя степени x.

Метод быстрого возведения в степень основан на представлении показателя степени x в двоичной форме:

$$x = x_0 + 2x_1 + \ldots + 2^{k-1} x_{k-1},$$

где x_i – элемент двоичного представления, т.е. $x_i = 0$ или 1; k – число двоичных разрядов. Тогда

$$y = a^x = a^{\sum_{j=0}^{k-1} 2^i xj} = \prod_{j=0}^{k-1} \left(a^{2^j}\right)^{xi} \bmod n,$$

и число перемножений последовательных квадратов числа a равно числу ненулевых двоичных разрядов x_i. С учетом возведения в квадрат общее число умножений при возведении в степень x не более $2 \times \log x$.

Вычисление дискретного логарифма

Логарифм x по основанию a от целого числа y по mod n,

$$\log_a y = x \bmod n,$$

является целым числом x, которое удовлетворяет уравнению

$$y = a^x \bmod n.$$

В отличие от логарифмов вещественных и комплексных чисел, которые всегда можно найти хотя бы приближенно, дискретные логарифмы существуют не всегда. Так, легко убедиться, что $\log_3 7 \bmod 17$ не существует.

Операция логарифмирования является сложной, т.е. до сих пор не найден алгоритм, который позволил бы вычислить логарифм за число операций, полиномиально зависящее от числа разрядов аргумента. В настоящее время известный наилучший алгоритм требует для вычисления логарифма по mod n количество операций порядка $\exp\sqrt{\ln(\ln(\ln n))}$.

Разложение на множители

Согласно основной теореме арифметики любое целое число n допускает однозначное разложение на простые множители:

$$n = p_1^{a_1} p_2^{a_2} \ldots p_s^{a_s},$$

где p_i – различные простые числа (т.е. числа, которые делятся без остатка только на 1 и на себя), a_i – натуральные числа, $i = 1, 2, \ldots, s$. Разложение чисел на простые множители более кратко называется *факторизацией.*

В настоящее время неизвестны простые алгоритмы факторизации. Наиболее быстрые алгоритмы требуют для факторизации целого числа n примерно столько же операций, сколько нужно для вычисления логарифма по mod n. Например, разложение на простые множители случайно выбранных натуральных чисел со ста десятичными разрядами по данным 1977 г. требовало 74 года машинного времени, а для чисел со 125 десятичными разрядами по данным 1994 г. та же оценка давала 10^6 лет.

Вычисление наибольшего общего делителя

Наибольшим общим делителем (НОД) двух чисел u и v называется такое наибольшее целое число НОД(u, v), которое делит оба числа u и v.

Прямой метод нахождения НОД состоит в разложении чисел u и v на простые множители и выборе среди них всех общих множителей. Эта задача является сложной, поскольку требует для своего вычисления числа операций, экспоненциально зависящего от числа разрядов чисел u и v. Однако существует быстрый метод вычисления НОД. Он основан на алгоритме Евклида и состоит в следующем.

Пусть, $u > v > 0$, тогда производятся вычисления до тех пор, пока не будет получено, что $b_k = 0$ или $b_k = 1$. Если $b_k = 1$, то НОД(u, v)=1, т.е. u и v взаимно простые числа. Иначе, когда $b_k = 0$, имеем НОД(u, v)= b_{k-1}, которое и дает решение задачи. Сложность данного алгоритма полиномиально зависит от числа разрядов u и v и требует O(log v) делений.

$$\begin{aligned}
U &= a_1 v + b_1 \quad b_1 = \mathrm{res}_v u \qquad 0 \le b_1 \le v \\
v &= a_2 b_1 + b_2 \quad b_2 = \mathrm{res} b_1 \, v \qquad 0 \le b_2 < b_1 \\
b_1 &= a_3 b_2 + b_3 \quad b_3 = \mathrm{res} b_2 \, b_1 \qquad 0 \le b_3 < b_2 \\
&\ldots \qquad\qquad \ldots \qquad\qquad \ldots \qquad\qquad \ldots \\
b_{k-2} &= a_k b_{k-1} + b_k \quad b_k = \mathrm{res} b_{k-1} \qquad 0 \le b_k < b_{k-1}
\end{aligned}$$

Обращение элементов по модулю

Элемент α, обратный к элементу a по модулю n

$$\alpha = a^{-1} \bmod n,$$

это такое целое число α, которое удовлетворяет уравнению

$$\alpha\, a = 1 \bmod n.$$

Для целых чисел a и n решение данной задачи (называемой решением сравнения) существует не всегда, а только при условии, что НОД чисел a и n равен единице. В частности, обратный элемент α существует всегда, когда a строго меньше n и само n – простое число.

В тех случаях, когда обратный элемент существует, его нахождение является простой задачей. Решается она с помощью алгоритма Евклида.

Действительно, если проанализировать алгоритм Евклида, можно видеть, что он дает возможность не только вычислить НОД чисел a и n, но и представить его в следующей форме:

$$\text{НОД}(a, n) = xa + yn,$$

где x и y – целые (не обязательно положительные) числа.

Поскольку предполагается, что обратный элемент существует, должно быть выполнено условие НОД(a,n)=1 или $xa+yn = 1 \bmod n$, но $yn = 0 \bmod n$, поскольку произведение yn кратно n .

В силу этого

$$a^{-1} = x \bmod n.$$

Тесты на простоту

Пусть задано число n и требуется проверить, является ли оно простым. При тривиальном методе решения его нужно делить на все меньшие, чем $\sqrt{n}$, простые числа и проверять, произошли ли эти деления нацело (без остатка). Задача факторизации, т.е. прямое разложение числа на простые множители, является сложной, и поэтому кажется, что сложна и задача проверки чисел на простоту.

Однако для проверки на простоту можно использовать вероятностный тест. Это тест, в котором может быть принято и ошибочное решение, но с весьма малой вероятностью.

Рассмотрим тест Ферма, в основу которого положена уже обсуждавшаяся малая теорема Ферма: если n простое число и b не кратно n, т.е. НОД $(b, n) = 1$, то

$$b^{n-1} = 1 \bmod n.$$

Для проверки простоты числа n генерируется k случайных целых чисел b_1, b_2,…,b_k , меньших, чем n, и k раз проверяется соотношение

$$\text{НОД}(b_i, n) = 1.$$

Если оно не выполняется хотя бы один раз, то n обязательно составное. Если же соотношение выполняется для всех k чисел, то далее производится проверка соотношений $b_i^{n-1} = 1 \bmod n$, $i = 1, 2,..., k$. Если хотя бы одно из этих соотношений не выполнится, то n необходимо составное.

Если n составное, то все тесты также могут выполниться, но это произойдет лишь с вероятностью $1/2^k$ (для так называемых составных чисел *Кармайкла* этот тест выполняется при любых b_i). Таким образом, увеличивая число испытаний, можно надежно протестировать любое число на простоту. Легко видеть, что вероятностный тест является простой операцией.

Помимо теста Ферма существуют и другие тесты на простоту. Наиболее эффективным из них считается тест Рабина-Миллера. Он тестирует составные числа, включая числа Кармайкла.

2.2. Модели шифрования/дешифрования дискретных сообщений

Наибольший интерес, как правило, вызывают вопросы, связанные с шифрованием и дешифрованием так называемых дискретных сообщений. Дискретные сообщения могут быть представлены сигналами с конечным числом состояний. К ним относятся данные, печатные тексты, а также речевые сигналы и изображения, если они предварительно преобразованы в дискретные (цифровые) сигналы. Для аналоговых сигналов используют другие методы защиты. Они будут рассмотрены отдельно.

Математической моделью системы шифрования/дешифрования дискретных сообщений называется пара функций

$$\overline{E} = f\left(\overline{M}, \overline{K}_{ш}\right), \quad \overline{M} = g\left(\overline{E}, \overline{K}_{д}\right),$$

которые преобразуют сообщение $\overline{M}$ в криптограмму $\overline{E}$ при помощи ключа шифрования $\overline{K}_{ш}$ и, наоборот, криптограмму $\overline{E}$ в сообщение $\overline{M}$ при помощи ключа дешифрования $\overline{K}_{д}$.

Обе функции, задающие криптосистему, должны удовлетворять следующим требованиям:

– $\overline{E} = f\left(\overline{M}, \overline{K}_{ш}\right)$ и $\overline{M} = g\left(\overline{E}, \overline{K}_{д}\right)$ при известных аргументах вычисляются сравнительно просто;

– $g(\overline{E}, ?)$ при неизвестном ключе дешифрования $\overline{K}_{д}$ вычисляется достаточно сложно.

Голландский криптограф А. Кергофф (1835–1903) предположил, что секретность шифров должна основываться только на секретности ключа, который может быть легко заменен, а не на секретности алгоритма шифрования, поскольку он, в конце концов, может оказаться известным противнику. Достоинство такого подхода несомненно: открытое опубликование алгоритма шифрования позволяет многим специалистам принять участие в анализе и обсуждении его свойств, стойкости и возможности технического применения. При рассмотрении современных систем шифрования предполагается, что ключ дешифрования $\overline{K}_{д}$ неизвестен нелегальным пользователям (посторонним наблюдателям, оппонентам), хотя они и могут знать функции $f\left(\overline{M}, \overline{K}_{ш}\right)$ и $g\left(\overline{E}, \overline{K}_{д}\right)$, а также ключ шифрования $\overline{K}_{ш}$, если он не совпадает с ключом дешифрования $\overline{K}_{д}$.

Следует различать три основных вида нападений (атак) оппонентов на криптосистему с целью узнать сообщение (в обобщенной постановке задачи, раскрыть ключ шифрования/дешифрования):

– атака при известной криптограмме $\overline{E}$;

– атака при известной криптограмме $\overline{E}$ и известном сообщении $\overline{M}$, которое соответствует определенной части криптограммы, полученной при использовании того же самого ключа (атака при частично известном открытом сообщении);

– атака при известной криптограмме и специально выбранной части сообщения, соответствующей части криптограммы, полученной на том же ключе (атака с частично выбранным открытым сообщением).

Современные криптосистемы считаются стойкими, если они стойки относительно атак всех трех видов.

Для криптосистем, которые шифруют сообщения с невысокими требованиями к вероятности ошибки при передаче (цифровая речь, цифровое и изображение) необходимо добавить четвертое, дополнительное, требование. Оно стоит в том, что дешифрование после передачи криптограммы по каналам с помехами не должно увеличивать число ошибок по сравнению с тем числом ошибок, которые образовались в канале связи из-за помех, иными словами не должно происходить размножение ошибок после дешифрования.

Если ключ шифрования равен ключу дешифрования, т.е.

$$\overline{K}_{ш} = \overline{K}_{д} = \overline{K} \ ,$$

то система называется *симметричной (одноключевой)*. Для ее работы в пункты шифрования и дешифрования должны быть секретным образом доставлены одинаковые ключи $\overline{K}$.

Если $\overline{K}_{ш} \neq \overline{K}_{д}$, т.е. ключ шифрования не равен ключу дешифрования, то система шифрования называется *несимметричной (двухключевой)*. В этом случае ключ $\overline{K}_{ш}$ доставляется в пункт шифрования, а ключ $\overline{K}_{д}$ – в пункт дешифрования. Оба ключа должны быть связаны функциональной зависимостью $\overline{K}_{д} = \varphi(\overline{K}_{ш})$, но такой, что по известному ключу шифрования $\overline{K}_{ш}$ нельзя было бы восстановить ключ дешифрования $\overline{K}_{д}$. Это означает, что для несимметричной системы шифрования функция φ() должна быть трудно вычисляемой.

В несимметричной системе шифрования имеется возможность распределять среди законных пользователей секретным образом только их личные ключи дешифрования, а ключи шифрования сделать открытыми и опубликовать, например, в общедоступном справочнике. Поэтому такие криптосистемы, предложенные

Диффи (W. Diffie) и Хелманом (V. Hellman), называют системами с общедоступным ключом (Public key cryptosystem).

В первой части дисциплины будут рассмотрены только одноключевые (симметричные) системы.

2.3. Понятие стойкость криптосистемы

Существуют два основных класса стойкости криптосистем.

Идеально (*безусловно*) *стойкие*, или *совершенные* криптосистемы. Для этих систем стойкость к криптоанализу (дешифрованию без знания ключа) не зависит от вычислительной мощности оппонента. Их называют *теоретически недешифруемыми* системами.

Вычислительно стойкие криптосистемы. У них стойкость к криптоанализу зависит от вычислительной мощности оппонента.

Система является теоретически недешифруемой, если никакая криптограмма $\overline{E}$ при условии, что ключ $\overline{K}$ неизвестен, не раскрывает никаких сведений о сообщении $\overline{M}$, зашифрованном в эту криптограмму. В соответствии с теорией информации это происходит при условии, что равна нулю взаимная информация между множеством сообщений и множеством криптограмм. Из этого следует, что при неизвестном ключе дешифрования вероятность угадывания переданного сообщения не зависит от того, используется криптограмма или нет.

Оппоненты имеют доступ к открытому каналу связи и перехватывают, подслушивая криптограммы сообщений, которые пересылаются от отправителя к получателю. Но при идеальном шифровании, если оппоненты ничего не знают о ключе, криптосистема «обрывает» канал передачи от информации, которой обмениваются легальные пользователи, к оппонентам.

Равносильное определение идеального шифрования устанавливает независимость элементов любой пары $\overline{M}$ и $\overline{E}$ из множества сообщений и множества криптограмм. Независимость означает, что перехват криптограммы не влияет на возможность расшифровки сообщения.

Можно показать, что если двоичные элементы ключа выбираются *взаимно независимыми и равновероятными,* то этого *доста-*

точно, чтобы система шифрования оказалась теоретически недешифруемой.

Рассмотрим пример построения теоретически недешифруемой системы (рис. 9).

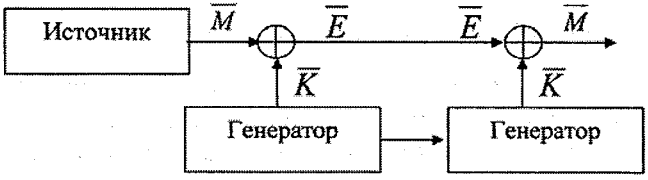


Рис. 9. Система передачи с шифрованием двоичных сообщений

Предположим, что сообщение является двоичной последовательностью $\overline{M}^n$ длины n. Тогда можно сформировать криптограмму как двоичную последовательность такой же длины по следующему правилу:

$$\overline{E}^n = \overline{M}^n \oplus \overline{K}^n,$$

используя поразрядное сложение $\oplus$ по модулю 2 сообщения $\overline{M}^n$ с ключом $\overline{K}^n$. Здесь ключ также является двоичной последовательностью длины n и играет роль маскирующего слагаемого, например, при $n = 21$ получаем

$$\begin{array}{ll} \overline{M} & 011001101111010001110 \\ \oplus & \oplus \\ \overline{K} & 010011110101100110101 \\ \hline \overline{E} & 001010011010110111011 \end{array}$$

При известном ключе $\overline{K}$, который должен быть передан на приемную сторону каким-либо секретным образом, сообщение легко восстанавливается по той же формуле, по которой производилось шифрование:

$$\overline{M}^n = \overline{E}^n \oplus \overline{K}^n,$$

поскольку $\overline{K}^n \oplus \overline{K}^n = 0$.

3. СПОСОБЫ ФОРМИРОВАНИЯ КРИПТОГРАММ

По способам формирования криптограммы из сообщения различают:

- блоковое шифрование,
- потоковое шифрование.

Для образования блокового шифра последовательность $\overline{M}$, состоящая из символов сообщения, разбивается на блоки $\overline{M}_1, \overline{M}_2, \ldots, \overline{M}_i \ldots$ одинаковой длины n. Если число символов последовательности $\overline{M}$ не кратно n, она дополняется необходимым числом нулей. После разбиения каждый такой блок преобразуется в блок криптограммы по одному и тому же правилу, зависящему от ключа шифрования $\overline{K}_ш$ (рис.10). Блоки криптограммы $\overline{E}_1, \overline{E}_2, \ldots, \overline{E}_i, \ldots$ обычно имеют ту же длину n, что и блоки сообщения.

