ФГОС 3+

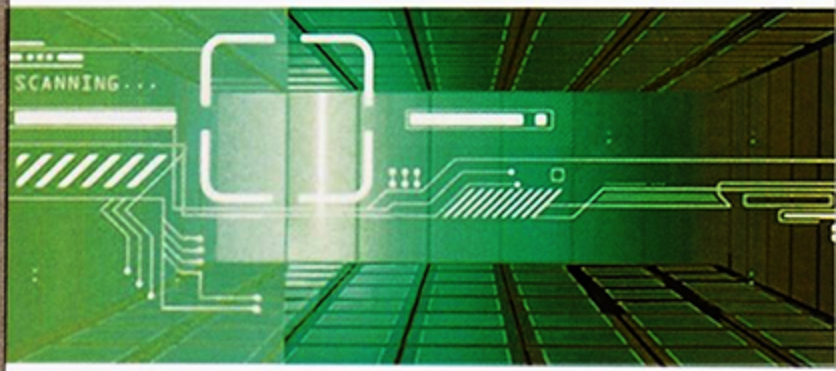
БАКАЛАВРИАТ
МАГИСТРАТУРА

А.В. Бабаш, Е.К. Баранова

КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

BOOK.ru
ONLINE МАТЕРИАЛЫ

Учебник



КНОРУС

Б А К А Л А В Р И А Т

ФГОС 3+

А.В. Бабаш, Е.К. Баранова

КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

Рекомендовано
УМО по образованию
в области прикладной информатики
в качестве **учебника**
для студентов высших учебных заведений,
обучающихся по направлению «Прикладная информатика»

BOOK.ru
ЭЛЕКТРОННО-БИБЛИОТЕЧНАЯ СИСТЕМА
КНОРУС • МОСКВА • 2016

УДК 681.3(075.8)
ББК 32.81я73
Б24

Рецензенты:
М.Г. Дмитриев, главный научный сотрудник Института системного анализа РАН, д-р физ.-мат. наук, проф.,
П.Б. Хорев, проф. Национального исследовательского университета МЭИ, канд. техн. наук

Авторский коллектив:
А.В. Бабаш, проф. НИУ «Высшая школа экономики»,
Е.К. Баранова, доц. НИУ «Высшая школа экономики»

Бабаш А.В.

Б24 **Криптографические методы защиты информации** : учебник / А.В. Бабаш, Е.К. Баранова. — М. : КНОРУС, 2016. — 192 с. — (Бакалавриат).

ISBN 978-5-406-04766-8
DOI 10.15216/978-5-406-04766-8

Рассмотрены основные вопросы криптографической защиты информации: математические основы криптографии; элементы истории развития криптографии; алгоритмы наиболее распространенных современных симметричных и асимметричных систем шифрования и современные области их применения. Учитывая специфику студентов гуманитарного профиля, доступно излагается необходимый математический аппарат криптографии, не прибегая в некоторых случаях к полным доказательствам теорем, приводя лишь их схемы. Каждая глава организована таким образом, чтобы студент мог изучать книгу самостоятельно, для чего в конце глав приводятся тесты, примеры и контрольные задания, выполнение которых дает возможность проконтролировать качество усвоения материала по теме.

Соответствует ФГОС ВО 3+.

Для студентов бакалавриата и магистратуры высших учебных заведений, где читается дисциплина «Криптографические методы защиты информации», а также обучающихся по специальностям, включающим в себя компонент по криптографической защите информации.

УДК 681.3(075.8)
ББК 32.81я73

Бабаш Александр Владимирович
Баранова Елена Константиновна
КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

Сертификат соответствия № РОСС RU. АЕ51. Н 16604 от 07.07.2014.

Изд. № 8568. Подписано в печать 01.07.2015. Формат 60×90/16.
Гарнитура «NewtonC». Печать офсетная.
Усл. печ. л. 12,0. Уч.-изд. л. 9,3. Тираж 500 экз.

ООО «Издательство «КноРус».
117218, г. Москва, ул. Кедрова, д. 14, корп. 2.
Тел.: 8-495-741-46-28.
E-mail: office@knorus.ru http://www.knorus.ru

Отпечатано в ОАО «Областная типография „Печатный двор“».
432049, г. Ульяновск, ул. Пушкарева, 27. E-mail: ulpd@mail.ru

IISBN 978-5-406-04766-8

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ

Курсы по криптографической защите информации в настоящее время читаются практически во всех ведущих университетах как в России, так и за рубежом. Зачастую их включают в программу математических факультетов или факультетов вычислительной техники. Однако в последние годы многие гуманитарные университеты видят необходимость включения этой дисциплины в свои учебные программы. Действительно, отдельный курс по криптографии необходим студентам всех упомянутых направлений, но подготовка специалистов на этих факультетах, как и цели, стоящие перед ними, различны. Некоторым из них достаточно краткого обзора существующих криптографических алгоритмов и умения их применять, в то время как другим необходимы глубинные математические знания, позволяющие проникнуть в суть криптопреобразований, чтобы подвести их к переднему краю современных исследований в области создания этих алгоритмов. Следовательно, существует практическая необходимость в учебнике, начинающемся с изложения базовых математических преобразований, используемых в криптографии, и ведущем студентов через общие понятия и методы к пониманию сути криптографических алгоритмов.

В настоящее время существует множество книг, которые дают общее представление об основных направлениях современной криптографии. Лучшие из них — книги Найджела Смарта (*N. Smart*)[1] и Брюса Шнайера (*B. Schneier*)[2] переведены на русский язык. Однако студентам гуманитарных факультетов для изучения этих мировых криптографических бестселлеров не хватает достаточной подготовки и особенно математических знаний.

Главная цель предлагаемой читателям книги — дать знания в области криптографической защиты информации, а также, учитывая специфику студентов гуманитарного профиля, доступно изложить необходимый математический аппарат криптографии, не прибегая в некоторых случаях к полным доказательствам теорем, приводя лишь их схемы. Каждая глава организована таким образом, чтобы студент мог изучать книгу самостоятельно, для чего в конце глав приводятся тесты, примеры и контрольные задания, выполнение которых дает возможность проконтролировать качество усвояемости материала по теме.

---

[1] *Смарт Н.* Криптография. М. : Техносфера, 2006.

[2] *Шнайер Б.* Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М. : Триумф, 2002.

Первая глава книги посвящена рассмотрению математических основ криптографии, что позволяет подготовить читателей к пониманию и адекватному восприятию материала остальных глав книги.

Ретроспективный взгляд на историю развития криптографии позволяет читателям не только познакомиться с классическими шифрами замены и перестановки, но и дать представление о тех эволюционных процессах, которые происходили в криптографии на протяжении веков.

В отдельных главах рассматриваются основные требования, предъявляемые к методам шифрования информации; типовые задачи криптоанализа; теоретическая и практическая стойкость шифров. Приводится подробное описание некоторых симметричных криптографических алгоритмов.

Для понимания математических преобразований, составляющих основу современных методов асимметричного шифрования, в отдельной главе приводятся базовые понятия модулярной арифметики и вычислений в конечных полях. Рассматриваются алгоритмы наиболее распространенных современных асимметричных систем шифрования.

Отдельная глава посвящена принципам построения алгоритмов электронной подписи и функций хэширования, а также вопросам управления криптографическими ключами.

Книга предназначена для студентов высших учебных заведений, где читается дисциплина «Криптографические методы защиты информации», а также обучающихся по специальностям, включающим компонент по криптографической защите информации. Материал книги адаптирован под магистерскую программу специализации «Управление информационной безопасностью» НИУ ВШЭ. Авторами книги подготовлен практикум по указанной дисциплине[1], который дополняет предлагаемый читателям теоретический курс по криптографии.

[1] *Баранова Е.К., Бабаш А.В.* Криптографические методы защиты информации. Лабораторный практикум : учеб. пособие (+ CD-ROM). М. : КНОРУС, 2015.

ГЛАВА 1

МАТЕМАТИЧЕСКИЕ ОСНОВЫ КРИПТОГРАФИИ

1.1. Операции над множествами

Под множеством понимают совокупность объектов любой природы, обладающих некоторым общим свойством.

Объекты, объединенные одним общим свойством, называют элементами множества и обозначают a, b, c, ... x, y, z. Множества обозначают A, B, C, ... X, Y, Z. Запись $a \in A$ означает, что элемент «a» принадлежит множеству A, $b \notin A$ означает, что элемент «b» не принадлежит множеству A.

Множество, число элементов которого, конечно, называют конечным и бесконечным в противном случае.

Бесконечные множества разделяются на счетные и несчетные. Если элементы бесконечного множества можно пронумеровать с помощью натурального ряда чисел, то оно называется счетным, и несчетным в противном случае. Так, множество четных чисел — счетное, множество действительных чисел — несчетное.

Если каждый элемент множества A является одновременно и элементом множества B, то множество A называется подмножеством (частью) множества B и обозначается $A \subseteq B$.

Если $A \subseteq B$ и $B \subseteq A$, то множества A и B называются равносильными (совпадающими) и обозначаются $A = B$.

Множество, не содержащее ни одного элемента, называется пустым и обозначается символом $\varnothing$. Пустое множество считают конечным множеством и подмножеством любого множества.

Любое множество A есть одновременно и подмножество самого себя. Такое подмножество A и пустое множество $\varnothing$ называют несобственными подмножествами множества A, в отличие от всех других подмножеств, которые называют собственными.

Примеры

Пусть $A = \{a_1, a_2, a_3\}$. Подмножества $\{a_1, a_2, a_3\}$ и $\varnothing$ — несобственные подмножества A. Собственные: $\{a_1\}$, $\{a_2\}$, $\{a_3\}$, $\{a_1, a_2\}$, $\{a_1, a_3\}$, $\{a_2, a_3\}$. Порядок выписанных элементов в записи множеств через его элементы не важен.

Число подмножеств любого конечного множества, содержащего «n» элементов, равно 2^n.

В определении операций над множествами фиксируют некоторое множество «U» и рассматривают операции над его подмножествами. Изначальное множество U называют универсальным.

На подмножествах A, B, C выбранного множества U определены следующие операции.

Объединением, или суммой подмножеств A и B, называется множество C, элементы которого принадлежат хотя бы одному из подмножеств A или B.

$$C = A \cup B = \{c_i : c_i \in A \text{ или } c_i \in B\}.$$

Пересечением подмножеств A и B называется подмножество C, элементы которого принадлежат как подмножеству A, так и подмножеству B.

$$C = A \cap B = \{c_i : c_i \in A \text{ и } c_i \in B\}.$$

Дополнением $\overline{A}$ подмножества A называется множество, элементы которого принадлежат универсальному множеству U и не принадлежат A.

$$C = \overline{A} = \{c_i : c_i \in U \text{ и } c_i \notin A\}.$$

Данные операции обладают следующими свойствами:

идемпотентность

$A \cup A = A$

$A \cap A = A$

коммутативность

$A \cup B = B \cup A$

$A \cap B = B \cap A$

ассоциативность

$(A \cup B) \cup C = A \cup (B \cup C)$

$(A \cap B) \cap C = A \cap (B \cap C)$

дистрибутивность

$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

законы де Моргана

$\overline{A \cup B} = \overline{A} \cap \overline{B}$

$\overline{A \cap B} = \overline{A} \cup \overline{B}$

и

$A \cup U = U$

$A \cap \varnothing = \varnothing$

$A \cap U = A$

$A \cup \varnothing = A$

$A \cup \overline{A} = U$

$A \cap \overline{A} = \varnothing$

$\overline{\overline{A}} = A$

$\overline{\varnothing} = U$

$\overline{U} = \varnothing$

$A \subset B$ равносильно $\overline{B} \subset \overline{A}$.

Указанные соотношения обладают свойствами двойственности: если в одной из формул поменять одновременно местами символы $\cup$ и $\cap$, U и $\varnothing$, $\subset$ и $\supset$, то получим другую формулу из этого списка.

Порядок выполнения операций следующий:

— дополнение (—);

— пересечение ($\cap$);

— объединение ($\cup$).

Названные операции и свойства к ним могут быть проиллюстрированы диаграммами Эйлера-Венна (рис. 1.1).

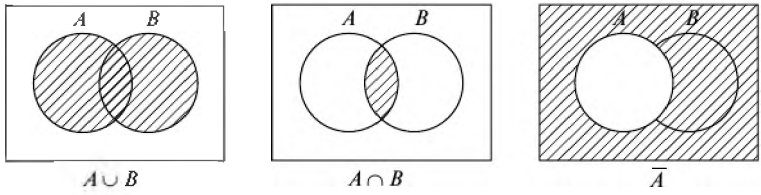


Рис. 1.1. Диаграммы Эйлера-Венна, иллюстрирующие выполнение операций объединения, пересечения, дополнения

Абстрактная алгебраическая система подмножеств некоторого универсального множества с введенными для них операциями объединения, пересечения, дополнения, обладающая перечисленными выше свойствами, образует Булеву алгебру.

К операциям над множествами относятся также следующие.

Разность множеств $A \setminus B$ — множество, состоящее из элементов множества A и не принадлежащих множеству B.

$$C = A \setminus B = \{c_i : c_i \in A \text{ и } c_i \notin B\}.$$

Очевидно, что справедлива формула $\overline{C} = A \setminus B = A \cap \overline{B}$.

Симметрическая разность $A \Delta B = (A \setminus B) \cup (B \setminus A)$.

Эти операции можно проиллюстрировать на диаграммах Эйлера-Венна (рис. 1.2).

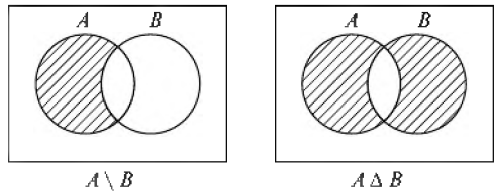


Рис. 1.2. Диаграммы Эйлера-Венна, иллюстрирующие разность множеств и симметрическую разность

Декартово (прямое) произведение множеств A и B: $A \times B = C$.

Декартовым произведением $A \times B$ является множество C всех упорядоченных пар $<a_i, b_j>$, где $a_i \in A$, $b_j \in B$, т.е.

$$C = A \times B = \{<a_i, b_j> : a_i \in A \text{ и } b_j \in B\}.$$

Иллюстрацией Декартова произведения множеств $A = \{a_1, a_2\}$ и $B = \{b_1, b_2, b_3\}$ является рис. 1.3.

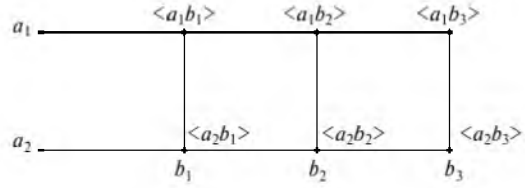


Рис. 1.3. Иллюстрация Декартова произведения множеств $A = \{a_1, a_2\}$ и $B = \{b_1, b_2, b_3\}$

В общем случае Декартовым произведением множеств $A_1, A_2, \ldots A_n$ называется множество

$$A_1 \times A_2 \times \ldots \times A_n = \{<a_1, a_2, \ldots a_n> : a_1 \in A_1, a_2 \in A_2 \ldots a_n \in A_n\}.$$

Приведем несколько упражнений, помогающих усвоить приведенные выше понятия.

Упражнения

1) Пусть заданы три числовых множества $A = \{2, 3, 4, 10\}$, $B = \{1, 2, 10, 12\}$, $C = \{1, 9, 10\}$. Требуется указать элементы множеств

$$\text{а) } A \cap B \cup B \cap C = D. \quad \text{б) } (A \cup C) \setminus (B \cap A) = E.$$

Множество «D» есть объединение двух множеств $A \cap B$ и $B \cap C$, что следует из порядка выполнения действий.

$$A \cap B = \{2, 10\},\ B \cap C = \{1, 10\} \text{ и } D = \{1, 2, 10\}.$$

Множество «E» есть разность между объединением $A \cup C$ и пересечением $B \cap A$.

$$A \cup C = \{1, 2, 3, 4, 10, 12\},\ B \cap A = \{2, 10\} \text{ и } E = \{1, 3, 4, 12\}.$$

2) Пусть множество A состоит из точек $M(x, y)$ плоскости, для которых $|x| \le 3$ и $|y| \le 5$, множество B — из точек плоскости, для которых $x^2 + y^2 \le 25$, C — из точек плоскости, для которых $x > 0$. Требуется изобразить множество $A \cap B \setminus C$.

Как следует из условия, множество A есть прямоугольник, B — круг, C — полуплоскость. Решение приведено на рис. 1.4.

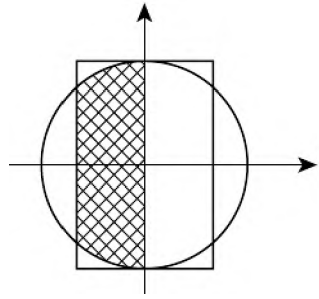


Рис. 1.4. Иллюстрация решения для упражнения 2

$A \cap B$ — «обрезанный» прямоугольник, обведенный на рисунке жирной линией.

$A \cap B \setminus C$ — множество точек, полученное удалением из $A \cap B$ точек полуплоскости $x > 0$. Результат изображен на рис. 1.4 штриховкой.

3) На диаграмме Эйлера-Венна убедиться в справедливости формул:

$$A \cup A \cap B = A \text{ и } (A \cup B) \cap A = A.$$

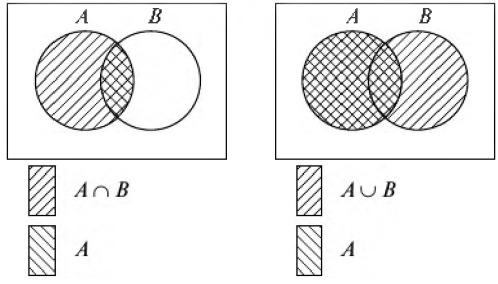


Рис. 1.5. Иллюстрация решения для упражнения 3

Данные формулы называют формулами поглощения, т.к. $A \cap B \subset A$ в первой формуле и $A \subset A \cup B$ во второй. Формулы поглощения помогают в преобразованиях, упрощающих выражения, задающие некоторые множества.

Помимо формул поглощения в преобразованиях использовались формулы склеивания $A \cup \overline{A} \cap E = A \cup E$ и $\overline{A} \cup A \cap E = \overline{A} \cup E$.

1.2. Отображение множеств

Если каждому элементу $x \in X$ поставлен в соответствие некоторый элемент $y \in Y$, то говорят, что определено отображение f множества X во множество Y. Обозначают $y = f(x)$. Элемент y есть образ элемента x при данном отображении f, x — один из прообразов элемента y. Множество всех прообразов y обозначают $f^{-1}(y)$, а $x \in f^{-1}(y)$.

Частным случаем отображения множества X в множество Y является отображение множества X на множество Y. Отображение f множества X в Y является отображением множества X на Y, если каждому элементу $y \in Y$ был поставлен в соответствие какой-либо элемент $x \in X$ при данном отображении f. Такое соотношение называется сюръективным, т.е. если каждый элемент множества y имеет прообраз, то отображение f сюръективно.

Пусть $X = \{a, b, c, d\}$, $Y = \{2, 4, 6\}$. Зададим отображения f_1 и f_2 так:

$$f_1:\ a \to 2 \qquad f_2:\ a \to 2$$
$$b \to 4 \qquad\qquad b \to 2$$
$$c \to 4 \qquad\qquad c \to 6$$
$$d \to 6 \qquad\qquad d \to 6,$$

т.е.

$$f_1(a) = \{2\} \quad f_1^{-1}(2) = \{a\} \quad f_2(a) = \{2\} \quad f_2^{-1}(2) = \{a, b\}$$
$$f_1(b) = \{4\} \quad f_1^{-1}(4) = \{b, c\} \quad f_2(b) = \{2\} \quad f_2^{-1}(4) = \{\varnothing\}$$
$$f_1(c) = \{4\} \quad f_1^{-1}(6) = \{d\} \quad f_2(c) = \{6\} \quad f_2^{-1}(6) = \{c, d\}.$$
$$f_1(d) = \{6\} \qquad\qquad f_2(d) = \{6\}$$

Отображение f_1 X в Y является сюръективным, т.е. отображением X на Y, т.к. каждый элемент множества Y имеет прообраз. Отображение f_2 несюръективно, элемент «4» не имеет прообраза.

Отображение X в Y называется инъективным, если для каждого элемента $y \in Y$ существует не более одного прообраза. Приведенные выше отображения f_1 и f_2 не являются инъективными.

$$X = \{x_1, x_2, x_3\} \qquad Y = \{y_1, y_2, y_3, y_4\} \qquad f_3:\ x_1 \to y_1$$
$$x_2 \to y_2$$
$$x_3 \to y_4$$

Отображение f_3 — инъективно.

Если отображение f сюръективно и инъективно, оно называется биективным (взаимно однозначным соответствием).

Очевидно, биективное отображение между конечными множествами X и Y возможно только в случае, когда число элементов этих множеств совпадает.

Примером биективного отображения для бесконечных множеств может служить отображение f, установленное между множеством натурального ряда чисел $A = \{1, 2, 3, \ldots n, \ldots\}$ и множеством четных положительных чисел $B = \{2, 4, 6, \ldots\}$ по типу $n \leftrightarrow 2n$.

На рисунке 1.6 показана возможность установления биективного отображения между множеством Z точек полуокружности и множеством X точек открытого отрезка (a, b), а также между множеством Z и множеством Y точек прямой — множеством Y.

$z, z_1 \in Z;$ Множества X, Y, Z — несчетные

$x, x_1 \in X;$

$y, y_1 \in Y.$

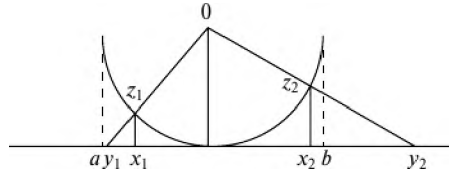


Рис. 1.6. Иллюстрация возможности установления биективного отображения между множеством Z точек полуокружности и множеством X точек открытого отрезка (a, b), а также между множеством Z и множеством Y точек прямой — множеством Y

Примеры

1) Установить биективное отображение между множеством $A = \{1, 6, 11, 16, 21, \ldots\}$ и натуральным рядом чисел.

Очевидно, это можно сделать, поставив в соответствие элементу натурального ряда «n» $a_n = 1 + 5(n - 1) \in A$, т.е. $n \leftrightarrow 1 + 5(n - 1)$.

2) Установить биективное отображение между множеством точек плоскости и множеством точек сферы, из которой выброшена одна точка.

Очевидно, это можно сделать геометрически (рис. 1.7).

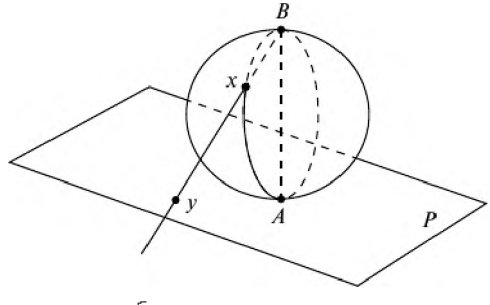


Рис. 1.7. Биективное отображение между множеством точек плоскости и множеством точек сферы, из которой выброшена одна точка

Обозначим множество точек плоскости P, множество точек сферы — M, точка A выброшена из сферы, $x \in M$, $y \in P$.

Чтобы установить биективное отображение между M и P, достаточно соединить точку B лучом с точкой «x» и получить соответствующую точку «y», или точку B соединить с точкой «y» и получить соответствующую точку «x», т.е. «x» $\leftrightarrow$ «y».

1.3. Мощность множеств

Два множества называются количественно эквивалентными (или просто эквивалентными), если между ними можно установить биективное отображение.

Исходя из этого определения, можно дать другую формулировку счетного множества: счетным называется множество, эквивалентное натуральному ряду чисел.

Очевидно, что справедливы следующие утверждения.

1. Конечные множества эквивалентны тогда и только тогда, когда они содержат одинаковое число элементов.
2. Два множества, порознь эквивалентные третьему, эквивалентны между собой.
3. Все счетные множества эквивалентны между собой.
4. Всякое множество, эквивалентное счетному множеству, счетно.

О двух эквивалентных множествах говорят, что они имеют одинаковую мощность.

Мощность — это то общее, что есть у эквивалентных множеств. Что общего имеют эквивалентные множества? Общим для них является число элементов. Мощность конечного множества есть число его элементов.

Все счетные множества имеют мощность, равную мощности натурального ряда чисел. Мощность натурального ряда чисел обозначается $\aleph_0$ — алеф-нуль.

Мощность континуума обозначается готической буквой C. Между этими мощностями существует следующая связь: $C = 2^{\aleph_0}$.

Как сравниваются мощности?

Рассмотрим два множества A и B. Если между ними можно установить биективное отображение, то мощности данных множеств равны. Если между множеством A и частью множества B можно установить биективное отображение, а между множеством B и частью A нельзя, то мощность множества A меньше мощности множества B.

Для конечных множеств это очевидно. Для бесконечных множеств оно также справедливо.

Мощность натурального ряда чисел — меньшая среди мощностей всех бесконечных множеств. Следующая по величине — мощность континуума. Пытаясь найти множество, мощность которого была бы промежуточной между мощностями континуума и натурального ряда чисел, Георг Кантор[1], основатель теории множеств, сформулировал так называемую гипотезу континуума — предложение, отрицающее множество промежуточной мощности. Попытки доказать это предложение привели к серьезным теоретическим исследованиям, связанным с пересмотром оснований математики.

Множества наибольшей мощности не существует, т.к. мощность множества подмножеств исходного множества всегда больше мощности исходного множества.

Примеры

1) Доказать, что если $A \setminus B$ эквивалентно $B \setminus A$, то A и B эквивалентны (рис. 1.8).

Решение: $A = (A \setminus B) \cup A \cap B \quad B = (B \setminus A) \cup A \cap B$.

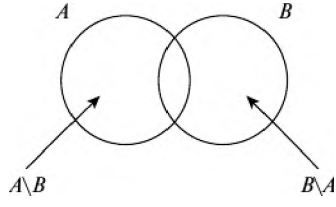


Рис. 1.8. Решение примера 1

Если $(A \setminus B)$ и $(B \setminus A)$ эквивалентны, то между элементами этих множеств существует биективное отображение. Элементы множества $(A \cap B)$ поставим в соответствие самим себе. Следовательно, между элементами множеств A и B существует биективное отображение, т.е. A и B эквивалентны, т.е. мощности множеств A и B одинаковы.

Сформулируем некоторые основные теоремы, справедливые для счетных множеств.

Теорема 1. Всякая часть счетного множества есть либо конечное, либо счетное множество.

[1] Георг Кантор (Georg Ferdinand Ludwig Philipp Cantor, 1845—1918) наиболее известен как создатель теории множеств, ставшей краеугольным камнем в математике. Кантор ввел понятие взаимно-однозначного соответствия между элементами множеств, дал определения бесконечного и вполне-упорядоченного множества.

Теорема 2. Сумма конечного или счетного числа конечных или счетных множеств есть счетное множество.

Теорема 3. Всякое бесконечное множество содержит счетное подмножество.

Теорема 4. Если M — несчетное множество, а $A \subset M$ есть конечное или счетное множество, то множества M и $M \setminus A$ эквивалентны.

Теорема 5. Присоединяя к некоторому бесконечному множеству M, счетному или несчетному, счетное или конечное множество A, получим множество $M \cup A$, эквивалентное множеству M.

Теорема 6. Всякое бесконечное множество M содержит часть $A \subset M$, эквивалентную всему множеству M.

Теорема 7. Множество всех пар натуральных чисел счетно. Под парой натуральных чисел понимают два натуральных числа, расположенных в определенном порядке.

Теорема 8. Множество всех рациональных чисел счетно.

Теорема 9. Множество всех конечных последовательностей, составленных из элементов данного счетного множества, есть счетное множество.

Теорема 10. Множество всех алгебраических чисел счетно.

Теорема 11. Множество континуума несчетно.

1.4. Отношения на множествах

Предложения «x — брат y», «$x < y$» выражают отношения между объектами некоторого множества.

Первое предложение свидетельствует, что два объекта «x» и «y» принадлежат общему классу — сыновья общих родителей. Второе предложение выражает относительный порядок в системе.

Об отношении можно говорить тогда, когда можно выделить множество объектов, на которых это отношение определено.

Приведенные примеры есть бинарные отношения (они выполняются для пары объектов). Тернарные отношения определены для трех объектов, n-арные — для n объектов.

Отношением A на множестве M называют подмножество A прямого произведения $M \times M$ множества M. Если $<x, y>$ входит в A, то обозначают xAy (или $<x, y> \in A$). Эта запись читается так: «x находится в отношении A с y».

Итак, отношением называется упорядоченная пара $<A, M>$, где $A \subseteq M \times M$. M — множество, на котором определено отношение; A —

множество пар, для которых это отношение определено (рассматриваем бинарные отношения).

Обратимся к примеру. Зададим отношение «x_i — победитель x_j» в шахматном турнире из пяти игроков x_1, x_2, x_3, x_4, x_5, турнир играл в один круг. Данные приведены в табл. 1.1.

Таблица 1.1

	x_1	x_2	x_3	x_4	x_5
x_1	0	1	1	0	0
x_2	0	0	1	1	1
x_3	0	0	0	0	0
x_4	1	0	0	0	0
x_5	0	0	1	0	0

$i^{\text{ая}}$ строка соответствует элементу x_i, $j^{\text{ый}}$ столбец элементу x_j, на их пересечении ставится 1, если отношение x_iAx_j выполнено, и 0, если нет. Так, единица, стоящая на пересечении $4^{\text{ой}}$ строки и $1^{\text{го}}$ столбца, соответствует тому, что игрок x_4 выиграл у игрока x_1, т.е. $<x_4Ax_1>$.

Итак, на множестве M (x_1, ... x_5) отношение «x_i — победитель y_j» задано матрицей

$$a_{ij} = \begin{cases} 1, \text{если выполнено } x_i A x_j \\ 0, \text{если не выполнено } x_i A x_j \end{cases}.$$

Такая матрица полностью задает отношение A на множестве M.

Прямое произведение $M \times M$ представлено двадцатью пятью элементами матрицы (табл. 1.1).

Если $a_{ij} \equiv 0$ $(i, j = \overline{1, n})$, то имеем пустое отношение, т.е. такое, которое не выполнено ни для какой пары x_ix_j. Если $a_{ij} \equiv 1$, имеем полное отношение, т.е. отношение, выполненное для всех пар. Единичная матрица E задает диагональное отношение, отношение равенства: $<x_iAx_j>$, если $x_i = x_j$.

Зададим отношение другим способом, а именно: элементы множества изобразим точками, проведем стрелку от x_i к x_j, если выполнено x_iAx_j, получим фигуру — ориентированный граф (рис. 1.9).

Точки x_1, x_2, x_3, x_4, x_5 — вершины графа, направленные линии — ребра графа.

Элементы теории графов рассмотрим во второй части данного пособия.

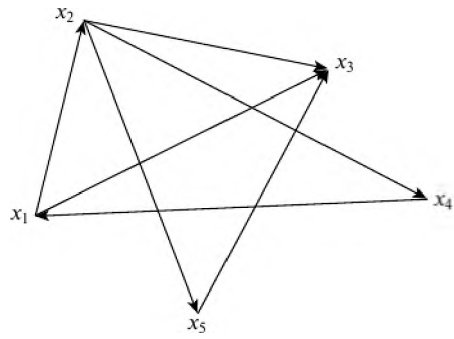


Рис. 1.9. Задание отношения в виде ориентированного графа

Свойства отношений

1) Отношение A рефлексивно, если оно выполнено между объектом и им самим, т.е. xAx.

2) Отношения «быть похожим», «быть знакомым» — рефлексивны. Отношение «быть братом» — нерефлексивно.

3) Если отношение A может выполняться лишь для несовпадающих объектов, то оно антирефлексивно, т.е. из xAy следует, что $x \neq y$.

4) Отношение A называется симметричным, если при выполнении xAy выполнено yAx.

5) Отношения «быть родственником», «быть похожим на» — симметричны.

6) Отношение A называется асимметричным, если из двух отношений xAy и yAx хотя бы одно не выполнено. Так, приведенный выше пример: отношение «x — победитель y» — асимметрично.

7) Справедлива теорема: если отношение асимметрично, то оно антирефлексивно.

8) Отношение называется транзитивным, если при выполнении xAy и yAz выполнено xAz.

9) Примером является отношение «быть больше (меньше)». Так, если $x < y$ и $y < z$, то $x < z$.

10) Отношение A называется антисимметричным, если оба соотношения xAy и yAx выполняются только тогда, когда $x = y$.

Эквивалентность. Если бинарное отношение A на множестве X рефлексивно (xAx для любого $x \in X$), симметрично (из xAy следует yAx), транзитивно (из xAy, yAz следует xAz), то оно называется отношением эквивалентности на множестве X.

Отношению эквивалентности соответствует разбиение множества X на классы. Множество X разбито на классы, если его можно представить в виде суммы непересекающихся подмножеств:

$$X = X_1 \cup X_2 \cup ... \cup X_n,$$

$$\text{где } X_k \subset X \ (k = \overline{1, n}) \text{ и } X_i \cap X_j = V \ (i \neq j) \ (i, j = \overline{1, n}).$$

Примеры

1) Множество X учащихся десятых классов некоторой школы разбивается на два класса: X_1 — учащиеся $10^{\text{А}}$ класса, X_2 — учащиеся $10^{\text{Б}}$ класса. $X = X_1 \cup X_2$ и $X_1 \cap X_2 = V$, т.к. нет учеников, обучающихся одновременно и в $10^{\text{А}}$, и в $10^{\text{Б}}$ классе.

Два элемента множества X эквивалентны, если они принадлежат одному и тому же классу.

Каждая пара учащихся $10^{\text{А}}$ класса — эквивалентные элементы множества X_1 (так же, как и пара $10^{\text{Б}} — X_2$).

Разбивая множество X на классы, мы осуществили сюръективное отображение множества всех учащихся X на множество Y, состоящее из двух элементов $y_1 = 10^{\text{А}}$, $y_2 = 10^{\text{Б}}$. Причем $f^{-1}(y_1) = X_1, f^{-1}(y_2) = X_2$.

2) Составление каталога по алфавиту. Множество всех книг в библиотеке X разбивается на конечное число классов — количество букв алфавита Y. Книги, начинающиеся с одной и той же буквы, принадлежат одному классу, и между любой парой таких книг существует отношение эквивалентности.

В то же время, составляя каталог по алфавиту, мы осуществляем сюръективное отображение множества всех книг в библиотеке X на множество букв алфавита Y.

Итак, отношение эквивалентности — рефлексивно, симметрично и транзитивно. Эти свойства являются необходимыми и достаточными условиями разбиения множества на классы.

Отношение A на множестве M называется *толерантностью*, если оно рефлексивно и симметрично.

Так, отношение «быть знакомым» соответствует определению толерантности.

Отношение A на множестве X называется *отношением порядка*, если оно транзитивно и антирефлексивно.

Отношение порядка характеризует соотношение объектов друг к другу по старшинству, по важности, оно не является симметричным. Отношение $x < y$ на множестве действительных чисел есть пример отношения порядка.

Множество, на котором задано отношение порядка, называется упорядоченным множеством. Понятие конечного упорядоченного множества совпадает с понятием конечной последовательности, состоящей из различных элементов. Простейшими примерами беско-

нечных упорядоченных множеств является множество всех целых чисел, множество рациональных чисел.

Заметим, что одно и то же множество можно упорядочить многими различными способами. Так, например, натуральные числа можно упорядочить «естественным образом»: 1, 2, 3, 4, ... Это же множество можно упорядочить так, что отдельно нечетные и отдельно четные числа расположены в порядке возрастания, а все нечетные числа считать предшествующими четным, т.е. 1, 3, 5, ... 2, 4, 6.

Биективное отображение «f» упорядоченного множества X на упорядоченное множество Y называют соответствием подобия или подобным соответствием, если оно сохраняет порядок.

Два упорядоченных множества называются подобными, или имеющими один и тот же порядковый тип, если одно из них можно подобно отобразить на другое. Так, два конечных упорядоченных множества X и Y, состоящих из одного и того же числа элементов, подобны между собой. Указанное выше биективное отображение между всей числовой прямой и интервалом (a, b) является соответствием подобия, и указанные множества подобны.

Заметим, что подобные множества имеют одну и ту же мощность.

Упорядоченное множество называется вполне упорядоченным, если каждое его непустое подмножество содержит первый элемент. Так, все конечные упорядоченные множества — вполне упорядочены. Примером бесконечного вполне упорядоченного множества является множество всех натуральных чисел.

Тест к главе 1

1. ***Будет ли пустое множество V каким-либо подмножеством некоторого множества*:**
   1) будет собственным подмножеством;
   2) будет несобственным подмножеством;
   3) не будет никаким подмножеством.
2. ***Что есть множество $A \setminus B$, если A — множество всех математических книг во всех библиотеках России, а B — множество всех книг в библиотеке университета по различным отделам науки и искусства*:**
   1) множество математических книг в России без математических книг в библиотеке университета;
   2) множество книг по искусству в библиотеке университета;

3) множество книг в библиотеке университета по искусству и науке, кроме математических.

3. ***Совпадают ли дистрибутивные законы Булевой алгебры и алгебры действительных чисел*:**
1) оба совпадают;
2) оба не совпадают;
3) один совпадает, другой — нет.

4. ***Вытекает ли из равенства*** $A \setminus B = C$, ***что*** $A = B \cup C$**:**
1) да;
2) нет;
3) вообще нет, но в частном случае да.

5. ***Есть ли законы для дополнений в алгебре действительных чисел*:**
1) да;
2) нет;
3) некоторые есть, некоторых нет.

6. ***Справедливы ли законы идемпотентности Булевой алгебры в алгебре действительных чисел*:**
1) справедливы;
2) несправедливы;
3) один справедлив, другой нет.

7. ***Можно ли поставить в соответствие единицу или ноль, соответственно, универсальному и пустому множеству, исходя из свойств операций*:**
1) можно;
2) единицу — можно, ноль — нет;
3) ноль — можно, единицу — нет.

8. ***Будет ли каждое из множеств*** A, B, C, D ***подмножеством другого, если*** A ***— множество действительных чисел,*** B ***— множество рациональных чисел,*** C ***— множество целых чисел,*** D ***— множество натуральных чисел*:**
1) да;
2) нет;
3) лишь некоторые из множеств являются подмножествами перечисленных множеств.

9. ***Задано отображение*** f ***множества*** X ***в*** Y. $X = \{x_1, x_2, x_3, x_4\}$ $Y = \{y_1, y_2, y_3\}$: $f(x_1) = y_1, f(x_2) = y_2, f(x_3) = y_2, f(x_4) = y_3$. ***Будет ли это отображение*** f**:**
1) сюръективно;
2) инъективно;
3) биективно.

10. ***Можно ли в любом бесконечном множестве выделить счетное подмножество*:**
1) нельзя;
2) можно;
3) можно, но не всегда.

11. ***Отношение «быть старше» : «x старше y» является*:**
1) рефлексивным;
2) симметричным;
3) асимметричным.

12. ***Отношение «x — победитель y» является*:**
1) антирефлексивным;
2) симметричным;
3) транзитивным.

13. ***Каково максимально возможное число классов, на которое можно разбить сумму трех пересекающихся множеств, не прибегая к произвольному делению отдельных областей на диаграммах Эйлера-Венна*:**
1) 3;
2) 5;
3) 7.