При сохранении неизменным ключа шифрования $\overline{K}_ш$ одинаковые блоки, появляющиеся в разных местах сообщения, дают и в криптограмме одинаковые блоки. Обычно такой способ блокового шифрования называется *шифрованием с помощью кодовой книги.*

В случае потокового шифрования каждый отдельный символ сообщения преобразуется в отдельный символ криптограммы независимо от других символов по правилу, заданному ключом (рис. 11). Правило может изменяться от символа к символу.

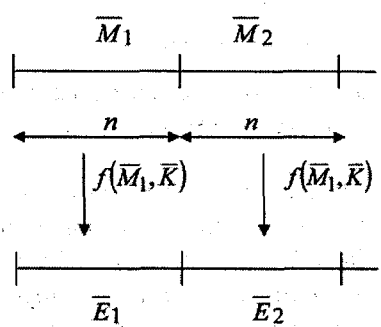


Рис. 10. Блоковое шифрование

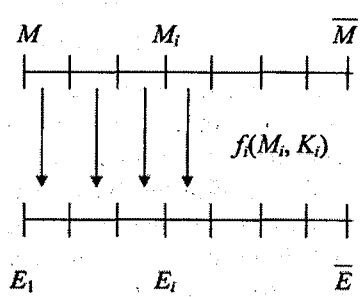


Рис. 11. Потоковое шифрование

Наиболее часто используемыми потоковыми шифрами являются так называемые аддитивные потоковые шифры, или шифры гаммирования. В случае гаммирования сообщение представляется последовательностью двоичных символов, а каждый символ криптограммы формируется сложением по модулю 2 соответствующих символов сообщения и символов специально формируемой двоичной последовательности (гаммы), зависящей от ключа:

$$E_i = M_i \oplus \Gamma_i(K),$$

где $\Gamma_i(K)$ – символ гаммы, функционально зависящий от ключа K и номера i символа сообщения.

По аналогичному правилу происходит и восстановление сообщения

$$M_i = E_i \oplus \Gamma_i(K),$$

где символ $\Gamma_i(K)$ совпадает с символом гаммы в предыдущем выражении, поскольку оба символа формируются с помощью одного и того же ключа.

Из этого следует, что ключ должен быть секретным образом доставлен на приемную сторону (рис. 12).

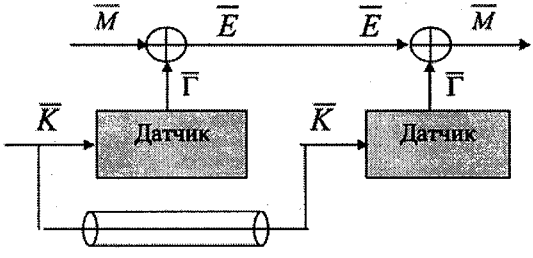


Рис. 12. Потоковое шифрование гаммированием

Различные варианты потоковых шифров определяются различными вариантами датчиков гаммы, некоторые из них могут быть основаны на блоковом шифре.

Таким образом, блоковые и потоковые шифры в действительности могут реализовываться одним и тем же устройством или программой, но на основе разных модификаций (мод) методов шифрования. Другой известный способ построения датчиков гаммы использует, так называемые, линейные рекуррентные регистры. Существуют и иные способы построения датчиков гаммы.

3.1. Блоковое шифрование

3.1.1. Симметричные блоковые шифры

Отличительной чертой блочных шифров является обработка исходного текста по несколько (десятки, сотни) бит, т.е. поблочно. Блочные шифры являются логичным продолжением идеи алфавитных замен, перенесенной в новый цифровой век. Очевидно, что при выполнении алфавитных подстановок двухбуквенных пар (биграмм) на другие двухбуквенные пары, частотный анализ становится уже достаточно трудоемкой задачей. А в этом случае, когда по какому-либо закону изменяются наборы из 4, 8 или даже 16 символов, ни о каком частотном вскрытии шифров речи идти не может – объем шифртекста, необходимого для подобной атаки растет экспоненциально. Проблема, из-за которой подобные схемы не были востребованы ранее, заключалась в отсутствии надежного и простого в употреблении алгоритма подобных преобразований, да еще в идеале и зависящего только от ключа шифрования.

Любой блочный шифр в общем виде представляет собой закон отображения множества входных блоков исходного текста на множество блоков зашифрованного текста (рис. 13). Кроме того, этот закон очень сильно зависит от секретного ключа шифрования.

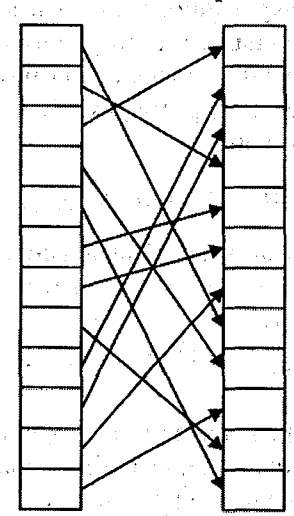


Рис. 13. Общий вид образования блочного шифра

В принципе блочный шифр как закон отображения - это шифр замены над очень большим алфавитом (из десятков и сотен бит). Потенциально любой шифр замены можно описать полной таблицей соответствий между входными и выходными данными. Так делали и для побуквенных замен. Но для того чтобы описать только одно отображение над 64-битным блоком (т.е. алфавитом из 2^{64} элементов) требуется 64×2^{64} бит запоминающего устройства, а ведь это отображение будет применено только при каком-то одном конкретном значении ключа. Поэтому современный блочный шифр – это закон, описывающий данное отображение алгоритмически.

Количество бит в обрабатывающем блоке называется разрядностью блока, количество бит в ключе – размером ключа алгоритма. Наиболее популярная на сегодняшний день разрядность блока – 64 и 128 бит, а размер ключа (128, 192 или 256). В общем виде процесс блочного шифрования описывается формулой:

$$\overline{E} = \text{EnCrypt}\,(\overline{M}, \overline{K}_{\text{ш}}),$$

где $\overline{M}$ – блок исходного текста, $\overline{E}$ – зашифрованный блок, $\overline{K}_{\text{ш}}$ – ключ шифрования.

Разрядность зашифрованного блока в классических блочных шифрах в точности совпадает с разрядностью исходного блока, поэтому за этап собственно шифрования поток данных не уменьшается и не увеличивается в размерах. Дешифрование описывается соответственно формулой

$$\overline{M} = \text{DeCrypt}(\overline{E}, \overline{K}_{\text{ш}}).$$

В тех случаях, когда и для шифрования и для дешифрования используется одна и та же последовательность действий, т.е. $\text{EnCrypt} \equiv \text{DeCrypt} \equiv \text{Crypt}$ и, соответственно, $\text{Crypt}(\text{Crypt}(\overline{M}, \overline{K}_{\text{ш}}), \overline{K}_{\text{ш}}) = \overline{M}$ блочный шифр называется абсолютно симметричным.

Основной идеей практически всех блочных шифров является тот факт, что последовательность бит любой длины Q можно представить в виде натурального числа из диапазона $[0...2^{Q}-1]$. Более того, один и тот же объем данных, разбивая на подобные

блоки разными способами, можно представить как в виде большого набора небольших чисел, например, от 0 до 15 или от 0 до 255, так и в виде нескольких огромных чисел, например, диапазона [0...4294967295]. Здесь все зависит от схемы разбиения потока данных на блоки бит. Над этими натуральными числами производится множество простейших математических и алгоритмических преобразований, так называемых криптопримитивов – из числа тех, которые легко выполняются на современной архитектуре ЭВМ.

Для криптопреобразований нет никакой разницы, представляет ли обрабатываемая группа бит (число) какой-либо определенный смысл или нет. Например, на каком-то промежуточном этапе в обрабатываемом 8-битном блоке первые два бита могут быть позаимствованы из третьей буквы шифруемого текста, следующие три – из шестой, а последние три бита, вычислены на основе первой и пятой букв. Главное, чтобы после восстановления на приемной стороне все биты в блоке заняли свои правильные места и получили правильные значения, т.е. снова превратились в осмысленный текст. В подобном абстрагировании от смысловой нагрузки на промежуточных стадиях и заключена сила двоичного представления данных.

Несомненно, что для однозначного дешифрования текста на приемной стороне все применяемые преобразования должны быть обратимы. Ни на одном из этапов шифрования из обрабатываемой последовательности не должно происходить "потери" информации. Перечень наиболее часто применяемых преобразований с их условными обозначениями на схемах приведен в табл. 4.

Отдельное место в криптографических преобразованиях занимают табличные подстановки. С помощью табличных подстановок реализуют наиболее общие законы изменения данных, зачастую не описываемые ни аналитически, ни алгоритмически. В тех случаях, когда в шифрующем алгоритме необходимо реализовать довольно сложную для микропроцессора или вообще не описываемую алгоритмом функцию, наиболее подходящим решением является занести все ее значения в массив, хранящийся в постоянной или оперативной памяти. Подобное задание функции называется табличным.

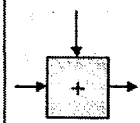
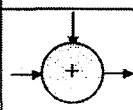
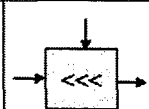
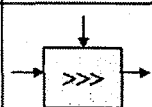
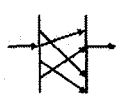
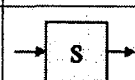
Количество бит в исходном операнде называется разрядностью входа табличной подстановки, количество бит в результате – разрядностью выхода. В большинстве случаев табличные подстановки используются в качестве биективной (однозначно обратимой) функции – в этом случае разрядность входа и выхода совпадает. Наиболее часто по такой схеме применяются подстановки 4×4 бита и 8×8 бит. Обратное преобразование, если в нем есть необходимость, задается на дешифрующей стороне также таблицей и, соответственно, такой же размерности. В тех случаях, когда от подстановки не требуется обратимость, наряду с традиционными, применяются подстановки 4×8, 8×32 бита и некоторые другие.

Запись функции в таблице имеет свои достоинства и недостатки. По сравнению с долгим вычислением значения функции выборка из таблицы производится гораздо быстрее (за две операции процессора). Однако эти две операции не сравнимы по быстродействию с обычным сложением или XOR – ведь одним из операндов в выборке всегда является ячейка памяти и предсказать заранее, какая именно ячейка будет востребована, невозможно. Поэтому на современных процессорах табличная выборка производится довольно быстро только тогда, когда вся таблица размещается в кэш-памяти процессора. В противном случае каждая операция выборки инициирует многотактовый процесс обращения к настоящей оперативной памяти. Все это накладывает жесткие ограничения на размер хранимой таблицы и, соответственно, разрядность ее входа и выхода.

Одним из неоспоримых преимуществ табличных подстановок является быстродействие в отношении нелинейных операций. Практически все приведенные в табл. 4 операции, кроме умножения по модулю 2^N+1 и табличных подстановок, обладают очень высокой линейной зависимостью между входными и выходными значениями. Это очень плохо в отношении криптостойкости шифра к линейному криптоанализу. В плане повышения стойкости шифра к этому классу атак табличная подстановка намного опережает операцию умножения по модулю 2^N+1 и является на сегодняшний день самым применяемым нелинейным преобразованием.

Таблица 4

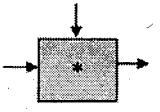
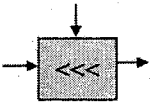
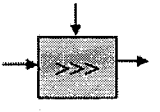
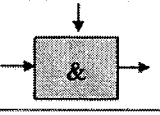
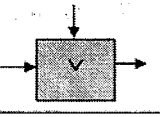
Основные обратимые криптопримитивы

Операция	Условные обозначения	Математическая нотация	Примечание
Сложение		$\overline{E} = \overline{M} + V$	При переполнении результатом разрядности операндов бит переноса отбрасывается – потери информации при этом не происходит
Исключающее «ИЛИ» (XOR)		$\overline{E} = \overline{M} \oplus V$ или $\overline{E} = \overline{M}$ XOR V	Операция обратима сама себе
Умножение по модулю 2^N+1		$\overline{A} = (\overline{M} * V) \bmod (2^N+1)$	Данное преобразование обратимо. Сомножитель по известному произведению и другому сомножителю находится по алгоритму Евклида.
Циклический битовый сдвиг влево		$\overline{E} = \overline{M}$ ROL V	Операция является обратной к "циклическому сдвигу вправо»
Циклический битовый сдвиг вправо		$\overline{E} = \overline{M}$ ROR V	Операция является обратной к "циклическому сдвигу влево»
Произвольная перестановка бит			Данная операция рассчитана в основном на аппаратную реализацию и поэтому сейчас в значительной степени устарела
Таблица подстановки		$\overline{E}$ = S–Box[$\overline{M}$]	

В тех случаях когда от преобразования не требуется обратимости по входным параметрам, дополнительно к описанным обратимым функциям применяются операции, приведенные в табл. 5.

Таблица 5

Основные необратимые криптопримитивы

Операция	Условные обозначения	Математическая нотация
Умножение		$\overline{E} = \overline{M} * V \bmod (2^N)$
Остаток (модуль) от деления		$\overline{E} = \overline{M} \bmod V$
Арифметический битовый сдвиг влево		$\overline{E} = \overline{M}$ SHL V
Арифметический битовый сдвиг вправо		$\overline{E} = \overline{M}$ SHR V
Логическое «И»		$\overline{E} = \overline{M}$ AND V
Логическое «ИЛИ»		$\overline{E} = \overline{M}$ OR V

В качестве параметра V в обратимых и необратимых операциях могут использоваться как фиксированные числовые константы, так и материал ключа. Материалом ключа называется блок данных, вычисляемый как основание ключа шифрования и зависящий только от него $V=f(\overline{K}_{ш})$. В простейшем случае материалом ключа может выступать сам ключ, (если его разрядность совпадает с разрядностью операции) либо произвольная его часть без какой-либо обработки. Операции, в которых в качестве параметра V использу-

ется материал ключа, называются перемешиванием с ключом или добавлением ключа (никакой связи с арифметическим сложением данный термин не несет). Подобных операций за время шифрования одного блока производится достаточно много. Таким образом, решается одно из главных требований к шифрам – невосстановимость ключа.

Еще в работе Шеннона «Теория связи секретных систем» были сформулированы следующие основные принципы преобразования сообщений, которые могут обеспечить высокую стойкость блокового шифрования (рис. 14).

1. Подстановка (нелинейное преобразование, зависящее от ключа), ставит на место одних символов другие как функцию символов сообщения и ключа, создает сложную зависимость между сообщением и шифртекстом;

2. Перестановка выполняет перепутывание, перемешивание символов, скрывает частоту появления устойчивых сочетаний символов;

3. Итерирование (повторение двух предыдущих пунктов многократно) распространяет влияние отдельных символов открытого теста и отдельных символов ключа на значительное число символов шифртекста.

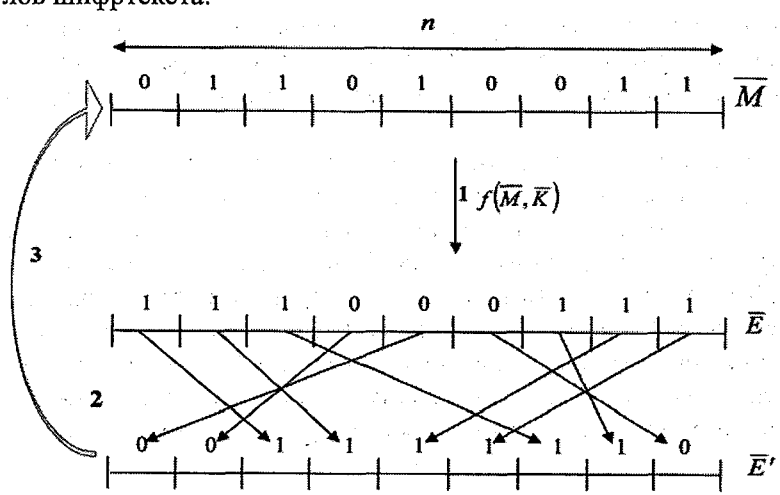


Рис. 14. Принцип построения блоковых шифров

Достаточно сложно задать общее нелинейное преобразование на блоках большой длины n. В то же время если выбирать блоки короткими, то в криптограмме сохранится статистика сообщения. Статистику сообщения оппонент может использовать для эффективного криптоанализа, как это реализуется, например, при криптоанализе шифра простой замены. Для разрешения указанного противоречия обычно выбирается умеренная длина блоков. Сообщение разбивается на блоки одинаковой длины n. Затем к каждому блоку многократно применяются нелинейное преобразование и перестановка символов в пределах блока.