14. ***Если отношение A на множестве M рефлексивно, симметрично и транзитивно, можно ли разбить множество M на классы*:**
1) да;
2) нет;
3) можно, но не всегда.

15. ***Пусть на множестве M задано отношение A: «x знаком с y». Почему нельзя разбить множество M на классы*:**
1) отношение A не рефлексивно;
2) отношение A не симметрично;
3) отношение A не транзитивно.

16. ***Почему множество действительных чисел и множество натуральных чисел не являются подобными*:**
1) множество натуральных чисел неупорядочено;
2) множество действительных чисел неупорядочено;
3) нет биективного соответствия между множествами.

17. ***Почему множество M точек отрезка* [0, 1] *не является вполне упорядоченным множеством*:**
1) M не упорядочено;
2) не все подмножества M содержат первый элемент;
3) ни одно из подмножеств M не содержит первый элемент.

18. ***Сколько несобственных подмножеств имеет конечное множество, состоящее из n элементов*:**

1) 1;
2) 2;
3) n.

19. ***Сколько собственных подмножеств имеет конечное множество*** $X = \{x_1, x_2, \ldots, x_n\}$**:**

1) $n - 1$;
2) $n \times n = n^2$;
3) $2^n - 2$.

20. ***В каком порядке нужно производить операции, преобразовывая формулу*** $S = A \cap B \cup C \cap \overline{B} \cup A$**:**

1) $\left(A \cap (B \cup C) \cap \overline{B}\right) \cup A$;
2) $(A \cap B) \cup \left(C \cap (\overline{B} \cup A)\right)$;
3) $(A \cap B) \cup (C \cap \overline{B}) \cup A$.

21. ***Пусть*** $n(A \cup B)$ ***— мощность множества, являющегося объединением конечных множеств*** A ***и*** B**,** $m_1 = n(A \cup B)$**, если множества пересекаются, т.е.** $A \cap B \neq \varnothing$ ***и*** $m_2 = n(A \cup B)$**, если** $A \cap B = 0$**. Равны ли мощности** m_1 ***и*** m_2**:**

1) $m_1 = m_2$;
2) $m_1 > m_2$;
3) $m_1 < m_2$.

22. ***Мощность какого множества больше*** X ***или*** Y**, если** X **— исходное конечное множество,** Y **— множество подмножеств множества** X**:**

1) мощность X больше мощности Y;
2) мощность X меньше мощности Y;
3) мощность X равна мощности Y.

23. ***Существует ли среди бесконечных множеств множества наименьшей и наибольшей мощности*:**

1) существуют множества как наибольшей, так и наименьшей мощности;
2) существует множество наибольшей, а наименьшей мощности нет;
3) существует множество наименьшей, а наибольшей мощности нет.

24. ***Является ли сюръективное отображение инъективным*:**

1) сюръективное отображение всегда инъективно;
2) сюръективное отображение неинъективно;

3) сюръективное отображение может быть инъективным, но может и не быть им.

25. ***Всегда ли биективное отображение сюръективно*:**
1) всегда;
2) никогда;
3) может быть сюръективным, но может и не быть им.

26. ***Когда сумма конечного или счетного числа конечных или счетных множеств является конечным множеством*:**
1) в случае конечного числа суммы счетных множеств;
2) в случае счетного числа суммы конечных множеств;
3) в случае конечного числа суммы конечных множеств.

27. ***Может ли конечное множество A содержать собственное подмножество, эквивалентное всему множеству A*:**
1) всегда содержит;
2) никогда не содержит;
3) иногда содержит, иногда нет.

28. ***Отсутствием какого из свойств отношений отличаются отношение толерантности от отношения эквивалентности*:**
1) рефлексивности;
2) симметрии;
3) транзитивности.

Таблица ответов на тест к главе 1

Номер вопроса	1	2	3	4	5	6	7
Правильный ответ	1	1	3	3	2	2	1
Номер вопроса	8	9	10	11	12	13	14
Правильный ответ	2	1	2	3	1	3	1
Номер вопроса	15	16	17	18	19	20	21
Правильный ответ	3	3	2	2	3	3	3
Номер вопроса	22	23	24	25	26	27	28
Правильный ответ	2	3	3	1	3	2	3

ГЛАВА 2

ЭВОЛЮЦИЯ СИММЕТРИЧНОГО ШИФРОВАНИЯ

2.1. Классические шифры

В истории криптографии, как в специфической области человеческой деятельности, выделяется три основных периода. Первый, наиболее продолжительный, — это эпоха «ручной криптографии». Ее начало теряется в глубокой древности, а окончание приходится на 30-е годы XX в. Криптография прошла путь от магического искусства до вполне прозаической прикладной специальности чиновников дипломатических и военных ведомств.

Второй период — создание и широкое внедрение в практику сначала механических, затем электромеханических и электронных устройств шифрования, организация целых сетей засекреченной связи. Его началом можно считать разработку Гилбертом Вернамом (*Gilbert Sandford Vernam*) в 1917 году схемы телеграфной шифровальной машины, использующей *одноразовую гамму* (рис. 2.1).

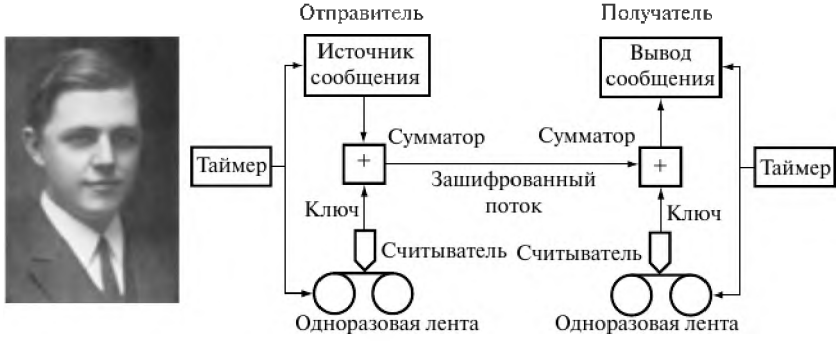


Рис. 2.1. Гилберт Вернам и его схема телеграфной шифровальной машины, использующей одноразовую гамму

К середине 70-х годов XX в. с развитием разветвленных коммерческих сетей связи, электронной почты и глобальных информационных систем на первый план вышли новые проблемы — проблемы снабжения ключами и проблемы подтверждения подлинности.

В 1976 году американские ученые Уитфилд Диффи (*Whitfield Diffie*) и Мартин Хеллман (*Martin Hellman*) предложили два новых принципа организации засекреченной связи без предварительного снабжения абонентов секретными ключами шифрования — принцип так называемого *открытого ключа шифрования* и принцип *открытого распределения ключей*. Этот момент можно считать началом нового периода в развитии криптографии. В настоящее время это направление современной криптографии очень интенсивно развивается.

Понятие «безопасность» охватывает широкий круг интересов как отдельных лиц, так и целых государств. Во все исторические времена существенное внимание уделялось проблеме информационной безопасности, обеспечению защиты конфиденциальной информации от ознакомления, кражи, модификации, подмены. Решением этих вопросов занимается *криптография*.

Рис. 2.2. Джон Валлис (*John Wallis*)

***Криптография* — тайнопись.** Термин ввел Джон Валлис (*John Wallis*, 1616—1703), английский математик, один из основателей и первых членов Лондонского королевского общества, профессор геометрии Оксфордского университета (1649).

Потребность шифровать и передавать шифрованные сообщения возникла очень давно. Так, еще в *V—VI* вв. до н.э. греки применяли специальное шифрующее устройство. По описанию Плутарха, оно состояло из двух цилиндрических стержней одинаковой длины и толщины. Один оставляли себе, а другой отдавали отъезжающему. Эти стержни называли *сциталами* (от греч. σκυτάλη — жезл) (рис. 2.3). При необходимости передачи сообщения длинную ленту папируса наматывали на сциталу, не оставляя на ней никакого промежутка.

Затем, оставляя папирус на сцитале, записывали на нем все, что необходимо, а написав, снимали папирус и без стержня отправляли адресату. Так как буквы оказывались разбросанными в беспорядке, то прочитать сообщение мог только тот, кто имел свою сциталу такой же длины и толщины, намотав на нее папирус.

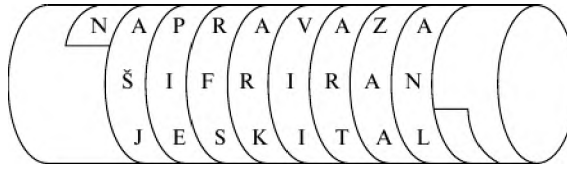


Рис. 2.3. Сцитала

Квадрат Полибия[1]. В Древней Греции (II в. до н.э.) был известен шифр, называемый «*квадрат Полибия*». Это устройство представляло собой квадрат 5×5, столбцы и строки которого нумеровали цифрами от 1 до 5. В каждую клетку записывалась одна буква (в греческом варианте одна клетка оказывалась пустой, а в латинском — в одну клетку помещали две буквы *I*, *J*). В результате каждой букве отвечала пара чисел по номеру строки и столбца.

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

13	34	22	24	44	34	15	42	22	34	43	45	32

Congito ergo sum — (лат.) «Я мыслю, следовательно, существую». (Р. Декарт).

Шифр Цезаря. В I в. н.э. Гай Юлий Цезарь во время войны с галлами, переписываясь со своими друзьями в Риме, заменял в сообщении первую букву латинского алфавита (*A*) на четвертую (*D*), вторую (*B*) — на пятую (*E*), наконец последнюю — на третью.

ABCDEFGHIJKLMNOPQRSTUVWXYZ
DEFGHIJKLMNOPQRSTUVWXYZABC

YHQL YLGL YLFL
Veni vidi vici (лат.) — «Пришел, увидел, победил».
(*Ю. Цезарь.* Донесение Сенату о победе над понтийским царем).

Шифр Цезаря относится к так называемому классу *моноалфавитных подстановок* и имеет множество модификаций.

Решетка Кардано. Широко известны шифры, принадлежащие к классу *перестановка*, в частности «*решетка Кардано*»[2]. Это прямоугольная карточка с отверстиями, чаще всего квадратная, которая при наложении на лист бумаги оставляет открытыми лишь некоторые его части. Число строк и столбцов на карточке четно. Карточка сделана

[1] Полибий (др. греч. Πολύβιος, 200—120 гг. до н.э.) — древнегреческий историк.

[2] Кардано Джероламо (лат. Hieronymus Cardanus, 1501—1576) — итальянский математик, философ и врач.

так, что при последовательном ее поворачивании каждая клетка лежащего под ней листа окажется занятой. Карточку поворачивают сначала вдоль вертикальной оси симметрии на 180°, а затем вдоль горизонтальной оси также на 180°. И вновь повторяют ту же процедуру (рис. 2.4).

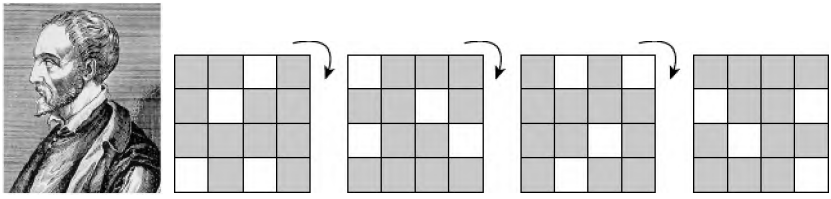


Рис. 2.4. Джероламо Кардано (лат. *Hieronymus Cardanus*). Решетка Кардано

Диск Альберти. Итальянец Леон Баттист Альберти (итал. *Leone Battista Alberti*, XVI в.) впервые выдвинул идею «двойного шифрования» — текст, полученный в результате первого шифрования, подвергался повторному зашифрованию. В трактате Альберти был приведен его собственный шифр, который он назвал «шифром, достойным королей». Он утверждал, что этот шифр недешифруем. Реализация шифра осуществлялась с помощью шифровального диска, положившего начало целой серии *многоалфавитных подстановок*. Устройство представляло собой пару дисков — внешний, неподвижный (на нем были нанесены буквы в естественном порядке и цифры от 1 до 4) и внутренний — подвижный — на нем буквы были переставлены. Процесс шифрования заключался в нахождении буквы открытого текста на внешнем диске и замену ее на соответствующую (стоящую под ней) букву шифрованного текста. После шифрования нескольких слов внутренний диск сдвигался на один шаг. Ключом данного шифра являлся порядок расположения букв на внутреннем диске и его начальное положение относительно внешнего диска (рис. 2.5).

Заметим также, что одна из первых в Европе книг, посвященных криптоанализу «Трактат о шифрах» (1466), написана Леоном Баттиста Альберти — итальянским ученым, гуманистом, писателем, одним из зачинателей новой европейской архитектуры и ведущим теоретиком искусства эпохи Возрождения. Своей работой он внес существенный вклад в развитие криптографии.

Таблица Виженера[1]. Неудобство рассмотренных выше шрифтов моноалфавитных подстановок очевидно, так как в случае использования

[1] Блез де Виженер (фр. Blaise de Vigenère 1523—1596) — французский дипломат, криптограф и алхимик, написал большой труд о шифрах. Квадратный шифр Виженера на протяжении почти 400 лет не был дешифрован, считался недешифруемым шифром.

стандартного алфавита таблица частот встречаемости букв алфавита позволяет определить один или несколько символов, а этого иногда достаточно для вскрытия шифра («Пляшущие человечки» Конан Дойля или «Золотой жук» Эдгара По). Поэтому использовались различные приемы для того, чтобы затруднить дешифрование, например таблица Виженера, которая представляет собой квадратную матрицу с числом строк и столбцов, равным количеству букв алфавита. Для шифрования по этой схеме используется таблица, где первая строка состоит из 26 букв латинского алфавита (ниже в примере приводится кириллическая таблица для шифра Виженера), а каждая последующая строка представляет собой циклический сдвиг предыдущей на одну позицию.

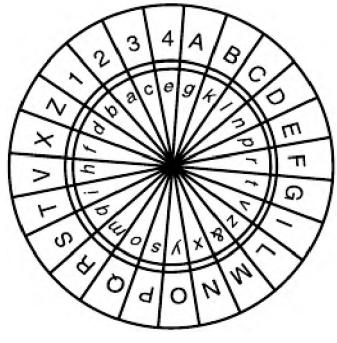


Рис. 2.5. Диск Альберти. Статуя Альберти в колоннаде Уффици (Флоренция)

Пароль или лозунг с повторениями выписывается буква к букве под исходным сообщением. Каждая буква зашифрованного текста берется из таблицы на пересечении соответствующей буквы исходного текста и столбца, соответствующего текущей букве лозунга. Подобный шифр получил название многоалфавитной замены. Кроме непосредственного увеличения стойкости шифр Виженера привнес в криптографию две качественно новые идеи. Во-первых, процесс шифрования стал зависеть в первую очередь от небольшого не известного третьей стороне слова — лозунга. 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, например, в шифре Цезаря.

Во-вторых, сама таблица, использованная Виженером, несет по своей сути первые намеки на идею цифрового шифрования. Если все буквы латинского алфавита пронумеровать по порядку от 0 до 25,

то процедура шифрования по такой таблице превратится в обычную операцию сложения по модулю 26.

Чтобы зашифровать какое-либо сообщение, выбирают слово — лозунг (например, «монастырь») и надписывают его над сообщением с необходимым повторением. Чтобы получить шифрованный текст, находят очередной знак лозунга, начиная с первого в вертикальном алфавите, а ему соответствующий знак сообщения в горизонтальном алфавите. На пересечении выделенных столбца и строки находим первую букву шифра. Очевидно, что ключом к такому шифру является используемый лозунг.

м о н а с т ы р ь м о н а с т ы р ь м о н

р а с к и н у л о с ь м о р е ш и р о к о

э о я к щ а п ы й ю й щ о в ч ф ш л ь ш ы

Таблица Виженера (для кириллического алфавита)

АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯ
БВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯА
ВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯАБ
ГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯАБВ
ДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯФБВГ
ЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯФБВГД
ЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯФБВГДЕ
ЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖ
ИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗ
ЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИ
КЛМНОПРСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙ
ЛМНОПРСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙК
МНОПРСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛ
НОПРСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМ
ОПРСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМН
ПРСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНО
РСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОП
СТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПР
ТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРС
УФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТ
ФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТУ
ХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТУФ
ЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТУФХ
ЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТУФХЦ
ШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧ
ЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШ
ЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩ
ЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬ
ЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫ
ЮЯАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭ
ЯАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮ

Одноразовый шифровальный блокнот. Примером нераскрываемого шифра может служить так называемый *одноразовый шифровальный блокнот* — шифр, в основе которого лежит та же идея, что в шифре Цезаря. Назовем *расширенным алфавитом* множество букв алфавита и знаков препинания {. , : ; ! ? ( ) — " < пробел >}, число символов расширенного кириллического алфавита в данном варианте будет равно 44. Занумеруем символы расширенного алфавита числами от 0 до 43. Тогда любой передаваемый текст можно рассматривать как последовательность $\{a_n\}$ чисел множества $A = \{0, 1, 2, ..., 43\}$.

Предположим, что имеем *ключ* — случайную последовательность $\{c_n\}$ чисел из множества A той же длины, что и передаваемый текст. Складывая по модулю 44 число из передаваемого текста a_n с соответствующим числом из множества ключа c_n:

$$a_n + c_n \equiv b_n \pmod{44},\ 0 \le b_n \le 43,$$

получим последовательность $\{b_n\}$ знаков шифрованного текста. Чтобы получить передаваемый текст, можно воспользоваться тем же ключом:

$$a_n \equiv b_n - c_n \pmod{44}, \quad 0 \le a_n \le 43,$$
$$\text{если} \quad b_n - c_n < 0, \quad \text{то} \quad a_n = b_n - c_n + 44.$$

У двух абонентов, находящихся в секретной переписке, имеются два одинаковых блокнота. В каждом из них на нескольких листах напечатана случайная последовательность чисел множества A. Отправитель свой текст шифрует указанным выше способом с помощью первой страницы блокнота. Зашифровав сообщение, он уничтожает использованную страницу и отправляет текст сообщения второму абоненту, получатель шифрованного текста расшифровывает его и также уничтожает использованный лист блокнота. Нетрудно видеть, что одноразовый шифр нераскрываем в принципе, так как символ в тексте может быть заменен любым другим символом и этот выбор совершенно случаен.

Одноалфавитный метод шифрования. Данный метод, пожалуй, самый древний из всех известных методов. В его основе лежит простой способ шифрования: отправитель и получатель зашифрованного документа заранее договариваются об определенном смещении букв относительно их обычного местоположения в алфавите. Например, для кириллицы, если смещение равно 1, то «А» соответствует букве «Б», «Б» — «В» и так далее, а когда алфавит подходит к концу, то начинают брать буквы из начала списка. И выходит, например, следующее: из слова «КОДИРОВАНИЕ» получается «ЛПЕЙСПГБОЙЖ».

Частным случаем данного метода является ранее рассмотренный шифр Цезаря. Очевидно, что произвольный шифр из класса одноалфавитных методов не является шифром Цезаря (если мощность алфавита текста равна n, то число шифров Цезаря равно n, а число всех одноалфавитных шифров равно $n!$). Однако и для таких методов легко предложить способы дешифрования, основанные на статистических свойствах шифрованных текстов, поскольку открытый и закрытый тексты имеют одинаковые статистические характеристики.

Шифрование методом перестановки символов. Суть этого метода заключается в том, что символы текста переставляются по определенным правилам, при этом используются только символы исходного (незашифрованного) текста. Перестановки в классической криптографии обычно получаются в результате записи исходного текста и чтения шифрованного текста по разным путям геометрической фигуры. Простейшим примером перестановки является запись исходного текста по строкам некоторой матрицы и чтение его по столбцам этой матрицы (рис. 2.6). Последовательность заполнения строк и чтения столбцов может быть любой и задается ключом. Таким образом, для матрицы размером 8×8 (длина блока 64 символа) возможно 1.6×10^9 ключей, что позволяет на современных компьютерах путем перебора дешифровать заданный текст. Однако для матрицы размером 16×16 (длина блока 256 символов) имеется 1.4×10^{26} ключей, и перебор их с помощью современных вычислительных средств весьма затруднителен.

1	И	Е	—	П
2	Е	Р	Е	С
3	О	В	А	Н
4	Т	А	Н	О
5	Ш	И	Ф	Р
6	В	К	О	Й
k_1/k_2	1	2	3	4

Открытый текст: «ШИФРОВАНИЕ_ПЕРЕСТАНОВКОЙ».
Матрица из четырех столбцов. Ключи: k_1 {5-3-1-2-4-6}; k_2 {4-2-3-1}.
Запись по строкам производится в соответствии с ключом k_1.
Чтение по столбцам в соответствии с ключом k_2.
Шифротекст: «ПСНОРЙЕРВАИК_ЕАНФОИЕОТШВ»

Рис. 2.6. Пример шифрования методом усложненной перестановки

Примером применения метода перестановки может быть также восьмиэлементная таблица (или граф), обладающая совокупностью маршрутов, носящих название маршрутов Гамильтона. Последовательность заполнения таблицы каждый раз соответствует нумерации ее элементов. Если длина шифруемого текста некратна числу элементов, то при последнем заполнении в свободные элементы заносится произвольный символ. Выборка из таблицы для каждого заполнения может выполняться по своему маршруту, при этом маршруты могут использоваться как последовательно, так и в порядке, задаваемом ключом. Наиболее сложные перестановки осуществляются по гамильтоновым путям, которых в графе может быть несколько. *Гамильтонов путь* — путь, содержащий каждую вершину графа ровно один раз. Необходимо отметить, что, например, для графа на рис. 2.7 из восьми вершин можно предложить несколько маршрутов записи открытого текста и несколько гамильтоновых путей для чтения криптограмм.

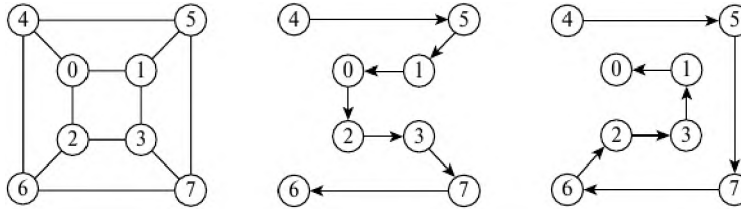


Рис. 2.7. Перестановки с использованием гамильтоновых путей на графе

Для методов перестановки характерны простота алгоритма, возможность программной реализации и низкий уровень защиты, так как при большой длине исходного текста в его зашифрованном варианте проявляются статистические закономерности ключа, что и позволяет его быстро раскрыть. Другой недостаток этих методов — легкое раскрытие, если удается направить в систему для шифрования несколько специально подобранных сообщений. Так, если длина блока в исходном тексте равна K символам, то для раскрытия ключа достаточно пропустить через шифровальную систему $K - 1$ блоков исходного текста, в которых все символы, кроме одного, одинаковы.

Шифрование инверсными символами (по дополнению до 255). Данный метод шифрования является частным случаем одноалфавитной замены в алфавите мощности 256. Суть метода заключается в замене символа ASCII-кодировки с номером i на символ с номером $255 - i$. Аналогично проводится и операция расшифрования.

Многоалфавитные методы шифрования. Многоалфавитное шифрование (многоалфавитная замена) заключается в том, что для последовательных символов шифруемого текста используются одноалфавитные методы с различными ключами. Например, первый символ заменяется по методу Цезаря со смещением 14, второй — со смещением 10 и так далее до конца заданного ключа. Затем процедура продолжается периодически. Более общей является ситуация, когда используется не шифр Цезаря, а последовательность произвольных подстановок, соответствующих одноалфавитным методам.

Более наглядным примером подобного шифрования является *метод гаммирования*. Данный способ преобразования заключается в том, что символы закрываемого текста последовательно складываются с символами некоторой специальной последовательности, именуемой гаммой. Такое преобразование иногда называют наложением гаммы на открытый текст.

Собственно, процедура наложения может осуществляться одним из двух способов.

1. Символы закрываемого текста и гаммы заменяются цифровыми эквивалентами, а затем складываются по модулю K, где K — количество символов алфавита

$$T_{ш} = (T_{о} \oplus T_{г}) \bmod K,$$

где $T_{ш}$ — шифротекст, $T_{о}$ — открытый текст, $T_{г}$ — гамма.

2. Символы текста и гаммы представляются в двоичных кодах, а затем каждая пара двоичных разрядов складывается по модулю 2.

Стойкость шифрования методом гаммирования определяется, главным образом, качеством гаммы, которое определяется двумя характеристиками: длиною периода и случайностью распределения символов по периоду. Длиною периода гаммы называется минимальное количество символов, после которого последовательность начинает повторяться. Случайность распределения символов по периоду означает отсутствие закономерностей между появлением различных символов в пределах периода.

Основные требования, которые предъявляются к методам шифрования информации

1. Сложность и трудоемкость процедур шифрования и расшифрования должны определяться в зависимости от степени секретности защищаемых данных.

2. Надежность закрытия должна быть такой, чтобы секретность не нарушалась даже в том случае, когда злоумышленнику известен способ закрытия.

3. Способ закрытия и набор используемых служебных данных (ключевых установок) не должны быть слишком сложными. Затраты на защитные преобразования должны быть приемлемые при заданном уровне сохранности информации.

4. Выполнение процедур прямого и обратного преобразования должно быть формальным и как можно проще.

5. Процедуры прямого и обратного преобразования не должны зависеть от длины сообщения.

6. Ошибки, возникающие в процессе преобразования, не должны распространяться по системе и вызывать потерю информации. Из-за появления ошибок передачи зашифрованного сообщения по каналам связи не должна исключаться возможность надежной расшифровки текста на приемном конце.

7. Избыточность сообщений, вносимая закрытием, должна быть как можно меньшей.

8. Объем ключа не должен затруднять его запоминание и пересылку.

Гистограмма текста. Одним из наиболее известных методов криптоанализа является изучение статистических характеристик шифрованных текстов. Графическое отображение совокупности частот встречаемости символов в тексте называют гистограммой этого текста. Предположим, что мы имеем дело с методом одноалфавитного шифрования. Зная частоту встречаемости букв в алфавите, можно предположить, какая буква была заменена на данную. Например, если часто встречаемая буква «О» заменена на редко встречающуюся букву «Щ», то в гистограмме шифрованного текста буква «Щ» встретится с той же частотой.

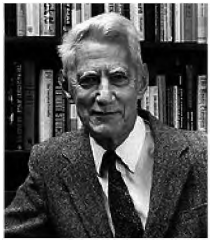
2.2. Основные понятия теории классических шифров

В 1963 г. Клод Шеннон[1] в книге «Работы по теории информации и кибернетике» (раздел «Теория связи в секретных системах») одним из первых ввел и систематически исследовал простую и естественную математическую модель шифра. Он рассматривал так называемые «се-

[1] Шеннон Клод Элвуд (англ. Claude Elwood Shannon, 1916—2001). К. Шеннону удалось решить фундаментальные проблемы в теоретической криптографии. Его работы стимулировали бурный рост научных исследований по теории информации и криптографии.

кретные системы», в которых смысл сообщения скрывается при помощи шифра или кода, но само шифрованное сообщение не скрывается и предполагается, что противник обладает любым специальным оборудованием, необходимым для перехвата и записи передаваемых сигналов.

В модели шифров по К. Шеннону рассматривается только дискретная информация, то есть считается, что сообщение, которое должно быть зашифровано, состоит из последовательности дискретных символов, каждый из которых выбран из некоторого конечного множества. Эти символы могут быть буквами или словами некоторого языка, амплитудными уровнями квантованной речи или видеосигнала. Ядром секретной системы является собственно шифр.



Алгебраическая модель шифра. Пусть X, K, Y — некоторые конечные множества, которые названы, соответственно, множеством открытых текстов, множеством ключей и множеством шифрованных сообщений (криптограмм). На прямом произведении $X \times K$ множеств X и K задана функция f: $X \times K \to Y (f(x, \chi) = y, x \in X, \chi \in K, y \in Y)$. Функции f соответствует семейство отображений $f\chi$: $X \to Y$, $\chi \in K$, каждое отображение задано так: для $x \in X$

$$f\chi(x) = f(x, \chi).$$

Таким образом, $f\chi$ — ограничение f на множестве $X \times \{\chi\}$. Здесь $\{\chi\}$ — множество, состоящее из одного элемента. Заметим, что задание семейства отображений $(f\chi)$ $\chi \in K, f\chi: X \to Y$ однозначно определяет отображение f: $X \times K \to Y, f(x, \chi) = f\chi(x)$.

Введенная четверка $A = (X, K, Y, f)$ определяет трехосновную универсальную алгебру, сигнатура которой состоит из функциональной единственной операции f.

ОПРЕДЕЛЕНИЕ. Введенная тройка множеств X, K, Y с функцией f: $X \times K \to Y$

$$A = (X, K, Y, f)$$

называется алгебраической моделью шифра, коротко — шифром, если выполнены два условия: 1) функция f — сюръективна (осуществляет отображение «на» Y); 2) для любого $\chi \in K$ функция $f\chi$ инъективна (образы двух различных элементов различны).

Из условия 2) данного определения вытекает, что $|X| \le |Y|$. Запись $f(x, \chi) = y$ называется уравнением шифрования. Имеется в виду, что от-

крытое сообщение x зашифровывается шифром A на ключе χ и получается шифрованный текст y. Уравнением расшифрования называют запись $f\chi - 1(y) = x(f - 1(y, \chi) = x)$, подразумевая, что шифрованный текст $y = f(x, \chi)$ расшифровывается на ключе χ и получается исходное открытое сообщение x. Для краткости в ряде случаев используют и более простые обозначения уравнений шифрования и расшифрования, а именно соответственно: $\chi x = y$ и $\chi - 1y = x$.

Требование инъективности отображений $(f\chi)$ $\chi \in K$ в определении шифра равносильно требованию возможности однозначного расшифрования криптограммы (однозначного восстановления открытого текста по известным шифрованному тексту и ключу). Требование же сюръективности отображения f не играет существенной роли, и оно обычно вводится для устранения некоторых технических, с точки зрения математики, дополнительных неудобств, то есть для упрощения изложения. Подчеркнем, что множество X названо множеством открытых текстов. Его можно понимать как множество текстов, возможных для зашифрования на данном шифре. Введенная модель шифра отражает лишь функциональные свойства шифрования и расшифрования в классических, с точки зрения истории криптографии, системах шифрования (в системах с симметричным ключом). В этой модели открытый текст (или шифрованный текст) — это лишь элемент абстрактного множества X (или Y), не учитывающий особенностей языка, его статистических свойств, вообще говоря, не являющийся текстом в его привычном понимании. При детализации модели шифра в ряде случаев указывают природу элементов множеств.

Рассмотрим примеры.

Обозначим через I некоторый алфавит, а через I^*– множество всех слов в алфавите I, то есть множество конечных последовательностей $(i_1, i_2, ..., i_L)$, $i_j \in I, j \in \{1, ..., L\}$, $L \in \{1, 2, ...\}$.

Шифр простой замены. Пусть $X = M$ — некоторое подмножество из I^*, а K — множество всех подстановок на I, т.е. $K = S(I)$ — симметрическая группа подстановок на I. Для каждого $g \in K$ определим f_g, положив для $(i_1, i_2, ..., i_L)$ из M

$$f_g(i_1, i_2, ..., i_L) = g(i_1), g(i_2), ..., g(i_L).$$

Положим дополнительно

$$f(i_1, i_2, ..., i_L, g) = f_g(i_1, i_2, ..., i_L)$$

и $Y = f(M) = \{f(i_1, i_2, ..., i_L, g): g \in S(I), (i_1, i_2, ..., i_L) \in M\}$. Таким образом, нами определен шифр $A = (M, S(I), Y, f)$ простой замены, более точно:

алгебраическая модель шифра простой замены с множеством открытых текстов $X = M$.

Шифр перестановки. Положим X — множество открытых (содержательных) текстов в алфавите I длины, кратной T. $K = S_T$ — симметрическая группа подстановок степени T, для $g \in S_T$ определим f_g, положив для $(i_1, i_2, ..., i_T) \in X$

$$f_g(i_1, i_2, ..., i_T) = (i_{g(1)}, i_{g(2)}, ..., i_{g(T)});$$

доопределим fg на остальных элементах из X по правилу: текст $x \in X$ делится на отрезки длины T и каждый отрезок длины T шифруется на ключе g по указанному выше закону шифрования. Последовательность, составленная из букв образов зашифрованных слов, является шифрованным текстом, соответствующим открытому тексту x и ключу g. Таким образом, нами определена функция $f: X \times K \to Y$ и шифр перестановки (X, S_T, Y, f). Для шифрования текста длины, не кратной T, его дополняют буквами до длины, кратной T.

Шифр гаммирования. Пусть буквы алфавита I упорядочены в некотором естественном порядке. «Отождествим» номера этих букв с самими буквами. То есть формально положим $I = \{1, 2, ..., n - 1, 0\}$, $|I| = n$. Положим X — некоторое подмножество множества I^L, $K \subseteq I^L$. Для ключа $\gamma = \gamma_1, \gamma_2, ..., \gamma_L$ из K и открытого текста $x = i_1, i_2, ..., i_L$ из X положим $f_\gamma(i_1, i_2, ..., i_L) = y_1, y_2, ..., y_L$, где $y_j = i_j + \gamma_j \bmod (n)$, $j \in \{1, ..., L\}$. Иногда под шифром гаммирования понимают и следующие способы шифрования: $y_j = i_j - \gamma_j$; $y_j = \gamma_j - i_j \bmod (n)$.

Поточный шифр. Шифр поточной замены. Введем сначала вспомогательный шифр (I, Γ, Y, f) для шифрования букв алфавита I. Для ключа $\gamma_1 \in \Gamma$ и буквы (открытого текста) $I \in I$ шифрованный текст имеет вид $f_{\gamma_1}(i) = y$. Обозначим через K — множество ключей поточного шифра. Для натурального числа L введем отображение Φ: $K \to \Gamma^L$, для фиксированного ключа $\chi \in K$ положим $\Phi(\chi) = \gamma_1, \gamma_2, ..., \gamma_L$. *Поточный шифр* (I^L, K, F, Y') для вспомогательного шифра $(X = I, K = \Gamma, Y, f)$ шифрует открытый текст $i_1, i_2, ..., i_L$ на ключе $\chi \in K$ по правилу

$$F\chi(i_1, i_2, ..., i_L) = f_{\gamma_1}(i_1), f_{\gamma_2}(i_2), ..., f_{\gamma_L}(i_L),$$

где $f_\gamma(i) = f(i, \gamma)$.

Поточным шифром замены мы называем поточный шифр, для которого опорный шифр имеет вид $(X = I, K = \Gamma, Y = I, f)$, а $(f_\gamma)\gamma \in \Gamma$ — семейство подстановок на I. Примерами поточных шифров служат шифры гаммирования, шифры простой замены. Поточный шифр с опорным шифром вида: $I = K = \{1, 2, ..., n\}$, $f(i, \gamma) = i + \gamma \bmod |I|$ так же называют шифром гаммирования. При этом условно различают программный

шифр гаммирования, в случае $|K| < |I|^L$, и случайный шифр гаммирования, в случае $K = I^L$, Ф — тождественное отображение.

Более общее понятие поточного шифра состоит в том, что в качестве множества открытых текстов рассматриваются все последовательности алфавита I длины, не превосходящей некоторого $L(0)$.

Для шифрования текстов длины L используется гамма $\Phi_L(\chi) = \gamma_1, \gamma_2, ..., \gamma_L$. Таким образом, используются L функций $\Phi_j, j \in \{1, ..., L(0)\}$.

Произведение шифров. Произведением шифров $A_1 = (X_1, K_1, Y_1, f_1)$, $A_2 = (X_2, K_2, Y_2, f_2)$, $Y_1 \subseteq X_2$ называют шифр $A = (X_1, K_1 x K_2, Y_2, f)$, для которого $f(x, (\chi_1, \chi_2)) = f_2(f_1(x, \chi_1), \chi_2)$, $(\chi_1, \chi_2) \in K_1 x K_2$.

Транзитивность шифра. Шифр $A = (X, K, Y, f)$ называют *транзитивным*, если при любых $x \in X$ и $y \in Y$ найдется $\chi \in K$, при котором $f(x, \chi) = y$. Исходя из введенных определений, легко доказывается, что для транзитивного шифра

$$|X| \le |Y| \le |K|.$$

Основные параметры шифра. Ряд требований к шифрам формулируются с использованием понятий, точное определение которых будет дано позднее. Тем не менее, на качественном уровне понимания эти параметры можно трактовать следующим образом.

Стойкость шифра. Ряд шифров являются совершенными в том смысле, что положение противника, стремящегося к их дешифрованию, не облегчается в результате перехвата шифротекста, то есть наличие криптограммы не уменьшает неопределенности в возможном выборе открытого текста. Такие шифры относят к так называемому классу *теоретически стойких* шифров. Ряд шифров, а это многие практически используемые шифры, таковы, что эта неопределенность при перехваченной криптограмме полностью исчезает, то есть становится известным, что данная криптограмма может быть получена шифрованием только единственного открытого текста (неизвестно только какого). Уровень стойкости таких систем оценивается по затратам времени и сил, необходимых для получения этого единственного открытого текста. При больших затратах или малой вероятности успеха в дешифровании говорят, что шифр *практически стойкий*.

Объем ключа. Ключ шифрования (он же ключ расшифрования) должен быть не известен противнику и находиться как в передающем пункте связи, так и в приемном пункте. Обычная практика использования ключа состоит в том, что он используется как одноразовый шприц — единожды, при шифровании лишь одного открытого текста. Для регулярной связи корреспондентов, следовательно, надо иметь в их пунктах связи достаточно большое количество ключей, то есть должна

решаться задача секретной доставки ключей. Эта задача решается более просто, если объем каждого ключа сравнительно небольшой.

Сложность выполнения операций шифрования и расшифрования. Эти операции должны быть по способу выполнения по возможности простыми. Если они выполняются вручную, то их сложность приводит к большим затратам времени на их выполнение и появлению ошибок. При использовании шифровальной аппаратуры возникают вопросы о простоте технической реализации аппаратуры, ее стоимости и о достижении необходимой скорости выполнения операций, связанных с процессами шифрования и расшифрования.

Разрастание числа ошибок. В некоторых типах шифров ошибка в одной букве, допущенная при шифровании, приводит к большому числу ошибок в расшифрованном тексте. Такие ошибки разрастаются в результате операций расшифрования, вызывая значительную потерю информации, и часто требует повторного зашифрования текста и передачи новой криптограммы. Естественно, при выборе шифра для связи стараются минимизировать возрастание числа ошибок.

Помехоустойчивость шифра. При действии помех в линиях связи происходит искажение текста криптограммы, что приводит при расшифровании к искажениям открытого текста, а зачастую и к нечитаемости текста. Свойство шифра противостоять разрастанию ошибок при расшифровании текстов называют *помехоустойчивостью.*

Имитостойкость шифра. К активным действиям противника в канале связи относят его попытки навязать абоненту сети связи ложную информацию. Это делается путем искажения шифрованного текста в канале связи, либо его замене на ранее переданный шифротекст. Бывают и другие действия противника, ведущие к принятию ложной информации. Шифры, обладающие свойством противостоять попыткам навязывания ложной информации, называются *имитостойкими.*

Увеличение объема сообщения. Для некоторых шифров объем сообщения увеличивается в результате операции шифрования. Этот нежелательный эффект проявляется, например, при попытке выровнять статистику сообщения путем добавления некоторых вспомогательных символов («пустышек») или при рандомизации открытого сообщения, то есть, по сути, применения к нему некоторого пропорционального кода.