Простейшим способом задания нелинейного преобразования является табличный способ, когда сначала двоичный блок преобразуется в число, затем это число по таблице преобразуется в другое число, которое, в свою очередь, снова преобразуется в двоичный блок. Если шифруются блоки длиной $n = 3$, им сопоставляются некоторые числа:

$$000 \rightarrow 0,\ 001 \rightarrow 1,\ 010 \rightarrow 2,\ldots,\ 111 \rightarrow 7,$$

а таблица подстановок может быть задана ключом так:

$$\{0,1,2,3,4,5,6,7\} \rightarrow \{4,2,6,7,0,5,1,3\}\ .$$

Это означает, что, например, входной блок 011 будет преобразован во входной блок 111. Сложность такого преобразования равна 2^n, т.е. числу возможных двоичных слов длиной n, где n – длина блока.

Нелинейные преобразования определяются секретным ключом. При выполнении каждой итерации обычно используется новый ключ. Для его получения производят специальные преобразования секретного ключа. Совокупность применяемых ключей (текущих ключевых элементов) называют расширенным ключом. Расширенный ключ вырабатывается из секретного ключа, например, при помощи циклических сдвигов символов исходного ключа. Определенные последовательности бит расширенного ключа применяют как текущие ключи.

3.1.2 Схемы образования блочных шифров с помощью сетей

Любой блочный шифр является многократной комбинацией описанных операций по определенной схеме или без таковой.

Одной из наиболее распространенных схем блочных шифров являются сети. Блочный шифр, построенный по такой схеме, состоит из многократных повторений, называемых циклами, или раундами, нескольких видов операций, называемых слоями. Разбиение всего процесса шифрования на несколько однотипных слоев позволяет:

– сократить размер программного кода использованием цикла;

– унифицировать «формулу» шифрования и, как следствие, упростить проверку стойкости шифра к известным видам криптоатак;

– сделать шифр легко усложняемым при необходимости (путем увеличения числа раундов);

– ввести понятие ключей раунда, на которые разбивается весь материал ключа.

Некоторые из слоев не содержат в себе процедур смешивания с ключом. Обычно они выполняют функции обратимого перемешивания данных внутри блока либо защищают блочный шифр от какого-либо конкретного класса криптоатак. Подобные слои выполняют свои преобразования всегда по одной и той же схеме. В тех слоях, где в качестве параметров операций используется материал ключа, на каждом раунде используются обычно все новые и новые блоки из материала ключа – ключи раунда. Поэтому каждый подобный слой на самом деле производит свое уникальное преобразование материала. Эта схема позволяет наряду с унифицированностью преобразований получить очень сложную для дешифрования при неизвестном ключе схему.

Одной из самых простых и ранних сетей является SP-сеть, каждый раунд которой состоит из двух слоев (рис. 15).

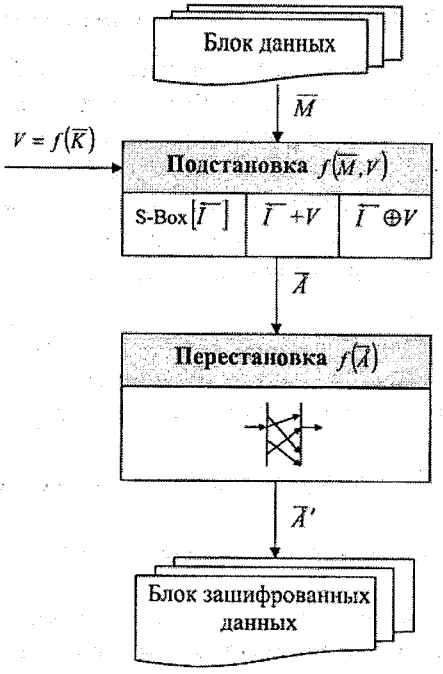


Рис. 15. Структура SP-сети

Свое название данная сеть получила от сокращений двух английских слов: подстановка (Substitution – S) и перестановка (permutation – P). В слое подстановки над данными производятся преобразования класса сложения, исключающего ИЛИ, табличных подстановок, в качестве параметров используются константы и материал ключа. В слое перестановки биты либо реже байты меняются местами внутри блока обычно по фиксированной (не зависящей от ключа) схеме.

Более современной модификацией технологии сетей является KASLT-сеть с тремя слоями в каждом раунде (рис. 16).

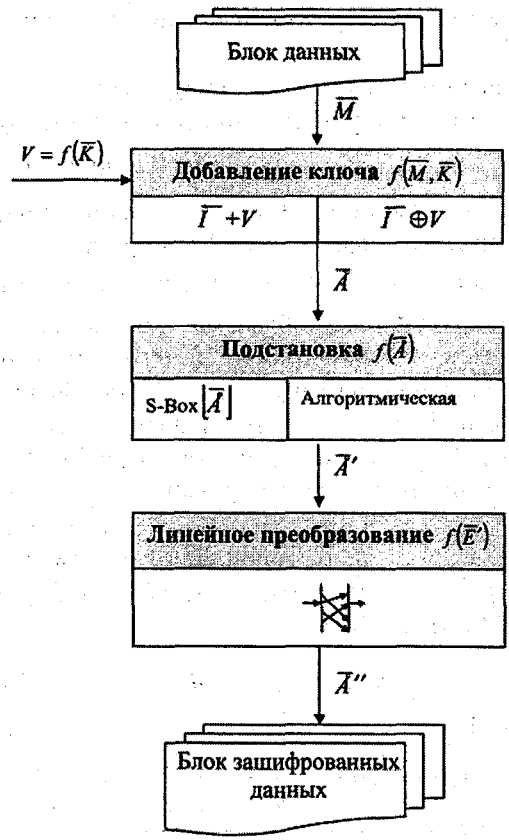


Рис. 16. Структура KASLT-сети

Первый слой KASLT-сети – добавление ключа (Key Addition – KA) – производит смешивание данных с ключом раунда. Смешивание производится обычно какой-либо простой операцией – сложением либо XOR. Второй слой – подстановка (Substitution – S) – производит алгоритмическую или чаще табличную подстановку для обеспечения нелинейных свойств шифра. Третий слой – линейное преобразование (Linear Transformation, LT) – производит активное обратимое перемешивание данных внутри блока. Как видим, по сравнению с SP-сетью здесь добавление ключа вынесено в отдельный слой, разделены линейная и нелинейная часть преобразования, а перемешивание данных производится с помощью

быстро реализуемых в программной реализации линейных преобразований.

Практически независимой от двух предыдущих разработок и очень популярной схемой блочных шифров является сеть Файстеля. Для этого предварительно из секретного ключа $\overline{K} = \overline{V}_0$ при помощи заданного детерминированного преобразования, входящего в описание алгоритма шифрования, формируется последовательность текущих ключей $\overline{V}_i$, $I = 1,2,...,r$, где r – число итераций.

В каждом из раундов сети Файстеля только по одному слою, т. е. все преобразования однотипны. Однако сама структура преобразования специфична: на какую-либо часть шифруемого блока обратимой операцией накладывается значение функции (возможно, необратимой), вычисленное от другой части блока.

Сама структура Файстеля предполагает следующий способ преобразования блоков. Сначала каждый блок сообщения, образованный последовательностью двоичных символе длиной n, разбивается пополам на подблоки длиной $n_0 = n/2$ (рис. 17), для этого n должно быть четным.

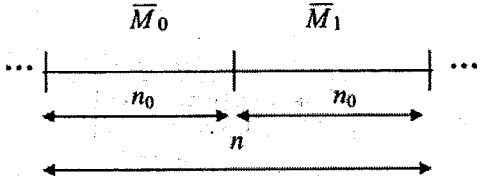


Рис. 17. Подготовка блока для шифрования на основе структуры Файстеля

Шифрование каждого блока выполняется отдельно с использованием рекуррентного соотношения

$$\overline{M}_{i+1} = \overline{M}_{i-1} \oplus f\left(\overline{K}_i, \overline{M}_i\right),\ i = 1, 2,..., r.$$

Здесь $f(\ )$ – нелинейная детерминированная открытая функция двух аргументов; $\overline{K}_i$ – текущие ключевые элементы (материал ключа), детерминированным образом сформированные из основного секретного ключа $\overline{K}$ данной криптосистемы.

Криптограмма образуется на заключительной итерации ($i = r$) и имеет вид блока $\overline{E} = \left(\overline{M}_r, \overline{M}_{r+1}\right)$ длиной n. Этот блок также со-

ставлен из последовательно записанных друг за другом подблоков, но теперь уже $\overline{M}_r$ и $\overline{M}_{r+1}$. Отметим, что примененное объединение подблоков в единый блок называется *конкатенацией.*

Классическая структура сети Файстеля приведена на рис. 18.

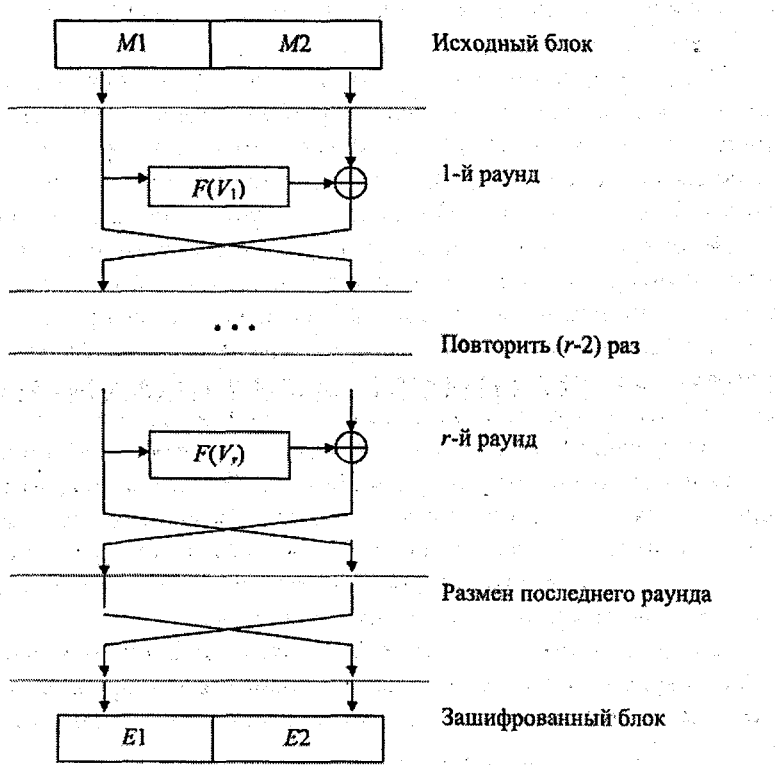


Рис. 18. Классическая сеть Файстеля

Независимые потоки информации, порожденные из исходного блока, называются ветвями сети. В классической схеме их две. Величины V_i именуются параметрами сети, обычно их роль играет материал ключа. Функция F называется образующей. Оптимальное число раундов r для сети Файстеля – от 8 до 32. Данная схема является обратимой. Сеть Файстеля обладает тем свойством, что даже если в качестве образующей функции F будет использовано необратимое преобразование, то и в этом случае вся цепочка будет

восстановима. Это происходит вследствие того, что для обратного преобразования сети Файстеля не нужно вычислять функцию F^{-1} – и при дешифровании вычисляется только прямое значение F. Обратимость требует только две операции наложения F на шифруемую в данном раунде ветвь, поэтому здесь чаще всего применяется XOR, реже – арифметическое сложение.

Более того, как нетрудно заметить, сеть Файстеля, в которой для наложения значения F используется операция XOR, симметрична. Исключающее ИЛИ обратимо своим же повтором и потому инверсия последнего обмена ветвей делает возможным дешифрование блока той же сетью Файстеля, но только в обратном порядке параметров V_i. Заметим. что для обратимости сети Файстеля не имеет значение, является ли число раундов четным или нечетным числом. В большинстве реализаций сети Файстеля прямое и обратное преобразования производятся одним и то же фрагментом программного кода, оформленного в виде процедуры, которому в качестве параметра передается вектор величин V_i либо в исходном (для шифрования), либо в инверсном (для дешифрования) порядке.

Преимущества структуры Файстеля состоят в том, что, во-первых, для нее выполняется принцип перемешивания подблоков и, во-вторых, хотя функция $F(,)$ может и не иметь обратную $F^{-1}(,)$, весьма просто реализуется алгоритм шифрования ключом, известным законному пользователю.

Имеется много различных блоковых шифров, основанных на структуре Файстеля, которые отличаются выбором функции F и способом формирования текущих ключевых элементов $\overline{V}_i$, $i = 1,2,\ldots,r$, из секретного ключа $\overline{K}$.

С незначительными доработками сеть Файстеля можно сделать и абсолютно симметричным шифром, выполняющим функции шифрования и дешифрования одним и тем же набором операций. Модификация сети Файстеля, обладающая подобными свойствами приведена на рис. 19. Как видно, основная ее хитрость в повторном использовании данных ключа в обратном порядке во второй половине цикла. Необходимо заметить, однако, что именно из-за этой недостаточно исследованной специфики такой схемы (т.е. потенциально возможности ослабления зашифрованного тек-

ста обратными преобразованиями) ее используют в криптоалгоритмах с большей осторожностью.

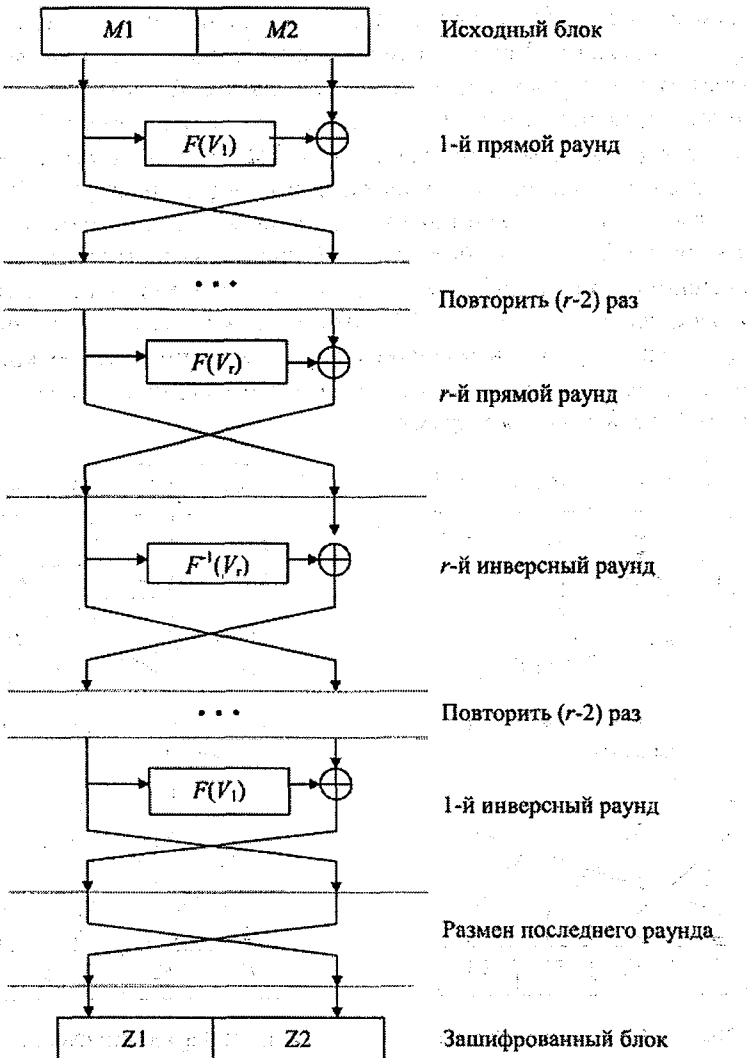


Рис. 19. Абсолютно симметричная сеть Файстеля

А вот модификацию сети Файстеля для большего числа ветвей применяют гораздо чаще. Это в первую очередь связано с тем, что при больших размерах шифруемых блоков (128 и более бит) становится неудобно работать с математическими функциями по модулю 64 и выше. Как известно, основные единицы информации, обрабатываемые процессорами, на сегодняшний день – это байт и двойное машинное слово (32 бита). Поэтому все чаще в блочных криптоалгоритмах встречается сеть Файстеля с 4-мя ветвями. Самый простой принцип ее модификации изображен на рис. 20. Для более быстрого перемешивания информации между ветвями (а это основная проблема сети Файстеля с большим количеством ветвей) применяются две модификации схемы, называемые тип-2 и тип-3. Они изображены на рис. 21, 22 соответственно. За счет частичного объединения обработки потоков в двух последних схемах количество раундов в них лишь не намного больше количества раундов в сети Файстеля с двумя ветвями.