Основные свойства модели шифра. Важным классом шифров является введенные выше так называемые *транзитивные шифры*, то есть шифры, для которых уравнение $f(x, \chi) = y$ разрешимо относительно $\chi \in K$ при любых парах $(x, y) \in X \times Y$ и так называемые t-транзитивные шифры, шифры, для которых система уравнений $f(x(j), \chi) = y(j)$,

$j \in \{1, ..., t\}$ имеет решение относительно $\chi \in K$ для любых подмножеств $\{x(1), x(2), ..., x(t)\} \subseteq X$ мощности t и любых подмножеств $\{y(1), y(2), ..., y(t)\} \subseteq Y$ мощности t.

Другой важный класс шифров представляют так называемые *эндоморфные шифры* (термин предложил К. Шеннон), то есть шифры (X, K, Y, f), для которых множество открытых текстов X совпадает с множеством криптограмм Y. Для таких шифров (X, K, Y, f) каждое преобразование $f\chi$, $\chi \in K$ является биекцией X в X (подстановкой на X). Множество таких биекций обозначают через $\Pi(K, f) = \{f\chi: \chi \in K\}$, а сам эндоморфный шифр — через $A = (X, \Pi(K, f))$ и называют *подстановочной моделью* эндоморфного шифра. При этом под ключом этого шифра понимают биекцию $\pi \in \Pi(K, f)$. Уравнение шифрования записывают в виде $\pi x = y$, уравнение расшифования записывают в виде $\pi^{-1} y = x$.

Эндоморфный шифр, у которого множество подстановок $\Pi(K, f)$ является смежным классом по некоторой подгруппе из $S(X)$, называют *групповым шифром*.

Транзитивный шифр, для которого $|X| = |K|$ называют *минимальным шифром*.

Для *эндоморфных шифров* $A_1 = (X, \Pi(K_1, f_1))$ $A_2 = (X, \Pi(K_2, f_2))$ используют понятие *произведения шифров* $A_1 \times A_2 = (X, \Pi(K_1, f_1) \times \Pi(K_2, f_2))$, где $\Pi(K_1, f_1) \times \Pi(K_2, f_2) = \{\pi_1\pi_2: \pi_1 \in \Pi(K_1, f_1), \pi_2 \in \Pi(K_2, f_2)\}$. Очевидно, произведение эндоморфных шифров будет транзитивным шифром, если таковым является хотя бы один из них.

Ключи χ, χ' шифра (X, K, Y, f) называются *эквивалентными*, если при любом $x \in X$

$$f(x, \chi) = f(x, \chi').$$

Вероятностная модель шифра. Одно из важнейших предположений К. Шеннона при исследовании секретных систем состояло в том, что каждому возможному передаваемому сообщению (открытому тексту) соответствует априорная вероятность, определяемая вероятностным процессом получения сообщения. Аналогично имеется и априорные вероятности использования различных ключей шифра. Эти вероятностные распределения на множестве открытых текстов и множестве ключей характеризуют априорные знания криптоаналитика противника относительно используемого шифра. При этом К. Шеннон предполагал, что сам шифр известен противнику.

ОПРЕДЕЛЕНИЕ. *Вероятностной моделью шифра* называется его алгебраическая модель с заданными дискретными, независимыми вероятностными распределениями $P(X) = (p(x), x \in X)$, $P(K) = (p(\chi), \chi \in K)$ на множествах X и K.

Естественно, вероятностные распределения на X и K индуцируют вероятностное распределение $P(Y) = (p(y), y \in Y)$ на Y, совместные распределения $P(X, K)$, $P(X, Y)$, $P(Y, K)$ и условные распределения $P(X/y) = (p(x/y), x \in X)$ и $P(K/y) = (p(\chi/y), \chi \in K)$.

Вероятностной модели шифра соответствует так называемая *матрица* $(p(y/x))$ размера $|X| \times |Y|$ *переходных вероятностей шифра*, составленная из условных вероятностей $p(y/x)$ — вероятности зашифрования открытого текста x в криптограмму y при случайном выборе ключа $\chi \in K$ в соответствии с $P(K)$.

Совершенные шифры. При определении теоретической стойкости шифра используют вероятностную модель шифра и следующие рассуждения.

Зная шифр и априорные вероятности открытых текстов и ключей, обладая перехватом шифртекста $y \in Y$, противник может вычислить условные вероятности $p(x/y)$ при всех $x \in X$. Если при этом окажется, что один из элементов $x(0)$ их X имеет значительную вероятность $p(x(0)/y) = 1 - \varepsilon$, а все остальные элементы их X, вместе взятые, имеют вероятность $\sum_{x \neq x(0)} p(x/y) = \varepsilon$, то это означает, что с надежностью $1 - \varepsilon$ найдено истинное открытое сообщение. В этом смысле говорят, что дешифрование сводится к вычислению апостериорных вероятностей $p(x/y)$ при всех $x \in X$. Напротив, если окажется, что при любом $x \in X$ выполняется равенство $p(x/y) = p(x)$, то перехваченная криптограмма y не несет никакой информации об открытом сообщении. Если это равенство выполняется дополнительно и при любом $y \in Y$, то это свидетельствует о высокой способности шифра противостоять попыткам дешифрования, то есть о высокой криптостойкости шифра. Последние шифры К. Шеннон назвал «совершенными» шифрами.

ОПРЕДЕЛЕНИЕ. Шифр (X, K, Y, f) с вероятностными распределениями $P(X) = (p(x), x \in X)$, $P(K) = (p(\chi), \chi \in K)$ называется *совершенным* (при нападении на $x \in X$ по перехвату $y \in Y$), если при любых $x \in X$ и $y \in Y$

$$p(x/y) = p(x).$$

Используя формулу условных вероятностей

$$p(x, y) = p(y/x)p(x) = p(x/y)p(y),$$

легко показывается, что *совершенство шифра равносильно условию*

$$p(y/x) = p(y)$$

при любых $x \in X$, $y \in Y$.

Несложно доказывается, что свойство совершенности шифра (X, K, Y, f), у которого $|X| = |K| = |Y|$, равносильно двум условиям: 1) $p(\chi) = \frac{1}{|K|}$, $\chi \in K$; 2) уравнение $f(x, \chi) = y$ однозначно разрешимо относительно $\chi \in K$ при любых $x \in X$ и $y \in Y$.

Одним из примеров совершенных шифров является шифр гаммирования $X = Y = K = I^L$ с равновероятным выбором ключа — гаммы. В качестве совершенных шифров выступают следующие шифры простой замены с множеством ключей $K = S(I)$ ($S(I)$ — симметрическая группа подстановок на I с равновероятным выбором ключа: 1) $X = I$ — алфавит текста; 2) X — множество всех слов алфавита I длины L, не содержащих одинаковых букв.

Способы представления реализаций шифров. В современном мире, говоря о реализациях шифров, обычно употребляют термины: шифратор, шифросистема, алгоритм шифрования, программа шифрования. Последние два два понятия общеизвестны и не требуют объяснений. Первый термин употребляется для указания собственно устройства, реализующего процесс шифрования и расшифрования заданного шифра. Термин же шифросистема используется в более широком смысле как обозначение всего устройства, включающего в себя и устройства, предназначенные для предварительной обработки шифруемых «текстов», так и ряд других вспомогательных устройств ввода и вывода информации. Термины же криптосхема, шифрующий автомат являются специфическими и малоизвестными. Для записи законов функционирования шифратора используют схемное описание, состоящее из прямоугольников с надписями узлов и блоков шифратора. При заданном их математическом описании функционирования описание функционирования шифратора задается указанием связей между отдельными блоками и узлами. Совокупность блоков с заданными связями и описанием функционирования каждого блока обычно и называют *криптосхемой* шифратора. В таких описаниях преследуют цель указать принцип функционирования шифратора и обычно не указывают конкретные, численные значения всех параметров узлов и блоков.

В таком виде может быть представлена принципиальная криптосхема *поточного шифратора* (рис. 2.8).

Управляющий блок предназначен для выработки управляющей последовательности шифрующим блоком. Шифрующий блок предназначен для зашифрования символов открытого текста с помощью управляющей последовательности.

Рис. 2.8. Принципиальная криптосхема поточного шифратора

Ключом криптосхемы может являться, например, заполнение памятей узлов и блоков, составляющих управляющий блок, а в ряде случаев и закон их функционирования. Ниже будет указана криптосхема и другого класса шифров — блочных шифров.

Для описания функционирования дискретных устройств, реализующих процесс шифрования или реализующих отдельные блоки шифратора, зачастую применяют язык *теории автоматов*.

ОПРЕДЕЛЕНИЕ. Три конечных непустых множества X, S, Y, два семейства отображений $(h_x)_{x\in X}$, $(f_x)_{x\in X}$, $h_x\colon S \to S$, $f_x\colon S \to Y$, $x \in X$ и приводимый ниже закон функционирования называют *конечным автоматом* (*конечным автоматом Мили*) и обозначают через $A = (X, S, Y, (h_x)_{x\in X}, (f_x)_{x\in X})$.

Полагают, что автомат моделирует работу многих дискретных устройств, при этом множество X называют входным алфавитом автомата, S — множеством состояний, Y — выходным алфавитом автомата функции $(h_x)_{x\in X}$, $(f_x)_{x\in X}$ называют, соответственно, частичными функциями переходов и выходов автомата. Автомат A функционирует в дискретном времени $t \in \{1, 2, ...\}$. При входной последовательности $P = x_1, x_2, ..., x_j \in X$ и начальном состоянии $s = s_1 \in S$ автомат вырабатывает последовательность состояний $A_M(s, P) = s_1, s_2, ...$ (последовательно находится в состояниях $s_1, s_2, ...$) и выходную последовательность $A(s, P) = y_1, y_2, ...$. Правила получения этих последовательностей таковы:

$s_2 = h_{x1}s_1$ — образ s_1 при отображении h_{x1}, $s_{j+1} = h_{xj}s_j$, $j \in \{1, 2, ...\}$,

$y_1 = f_{x1}s_1$ — образ s_1 при отображении f_{x1}, $y_j = f_{xj}s_j$, $j \in \{1, 2, ...\}$.

В случае $f_x = f_{x'}$ для любых $x, x' \in X$ автомат A называют *автоматом Мура*.

Автономным конечным автоматом называют двойку конечных множеств S, Y и два отображения: $h\colon S \to S$ и $\lambda\colon S \to Y$ и обозначают через $A = (S, Y, h, \lambda)$. В ряде случаев используют и другое определение: Автомат $A = (X, S, Y, (h_x)_{x\in X}, (f_x)_{x\in X})$ называют автономным в случае, когда $|X| = 1$.

Для автомата определяют его *граф переходов*: совокупность точек на плоскости, обозначенные состояниями автомата, некоторые из которых соединены ориентированными ребрами (стрелками →). На ре-

бре, соединяющем состояние s с состоянием s', ставятся две пометки: x/y, где x и y определены из соотношений $h_x s = s', f_x s = y$. Переход из состояния в состояние по стрелкам называется *путем* в графе переходов автомата. Путь определяет последовательность состояний $A_M(s, P)$ и выходную последовательность $A(s, P)$, отвечающую входной последовательности P автомата A и его начальному состоянию $s_1 = s \in S$.

Отметим, что задание семейства отображений $(h_x)_{x \in X}, ((f_x)_{x \in X})$ равносильно заданию отображения h: $X \times S \to S$, $h(x, s) = h_x(s)$ (аналогично, f: $X \times S \to Y$, $f(x, s) = f_x(s)$). В связи с чем чаще автомат A определяют тремя множествами X, S, Y, двумя отображениями h, f: $A = (X, S, Y, h, f)$ и законом фукционирования.

Обозначим через X^* множество всех слов конечной длины в алфавите X. Автомат A с начальным состоянием s задает отображение $\varphi_{A, s}$ $X^* \to Y^*$, именно, для $P \in X^*$

$$\varphi_{A, s}(P) = A(s, P).$$

Такие отображения называются конечно-автоматными, или просто *автоматными*.

При фиксированных множествах X, S, Y автомат задается отображениями h, f. При моделировании функционирования шифратора, или его криптосхемы конечным автоматом начальные состояния автомата моделируют так называемые части ключа, иногда и весь ключ, заключенный в памяти криптосхемы, часть же ключа «логики криптосхемы», иногда и весь ключ, моделируется выбором функций переходов h, f автомата. При этом полагают, что входными последовательностями автомата являются открытые тексты, подлежащие шифрованию, а выходные последовательности автомата трактуются как шифрованные тексты, соответствующие открытым текстам и ключам. При моделировании блочных шифров открытыми текстами являются начальные состояния автомата, ключами (раундовыми ключами) являются элементы алфавита X. Шифрованным текстом, который соответствует открытому тексту и последовательности P раундовых ключей, является заключительное состояние автомата, соответствующее начальному состоянию и входной последовательности P. Более полное представление об автоматах читатель может составить, ознакомившись с изданными материалами автора.

Понятие *шифрующего автомата* трактуется неоднозначно.

Первое определение состоит в том, что *шифрующий автомат* есть множество автоматов $A(r)$, $r \in R$ с начальными состояниями $s(r) \in S(r)$. Такое определение равносильно тому, что под шифрующим автоматом понимают некоторое *множество автоматных отображений* множества открытых текстов в шифрованные.

Второе определение шифрующего автомата состоит в том, что автомат $A = (X, S, Y, (h_x)_{x\in X}, (f_x)_{x\in X})$ является *шифрующим автоматом, если его автоматные отображения* $\varphi_{A,s}\ X^* \to Y^*$, $s \in S$ *являются инъективными* отображениями. Такое определение согласуется с определением шифра (X, K, Y, f) в том смысле, что в качестве отображений $f\chi$ берутся автоматные инъективные отображения.

Для большей общности иногда второе определение обобщают. Именно рассматривают автоматы, у которых $X = \Gamma \times I$, где Γ — алфавит внешней части ключа, часть ключа: $\gamma = \gamma_1, \gamma_2, ..., \gamma_L, \gamma_j \in \Gamma$; I — алфавит открытого текста. При фиксированых частях ключа $\gamma \in \Gamma^L$ и $s \in S$ требуют инъективность отображения I^L в Y^L, то есть при входных различных последовательностях вида $P = (a_1, \gamma_1), (a_2, \gamma_2), ..., (a_L, \gamma_L)$ и $P' = (a'_1, \gamma_1), (a'_2, \gamma_2), ..., (a'_L, \gamma_L)$ требуют, чтобы $A(s, P) \neq A(s, P')$ при любом натуральном L.

Выясним условия, при которых автомат $A = (X, S, Y, h, f)$ является шифрующим автоматом, то есть автоматные отображения $\varphi_{A,s}\ X^* \to$ $\to Y^*$, $s \in S$ являются инъективными отображениями. Для отображения $f: X \times S \to Y$ обозначим через f_s отображение X в Y: $f_s(x) = f(x, s)$. Через S_s обозначим множество состояний автомата A, содержащее s и все состояния $s' \in S$, достижимые из s в графе переходов автомата A, то есть для которых есть пути из s в s'. На множестве S_s определен подавтомат $A_s = (X, S_s, Y, h, f)$ автомата A (здесь ограничения отображений h, f обозначены теми же буквами).

С использованием введенных определений несложно доказывается.

УТВЕРЖДЕНИЕ. Автоматное отображение $\varphi_{A,s}\ X^* \to Y^*$, $s \in S$ является инъективным тогда и только тогда, если при каждом состоянии s' из S_s отображение $f_{s'}$ инъективно.

В *третьем определении* под шифрующим автоматом понимают автомат, моделирующий устройство шифрования либо некоторого его блока. В таком понимании устройство шифрования моделируют автоматом $A = (X, S, Y, (h_x)_{x\in X}, (f_x)_{x\in X})$, у которого отображения $(h_x)_{x\in X}$ S в S являются биекциями S в S. Такие автоматы обычно называют *перестановочными*. Часто гаммообразующее устройство шифратора называют *шифрующим автоматом*.

Эквивалентность ключей шифрующего автомата

ОПРЕДЕЛЕНИЕ. Состояния s, s' автомата A называются *неотличимыми*, если

$$A(s, P) = A(s', P)$$

при любом входном слове $P \in I^*$.

Автомат A называется *приведенным*, если он не имеет различных неотличимых состояний.

Ключи s, s' шифрующего автомата A называются *эквивалентными*, если s, s' — неотличимые состояния автомата A.

2.3. Особенности построения блочных шифров

Современные алгоритмы шифрования возникли на базе развития и совершенствования простейших шифров, путем устранения имеющихся у них криптографических слабостей. Большинство современных шифров можно рассматривать как усиление и модернизацию известных с древних времен шифров простой замены, перестановки, о которых мы говорили в предыдущих разделах этой главы.

Всем очевидна возможность дешифрования шифра простой замены в случае, если достаточно велико соотношение между объемом материала и размером алфавита открытого и шифрованного текстов. Однако шифр простой замены сложно вскрыть, если это соотношение мало, например, в случае, когда шифруются тексты единичной длины (одна буква). Отсюда и вытекает подход к усилению шифра простой замены — разрабатывают шифры, для которых это соотношение чрезвычайно мало. Делают это двумя способами: увеличивают алфавит либо максимально уменьшают объем сообщения, шифруемого с помощью одной и той же замены.

Первый путь (увеличение алфавита) реализован в шифрах многозначной замены, в кодах и современных блочных шифрах.

По второму пути (уменьшение числа знаков, шифруемых по одной замене) пошли при создании поточных шифров замены.

Шифры многозначной замены. Шифр многозначной замены задается табл. 2.1.

Таблица 2.1

А	Б	В	Г	Д	...
а1	б1	в1	г1	д1	...
а2	б2	в2	г2	д2	...
а3	б3	в3	г3	д3	...
а4	...	...	...	...	...
...	...	...	...	...	...

Здесь каждой букве отвечает несколько символов шифротекста. Алфавит шифротекста больше алфавита открытого текста. Шифровальщик, зашифровывая открытый текст, должен выбрать для каждой буквы одно из обозначений, например, А зашифровывается в a1, или в a2, или в a3 и т.д. Данный шифр при грамотном использовании может значительно выровнять диаграмму встречаемости символов в шифротексте.

Коды. Идея увеличения алфавита открытого текста реализована в кодах. Код представляет собой два словаря. Первый словарь предназначен для зашифрования, а второй — для расшифрования. В словаре для зашифрования в алфавитном порядке написаны символы алфавита открытого текста: отдельные буквы, слова, целые предложения и для каждого символа указано его кодообозначение. При шифровании шифровальщик каждый символ (или слово, предложение) заменяет с помощью первого словаря на кодообозначение. Это преобразование неоднозначно. Слово можно заменить по буквам или попытаться подобрать кодообозначение для целого слова или фразы. При расшифровании используется второй словарь, в котором в алфавитном порядке записаны кодообозначения, и расшифрование сводится к замене их на символы открытого текста. Словарь может состоять из тысяч, десятков тысяч слов. Дешифровать код достаточно сложно. Для этого необходимо набрать достаточно много материала. Коды находят определенное применение, например есть военно-морские коды, дипломатические коды и т.п. Недостаток этой системы шифрования: каждый код — это две книги, их надо напечатать без ошибок, разослать всем участникам закрытой связи. Если одна такая книга попадет злоумышленнику, то код надо срочно менять, система инерционная.

Промежуточным вариантом между простой заменой и кодом является блочный шифр. В нем текст делится на блоки и проводится простая замена блоков. Когда длина блока достаточно велика, таблица замены становится необозримой и саму замену приходится задавать не таблицей, а некоторым алгоритмом преобразования.

Блочный шифр «два квадрата». Блочные шифры, в которых заменялись пары букв, применяли во время Второй мировой войны немцы в низовых линиях связи. Они основаны на блочном шифре Плейфейра (у него был один квадрат). При шифровании открытый текст разбивался на блоки по две буквы, например:

КР ИП ТО ГР АФ ИЯ

Ключом являлись два квадрата, в которых записывался алфавит в произвольном порядке.

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

Первая буква выбиралась в левом квадрате, вторая — в правом. Мысленно строился прямоугольник, и в шифротекст включались буквы из не занятых его углов. Так, в примере на место **К** ставилась буква из соответствующего незанятого угла прямоугольника второго квадрата **Ж**, а вместо **Р** ставилась **Ф**. Если буквы оказывались в одной строке, то буква заменялась на букву той же позиции, но из другого квадрата. Так, в примере вместо **И** ставилась буква этого же столбца второго квадрата **К**, а вместо **П** — соответствующая буква первого квадрата **З**. Таким образом, слово криптография после зашифрования имело бы вид

ЖФКЗФБЕРРУЩР

Советским криптографам в годы войны шифры типа «*два квадрата*» удавалось дешифровывать, но это требовало значительных усилий и опыта.

Преимуществом данных шифров перед кодами были достаточная простота и быстрота зашифрования и расшифрования, отсутствие потребности в словарях, простота в смене ключевых квадратов. Практически шифр «*два квадрата*» является забытым шифром англичанина Чарльза Уинстона[1], который назывался «двойным квадратом».

Сеть Фейстеля. Блочные шифры — это шифры простой замены с большим алфавитом «открытого текста». Многие блочные шифры используют в своем построении так называемую идею Хорста Фейстеля, состоящую в реализации многих «раундов» шифрования, каждый из которых реализуется криптосхемой.

Хорст Фейстель (*Horst Feistel*, 1915—1990) в 1971 г. запатентовал два устройства, реализовавшие различные алгоритмы шифрования, назван-

[1] В 1854 г. англичанин Чарльз Уинстон разработал метод шифрования биграммами, который называют «двойным квадратом». Свое название этот шифр получил по аналогии с полибианским квадратом.

ные затем общим названием «Люцифер» (*Lucifer*). Одно из устройств использовало конструкцию, впоследствии названную «сетью Фейстеля» («*Feistel cipher*», «*Feistel network*»).

Сеть Фейстеля имеет следующую структуру. Входной блок делится на несколько равной длины подблоков, называемых ветвями. В случае, если блок имеет длину 64 бита, используются две ветви по 32 бита каждая. Каждая ветвь обрабатывается независимо от другой, после чего осуществляется циклический сдвиг всех ветвей влево. Такое преобразование выполняется несколько циклов или раундов. В случае двух ветвей каждый раунд имеет структуру, показанную на рис. 2.9.

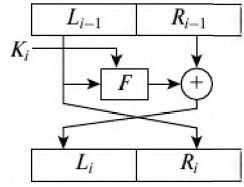


Рис. 2.9. I раунд сети Фейстеля

Функция F называется образующей. Каждый раунд состоит из вычисления функции *F* для одной ветви и побитового выполнения операции XOR результата *F* с другой ветвью. После этого ветви меняются местами. Считается, что оптимальное число раундов — от 8 до 32.

Важно то, что увеличение количества раундов значительно увеличивает криптостойкость алгоритма. Возможно, эта особенность и повлияла на столь активное распространение сети Фейстеля, так как для большей криптостойкости достаточно просто увеличить количество раундов, не изменяя сам алгоритм. В последнее время количество раундов не фиксируется, а лишь указываются допустимые пределы.

Сеть Фейстеля является обратимой даже в том случае, если функция *F* не является таковой, так как для дешифрования не требуется вычислять F^{-1}. Для дешифрования используется тот же алгоритм, но на вход подается зашифрованный текст, и ключи используются в обратном порядке.

В настоящее время все чаще используются различные разновидности сети Фейстеля для 128-битного блока с четырьмя ветвями. Увеличение количества ветвей, а не размерности каждой ветви связано с тем, что наиболее популярными до сих пор остаются процессоры с 32-разрядными словами, следовательно, оперировать 32-разрядными словами эффективнее, чем с 64-разрядными.

Основной характеристикой алгоритма, построенного на основе сети Фейстеля, является функция *F*. Различные варианты касаются также начального и конечного преобразований. Подобные преобразования, называемые забеливанием (*whitening*), осуществляются для того, чтобы выполнить начальную рандомизацию входного текста.

На основе различных модицикаций сети Фейстеля построены многие современные алгоритмы симметричного шифрования: DES, ГОСТ 28147—89, Blowfish, RC5, FEAL и ряд других. Сети Фейстеля были широко изучены криптографами в силу их обширного распространения. В 1988 г. Майкл Люби (*Michael Luby*) и Чарльз Ракофф (*Charles Rackoff*) провели исследования сети Фейстеля и доказали, что если раундовая функция является криптостойкой псевдослучайной и используемые ключи независимы в каждом раунде, то трех раундов будет достаточно для того, чтобы блочный шифр являлся псевдослучайной перестановкой.

Иногда сеть Фейстеля в западной литературе называют «*Luby-Rackoff block cipher*» в честь Люби и Ракоффа, которые проделали большой объем теоретических исследований в этой области.

Во многих блочных шифрах на основе сети Фейстеля были найдены те или иные уязвимости, однако в ряде случаев эти уязвимости являются чисто теоретическими и при нынешней производительности компьютеров использовать их на практике для взлома невозможно.

2.4. Блочный шифр DES

Стандарт шифрования данных DES (*Data Encryption Standard*) был опубликован в 1977 г. Национальным бюро стандартов США (*National Bureau of Standards*, *NBS*). Стандарт DES был предназначен для защиты от несанкционированного доступа к важной, но несекретной информации в государственных и коммерческих организациях США. Долгое время DES являлся самым распространенным алгоритмом, используемым в системах защиты коммерческой информации.

В конце 1996 г. Национальным институтом стандартов США (*NIST*) был объявлен конкурс на создание нового общенационального стандарта шифрования, который должен был прийти на замену DES. В настоящее время в качестве американского стандарта блочного шифрования AES (*Advanced Encryption Standard*) испльзуется алгоритм *Rijndael*.

Основные достоинства алгоритма DES:

— используется только один ключ длиной 56 бит;

— зашифровав сообщение с помощью одного пакета программ, для расшифровки можно использовать любой другой пакет программ, соответствующий стандарту DES;
— относительная простота алгоритма обеспечивает высокую скорость обработки;
— достаточно высокая стойкость алгоритма.

DES осуществляет шифрование 64-битовых блоков данных с помощью 64-битового ключа, в котором значащими являются 56 бит (остальные 8 бит — проверочные биты для контроля на четность). Дешифрование в DES является операцией, обратной шифрованию, и выполняется путем повторения операций шифрования в обратной последовательности. Обобщенная схема процесса шифрования в алгоритме DES показана на рис. 2.10. Процесс шифрования заключается в начальной перестановке битов 64-битового блока, шестнадцати циклах шифрования и, наконец, в конечной перестановке битов.

Рис. 2.10. Обобщенная схема шифрования в алгоритме DES

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

Все перестановки и коды в таблицах подобраны разработчиками таким образом, чтобы максимально затруднить процесс расшифровки путем подбора ключа. При описании алгоритма DES (рис. 2.11) применены следующие обозначения:

— L и R — последовательности битов (левая (*left*) и правая (*right*));
— LR — конкатенация последовательностей L и R, т.е. такая последовательность битов, длина которой равна сумме длин L и R; в последовательности LR биты последовательности R следуют за битами последовательности L;
— $\oplus$ — операция побитового сложения по модулю 2.

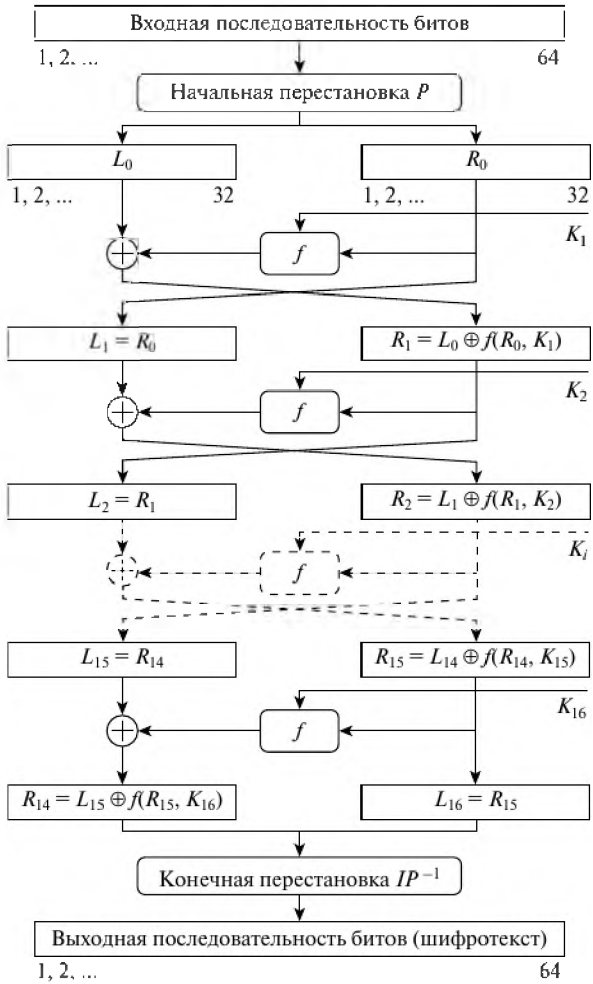


Рис. 2.11. Структура алгоритма DES

Пусть из файла исходного текста считан очередной 64-битовый (8-байтовый) блок T. Этот блок T преобразуется с помощью *матрицы начальной перестановки* P (табл. 2.2).

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

Таблица 2.2

Матрица начальной перестановки *P*

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

Затем выполняется итеративный процесс шифрования, состоящий из 16 шагов (циклов). Пусть T_i — результат i-й итерации: $T_i = L_i R_i$, где $L_i = t_1 t_2 \ldots t_{32}$ (первые 32 бита); $R_i = t_{33} t_{34} \ldots t_{64}$ (последние 32 бита). Тогда результат i-й итерации описывается следующими формулами:

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

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

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

По окончании шифрования осуществляется восстановление позиций битов с помощью матрицы обратной перестановки P^{-1} (табл. 2.3).

Таблица 2.3

Матрица обратной перестановки P^{-1}

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

Пример того, как соотносятся элементы первой строки матрицы P^{-1} с элементами матрицы P, приведен в таблице 2.4.

Таблица 2.4

Связь элементов матриц

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

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

Итеративный процесс расшифрования может быть описан следующими формулами:

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

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

Теперь рассмотрим, что скрывается под преобразованием, обозначенным буквой f. Схема вычисления функции шифрования $f(R_{i-1}, K_i)$ показана на рис. 2.12.

Для вычисления значения функции f используются:

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

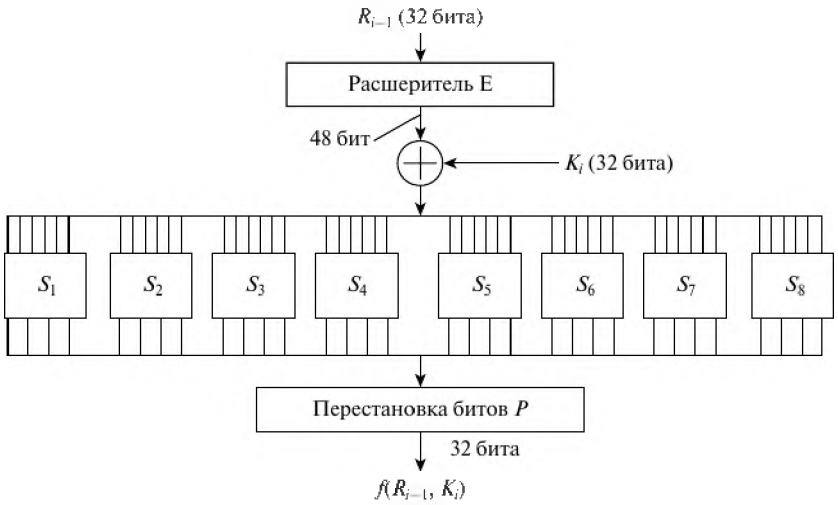


Рис. 2.12. Схема вычисления функции шифрования f

Приведем определения этих функций.

Аргументами функции шифрования f являются R_{i-1} (32 бита) и K_i (48 бит). Результат функции $E(R_{i-1})$ есть 48-битовое число. Функция расширения E, выполняющая расширение 32 бит до 48 (принимает блок из 32 бит и порождает блок из 48 бит), определяется в табл. 2.5.

Таблица 2.5

Функция расширения E

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

В соответствии с таблицей 2.5 первые три бита $E(R_{i-1})$ — это биты 32, 1 и 2, а последние — 31, 32, 1. Полученный результат (обозначим его $E(R_{i-1})$) складывается по модулю 2 (операция *XOR*) с текущим значением ключа K_i и затем разбивается на восемь 6-битовых блоков $B_1, B_2, ..., B_8$:

$$E(R_{i-1}) \oplus K_i = B_1, B_2, ..., B_8.$$

Далее каждый из этих блоков используется как номер элемента в функциях-матрицах S_1, S_2, ..., S_8, содержащих 4-битовые значения (табл. 2.6).

Таблица 2.6

Функции преобразования S_1, S_2, ..., S_8

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

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

В результате получаем $S_1(B_1)$ $S_2(B_2)$ $S_3(B_3)$ ... $S_8(B_8)$, т.е. 32-битовый блок (поскольку матрицы S_j содержат 4-битовые элементы). Этот 32-битовый блок преобразуется с помощью функции перестановки битов P (табл. 2.7).

Таблица 2.7

Функция P перестановки битов

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

Таким образом, функция шифрования

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

Как нетрудно заметить, на каждой итерации используется новое значение ключа K_i (длиной 48 бит). Новое значение ключа K_i вычисляется из начального ключа K (рис. 2.13). Ключ K представляет собой 64-битовый блок с 8 битами контроля по четности, расположенными в позициях 8, 16, 24, 32, 40, 48, 56, 64. Для удаления контрольных битов и подготовки ключа к работе используется функция G первоначальной подготовки ключа (табл. 2.8).

Табл. 2.8 разделена на две части. Результат преобразования $G(K)$ разбивается на две половины C_0 и D_0 по 28 бит каждая. Первые четыре строки матрицы G определяют, как выбираются биты последователь-

ности C_0 (первым битом C_0 будет бит 57 ключа шифра, затем бит 49 и т.д., а последними битами — биты 44 и 36 ключа).

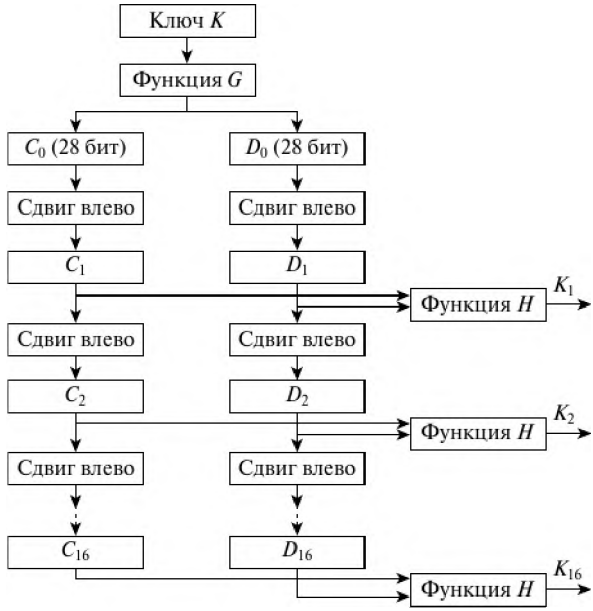


Рис. 2.13. Схема алгоритма вычисления ключей K_i

Таблица 2.8

Функция *G* первоначальной подготовки ключа (переставленная выборка 1)

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

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

Как видно из табл. 2.8, для генерации последовательностей C_0 и D_0 не используются биты 8, 16, 24, 32, 40, 48, 56 и 64 ключа шифра. Эти биты не влияют на шифрование и могут служить для других целей (на-

пример, для контроля по четности). Таким образом, в действительности ключ шифра является 56-битовым.

После определения C_0 и D_0 рекурсивно определяются C_i и D_i, $i \in \{1, 2, ..., 16\}$. Для этого применяются операции циклического сдвига влево на один или два бита в зависимости от номера шага итерации, как показано в табл. 2.9.

Таблица 2.9

Таблица сдвигов s_i для вычисления ключа

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

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

Ключ K_i, определяемый на каждом шаге итерации, есть результат выбора конкретных битов из 56-битовой последовательности C_i D_i и их перестановки. Другими словами, ключ $K_i = H(C_i D_i)$, где функция H определяется матрицей, завершающей обработку ключа (табл. 2.10).

Таблица 2.10

Функция H завершающей обработки ключа (переставленная выборка 2)

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Как следует из табл. 2.10, первым битом ключа K_i будет 14-й бит последовательности C_iD_i, вторым — 17-й бит, 47-м битом ключа K_i будет 29-й бит C_iD_i, а 48-м битом — 32-й бит C_iD_i.

Основные режимы работы алгоритма DES

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

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

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

Режим «Электронная кодовая книга». Длинный файл разбивают на 64-битовые отрезки (блоки) по 8 байтов. Каждый из этих блоков шифруют независимо с использованием одного и того же ключа шифрования (рис. 2.14).

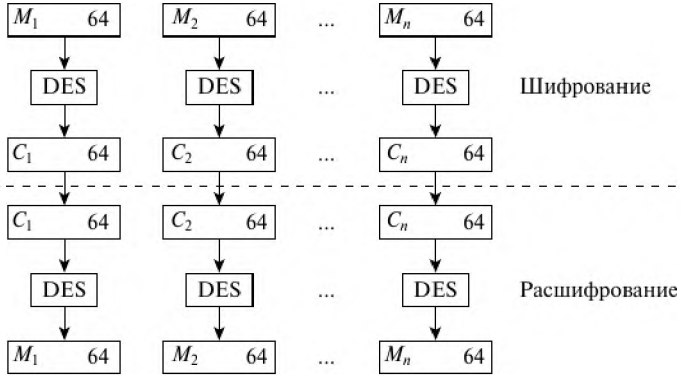


Рис. 2.14. Схема алгоритма DES в режиме электронной кодовой книги

Основное достоинство — простота реализации. Из-за фиксированного характера шифрования при ограниченной длине блока 64 бита возможно проведение криптоанализа «со словарем». Блок такого размера может повториться в сообщении вследствие большой избыточности в тексте на естественном языке. Это приводит к тому, что идентичные блоки открытого текста в сообщении будут представлены

идентичными блоками шифртекста, что дает криптоаналитику некоторую информацию о содержании сообщения.

Режим «Сцепление блоков шифра». В этом режиме (рис. 2.15) исходный файл M разбивается на 64-битовые блоки: $M = M_1 M_2 ... M_n$. Первый блок M_1 складывается по модулю 2 с 64-битовым начальным вектором IV, который меняется ежедневно и держится в секрете. Полученная сумма затем шифруется с использованием ключа DES, известного и отправителю, и получателю информации. Полученный 64-битовый шифр C_1 складывается по модулю 2 со вторым блоком текста, результат шифруется и получается второй 64-битовый шифр C_2, и т.д. Процедура повторяется до тех пор, пока не будут обработаны все блоки текста.