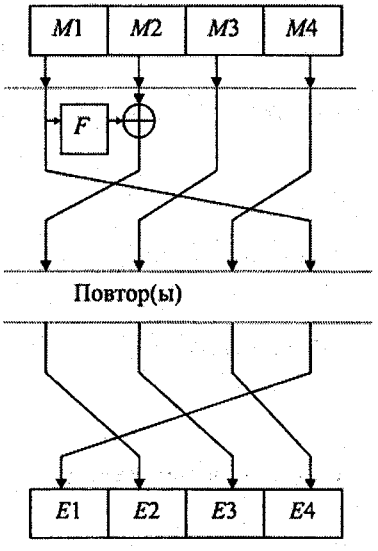


Рис. 20. Раунд сети Файстеля с 4-мя ветвями, тип 1

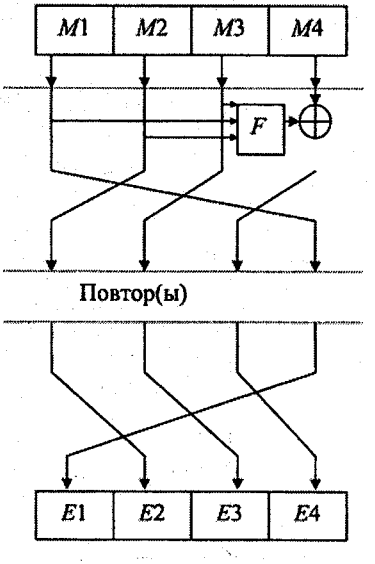


Рис. 21. Раунд сети Файстеля с 4-мя ветвями, тип 2

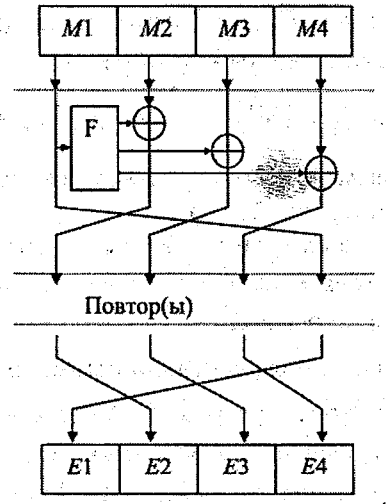


Рис. 22. Раунд сети Файстеля
с 4-мя ветвями, тип 3

Основная часть современных криптоисследований, посвященных сети Файстеля, относится к структуре образующей функции F и правилу вычисления материала ключа V_i. Из других решений интерес представляют только дополнительные начальные и оконечные преобразования данных («забеливание») над шифруемым блоком. Подобные обратимые преобразования выполняются либо операцией XOR, либо сложением. Начальное «забеливание» имеет целью произвести перемешивание данных блока с частью материала ключа на самой ранней стадии и, тем самым, повысить рандомизацию входного текста и, как следствие, еще сильнее снизить вероятность успешной атаки по известному открытому тексту. Выходное преобразование усложняет обратную трассировку при частично известном злоумышленнику ключе, кроме того, делает сеть Файстеля с «забеливанием» снова симметричной.

3.1.3. Многократное шифрование блоков

На первый взгляд представляется очевидным, что можно значительно повысить стойкость шифра, если криптограмму, полученную с помощью ключа $\overline{K}_1$, зашифровать еще раз с по-

мощью другого ключа $\overline{K}_2$, т.е. реализовать процедуры шифрования и дешифрования следующим образом:

$$\overline{E} = f_{K_2}\left(f_{K_1}\left(\overline{M}\right)\right), \qquad \overline{M} = g_{K_1}\left(g_{K_2}\left(\overline{E}\right)\right).$$

Здесь применяемые ключи указываются как нижние индексы функций шифрования/дешифрования.

Пусть длина используемых ключей одинакова и равна N, а криптоанализ криптограммы $\overline{E}$ выполняется с полным перебором всех возможных пар ключей $\overline{K}_1$ и $\overline{K}_2$.

Сначала выбирают первого кандидата на ключ $\overline{K}_2$ и производят дешифрование криптограммы $\overline{E}$. Полученный результат дешифруют 2^N раз, перебирая 2^N возможных кандидатов на ключ $\overline{K}_1$ в надежде получить читаемое сообщение. При неудаче производят дешифрование исходной криптограммы с помощью второго кандидата на ключ $\overline{K}_2$ и снова дешифруют полученный результат, перебирая 2^N возможных кандидатов на ключ $\overline{K}_1$ и т.д. Поскольку число возможных кандидатов на ключ $\overline{K}_2$ также составляет 2^N, может показаться, что верхняя граница необходимого числа расшифровок при переборе двух ключей должна быть 2^{2N}.

Однако можно показать, что это не так. Если длина ключей равна N, то верхняя граница необходимого числа операций для криптоанализа двукратного шифрования равна только 2^{N+1}, поэтому двукратное шифрование не является эффективным.

В силу сказанного для повышения стойкости шифрования используют не двойное, а тройное шифрование на трех различных ключах, т.е. формируют криптограмму по следующему правилу:

$$\overline{E} = f_{K_3}\left(g_{K_2}\left(f_{K_1}\left(\overline{M}\right)\right)\right), \qquad \overline{M} = g_{K_1}\left(f_{K_2}\left(g_{K_3}\left(\overline{E}\right)\right)\right).$$

Доказывается, что в этом случае наилучший метод криптоанализа при помощи тотального перебора ключей потребует 2^{2N} шагов, т.е. стойкость шифра существенно увеличивается.

В упрощенном способе повышения криптостойкости блоковых шифров с использованием трех ключей два из них применяют в качестве маскирующих слагаемых:

$$\overline{E} = \overline{K}_3 \oplus f_{K_2}\left(\overline{K}_1 \oplus \overline{M}\right), \qquad \overline{M} = \overline{K}_1 \oplus g_{K_2}\left(\overline{K}_3 \oplus \overline{E}\right),$$

длина маскирующих ключей $\overline{K}_1$ и $\overline{K}_3$ может не совпадать с длиной ключа $\overline{K}_2$.

3.2. Потоковые шифры

3.2.1. Аддитивные потоковые шифры

Идеально стойкая, теоретически недешифруемая система потокового шифрования имеет ключ, образованный истинно случайной последовательностью, символы который равновероятны и взаимонезависимы. Эта система шифрования имеет два очевидных и существенных недостатка. Во-первых, разрядности ключа и сообщения, представленного в цифровой форме должны быть одинаковыми, а во-вторых, ключ может быть использован только один раз.

При передаче больших объемов информации среди многих пользователей такая система шифрования оказывается совершенно непрактичной и чрезвычайно дорогостоящей, поскольку требует не только генерации большого числа длинных ключевых последовательностей, но и секретного распределения созданных данных ключей среди соответствующих законных отправителей и получателей сообщений.

По этой причине совершенные криптосистемы, для которых стойкость к дешифрованию без знания ключа не зависит от вычислительной мощности оппонента, применяют лишь в исключительных случаях, а при построении эффективных широко используемых алгоритмов шифрования ограничиваются вычислительно стойкими алгоритмами.

Криптосистема является вычислительно стойкой, если наилучший алгоритм дешифрования без знания ключа требует времени больше, чем имеется в распоряжении оппонента, в частности, если время криптоанализа превышает время, в течение которого сообщение остается актуальным.

Вычислительно стойкая система аддитивного потокового шифрования имеет ключ разумной длины. Этот ключ вводится в генератор псевдослучайных последовательностей (ГПСП) и

управляет его работой. Генератор ПСП по известному алгоритму расширяет относительно короткий ключ $\overline{K}$ в гамму $\overline{\Gamma}$, длина которой должна быть не меньше длины сообщения и которая может использоваться только один раз. Для шифрования следующего сообщения на том же ключе генерируется новая гамма.

Гамма является детерминированной функцией ключа. Но ее статистические свойства не отличаются от свойств случайной последовательности, в силу чего она и называется псевдослучайной последовательностью. При известном ключе шифрования гамма может быть легко повторена и отправителем, и получателем.

Главной отличительной чертой поточных шифров является побитная обработка информации. Как следствие, шифрование и дешифрование в таких схемах может обрываться в произвольный момент времени, как только выясняется, что передаваемый поток прервался, и также восстанавливаться при обнаружении факта продолжения передачи. Подобная обработка информации может быть представлена в виде автомата, который на каждом такте:

– генерирует по какому-либо закону один бит шифрующей последовательности;

– каким-либо обратным преобразованием накладывает один бит открытого потока на данный шифрующий бит, получая зашифрованный бит.

Все современные поточные шифры действуют по данной схеме. Бит шифрования, получающийся на каждом новом шаге автомата, как впрочем, и целый набор таких бит, принято обозначать символом $\overline{\Gamma}$ (гамма), а сами поточные шифры получили из-за этого второе название – шифры гаммирования. Общая схема шифрования поточным шифром приведена на рис. 23.

Гамма $\overline{\Gamma}$ формируется независимо от символов сообщения $\overline{M}$. Каждому символу сообщения и соответствующему символу криптограммы соответствует один и тот же символ гаммы, поэтому ни удаление, ни вставка каких бы то ни было символов в криптограмму недопустимо. Вся передача осуществляется в синхронном режиме. Ошибка при приеме символа криптограммы вызывает ошибку только в соответствующем символе расшифрованного текста. Размножения ошибок нет.

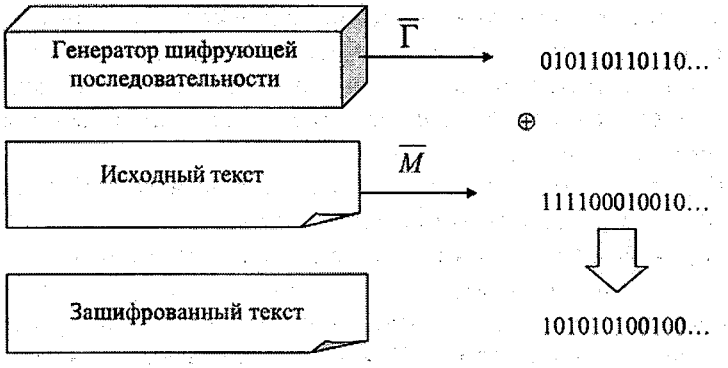


Рис. 23. Шифрование поточным шифром в общем виде

Потоковые шифры имеют следующие преимущества перед блоковыми:

– проще и дешевле аппаратная реализация;
– высокая скорость шифрования;
– отсутствует размножение ошибок, возникающих в каналах связи.

Эти преимущества привели к тому, что потоковые шифры получили наибольшее распространение при шифровании оцифрованных речевых сигналов и цифровых данных, требующих оперативной доставки потребителю. Наиболее типичным примером использования потоковых шифровых является защита информации в сетях GSM.

Датчики гаммы весьма разнообразны. Во многих потоковых шифрах в качестве датчиков гаммы применяется хорошо известный из различных технических приложений рекуррентный метод формирования псевдослучайных последовательностей с помощью так называемого линейного рекуррентного регистра с обратными связями (ЛРР).

3.2.2. Применение линейных рекуррентных регистров для потокового шифрования

Индивидуальные свойства линейного рекуррентного регистра сдвига с обратными связями (рис.24) определяет двоичная последовательность множителей $h_0, h_1, \ldots, h_n$, где $h_0=h_n=1$. По-

скольку среди элементов последовательности некоторые имеют нулевые значения, отводы оказываются подключенными не ко всем сумматорам, а только к тем, которым соответствуют h_i=0. Текущее состояние ЛРР определяется конкретным заполнением всех ячеек памяти $a_0, a_1, \ldots, a_{n-1}$.

Перед пуском производится заполнение ячеек ЛРР элементами начальной двоичной последовательности *C*.

Под действием каждого поступающего тактового импульса вычисляется по модулю 2 сумма $a_0 + h_1a_1 + h_2a_2 + \ldots + h_{n-1}a_{n-1}$ и происходит сдвиг содержимого всех ячеек в сторону выхода. Шаг сдвига – одна ячейка. Освободившаяся ячейка с номером $n-1$ заполняется найденной суммой, а содержимое ячейки a_0 с нулевым номером покидает регистр и становится очередным символов b_j выходной последовательности. По окончанию следующего такта на выходе появится содержимое предыдущей ячейки a_1, в качестве символа b_{j+1} и т.д.

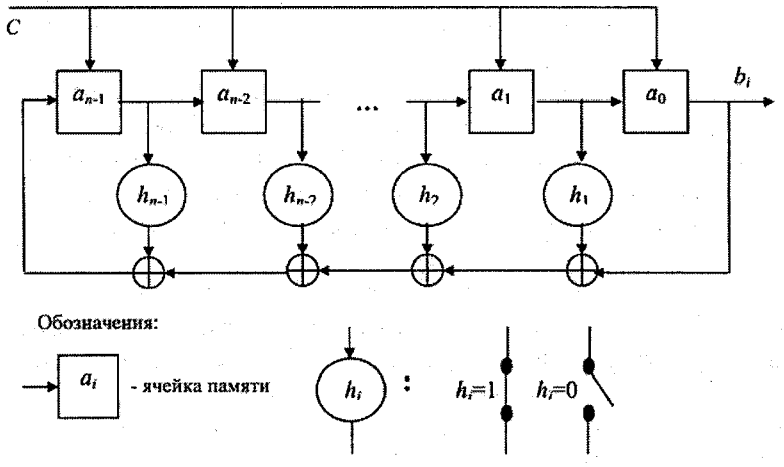


Рис. 24. Линейный рекуррентный регистр сдвига с обратными связями

Каждому ЛРР можно сопоставить полином обратной связи

$$h(x) = x^n + h_{n-1}x^{n-1} + h_{n-2}x^{n-2} + \ldots + h_1x + 1,$$

с двоичными коэффициентами h_i, где $h_0 = h_n = 1$. Верно и обратное утверждение: по заданному степенному полиному $h(x)$ всегда можно построить ЛРР. Например, полиному $h(x) = x^4 + x + 1$ соот-

ветствует линейный рекуррентный регистр сдвига (рис. 25) из 4 ячеек памяти ($n = 4$).

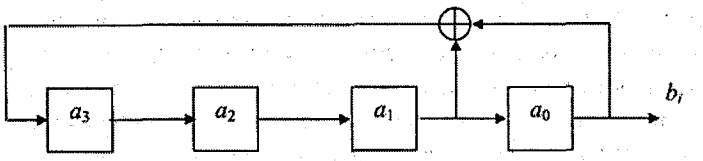


Рис. 25. ЛРР, соответствующий полиному $h(x) = x^4+x+1$

Линейный рекуррентный регистр сдвига длиной n имеет следующие основные свойства:

Период T выходной последовательности из символов b_i при любом начальном заполнении ячеек памяти ограничен неравенством $T \le 2^n - 1$.

Период T выходной последовательности максимален, т.е. определяется равенством $T = 2^n - 1$ при любом начальном заполнении тогда и только тогда, когда полином обратной связи $h(x)$ является примитивным полиномом.

Полином $h(x)$ степени n с двоичными коэффициентами называется примитивным, если он:

– является неприводимым, т.е. не представим в виде произведения двух или более многочленов с двоичными коэффициентами (при этом все действия выполняются по модулю 2);

– делит полином $x^{2^{n-1}} + 1$ без остатка, но не делит без остатка ни один полином вида $x^{n'}+1$ при любом $n' < 2^n - 1$.

Пример примитивного полинома: x^4+x+1. Полином x^4+x+1 делит $x^{15}+1$, поскольку

$$x^{15} + 1 = (x^4 + x + 1)(x^{11} + x^8 + x^7 + x^5 + x^3 + x^2 + x + 1).$$

Например, полином x^2+1 примитивным не является.

При больших значениях степени n число примитивных полиномов весьма велико. В настоящее время имеются подробные таблицы примитивных полиномов, в том числе и больших степеней. Имеются также алгоритмы, которые позволяют проверить, является ли случайно сгенерированный полином, даже весьма высокой степени, примитивным. Таким образом, не существует проблемы выбора примитивного полинома обратной связи

$h(x)$ для ЛРР достаточно большой длины n, т.е. такого полинома, который обеспечит достаточно большой период выходной последовательности $T = 2^n-1$ при любом начальном заполнении. Отметим также, что для сокращения числа отводов в конструкции ЛРР среди примитивных полиномов данной степени используют полиномы с наименьшим числом ненулевых коэффициентов.

Выходная последовательность линейного рекуррентного регистра сдвига, реализованного на примитивном полиноме, обладает свойствами «баланса» и «окна». Свойство «баланса» состоит в том, что число нулей на периоде $T = 2^n - 1$ точно равно $2^{n-1}-1$, а число единиц равно 2^{n-1}. Свойство «окна» гарантирует, что во всех 2^{n-1} «окнах» из n символов, расположенных друг за другом на периоде, все возможные 2^{n-1} ненулевые двоичные последовательности появятся только по одному разу.

Перечисленные свойства ЛРР, а также некоторые другие, приводят к тому, что последовательность символов на выходе ЛРР может быть принята за чисто случайную. Она практически не отличается от последовательностей, получаемых при бросании симметричной монеты. Однако в действительности выходная последовательность ЛРР является строго детерминированной. Она однозначно задана начальным заполнением $a_j \in \{0,1\}$, $j = 0,\ldots,n-1$ и полиномом связи $h(x)$, в силу чего ее называют псевдослучайной последовательностью.

Использовать ЛРР непосредственно в качестве датчика гаммы $\overline{\Gamma}$ (рис. 26) в потоковом шифраторе нельзя. ЛРР обладает следующим свойством: если известна выходная последовательность ЛРР длиной всего лишь $2n$, можно вычислить как начальное заполнение, так и полином обратной связи однозначно, причем сложность решения этой задачи имеет порядок $O(n^3)$, т.е. требует примерно n^3 операций. Данное свойство является непосредственным следствием линейности ЛРР. Оно позволяет свести решение задачи криптоанализа к решению системы линейных уравнений.

Чтобы избежать описанного нападения, нужно нарушить линейность датчика гаммы, например, используя два линейных рекуррентных регистра, из которых первый ЛРР служит для второго ЛРР генератором тактовых импульсов, или применяя нелинейные узлы усложнения.

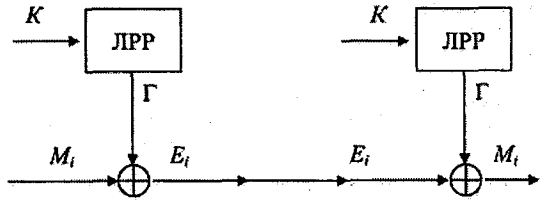


Рис. 26. Применение ЛРР для потокового шифрования

На сегодняшний день из открытых разработок нелинейных поточных шифров наибольшую популярность получили три класса алгоритмов. Это комбинирующие, фильтрующие и динамические поточные шифры. Все они в качестве базовых элементов используют линейные рекуррентные регистры сдвига, причем только те, которые порождают последовательности наибольшей длины. Во всех трех методах основной целью становится получение большого периода порождаемой гаммы, высокой нелинейности, устойчивости к атакам по открытому тексту и корреляционным атакам.

Фильтрующие шифры имеют, пожалуй, наиболее простую структуру из нелинейных поточных шифров. Они строятся на базе одного ЛРР созданием от его ячеек дополнительных отводов, никак не связанных с отводами обратной связи. Значения, получаемые по этим отводам, смешиваются на основе какой-нибудь нелинейной функции – фильтра (рис. 27). Бит-результат данной функции и подается на выход схемы как очередной бит гаммы.

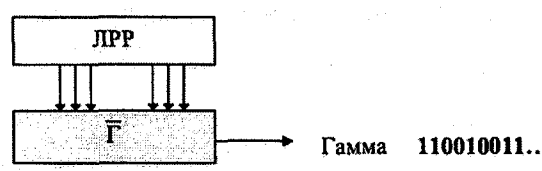


Рис. 27. Фильтрующий поточный шифр

Комбинирующие шифры строятся на основе нескольких ЛРР, объединяя нелинейной функцией биты, порождаемые каждым из них на очередном шаге (рис. 28).

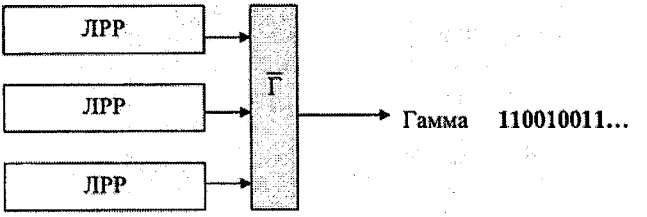


Рис. 28. Комбинирующий поточный шифр

Одной из основных проблем комбинирующих шифров является возможность просачивания информации из одного или нескольких ЛРР в выходную гамму. Так, в простейшем примере, приведенном на рис. 29, итоговая шифрующая последовательность порождается по формуле $\overline{\Gamma} = a \oplus b(b\&c)$. Несмотря на то что подобная гамма является сбалансированной, т. е. вероятности появления нулей и единиц в ней совпадают, в нее с неприемлемо большой вероятностью проходит поток бит, порождаемый первым ЛРР. Действительно, вероятность $P(\overline{\Gamma} = b) = 4/8 = 1/2$, $P(\overline{\Gamma} = c) = 4/8 = 1/2$, но $P(\overline{\Gamma} = a) = 6/8$. Это позволяет при несколько большем объеме перехваченной информации, чем это требовалось ранее, проводить на первый ЛРР все те же атаки, что и на одиночный ЛРР. А получив всю необходимую информацию по первому ЛРР, можно либо непосредственно воспользоваться ею для чтения с «домысливанием» открытого текста, либо попытаться узнать больше информации о втором и третьем ЛРР.

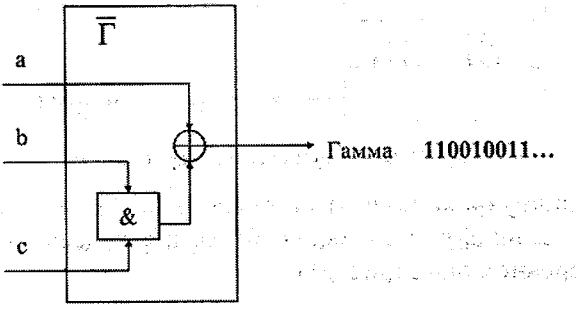


Рис. 29. Схема $\overline{\Gamma} = a \oplus (b\&c)$

Описанный метод называется корреляционной атакой на комбинирующий шифр. Комбинирующий шифр называется корреляционно-стойким, если для любой комбинации ЛРР, входящих в его состав, значение коэффициента корреляции между битами этих ЛРР и результирующим битом шифра равно 1/2.

Однако после проведения математических исследований оказалось, что чем выше степень нелинейности объединяющей функции нелинейного фильтра, тем ниже его корреляционная стойкость и наоборот. Разработчики оказались между двумя криптоаналитическими атаками. Одним из методов выхода из данной ситуации оказалось использование в объединяющей функции элементов памяти. Подобные ячейки хранят несколько бит, характеризующих предыдущее состояние функции и «подмешивают» его к ней же на этапе вычисления (рис. 30).

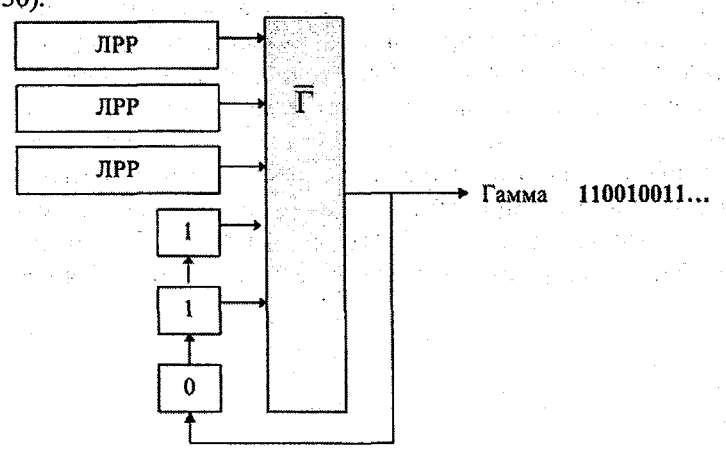


Рис. 30. Комбинирующий поточный шифр с элементами памяти

Динамические шифры строятся также на базе нескольких ЛРР, но объединяют их не на равноправных условиях, а по схеме «начальник-подчиненный». Так, например, в схеме, приведенной на рис.31, бит, порождаемый управляющим регистром, определяет бит, от какого из подчиненных ЛРР (первого или второго) будет подан на выход всего алгоритма. Бит от противоположного выбранному регистра просто отбрасывается.

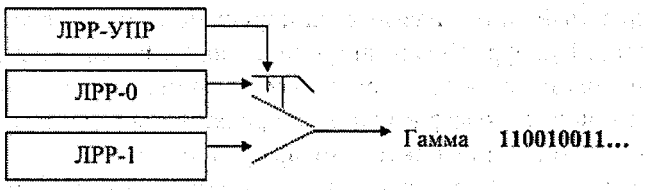


Рис. 31. Динамический поточный шифр с выбором ЛРР

По другой схеме, если на очередном своем шаге управляющий ЛРР генерирует «1», то бит подчиненного ЛРР, полученный на этом же шаге, поступает на выход системы, а если же управляющий ЛРР выдал «0», то бит подчиненного регистра просто отбрасывается. Возможны и сложные схемы взаимодействия ЛРР и их комбинации с уже описанными методами.

При правильном построении закона объединения нескольких ЛРР как в комбинирующих, так и в динамических шифрах период результирующей последовательности будет равен произведению периодов или полупериодов каждого отдельно взятого ЛРР (в зависимости от схемы их объединения). Единственным необходимым условием для этого является то, чтобы разрядности всех ЛРР были попарно взаимно просты, т.е. не имели общих делителей кроме единицы. Чаще всего разрядности всех ЛРР, входящих в общий блок, выбираются как простые числа из подходящего диапазона.

4. АСИММЕТРИЧНЫЕ КРИПТОСИСТЕМЫ

4.1. Особенности асимметричных криптосистем

Асимметричная криптосистема – это такая криптосистема, в которой ключ шифрования $K_ш = K$ не совпадает с ключом дешифрования $K_д = k$, т.е. $k \neq K$, причем

$$E = f_k(M) = E_k(M),\ M = g_k(E) = M_k(E).$$

В дальнейшем для обозначения ключей шифрования и дешифрования в асимметричных криптосистемах будем использовать прописную K и строчную k.

Эти системы, имеют еще одно название в связи со способом их применения – *криптосистемы с открытым ключом.*

Сформулируем основные требования, которым должна удовлетворять практически реализуемая криптосистема с открытым ключом.

1. Вычисление пары ключей K и k должно быть просто решаемой задачей при наличии некоторых данных, известных центру распределения ключей (ЦРК).

2. При известном ключе шифрования K вычисление криптограммы $E = f_k(M)$ должно быть достаточно простым.

3. При известном ключе дешифрования k восстановление сообщения $M = g_k(E)$ должно быть достаточно простым.

4. При известном ключе шифрования K вычисление ключа дешифрования k должно быть весьма сложным.

5. При известном ключе шифрования K, но неизвестном ключе дешифрования k, вычисление сообщения M по известной криптограмме E должно быть весьма сложным.

Если условие 4 выполнено, то, очевидно, что ключ шифрования K можно сделать открытым и при этом стойкость системы не нарушится. Именно вследствие этого и появилось название таких криптосистем.

Однако чтобы показать, что асимметричные криптосистемы имеют смысл, необходимо дать ответы на следующие два вопроса:

– зачем нужны криптосистемы с открытым ключом, если существуют достаточно стойкие симметричные криптосистемы (блоковые или потоковые)? Каковы их преимущества перед обычными системами?

– существует ли вообще асимметричные системы шифрования, удовлетворяющие основным требованиям?

Ответим на эти вопросы.

Основное преимущество асимметричных систем шифрования с открытым ключом состоит в том, что они существенно упрощают проблему распределения ключей в сети с большим числом пользователей.

Действительно, пусть имеется n пользователей, каждый из которых хочет вести переговоры с любым другим секретно от остальных, используя симметричную криптосистему. В этом случае каждый пользователь должен получить и хранить в секрете n–1 ключ, а общее число секретных ключей в сети, генерируемых и распределяемых ЦРК, должно составлять $C_n^2 = n(n-1)/2$.

Если же использовать асимметричные криптосистемы, то все пользователи должны получить от ЦРК только по одному секретному ключу дешифрования k_i, $i = 1,2,...,n$, в то время как соответствующие ключи шифрования K_i могут быть помещены в открытый справочник, доступный для всех. Если пользователь A хочет переслать сообщение пользователю B, то он формирует криптограмму $E = f_{K_B}(M)$ и пересыпает ее по открытому каналу связи (рис. 32). Пользователь B с помощью своего секретного ключа дешифрования восстанавливает сообщение $M = g_{K_B}(E)$.

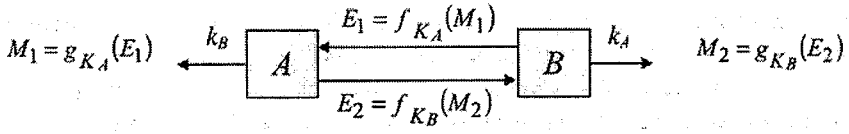


Рис. 32. Обмен сообщениями в двухключевой криптосистеме

Второе преимущество асимметричных систем по сравнению с симметричными состоит в том, что они, помимо функции шифрования/дешифрования, могут реализовывать и ряд других дополнительных функций (например, цифровую подпись).

Наконец, третьим преимуществом является то, что асимметричные системы имеют более «прозрачный» криптоанализ (т.е. проще оценивается стойкость конкретной системы), поскольку, как будет показано в дальнейшем, их криптостойкость базируется на сложности решения четко сформулированных математических задач.