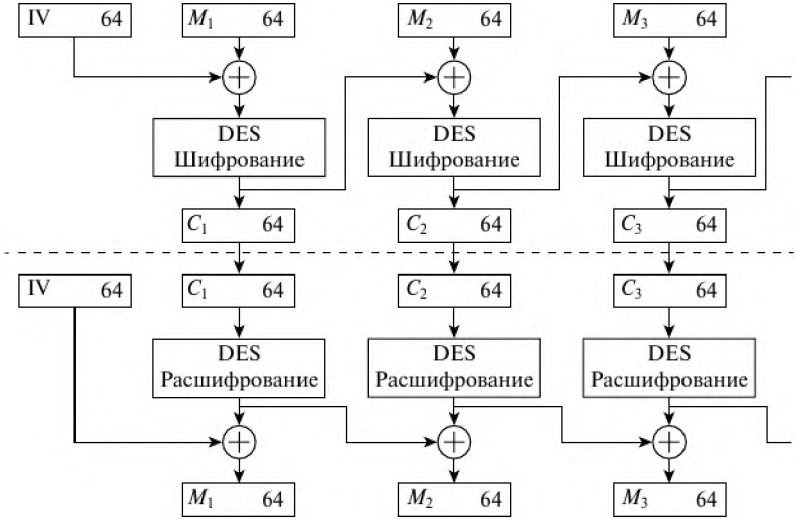


Рис. 2.15. Схема алгоритма DES в режиме сцепления блоков шифра

Таким образом, для всех i из $\{1, ..., n\}$ (n — число блоков) результат шифрования C_i определяется следующим образом:

$$C_i = \mathrm{DES}(M_i \oplus C_{i-1}),$$

где $C_0 = \mathrm{IV}$ — начальное значение шифра, равное начальному вектору (*вектору инициализации*).

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

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

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

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

Режим «Обратная связь по шифру». В этом (рис. 2.16) режиме размер блока длиной k битов может отличаться от 64 бит. Файл, подлежащий шифрованию (расшифрованию), считывается последовательными блоками длиной k битов, $k \le 64$).

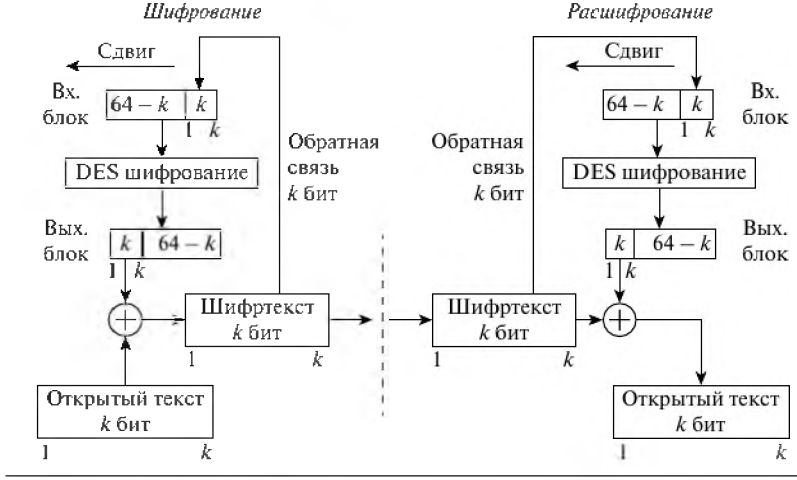


Рис. 2.16. Схема алгоритма DES в режиме обратной связи по шифртексту

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

Предположим, что в результате разбиения на блоки мы получили n блоков длиной k битов каждый (остаток дописывается нулями или пробелами). Тогда для любого i из {1, ..., n} блок шифротекста получается по правилу

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

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

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

Режим «Обратная связь по выходу». Этот режим (рис. 2.17) тоже использует переменный размер блока и сдвиговый регистр, инициализируемый так же, как в режиме CFB, а именно — входной блок вначале содержит вектор инициализации IV, выровненный по правому краю. При этом для каждого сеанса шифрования данных необходимо использовать новое начальное состояние регистра, которое должно пересылаться по каналу открытым текстом.

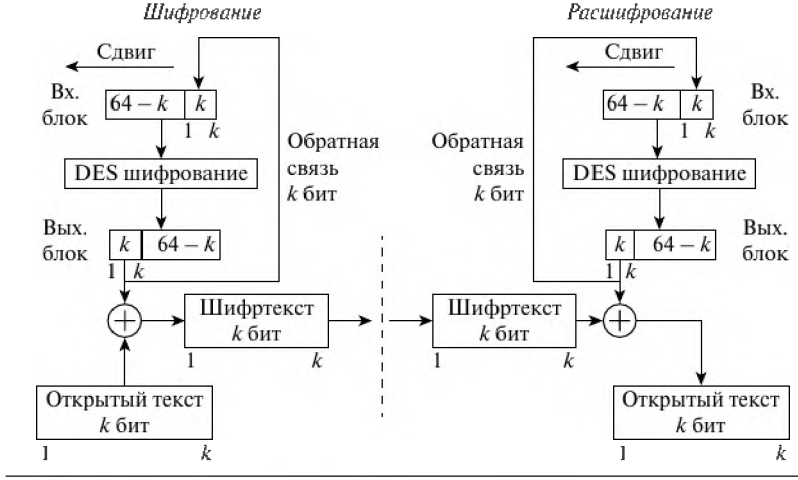


Рис. 2.17. Схема алгоритма DES в режиме обратной связи по выходу

Положим $M = M_1 M_2 ... M_n$. Для всех i из $\{1, ..., n\}$

$$C_i = M_i \oplus P_i,$$

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

Отличие от режима обратной связи по шифртексту состоит в методе обновления сдвигового регистра. Это осуществляется путем отбрасывания старших k битов и дописывания справа P_i.

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

Режим ECB хорошо подходит для шифрования ключей: режим CFB, как правило, предназначается для шифрования отдельных сим-

волов, а режим OFB нередко применяется для шифрования в спутниковых системах связи.

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

— интерактивного шифрования при обмене данными в телекоммуникационных сетях;
— шифрования криптографических ключей в практике автоматизированного распространения ключей;
— шифрования файлов, почтовых отправлений, данных спутников и других практических задач.

2.5. Отечественный стандарт шифрования данных

В нашей стране установлен единый алгоритм криптографического преобразования данных для систем обработки информации в телекоммуникационных сетях и отдельных вычислительных комплексах, который определяется ГОСТ 28147—89. Стандарт обязателен для организаций, предприятий и учреждений, применяющих криптографическую защиту данных.

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

При описании алгоритма используются следующие обозначения:

L и R — последовательности битов;

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

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

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

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

Примеры

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

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

представленные в двоичном виде, т.е.

$$a = a_{32} \cdot 2^{31} + a_{31} \cdot 2^{30} + ... + a_2 \cdot 2^1 + a_1,$$
$$b = b_{32} \cdot 2^{31} + b_{31} \cdot 2^{30} + ... + b_2 \cdot 2^1 + b_1,$$

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

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

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

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

Алгоритм предусматривает четыре режима работы:

— шифрование данных в режиме простой замены;
— шифрование данных в режиме гаммирования;
— шифрование данных в режиме гаммирования с обратной связью;
— выработка имитовставки.

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

N_1, N_2 — 32-разрядные накопители;

CM_1 — 32-разрядный сумматор по модулю 2^{32} ($\boxplus$);

CM_2 — 32-разрядный сумматор по модулю 2 ($\oplus$);

R — 32-разрядный регистр циклического сдвига;

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

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

Зашифрование открытых данных в режиме простой замены (рис. 2.18). Открытые данные, подлежащие зашифрованию, разбивают на 64-разрядные блоки T_0. Процедура зашифрования 64-разрядного блока T_0 в режиме простой замены включает 32 цикла ($j \in \{1, ..., 32\}$). В ключевое запоминающее устройство вводят 256 бит ключа K в виде восьми 32-разрядных подключей (чисел) K_i:

$$K = K_7 K_6 K_5 K_4 K_3 K_2 K_1 K_0.$$

Последовательность битов блока

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

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

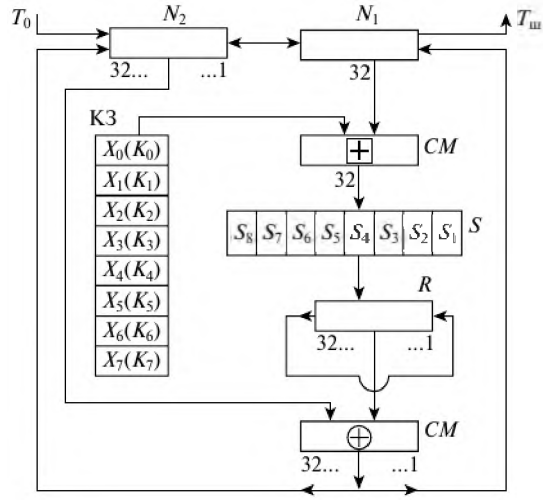


Рис. 2.18. Схема реализации режима простой замены

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

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

начальное заполнение накопителя N_2

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

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

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

Здесь $a(1)$ — заполнение N_1 после 1-го цикла зашифрования;

$b(1)$ — заполнение N_2 после 1-го цикла зашифрования;

f — функция шифрования.

Аргументом функции f является сумма по модулю 2^{32} числа $a(0)$ (начального заполнения накопителя N_1) и числа K_0 — подключа, считываемого из накопителя X_0 КЗУ. Каждое из этих чисел равно 32 битам. Функция f включает две операции над полученной 32-разрядной суммой $(a(0) \boxplus K_0)$.

Первая операция называется *подстановкой* (*заменой*) и выполняется блоком подстановки S. Блок подстановки S состоит из восьми узлов замены (S-блоков замены) S_1, S_2, ..., S_8 с памятью 64 бит каждый. Поступающий из CM_1 на блок подстановки S 32-разрядный вектор разбивают на восемь последовательно идущих 4-разрядных векторов, каждый из которых преобразуется в четырехразрядный вектор соответствующим узлом замены. Каждый узел замены можно представить в виде таблицы-перестановки шестнадцати четырехразрядных двоичных чисел в диапазоне 0000...1111. Входной вектор указывает адрес строки в таблице, а число в этой строке является выходным вектором. Затем четырехразрядные выходные векторы последовательно объединяют в 32-разрядный вектор. Узлы замены (таблицы-перестановки) представляют собой ключевые элементы, которые являются общими для сети и редко изменяются. Эти узлы замены должны сохраняться в секрете.

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

Далее результат работы функции шифрования f суммируют поразрядно по модулю 2 в сумматоре CM_2 с 32-разрядным начальным заполнением $b(0)$ накопителя N_2. Затем полученный на выходе CM_2 результат (значение $a(1)$) записывают в накопитель N_1, а старое значение N_1 (значение $a(0)$) переписывают в накопитель N_2 (значение $b(1) = a(0)$). Первый цикл завершен.

Последующие циклы осуществляются аналогично, при этом во втором цикле из КЗУ считывают заполнение X_1 — подключ K_1, в третьем цикле — подключ K_2 и т.д., в восьмом цикле — подключ K_7. В циклах с 9-го по 16-й, а также в циклах с 17-го по 24-й подключи из КЗУ считываются в том же порядке: K_0, K_1, K_2, ..., K_6, K_7. В последних восьми циклах с 25-го по 32-й порядок считывания подключей из КЗУ обратный: K_7, K_6, ..., K_2, K_1, K_0. Таким образом, при зашифровании в 32 циклах осуществляется следующий порядок выборки из КЗУ подключей:

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

В 32-м цикле результат из сумматора CM_2 вводится в накопитель N_2, а в накопителе N_1 сохраняется прежнее заполнение. Полученные после 32-го цикла зашифрования заполнения накопителей N_1 и N_2 являются блоком зашифрованных данных $T_ш$, соответствующим блоку открытых данных T_0.

Уравнения зашифрования в режиме простой замены имеют вид:

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

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

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

где

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

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

$$T_{\text{ш}} = (a_1(32), a_2(32), ..., a_{32}(32), b_1(32), b_2(32), ..., b_{32}(32)).$$

Остальные блоки открытых данных зашифровываются в режиме простой замены аналогично.

Зашифрование открытых данных в режиме гаммирования. Криптосхема, реализующая алгоритм зашифрования в режиме гаммирования, показана на рис. 2.19. Открытые данные разбивают на 64-разрядные блоки

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

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

Эти блоки поочередно зашифровываются в режиме гаммирования путем поразрядного сложения по модулю 2 в сумматоре CM_5 с гаммой шифра $\Gamma_{\text{ш}}$, которая вырабатывается блоками по 64 бита, т.е.

$$\Gamma_{\text{ш}} = (\Gamma_{\text{ш}}^{(1)}, \Gamma_{\text{ш}}^{(2)}, ..., \Gamma_{\text{ш}}^{(i)}, \Gamma_{\text{ш}}^{(m)}),$$

где $\Gamma_{\text{ш}}^{(i)}$ — i-й 64-разрядный блок, $i \in \{1, ..., m\}$.

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

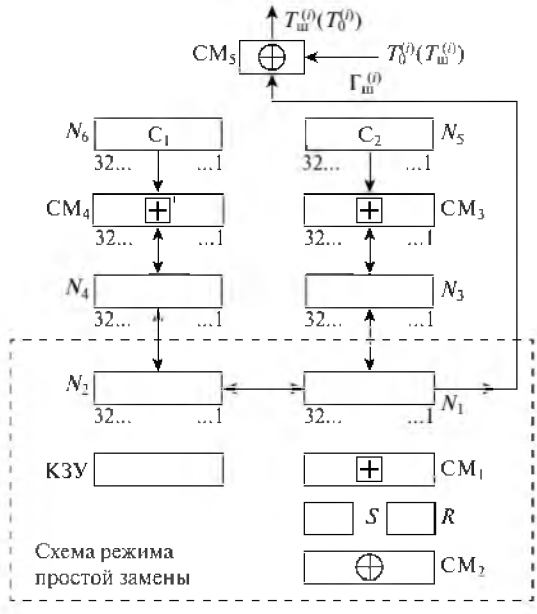


Рис. 2.19. Схема реализации режима гаммирования

Уравнение зашифрования данных в режиме гаммирования имеет вид

$$T_{ш}^{(i)} = T_0^{(i)} \oplus \Gamma_{ш}^{(i)},$$

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

Величины Y_i, Z_i определяются итерационно по мере формирования гаммы $\Gamma_{ш}$ следующим образом:

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

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

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

Рассмотрим реализацию процедуры зашифрования в режиме гаммирования.

В накопители N_6 и N_5 заранее записаны 32-разрядные двоичные константы C_1 и C_2, имеющие следующие значения (в шестнадцатеричной форме):

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

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

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

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

Исходное заполнение накопителя N_1:

$$(S_{32}, S_{31}, ..., S_2, S_1);$$
$$32,\ 31,\ ...,\ 2,\ 1 \quad \leftarrow \text{номер разряда } N_1$$

исходное заполнение накопителя N_2:

$$(S_{64}, S_{63}, ..., S_{34}, S_{33});$$
$$32,\ 31,\ ...,\ 2,\ 1 \quad \leftarrow \text{номер разряда } N_2$$

Исходное заполнение N_1 и N_2 (синхропосылка $\bar{S}$) зашифровывается в режиме простой замены. Результат зашифрования

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

переписывается в 32-разрядные накопители N_3 и N_4 так, что заполнение N_1 переписывается в N_3, а заполнение N_2 — в N_4.

Заполнение накопителя N_4 суммируют по модулю $(2^{32} - 1)$ в сумматоре CM_4 с 32-разрядной константой C_1 из накопителя N_6. Результат записывается в N_4. Заполнение накопителя N_3 суммируется по модулю 2^{32} в сумматоре CM_3 с 32-разрядной константой C_2 из накопителя N_5. Результат записывается в N_3. Заполнение N_3 переписывают в N_1, а заполнение N_4 — в N_2, при этом заполнения N_3, N_4 сохраняются. Заполнение накопителей N_1 и N_2 зашифровывается в режиме простой замены.

Полученное в результате зашифрования заполнение накопителей N_1, N_2 образует первый 64-разрядный блок гаммы шифра $\Gamma_{ш}^{(1)} = (\gamma_1^{(1)}, \gamma_2^{(1)}, ..., \gamma_{63}^{(1)}, \gamma_{64}^{(1)})$, который суммируют поразрядно по модулю 2 в сумматоре CM_5 с первым 64-разрядным блоком открытых данных

$$T_0^{(1)} = (t_1^{(1)}, t_2^{(1)}, ..., t_{63}^{(1)}, t_{64}^{(1)}).$$

В результате суммирования по модулю 2 значений $\Gamma_{ш}^{(1)}$ и $T_0^{(1)}$ получают первый 64-разрядный блок зашифрованнных данных:

$$T_{ш}^{(1)} = \Gamma_{ш}^{(1)} \otimes T_0^{(1)} = (\tau_1^{(1)}, \tau_2^{(1)}, ..., \tau_{63}^{(1)}, \tau_{64}^{(1)}),$$

где $\tau_i^{(1)} = t_i^{(1)} \oplus \gamma_i^{(1)}$, $i \in \{1, ..., 64\}$.

Для получения следующего 64-разрядного блока гаммы шифра $\Gamma_{ш}^{(2)}$ заполнение N_4 суммируется по модулю $(2^{32}-1)$ в сумматоре CM_4 с константой C_1 из N_6. Результат записывается в N_4. Заполнение N_3 суммируется по модулю 2^{32} в сумматоре CM_3 с константой C_2 из N_5. Результат записывается в N_3. Новое заполнение N_3 переписывают в N_1, а новое заполнение N_4 — в N_2, при этом заполнения N_3 и N_4 сохраняют. Заполнения N_1, N_2 зашифровывают в режиме простой замены.

Полученное в результате зашифрования заполнение накопителей N_1 и N_2 образует второй 64-разрядный блок гаммы шифра $\Gamma_{ш}^{(2)}$, который суммируется поразрядно по модулю 2 в сумматоре CM_5 со вторым блоком открытых данных $T_0^{(2)}$:

$$T_{ш}^{(2)} = \Gamma_{ш}^{(2)} \oplus T_0^{(2)}.$$

Аналогично вырабатываются блоки гаммы шифра $\Gamma_{ш}^{(3)}$, $\Gamma_{ш}^{(4)}$, ..., $\Gamma_{ш}^{(m)}$ и зашифровываются блоки открытых данных $T_0^{(3)}$, $T_0^{(4)}$, ..., $T_0^{(m)}$.

В канал связи или память передаются синхропосылка $\tilde{S}$ и блоки зашифрованных данных $T_{ш}^{(1)}$, $T_{ш}^{(2)}$, ..., $T_{ш}^{(m)}$.

Режим гаммирования с обратной связью. Зашифрование открытых данных в режиме гаммирования с обратной связью. Криптосхема, реализующая алгоритм зашифрования в режиме гаммирования с обратной связью, имеет вид, показанный на рис. 2.20.

Открытые данные, разбитые на 64-разрядные блоки $T_0^{(1)}$, $T_0^{(2)}$, ..., $T_0^{(m)}$, зашифровываются в режиме гаммирования с обратной связью путем поразрядного сложения по модулю 2 с гаммой шифра $\Gamma_{ш}$, которая вырабатывается блоками по 64 бита:

$$\Gamma_{ш} = (\Gamma_{ш}^{(1)}, \Gamma_{ш}^{(2)}, ..., \Gamma_{ш}^{(m)}).$$

Число двоичных разрядов в блоке $T_0^{(m)}$ может быть меньше 64, при этом неиспользованная для шифрования часть гаммы шифра из блока $\Gamma_{ш}^{(m)}$ отбрасывается. Уравнения зашифрования в режиме гаммирования с обратной связью имеют вид:

$$T_{ш}^{(1)} = A(\tilde{S}) \oplus T_0^{(1)} = \Gamma_{ш}^{(1)} \oplus T_0^{(1)},$$
$$T_{ш}^{(i)} = A(T_{ш}^{(i-1)}) \oplus T_0^{(i)} = \Gamma_{ш}^{(i)} \oplus T_0^{(i)}, \quad i \in \{2, ..., m\}.$$

Здесь $T_{ш}^{(i)}$ — i-й 64-разрядный блок зашифрованного текста; $A(\cdot)$ — функция зашифрования в режиме простой замены; m — определяется объемом открытых данных.

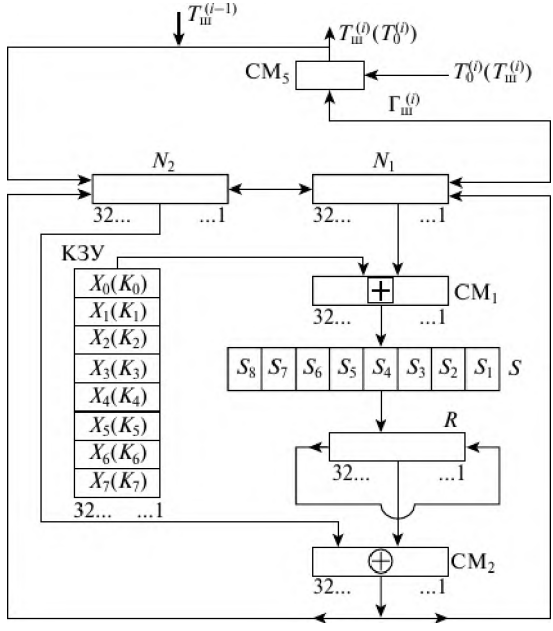


Рис. 2.20. Схема реализации режима гаммирования с обратной связью

Аргументом функции $A(\cdot)$ на первом шаге итеративного алгоритма является 64-разрядная синхропосылка $\tilde{S}$, а на всех последующих шагах — предыдущий блок зашифрованных данных $T_{ш}^{(i-1)}$.

Процедура зашифрования данных в режиме гаммирования с обратной связью реализуется следующим образом. В КЗУ вводятся 256 бит ключа. В накопители N_1 и N_2 вводится синхро-посылка $\tilde{S} = (S_1, S_2, ..., S_{64})$ из 64 бит. Исходное заполнение накопителей N_1 и N_2 зашифровывается в режиме простой замены. Полученное в результате зашифрования заполнение накопителей N_1 и N_2 образует первый 64-разрядный блок гаммы шифра $\Gamma_{ш}^{(1)} = A(\tilde{S})$, который суммируется поразрядно по модулю 2 в сумматоре CM_5 с первым 64-разрядным блоком открытых данных

$$T_0^{(1)} = (t_1^{(1)}, t_2^{(1)}, ..., t_{64}^{(1)}).$$

В результате получают первый 64-разрядный блок зашифрованных данных

$$T_{ш}^{(1)} = \Gamma_{ш}^{(1)} \oplus T_0^{(1)},$$

где $T_{ш}^{(1)} = (\tau_1^{(1)}, \tau_2^{(1)}, ..., \tau_{64}^{(1)})$.

Блок зашифрованных данных $T_{ш}^{(1)}$ одновременно является также исходным состоянием накопителей N_1, N_2 для выработки второго блока гаммы шифра $\Gamma_{ш}^{(2)}$, и поэтому по обратной связи $T_{ш}^{(1)}$ записывается в указанные накопители N_1 и N_2.

Заполнение накопителя N_1

$$(\tau_{32}^{(1)}, \tau_{31}^{(1)}, ..., \tau_{2}^{(1)}, \tau_{1}^{(1)}).$$
$$32,\ 31, ...,\ 2,\ \ 1 \quad \leftarrow \text{номер разряда } N_1$$

Заполнение накопителя N_2

$$(\tau_{64}^{(1)}, \tau_{63}^{(1)}, ..., \tau_{34}^{(1)}, \tau_{33}^{(1)}).$$
$$32,\ 31, ...,\ 2,\ \ 1 \quad \leftarrow \text{номер разряда } N_2$$

Заполнение накопителей N_1 и N_2 зашифровывается в режиме простой замены.

Полученное в результате зашифрования заполнение накопителей N_1 и N_2 образует второй 64-разрядный блок гаммы шифра $\Gamma_{ш}^{(2)}$, который суммируется поразрядно по модулю 2 в сумматоре CM_3 со вторым блоком открытых данных $T_0^{(2)}$:

$$\Gamma_{ш}^{(2)} \oplus T_{0}^{(2)} = T_{ш}^{(2)}.$$

Выработка последующих блоков гаммы шифра $\Gamma_{ш}^{(i)}$ и зашифрование соответствующих блоков открытых данных $T_0^{(i)}$ ($i \in \{3, ..., m\}$) производится аналогично.

Если длина последнего m-го блока открытых данных $T_0^{(m)}$ меньше 64 разрядов, то из $\Gamma_{ш}^{(m)}$ используется только соответствующее число разрядов гаммы шифра, остальные разряды отбрасываются.

В канал связи или память ЭВМ передаются синхропосылка $\tilde{S}$ и блоки зашифрованных данных $T_{ш}^{(1)}$, $T_{ш}^{(2)}$, ..., $T_{ш}^{(m)}$.

Режим выработки имитовставки. Имитовставка — это блок из P бит, который вырабатывают по определенному правилу из открытых данных с использованием ключа и затем добавляют к зашифрованным данным для обеспечения их имитозащиты.

Имитозащита — это защита системы шифрованной связи от навязывания ложных данных.

В стандарте ГОСТ 28147—89 определяется процесс выработки имитовставки, который единообразен для любого из режимов шифрования данных. Имитовставка $И_р$ вырабатывается из блоков открытых

данных либо перед шифрованием всего сообщения, либо параллельно с шифрованием по блокам. Первые блоки открытых данных, которые участвуют в выработке имитовставки, могут содержать служебную информацию (например, адресную часть, время, синхропосылку) и не зашифровываются.

Значение параметра P (число двоичных разрядов в имитовставке) определяется криптографическими требованиями с учетом того, что вероятность навязывания ложных помех равна $1/2^p$.

Для выработки имитовставки открытые данные представляют в виде последовательности 64-разрядных блоков $T_0^{(i)}$, $i \in \{1, ..., m\}$.

Первый блок открытых данных $T_0^{(1)}$ подвергают преобразованию $\tilde{A}(\cdot)$, соответствующему первым 16 циклам алгоритма зашифрования в режиме простой замены. В качестве ключа для выработки имитовставки используют ключ длиной 256 бит, по которому шифруют данные.

Полученное после 16 циклов 64-разрядное число $\tilde{A}(T_0^{(1)})$ суммируют по модулю 2 со вторым блоком открытых данных $T_0^{(2)}$. Результат суммирования $(\tilde{A}(T_0^{(1)}) \oplus T_0^{(2)})$ снова подвергают преобразованию $\tilde{A}(\cdot)$.

Полученное 64-разрядное число $\tilde{A}\ (\tilde{A}(T_0^{(1)}) \oplus T_0^{(2)})$ суммируют по модулю 2 с третьим блоком $T_0^{(3)}$ и снова подвергают преобразованию $\tilde{A}(\cdot)$, получая 64-разрядное число $\tilde{A}\ (\tilde{A}(\tilde{A}(T_0^{(1)}) \oplus T_0^{(2)}) \oplus T_0^{(3)})$, и т.д.

Последний блок $T_0^{(m)}$ (при необходимости дополненный нулями до полного 64-разрядного блока) суммируют по модулю 2 с результатом вычислений на шаге $(m-1)$, после чего зашифровывают в режиме простой замены, используя преобразование $\tilde{A}(\cdot)$.

Из полученного 64-разрядного числа выбирают отрезок $И_p$ (имитовставку) длиной P бит:

$$И_p = [a_{32-p+1}^{(m)}(16), a_{32-p+2}^{(m)}(16), ..., a_{32}^{(m)}(16)],$$

где $a_i^{(m)}$ — i-й бит 64-разрядного числа, полученного после 16-го цикла последнего преобразования $\tilde{A}(\cdot)$, $32 - p + 1 \le i \le 32$.

Имитовставка $И_p$ передается по каналу связи или в память в конце зашифрованных данных, т.е.

$$T_ш^{(1)}, T_ш^{(2)}, ..., T_ш^{(m)}, И_p.$$

Поступившие к получателю зашифрованные данные

$$T_{\text{ш}}^{(1)}, T_{\text{ш}}^{(2)}, \ldots, T_{\text{ш}}^{(m)}$$

расшифровываются и из полученных блоков открытых данных $T_0^{(1)}$, $T_0^{(2)}$, ..., $T_0^{(m)}$ аналогичным образом вырабатывается имитовставка $И'_p$. Эта имитовставка $И'_p$ сравнивается с имитовставкой $И_p$, полученной вместе с зашифрованными данными из канала связи или из памяти. В случае несовпадения имитовставок полученные при расшифровании блоки открытых данных $T_0^{(1)}$, $T_0^{(2)}, \ldots, T_0^{(m)}$ считают ложными.

Сравнивая схемы DES и ГОСТ, следует заметить, что они очень похожи, но есть и существенные отличия:

— в ГОСТе проводится в 2 раза большее число итераций, определяющих криптографическую сложность результирующих преобразований;

— в ГОСТе существенно больше ключей ($2^{56} = 6.4 \cdot 10^{16}$ вариантов ключевых установок в DES и $2^{256} = 6.4 \cdot 10^{76}$ ключевых установок в ГОСТ).

Обе схемы не являются теоретически стойкими. При достаточном количестве шифрованного текста тотальным методом, т.е. перебором всех ключей, проведением пробного расшифрования и отсева по статистическим критериям ложных вариантов получаемого открытого текста, можно найти ключ для обоих шифров.

Тест к главе 2

1. ***Преобразование открытого текста сообщения в закрытый называется*:**
   1) процедура шифрования;
   2) алгоритм шифрования;
   3) обеспечение аутентификации;
   4) цифровая запись.
2. ***Входные параметры процесса шифрования (несколько верных ответов)*:**
   1) зашифрованный текст;
   2) ключ;
   3) открытый текст;
   4) алгоритм.

3. ***Какие из сервисов реализуются при использовании криптографических преобразований (несколько верных ответов)*:**
   1) контроль целостности;
   2) аутентификация;
   3) шифрование;
   4) алгоритм.
4. ***Что позволяет предотвратить использование криптографических преобразований*:**
   1) отказ от информации;
   2) обеспечение аутентификации;
   3) утечку информации;
   4) использование алгоритмов асимметричного шифрования.
5. ***Знание ключа позволяет*:**
   1) использовать криптографические сервисы безопасности;
   2) обеспечить аутентификацию;
   3) предотвратить утечку информации;
   4) выполнить обратное преобразование.
6. ***Что в криптографии понимается под термином «элементарное опробование»*:**
   1) операция над двумя n-разрядными двоичными числами;
   2) проверка ключа на целостность;
   3) сопоставление двух паролей;
   4) передача ключа по какому-либо каналу связи.
7. ***Чем определяется уровень надежности применяемых криптографических преобразований*:**
   1) значением допустимой вероятности неисправностей или сбоев, приводящих к получению злоумышленником дополнительной информации о криптографических преобразованиях;
   2) сложностью комбинации символов, выбранных случайным образом;
   3) использованием большого числа ключей для шифрования;
   4) отношением количества дешифрованной информации к общему количеству шифрованной информации, подлежащей дешифрованию.
8. ***Ниже перечислены механизмы защиты информационных систем от несанкционированного доступа. Что здесь лишнее*:**
   1) идентификация и аутентификация пользователей и субъектов доступа;
   2) управление доступом;
   3) обеспечение постоянного числа пользователей сети;

4) обеспечения целостности;
5) регистрация и учет.

9. ***Что называется имитовставкой*:**
1) это блок данных, переменной длины, который вырабатывают по определенному правилу из открытых данных с использованием ключа и затем добавляют к зашифрованным данным для обеспечения их имитозащиты;
2) это блок данных фиксированной длины, который вырабатывают по определенному правилу из открытых данных с использованием ключа и затем добавляют к зашифрованным данным для обеспечения их имитозащиты.

10. ***Как иначе называется симметричное шифрование*:**
1) шифрование с закрытым ключом;
2) шифрование методом Бейтса;
3) шифрование с открытым ключом;
4) шифрование с переменным ключом.

11. ***Какой алгоритм не используется при симметричном шифровании*:**
1) поточное шифрование;
2) побитовое шифрование;
3) блочное шифрование;
4) алгоритм Эль-Гамаля.

12. ***Какой из режимов алгоритма DES используется для построения шифров гаммирования?***
1) электронная кодовая книга;
2) сцепление блоков шифра;
3) обратная связь по шифротексту;
4) обратная связь по выходу.

13. ***Какова длина блока алгоритма шифрования DES*:**
1) 16 бит;
2) 56 бит;
3) 64 бита;
4) 5 байт.

14. ***Сколько всего циклов выполняется операция зашифровывания в алгоритме DES*:**
1) 10;
2) 14;
3) 16;
4) 20.

15. ***Что является преимуществом симметричного шифрования*:**
1) скорость выполнения криптографических преобразований;
2) легкость внесения изменений в алгоритм шифрования;

3) секретный ключ известен только получателю информации и первоначальный обмен не требует передачи секретного ключа;
4) применение в системах аутентификации (электронная подпись).

16. ***Какой размер ключа в отечественном стандарте симметричного шифрования*:**
1) 56 бит;
2) 124 бит;
3) 256 бит.

17. ***Какие из режимов шифрования данных не включает в себя отечественный стандарт симметричного шифрования*:**
1) режим гаммировани;
2) режим простой замены;
3) режим обратной связи по шифротексту;
4) режим гаммирования с обратной связью.

18. ***Использует ли отечественный стандарт симметричного шифрования дополнительный ключ*:**
1) да;
2) нет.

19. ***Какое из этих утверждений является верным*:**
1) у *S*-блоков ГОСТ 4-битовые входы и выходы;
2) у *S*-блоков ГОСТ 4-битовые входы и 8-битовые выходы;
3) у *S*-блоков ГОСТ 8-битовые входы и 4-битовые выходы.

20. ***Используется ли в отечественном стандарте симметричного шифрования процедура генерации подключей из ключей, как в DES*:**
1) да, но эта процедура сравнительно проста;
2) не используется;
3) используется аналогичная по сложности процедура.

21. ***В отечественном стандарте симметричного шифрования применяется подстановка, основанная на применении S-блоков. Сколько таких блоков используется в ГОСТ*:**
1) 8;
2) 12;
3) 16;
4) 24.

22. ***Длина раундового ключа в отечественном стандарте симметричного шифрования*:**
1) 8 бит;
2) 32 бита;
3) 48 бит.

23. ***Выберите правильное утверждение*:**

1) в отечественном стандарте симметричного шифрования есть начальная, но нет конечной битовых перестановок шифруемого блока;
2) в отечественном стандарте симметричного шифрования нет начальной и конечной битовых перестановок шифруемого блока, так как они не влияют на стойкость шифра;
3) в DES нет начальной и конечной битовых перестановок шифруемого блока.

24. ***Что означает «многократное шифрование» применительно к блочным шифрам*:**

1) повторное применение алгоритма шифрования к шифротексту с теми же ключами;
2) шифрование одного и того же блока открытого текста несколько раз с несколькими ключами;
3) повторное применение алгоритма шифрования к шифротексту с другими ключами;
4) увеличение числа этапов шифрования открытого текста.

Таблица ответов на тест к главе 2

Номер вопроса	1	2	3	4	5	6	7	8
Правильный ответ	1	2, 3	1, 2	3	4	1	1	3
Номер вопроса	9	10	11	12	13	14	15	16
Правильный ответ	2	1	4	4	3	3	1	3
Номер вопроса	17	18	19	20	21	22	23	24
Правильный ответ	3	1	1	1	1	2	2	2

ГЛАВА 3

ЭЛЕМЕНТЫ КРИПТОАНАЛИЗА КЛАССИЧЕСКИХ ШИФРОВ

3.1. Открытые сообщения и их простейшие характеристики

Понятие открытого сообщения в криптографической литературе понимается двояко: либо это содержательный текст, поддающийся смысловому чтению, либо это текст, подлежащий зашифрованию, в последнем случае это текст, возможно, и нечитаемый, например, шифрованный текст, в случае двойного перешифрования. В данном параграфе под открытым текстом, как правило, мы понимаем содержательный читаемый текст на каком-либо языке. Модели шифров, криптосхем, шифраторов, модели ключевых систем шифров строятся с использованием моделей открытого текста (или моделей источника открытых сообщений). Все модели, в том числе и модели открытых сообщений, обычно делятся на два класса: детерминированные и вероятностные.

Детерминированный источник сообщений. В этой модели открытые тексты (как и шифрованные) представляют собой последовательности символов, взятых из конечного множества символов, называемого алфавитом открытого текста. Например, алфавит русского языка, алфавит английского языка.

Число символов в алфавите математики называют мощностью алфавита. Например, алфавит $A(1) = \{A, B, C, D, ..., X, Y, Z\}$ — прописные буквы английского языка. Мощность алфавита 26 (иногда вместо пробела используют букву Z). Алфавит $A(2) = \{A, B, C, D, ..., X, Y, Z, a, b, c, ..., x, y, z, 0, 1, 2, ..., 9, ... « » ? !\}$ — мощность алфавита 70. Алфавит $A(3) = \{0, 1\}$ — мощность 2. Часто используются алфавиты, представляющие собой двоичные наборы длиной n (как правило $5 \le n \ge 8$) или двоичные коды, например, международный телеграфный код (МТК-2). Полный русский алфавит состоит из 33 букв:

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

пробела, точки, запятой.

Отождествляют: в ряде случаев Е и Ё, иногда И и Й, а иногда Е и Э, часто отождествляют Ь и Ъ. Добавляя пробел, говорят, что алфавит состоит из 34 букв, а с некоторыми отождествлениями букв алфавит содержит от 28 до 33 букв.

Пусть I — некоторый алфавит, мощности $|I|$. Текст, записанный в алфавите I, имеет длину — число символов в соответствующей последовательности. Последовательность k символов называют k-граммой в алфавите I. Математики, как правило, под последовательностью обычно понимают бесконечную последовательность символов алфавита I, конечную же последовательность называют словом в алфавите I. Собственно источниками открытого текста является отдельный человек или группа людей, радиопередающие станции, пункты телеграфной и телефонной сети и т.д. Каждый источник открытого текста (сообщений) характеризуется своими особенностями: используемым алфавитом, например, русским алфавитом; определенной структурой тематики сообщений, например, о погоде, о политике; математический текст, физический и т.д.; частотными характеристиками сообщений и другими особенностями, например, так называемыми вероятными словами: «Сообщаю Вам», «Докладываю», «На Ваш номер „...“ сообщаю», «старший оперуполномоченный провинции Логар майор Тарасов» и т.д. Сообщения на английском языке передаваемые по телетайпу, скорее всего используют алфавит $A(2)$. Частный корреспондент, намеревающийся шифровать свои сообщения, во многих случаях предпочтет использовать алфавит $A(1)$. Данные, передаваемые в телекоммуникационных сетях, удобнее отображать с использованием алфавита $A(3)$.

Каждый источник сообщений порождает тексты в соответствии с правилами грамматики, что находит отражение и в других характеристиках сообщений. Например, в содержательных текстах на английском языке за буквой «q» всегда следует буква «u», в русских текстах буквы «ь» и «ъ» никогда не располагаются рядом и не следуют за гласными буквами. Всякий источник сообщений можно моделировать списком *допустимых* (т.е. встречающихся в каких-либо текстах) k-грамм при $k = 1, 2, 3, \ldots$ Если k-грамма не является допустимой, то ее называют *запретной*, или *запрещенной*.

Детерминированная модель источника открытых сообщений. Разделение множества k-грамм на допустимые и запретные определяет детерминированную модель источника открытых сообщений. В такой модели открытый текст рассматривается как последовательность символов некоторого алфавита, не содержащую запретных k-грамм. Построение детерминированной модели исследуемого источника от-

крытых сообщений можно реализовать в результате статистической обработки генерируемых им текстов. Для этого следует «просмотреть» достаточно большое количество текстов, сгенерированного данным источником, и все не встретившиеся k-граммы отнести к множеству запретных k-грамм источника. Естественно, чем большее количество материала отработано, тем эффективнее применение построенной модели сообщений на практике для решения различных криптографических задач. В ряде криптографических задач данная модель источника сообщений используется для различения открытых текстов от случайных последовательностей с помощью вычислительной техники.

Источник передачи данных. Появление систем телеобработки привело к появлению нового вида связи, так называемого «передача данных». Целью передачи данных является передача информации для обработки ее вычислительным машинам или же выдача ее этими машинами. Принципиальная новизна вида связи — передачи данных состоит в том, что эта связь осуществляет обмен информацией между компьютерами, а также между компьютерами и человеком. Данные, предназначенные для машин, называют «формализованным языком», языком машин. Этим подчеркивается, что они не предназначены непосредственно для восприятия человеком. Эти данные передаются в цифровом виде (часто в виде двоичной последовательности). Осмысливание их человеком может происходить только после их представления в соответствующей форме. В криптографических терминах понятия формализованного языка представляют собой словарные величины, а их условные формы — кодобозначения, последние изображаются в виде буквенных, цифровых и смешанных групп различной длины (разрядности). Формализованный документ оформляется в виде так называемого «формата», т.е. формы, в которой размещение данных осуществляется по некоторым жестким правилам на местах, определяемых для данного формата шаблоном. Таким образом, для чтения таких документов необходимо знать формальный язык и форматы документов. Можно сказать, что фактически открытое сообщение в формализованном языке представляет собой кодограмму, а перевод открытого сообщения в формализованный язык есть позиционное кодирование смысловых сообщений нескольких языков, каждый из которых кодируется своим кодом и располагается на определенном месте формата сообщения. Таким образом, для формализованных сообщений исчезает понятие открытого текста в общепринятом его понимании «читаемого» текста. Признаками «открытого текста» текста формализованного являются не его читаемость, а различные его детерминированные и статистические признаки, связанные с применяе-

мыми способами сжатия и кодирования в системах дискретного фототелеграфа, телевидения, телекоммуникационных сетей.

Простейшие вероятностные источники сообщений. В этих моделях источник открытого текста рассматривается как источник случайных последовательностей. Считается, что источник генерирует конечную или бесконечную последовательность случайных символов $x(1)$, $x(2)$, ..., $x(n)$ из алфавита I. Вероятность случайного сообщения «$i(1)$, $i(2)$, ..., $i(n)$» определяется как вероятность совместного события

$$P(i(1), i(2), ..., i(n)) = P(x(1) = i(1), x(2) = i(2), ..., x(n) = i(n))).$$

При этом, естественно, требуют выполнения условий:

1) для любого случайного сообщения «$i(1)$, $i(2)$, ..., $i(n)$»

$$P(i(1), i(2), ..., i(n)) \geq 0;$$

2) $$\sum_{i(1), i(2), ..., i(n)} P(i(1), i(2), ..., i(n)) = 1;$$

3) для любого случайного сообщения «$i(1)$, $i(2)$, ..., $i(n)$»

$$P(i(1), i(2), ..., i(n)) = \sum_{i(1), i(2), ..., i(s)} P(i(1), i(2), ..., i(s)),\ s \geq n + 1.$$

Смысл последнего условия состоит в том, что вероятность всякого случайного сообщения длины n есть сумма вероятностей всех «продолжений» этого сообщения до длины $s > n$ (некоторый вариант аксиомы Колмогорова). Текст, порождаемый таким источником, является вероятностным аналогом языка. Он обладает одинаковыми с языком частотными характеристиками k-грамм. Задавая конкретное вероятностное распределение на множестве открытых текстов, мы задаем соответствующую модель источника сообщений. Рассмотрим некоторые частные случаи этой общей модели.

Стационарный источник независимых символов алфавита. В этой модели предполагается, что вероятности сообщений полностью определяются вероятностями отдельных символов алфавита:

$$P(i(1), i(2), ..., i(n)) = \prod_{j=1}^{n} P(x(j) = i(j))$$

$$\text{и} \quad P(x(j) = i) > 0,\ \sum_{i \in I} P(x(j) = i) = 1.$$

Под открытым текстом понимается реализация последовательности независимых испытаний в полиномиальной вероятностной схеме с числом исходов $|I| = m$. Исходу взаимно однозначно соответствует символ алфавита I. Эта модель позволяет разделить буквы алфавита

на классы высокой, средней и низкой частот использования. Ниже, в табл. 3.1, приводятся буквы высокой частоты использования для некоторых европейских языков (частота указана в процентах).

Таблца 3.1

Язык	Буквы алфавитов и частоты их использования в текстах											
Английский	E	12,86	*T*	9,72	*A*	7,96	*I*	7,77	*N*	7,51	*R*	7,03
Испанский	E	14,15	*A*	12,90	*O*	8,84	*S*	7,64	*I*	7,01	*R*	6,95
Итальянский	*I*	12,04	E	11,60	*A*	11,10	*O*	8,92	*N*	7,68	*T*	7,07
Немецкий	*E*	19,18	*N*	10,20	*I*	8,21	*S*	7,07	*R*	7,01	*T*	5,86
Французский	*E*	17,76	*S*	8,23	*A*	7,68	*N*	7,61	*T*	7,30	*I*	7,23
Русский	*O*	11,00	И	8,90	Е	8,30	А	7,90	Н	6,90	Т	6,00

Для сравнения частот редких букв и букв, приведенных в таблице, укажем, что, например, в английском языке редкими буквами являются буквы *J*, *Q*, *Z*, а их частоты в процентах оцениваются величинами 0,13; 0,12; 0,08 соответственно. Из этой таблицы видно, что не случайно итальянский и испанский языки считаются певучими: на долю гласных приходится около половины всех букв. Самыми частыми биграммами в русском языке являются (в процентах) СТ (1,74), НО (1,29), ЕН (1,23), ТО (1,21), НА (1,20), ОВ (1,16), НИ (1,15), РА (1,14), ВО (1,08), КО (1,07). Наиболее частые триграммы: СТО, ЕНО, НОВ, ТОВ, ОВО, НАЛ, РАЛ, НИС.

Рассматриваемая модель открытого текста весьма просто строится для любого источника открытых сообщений с использованием относительно небольшого количества материала и удобна для практического применения. В то же время, некоторые свойства модели противоречат свойствам языков. В частности, согласно этой модели любая k-грамма, $k > 1$, имеет ненулевую вероятность появления в сообщении.

3.2. Дешифрование некоторых классических шифров

Устойчивые закономерности открытого текста и их использование при дешифровании шифров простой замены и перестановки. Возможность дешифрования какого либо шифра в значительной мере зависит от того, в какой степени криптографические преобразования разрушают вероятностно-статистические закономерности, присутствующие в открытом содержательном тексте. Так в осмысленных текстах любо-

го естественного языка различные буквы встречаются с разной частотой, при этом относительные частоты букв в различных текстах одного языка близки между собой. То же самое можно сказать и о частотах пар, троек букв открытого текста. Кроме того, любой естественный язык обладает так называемой избыточностью, что позволяет с большой вероятностью «угадывать» смысл сообщения, даже если часть букв в сообщении не известна.

В табл. 3.2 приведены относительные частоты букв алфавита русского языка.

Таблица 3.2

1	а — 0,062	12	л — 0,035	23	ц — 0,004
2	б — 0,014	13	м — 0,026	24	ч — 0,012
3	в — 0,038	14	н — 0,053	25	ш — 0,006
4	г — 0,013	15	о — 0,090	26	щ — 0,003
5	д — 0,025	16	п — 0,023	27	ы — 0,016
6	е, ё — 0,072	17	р — 0,040	28	ъ, ь — 0,014
7	ж — 0,077	18	с — 0,045	29	э — 0,003
8	з — 0,016	19	т — 0,053	30	ю — 0,006
9	и — 0,062	20	у — 0,021	31	я — 0,018
10	й — 0,010	21	ф — 0,002	32	— 0,175
11	к — 0,28	22	х — 0,009		

Подобные статистические таблицы приводятся в разных книгах. Они получены на основе подсчетов частот на больших объемах открытого текста. Учитывая, что для экспериментов берется различный исходный материал, значения вероятностей несколько отличаются между собой.

Если упорядочить буквы по убыванию вероятностей, то мы получим вариационный ряд

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

В слове **СЕНОВАЛИТР** содержатся 10 наиболее частых букв.

Частоты знаков алфавита зависят не только от языка, но и от характера текста. Так в тексте по криптографии будет повышена вероятность букв Ф, Ш (из-за часто встречающихся слов «шифр», «криптография»). В некоторых математических текстах может быть завышена частота буквы Ф (из-за слов «функция». «функционал» и т.п.). В стандартных текстовых файлах наиболее частым является символ «про-

бел». Частотная диаграмма содержательных текстов является устойчивой характеристикой текста. Из теории вероятностей следует, что при достаточно слабых ограничениях на вероятностные свойства случайного процесса справедлив закон больших чисел, т.е. относительные частоты $\frac{\vartheta_k}{N}$ знаков сходятся по вероятности к значениям их вероятностей p_k

$$P\left\{\left|\frac{\vartheta_k}{N}-p_k\right|>\varepsilon\right\}\xrightarrow[N\to\infty]{}0.$$

Шифры перестановки и простой замены неполностью разрушают вероятностно-статистические свойства, имеющиеся в открытом сообщении.

Дешифрование шифра простой замены. При дешифровании текста, зашифрованного шифром простой замены, используют частотные характеристики открытого текста. Именно, если подсчитать частоты встречаемости знаков в шифрованном тексте, упорядочить их по убыванию и сравнить с вариационным рядом вероятностей открытого текста, то эти две последовательности будут близки. Скорее всего на первом месте окажется пробел, далее будут следовать буквы О, Е, А, И.

Конечно, если текст не очень длинный, то не обязательно полное совпадение. Может оказаться на втором месте О, а на третьем Е, но в любом случае в первых и вторых рядах одинаковые буквы будут располагаться недалеко друг от друга, и чем ближе к началу (чем больше вероятность знаков), тем меньше будет расстояние между знаками.

Аналогичная картина наблюдается и для пар соседних букв (биграмм) открытого текста (наиболее частая биграмма русского открытого текста — СТ). Однако для получения устойчивой картины длина последовательности должна быть существенно больше. На сравнительно небольших отрезках открытого текста эта картина как-то смазана. Более устойчивой характеристикой биграмм является отсутствие в осмысленном тексте некоторых биграмм, как говорят, наличие запретных биграмм, имеющих вероятность равную практически 0.

Видели ли вы когда-нибудь в открытом тексте биграмму ЪЬ или биграммы вида: «гласная» Ь; «пробел» Ь? Знание и использование указанных особенностей открытого текста значительно облегчает дешифрование шифра простой замены.

Используя указанные свойства открытых текстов, под каждой буквой шифрованного текста строится колонка возможных букв открытого текста, соответствующая данной шифрованной букве. Буквы в колонках ранжируются по вероятности. Наиболее вероятные буквы ставятся выше менее вероятных букв. Далее стараются набрать осмысленный текст, выбирая в колонках по одной букве.

О степени неоднозначности восстановления открытого текста. При уменьшении материала задача дешифрования существенно усложняется. Действительно, представьте себе, что вам предложено дешифровать зашифрованный с помощью простой замены шифртекст, состоящий из одного первого слова ДОЧАЛЬ. Для каждого слова из шести различных букв найдется простая замена, при применении которой мы получим данный шифртекст. Проблема заключается не в том, чтобы найти ключевую подстановку, а в том, как выбрать из обширного множества вариантов восстановленных открытых текстов именно тот, который был зашифрован. Таким образом, важной характеристикой эффективности криптографической защиты является степень неоднозначности восстановления открытого текста по шифрованному.

Дешифрование шифра перестановки. При известном порядке d ключевой подстановки задача дешифрования сводится к следующим действиям:

1) к записи шифрованного текста в таблицу с d колонками;
2) поиску подстановки с помощью которой, переставив столбцы таблицы, получится искомый открытый текст, записанный в таблице.

Остановимся сначала на решении второй задачи.

Рассмотрим пример дешифрования шифра перестановки восьми столбцов. Пусть шифртекст имеет следующий вид (табл. 3.3).

Сопоставим перестановке столбцов таблицу 8×8, при этом поставим на пересечении i-й строки и j-го столбца единицу, если j-я колонка после обратной перестановки должна следовать за i-й. Наша задача — восстановить таблицу, отвечающую правильной перестановке столбцов.

Давайте теперь попарно пристраивать один столбец к другому. Если при этом в некоторых строках появляются запретные биграммы, то столбцы не могут в открытом тексте следовать друг за другом, и соответствующая клеточка зачеркиваются. В нашем примере 6-й столбец не может следовать за 4-м, так как иначе в тексте в первой строке будет подряд два пробела. Посмотрим, например, 6-ю строку. Если бы 4-йстолбец следовал за 1-м, то в тексте были бы слова, начинающиеся с Ь.

Таблица 3.3

1	2	3	4	5	6	7	8
п	а	я		в		и	м
о	ч	ш	г		у		е
е	б	ж	л		е		о
м		ч		о	т	о	я
	е	г	е		у	с	щ
	а	к	ь	з	а	т	т
я	р	е		е	п		ь
	ю	з	в	а	н	в	
о	й	а	в	е	ш	л	
	е	е	я	м		п	н
ь	р	р	н	з	е	е	е
з	а	м	а	н		а	к
ч	с	т	а		ь	а	н
о	я	л	м		а	л	
о	ь	ч	х	т	а	т	
в		е	о	а	л	е	п
о	е	р	м	т	ь	е	
д	с	г	ы		о	а	т
е	б	в	н		ы		
	а	у	и	н	з	н	л
	г	и	а	о	к	к	д
	а	о	б	д	г	н	
	ж	а	у	е	д	я	д
х	л	и		е	м	о	а
к	р	т	д		ь	о	е
	ь	х	в	т	о	н	
р	л	е		е	д	а	ю
р		з	е	в		е	д
ш		в	а	е	н	е	н
т	и	й	е	в			д
		в		с	д		

После просмотра всех строк мы получим табл. 3.4.

Таблица 3.4

	1	2	3	4	5	6	7	8
1	x	x		x	x	x		x
2		x		x		x		
3			x					x
4	x	x		x		x	x	x
5	x				x	x	x	x
6	x	x		x		x	x	
7	x			x	x	x	x	x
8	x	x			x	x	x	x

Если бы текст был бы подлиннее и строк было бы побольше, то в каждой строке и в каждом столбце осталось бы ровно по одной незачеркнутой клетке и перестановка была бы восстановлена. В нашей таблице мы только можем утверждать, что 6-й столбец может следовать за 3-м (обозначим это событие следующим образом $3 \to 6$), или за 5-м, или за 8-м. Если 6-ой столбец не является последним. Следовательно, если начинать располагать столбцы начиная с 3, то надо рассмотреть 4 варианта.

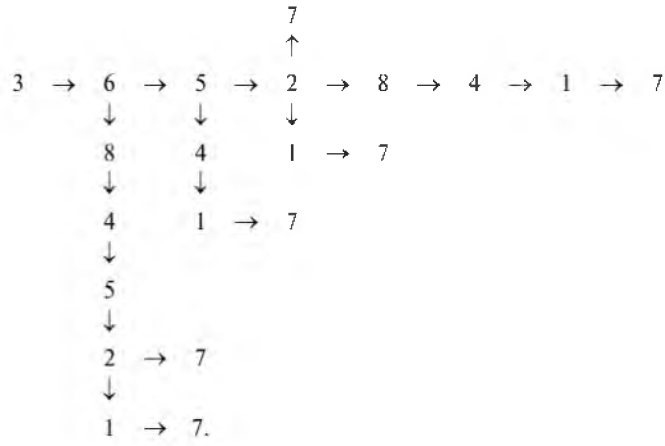
Ниже мы будем рассматривать вариант $3 \to 6$. Тогда для 6-го столбца может быть два варианта продолжения

$$\begin{array}{ccccccc} & & & & 8 & & \\ & & & & \uparrow & & \\ 3 & \to & 8 & \to & 5. & & \end{array}$$

Нам надо рассмотреть оба и постараться отсеять ложный вариант. Если отсеять ложный вариант не удастся, то надо продолжать оба варианта

$$\begin{array}{ccccccccc} & & & & 8 & \to & 4 & & 1 \\ & & & & \uparrow & & & & \uparrow \\ 3 & \to & 6 & \to & 5 & \to & 2 & \to & 7. \end{array}$$

В итоге получаем некоторое дерево возможного следования столбцов в открытом тексте



Каждой ветви дерева соответствует некоторая перестановка столбцов. Далее проверяем каждый вариант на осмысленность и получаем правильный вариант.

$$3 \to 6 \to 5 \to 2 \to 8 \to 4 \to 1 \to 7.$$

Заметим, что не обязательно было строить дерево до конца. Например, ветвь

$$3 \to 6 \to 8 \to 4 \to 5$$

можно было отсеять сразу. Разве можно признать осмысленным следующий фрагмент текста (табл. 3.5).

Таблица 3.5

3	6	8	4	5
я		м		в
ш	у	е	г	
ж	е	о	л	
ч	т	я		о
г	у	щ	е	з
к	а	т	ь	е
е	а	т		а

Такая процедура отсечения ветвей была бы просто необходима, если бы строк было поменьше и дерево было бы соответственно гораздо ветвистей. Предложенную процедуру легко автоматизировать и сделать пригодной для программной реализации. Алгоритм дешифрования должен состоять из следующих этапов.

1. Предварительная работа. Анализируя достаточно представительный объем открытых текстов, построить множество запретных биграмм.

2. Предварительная работа. Составить словарь всех возможных v-грамм для $v = 2, 3, ..., d$, которые могут встретиться в открытом тексте. Число d выбирается исходя из возможностей вычислительной техники.

Построить таблицу 8×8. При этом перебираются последовательно все запретные биграммы и для каждой опробуемой биграммы — последовательно все строки. Если хотя бы в одной строке первый символ биграммы встречается в i-том столбце, а второй в — j-ом, то клеточка $i \times j$ таблицы зачеркивается.

3. Выбрать некоторый столбец в качестве начального столбца.

4. Начать процедуру построения дерева путем пристраивания к исходному столбцу всех вариантов столбцов.

5. Для каждого полученного варианта добавить еще один из оставшихся столбцов. Если хотя бы в одной из строк таблицы встретится 3-грамма, которая отсутствует в словаре размещенных 3-грамм, то вариант отсеивается.

6. Для каждого из неотсеянных вариантов добавляем еще один столбец и проводим отсев ложных вариантов по словарю разрешенных 4-грамм.

Если словарь был построен только для $d \leq 3$, то отсев проводится путем проверки на допустимость 3-грамм, встретившихся в последних трех столбцах каждой строки. Продолжаем этот процесс до получения полной перестановки.

Ниже в табл. 3.6 приведен восстановленный для нашего примера текст.

Таблица 3.6

	1	2	3	4	5	6	7	8
1	я		в	а	м		п	и
2	ш	у		ч	е	г	о	
3	ж	е		б	о	л	е	
4	ч	т	о		я		м	о
5	г	у		е	щ	е		с
6	к	а	з	а	т	ь		т
7	е	п	е	р	ь		я	
8	з	н	а	ю		в		в
9	а	ш	е	й		в	о	л

Окончание

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

Метод дешифрования шифра перестановки при известном порядке ключевой подстановки d.

Записываем шифротекст $i_1^*, i_2^*, \ldots, i_L^*$, $L = k \times d$ в табл. 3.7.

Таблица 3.7

i_1^*	i_2^*	...	i_d^*
i_{d+1}^*	i_{d+2}^*	...	i_{2d}^*
.	.	.	.
$i_{(k-1)d+1}^*$	...	...	i_{kd}

Определим теперь множество возможных вариантов следования столбцов друг за другом. Для этого используем вероятности $P(xy)$ встречаемости биграмм xy, $x \in I$, $y \in I$ в открытом содержательном тек-

сте. Сначала будем искать первый и последний столбцы в исходной табл. 3.8, содержащей открытый текст. Если в табл. 3.7 с шифрованным текстом столбец

$$*j = \begin{pmatrix} i_j^* \\ i_{d+j}^* \\ . \\ i_{(k-1)d+j}^* \end{pmatrix}$$

является первым в табл. 3.8, а столбец

$$*j' = \begin{pmatrix} i_{j'}^* \\ i_{d+j'}^* \\ . \\ i_{(k-1)d+j'}^* \end{pmatrix}$$

последним в ней же, т.е.

$$\begin{pmatrix} i_j^* \\ i_{d+j}^* \\ . \\ i_{(k-1)d+j}^* \end{pmatrix} = \begin{pmatrix} i_1 \\ i_{1+d} \\ . \\ i_{1+(k-1)d} \end{pmatrix}, \begin{pmatrix} i_{j'}^* \\ i_{d+j'}^* \\ . \\ i_{kd+j'}^* \end{pmatrix} = \begin{pmatrix} i_d \\ i_{2d} \\ . \\ i_{kd} \end{pmatrix},$$

то последовательность биграмм:

$$(i_j^* i_{d+j'}^*), (i_{d+j'}^*, i_{2d+j}^*), \ldots, (i_{(k-1)d+j'}^*, i_{(k-1)d+j}^*) =$$
$$= (i_d, i_{1+d}), (i_{2d}, i_{1+2d}), \ldots, (i_{(k-1)d}, i_{1+(k-1)d})$$

является выборкой из дискретного распределения $(P(xy), x \in I, y \in Y)$ биграмм открытого текста (гипотеза H0). Построенные последовательности биграмм для других пар столбцов можно считать выборкой из дискретного распределения с вероятностями $p(xy) = p(x)p(y)$, $x \in I$, $y \in I$ (гипотеза H1). Данный факт позволяет статистическими критериями определить (возможно неоднозначно) пару: первый и последний столбец табл. 3.8.

Таблица 3.8

i_1	i_2	...	i_d
i_{d+1}	i_{d+2}	...	i_{2d}
.	.	.	.
$i_{(k-1)d+1}$	...	...	i_{kd}

Далее определим множество возможных вариантов следования столбцов друг за другом. Для этого используем вероятности $P(xy)$ встречаемости биграмм xy, $x \in I$, $y \in I$ в открытом содержательном тексте. Если столбец

$$*j = \begin{pmatrix} i_j^* \\ i_{d+j}^* \\ . \\ i_{(k-1)d+j}^* \end{pmatrix}$$

предшествует столбцу

$$*j' = \begin{pmatrix} i_{j'}^* \\ i_{d+j'}^* \\ . \\ i_{(k-1)d+j'}^* \end{pmatrix}$$

в истинной табл. 3.7, то

$$\begin{pmatrix} i_j^* \\ i_{d+j}^* \\ . \\ i_{j+(k-1)d}^* \end{pmatrix} = \begin{pmatrix} i_v \\ i_{v+d} \\ . \\ i_{v+(k-1)d} \end{pmatrix}, \quad \begin{pmatrix} i_{j'}^* \\ i_{d+j'}^* \\ . \\ i_{j'+(k-1)d}^* \end{pmatrix} = \begin{pmatrix} i_{v+1} \\ i_{v+1+d} \\ . \\ i_{v+1+(k-1)d} \end{pmatrix}$$

при некотором $v \in \{1, 2, ..., d-1\}$, в соседних столбцах

$$\begin{pmatrix} i_v \\ i_{v+d} \\ . \\ i_{v+(k-1)d} \end{pmatrix} \begin{pmatrix} i_{v+1} \\ i_{v+1+d} \\ . \\ i_{v+1+(k-1)d} \end{pmatrix}$$

пары букв алфавита I: $(i_v i_{v+1})$ $(i_{v+d} i_{v+1+d})$, ..., $(i_{v+(k-1)d} i_{v+1+(k-1)d})$ можно считать выборкой размера k биграмм из открытых содержательных текстов (гипотеза H0). Считаем, что для пар столбцов, не являющихся соседними, соответствующая последовательность биграмм является выборкой из вероятностного распределения с вероятностями $p(xy) = p(x)p(y)$, $x \in I$, $y \in I$ (гипотеза H1). Используя какой-либо статистический критерий разделения сформулированных гипотез, можно найти варианты, соседних столбцов, упорядоченных по следованию друг за другом. Опробуя эти варианты, можно найти открытый текст.

При малом значении k мы рекомендуем использовать весовой комбинированный критерий: для столбца

$$*j = \begin{pmatrix} i^*_j \\ i^*_{d+j} \\ \cdot \\ i^*_{(k-1)d+j} \end{pmatrix}$$

и опробуемого на предмет следования за ним столбца

$$*j' = \begin{pmatrix} i^*_{j'} \\ i^*_{d+j'} \\ \cdot \\ i^*_{(k-1)d+j'} \end{pmatrix}.$$

Из множества $\{1, ..., d\} \setminus \{*j\}$ вычисляем сумму:

$$Z(*j, *j') = P(i^*_j i^*_{j'}) + P(i^*_{d+j} i^*_{d+j'}) + ... + P(i^*_{(k-1)d+j} i^*_{(k-1)d+j'}).$$

В качестве искомого столбца берем столбец с максимальной суммой. Для удобства расчетов рекомендуем искать максимум, используя логарифмы вероятностей.

Дешифрование шифра гаммирования при некачественной гамме. Пусть I — некоторый алфавит, буквы которого упорядочены в естественном порядке. Запись $i + i' = i''$ будем понимать как модульное сложение номеров букв из $\{1, 2, ..., |I| - 1, 0\}$ по модулю $|I|$. То есть записывая уравнения, связанные с гаммированием, мы отождествляем буквы с их номерами в алфавите.

Предположим, что в качестве знаков гаммы в шифре гаммирования могут быть лишь знаки $i, i' \in I$, то есть уравнения образования шифротекста $b_1, b_2, ..., b_N$ имеют вид $a_j + \gamma_j = b_j, j \in \{1, 2, ..., N\}, \gamma_j \in \{i, i'\}$ (можно сказать, что в данном примере используют всего 1 бит гаммы).

В этом случае восстановление открытого текста $a_1, a_2, ..., a_N$ по шифротексту b_j не вызывает труда. Действительно, по известному шифротексту выпишем колонки букв (переведя их в положительные вычеты):

$$\begin{array}{cccc} b_1 - i & b_2 - i & ..... & b_N - i \\ b_1 - i' & b_2 - i' & ..... & b_N - i'. \end{array}$$

Очевидно, что в каждой колонке содержится по одной букве открытого текста. Этот текст можно попытаться восстановить, используя его избыточность. Делается это примерно так же, как и при дешифрова-

нии шифра простой замены. Не вдаваясь в подробности, приведем один пример на чтение в колонках.

Пример

К	Ш	И	П	Ш	О	Л	Ш	А	Э	И	Ж
В	Р	А	З	Т	Ж	Г	Р	Ш	Ф	А	Я

Прочитали слово КРИПТОГРАФИЯ?

Если бы для зашифрования использовалось 4 буквы, то глубина колонки была бы равна 4. Если используется полная гамма — для русского языка все 32 знака, то глубина колонок равна 32 и вы можете в ней прочитать любой текст.

Предположим, что гамма шифрования принимает все значения, но с разными вероятностями. В этом случае для дешифрования также существуют определенные подходы. Один из них заключается в чтении в колонках, где порядок (сверху вниз) в колонке возможных открытых букв $a \in I$ определен убыванием (точнее, не возрастанием) вероятностей

$$P(a/b) = P(a,b)/p(b) = \frac{p(a)q(b-a)}{\sum_{a'} p(a')q(b-a')}$$

при известной фиксированной букве b. Здесь $(p(a),\ a \in I)$ — вероятностное распределение букв открытого текста, $(q(c),\ c \in I)$ — вероятностное распределение на знаках гаммы.

Метод чтения по колонкам. Пусть $a_1, a_2, \ldots, a_N$ — неизвестный открытый текст алфавита I, $b_1, b_2, \ldots, b_N$ — известный шифротекст. В шифре гаммирования уравнение шифрования $b_t = a_t + \gamma t \bmod |I|$ может быть записано в виде $\sigma_t(a_t) = b_t$, где σ_t подстановка на I: $\sigma_t(i) = i + \gamma_t$ такова, что она определяется однозначно любым ее переходом. Пусть d — период ключевой последовательности, $N = k_d + r$.

Рассмотрим две подпоследовательности шифротекста

$$b_1, b_2 \ldots, b_j, \ldots, b_{(k-1)d+r},$$
$$b_{1+d}, b_{2+d}, \ldots, b_{j+d}, \ldots, b_{kd+r}.$$

Для изложения *метода чтения по колонкам* введем необходимые обозначения. Будем предполагать, что открытыми текстами, подлежащими шифрованию, являются содержательные тексты с вероятностями $P_1, P_2, \ldots, P_{|I|}$ букв алфавита I, P_j — вероятность буквы с номером j в содержательных текстах. Пусть на множестве $|I|^N$ ключей шифра задано равномерное распределение (ключом является реализация выборки объема N из равномерного распределения на $|I|$). Тогда вероятность $P(a_j = i,\ a_{j+d} = i'/\sigma_j(a_j) = b_j,\ \sigma_j(a_{j+d}) = b_{j+d})$ того, что j-тая и $j + d$-я

буквы открытого текста были равны, соответственно, i и i' при условии, что j-я и $j + d$-я буквы шифрованного текста равны b_j и b_{j+d} выражается формулой:

$$P(a_j = i, a_{j+d} = i' / s_j(a_j) = b_j, s_j(a_{j+d}) = b_{j+d}) =$$
$$= \frac{P(a_j = i, a_{j+d} = i'; \sigma_j(a_j) = b_j, \sigma_j(a_{j+d}) = b_{j+d})}{P(\sigma_j(a_j) = b_j, \sigma_j(a_{j+d}) = b_{j+d})}.$$

Напомним, что используемый ключ σ однозначно определен любым переходом в своей подстановке. Поэтому если числитель последнего выражения не равен нулю, то

$$\frac{P(a_j = i, a_{j-d} = i'; \sigma_j(a_j) = b_j, \sigma_j(a_{j+d}) = b_{j+d})}{P(\sigma_j(a_j) = b_j, \sigma_j(a_{j+d}) = b_{j+d})} =$$
$$= \frac{P(a_j = i, a_{j+d} = i'; \sigma_j(i) = b_j, \sigma_j(i') = b_{j+d})}{P(\sigma_j(a_j) = b_j, \sigma_j(a_{j+d}) = b_{j+d})} = \frac{P_i P_{i'}}{\sum_{\sigma \in K} P_{\sigma^{-1}(b_j)} P_{\sigma^{-1}(b_{j+d})}}.$$

Для рассматриваемых букв шифрованного текста b_j и b_{j+d} упорядочим в соответствии с невозрастанием полученных значений условных вероятностей и запишем в вертикальную колонку пары букв открытого текста $\frac{i}{i'}$, при этом верхние пары будут иметь большую условную вероятность, чем нижние. Построив такие колонки для каждого $j \in \{1, ..., N\}$ и расположив их по порядку слева направо, можно утверждать, что пары

$$\frac{a_1}{a_{1+d}}, \frac{a_2}{a_{2+d}}, \frac{a_3}{a_{3+d}}, \frac{a_4}{a_{4+d}}, ..., \frac{a_j}{a_{j+d}}, ...$$

букв искомого читаемого содержательного текста будут находиться, соответственно, в первой, второй, и т.д. колонках, и наша задача состоит в подборе пар букв в каждой колонке так, чтобы получить читаемые тексты, как по первым позициям пар, так и по вторым. При этом большинство истинных пар будет находиться ближе к верхнему краю колонок. Конечно, не исключен случай и получения нескольких начальных читаемых пар текстов на небольшой длине.

3.3. Типовые задачи криптоанализа

В отличие от задач дешифровния немаловажное место в криптографии занимют задачи криптоанализа, решение которых приводит к выводам:

— рассматриваемый шифр криптографически стойкий и его можно использовать;
— рассматриваемый шифр криптографически нестойкий и его нельзя использовать.

Перечислим основные задачи криптоанализа.

1. Известны только один или несколько шифротекстов y. В этих условиях решают следующие задачи:
— найти шифр A (определение типа шифра);
— найти шифр A, открытый текст x (дешифрование по шифротексту y).

2. Известны одна или несколько пар (x, y). В этих условиях надо определить вид шифра A и найти ключ χ.

3. Известны вид шифра A, один или несколько шифротекстов y, найти:
— x (бесключевое чтение);
— χ, x (дешифрование по шифротексту при известном шифре).

4. Известны: шифр A, одна или несколько пар (x, y), найти ключ χ.

Это типичные задачи криптоанализа. Стойкость шифров оценивается именно в этих условиях. Иногда отдельно возникают ситуации, когда известно много пар (x, y) так, что можно подобрать x или y, удовлетворяющие некоторым дополнительным условиям. Тогда говорят об атаке с использованием выбранного открытого или шифрованного текстов.

3.4. Теоретическая и практическая стойкость шифров

Понятие *теоретической стойкости шифров* обычно ассоциируется с понятием совершенного шифра по К. Шеннону.

ОПРЕДЕЛЕНИЕ. Шифр (X, K, Y, f), $Y = f(X \times K)$ с заданными вероятностными распределениями $P(x)$, $x \in X$ на X и $P(\chi)$, $\chi \in K$ называют *теоретически стойким*, если он совершенный по Шеннону, то есть при любом $y \in Y$

$$P(x/y) = P(x)$$

при любом $x \in X$.

Таким образом, теоретическая стойкость шифра (его совершенность) состоит в том, что знание шифрованного текста не влечет перераспределения вероятностей на множестве шифруемых текстов X.

В ряде случаев понятие теоретической стойкости шифра трактуют и по-другому. Теоретически стойкими шифрами относительно криптографических методов определения открытых текстов считаются те шифры, для которых эти методы приводят к неоднозначному определению открытых (содержательных) текстов. Например, теоретически стойкими шифрами относительно методов, приводящих к чтению текстов в колонках, считаются шифры, для которых доказана неоднозначность такого чтения.

Теоретически стойкими шифрами относительно теоретико-информационного представления шифра в виде канала связи без памяти считаются шифры, для которых средняя вероятность правильного декодирования открытого сообщения с заданной многозначностью по шифрованному тексту стремится к нулю с ростом длины сообщений.

Другой подход к определению стойкости связан также с именем К. Шеннона и называется сложностной подход к стойкости, или *практической стойкостью.* Сложность характеризуется двумя параметрами: число операций для вычисления результата (трудоемкость алгоритма) и объем необходимой памяти. Число операций при данном уровне развития вычислительной техники связано со временем работы алгоритма, поэтому стойкость можно выразить в терминах времени работы алгоритма дешифрования. Естественное требование надежности шифра — высокая сложность всех возможных алгоритмов дешифрования.

Меры практической стойкости шифра. Основными количественными мерами стойкости шифра служат так называемые «трудоемкость метода криптографического анализа» и «надежность его». Обозначим через A — класс применимых к шифру алгоритмов дешифрования и через $T(\varphi)$ — трудоемкость реализации алгоритма φ на некотором вычислительном устройстве.

Трудоемкость дешифрования. Данная трудоемкость обычно измеряется усредненным по ключам шифра и открытым текстам количеством времени или условных вычислительных операций, необходимых для реализации алгоритма. За трудоемкость дешифрования принимают величину

$$\min_{\varphi \in A} ET(\varphi).$$

Последняя величина (по определению) совпадает со средней трудоемкостью $ET(\varphi)$ лучшего из известных и применимых к шифру алгоритмов. При попытке практического использования этой формулы выявляются некоторые проблемы.

Поясним более подробно введенное понятие. Алгоритмы дешифрования применяются обычно к входным данным. В нашем случае это шифрованный текст «*y*» и шифр. Следовательно, результатом применения алгоритма должен быть открытый текст. Наша же цель состоит в определении трудоемкости — времени $T(\varphi)$, требуемого на реализацию алгоритма. Возможно, что $T(\varphi)$ будет зависеть от ряда дополнительных параметров, например, от шифртекста «*y*» и от порядка опробования ключей в алгоритме. Криптоанализ проводится, как правило, без наличия конкретного шифротекста и без прямой реализации алгоритма. Сам алгоритм в ряде случаев становится вероятностным алгоритмом, в его фрагментах используются вероятностные правила принятия решения о выполнении последующих действий, например, опробование ключей. Таким образом, умозрительное построение процесса нахождения открытого текста шифра, скорее всего следует назвать *криптографическим методом* решения задачи. В предположении о вероятностных распределениях случайных действий алгоритма и не известных нам входных данных алгоритма, а также вероятностных характеристик выбора ключа в шифре при зашифровании случайного содержательного открытого текста подсчитывается среднее число операций (действий) алгоритма, которое и называется *трудоемкостью метода криптоанализа*. При фиксации в предположениях вычислительных способностей противника (производительность вычислительных средств, объем возможной памяти и т.д.) это среднее число операций адекватно переводится в среднее время, необходимое для дешифрования шифра.

Надежность дешифрования. Второй количественной мерой стойкости шифра относительно метода криптоанализа является *надежность* метода $\pi(\varphi)$ — вероятность дешифрования. Раз метод несет в себе определенную случайность, например, неполное опробование ключей, то и положительный результат метода возможен с некоторой вероятностью. Блестящим примером является метод дешифрования, заключающийся в случайном отгадывании открытого текста. В ряде случаев представляет интерес и средняя доля информации, определяемая с помощью метода. В методах криптоанализа с предварительным определением ключа можно полагать, что средняя доля информации — это произведение вероятности его определения на объем дешифрованной информации.

Конечно, используют и другие характеристики эффективности методов криптоанализа, например, вероятность дешифрования за время, не превосходящее T.

Под количественной мерой криптографической стойкости шифра понимается наилучшая пара $(T(\varphi), \pi(\varphi))$ из всех возможных методов

криптографического анализа шифра. Смысл выбора наилучшей пары состоит в том, чтобы выбрать метод с минимизацией трудоемкости и одновременно максимизацией его надежности.

Предположения о возможностях противника. Криптограф, оценивая стойкость шифра, как правило, имитирует атаку на шифр со стороны криптоаналитика противника. Для этого он строит модель действий и возможностей противника, в которой максимально учитываются интеллектуальные, вычислительные, технические, агентурные и другие возможности противника. До настоящего времени криптографы не нашли практически приемлемого алгоритма дешифрования для алгоритма DES. Но небольшой размер ключа DES не позволил прогнозировать его практическую стойкость как достаточную на длительный срок, что привело к решению отказаться от использования алгоритма DES в государственных учреждениях для защиты информации.