Таблица 5

Длина ключей симметричных и асимметричных криптосистем при одинаковом уровне секретности

Длина ключа в битах	
симметричная криптосистема	асимметричная криптосистема
56	384
64	512
80	768
112	1792
128	2304

Однако в асимметричных криптосистемах приходится использовать более длинные ключи и процедура шифрования/дешифрования совершается медленнее. В табл. 5 сопоставлена длина ключей симметричных и асимметричных криптосистем при одинаковом уровне секретности.

4.2 Основы построения асимметричных систем

Для обеспечения безопасности распределения ключей с помощью ЦРК в асимметричных криптосистемах (рис. 33), где жирными линиями отмечены закрытые каналы распределения ключей) необходимо обеспечить защиту открытых ключей от подмены. Иначе злоумышленник может действительный открытый ключ какого-либо пользователя заменить своим открытым ключом и затем дешифровать все криптограммы, приходящие в адрес пользователя, ключ которого он подменил.

В асимметричной системе возможна также передача криптограмм между парой пользователей даже в том случае, когда они не имеют предварительно распределенных ключей, т.е. без содействия вспомогательного ЦРК.

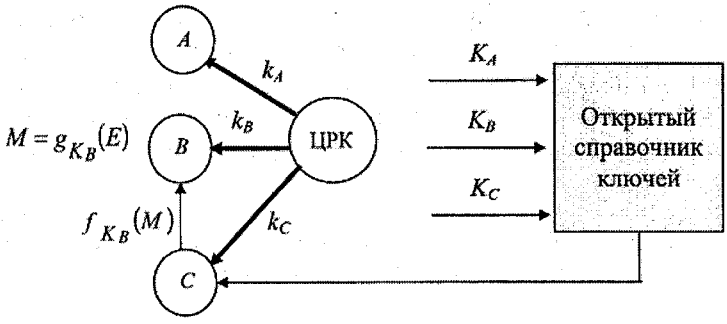


Рис. 33. Распределение ключей в асимметричной криптосистеме с помощью ЦРК в сети с тремя пользователями

Криптосистемы с открытым ключом могут быть построены на основе ряда известных математических задач, решение которых в отсутствие дополнительной информации является весьма сложным, а при выборе соответствующих параметров приводит к нереализуемо большому числу операций. Как правило, все криптосистемы с открытым ключом требуют выполнения действий с весьма большими целыми числами, причем безо всякого округления. Практически все действия производятся по модулю большого натурального числа n. Ключи также представляют собой большие целые числа.

Для построения асимметричных систем, удовлетворяющих условиям 1–5, обычно используют следующие трудные задачи из теории чисел, алгебры и комбинаторики:

– разложение большого целого числа N на множители (факторизация, $N = p_1^{\bullet} p_2 \ldots p_s$, где p_i – простые числа, $i = 1,2,\ldots,s$);

– нахождение квадратных корней по модулю n. Задано a, найти $\sqrt{a} \mod n$: $\sqrt{a}^{\bullet}\sqrt{a} = \mod n$;

– вычисление дискретных логарифмов по модулю n. Задано a, найти число c такое, что $c = \log_b a \mod n$, т.е. удовлетворяет уравнению $b^c = a \mod n$;

– задача об «укладке ранца». Даны натуральные числа a_1, a_2 ,…, a_s и n; найти такой двоичный вектор b_1, b_2, …, b_s, если он существует, чтобы выполнялось равенство;

– исправление ошибок произвольным линейным корректирующим кодом;

– дискретное логарифмирование на эллиптических кривых.

В настоящее время наиболее разработаны следующие виды криптосистем открытым ключом:

RSA, названной так по первым буквам фамилий ее авторов: Rumley, Shamir, Adleman (на основе трудных задач факторизации и дискретного логарифмирования);

Эль-Гамаля (на основе трудности нахождения дискретного логарифма);

Меркля-Хеллмана (на проблеме об «укладке ранца»);

Мак-Элиса (на задаче о декодировании линейного кода);

Коблица (на задаче о дискретном логарифмировании на эллиптических кривых).

Все криптосистемы с открытым ключом используют представление сообщений в виде целых чисел и преобразование этих целых чисел в криптограммы, представляющие собой также целые числа, поэтому математической основой всех систем с открытым ключом является, прежде всего, теория чисел.

Алгоритмы шифрования с открытым ключом получили широкое распространение в современных информационных системах. Так, алгоритм RSA стал мировым стандартом де-факто для открытых систем и рекомендован МККТТ.

Перейдем к детальному рассмотрению схемы RSA.

Для шифрования выбирают целое число $N = pq$, где p и q – два больших простых числа.

Сообщение M представляется одним из чисел

$$M \in \{1,2,\ldots, N-1\}.$$

Формулы, описывающие шифрование/дешифрование, имеют следующий вид:

$$E = M^K \bmod N, \qquad M = E^k \bmod N,$$

где K – открытый ключ, k – секретный ключ.

Покажем, что возведение в степень

$$E^k = \left(M^K\right)^k \bmod N,$$

выполняемое при дешифровании, действительно восстанавливает сообщение M.

В RSA пара ключей формируется по правилу

$$kK = 1 \bmod \varphi(N), \tag{1}$$

что эквивалентно

$$kK = 1 + l\varphi(N), \tag{2}$$

где l – целое число, а $\varphi(N)$ – функция Эйлера, которая определяет чисел меньших N и взаимно простых с ним.

По теореме Эйлера, если M и N – взаимно простые числа, то

$$M^{\varphi(N)} = 1 \bmod N.$$

Отсюда

$$M^{kK} = M^{1+l\varphi(N)} = M\left(M^{\varphi(N)}\right)^l = M \cdot 1^l = M \bmod N,$$

т.е. происходит дешифрование сообщения.

В то же время можно доказать, что специальный выбор модуля N в виде произведения $N = pq$ двух простых чисел p и q гарантирует выполнение равенства (2), т.е. дешифрование при любых M, не обязательно взаимно простых с N.

Отметим также, что степень M^K не должны быть меньше N.

При невыполнении данного условия криптограмма принимает вид

$$E = M^K,$$

и легко вскрывается без вычислений по модулю, так как открытый ключ K известен.

Рассмотренные соотношения указывают путь формирования ключей. Вначале выбирают очень большие случайные простые числа p и q, отличающиеся друг от друга на несколько десятичных разрядов, так чтобы произведение $N=pq$ имело длину не менее 768 бит (данные 2001 года). Вычисляют функцию Эйлера

$$\varphi(N) = (p-1)(q-1).$$

Из равенства (1)

$$Kk = 1 \bmod \varphi(N)$$

следует, что оба ключа взаимно обратные числа по модулю $\varphi(N)$.

Открытый ключ K выбирают среди чисел меньших и взаимно простых с $\varphi(N)$.

Закрытый ключ k вычисляют

$$k=K^{-1} \bmod \varphi(N)$$

с помощью алгоритма Евклида, что не является сложной задачей.

Завершив подготовительные операции, открытый ключ K и модуль N помещают в открытый справочник, приняв меры, гарантирующие невозможность подмены. Числа k, p и q хранят в секрете.

Таким образом, оба ключа RSA задаются парами целых чисел: (K, N) и (k, N), причем числа K и k равноправны, т.е. ключи можно поменять местами, и использовать k как открытый ключ шифрования, а K как закрытый ключ дешифрования

Когда авторы R. Rivest, A. Shamir, L. Adleman (Ривест, Шамир, Адлеман) в 1977 г. предложили криптосистему RSA, они зашифровали фразу «Its all Greek to me», представленную в цифровой форме M. Были опубликованы значения E, K, N с указанием, что 129 десятичных разрядов N получены при умножении 6 и 65-разрядных чисел p и q. Факторизация N и вскрытие криптограммы были выполнены примерно за 220 дней лишь в 1994 г. после предварительной теоретической подготовки. В работе принимали участие 600 добровольцев и 160 компьютеров, объединенных в сеть с помощью Интернет.

4.3. Гибридная система шифрования

Стойкость системы с открытым ключом, и в частности RSA, зависит от выбора ее параметров. Если выбрать $\log_2 N$=200 бит, то для факторизации требуется примерно $2{,}7 \cdot 10^{11}$ операций, что на персональном компьютере несколько дней. Но если выбрать $\log_2 N$ = 664 бит, то для факторизации требуется уж $12 \cdot 10^{23}$ операций. При скорости выполнения 10^6 операций в секунду на факторизацию уйдет несколько миллиардов лет.

Таким образом, если параметры шифра RSA выбраны правильно и модуль N взять достаточно большим, например $\log_2 N \geq$ 1024, то данную систему можно считать достаточно стойкой. На сегодняшний день отсутствуют какие-либо методы ее успешного криптоанализа.

Реализация шифра RSA отработана как в программном, так и в аппаратном вариантах. Используется RSA и для шифрования в

смарт-картах. В программном варианте скорость шифрования имеет порядок 1 Кбит/с, в аппаратном – 10–50 Кбит/с.

Сравнительно низкая скорость шифрования и дешифрования по сравнению с симметричными, блоковыми или потоковыми системами является недостатком всех асимметричных систем шифрования, в том числе и для RSA.

Для преодоления этого недостатка используют так называемые гибридные системы шифрования, в которых непосредственное шифрование сообщений производится при помощи быстрого блокового или потокового шифра, а закрытый ключ, нужный для быстрого симметричного шифрования, шифруется при помощи криптосистемы с открытым ключом и передается в виде криптограммы.

Асимметричные шифры, например RSA, используются только для шифрования ключа при его смене. Поскольку в обычной симметричной системе ключ является достаточно коротким (< 512 бит) и меняют его достаточно редко, то шифрование и дешифрование ключа в гибридной системе в среднем не занимает много времени.

Относительный вклад медленного шифрования с открытым ключом в данной системе мал. Основная скорость шифрования определяется симметричным шифрованием.

Рассмотрим пример применения гибридной системы шифрования при условии, что центр распределения ключей снабдил участников информационного обмена их личными секретными ключами, а соответствующие открытые ключи поместил в общедоступный справочник. Теперь, если пользователь *A* должен послать пользователю *B* достаточно длинное секретное сообщение, он вырабатывает сеансовый ключ симметричного шифрования и с его помощью шифрует передаваемое сообщение. К полученной криптограмме он присоединяет еще и криптограмму использованного ключа, применив для ее создания открытый ключ получателя *B*, опубликованный в справочнике. В свою очередь получатель, приняв обе криптограммы, сначала использует свой закрытый ключ и расшифровывает сеансовый ключ симметричного шифрования, после чего расшифровывает криптограмму, содержащую сообщение.

5. ПРИЛОЖЕНИЕ КРИПТОГРАФИИ

5.1. Электронная цифровая подпись

При обмене электронными документами по сети связи существенно снижаются затраты на обработку и хранение документов, убыстряется их поиск. Но при этом возникает проблема аутентификации автора документа и самого документа, т.е. установления подлинности автора и отсутствия изменений в полученном документе. Обычной «бумажной» подписью традиционно заверяется подлинность документа. Стойкость подписи, то есть невозможность ее подделки посторонними лицами обеспечивается двумя основными условиями: во-первых, ее уникальностью, основанной на индивидуальных особенностях почерка, а во-вторых, физической целостностью бумажного документа, на котором произведена подпись. При этом подпись не может быть подделана даже тем лицом, которое проверяет ее подлинность.

Однако при передаче электронных документов по компьютерным сетям или хранении на машинных носителях воспользоваться данными условиями невозможно, в том числе и при передаче факсимильных сообщений (ФАКС), поскольку они допускают простую подделку.

Целью аутентификации электронных документов является их защита от возможных видов злоумышленных действий, к которым относятся:

– *активный перехват* – нарушитель, подключившийся к сети, перехватывает документы (файлы) и изменяет их;

– *маскарад* – абонент *C* посылает документ абоненту *B* от имени абонента *A*;

– *ренегатство* – абонент *A* заявляет, что не посылал сообщения абоненту *B*, хотя на самом деле послал;

– *подмена* – абонент *B* изменяет или формирует новый документ и заявляет, что получил его от абонента *A*;

– *повтор* – абонент *C* повторяет ранее переданный документ, который абонент *A* посылал абоненту *B*.

Эти виды злоумышленных действий могут нанести существенный ущерб банковским и коммерческим структурам, государственным предприятиям и организациям, а также частным лицам, применяющим в своей деятельности компьютерные информационные технологии.

При обработке документов в электронной форме совершенно непригодны традиционные способы установления подлинности по рукописной подписи и оттиску печати на бумажном документе. Принципиально новым решением является электронная цифровая подпись – ЭЦП.

Электронная цифровая подпись используется для аутентификации текстов, передаваемых по телекоммуникационным каналам. Цифровая подпись решает проблему возможного спора между отправителем и получателем, в том числе и в суде, при наличии юридической базы для ее применения.

Функционально она аналогична обычной рукописной подписи и обладает ее основными достоинствами:

– удостоверяет, что подписанный текст исходит от лица, поставившего подпись;

– не дает самому этому лицу возможности отказаться от обязательств, связанных с подписанным текстом;

– гарантирует целостность подписанного текста.

Одновременно с этими ЭЦП должна представлять собой относительно небольшое количество дополнительной цифровой информации (цепочку данных), которые можно передавать по сетям вместе с подписываемым текстом, т.е. она должна удовлетворять следующим четырем основным требованиям:

– цифровая подпись должна быть уникальной, т.е. никто, кроме автора, не может создать такую же подпись, включая лиц, проверяющих ее подлинность;

– каждый пользователь сети, законный или незаконный, может проверить истинность цифровой подписи;

– подписавший не может отказаться от сообщения, заверенного его цифровой подписью;

– как реализация, так и проверка цифровой подписи, должны быть достаточно простыми.

Чтобы удовлетворить всем перечисленным требованиям цифровая подпись, в отличие от «бумажной», должна зависеть от всех бит сообщения и изменяться даже при изменении одного бита подписанного сообщения. Для реализации цифровой подписи на основе симметричных криптосистем необходимо участие доверенного лица – арбитра. Реализация цифровой подписи без арбитра возможна только на основе использования асимметричных систем.

Система ЭЦП включает две процедуры:

1) процедуру постановки подписи;
2) процедуру проверки подписи.

В процедуре постановки подписи используется секретный ключ отправителя сообщения, в процедуре проверки подписи – открытый ключ отправителя.

При формировании ЭЦП отправитель, прежде всего, вычисляет хэш-функцию $h(M)$ подписываемого текста M. Вычисленное значение хэш-функции $h(M)$ представляет собой один короткий блок информации m, характеризующий весь текст M в целом. Затем число m шифруется секретным ключом отправителя. Получаемая при этом пара чисел представляет собой ЭЦП для данного текста M.

При проверке ЭЦП получатель сообщения снова вычисляет хэш-функцию $m = h(M)$ принятого по каналу текста M, после чего при помощи открытого ключа отправителя проверяет, соответствует ли полученная подпись вычисленному значению m хэш-функции.

Принципиальным моментом в системе ЭЦП является невозможность подделки ЭЦП пользователя без знания его секретного ключа подписывания.

В качестве подписываемого документа может быть использован любой файл. Подписанный файл создается из неподписанного путем добавления в него одной или более электронных подписей.

Каждая подпись содержит следующую информацию:

– дату подписи;
– срок окончания действия ключа данной подписи;
– информацию о лице, подписавшем файл (Ф.И.О., должность, краткое наименование фирмы);
– идентификатор подписавшего (имя открытого ключа);

– собственно цифровую подпись.

Существуют различные виды цифровой подписи, основанные на криптосистемах с открытым ключом. Одни опираются на сложность разложения больших чисел на простые множители, другие – на сложность дискретного логарифмирования в одной из конечных групп.

С 1-го июля 2002 г. в России введен стандарт электронной цифровой подписи ГОСТ Р.34.10-2001, опирающийся на сложность решения задачи «дискретного логарифмирования» в группе точек на эллиптической кривой. Эллиптические кривые изучаются в алгебраической геометрии и широко применяются в различных разделах математики, в том числе в теории чисел.