Учет интеллектуальных возможностей противника нередко проявляется в постановках задач криптоанализа шифра. В шифрах гаммирования нередко оценивают трудоемкости и надежности методов определения открытого текста по параметрам эффективности методов определения ключа по известной гамме наложения (или, что то же самое, по известным открытому и шифрованному текстам). Аналогично иногда поступают и с другими поточными шифрами, например, при анализе шифров поточной замены переходят к решению задачи определения ключа по известной управляющей последовательности шифрующего блока. В задачах чтения открытого текста по шифрованному тексту иногда «добавляют» и другой известной информации, облегчающей нахождение решения задачи. Таким образом, учет интеллектуальных возможностей противника проводится путем постановки и решения «облегченных» задач криптоанализа. При этом полагают, что криптографическая стойкость шифра, вычисленная по таким «облегченным» задачам, не превышает стойкость шифра, анализируемого в реальных условиях эксплуатации. Нередко в качестве таких задач выделяют задачи, возникающие на промежуточных этапах анализируемого метода криптоанализа. Примерами таких задач являются разнообразные математические задачи, к которым сводится метод криптоанализа, например, задача решения систем нелинейных уравнений в разнообразных алгебраических структурах, определение начального состояния автомата по его выходной и входной последовательностям, определение входной последовательности автомата по его начальному состоянию и выходной последовательности и др. Нахождение эффективных алгоритмов решения какой-либо из этих математических задач может значительно понизить криптографическую стойкость многих шифров.

Учет старения дешифруемой информации. Что лучше? Дешифровать за пять лет пять телеграмм или за один год одну телеграмму. Ответ на этот вопрос неоднозначен. Конечно, чем больше дешифрованной информации, тем лучше, но хороша ложка к обеду! В ряде случаев не полученная вовремя информация теряет свою ценность. Так, сведения о погоде, о временных дорогах и переправах и т.д. теряют свою ценность по истечении определенного времени. Учет «старения информации» может быть проведен аналогично учету порчи продуктов питания на овощных и продовольственных складах, старения словарей, т.е. учету неиспользуемых слов из старых словарей. Такой учет может проводиться по так называемому правилу «постоянного процента», $U(t) = U(0)\, e^{-\lambda t}$, здесь $U(0)$ — начальное количество «продукта», $U(t)$ — количество «продукта» через t единиц времени. Эту же формулу иногда записывают в виде $U(t) = U(0)(1 - \alpha)^t$, α — коэффициент старения $0 < \alpha < 1$.

3.5. Имитостойкость шифров в модели К. Шеннона

Предположим, что имеется связь между абонентами A и B. Абонент A может в определенный момент времени отправить абоненту B сообщение «y» — криптограмму, зашифрованную шифром (X, K, Y, f) на ключе $\chi \in K$ (здесь в рассуждениях мы используем модель шифра Шеннона, где под X понимается множество открытых (содержательных) текстов). До момента передачи $y \in Y$ канал связи «пуст», но в шифратор абонента B введен ключ χ в ожидании получения сообщения от абонента A.

Возможные действия противника сформулируем в виде предположений о его возможностях:

- он знает действующий шифр;
- он имеет доступ к каналу связи;
- он может считывать из канала любое сообщение;
- он может формировать и вставлять в канал связи любое сообщение;
- он может заменять передаваемое сообщение любым другим сообщением;
- все указанные действия он может совершать «мгновенно» (располагая соответствующими техническими средствами);
- он не знает действующего ключа шифра.

Ниже используется обобщенная модель шифра. Именно пусть (Y, K, X', X, Φ) — шифр расшифрования, X — множество содержательных открытых текстов, $X \subset X'$. Уравнение расшифрования $y \in Y$ при ключе $\chi \in K$ будет записываться в виде $\chi^{-1}y = x, x \in X'$.

Имитация при пустом канале. Пусть канал связи «пуст» и противник встраивает в канал связи некоторое шифрованное сообщение — элемент $y \in Y$.

При действующем ключе χ абонент B расшифрует y. Априори возможны два исхода:

1) он получит $\chi^{-1}y \notin X$, то есть абонент B не прочтет сообщение и сочтет его за ложное сообщение (напомним, что в качестве X выступает множество всех содержательных текстов);

2) $\chi^{-1}y \in X$. В этом случае B получит открытое ложное сообщение $x = \chi^{-1}y$ и, действуя в соответствии с этим сообщением, может нанести себе ущерб.

Заметим, что при таком «навязывании» ложной информации противник не знает, какое именно сообщение $x \in X$ (во втором случае) он «навязал» абоненту B. Такое навязывание ложной информации можно назвать навязыванием «наугад». При случайно выбираемом ключе χ событие: $\chi^{-1}y \in X$ имеет свою вероятность

$$P_{\text{н}}(y) = P(\chi^{-1}y \in X),$$

называемую вероятностью навязывания криптограммы y. Считается естественным, что противник выберет навязываемый $y \in Y$ так, чтобы максимизировать вероятность $P_{\text{н}}(y)$. Защищающаяся сторона, в свою очередь, зная о возможных действиях противника будет выбирать такой шифр, в котором максимальное значение вероятностей навязывания:

$$P_{\text{н}} = \max_{y \in Y} P(\chi^{-1}y \in X)$$

минимально.

Величину $1/P_{\text{н}}$ можно назвать *коэффициентом имитозащиты в пустом канале*. При навязывании противником «наилучших» криптограмм y': $P_{\text{н}}(y') = P_{\text{н}}$ величина $1/P_{\text{н}}$ характеризует среднее число попыток противника до навязывания ложной информации (предполагается случайный выбор ключей при каждой попытке).

Противник может ужесточить свои действия: пусть $x(0) \in X$ — сообщение, которое наносит максимальный ущерб абоненту B; противник может попытаться навязывать такое сообщение $y \in Y$, при котором вероятность $P(\chi^{-1}y = x(0))$ максимальна. Такой способ навязывания можно назвать целевым (прицельным).

Имитация при передаваемом сообщении. Аналогичные ситуации возникают и при навязывании путем подмены передаваемого сообщения $y \in Y$ на $y' \in Y, y \neq y'$.

Имитация при знании открытого текста. Противник может обладать и дополнительной информацией (не отраженной выше в перечне его возможностей). Например, может знать открытый текст $x \in X$, который скрывается за криптограммой y в канале связи. В этой ситуации аналогично вводится величина $\max_{y \in Y} P(\chi^{-1} y \in X / \chi x = y)$, характеризующая стремление противника максимизировать условную вероятность навязывания, а защищающаяся сторона стремится минимизировать этот максимум.

Общая задача защищающийся стороны состоит в выборе такого шифра, который минимизирует ущерб от имитационных действий противника во всех реально возможных ситуациях.

Имитация в широком смысле. К имитации в широком смысле относятся и некоторые другие действия противника. Во-первых, противник может переадресовать сообщение y, идущее от A к B, другому абоненту C. Такая переадресация (со срывом сообщения абоненту B или без срыва) может привести к негативным последствиям для защищающейся стороны. Следовательно, абонент C, получив шифрованное сообщение $y \in Y$, должен быть уверен в том, что это сообщение предназначено именно ему. Во-вторых, противник может изменить «подпись» абонента A, передающего «y» абоненту B (заменив каким- либо способом в открытом тексте адрес отправителя сообщения A на адрес другого абонента A'). Поэтому абонент B, приняв сообщение, должен быть уверен, что это сообщение передано ему именно абонентом A. В-третьих, противник, перехватив $y \in Y$, может задержать это сообщение и вставить его в канал связи в другое время. Следовательно, абонент B, получая «y», должен быть уверен в том, что сообщение он получил вовремя.

Тест к главе 3

1. ***Всякий источник сообщений можно моделировать списком допустимых (т.е. встречающихся в каких-либо текстах) k-грамм при k = 1, 2, 3, ... Какие из приведенных k-грамм не являются допустимыми в русском языке (несколько верных ответов):***
   1) «ШЕЕ»;
   2) «ЖФ»;

3) «АУ»;
4) «ЮЪХ»;
5) «ЖЬН».

2. ***Криптограмма получена в результате простой замены*:**
«ВГАДЮБКГЖЯАО МЮБ ЕДБЕБЗ ЛЖФАЮТ АЬЯБГЫЖРАА»
***Ключ-подстановка*:**

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

***Восстановленный исходный текст*:**
1) «КРИПТОЛОГИЯ ЭТО НАУКА О ЗАЩИТЕ ИНФОРМАЦИИ»;
2) «КРИПТОГРАФИЯ ЭТО СПОСОБ ЗАЩИТЫ ИНФОРМАЦИИ»;
3) «КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ».

3. ***Что означает термин «многократное шифрование» применительно к блочным шифрам*:**
1) повторное применение алгоритма шифрования к шифротексту с теми же ключами;
2) шифрование одного и того же блока открытого текста несколько раз с несколькими ключами;
3) увеличение числа этапов шифрования открытого текста.

4. ***Гаммирование чаще всего осуществляется* (*несколько верных ответов*):**
1) по модулю 2, если открытый текст представляется в виде бинарной последовательности;
2) по модулю 256, если открытый текст представляется в виде последовательности байтов;
3) по модулю 16, если открытый текст представлен в цифровом виде;
4) по модулю 10, если открытый текст представлен в виде последовательности цифр, что иногда делается в ручных системах шифрования.

5. ***Основой построения большинства поточных шифров являются*:**
1) генераторы псевдослучайных чисел, в частности, различные комбинации регистров сдвига;
2) схемы суммирования по mod16;
3) таблицы подстановок.

6. ***Зашифрованный методом перестановки открытый текст*: «*Сертификаты ключей ЭЦП*» *при ключе длиной 7 и перестановке*: {*4132756*} *имеет вид*:**
1) тСреиифыктал кйюечЦ Э П;

2) юклчТи ЭСЦ еиртфаикт ы;
3) чКилют рСекиафиЭтПы Ц.

7. ***Зашифровать слово «выборочность» методом перестановки с ключом {3142}:***
   1) бвоычрнотоьс;
   2) ыовбрчоонсьт;
   3) ывброончотсь.
8. ***Зашифровать открытый текст*** — «field» ***методом Виженера, ключ*** — «moon» (***алфавит — латиница***):
   1) rwcup;
   2) rwsyp;
   3) rvsyp.
9. ***Частотный анализ может эффективно применяться для дешифрования шифров:***
   1) перестановки;
   2) многоалфавитной замены;
   3) простой замены.
10. ***Какие меры практической стойкости шифра относительно метода криптоанализа вы можете выделить (несколько верных ответов):***
    1) вероятность дешифрования за время, не превосходящее T;
    2) среднее время, необходимое для дешифрования шифра;
    3) скорость дешифрования шифра.
11. ***Какие шифры можно называть имитостойкими:***
    1) шифры, обладающие свойством противостоять разрастанию ошибок при расшифровании текстов;
    2) шифры, обладающие свойством противостоять попыткам навязывания ложной информации.
12. ***Какие шифры можно называть помехоустойчивыми:***
    1) шифры, обладающие свойством противостоять разрастанию ошибок при расшифровании текстов;
    2) шифры, обладающие свойством противостоять попыткам навязывания ложной информации.
13. ***Разрастание числа ошибок означает, что:***
    1) ошибка в одной букве, допущенная при шифровании, приводит к большому числу ошибок в расшифрованном тексте;
    2) ошибка в одной букве, допущенная при расшифровании, приводит к последующим ошибкам.
14. ***Шифр считается совершенным:***
    1) если он не поддается дешифрованию;
    2) если положение противника, стремящегося к его дешифрованию, не облегчается в результате перехвата шифротекста;

3) если требуются большие затраты или мала вероятность успеха его дешифрования.

15. ***Шифр считается практически стойким*:**
1) если он не поддается дешифрованию;
2) если положение противника, стремящегося к его дешифрованию, не облегчается в результате перехвата шифротекста;
3) если требуются большие затраты или мала вероятность успеха его дешифрования.

16. ***Степень неоднозначности восстановления открытого текста при дешифровании*:**
1) возрастает при уменьшении материала;
2) снижается при уменьшении материала.

Таблица ответов на тест к главе 3

Номер вопроса	1	2	3	4	5	6	7	8
Правильный ответ	2, 4, 5	2	2	1, 2, 4	1	1	1	2
Номер вопроса	9	10	11	12	13	14	15	16
Правильный ответ	3	1, 2	2	1	1	2	3	1

ГЛАВА 4

ОСНОВЫ АСИММЕТРИЧНОГО ШИФРОВАНИЯ

4.1. Модулярная арифметика

Множество всех целых чисел с операциями сложения и умножения называют *кольцом целых чисел*.

Для любого целого $a > 0$ справедливо равенство $a = k \times n + r_a$, где k — целое число — частное от деления a на натуральное число n, r_a остаток от деления a на n. Величина r_a может принимать значения лишь из множества $Z/n = \{0, 1, 2, ..., (n - 1)\}$. Равенство $a = k \times n + r_a$ записывают в новом виде $a \equiv r_a \bmod n$ и читают так: «a сравнимо с r_a по модулю n». На множестве Z/n вводят операции сложения по модулю n, умножения по модулю n. Результатом сложения по модулю n чисел a, b является остаток r_{a+b} от деления $(a + b)$ на n. Это записывают в виде $(a + b) = r_{a+b} \bmod n$. Аналогично вводится операция умножения по модулю n: $(a \times b) = r_{a \times b} \bmod n$. Напоминаем, что операции проводятся над возможными остатками — вычетами $\{0, 1, 2, ..., (n - 1)\}$ от деления на n целых неотрицательных чисел.

Примеры

Для $n = 12$ полный набор вычетов: $Z/12 = \{0, 1, 2, ..., 11\}$, $2 + 5 = 7 \bmod 12$, $9 + 4 = 1 \bmod 12$, $11 + 11 = 10 \bmod 12$, $2 + 0 = 2 \bmod 12$, $2 \times 5 = 10 \bmod 12$, $9 \times 4 = 0 \bmod 12$, $11 \times 11 = 1 \bmod 12$. Здесь и ниже индексы у остатков опускаются.

Множество $Z/n = \{0, 1, 2, ..., (n - 1)\}$ с введенными операциями сложения и умножения по модулю n называют *кольцом целых по модулю n* (*кольцом вычетов по модулю n, точнее, кольцом положительных вычетов по модулю n*).

Получение остатка $a \bmod n$ от деления на n произвольного целого числа a называют приведением по модулю n. Эта операция обладает хорошим свойством, называемым гомоморфизмом: мы можем либо сначала приводить числа по модулю n, а затем выполнять операции сложения и умножения, либо сначала выполнять операции, а затем приводить полученное число по модулю n. Более точно: приведение по модулю n является *гомоморфным отображением* кольца целых чисел в кольцо целых по модулю n. Легко проверяется, что

$$(a + b) \bmod n = [a(\bmod n) + b(\bmod n)] \bmod n,$$
$$(a - b) \bmod n = [a(\bmod n) - b(\bmod n)] \bmod n,$$
$$(a \times b) \bmod n = [a(\bmod n) \times b(\bmod n)] \bmod n,$$
$$[a \times (b + c)] \bmod n = \{[a \times b(\bmod n)] + [a \times c(\bmod n)]\} \bmod n.$$

Вычисление $a^x \bmod n$ — степени числа a по модулю n, как следует из записи, можно выполнить как ряд умножений и последним действием выполнить деление, получая остаток. Можно вычислить эту степень быстрее, производя возведение в степень как ряд последовательных умножений совместно с приведением по модулю. Это особенно заметно, если работать с большими числами (200 бит и более).

Примеры

Если нужно вычислить $a^8 \bmod n$, то выполняют три малых умножения и три малых приведения по модулю:

$$((a^2 \bmod n)^2 \bmod n)^2 \bmod n.$$

Тем же способом вычисляют

$$a^{16} \bmod n = (((a^2 \bmod n)^2 \bmod n)^2 \bmod n)^2 \bmod n.$$

Вычисление $a^x \bmod n$, где x не является степенью 2, лишь немного сложнее.

Двоичная запись числа x позволяет представить число x как сумму степеней 2: $x = 25_{(10)} \rightarrow 11001_{(2)}$ — двоичное представление числа 25, поэтому $25 = 2^4 + 2^3 + 2^0$. Тогда

$$a^{25} \bmod n = (a \cdot a^{24}) \bmod n = (a \cdot a^8 \cdot a^{16}) \bmod n =$$
$$= a \cdot ((a^2)^2)^2 \cdot (((a^2)^2)^2)^2 \bmod n = ((((a^2 \cdot a)^2)^2)^2 \cdot a) \bmod n.$$

4.2. Алгоритм Евклида для нахождения наибольшего общего делителя

Целое число a делит без остатка другое целое число b, если и только если $b = k \times a$ для некоторого целого числа k (т.е. остаток отделения равен 0). В этом случае a называют *делителем числа b* или *множителем в разложении* числа b на множители.

Пусть a — целое число, большее 1. Тогда a является *простым числом*, если его единственными положительными делителями будут 1 и само a, в противном случае a называется *составным*. Любое целое $n > 1$ может быть представлено единственным образом с точностью до порядка сомножителей как произведение простых.

Существенный с точки зрения криптографии факт состоит в том, что не известно никакого эффективного алгоритма разложения чисел

на множители. Более точно: считают, что *криптографическая стойкость ряда шифров с открытым ключом держится именно на отсутствии эффективного алгоритма разложения чисел на множители (алгоритма факторизации)*.

Наибольший общий делитель чисел a и b, обозначаемый как *НОД* (a, b), *или* просто (a, b), — это наибольшее целое, делящее одновременно числа a и b. В эквивалентной форме (a, b) — это то, единственное натуральное число, которое делит a и b и делится на любое целое, делящее и a, и b. Если НОД$(a, b) = 1$, то целые a и b — *взаимно простые*.

Наибольший общий делитель может быть вычислен с помощью *алгоритма Евклида*. Опишем алгоритм Евклида для нахождения НОД(a, b). Введем обозначения: q_i — частное; r_i — остаток. Тогда алгоритм можно представить в виде следующей цепочки равенств:

$$\begin{aligned}
&a = b \times q_1 + r_1, && 0 < r_1 < b,\\
&b = r_1 \times q_2 + r_2, && 0 < r_2 < r_1,\\
&r_1 = r_2 \times q_3 + r_3, && 0 < r_3 < r_2,\\
&\vdots && \vdots\\
&r_{k-2} = r_{k-1} \times q_k + r_k, && 0 < r_k < r_{k-1},\\
&r_{k-1} = r_k \times q_{k+1}.
\end{aligned}$$

Остановка гарантируется, поскольку остатки r_i от делений образуют строго убывающую последовательность натуральных чисел. Из этой цепочки немедленно получаем, что r_k есть общий делитель чисел a и b и, более того, что любой общий делитель чисел a и b делит и r_k. Таким образом, r_k = НОД(a, b) или $r_k = (a, b)$.

Отметим, что при *операции умножения действительных чисел* нетрудно вычислить мультипликативную обратную величину a^{-1} для ненулевого числа a: $a^{-1} = 1/a$ или $a \times a^{-1} = 1$. Например, мультипликативная обратная величина от числа 4 равна $1/4$, поскольку $4\frac{1}{4} = 1$.

Обращаем особое внимание, что при операции *умножения по модулю n в кольце* $Z/n = \{0, 1, 2, ..., (n - 1)\}$ вычисление обратной величины для $a \in Z/n$, т.е. величины x, для которой $a \times x \bmod n = 1$ является более сложной задачей. Например, решение сравнения $4 \times x \equiv 1 (\bmod 7)$ эквивалентно нахождению таких значений x и k, что $4 \times x \equiv 7 \times k + 1$, где x и k — целые числа. Общая формулировка этой задачи — нахождение такого целого числа x, что

$$a \times x \bmod n = 1, \quad \text{или} \quad a^{-1} \equiv x \bmod n.$$

Решение этой задачи иногда существует, а иногда его нет. Например, обратная величина для числа 5 по модулю 14 равна 3, поскольку $5 \times 3 = 15 \equiv 1 (\text{mod } 14)$. С другой стороны, число 2 не имеет обратной величины по модулю 14. *Вообще сравнение* $a^{-1} \equiv x (\text{mod } n)$ *имеет единственное решение тогда и только тогда, когда a и n — взаимно простые числа.*

В этом случае говорят, что элемент *a обратим* и его *обратный элемент* a^{-1} равен *x*. *Часто элемент* a^{-1} *обозначают через* $\frac{1}{a}$, *что вводит обычно студентов в смущение.*

Но надо всегда понимать, что это целый положительный вычет по модулю n.

Множество всех обратимых элементов кольца Z/n называется *мультипликативной группой кольца вычетов* Z/n. Эту группу обозначают через G или (Z/n). Оказывается, что элемент a из Z/n обратим, тогда и только тогда, когда он взаимно прост с n, $(a, n) = 1$. Через $\varphi(n)$ обозначают так называемую функцию Эйлера, значение $\varphi(n)$ от натурального числа n равно числу положительных целых чисел меньших n и взаимно простых с n. В связи с чем число $|G|$ элементов в *мультипликативной группе вычетов кольца* Z/n равно $\varphi(n)$, $\varphi(n) = |G|$.

Примеры

Пусть модуль $n = 10$. Полный набор вычетов по модулю $n = 10$ {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}. Из них только 1, 3, 7, 9 не имеют общего сомножителя с числом 10, то есть обратимы, $\varphi(10) = 4$, $G = \{1, 3, 7, 9\}$, $|G| = 4$.

Для произведения простых чисел p, q

$$\phi(p \cdot q) = (p-1) \cdot (q-1) = n, \quad |G| = (p-1) \cdot (q-1).$$

В таблице 4.1 приведены значения функции Эйлера при различных знаениях n.

Таблица 4.1

Модуль n	Функция $\varphi(n)$
n — простос	$n - 1$
n^2	$n(n-1)$
...	...
n^r	$n^{r-1}(n-1)$
$n = p \times q$ (p, q — простые)	$(p-1)(q-1)$
...	...
$n = \prod_{i=1}^{t} p_i^{e_i}$ (p_i — простые)	$\prod_{i=1}^{t} p_i^{e_i-1}(p_i-1)a$

Любой элемент a из группы G порождает подгруппу $H = <a> = \{a, a^2, a^3, ..., a^m\}$.

Относительно умножения по модулю n *порядок группы* $= <a> = \{a, a^2, a^3, ..., a^m\}$ (число ее элементов) совпадает с *порядком элемента* a — наименьшим натуральным числом m, при котором $a^m \equiv 1 (\mathrm{mod}\, n)$. Для g из G и ее подгруппы $H = \{h_1, h_2, ..., h_m\}$ положим, $gH = \{gh_1, gh_2, ..., gh_m\}$. Легко видеть, что $|H| = |gH|$ и при g, не принадлежащим H их пересечение пусто. Любая группа имеет разложение по *левым* (правым) *смежным классам* по своей подгруппе, $G = eH \bigcup g_2 H \bigcup ... \bigcup g_k H$ ($G = He \bigcup Hg_2 \bigcup ... \bigcup Hg_k$), e — нейтральный элемент (в нашем случае $e = 1$). Классы — подмножества $H, g_2H, ..., g_kH$ (попарно не пересекаются и в объединении дают G). Из сказанного следует, что $k|H| = |G|$ (теорема Лагранжа). Положим $H = <a>$, $<a> = \{a, a^2, a^3, ..., a^m\}$. Тогда $|H|$ в нашем случае совпадает с порядком элемента a, $|H| = m$, то m делит $|G| = \varphi(n)$. По этой причине $a\varphi^{(n)} = a^{cm}$ для некоторого c и $a\varphi^{(n)} = a^{cm} = (a^m)^c \equiv (1)^c (\mathrm{mod}\, n)$ для любого элемента a из G. Таким образом, справедливы теоремы:

— *малая теорема Ферма*: *если* n — *простое и* $НОД(a, n) = 1$, *то* $a^{n-1} \equiv 1 (\mathrm{mod}\, n)$;

— *теорема Эйлера*: *если* $НОД(a, n) = 1$, *то* $a^{\varphi(n)} \equiv 1 \mathrm{mod}\, n$.

Основные способы нахождения обратных величин $a^{-1} \equiv 1 (\mathrm{mod}\, n)$.

1. Проверить поочередно значения $1, 2, ..., n - 1$, пока не будет найден $a^{-1} \equiv 1 (\mathrm{mod}\, n)$, такой, что $a \cdot a^{-1} (\mathrm{mod}\, n) \equiv 1$.

2. Если известна функция Эйлера $\varphi(n)$, то можно вычислить $a^{-1} (\mathrm{mod}\, n) \equiv a^{\varphi(n)-1} \mathrm{mod}\, n$, используя алгоритм быстрого возведения в степень.

3. Если функция Эйлера $\varphi(n)$ не известна, можно использовать расширенный алгоритм Евклида.

Примеры

Проиллюстрируем способы нахождения обратных величин $a^{-1} \equiv 1 (\mathrm{mod}\, n)$ на числовых примерах.

Поочередная проверка значений $1, 2, ..., n - 1$, *пока не будет найден*

$$x = a^{-1} (\mathrm{mod}\, n), \text{ такой что } a \times x \equiv 1 (\mathrm{mod}\, n).$$

Пусть $n = 7$, $a = 5$. Требуется найти $x = a^{-1} (\mathrm{mod}\, n)$.

$$a \times x \equiv 1 (\mathrm{mod}\, n) \quad \text{или} \quad 5 \times x \equiv 1 (\mathrm{mod}\, 7).$$

Получаем $x = 5^{-1} (\mathrm{mod}\, 7) = 3$. Результаты проверки сведены в табл. 4.2.

Нахождение $a^{-1} (\mathrm{mod}\, n)$, *если известна функция Эйлера* $\varphi(n)$.

Пусть $n = 7$, $a = 5$. Найти $x = a^{-1} (\mathrm{mod}\, n) = 5^{-1} (\mathrm{mod}\, 7)$.

Таблица 4.2

x	$5 \times x$	$5 \times x (\text{mod } 7)$
1	5	5
2	10	3
3	15	1
4	20	6
5	25	4
6	30	2

Модуль $n = 7$ — простое число. Поэтому функция Эйлера $\varphi(n) = \varphi(7) = n - 1 = 6$. Обратная величина от 5 по mod 7

$$a^{-1}(\text{mod } n) = a\varphi^{(n)-1}(\text{mod } n) = 5^{6-1} \text{mod } 7 = 5^5 \text{ mod } 7 = (5^2 \text{ mod } 7)(5^3 \text{ mod } 7) \text{mod } 7 =$$
$$= (25 \text{ mod } 7)(125 \text{ mod } 7) \text{mod } 7 = (4 \cdot 6) \text{mod } 7 = 24 \text{ mod } 7 = 3.$$

Итак, $x = 5^{-1}(\text{mod } 7) = 3$.

Нахождение обратной величины $a^{-1}(\text{mod } n)$ с помощью расширенного алгоритма Евклида.

Пусть НОД$(a, n) = 1$, $a > 0$, $n > 0$. Рассматривается задача поиска целочисленного решения (x, y) уравнения $a \cdot x - n \cdot y = 1$. Для решения задачи используют сначала алгоритм Евклида:

$$a = n \times q_0 + r_1$$
$$n = r_1 \times q_1 + r_2$$
$$r_1 = r_2 \times q_2 + r_3$$
$$r_2 = r_3 \times q_3 + r_4$$
$$\ldots\ldots\ldots\ldots$$
$$r_{k-2} = r_{k-1} \times q_{k-1} + r_k$$
$$r_{k-1} = r_k \times q_k + 0.$$

Находят последовательность $q_0, q_1, q_2, \ldots, q_k$. Затем рекуррентно строят $P_0, P_1, P_2, \ldots, P_k$ и $Q_0, Q_1, Q_2, \ldots, Q_k$. Полагают

$$P_{-2} = 0, \quad P_{-1} = 1 \quad \text{и} \quad P_j = q_j P_{j-1} + P_{j-2} \quad \text{для } j \geq 0,$$
$$Q_{-2} = 1, \quad Q_{-1} = 0 \quad \text{и} \quad Q_j = q_j Q_{j-1} + Q_{j-2} \quad \text{для } j \geq 0.$$

Искомые значения x, y находятся по формулам

$$x = (-1)^{k-1} Q_{k-1}; \quad y = (-1)^{k-1} P_{k-1}.$$

Обратный элемент $a^{-1} = (-1)^{k-1} Q_{k-1}(\text{mod } n)$ для числа a есть решение уравнения

$$a \cdot x = 1(\text{mod } n),$$

в случае отрицательного значения a^{-1} для получения обратного элемента надо прибавить n.

4.3. Вычисления в конечных полях

Поле F есть множество, на котором определены операции сложения и умножения, удовлетворяющие требованиям: ассоциативности, коммутативности, дистрибутивности, существования аддитивного 0 и мультипликативной 1, аддитивных обратных и мультипликативных обратных для всех элементов за исключением 0. Примерами простейших конечных полей являются кольца вычетов по простому модулю Z/p.

Конечное поле $F(q)$ с конечным числом q элементов играет важную роль в криптографии. В общем случае число элементов конечного поля имеет вид

$$q = p^n,$$

где p — некоторое простое число и $n \geq 1$.

Конечные поля называют *полями Галуа* и обозначают $GF(p^n)$ или $GF(p)$ при $n = 1$. Многие криптосистемы шифров с открытым ключом базируются на полях Галуа $GF(p)$, где p — большое простое число.

Примеры

Поле Галуа $GF(5)$ имеет элементы 0, 1, 2, 3, 4 и описывается следующими таблицами сложения (табл. 4.3) и умножения (табл. 4.4).

Таблица 4.3

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

Таблица 4.4

x	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

Если p — простое число, то число $a \in \{1, ..., p - 1\}$ является взаимно простым с p, и поэтому обратный элемент a^{-1} существует. Тем самым однозначно определяется операция деления.

Обозначим через $GF^*(p)$ множество всех ненулевых элементов поля $GF(p)$. Некоторый элемент g из $GF^*(p)$ называют *образующим* или *порождающим элементом* $GF^*(p)$, если для всех a из $GF^*(p)$ найдется такое целое x, что $g^x = a \bmod p$. Всего имеется $\varphi(p-1)$ образующих элементов g. Число x называют *дискретным логарифмом элемента* a по основанию g и модулю p. Вычисление дискретных логарифмов (когда заданы g, a и p) примерно такая же труднорешаемая задача, как и задача разложения целого числа на множители. Криптографическая стойкость ряда шифров с открытым ключом держится именно на отсутствии эффективного алгоритма вычисления дискретных логарифмов.

4.4. Схема асимметричного шифрования

Асимметричные криптосистемы (*системы открытого шифрования*, *с открытым ключом — public key systems*) — смысл данных криптосистем состоит в том, что для зашифрования и расшифрования используются разные преобразования. Одно из них — зашифрование — является абсолютно открытым для всех. Другое же — расшифрование — остается секретным за счет секретности ключа расшифрования. Таким образом, любой, кто хочет что-либо зашифровать, пользуется открытым преобразованием. Но расшифровать и прочитать это сможет лишь тот, кто владеет секретным ключом (рис. 4.1).

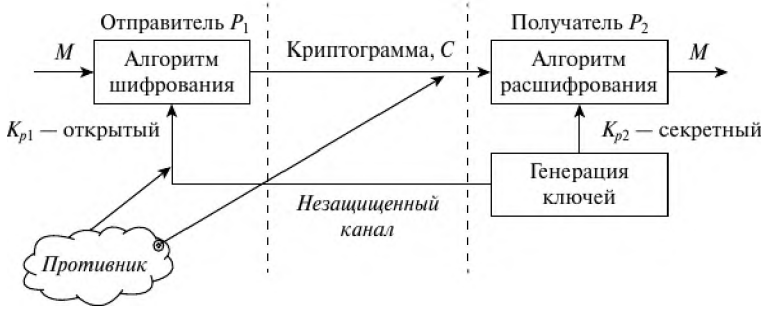


Рис. 4.1. Обобщенная схема асимметричной криптосистемы

В настоящий момент во многих асимметричных криптосистемах вид преобразования определяется ключом. У пользователя есть два ключа — секретный и открытый. Открытый ключ публикуется в общедоступном месте, и каждый, кто захочет послать сообщение этому

пользователю, — зашифровывает текст открытым ключом. Расшифровать сможет только упомянутый пользователь с секретным ключом. Таким образом, отпадает проблема передачи секретного ключа, как в симметричных системах. Однако, несмотря на все свои преимущества, эти криптосистемы достаточно трудоемки и медлительны. Стойкость асимметричных криптосистем базируется в основном на алгоритмической трудности решить за приемлемое время какую-либо задачу. Если злоумышленнику удастся построить такой алгоритм, то дискредитирована будет вся система и все сообщения, зашифрованные с помощью этой системы. В этом состоит главная опасность асимметричных криптосистем в отличие от симметричных.

4.5. Алгоритм Диффи — Хеллмана

В 1976 г. Мартин Хеллман (*Martin E. Hellman*, род. 1945) и Уитфилд Диффи (*Whitfild Diffie*, род. 1944) предложили на тот момент революционную концепцию криптографии с открытым ключом.

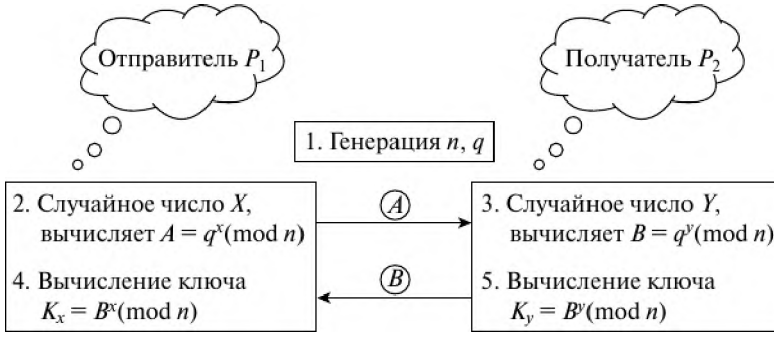
Алгоритм Диффи — Хеллмана (*Diffie — Hellman*) использует функцию дискретного возведения в степень. Сначала генерируются два больших простых числа n и q. Эти два числа не обязательно хранить в секрете. Далее один из партнеров P_1 генерирует случайное число x и посылает другому участнику будущих обменов P_2 значение

$$A = q^x \bmod n.$$

По получении A партнер P_2 генерирует случайное число y и посылает участнику обмена P_1 вычисленное значение

$$B = q^y \bmod n.$$

Партнер P_1, получив B, вычисляет $K_x = B^x \bmod n$, а партнер P_2 вычисляет $K_y = A^y \bmod n$. Алгоритм гарантирует, что числа K_y и K_x равны и могут быть использованы в качестве секретного ключа для шифрования. Ведь даже перехватив числа A и B, трудно вычислить K_x или K_y. Схематично работа алгоритма Диффи — Хеллмана представлена на рис. 4.2.



Примеры
$n = 5, q = 7, x = 3, y = 2$

$A = 7^3 \pmod 5 = 343 \pmod 5 = 3$
$K_x = 4^3 \pmod 5 = 64 \pmod 5 = 4$

$B = 7^2 \pmod 5 = 49 \pmod 5 = 4$
$K_y = 3^2 \pmod 5 = 4$

Рис. 4.2. Алгоритм Диффи — Хеллмана

Алгоритм Диффи — Хеллмана, обеспечивая конфиденциальность передачи ключа, не может гарантировать того, что он прислан именно тем партнером, который предполагается. Для решения этой проблемы был предложен протокол *STS* (*station-to-station*). Этот протокол для идентификации отправителя использует технику электронной подписи. Подпись шифруется общим секретным ключом, после того как он сформирован. Подпись включает в себя идентификаторы как P_1, так и P_2 (см. также *RFC-2786* «*Diffie — Hellman USM Key Management Information Base and Textual Convention. M.St. Johns. March 2000*»).

4.6. Алгоритм RSA

Первое практическое воплощение принцип открытого шифрования получил в системе *RSA*, разработанной в 1977 г. в Массачусетском технологическом Институте (США) и получившей свое название от первых букв фамилий авторов: Рональд Ривест (*R. Rivest*), Эди Шамир (*A. Shamir*), Леонард Адлеман (*L. Adleman*).

В криптосистеме RSA открытый ключ K_o, секретный ключ k_c, сообщение M и криптограмма C при-

надлежат кольцу целых чисел $Z/N = \{0, 1, 2, ..., N - 1\}$ по модулю N, где $N = P$. В криптосистеме RSA открытый ключ K_o, секретный ключ k_c, сообщение M и криптограмма C принадлежат кольцу целых чисел $Z/N = \{0, 1, 2, ..., N - 1\}$ по модулю N, где $N = P \times Q$, P и Q — случайные большие *простые* числа. Открытый ключ K_o выбирают случайным образом так, чтобы выполнялись условия:

$$1 < K_o \le \varphi(N),$$
$$\text{НОД}(K_o, \varphi(N)) = 1,$$
$$\varphi(N) = (P - 1)(Q - 1),$$

где $\varphi(N)$ — функция Эйлера — количество положительных целых чисел в интервале от 1 до N взаимно простых с N.

В силу $\text{НОД}(K_o, \varphi(N)) = 1$ однозначно, используя расширенный алгоритм Евклида, вычисляется секретный ключ k_c, такой, что

$$k_c \times K_o \equiv 1 (\text{mod}\, \varphi(N)).$$

Это можно осуществить, так как получатель B знает пару простых чисел (P, Q) и может легко найти $\varphi(N)$. Открытый ключ K_o используют для шифрования данных, а секретный ключ k_c — для расшифрования. Криптограмма C определяется через пару (открытый ключ K_o, сообщение M)

$$C = M^{K_o} (\text{mod}\, N).$$

В качестве алгоритма быстрого вычисления значения C используют ряд последовательных возведений в квадрат целого M и умножений на M с приведением по модулю N.

Обращение функции $C = M^{K_o} (\text{mod}\, N)$, то есть определение значения M по известным значениям C, K_o и N, практически неосуществимо при $N \approx 2^{512}$. Однако обратную задачу, т.е. задачу расшифрования криптограммы C, можно решить, используя пару (секретный ключ k_c, криптограмма C) по следующей формуле расшифрования: $M = C^{k_c} (\text{mod}\, N)$.

Проведем подробное обоснование справедливости этой формулы. Процесс расшифрования можно записать так:

$$M^{K_o k_c} = M (\text{mod}\, N).$$

Величина $\varphi(N)$ играет важную роль в теореме Эйлера, которая утверждает, что если $\text{НОД}(x, N) = 1$, то $x\varphi^{(N)} \equiv 1 (\text{mod}\, N)$, или в несколько более общей форме

$$x^n \cdot \varphi^{(N)+1} \equiv x (\text{mod}\, N).$$

Но как раз из $k_c \cdot K_o \equiv 1(\mathrm{mod}\,\varphi(N))$ следует $k_c \cdot K_o = n \cdot \varphi(N) + 1$ при некотором n. Поэтому для $(M, N) = 1$ вытекает

$$M^{K_o k_c} = M^{n\varphi(N)+1} = (M^{\varphi(N)})^n M \equiv M(\mathrm{mod}\,N).$$

При $(M, N) \neq 1$ следует воспользоваться следующим утверждением: *если P и Q — большие простые числа*, $K_0 \times k_c(\mathrm{mod}\,\varphi(N)) = 1$, *то для* ***любого*** x, $0 \le x < N$, $N = P \cdot Q$:

$$(x^{K_o})^{k_c}(\mathrm{mod}\,N) = x.$$

ДОКАЗАТЕЛЬСТВО. Пусть НОД(x, N) = 1. Тогда

$$(x^{K_o})^{k_c} = x^{K_o k_c} = x^{m\varphi(N)+1}.$$

Поэтому по теореме Эйлера

$$(x^{K_o})^{k_c}\ (\mathrm{mod}\,N) = (x(x^{m\varphi(N)}(\mathrm{mod}\,N)))(\mathrm{mod}\,N) = (x \times 1)(\mathrm{mod}\,N) = x.$$

Если НОД(x, N) $\neq 1$, то или $x = 0(\mathrm{mod}\,N)$, или НОД(x, N) = P, или НОД(x, N) = Q. Если $x = 0(\mathrm{mod}\,N)$, то $x^{K_o k_c} = 0(\mathrm{mod}\,N)$.

Пусть НОД(x, N) = P. Тогда $x = x_1 P$, где $(x_1, N) = 1$.

$$x^{K_o k_c} = x^{m(P-1)(Q-1)+1} = Px_1 P^{m(P-1)(Q-1)} x_1^{m(P-1)(Q-1)} \equiv y(\mathrm{mod}\,P \times Q).$$

Если $rP = y(\mathrm{mod}\,(P \times Q))$, то $rP = PQc + y$, следовательно, $y = Py_1$. Тогда $r \equiv y_1(\mathrm{mod}\,Q)$. Следовательно,

$$x_1((Px_1)^{m(P-1)})^{Q-1} \equiv y_1(\mathrm{mod}\,Q).$$

По теореме Ферма $z_1^{Q-1} \equiv 1(\mathrm{mod}\,Q)$. Поэтому

$$x = x_1 P(\mathrm{mod}(PQ)) = y(\mathrm{mod}\,N) \equiv x^{K_o k_c}(\mathrm{mod}\,N),$$

что и требовалось доказать.

Открытый и шифрованный тексты эффективно вычисляются, если известны K_o и k_c с помощью алгоритма быстрого возведения в степень. Если искать секретный ключ k_c по известному открытому ключу K_o, то надо знать $\varphi(N)$.

Таким образом, получатель B, который создает криптосистему, защищает два параметра:

1) секретный ключ $k_в$;

2) пару чисел (P, Q), произведение которых дает значение модуля N.

С другой стороны, получатель B открывает значение модуля N и открытый ключ $K_в$.

Противнику известны лишь значения $K_в$ и N. Если бы он смог разложить число N на множители P и Q, то он узнал бы «потайной ход» — тройку чисел $\{P, Q, K_в\}$, вычислил значение функции Эйлера $\varphi(N) = (P-1)(Q-1)$ и определил значение секретного ключа $k_в$.

Однако, как уже отмечалось, разложение очень большого N на множители вычислительно неосуществимо (при условии, что длины выбранных P и Q составляют не менее 100 десятичных знаков). Для обеспечения максимальной безопасности выбирают P и Q равной длины и хранят в секрете.

Процедуры шифрования и расшифрования в криптосистеме RSA. Предположим, что пользователь A хочет передать пользователю B сообщение в зашифрованном виде, используя криптосистему RSA. В таком случае пользователь A выступает в роли отправителя сообщения, а пользователь B — в роли получателя. Как отмечалось выше, криптосистему RSA должен сформировать получатель сообщения, т.е. пользователь B. Рассмотрим последовательность действий пользователя B и пользователя A.

1. Пользователь B выбирает два произвольных больших простых числа P и Q (в современных асимметричных криптосистемах длины чисел P и Q выбираются от 512 бит).

2. Пользователь B вычисляет значение модуля $N = P \times Q$.

3. Пользователь B вычисляет функцию Эйлера $\varphi(N) = (P-1)(Q-1)$ и выбирает случайным образом значение открытого ключа K_o с учетом выполнения условий: $1 < K_o \le \varphi(N)$, НОД$(K_o, \varphi(N)) = 1$.

4. Пользователь B вычисляет значение секретного ключа k_c, используя расширенный алгоритм Евклида при решении сравнения $k_c \equiv K_o^{-1} \pmod{\varphi(N)}$.

5. Пользователь B пересылает пользователю A пару чисел (N, K_o) по незащищенному каналу.

Если пользователь A хочет передать пользователю B сообщение M, он выполняет следующие шаги.

6. Пользователь A разбивает исходный открытый текст M на блоки, каждый из которых может быть представлен в виде числа $M_i \in \{0, 1, 2, ..., N-1\}$.

7. Пользователь A шифрует текст, представленный в виде последовательности чисел M_i по формуле $C_i = M_i^{K_o} \pmod{N}$ и отправляет криптограмму $C_1, C_2, C_3, ..., C_i, ...$ пользователю B.

8. Пользователь B расшифровывает принятую криптограмму $C_1, C_2, C_3, ..., C_i, ...$, используя секретный ключ k_c, по формуле $M_i = C_i^{k_c} \pmod{N}$.

В результате будет получена последовательность чисел M_i, которые представляют собой исходное сообщение M. Чтобы алгоритм RSA имел практическую ценность, необходимо иметь возможность без существенных затрат генерировать большие простые числа, уметь оперативно вычислять значения ключей K_o и k_c.

Примеры

1. Пусть пользователь A выбираст $P = 3$, $Q = 11$.
2. Тогда $N = P \times Q = 33$.
3. Функция Эйлера: $\varphi(N) = (P - 1)(Q - 1) = 2 \times 10 = 20$, и значение открытого ключа можем выбрать $K_o = 7$ с учетом выполнения условий: $1 < K_o \le \varphi(N)$, НОД(K_o, $\varphi(N)$) = 1.
4. Значение секретного ключа $k_c = 3$, вычислили из сравнения $k_c \times K_o \equiv 1 (\mathrm{mod}\, \varphi(N))$, т.к. $7 \times 3 \equiv 1 (\mathrm{mod}\, 20)$.
5. Пара чисел ($N = 33$, $K_o = 7$) по незащищенному каналу передается пользователю B.
6. Пусть текст пользователя A для зашифрования имеет вид: $M_1 = 3$, $M_2 = 2$.
7. Тогда, исходя из процедуры шифрования:

$C_i = M_i^{K_o} (\mathrm{mod}\, N)$, криптограмма будет иметь вид: (9, 29)

$C_1 = 3^7 \bmod 33 = 2187 \bmod 33 = 9$,
$C_2 = 2^7 \bmod 33 = 128 \bmod 33 = 29$.

8. Пользователь B, расшифровывая принятую криптограмму C_1, C_2, C_3, ..., C_i, ..., на секретном ключе $k_c = 3$, по формуле $M_i = C_i^{k_c} (\mathrm{mod}\, N)$:

$M_1 = 9^3 \bmod 33 = 729 \bmod 33 = 3$,
$M_2 = 29^3 \bmod 33 = 243899 \bmod 33 = 2$,
востанавливаст открытый текст (3, 2).

4.7. Схема шифрования Эль Гамаля

Тахер Эль Гамаль (араб. طاهر الجمل; род. 1955) — египетский криптограф. В 1985 году он опубликовал статью под названием «Криптосистема с открытым ключом и схема цифровой подписи на основе дискретных логарифмов», в которой представил свои разработки по созданию систем асимметричного шифрования и цифровой подписи, основанных на сложности проблемы дискретного логарифмирования.

Схема Эль Гамаля может быть использована как для шифрования, так и для цифровых подписей. Безопасность схемы Эль Гамаля обусловлена сложностью вычисления дискретных логарифмов в конечном поле.

Для того чтобы генерировать пару ключей (открытый ключ — секретный ключ), сначала выбирают некоторое большое простое число P и большое целое число G, причем $G < P$. Числа P и G могут быть распространены среди группы пользователей.

Затем выбирают случайное целое число X, причем $X < P$. Число X является секретным ключом и должно храниться в секрете.

Далее вычисляют $Y = G^X \bmod P$. Число Y является открытым ключом.

Для того чтобы зашифровать сообщение M, выбирают случайное целое число K, $1 < K < P - 1$, такое, что числа K и $(P - 1)$ являются взаимно простыми.

Затем вычисляют числа

$$a = G^K \bmod P,$$
$$b = Y^K M \bmod P.$$

Пара чисел (a, b) является шифртекстом. Заметим, что длина шифротекста вдвое больше длины исходного открытого текста M.

Для того чтобы расшифровать шифртекст (a, b), вычисляют

$$ba^{-X} = M \bmod P$$

Поскольку

$$a^X \equiv G^{KX} \bmod P \quad \text{и} \quad ba^{-X} = Y^K M a^{-X} = G^{KX} M G^{-KX} = M \bmod P,$$

то исходное соотношение справедливо.

Примеры

1. Выберем $P = 11$, $G = 2$ (помним, что в реальных криптосистемах с открытым ключом эти числа имеют длину порядка 512 бит), секретный ключ $X = 8$ выбрали, исходя из условия $X < P$.
2. Вычисляем

$$Y = G^X \bmod P = 2^8 \bmod 11 = 256 \bmod 11 = 3.$$

Итак, открытый ключ $Y = 3$.

3. Пусть сообщение $M = 5$.
4. Выберем некоторое случайное число $K = 9$ и убедимся, что НОД(K, $P - 1$) = 1 — действительно, НОД(9, 10) = 1.
5. Вычисляя пару чисел a и b:

$$a = G^K \bmod P = 2^9 \bmod 11 = 512 \bmod 11 = 6,$$
$$b = Y^K M \bmod P = 3^9 \cdot 5 \bmod 11 = 19683 \cdot 5 \bmod 11 = 9,$$

получим шифротекст $(a, b) = (6, 9)$.

6. Выполним расшифрование этого шифротекста, используя секретный ключ X:

$$M = b \cdot a^{-X} \bmod P = 9 \cdot 6^{-8} \bmod 11,$$

выражение $9 \cdot 6^{-8} \bmod 11$ можно представить в виде $6^8 \times M \equiv 9 \bmod 11$ или $1679616 \times M \equiv 9 \bmod 11$.

Решая данное сравнение, находим открытый текст: $M = 5$.

4.8. Схема шифрования Полига — Хеллмана

Алгоритм шифрования Полига — Хеллмана был впервые описан американскими математиками Роланом Силвером (*Roland Silver*), Стефаном Полигом (*Stephan Pohlig*) и Мартином Хеллманом (*Martin Hellman*) в 1978 г. в статье «*An improved algorithm for computing logarithms over GF(p) and its cryptographic significance*».

Шифр Полига — Хеллмана похож на RSA, он представляет собой несимметричный алгоритм, поскольку используются различные ключи для шифрования и расшифрования. В то же время эту схему нельзя отнести к классу криптосистем с открытым ключом, так как ключи шифрования и расшифрования легко выводятся один из другого. Оба ключа (шифрования и расшифрования) нужно держать в секрете.

Аналогично схеме RSA криптограмма C и открытый текст P определяются из соотношений:

$$C = P^e \bmod n,$$
$$P = C^d \bmod n,$$

где $e \times d \equiv 1$ (по модулю некоторого составного числа).

В отличие от алгоритма RSA в этой схеме число n не определяется через два больших простых числа; число n должно оставаться частью секретного ключа. Если кто-либо узнает значения e и n, он сможет вычислить значение d. Не зная значений e или d, противник будет вынужден вычислять значение:

$$e = \log_P C (\bmod n).$$

Известно, что такая задача является вычислительно сложной.

Тест к главе 4

1. ***В асимметричных криптографических алгоритмах ключи зашифрования и расшифрования всегда*:**
   1) разные, хотя и связанные между собой;
   2) разные, никак не связанные между собой;
   3) совпадают;
   4) ключ зашифрования представляет собой ключ расшифрования, записанный в обратном порядке.
2. ***Безопасность системы RSA основана*:**
   1) на трудности задачи разложения на простые множители;
   2) комбинации символов, выбранных случайным образом;
   3) использовании секретного ключа для шифрования;
   4) использовании простого делителя в качестве открытого ключа.
3. ***Кроме алгоритма RSA часто используемыми алгоритмами асимметричного шифрования являются (несколько верных ответов)*:**
   1) алгоритм Эль Гамаля;
   2) алгоритм шифрования Месси-Омуры;
   3) алгоритм Вильяма-Шафрама;
   4) алгоритм Грищенкова.
4. ***Сколько ключей используется в криптографических преобразованиях*:**
   1) 1;
   2) не менее 2;
   3) от 2 до 4;
   4) не менее 6.
5. ***Преимуществами асимметричных криптографических алгоритмов являются (несколько верных ответов)*:**
   1) скорость выполнения криптографических преобразований;
   2) легкость внесения изменений в алгоритм шифрования;
   3) секретный ключ известен только получателю информации и первоначальный обмен не требует передачи секретного ключа;
   4) применение в системах аутентификации (электронная цифровая подпись).
6. ***Что позволяют обеспечивать криптографические методы защиты*:**
   1) целостность сообщений;
   2) конфиденциальность сообщений;
   3) определять подлинность источников сообщений;
   4) гарантировать невозможность отказа от совершенных действий.

7. ***Для генерации двух ключей используется*:**
   1) два больших случайных простых числа p и q;
   2) простое число p и два случайных числа q и x;
   3) выбирается простое большое число p;
   4) выбираются два простых числа p и q, конгруэнтных $3 \bmod 4$.
8. ***Для максимальной безопасности в алгоритме RSA выбираются*:**
   1) $p < q$ по длине;
   2) $p > q$ по длине;
   3) p и q равной длины;
   4) $p >> q$ по длине.
9. ***Каким образом выбирается открытый ключ e в алгоритме RSA*:**
   1) $1 < e < (p-1)(q-1)$;
   2) $1 < e < \varphi(n)$ и e взаимно простое с $\varphi(n)$;
   3) $0 < e < \varphi(n)$ и e взаимно простое с $\varphi(n)$;
   4) может быть любым целым числом.
10. ***Каким образом вычисляется секретный ключ d в алгоритме RSA*:**
    1) $d = e^{-1} \log((p-1)(q-1))$;
    2) $d = e^{-1} \bmod((p-1)(q-1))$;
    3) $d = m \times e \bmod n$;
    4) $d = c \times e \bmod n$.
11. ***Какие значения могут быть уничтожены после генерации пары ключей в RSA*:**
    1) n;
    2) $\varphi(n)$;
    3) e;
    4) p и q.
12. ***Для расшифрования сообщения m в RSA нужно вычислить*:**
    1) $m = c^d \times e^{-1} \bmod n$;
    2) $m = c^e \bmod n$;
    3) $m = c \times e^{-1} \bmod n$;
    4) $m = c^d \bmod n$.
13. ***По какой формуле вычисляется модуль n в RSA*:**
    1) $n = k \times \varphi(n) + 1$;
    2) $n = p \times q$;
    3) $n = (p-1)(q-1)$;
    4) $n = p \times q \bmod n$.
14. ***Каково соотношение исходного текста и шифротекста в криптоалгоритме Эль Гамаля*:**
    1) длина шифротекста вдвое больше длины исходного открытого текста;
    2) длины открытого и шифротекста одинаковые;
    3) длина шифротекста вдвое меньше длины открытого текста.

15. ***Можно ли шифр Полига — Хеллмана отнести к классу криптосистем с открытым ключом*:**
    1) да;
    2) нет.
16. ***Безопасность криптосхемы Эль Гамаля обусловлена*:**
    1) сложностью факторизации больших чисел;
    2) сложностью вычисления дискретных логарифмов в конечном поле;
    3) сложностью получения значений простых чисел большой длины.

Таблица ответов на тест к главе 4

Номер вопроса	1	2	3	4	5	6	7	8
Правильный ответ	1	1	1, 2	2	3, 4	2, 3	1	3
Номер вопроса	9	10	11	12	13	14	15	16
Правильный ответ	2	2	4	4	2	1	2	2

ГЛАВА 5

ИДЕНТИФИКАЦИЯ И АУТЕНТИФИКАЦИЯ. УПРАВЛЕНИЕ КРИПТОГРАФИЧЕСКИМИ КЛЮЧАМИ

5.1. Идентификация и аутентификация

С каждым объектом компьютерной системы связана некоторая информация, однозначно идентифицирующая его. Это может быть число, строка символов, алгоритм, определяющий данный объект. Эту информацию называют *идентификатором объекта*. Если объект имеет некоторый идентификатор, зарегистрированный в сети, он называется *законным* (легальным) объектом; остальные объекты относятся к *незаконным* (нелегальным).

При попытке некоего объекта получить доступ к информационным ресурсам, обычно выполняются следующие шаги.

1 шаг. Идентификация объекта. Если подсистема защиты приняла идентификатор объекта, то данный объект считается законным для данной сети.

2 шаг. Подтверждение подлинности объекта. Проверка подлинности объекта (его *аутентификация*) устанавливает, является ли данный объект именно таким, каким он себя объявляет.

3 шаг. Предоставление полномочий (*авторизация*) *объекта*. Устанавливается сфера действия и доступные объекту ресурсы.

Перечисленные три процедуры инициализации являются процедурами защиты и относятся к одному объекту компьютерной системы. Для получения доступа к ресурсам компьютерной системы пользователь должен пройти процесс представления компьютерной системе, который включает две стадии:

— идентификацию, когда пользователь сообщает системе по ее запросу свое имя (идентификатор);

— аутентификацию, когда пользователь подтверждает идентификацию, вводя в систему уникальную, не известную другим пользователям информацию о себе (например, пароль).

Обычно процедуры идентификации и аутентификации проводятся совместно в комплексе. Для проведения процедур идентификации и аутентификации пользователя необходимо:

— наличие соответствующего субъекта (модуля) аутентификации;
— наличие аутентифицирующего объекта, хранящего уникальную информацию для аутентификации пользователя.

Основные принципы построения протоколов идентификации и аутентификации

Протокол идентификации и аутентификации — это протокол, включающий двух участников: доказывающего (проверяемого) участника A, проходящего идентификацию, и проверяющего участника B.

Целью протокола идентификации и аутентификации является проверка участником обмена B того, что проверяемый действительно является участником обмена A. Такие протоколы могут использовать как симметричные, так и асимметричные криптоситемы. Различают два вида методов идентификации. Слабая идентификация — идентификация, использующая фиксированные пароли. Сильная идентификация — идентификация на основе протоколов «запрос-ответ» или изменяющейся (без повторения) информации.

При построении протоколов идентификации важно знать возможные атаки на них.

1. *Атака «подмена»*. Попытка подменить одного пользователя другим.

2. *Атака «повторное навязывание сообщений»*. Использование информации ранее проверенного протокола идентификации того же самого или другого пользователя.

3. *Комбинированная атака*. Использование комбинации данных из ранее выполненных протоколов, в том числе протоколов, ранее навязанных противником.

4. *Атака отражением*. Комбинированная атака, использующая посылку части информации только что проверенного протокола доказывающему.

5. *Атака «задержка передачи сообщения»*. Перехват сообщения и навязывание его в более поздний период времени.

Идентификация с использованием симметричных криптосистем

Рассмотрим несколько протоколов сильной идентификации в симметричных криптосистемах. Через $(c \| d)$ ниже будет обозначаться конкатенация пары слов c и d.

Для дальнейшего изложения введем следующие обозначения:

E_k — функция шифрования на ключе k;

D_k — функция дешифрования на ключе k;

$\|$ — знак конкатенации (соединения) слов, символов;

$id(B)$ — идентификатор абонента B;

$A \to B$: y — A передает B y.

1. *Пример односторонней идентификации с использованием временной метки t_A.*

Доказательство проверяемого A: $A \to B$: $y = E_k(t_A \| id(B))$.

Проверка принятого сообщения y' проверяющим B: $(t \| i)) = D_k(y')$, удостоверение, что $t = t_A \in \Delta T$, и $i = id(B)$.

Вследствие трудности синхронизации часов передающей и принимающей стороны интервал ΔT не может быть очень малым. Поэтому технически оснащенный противник имеет возможность после перехвата сообщения y идентифицировать себя для другого участника C, изменив часть $id(B)$ этого кода. Это можно сделать, если, например, шифрование осуществляется наложением гаммы.

2. *Пример односторонней идентификация с использованием случайных чисел z_B.*

Запрос проверяющего B (пересылка случайного числа): $B \to A$: z_B.

Доказательство проверяемого A (ответ): $A \to B$: $y = E_k(z_B \| id(B))$.

Проверка принятого сообщения y' проверяющим B: $(z \| i) = D_k(y')$, удостоверение, что $z = z_B$ и $i = id(B)$.

3. *Пример взаимной идентификации с использованием случайных чисел.*

Запрос проверяющего B (пересылка случайного числа): $B \to A$: z_B.

Доказательство проверяемого A (ответ-запрос): $A \to B$: $y = E_k(z_A \| z'_B \| id(B))$.

Проверка y' проверяющим B: B: $(z_1 \| z_2 \| i) = D_k(y')$, удостоверение, что: $z_2 = z_B$, $i = id(B)$.

Доказательство проверяемого B (ответ): $B \to A$: $y = E_k(z_1 \| z_B)$.

Проверка проверяющим A: $(z'_1 \| z'_2) = D_k(y')$, удостоверение, что $z'_1 = z_A$, $z'_2 = z'_B$.

Идентификация с использованием асимметричных криптосистем

Теперь опишем несколько примеров протоколов сильной идентификации в асимметричных криптосистемах.

1. *Пример идентификации расшифрованием запроса, зашифрованного на открытом ключе K_{3_A} абонента A.*

Запрос от проверяющего B: $B \to A$: $h(z_B)$, $id(B)$, $y = E_{k_{3_A}}(z_B \| id(B))$.

Проверка запроса проверяемым A: h', i', $(z \| i) = D_{k_{p_A}}(y')$, удостоверение, что $h(z) = h'$, $i = i'$.

Доказательство проверяемого A (ответ): $A \to B$: z.

Проверка проверяющим B: z', удостоверение, что $z' = z_B$.

Примечание. Проверяемый A доказал проверяющему B, что он владеет ключом расшифрования K_{p_A}. При этом проверяемый имеет возможность учитывать запросы и фиксировать случаи их повторного поступления.

2. *Пример идентификации зашифрованием запроса секретным ключом* k_{p_A}.

Запрос от проверяющего B: $B \to A$: z_B.

Доказательство проверяемого A: $A \to B$: $y = D k_{p_A}(z'_B)$.

Проверка проверяющим B: $z = E k_{3_A}(y')$, удостоверение, что $z = z_A$.

Недостаток: C может получить от A зашифрованное на ключе k_{p_A} выгодное для C сообщение z_C (например, заархивированное долговое обязательство A). Устраняется использованием хэш-функции: запрос от проверяющего B $B \to A$, состоящий в том, что B намерен связаться с A. Доказательство проверяемого A: $A \to B$: $y = D k_{p_A}((M \| h(M)))$, где M — любое сообщение. Проверка проверяющим B: $(m \| h) = E k_{3_A}(y')$, удостоверение, что $h = h(m)$.

5.2. Управление криптографическими ключами

Под *ключевой информацией* понимают совокупность всех действующих ключей в информационной системе. Если не обеспечено достаточно надежное управление ключевой информацией, то, завладев ею, злоумышленник получает неограниченный доступ ко всей информации.

Управление ключами — информационный процесс, включающий реализацию следующих основных функций: *генерация ключей*; *хранение ключей*; *распределение ключей*.

Генерация ключей. Безопасность любого криптографического алгоритма определяется используемым криптографическим ключом. Добротные криптографические ключи должны иметь достаточную длину и случайные значения битов. Для получения ключей используются аппаратные и программные средства генерации случайных значений

ключей. Как правило, применяют датчики псевдослучайных чисел (ПСЧ). Однако степень случайности генерации чисел должна быть достаточно высокой. Идеальными генераторами являются устройства на основе «натуральных» случайных процессов, например на основе белого радиошума.

В информационной системе со средними требованиями защищенности вполне приемлемы программные генераторы ключей, которые вычисляют ПСЧ как сложную функцию от текущего времени и (или) числа, введенного пользователем.

Генерация сеансового ключа для симметричных криптосистем. Один из методов генерации сеансового ключа для симметричных криптосистем описан в стандарте ANSI X 9.17. Он предполагает использование криптографического алгоритма DES (хотя можно применить и другие симметричные алгоритмы шифрования). Введем следующие обозначения:

$E_K(X)$ — результат шифрования алгоритмом DES значения X;

K — ключ, зарезервированный для генерации секретных ключей;

V_0 — секретное 64-битовое начальное число;

T — временная отметка.

Схема генерации случайного сеансового ключа R_i в соответствии со стандартом ANSI X 9.17 показана на рис. 5.1. Случайный ключ R_i генерируют, вычисляя значение

$$R_i = E_K(E_K(T_i) \oplus V_i).$$

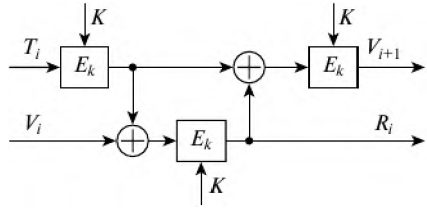


Рис. 5.1. Схема генерации случайного сеансового ключа R_i в соответствии со стандартом ANSI X 9.17

Следующее значение V_{i+1} вычисляют так:

$$V_{i+1} = E_K(E_K(T_i) \oplus R_i).$$

Если необходим 128-битовый случайный ключ, генерируют пару ключей R_i, R_{i+1} и объединяют их вместе.

Если ключ не меняется регулярно, это может привести к его раскрытию и утечке информации. Регулярную замену ключа можно осуществить, используя процедуру модификации ключа.

Модификация ключа — это генерирование нового ключа из предыдущего значения ключа с помощью односторонней (однонаправленной) функции. Участники информационного обмена разделяют один и тот же ключ и одновременно вводят его значение в качестве аргумента в одностороннюю функцию, получая один и тот же результат. Затем они берут определенные биты из этих результатов, чтобы создать новое значение ключа.

Генерация ключей для асимметричных криптосистем с открытыми ключами много сложнее, потому что эти ключи должны обладать определенными математическими свойствами (они должны быть очень большими и простыми и т.д.).

Хранение ключей. Под *функцией хранения ключей* понимают организацию их безопасного хранения, учета и удаления. Ключ является самым привлекательным для злоумышленника объектом, открывающим ему путь к конфиденциальной информации. Поэтому вопросам безопасного хранения ключей следует уделять особое внимание. Секретные ключи никогда не должны записываться в явном виде на носителе, который может быть считан или скопирован.

Носители ключевой информации. Ключевой носитель может быть технически реализован различным образом на разных носителях информации — магнитных дисках, устройствах хранения ключей типа *TOUCH MEMORY*, пластиковых картах и т.д.

Магнитные диски до недавнего времени представляли наиболее распространенный тип носителя ключевой информации. Применение магнитного диска в качестве носителя ключа позволяло реализовать необходимое свойство отчуждаемости носителя ключа от защищенной информационной системы, т.е. осуществлять временное изъятие диска из состава технических средств компьютерной системы. Для предотвращения возможности перехвата ключевой информации в процессе ее чтения с диска применялось хранение ключевой информации в зашифрованном виде. В настоящее время магнитные диски используются достаточно редко.

Электронные пластиковые карты сейчас становятся наиболее распространенным и универсальным носителем конфиденциальной информации, который позволяет идентифицировать и аутентифицировать пользователей, хранить криптографические ключи, пароли и коды.

Интеллектуальные карты (*смарт-карты*), обладающие наибольшими возможностями, эффективно применяются не только для хранения ключевой информации, но и широко используются в электронных платежных системах, в комплексных решениях для медицины, транспорта, связи, образования и т.п.

Концепция иерархии ключей. Любая информация об используемых ключах должна быть защищена, в частности храниться в зашифрованном виде.

Необходимость в хранении и передаче ключей, зашифрованных с помощью других ключей, приводит к концепции *иерархии ключей*. В стандарте ISO 8532 (*Banking-Key Management*) подробно изложен метод главных/сеансовых ключей (*master/session keys*). Суть метода состоит в том, что вводится иерархия ключей: *главный ключ* (ГК), *ключ шифрования ключей* (КК), *ключ шифрования данных* (КД). Иерархия ключей может быть:

— двухуровневой (КК/КД);
— трехуровневой (ГК/КК/КД).

Самым нижним уровнем являются *рабочие или сеансовые КД*, которые используются для шифрования данных, персональных идентификационных номеров (*PIN*) и аутентификации сообщений. Когда эти ключи надо зашифровать с целью защиты при передаче или хранении, используют ключи следующего уровня — *ключи шифрования ключей*. Ключи шифрования ключей никогда не должны использоваться как сеансовые (рабочие) КД, и наоборот.

Такое разделение функций необходимо для обеспечения максимальной безопасности. Фактически стандарт устанавливает, что различные типы рабочих ключей (например, для шифрования данных, для аутентификации и т.д.) должны всегда шифроваться с помощью различных версий ключей шифрования ключей.

В частности, ключи шифрования ключей, используемые для пересылки ключей между двумя узлами сети, известны также как *ключи обмена между узлами сети* (*cross domain keys*). Обычно в канале используются два ключа для обмена между узлами сети, по одному в каждом направлении. Поэтому каждый узел сети будет иметь *ключ отправления* для обмена с узлами сети и *ключ получения* для каждого канала, поддерживаемого другим узлом сети.

На верхнем уровне иерархии ключей располагается *главный ключ*, *мастер-ключ*. Этот ключ применяют для шифрования *КК*, когда требуется сохранить их на диске. Обычно в каждом компьютере используется только один мастер-ключ.

Мастер-ключ распространяется между участниками обмена неэлектронным способом — при личном контакте, чтобы исключить его перехват и/или компрометацию. Раскрытие противником значения мастер-ключа полностью уничтожает защиту компьютера.

Значение мастер-ключа фиксируется на длительное время (до нескольких недель или месяцев). Поэтому генерация и хранение ма-

стер-ключей являются критическими вопросами криптографической защиты. На практике мастер-ключ компьютера создается истинно случайным выбором из всех возможных значений ключей. Мастер-ключ помещают в защищенный по считыванию и записи и от механических воздействий блок криптографической системы таким образом, чтобы раскрыть значение этого ключа было невозможно. Однако все же должен существовать способ проверки, является ли значение ключа правильным.

Проблема аутентификации мастер-ключа может быть решена различными путями. Один из способов аутентификации показан на рис. 5.2.

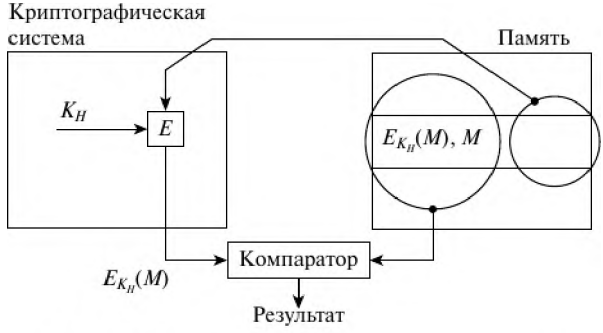


Рис. 5.2. Схема аутентификации мастер-ключа

Администратор, получив новое значение мастер-ключа K_H хост-компьютера, шифрует некоторое сообщение M ключом K_H. Пара (криптограмма $Ek_H(M)$, сообщение M) помещается в память компьютера. Всякий раз, когда требуется аутентификация мастер-ключа хост-компьютера, берется сообщение M из памяти и подается в криптографическую систему. Получаемая криптограмма сравнивается с криптограммой, хранящейся в памяти. Если они совпадают, считается, что данный ключ является правильным.

Рабочие ключи (например, сеансовый) обычно создаются с помощью псевдослучайного генератора и могут храниться в незащищенном месте. Это возможно, поскольку такие ключи генерируются в форме соответствующих криптограмм, т.е. генератор ПСЧ выдает вместо ключа K_S его криптограмму $E_{K_H}(K_S)$, получаемую с помощью мастер-ключа хост-компьютера. Расшифровывание такой криптограммы выполняется только перед использованием ключа K_S.

Схема защиты рабочего (сеансового) ключа показана на рис. 5.3.

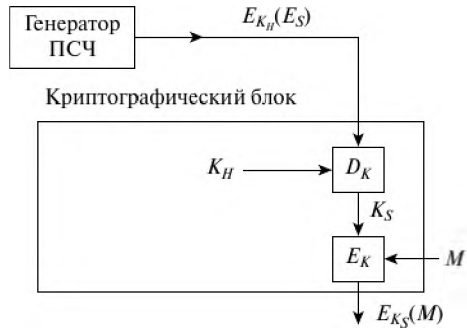


Рис. 5.3. Схема защиты сеансового ключа

Чтобы зашифровать сообщение M ключом K_S, на соответствующие входы криптографической системы подается криптограмма $E_{K_H}(K_S)$ и сообщение M. Криптографическая система сначала восстанавливает ключ K_S, а затем шифрует сообщение M, используя открытую форму сеансового ключа K_S.

Таким образом, безопасность сеансовых ключей зависит от безопасности криптографической системы. Очень важным условием безопасности информации является периодическое обновление ключевой информации в информационной системе. При этом должны переназначаться как рабочие ключи, так и мастер-ключи. В особо ответственных информационных системах обновление ключевой информации (сеансовых ключей) желательно делать ежедневно. Вопрос обновления ключевой информации тесно связан с третьим элементом управления ключами — распределением ключей.

Распределение ключей. Распределение ключей — самый ответственный процесс в управлении ключами. К нему предъявляются следующие требования:

— оперативность и точность распределения;

— скрытность распределяемых ключей.

Распределение ключей между пользователями компьютерной сети реализуется двумя способами:

1) с использованием одного или нескольких центров распределения ключей;

2) прямым обменом сеансовыми ключами между пользователями сети.

Недостаток первого подхода состоит в том, что центру распределения ключей известно, кому и какие ключи распределены, и это позволяет читать все сообщения, передаваемые по сети. Возможные

злоупотребления существенно влияют на защиту. При втором подходе проблема состоит в том, чтобы надежно удостоверить подлинность субъектов сети.

В обоих случаях должна быть обеспечена подлинность сеанса связи. Это можно осуществить, используя механизм запроса-ответа или механизм отметки времени.

Механизм запроса-ответа заключается в следующем. Пользователь *A* включает в посылаемое сообщение (запрос) для пользователя *B* непредсказуемый элемент (например, случайное число). При ответе пользователь *B* должен выполнить некоторую операцию с этим элементом (например, добавить единицу), что невозможно осуществить заранее, поскольку неизвестно, какое случайное число придет в запросе. После получения результата действий пользователя *B* (ответ) пользователь *A* может быть уверен, что сеанс является подлинным.

Механизм отметки времени предполагает фиксацию времени для каждого сообщения. Это позволяет каждому субъекту сети определить, насколько старо пришедшее сообщение, и отвергнуть его, если появится сомнение в его подлинности. При использовании отметок времени необходимо установить допустимый временной интервал задержки.

В обоих случаях для защиты элемента контроля используют шифрование, чтобы быть уверенным, что ответ отправлен не злоумышленником и не изменен штемпель отметки времени.

Задача распределения ключей сводится к построению протокола распределения ключей, обеспечивающего:

- взаимное подтверждение подлинности участников сеанса;
- подтверждение достоверности сеанса механизмом запроса-ответа или отметки времени;
- использование минимального числа сообщений при обмене ключами;
- возможность исключения злоупотреблений со стороны центра распределения ключей (вплоть до отказа от него).

В основу решения задачи распределения ключей целесообразно положить принцип отделения процедуры подтверждения подлинности партнеров от процедуры собственно распределения ключей. Цель такого подхода состоит в создании метода, при котором после установления подлинности участники сами формируют сеансовый ключ без участия центра распределения ключей с тем, чтобы распределитель ключей не имел возможности выявить содержание сообщений.

Распределение ключей с участием центра распределения ключей. При распределении ключей между участниками предстоящего информаци-

онного обмена должна быть гарантирована подлинность сеанса связи. Для взаимной проверки подлинности партнеров приемлема *модель рукопожатия*. В этом случае ни один из участников не будет получать никакой секретной информации во время процедуры установления подлинности.

Взаимное установление подлинности гарантирует вызов нужного субъекта с высокой степенью уверенности, что связь установлена с требуемым адресатом и никаких попыток подмены не было. Реальная процедура организации соединения между участниками информационного обмена включает как этап распределения, так и этап подтверждения подлинности партнеров.

При включении в процесс распределения ключей центра распределения ключей (ЦРК) осуществляется его взаимодействие с одним или обоими участниками сеанса с целью распределения секретных или открытых ключей, предназначенных для использования в последующих сеансах связи.

Следующий этап — подтверждение подлинности участников — содержит обмен удостоверяющими сообщениями, чтобы иметь возможность выявить любую подмену или повтор одного из предыдущих вызовов.

Протокол аутентификации и распределения ключей для классических (симметричных) криптосистем. Рассмотрим протоколы для симметричных криптосистем с секретными ключами и для асимметричных криптосистем с открытыми ключами. Вызывающий (исходный объект) обозначается через A, а вызываемый (объект назначения) — через B. Участники сеанса A и B имеют уникальные идентификаторы Id_A и Id_B соответственно.

Рассмотрим в качестве примера протокол *аутентификации и распределения ключей Kerberos* (по-русски — Цербер). Протокол Kerberos спроектирован для работы в сетях TCP/IP и предполагает участие в аутентификации и распределении ключей третьей доверенной стороны. Kerberos обеспечивает надежную аутентификацию в сети, разрешая законному пользователю доступ к различным машинам в сети. Протокол Kerberos основывается на симметричных шифрах (реализован алгоритм DES, хотя возможно применение и других симметричных криптоалгоритмов). Kerberos вырабатывает отдельный секретный ключ для каждого субъекта сети, и знание такого секретного ключа равносильно доказательству подлинности субъекта сети.

Основной протокол Kerberos является вариантом протокола аутентификации и распределения ключей Нидхема-Шредера. В версии 5 основного протокола Kerberos участвуют две взаимодействующие сто-

роны A и B и доверенный сервер KS (*Kerberos Server*). Стороны A и B, каждая по отдельности, разделяют свой секретный ключ с сервером KS. Доверенный сервер KS выполняет роль центра распределения ключей.

Пусть сторона A хочет получить сеансовый ключ для информационного обмена со стороной B. Сторона A инициирует фазу распределения ключей, посылая по сети серверу KS идентификаторы Id_A и Id_B:

(1) $A \rightarrow$ KS: Id_A, Id_B.

Сервер KS генерирует сообщение с временной отметкой T, сроком действия L, случайным сеансовым ключом K и идентификатором Id_A. Он шифрует это сообщение секретным ключом, который разделяет со стороной B. Затем сервер KS берет временную отметку T, срок действия L, сеансовый ключ K, идентификатор Id_B стороны B и шифрует все это секретным ключом, который разделяет со стороной A. Оба эти зашифрованные сообщения он отправляет стороне A:

(2) KS $\rightarrow A$: $E_A(T, L, K, Id_B)$, $E_B(T, L, K, Id_A)$.

Сторона A расшифровывает первое сообщение своим секретным ключом, проверяет отметку времени T, чтобы убедиться, что это сообщение не является повторением предыдущей процедуры распределения ключей.

Затем сторона A генерирует сообщение со своим идентификатором Id_A и отметкой времени T, шифрует его сеансовым ключом K и отправляет стороне B. Кроме того, A отправляет для B сообщение от KS, зашифрованное ключом стороны B:

(3) $A \rightarrow B$: $E_K(Id_A, T)$, $E_B(T, L, K, Id_A)$.