Рассмотрим (как более простую) реализацию цифровой подписи на основе криптосистемы RSA. Пользователь A, подписывающий сообщение M, генерирует пару ключей k_A, K_A сообщает пользователям сети значения K_A и N.

Далее пользователь A создает контрольную группу

$$E_s = M^k A \mod N,$$

которая и будет цифровой подписью сообщения M (рис. 34). Контрольная группа приписывается к сообщению и передается вместе с ним.

Рис. 34. Цифровая подпись на основе RSA

Любой другой пользователь сети, приняв сообщение M', при необходимости проверки цифровой подписи E'_s производит вычисление

$$(E'_s)^{K_A} \mod N.$$

Далее он сравнивает результат с полученным значением M'. При совпадении подпись пользователя A считается подлинной.

Легко убедиться, что в данном случае все четыре требования к цифровой подписи, сформулированные ранее, выполняются, если шифр RSA выбран достаточно стойким.

Рассмотренная система выработки цифровой подписи имеет два существенных недостатка:

– возникает значительная избыточность за счет приписывания контрольной группы, длина которой равна длине сообщения, что требует удвоения объема памяти, времени передачи и т.п.;

– при сообщениях большой длины операция возведения в степень при вычислении контрольной группы и ее проверке будет выполняться за недопустимо большое время.

Первая схема ЭЦП – RSA – была разработана еще в конце 1970-х гг. Однако проблема подтверждения авторства стала актуальной настолько, что потребовалось установление стандарта только в 1990-х гг. во время взрывного роста глобальной сети Интернет и массового распространения электронной торговли и оказания услуг. Именно поэтому стандарты ЭЦП в России и США были приняты практически одновременно в 1994 г. Из предложенных криптологами схем ЭЦП наиболее удачными оказались RSA и схема Эль-Гамаля. Первая была запатентована в США и ряде других стран (патент на RSA прекратил свое действие совсем недавно). Вторая имеет множество модификаций, которые весьма затруднительно запатентовать. Таким образом, схема ЭЦП Эль-Гамаля осталась по большей части свободной от патентов. Обладает она и целым рядом практических преимуществ: размер блоков, которыми оперируют алгоритмы, и соответственно размер ЭЦП в ней оказались значительно меньше, чем в RSA, при той же стойкости. Именно поэтому стандарты ЭЦП России и США базируются на схеме Эль-Гамаля. Согласно подписанному Президентом РФ 10 января 2002 г. Федеральному Закону «Об электронной цифровой подписи» ЭЦП должна обеспечить юридическую значимость как самих электронных документов, так и действий пользователей над ними (визирование, подписание, утверждение и т. п.).

5.2. Хеширование

Чтобы успешно реализовать цифровую подпись, в том числе и для длинных сообщений, сообщения предварительно подвергают сжатию в относительно короткие последовательности. Затем на основе асимметричного шифра, например RSA, производится цифровая подпись короткой цепочки бит.

Хэш-функция (англ. hash – мелко измельчать и перемешивать) предназначена для сжатия подписываемого документа до нескольких десятков или сотен бит. Хэш-функция $h(\cdot)$ принимает в качестве аргумента сообщение (документ) M произвольной длины и возвращает хэш-значение $h(M) = H$ фиксированной длины. Обычно хэшированная информация является сжатым двоичным представлением основного сообщения произвольной длины. Следует отметить, что значение хэш-функции $h(M)$ сложным образом зависит от документа M и не позволяет восстановить сам документ M.

Чтобы исключить возможность подделки цифровой подписи, должно использоваться однонаправленное хеширование, обладающее следующими свойствами.

1. При известном сообщении M вычисление $h = h(M)$ должно быть простым, причем без использования какого-либо ключа. Это означает, что процедура хеширования может быть выполнена любым пользователем. Применяемые при этом хеш-функции называются бесключевыми (бесключевое хеширование).

2. Хэш-функция должна обладать свойством необратимости, т.е. задача подбора документа M, который обладал бы требуемым значением хэш-функции $h = h(M)$, должна быть вычислительно неразрешима.

3. По заданному сообщению M должно быть трудно найти другое такое сообщение M' $(M \neq M')$, чтобы $h(M) = h(M')$.

4. Должно быть трудно, найти любую пару таких сообщений M' и M'', для которых $h(M') = h(M'')$.

Необходимость в выполнении четвертого условия связана с опасной атакой на цифровую подпись, использующую хеш-функцию. Эта атака, основана на так называемом парадоксе дней рождения.

Возьмем группу из m человек и событие, состоящее в том, что, по крайней мере, два человека в группе имеют один и тот же

день рождения. Оказывается, что если $m \geq 23$, то вероятность такого события не меньше чем 0,5.

Предположим, что длина хеш составляет L бит, тогда общее число возможных вариантов хеш равно 2^L. Основываясь на концепции дней рождения, видим, что достаточно перебрать $2^{L/2}$ сообщений и хотя бы для одной пары сообщений M и M' с вероятностью, близкой к единице, найдется общее значение хеш. При использовании хеш-функции для получения цифровой подписи это означает, что при случайном выборе $\sqrt{2^L}$ сообщений, где L – длина хеш, хотя бы одна пара из них будет иметь общее значение хеш и, следовательно, одну и ту же цифровую подпись.

Рассмотрим стратегию активного оппонента, который пытается сфабриковать фальшивую цифровую подпись на основе концепции дней рождения. Для начала злоумышленник формирует два сообщения, одно из которых M благоприятно для подписания неким пользователем A, и другое M' благоприятно для самого злоумышленника. После этого он производит небольшие изменения в сообщениях, не меняя их основного содержания и формируя тем самым две группы сообщений, отличающиеся по смыслу.

Основываясь на концепции дней рождения, нетрудно прийти к выводу, ему достаточно перебрать $2^{L/2}$ сообщений, чтобы найти общее значение хеш h для сообщений, принадлежащих разным группам.

Найдя для двух сообщений из разных групп общее значение хеш, активный оппонент предъявляет пользователю A сообщение благоприятное для подписания (из первой группы) и, в случае подписания, получает цифровую подпись h для сообщения из второй группы, т.е. для сообщения, имеющего смысл нужный оппоненту.

Если длина хеш будет недостаточна, например $L = 64$, то перебор 2^{32} вариантов вполне возможен. Поэтому длину хеш следует выбирать больше 256 или 512 бит.

Существуют различные способы построения хеш-функций. В основном они образуют две группы: в одной используются таблицы, в другой – стойкие блоковые шифры.

Большинство хэш-функций строится на основе однонаправленной функции $f(\cdot)$, которая образует выходное значение длиной

n при задании двух входных значений длиной n. Этими входами являются блок исходного текста M_i и хэш-значение H_{i-1} предыдущего блока текста (рис.35).

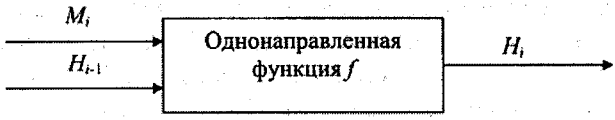


Рис. 35. Схема построения однонаправленной хэш-функции $H_i = f(M_i, H_{i-1})$

Хэш-значение, вычисляемое при вводе последнего блока текста, становится хэш-значением всего сообщения M.

В результате однонаправленная хэш-функция всегда формирует выход фиксированной длины n (независимо от длины входного текста).

Общепринятым принципом построения хэш-функций является итеративная последовательная схема. По этой методике ядром алгоритма является преобразование k бит в n бит. Величина n – разрядность результата хэш-функции, а k – произвольное число, большее n. Базовое преобразование должно обладать всеми свойствами хэш-функции, т.е. необратимостью и невозможностью инвариантного изменения входных данных.

Хэширование производится с помощью промежуточной вспомогательной переменной разрядностью в n бит. В качестве ее начального значения выбирается произвольное известное всем сторонам значение, например, 0.

Входные данные разбиваются на блоки по $(k - n)$ бит. На каждой итерации хэширования со значением промежуточной величины, полученной на предыдущей итерации, объединяется очередная $(k - n)$-битная порция входных данных и над получившимся k-битным блоком производится базовое преобразование. В результате весь входной текст оказывается «перемешанным» с начальным значением вспомогательной величины. Из-за характера преобразования базовую функцию часто называют сжимающей. Значение вспомогательной величины после финальной итерации поступает на выход хэш-функции (рис. 36). Иногда над получившимся значением производят дополнительные преобразования. Но в том случае, если сжимающая функция спроектирована с достаточной степенью стойкости, эти преобразования излишни.

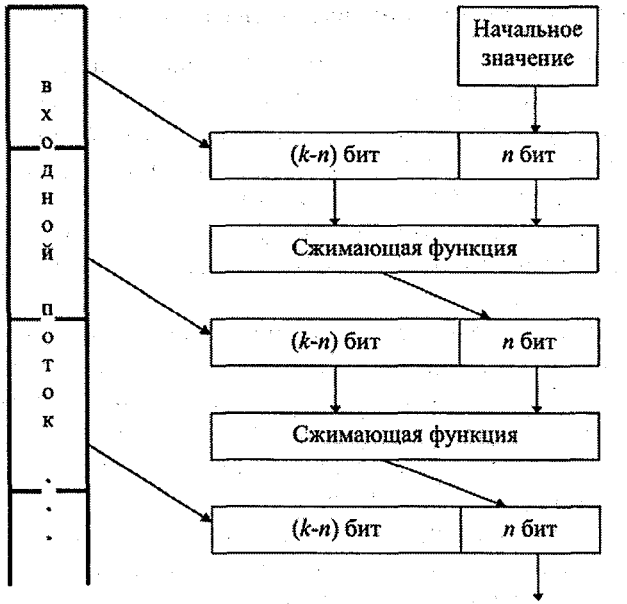


Рис. 36. Итеративная хэш-функция

При проектировании хэш-функции по итеративной схеме возникают два взаимосвязанных вопроса: как поступать с данными, не кратными числу $(k-n)$, и как добавлять в хэш-сумме длину документа, если это требуется. Есть два варианта решения этих вопросов. В первом варианте в начало документа перед хэшированием добавляется поле фиксированной длины (например, 32 бита), в котором в двоичном виде записывается исходная длина текста. Затем объединенный блок данных дополняется нулями до размера равного ближайшему кратному $(k-n)$ бит. Во втором варианте документ дополняется справа одним битом "1", а затем битами "0" до размера равного кратному $(k-n)$ бит. В этом варианте необходимость в поле длины отпадает – никакие два разных документа после выравнивания по границе порций не станут одинаковыми.

Кроме более популярных однопроходных алгоритмов хэширования существуют и многопроходные алгоритмы. В этом случае входной блок данных на этапе расширения неоднократно повторяется, а уже затем дополняется до ближайшей границы порции.

5.3. Обеспечение безопасности электронных платежей

В настоящее время значительная часть переводов между банками, а также переводы денежных средств клиентов с одного счета на другой и оплата покупок в магазинах производятся при помощи электронных коммуникационных средств. Это в большей степени относится к системе западных банков, но такая практика развивается и в России. Все это направление носит общее название электронных платежей (Electronics Found Transfer).

В случае когда клиенты осуществляют покупки или переводят денежные средства с помощью телекоммуникационных технологий без использования реальных денег, применяемая технология называется обычно «электронными деньгами» (Digital Money, DM), которые, конечно, не следует путать с цифровой наличностью (Digital Cash, DC).

Основные преимущества электронных денег состоят в увеличении безопасности платежей, повышении скорости операций и в полном отслеживании операций. В тех случаях когда это становится недостатком, можно использовать цифровую наличность (DC).

Чтобы обеспечить безопасность сделок между клиентами и банками, необходимо решить следующие задачи: выполнить идентификацию клиентов перед сделкой, гарантировать подлинность сделки (аутентификацию сделки), обеспечить доверие клиентов к автоматизированной системе и банку, иногда частичное.

Основой решения данных задач являются:

– существование телекоммуникационных систем между банками и терминалами, наличие банковских телекоммуникационных сетей;

– наличие терминалов универсального или специального типа, например, по продаже товаров в магазинах, по продаже и покупке валюты, приему чеков, оплате счетов и т.п.;

– пользование стойких криптографических методов для защиты платежей;

– использование физически защищенных технологий (криптомодулей, идентификационных карт, а также идентификаторов физических признаков клиентов).

Широко применяются системы электронных платежей на основе идентификационных карт. Эти карты известны и под другими названиями: пластиковые карты, кредитные карты, смарт-карты и т.п. Они предназначены для взаимодействия человека с различными автоматизированными системами, включая банковские. По смыслу карта – это аппаратное или аппаратно-программное средство, которое выполняет идентификацию субъекта (законного пользователя) системы, для чего хранит в памяти идентификационную информацию пользователя.

Современная технология защиты информации в финансово-кредитных системах электронных платежей состоит в использовании идентификационных карт.

Чтобы начать работу в сети, законный пользователь должен иметь возможность назвать (идентифицировать) себя. Для этого он помещает свою личную идентификационную карту в считывающее устройство терминала. Далее следует проверка подлинности идентифицирующих данных (аутентификация), а в случае ее успешного завершения следует признание пользователя законным и его допуск к автоматизированной системе.

В настоящее время система электронных платежей между клиентами и банками ориентирована, прежде всего, на использование технологий с идентификационными пластиковыми картами.

Первые применения идентификационных карт относятся к 50-м гг. XX в. Одновременно их стали использовать и для выполнения некоторых дополнительных функций. Классифицируют идентификационные карты по разным признакам (рис. 37). По материалу различают бумажные (картонные), пластиковые и металлические карты. Пластиковые настолько удобны и широко распространены, что стали синонимом идентификационных карт. На нагретом пластике хорошо выдавливаются изображения и рисунки, необходимые при персонализации карт. Пластиковая карта достаточно долговечна. В тяжелых условиях эксплуатации применяют металлические карты. Бумажные ламинированные карты сравнительно быстро изнашиваются, но дешевы и вполне пригодны для обеспечения небольших краткосрочных платежей.

Запись информации на картах может выполняться выдавливанием, заноситься в цифровом виде на магнитную пленку и в мик-

росхему (чип), создаваться лазером. Самым простым видом записи является нанесение графического изображения, которое может включать имя, образец подписи и цветное фото владельца, название распространителя карт (эмитента).

Там, где не требуется сложных расчетов, могут использоваться штрих коды, легко считываемые, но скрытые слоем, непрозрачным в видимом свете. Однако наиболее широкое распространение получила цифровая запись информации на пластиковых катах с магнитной полосой, микрочипом и оптической записью на более дорогих.

Рис. 37. Классификация пластиковых карт

Идентификационные карты с интегральной микросхемой, содержащей микроконтроллер или микропроцессор, часто называют интеллектуальными или разумными (smart) картами. Параметры их операционных возможностей в настоящее время близки к параметрам первых компьютеров.

Идентификационные карты применяются как удостоверение личности в охранных системах, социальных системах и в качестве платежного инструмента в кредитно-финансовой сфере.

В России сложилось деление карт на личные и корпоративные. Так называемые зарплатные карты совмещают в себе призна-

ки и тех и других. По картам, которые действуют в пределах остатка на счете, кредит не предоставляется, и они могут быть выданы любому клиенту.

Первый банкомат, в котором были применены пластиковые карты с магнитной полосой, появился в 1969 г. Эти карты изготавливаются строго в соответствии с международными стандартами. Стандарты определяют геометрические размеры карты, положение магнитной полосы, структуру записи цифровой формации на магнитной полосе и положение выдавливаемых изображений. Карты могут использоваться в интервале температур от –35 до +50°С при относительной влажности воздуха 5 – 95 %.

Магнитная полоса состоит из трех дорожек. Для записи на первой дорожке используется алфавит, в котором имеются 64 буквенно-цифровых символа. Каждый символ образуют 7 бит, причем последний бит нужен для проверки на четность. Для записей на второй и третьей дорожках алфавит содержит 16 буквенно-цифровых символов. Каждый символ образован цепочкой из 5 бит, последний также используется для проверки на четность.

На первой дорожке, предназначенной только для чтения, помещается 76 – 79 буквенно-цифровых символов: стартовая метка, идентификационный номер, имя владельца, дата окончания срока действия карты, трехзначное число, первая цифра которого определяет регион применения карты, вторая – платежный лимит, третья – необходимость запроса PIN-кода. Записываются на первую дорожку и другие сведения.

PIN-код (персональный идентификационный номер) применяется для подтверждения подлинности клиента (т.е. для аутентификации). Владелец карты вводит его в терминал отдельно от карты, например, с помощью клавиатуры.

На второй дорожке, размещенной в середине магнитной полосы и также предназначенной для чтения, помещается 37–40 буквенно-цифровых символов, которые используются для выполнения различных банковских операций. Наконец, третья дорожка предназначена как для считывания, так и для записи информации, нужной при проведении электронных транзакций. Она содержит 107 цифровых символов.

Режим взаимодействия пластиковой карты с банком в реальном времени (on-line) дает возможности повысить устойчивость системы к возможной компрометации ключей шифрования. Централизованная проверка идентификатора пользователя делает возможным оперативное обновление списков запрещенных к использованию карт. Этот режим возможен лишь при наличии надежных каналов связи между терминалами, которые устанавливаются в точках продажи или в любых других местах соприкосновения с клиентом, и с компьютером банка-распространителя, что делает этот режим довольно дорогим. Кроме того, наличие канала связи порождает и другие угрозы безопасности, например, анализ трафика и имитация работы компьютера банка-эмитента компьютером злоумышленника.

Пластиковая смарт-карта (SC) имеет в памяти первичный номер счета клиента – PAN (Primary Account Number). В память карты вместе с PAN также может быть включена дополнительная информация о названии банка, о размере счета и т.д.

При использовании карты владельцу необходимо вводить дополнительный код аутентификации. Этот код, по традиции называемый PIN (Personal Identification Number), предназначен для персональной аутентификации клиента (рис. 38). Он никогда не записывается на пластиковой карте и должен секретным образом храниться у владельца отдельно от карты. PIN-код никогда не передается по каналам связи и не хранится в компьютере банка-эмитента в открытом виде.

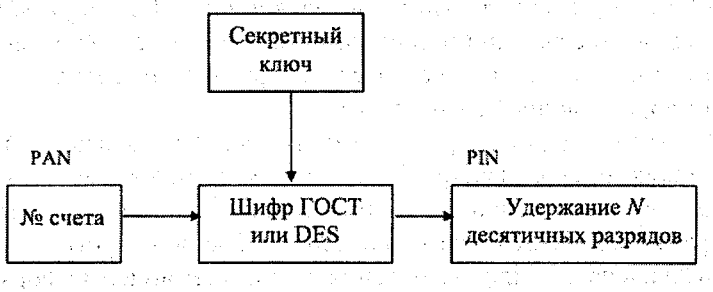


Рис. 38. Структурная схема формирования кода аутентификации пластиковой карты

Код аутентификации PIN содержит 4–8 цифр. Он может генерироваться как банком, так и владельцем карты. В последнем случае владелец создает случайное число и сообщает свой PIN в банк-распространитель карт. Там PIN зашифровывается специальным ключом банка-распространителя и хранится в виде криптограммы. Оба набора данных, № счета (PAN) и PIN некоторым образом связаны друг с другом.

Первые пластиковые карты с интегральной микросхемой появились во Франции в 1980 г. (патент журналиста Ролана Морено). В настоящее время существенно расширяется сфера применения интеллектуальных карт с металлическими контактами, вступающими в непосредственное физическое соединение с контактами считывающего/записывающего устройства при помещении карты в терминал. Существуют карты, использующие встроенную индукционную катушку для считывания и перезаписи информации по радио на расстоянии до 4 м от терминала. Скорость передачи данных до 9600 бит/с.

Известны следующие основные разновидности интеллектуальных карт.

– Карты с программируемым запоминающим устройством (PROM), применяемые для телефонных переговоров и для хранения ключей.

– Карты с энергонезависимым перепрограммируемым запоминающим устройством (EEPROM), применяемые для хранения индивидуальных данных.

– Карты с защищенным перепрограммируемым запоминающим устройством, которые обеспечивают доступ к чтению и записи после предъявления пароля или ключа.

По исполняемым функциям различают:

– Многофункциональные карты, применяемые с различными целями, в том числе и для аутентификации пользователей.

– Карты-счетчики с предварительно оплаченной суммой, если сумма не велика и не требуют защиты. При реализации сделки они вычитают истраченные суммы из остающейся в памяти, например, при плате за телефон, проезд, стоянку и т.п.

Различают карты, уничтожаемые после использования и возобновляемые. Объем памяти таких карт колеблется от 2048 до 8192 бит.

Карты памяти для хранения информации (пластиковые деньги, медицинские индивидуальные данные). Среди таких карт различают карты без зашиты и с защитой информации, с ограничением и без ограничения доступа к чтению/записи и стиранию. При выполнении защищенных операций требуется ключ. Объем памяти, защищенной PIN-кодом, составляет здесь 256 – 8192 бита.

Микропроцессорные карты, применяются для обеспечения электронных платежей с шифрованием данных по стандарту DES, с объемом постоянной памяти до 16 Кбайт, с объемом перезаписываемой памяти до 8 Кбайт, с асинхронным режимом обмена информацией при скорости 9600 бит/с.

Криптоконтроллеры, выполняющие с помощью встроенных алгоритмов DES с секретным ключом и RSA с открытым ключом шифрование/дешифрование информации, а также генерацию цифровой подписи.

Интеллектуальные карты имеют существенные преимущества перед обычными магнитными пластиковыми картами. Они значительно лучше защищены от использования посторонними лицами, а также от несанкционированных изменений записанной на карте суммы для сделок. Магнитные карты позволяют производить платежи в режиме on-line, т.е. при обмене данными с банком, в то время как smart-card позволяют при определенных условиях использовать и автономный (off-line) режим для выполнения определенных функций (платежей).

Смарт-карта обеспечивает следующий набор функций:

– разграничение полномочий доступа к внутренним ресурсам благодаря работе в защищенной файловой системе,

– шифрование данных с применением различных алгоритмов,

– формирование электронной цифровой подписи,

– ведение ключевой системы,

– выполнение всех операций взаимодействия владельца карты, банка и торговца.

Для работы с пластиковыми картами необходимо следующее дополнительное оборудование: расчетные терминалы, стационарные (переносные) кассовые аппараты, устройства, встроенные в различные торговые автоматы, банкоматы – устройства для выдачи наличности.

Каждый терминал обычно реализуется на базе персонального компьютера с дополнительными устройствами считывания и вместе со смарт-картой участвует в обеспечении логической безопасности.

Пластиковые карты обладают достаточной механической прочностью в интервале температур от –20 до +70°С. Выдерживают 1000 циклов (по 30 циклов в минуту) перекручивания на 30 градусов в обе стороны. Время хранения информации порядка 10 лет, минимальное число перезаписей около 10^4.

Информационная безопасность пластиковых карт основывается на криптографических методах, а также на использовании специальных безопасных микрокомпьютеров. Конструктивной и алгоритмической основой безопасного микрокомпьютера является невозможность доступа к его внутренним состояниям через входные и выходные порты системы.

Обычно на карте имеется несколько зон памяти:

– открытая (свободная от контроля),

– защищенная (для доступа к записи необходим пароль, а чтение свободное),

– конфиденциальная (требующая пароль для чтения),

– секретная (содержащая пароли и ключи).

Смарт-карты реализуются на основе особой технологии (MOS), которая исключает возможность микроскопического считывания содержимого памяти и внутреннего состояния при механическом вскрытии карт.

ЗАКЛЮЧЕНИЕ

Примеры приложений криптографии можно было бы продолжить. Важно отметить, что в течение длительного времени криптография была уделом отдельных талантливых изобретателей, как тех, кто создавал шифры, так и тех, кто пытался их разгадывать. Но постепенно, особенно начиная с XVII–XVIII вв., благодаря исследованиям и работам ряда ученых и, в частности, таких известных, как Л. Эйлер, Ф. Эпинус, Д. Кардано, Ф. Виет и др., позднее – К. Шеннон, А. Тьюринг, криптография прочно становится на научную основу.

В настоящее время криптография является самостоятельным, активно развивающимся направлением научных исследований, ее результаты имеют весьма широкую область практических приложений.

С гордостью можно отметить, что большой вклад в развитие криптографии как науки внесли наши отечественные ученые, всемирно известные по своим трудам в других областях научных исследований: академики А.Н. Колмогоров, В.А. Котельников, А.А. Марков, А.О. Гельфонд, Ю.В. Линник и др.

Россия, кстати, единственная страна, в которой образована и пользуется реальной государственной поддержкой Академия криптографии РФ, имеющая весьма высокий научный потенциал. В результате деятельности российских ученых работы ряда крупных научно-исследовательских институтов (один из которых, в частности, описан в произведениях Солженицына А.И.) в Советской России практически с нуля была создана целая отрасль разработки и производства надежных средств криптографической защиты информации. Она сыграла огромную роль в победе нашей страны в Великой отечественной войне и в ее послевоенном восстановлении в условиях жесткого противостояния с Западом и, к счастью, выжила и в трудных условиях смены в России форм собственности и соответствующего изменения структуры государственной власти.

Помимо упомянутых выше, а также ставших уже достаточно общеизвестными приложениями криптографии в области защи-

щенного обмена информацией по сетям связи общего пользования, в частности, по интернету, защиты электронных баз данных, закрытия информации, хранящейся на жестких дисках, на «интеллектуальных» картах и т.п., имеются и менее известные примеры применения криптографических методов, также заслуживающие упоминания.

Так, известно широкое распространение голограмм для защиты от подделок документов, различного рода продукции и т.д. Оказывается, что наиболее надежная голографическая защита может быть построена на основе помещения в голограмму абсолютно надежно защищенного методами криптографии цифрового кода, содержащего всю необходимую идентификационную информацию о защищаемом объекте. Такие технологии уже разработаны российскими учеными, правда, пока их практическая реализация выглядит достаточно сложной.

Другой пример – проблема защиты контрольно-кассовых машин (ККМ) от изменения недобросовестными пользователями содержащейся в их фискальной памяти информации о результатах торгового дня. К настоящему времени уже разработаны (кстати, нашими отечественными фирмами) криптографические способы защиты от такого рода злоупотреблений, позволяющие практически со стопроцентной надежностью проверять контрольным органам подобное вмешательство в фискальную память контрольно-кассовой машины. Более того, при оснащении контролируемых объектов соответствующими средствами связи можно было бы такой контроль делать дистанционно и незаметно для объекта контроля. Последнее удовольствие, правда, пока весьма дорогое.

Говоря о современном состоянии криптографии, следует сказать, что она переживает этап бурного развития. Этому способствуют два обстоятельства. Во-первых, стремительно развиваются ИТ-технологии, являющиеся основной сферой приложений для криптографии. Ее роль будет возрастать в связи с расширением ее областей приложения (цифровая подпись, аутентификация и подтверждение подлинности и целостности электронных документов, безопасность электронного бизнеса, защита информации, передаваемой через интернет и др.). Знакомство с криптографией потребуется каждому пользователю электронных средств обмена ин-

формацией. Во-вторых, криптография, наконец, вышла в открытый мир, и в ней нашли поле приложения своих сил большое количество талантливых ученых.

К числу наиболее интересных и значительных достижений современности в криптографии, безусловно, следует отнести изобретение американскими учеными Диффи и Хеллманом в 70-х гг. прошлого столетия нового принципа построения шифров, не требующего организации довольно трудоемкого процесса постоянного защищенного снабжения связывающихся сторон секретными ключами. В новом способе эти ключи вырабатываются с помощью специальной процедуры обмена некоторыми данными по обычным открытым каналам связи. К сожалению, идея этого способа основана на глубоких математических результатах и не может быть доступно изложена в популярной форме. Вместе с тем, хотя этот способ уже широко используется на практике, некоторые основополагающие математические гипотезы, на которых он основан, пока строго не доказаны. Это создает определенную тревогу с точки зрения надежности такого рода шифровальных систем.

Еще одним направлением исследований, сулящим большие перспективы для криптографии, являются работы в области так называемой «квантовой криптографии», основанной уже на результатах теоретической физики. И хотя на пути практической реализации этого направления встретились серьезные трудности научно-технического характера, специалисты достаточно обоснованно надеются на возможность их преодоления, и тогда мы станем свидетелями новых крупных успехов в теоретической и практической криптографии.

В заключение один конкретный пример использования криптографии в банковской сфере. Речь идет об автоматизированных платежных системах с дистанционным управлением банковским счетом на базе общедоступных (открытых) сетей связи (пластиковые карты плюс интернет, системы мобильной связи и т.п.). Здесь такие важные функции платежных систем, как защита передаваемых по каналам общей связи платежных поручений, юридическая значимость электронных транзакций, неотказуемость участников системы от совершенных операций должны базироваться исключительно на криптографических средствах защиты.

Таким образом, криптографические методы, характеризующиеся большими возможностями по обеспечению конфиденциальности и достоверности информации, а также по обеспечению «неотказуемости» от авторства в процессе обмена информацией, находят себе применение во все более и более многочисленных приложениях. Можно предположить в связи с этим, что определенное знакомство с криптографией, ее методами и возможностями в недалеком будущем потребуется каждому пользователю электронных средств обработки и обмена информацией. Более того, оптимисты криптографии считают, что криптография рано или поздно станет «третьей грамотностью» наряду со «второй» грамотностью – владением компьютером и информационными технологиями.

ЛИТЕРАТУРА

1. *Соболева Т.А.* Тайнопись в истории России. М.,1994.
2. *Жельников В.* Криптография от папируса до компьютера. М. 1996.
3. *Brassard G.* Modern Kryptology. Springer-Verlag, 1988.
4. *Саломаа А.* Криптография с открытым ключом. М.,1996.
5. *Schneier B.* Applied Cryptography. John Wiley & Sons, 1996.
6. *Н.А. Молдовян, А.А. Молдовян, М.А. Еремеев* Криптография: от примитивов к синтезу алгоритмов. – СПб.: БХВ-Петербург, 2004.
7. *Иванов М.А.* Криптографические методы защиты информации в компьютерных системах и сетях. – М.: КУДИЦ-ОБРАЗ, 2001
8. *Белов Е.Б., Зубов А.Ю., Погорелов Б.А., Проскурин Г.В., Черемушкин А.В., Шанкин Б.П., Шурупов А.Н.* Криптографические методы защиты информации. Учебно-методическое пособие. Ч.2.М.: ИКСИ, 2003.
9. *Столингс В.* Криптография и защита сетей. Принципы и практика. 2-е изд. – М.: Вильямс, 2001.
10. *Конеев И.Р., Беляев А.В.* Информационная безопасность предприятия. – СПб.: БХВ-Петербург, 2003.
11. Отечественный стандарт цифровой подписи ГОСТ Р 34.10-2001.
12. Федеральный закон Российской Федерации «Об электронной цифровой подписи».

СОДЕРЖАНИЕ

CONTENTS

Учебное издание

Павел Павлович Бескид
Татьяна Михайловна Татарникова

КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ
Часть 1
Основы криптографии

Учебное пособие

Редактор О.С. Крайнова
Компьютерная верстка Н.И. Афанасьевой

ЛР № 020309 от 30.12.96

Подписано в печать 10.09.10. Формат 60×90 1/16. Гарнитура Times New Roman.
Бумага офсетная. Печать офсетная. Усл. печ. л. 6,0. Тираж 250 экз. Заказ № 32/10
РГГМУ, 195196, Санкт-Петербург, Малоохтинский пр., 98.
ЗАО «НПП «Система», 197045, Санкт-Петербург, Ушаковская наб., 17/1.

70-40