Только сторона B может расшифровать сообщения (3). Сторона B получает отметку времени T, срок действия L, сеансовый ключ K и идентификатор Id_A. Затем сторона B расшифровывает сеансовым ключом K вторую часть сообщения (3). Совпадение значений T и Id_A в двух частях сообщения подтверждают подлинность A по отношению к B.

Для взаимного подтверждения подлинности сторона B создает сообщение, состоящее из отметки времени T плюс 1, шифрует его ключом K и отправляет стороне A:

(4) $B \rightarrow A$: $E_K(T + 1)$.

Если после расшифрования сообщения (4) сторона A получает ожидаемый результат, она знает, что на другом конце линии связи находится действительно B.

Этот протокол успешно работает при условии, что часы каждого участника синхронизированы с часами сервера KS. Следует отметить, что в этом протоколе необходим обмен с KS для получения сеансово-

го ключа каждый раз, когда *A* желает установить связь с *B*. Протокол обеспечивает надежное соединение объектов *A* и *B* при условии, что ни один из ключей не скомпрометирован и сервер KS защищен.

Система Kerberos обеспечивает защиту сети от несанкционированного доступа, базируясь исключительно на программных решениях, и предполагает многократное шифрование передаваемой по сети управляющей информации.

Система Kerberos имеет структуру типа клиент-сервер и состоит из клиентских частей *C*, установленных на все машины сети (рабочие станции пользователей и серверы), и Kerberos-сервера KS, располагающегося на каком-либо (не обязательно выделенном) компьютере.

Kerberos-сервер, в свою очередь, можно разделить на две части: сервер идентификации AS (*Authentication Server*) и сервер выдачи разрешений TGS (*Ticket Granting Server*). Информационными ресурсами, необходимыми клиентам *C*, управляет сервер информационных ресурсов RS (рис. 5.4).

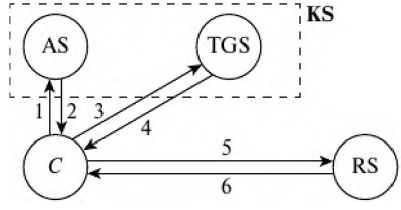


Рис. 5.4. Схема протокола Kerberos

Область действия системы Kerberos распространяется на тот участок сети, все пользователи которого зарегистрированы под своими именами и паролями в базе данных Kerberos-сервера.

Обозначения на схеме:

KS — сервер системы Kerberos;

AS — сервер идентификации;

TGS — сервер выдачи разрешений;

RS — сервер информационных ресурсов;

C — клиент системы Kerberos;

1: *C* → AS: — запрос разрешить обратиться к TGS;

2: AS → *C*: — разрешение обратиться к TGS;

3: *C* → TGS: — запрос на допуск к RS;

4: TGS → *C*: — разрешение на допуск к RS;

5: *C* → RS: — запрос на получение информационного ресурса от RS;

6: RS → *C*: — подтверждение подлинности сервера RS и предоставление информационного ресурса.

Укрупненно процесс идентификации и аутентификации пользователя в системе Kerberos можно списать следующим образом. Пользователь (клиент) *C*, желая получить доступ к ресурсу сети, направляет запрос серверу идентификации AS. Последний идентифицирует пользователя с помощью его имени и пароля и выдает разрешение на доступ к серверу выдачи разрешений TGS, который, в свою очередь, по запросу клиента *C* разрешает использование необходимых ресурсов сети с помощью целевого сервера информационных ресурсов RS.

Данная модель взаимодействия клиента с серверами может функционировать только при условии обеспечения конфиденциальности и целостности передаваемой управляющей информации. Без строгого обеспечения информационной безопасности клиент не может отправлять серверам AS, TGS и RS свои запросы и получать разрешения на доступ к обслуживанию в сети. Чтобы избежать возможности перехвата и несанкционированного использования информации, Kerberos применяет при передаче любой управляющей информации в сети сложную систему многократного шифрования с использованием комплекса секретных ключей (секретный ключ клиента, секретный ключ сервера, секретные сеансовые ключи, клиент-сервер).

Протокол для асимметричных криптосистем с использованием сертификатов открытых ключей. В этом протоколе используется идея сертификатов открытых ключей.

Сертификатом открытого ключа C называется сообщение ЦРК, удостоверяющее целостность некоторого открытого ключа объекта. Например, сертификат открытого ключа для пользователя *A*, обозначаемый C_A, содержит отметку времени *T*, идентификатор Id_A и открытый ключ K_A, зашифрованные секретным ключом удостоверяющего центра $k_{\text{ЦРК}}$, т.е.

$$C_A = E_{k_{\text{ЦРК}}}(T, Id_A, K_A).$$

Отметка времени *T* используется для подтверждения актуальности сертификата и тем самым предотвращает повторы прежних сертификатов, которые содержат открытые ключи и для которых соответствующие секретные ключи несостоятельны.

Секретный ключ $k_{\text{ЦРК}}$ известен только менеджеру ЦРК. Открытый ключ $K_{\text{ЦРК}}$ известен участникам *A* и *B*. ЦРК поддерживает таблицу открытых ключей всех объектов сети, которые он обслуживает.

Вызывающий объект *A* инициирует стадию установления ключа, запрашивая у ЦРК сертификат своего открытого ключа и открытого ключа участника *B*:

(1) $A \rightarrow$ ЦРК: Id_A, Id_B, *'Вышлите сертификаты ключей A и B'*.

Здесь Id_A и Id_B — уникальные идентификаторы соответственно участников A и B.

Менеджер ЦРК отвечает сообщением

(2) ЦРК $\rightarrow A$: $E_{k_{\text{ЦРК}}}(T, Id_A, K_A)$, $E_{k_{\text{ЦРК}}}(T, Id_B, K_B)$.

Участник A, используя открытый ключ ЦРК $K_{\text{ЦРК}}$, расшифровывает ответ ЦРК, проверяет оба сертификата. Идентификатор Id_B убеждает A, что личность вызываемого участника правильно зафиксирована в ЦРК и K_B — действительно открытый ключ участника B, поскольку оба зашифрованы ключом $k_{\text{ЦРК}}$.

Хотя открытые ключи предполагаются известными всем, посредничество ЦРК позволяет подтвердить их целостность. Без такого посредничества злоумышленник может снабдить A своим открытым ключом, который A будет считать ключом участника B. Затем злоумышленник может подменить собой B и установить связь с A, и его никто не сможет выявить.

Следующий шаг протокола включает установление связи A с B:

(3) $A \rightarrow B$: C_A, $E_{k_A}(T)$, $E_{K_B}(r_1)$.

Здесь C_A — сертификат открытого ключа пользователя A; $E_{k_A}(T)$ — отметка времени, зашифрованная секретным ключом участника A и являющаяся подписью участника A, поскольку никто другой не может создать такую подпись; r_1 — случайное число, генерируемое A и используемое для обмена с B в ходе процедуры подлинности.

Если сертификат C_A и подпись A верны, то участник B уверен, что сообщение пришло от A. Часть сообщения $E_{K_B}(r_1)$ может расшифровать только B, поскольку никто другой не знает секретного ключа k_B, соответствующего открытому ключу K_B. Участник B расшифровывает значение числа r_1 и, чтобы подтвердить свою подлинность, посылает участнику A сообщение

(4) $B \rightarrow A$: $E_{K_A}(r_1)$.

Участник A восстанавливает значение r_1, расшифровывая это сообщение с использованием своего секретного ключа k_A. Если это ожидаемое значение r_1, то A получает подтверждение, что вызываемый участник действительно B.

Протокол, основанный на симметричном (классическом) шифровании, функционирует быстрее, чем протокол, основанный на криптосистемах с открытыми ключами. Однако способность систем с открытыми ключами генерировать цифровые подписи, обеспечивающие

различные функции защиты, компенсирует избыточность требуемых вычислений.

Прямой обмен ключами между пользователями. При использовании для информационного обмена криптосистемы с симметричным секретным ключом два пользователя, желающие обменяться криптографически защищенной информацией, должны обладать общим секретным ключом. Пользователи должны обменяться общим ключом по каналу связи безопасным образом. Если пользователи меняют ключ достаточно часто, то доставка ключа превращается в серьезную проблему.

Для решения этой проблемы применяют два способа:

1) использование криптосистемы с открытым ключом для шифрования и передачи секретного ключа симметричной криптосистемы;

2) использование системы открытого распределения ключей Диффи — Хеллмана.

Использование криптосистемы с открытым ключом для шифрования и передачи секретного ключа симметричной криптосистемы. Алгоритмы, лежащие в основе криптосистем с открытым ключом, имеют следующие недостатки:

— генерация новых секретных и открытых ключей основана на генерации новых больших простых чисел, а проверка простоты чисел занимает много процессорного времени;

— процедуры шифрования и расшифрования, связанные с возведением в степень многозначного числа, достаточно громоздки.

Комбинированный метод шифрования позволяет сочетать преимущества высокой секретности, предоставляемые асимметричными криптосистемами с открытым ключом, с преимуществами высокой скорости работы, присущими симметричным криптосистемам с секретным ключом. При таком подходе криптосистема с открытым ключом применяется для шифрования, передачи и последующего расшифрования только секретного ключа симметричной криптосистемы, а симметричная криптосистема применяется для шифрования и передачи исходного открытого текста. В результате криптосистема с открытым ключом не заменяет симметричную криптосистему с секретным ключом, а лишь дополняет ее, позволяя повысить в целом защищенность передаваемой информации. Если пользователь A хочет передать зашифрованное комбинированным методом сообщение M пользователю B, то порядок его действий будет таков.

1. Создать (например, сгенерировать случайным образом) симметричный ключ, называемый в этом методе сеансовым ключом K_S.

2. Зашифровать сообщение M на сеансовом ключе K_S.

3. Зашифровать сеансовый ключ K_S на открытом ключе K_B пользователя B.

4. Передать по открытому каналу связи в адрес пользователя B зашифрованное сообщение вместе с зашифрованным сеансовым ключом.

Действия пользователя B при получении зашифрованного сообщения и зашифрованного сеансового ключа должны быть обратными.

5. Расшифровать на своем секретном ключе k_B сеансовый ключ K_S.

6. С помощью полученного сеансового ключа K_S расшифровать и прочитать сообщение M.

Использование системы открытого распределения ключей Диффи — Хеллмана. Для решения проблемы доставки ключа применяют системы открытого распределения ключей. Эти системы позволяют пользователям обмениваться ключами по незащищенным каналам связи. Интересно отметить, что системы открытого распределения ключей базируются на тех же принципах, что и системы шифрования с открытыми ключами. Для примера рассмотрим алгоритм открытого распределения ключей Диффи — Хеллмана.

Алгоритм открытого распределения ключей Диффи — Хеллмана. Его безопасность обусловлена трудностью вычисления дискретных логарифмов в конечном поле, в отличие от легкости дискретного возведения в степень в том же конечном поле.

Предположим, что два пользователя A и B хотят организовать защищенный коммуникационный канал.

1. Обе стороны заранее уславливаются о модуле N (N должен быть простым числом) и примитивном элементе $g \in Z_N$, $(1 \le g \le N-1)$, который образует все ненулевые элементы множества Z_N, т.е.

$$\{g, g^2, \ldots, g^{N-1} = 1\} = Z_N - \{0\}.$$

Эти два целых числа N и g могут не храниться в секрете. Как правило, эти значения являются общими для всех пользователей системы.

2. Затем пользователи A и B независимо друг от друга выбирают собственные секретные ключи k_A и k_B (k_A и k_B — случайные большие целые числа, которые хранятся пользователями A и B в секрете).

3. Далее пользователь A вычисляет открытый ключ

$$y_A = g^{k_A} \pmod N,$$

а пользователь B — открытый ключ

$$y_B = g^{k_B} \pmod N.$$

4. Затем стороны A и B обмениваются вычисленными значениями открытых ключей y_A и y_B по незащищенному каналу.

5. Далее пользователи A и B вычисляют общий секретный ключ, используя следующие сравнения:

пользователь A: $K = (y_B)^{k_A} = (g^{k_B})^{k_A} \pmod N$;

пользователь B: $K' = (y_A)^{k_B} = (g^{k_A})^{k_B} \pmod N$.

При этом $K = K'$, так как $(g^{k_B})^{k_A} = (g^{k_A})^{k_B} \pmod N$.

Схема реализации алгоритма Диффи — Хеллмана показана на рис. 5.5.

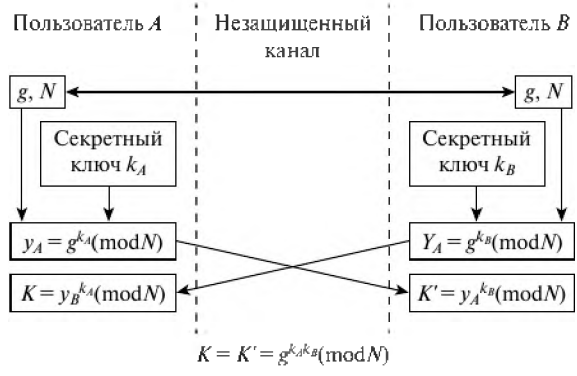


Рис. 5.5. Алгоритм открытого распределения ключей Диффи — Хеллмана

Ключ K может использоваться в качестве общего секретного ключа (ключа шифрования ключей) в симметричной криптосистеме.

Кроме того, обе стороны A и B могут шифровать сообщения, используя следующее преобразование шифрования (типа RSA): $C = E_K(M) = M^K \pmod N$.

Для выполнения расшифрования получатель сначала находит ключ расшифрования K^* с помощью сравнения

$$K \times K^* \equiv 1 (\mathrm{mod}\, N - 1),$$

а затем восстанавливает сообщение

$$M = D_K(C) = C^{K^*} (\mathrm{mod}\, N).$$

Примеры

Допустим, модуль $N = 47$, а примитивный элемент $g = 23$. Предположим, что пользователи A и B выбрали свои секретные ключи: $k_A = 12 (\mathrm{mod}\, 47)$ и $k_B = 33 (\mathrm{mod}\, 47)$.

Для того чтобы иметь общий секретный ключ K, они вычисляют сначала значения частных открытых ключей:

$$y_A = g^{k_A} = 23^{12} = 27 (\text{mod } 47),$$
$$y_B = g^{k_B} = 23^{33} = 33 (\text{mod } 47).$$

После того как пользователи A и B обменяются своими значениями y_A и y_B, они вычисляют общий секретный ключ

$$K = (y_B)^{k_A} = (y_A)^{k_B} = 33^{12} = 27^{33} = 23^{12 \cdot 33} = 25 (\text{mod } 47).$$

Кроме того они находят секретный ключ расшифрования, используя следующее сравнение:

$$K \times K^* \equiv 1 (\text{mod } N - 1),$$

откуда $K^* = 35 (\text{mod } 46)$.

Теперь, если сообщение $M = 16$, то криптограмма

$$C = M^K = 16^{25} = 21 (\text{mod } 47).$$

Получатель восстанавливает сообщение так:

$$M = C^{K^*} = 21^{35} = 16 (\text{mod } 47).$$

Злоумышленник, перехватив значения N, g, y_A и y_B, тоже хотел бы определить значение ключа K. Очевидный путь для решения этой задачи состоит в вычислении такого значения k_A по N, g, y_A, что $g^{k_A} \bmod N = y_A$ (поскольку в этом случае, вычислив k_A, можно найти $K = (y_B)^{k_A} \bmod N$). Однако нахождение k_A по N, g и y_A — задача нахождения дискретного логарифма в конечном поле, которая считается неразрешимой.

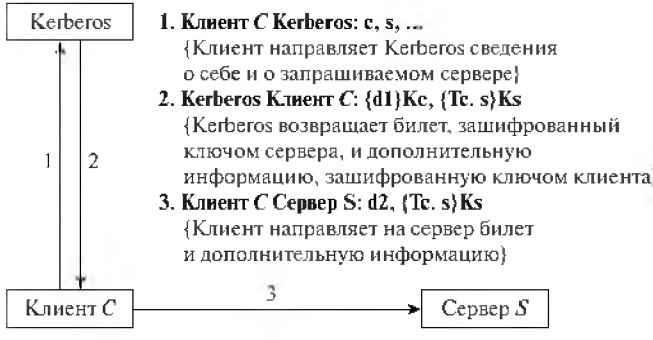
Выбор значений N и g может иметь существенное влияние на безопасность этой системы. Модуль N должен быть большим и простым числом, так же как и $(N - 1)/2$ должно быть простым числом. Число g желательно выбирать таким, чтобы оно было примитивным элементом множества Z_N. (В принципе, достаточно, чтобы число g генерировало большую подгруппу мультипликативной группы по mod N.)

Алгоритм открытого распределения ключей Диффи — Хеллмана позволяет обойтись без защищенного канала для передачи ключей. Однако, работая с этим алгоритмом, необходимо иметь гарантию того, что пользователь A получил открытый ключ именно от пользователя B, и наоборот. Эта проблема решается с помощью электронной подписи, которой подписываются сообщения об открытом ключе.

Тест к главе 5

1. ***Для проведения процедур идентификации и аутентификации пользователя необходимы (несколько верных ответов):***
   1) наличие электронной подписи пользователя;
   2) наличие соответствующего субъекта (модуля) аутентификации;
   3) наличие аутентифицирующего объекта, хранящего уникальную информацию для аутентификации пользователя.
2. ***Слабая идентификация — это:***
   1) идентификация, использующая одноразовые пароли;
   2) идентификация, использующая имитовставки;
   3) идентификация, использующая фиксированные пароли;
   4) идентификация на основе протоколов «запрос-ответ» или изменяющейся (без повторения) информации.
3. ***Сильная идентификация — это:***
   1) идентификация, использующая одноразовые пароли;
   2) идентификация, использующая имитовставки;
   3) идентификация, использующая фиксированные пароли;
   4) идентификация на основе протоколов «запрос-ответ» или изменяющейся (без повторения) информации.
4. ***Что служит аутентификатором, то есть используется для подтверждения подлинности субъекта (несколько верных ответов):***
   1) нечто, что субъект знает (пароль, личный идентификационный номер, криптографический ключ и т.п.);
   2) нечто, чем субъект владеет (личную карточку или иное устройство аналогичного назначения);
   3) нечто, что есть часть самого субъекта (голос, отпечатки пальцев и т.п., то есть свои биометрические характеристики);
   4) электронная подпись субъекта.
5. ***Какие меры позволяют повысить надежность парольной защиты (несколько верных ответов):***
   1) наложение технических ограничений (пароль должен быть не слишком коротким, он должен содержать буквы, цифры и т.п.);
   2) управление сроком действия паролей, их периодическая смена;
   3) ограничение доступа к файлу паролей;
   4) неиспользование программных генераторов паролей;
   5) ограничение числа неудачных попыток входа в систему; обучение пользователей.

6. ***Определите верную последовательность шагов программного генератора одноразовых паролей (три шага):***
   1) сервер присылает на пользовательскую систему число $(n - 1)$;
   2) пользователь верифицирует секретный ключ K;
   3) сервер применяет однонаправленную функцию f к полученному от пользователя значению и сравнивает результат с ранее сохраненной величиной: в случае совпадения подлинность пользователя считается установленной, сервер запоминает новое значение (присланное пользователем) и уменьшает на единицу счетчик (n);
   4) пользователь применяет однонаправленную функцию f к секретному ключу K $(n - 1)$ раз и отправляет результат по сети на сервер аутентификации.
7. ***Верно ли расставлены на стрелках номера действий при проверке сервером S подлинности клиента C в протоколе Kerberos:***



   1) да;
   2) нет.
8. ***Отношение «субъекты-объекты» можно представить в виде матрицы доступа:***
   1) в строках которой перечислены субъекты, в столбцах — объекты, а в клетках, расположенных на пересечении строк и столбцов, записаны дополнительные условия (например, время и место действия) и разрешенные виды доступа;
   2) в строках и столбцах которой перечислены идентификаторы и аутентификаторы пользователей;
   3) в строках и столбцах которой перечислены идентификаторы пользователей.
9. ***При принятии решения о предоставлении доступа обычно анализируется следующая информация (несколько верных ответов):***
   1) электронная подпись субъекта, для которой ключ проверки электронной подписи указан в квалифицированном сертификате;

2) идентификатор субъекта (например, идентификатор пользователя, сетевой адрес компьютера);
3) атрибуты субъекта (например, метка безопасности, группа пользователя).

10. ***Управление ключами — это*:**
1) реализация протокола распределения ключей между пользователями информационной системы;
2) информационный процесс, включающий реализацию следующих основных функций: генерация ключей; хранение ключей; распределение ключей.

11. ***Иерархия ключей может быть (несколько верных ответов):***
1) двухуровневой (ключ шифрования ключей/ключ шифрования данных);
2) трехуровневой (главный ключ/ключ шифрования ключей/ключ шифрования данных);
3) двухуровневой (главный ключ/ключ шифрования данных).

12. ***Распределение ключей между пользователями компьютерной сети реализуется двумя способами*:**
1) непосредственным обменом между пользователями сети хэш-образами ключей;
2) с использованием одного или нескольких центров распределения ключей;
3) прямым обменом сеансовыми ключами между пользователями сети.

13. ***Как может быть обеспечена подлинность сеанса связи между пользователями компьютерной сети (несколько верных ответов):***
1) можно использовать механизм запроса-ответа;
2) можно использовать механизм электронной подписи;
3) можно использовать механизм отметки времени.

14. ***Протокол Kerberos предполагает участие в аутентификации и распределении ключей третьей доверенной стороны*:**
1) нет;
2) да.

15. ***Для решения проблемы прямого обмена ключами между пользователями применяют два способа*:**
1) использование криптосистемы с открытым ключом для шифрования и передачи секретного ключа симметричной криптосистемы;
2) использование для обмена между пользователями хэш-образа ключей;

3) использование системы открытого распределения ключей Диффи — Хеллмана.

16. ***Безопасность алгоритма открытого распределения ключей Диффи — Хеллмана обусловлена*:**
    1) трудностью факторизации больших чисел;
    2) трудностью вычисления дискретных логарифмов в конечном поле;
    3) трудностью дискретного возведения в степень в конечном поле.

Таблица ответов на тест к главе 5

Номер вопроса	1	2	3	4	5	6	7	8
Правильный ответ	2, 3	3	4	1, 2, 3	1, 2, 3, 5	1, 4, 3	1	1
Номер вопроса	9	10	11	12	13	14	15	16
Правильный ответ	2, 3	2	1, 2	2, 3	1, 3	2	1, 3	2

ГЛАВА 6

ЭЛЕКТРОННАЯ ПОДПИСЬ

6.1. Процедуры постановки и проверки подписи

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 подпись*.

Электронная подпись используется для аутентификации текстов, передаваемых по телекоммуникационным каналам. Функционально она аналогична обычной рукописной подписи и обладает основными ее достоинствами:

— удостоверяет, что подписанный текст исходит от лица, поставившего подпись;
— не дает самому этому лицу возможность отказаться от обязательств, связанных с подписанным текстом;
— гарантирует целостность подписанного текста.

Электронная подпись представляет собой относительно небольшое количество дополнительной цифровой информации, передаваемой вместе с подписываемым текстом, и включает две процедуры:

- *процедуру постановки подписи*, в которой используется секретный ключ отправителя сообщения;
- *процедуру проверки подписи*, в которой используется открытый ключ отправителя.

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

Процедура проверки подписи. При проверке электронной подписи получатель сообщения снова вычисляет хэш-функцию $m = h(M)$ принятого по каналу текста M, после чего с помощью открытого ключа

отправителя проверяет, соответствует ли полученная подпись вычисленному значению m хэш-функции.

Принципиальным моментом в системе электронной подписи является невозможность подделки электронной подписи пользователя без знания его секретного ключа.

Каждая подпись, как правило, содержит следующую информацию:

— дату подписи;
— срок окончания действия ключа данной подписи;
— информацию о лице, подписавшем текст;
— идентификатор подписавшего (имя открытого ключа);
— собственно электронную подпись.

6.2. Хэш-функции

Пусть M' — множество всевозможных сообщений. Множество M' является, вообще говоря, бесконечным. Хэш-функцией называется отображение

$$H : M' \to M,$$

где M — некоторое множество.

Хэш-функция должна удовлетворять нескольким требованиям, что и позволяет ее использовать в криптографии, в частности, для подписи сообщений:

1) для данного $m' \in M'$ легко вычислить $H(m') \in M$;

2) для данного $m \in M$ трудно вычислить $m' \in M'$ такое, что $H(m') \in m$;

3) для данного $m' \in M'$ трудно вычислить $m'' \in M'$, $m'' \neq m'$ такое, что $H(m') = H(m'')$.

Отображение H, удовлетворяющее требованиям 1) и 2), является однонаправленным. Иногда вместо 3) требуют выполнения более сильного свойства;

3') трудно вычислить пару m', $m'' \in M'$, $m' \neq m''$, для которой $H(m') = H(m'')$.

Пара элементов множества M' (пара сообщений), о которой идет речь в 3'), называется *коллизией* для отображения H. Т. к. мощность M' больше мощности M, то H допускает коллизии.

Очевидно, что определение однонаправленной хэш-функции и коллизии не зависит от того, является ли множество M' конечным или бесконечным. Несложно показывается, что свойство 3') влечет свойство 3). Оказывается, что при выполнении нескольких естественных предположений условие 3') влечет выполнение условия 2).

Как правило, в качестве M' берут множество I^* всех слов конечной длины в некотором алфавите $I = \{i_1, ..., i_m\}$, а в качестве M множество I^L всех слов длины L алфавита. На практике, чаще всего $I = F_2$, $I^* = F_2^L$ — множество слов конечной длины в двоичном алфавите $F_2 = \{0, 1\}$. От функции H: $I^* \to I^L$ требуют, чтобы она обладала свойством: значения H на словах, которые даже имеют отличие друг от друга только в одном знаке, дают значительно отличающиеся хэш-значения. Тогда, получив на приемном конце сообщение и его хэш $(m', H(m'))$, можно вычислить значение хэш от сообщения, сравнить с полученным хэш по каналу связи и подтвердить или опровергнуть, что сообщение не искажено.

Если функция H зависит также от ключа $k \in K$, то помимо проверки целостности добавление значения хэш к сообщению подтверждает истинность сообщения. Такой способ подтверждения истинности называется *кодом аутентификации* (*Message Autentification Code — MAC*). Однако такое подтверждение истинности еще не является электронной подписью. Подтверждение истины называется подписью, если ее могут проверить все, не знающие ключи. Например, в суде можно поверить истинность, не раскрывая ключи. Приведенный выше способ проверки подлинности сообщения непригоден, так как влечет раскрытие ключа. Для того чтобы код аутентификации стал электронной подписью сообщения, необходимо использовать хэш-функции с дополнительными свойствами. Например, использовать систему с открытым ключом. Напомним вкратце суть электронной подписи. Пусть у корреспондента B имеются два алгоритма E_B и D_B, каждый из которых преобразует слово из I^L в слово из I^L и каждый определен на I^L. Первый алгоритм известен всем, а второй — только владельцу B. Обладание алгоритмом D_B однозначно (юридически) определяет корреспондента B. Считаем, что E_B и D_B удовлетворяют соотношениям для любого $\alpha \in I^L$

$$E_B D_B(\alpha) = \alpha,\ D_B E_B(\alpha) = \alpha.$$

Тогда электронной подписью документа $M \in A^*$ называется

$$C = D_B(h(M)).$$

Проверка подписи под документом M возможна любым лицом, у которого есть E_B. Для этого проверяющий вычисляет $H(M)$, причем $H()$ — известна всем, а M проверяющий получает вместе с подписью C. Второй шаг проверки — вычисление

$$E_B(C) = E_B D_B(H(M)) = H(M).$$

Если вычисленное значение хэш от M совпало с результатом применения алгоритмов E_B к C, то подпись B считается подтвержденной (даже в суде).

Обычно хэш $h()$ строится следующим образом. Выбирается функция h: $A^L \times A^L \to A^L$, удовлетворяющая свойствам 1—3. Например, когда длина L хэш была 64 бита, то брали следующую схему (рис. 6.1).

$$\forall\ \alpha \in A^*,\ \alpha = \alpha_1 \ldots \alpha_t,$$

где длина блока α_t равна L, а блок α_t дополнен до блока длины L (если это необходимо). $H(\alpha)$ вычисляется по следующему алгоритму.

$$\beta_1 = h(\alpha_1, \alpha_2)$$
$$\beta_2 = h(\beta_1, \alpha_3)$$
$$\ldots\ldots\ldots\ldots$$
$$\beta_{t-1} = h(\beta_{t-2}, \alpha_t).$$

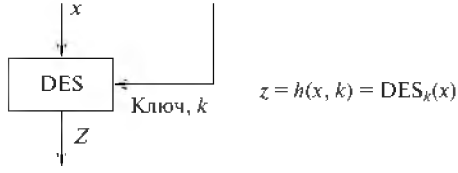


Рис. 6.1. Схема построения хэш-фукнции

Известно следующее.

УТВЕРЖДЕНИЕ. Пусть M', M — конечные множества. Задано произвольное отображение H: $M' \to M$. Пусть для любого $m \in M$ имеется эффективный алгоритм вычисления m' такого, что $H(m') = m$ (если m' существует), при этом алгоритм задает равномерное распределение на множестве таких m' при всех $m \in M$. Тогда имеется эффективный алгоритм вычисления коллизии для функции H.

ДОКАЗАТЕЛЬСТВО. Обозначим через A алгоритм обращения h, о котором идет речь в формулировке леммы. То есть $A(m) = m'$ для некоторого m' такого, что $H(m') = m$. Сформулируем алгоритм вычисления коллизии.

Входные данные алгоритма: хэш-функция H: $M' \to M$, алгоритм A. Выходные данные алгоритма: коллизия для H.

Шаг 1. Выбрать случайное $m' \in M'$, вычислить $m = H(m')$.

Шаг 2. Вычислить $m'' = H(m)$.

Шаг 3. Если $m' = m''$, то перейти к шагу 1; в противном случае m', m'' — коллизия для H. Алгоритм заканчивает работу.

Оценим среднее число прохождений алгоритма через шаг 1. Имеем вероятностную схему, исходами которой являются пары m', m'', где $H(m') = H(m'')$. Обозначим через $C_{m'}$ множество $m'' \in M'$, для которых $H(m'') = H(m')$. Заметим, что число классов m', m'' не больше $|M'|$. Тогда вероятность исхода m', m'' равна $\frac{1}{|M'|}\frac{1}{|C_{m'}|}$. Благоприятными являются исходы m', m'', где $m' \neq m''$. Найдем вероятность неблагоприятного исхода. Она равна

$$\sum_{\substack{m',\, m'' \\ m'=m''}} \frac{1}{|M'|}\frac{1}{|C_{m'}|} = \frac{1}{|M'|}\sum_{m'}\frac{1}{|C_{m'}|} = \frac{1}{|M'|}\sum_{C_{m'}}\sum_{m'' \in C_{m'}}\frac{1}{|C_{m'}|} = \frac{1}{|M'|}\sum_{C_{m'}} 1 \leq \frac{|M|}{|M'|}.$$

Отсюда вероятность благоприятного исхода не меньше $\frac{1}{2}$. Следовательно, среднее число прохождений алгоритма через шаг 1 не превосходит 2, то есть этот алгоритм эффективен.

Хэш-функция Шаумома, ван Хейста, Пфицмана

Она основана на возведении в степень в конечном простом поле F_p, где $p - 1 = 2q$ и q — простое число. Пусть a примитивный элемент в F_p, и $b \in F_p^*$. Рассмотрим отображение

$$H : Z/q \times Z/q \to F_p^*,$$

определенное равенством

$$H(x_1, x_2) = a^{x_1} b^{x_2} \pmod p.$$

Доказано, что эффективный алгоритм вычисления коллизии для функции H существует тогда и только тогда, когда существует эффективный алгоритм вычисления такого $e \pmod{p-1}$, что $a^e \equiv b \pmod p$.

Эта хэш-функция считается хорошей. Так как известно, что вычисление дискретных логарифмов является трудной задачей. Однако она имеет два недостатка. Во-первых, не позволяет сжимать сообщения сколь угодно большой длины, во-вторых, это очень медленная функция, она использует возведение в степень в конечном поле. Если первый из недостатков может быть сравнительно легко устранен, то второй — нет.

Хэш-функции и блочные шифры

Рассмотрим один общий метод построения хэш-функций. Пусть f — любая функция, V_j — векторное пространство над полем F_2

$$f\colon V_n \times V_t \to V_t$$

при некоторых натуральных n, t. Требуется построить функцию

$$H\colon \bigcup_{k \geq n} V_k \to V_t.$$

Сообщение $m' \in \bigcup_{k \geq n} V_k$ разбивается на блоки

$$m' = m_1 \| m_2 \| \ldots \| m_r,$$

где $m_i \in V_n$. Если длина сообщения m' в битах не кратна n, то последний блок m_r дополняется таким образом, что $m_r \in V_n$. Тогда $H(m') = R_r$, где последовательность $R_0, R_1, \ldots, R_r$ вычисляется по следующему правилу. Блок R_0 фиксирован, например, $R_0 = 0^t$. Далее

$$R_i = f(m_i, R_{i-1}), \quad 1 \leq i \leq r.$$

Функцию f часто получают применением стойкого блочного шифра. Основная идея использования здесь шифра заключается в том, что трудно вычислить ключ, зная открытый текст и соответствующий ему шифротекст. То есть трудно восстановить k из равенства $E_k(m) = c$, где $m \in M$, $c \in C$ известны. Иначе говоря, функция $E_k(m)\colon K \to C$ является однонаправленной при фиксированном m. Пусть для дальнейшего $M = C = K = V_t$ и имеется блочный шифр $E_k\colon M \to C$, $k \in K$. Тогда в качестве f при $n = t$ берут:

$$f(m, R) = E_R(m) \oplus m,$$
$$f(m, R) = E_R(m) \oplus m \oplus R,$$
$$f(m, R) = E_R(m \oplus R) \oplus m,$$
$$f(m, R) = E_R(m \oplus R) \oplus m \oplus R,$$

где $\oplus$ — операция разрядного суммирования по $\bmod 2$ векторов из V_t. Возможны также и другие варианты для $f(m, R)$. Однако неосмотрительность при выборе функции $f(m, R)$ может привести к тому, что итоговая хэш-функция не будет однонаправленной. Это так при $f(m, R) = E_R(m)$.

Общий рецепт таков. В качестве f выбирать функцию, для которой трудно найти коллизию. То есть для f должно выполняться условие 3'). Очевидно, что для функций

$$f(m, R) = E_{m \oplus R}(R) \quad \text{и} \quad f(m, R) = E_C(m \oplus R) \oplus m \oplus R$$

при постоянном $c \in V_t$ это условие не выполняется.

Однонаправленные хэш-функции

Обычно на практике хэш-функция сжимает подписываемый документ M до нескольких десятков или сотен бит. Хэш-функция $H(\cdot)$ принимает в качестве аргумента сообщение (документ) M произвольной длины и возвращает хэш-значение $H(M) = z$ *фиксированной длины*.

Как было отмечено ранее, хэшированная информация является сжатым двоичным представлением основного сообщения произвольной длины. Напомним, что значение хэш-функции $H(M)$ сложным образом зависит от документа M и не позволяет восстановить сам документ M.

Хэш-функция должна удовлетворять целому ряду условий:

- хэш-функция должна быть чувствительна к всевозможным изменениям в тексте M, таким как вставки, выбросы, перестановки и т.п.;
- хэш-функция должна обладать свойством необратимости, то есть задача подбора документа M', который обладал бы требуемым значением хэш-функции, должна быть вычислительно неразрешима;
- вероятность того, что значения хэш-функций двух различных документов (вне зависимости от их длин) совпадут, должна быть ничтожно мала.

Большинство хэш-функций строится на основе однонаправленной функции $f(\cdot)$, которая образует выходное значение длиной n при задании двух входных значений длиной n (рис. 6.2). Этими входами являются блок исходного текста M_i и хэш-значение R_{i-1} предыдущего блока текста:

$$R_i = f(M_i, R_{i-1}).$$

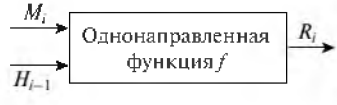


Рис. 6.2. Построение однонаправленной хэш-функции

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).

Однонаправленную хэш-функцию можно построить, используя симметричный блочный алгоритм. Наиболее очевидный подход со-

стоит в том, чтобы шифровать сообщение M посредством блочного алгоритма в специальных режимах CBC — сцепление блоков шифра или CFB — обратная связь по шифртексту с помощью фиксированного ключа и некоторого вектора инициализации. Последний блок шифртекста можно рассматривать в качестве хэш-значения сообщения M. При таком подходе не всегда возможно построить безопасную однонаправленную хэш-функцию, но всегда можно получить код аутентификации сообщения MAC (*Message Authentication Code*).

Более безопасный вариант хэш-функции можно получить, используя блок сообщения в качестве ключа, предыдущее хэш-значение — в качестве входа, а текущее хэш-значение — в качестве выхода. Реальные хэш-функции проектируются еще более сложными. Длина блока обычно определяется длиной ключа, а длина хэш-значения совпадает с длиной блока.

Поскольку большинство блочных алгоритмов являются 64-битовыми, некоторые схемы хэширования проектируют так, чтобы хэш-значение имело длину, равную двойной длине блока.

Если принять, что получаемая хэш-функция корректна, безопасность схемы хэширования базируется на безопасности лежащего в ее основе блочного алгоритма. Схема хэширования, у которой длина хэш-значения равна длине блока, показана на рис. 6.3.

Рис. 6.3. Обобщенная схема формирования хэш-функции

Ее работа описывается выражениями:

$$R_0 = I_R,$$
$$R_i = E_A(B) \oplus C,$$

где I_R — некоторое случайное начальное значение; A, B и C могут принимать значения M_i, R_{i-1}, $(M_i \oplus R_{i-1})$ или быть константами; E_A — функция шифрования на ключе A.

Три различные переменные A, B и C могут принимать одно из четырех возможных значений, поэтому в принципе можно получить 64 варианта общей схемы этого типа. Из них 52 варианта являются либо тривиально слабыми, либо небезопасными. Остальные 12 безопасных схем хэширования перечислены в табл. 6.1.

Таблица 6.1

Номер схемы	Функция хэширования
1	$R_i = E_{H_{i-1}}(M_i) \oplus M_i$
2	$R_i = E_{H_{i-1}}(M_i \oplus R_{i-1}) \oplus M_i \oplus R_{i-1}$
3	$R_i = E_{H_{i-1}}(M_i) \oplus R_{i-1} \oplus M_i$
4	$R_i = E_{H_{i-1}}(M_i \oplus R_{i-1}) \oplus M_i$
5	$R_i = E_{M_i}(R_{i-1}) \oplus R_{i-1}$
6	$R_i = E_{M_i}(M_i \oplus R_{i-1})\, M_i \oplus R_{i-1}$
7	$R_i = E_{M_i}(R_{i-1})\, M_i \oplus R_{i-1}$
8	$R_i = E_{M_i}(M_i \oplus R_{i-1}) \oplus R_{i-1}$
9	$R_i = E_{M_i \oplus H_{i-1}}(M_i) \oplus M_i$
10	$R_i = E_{M_i \oplus H_{i-1}}(R_{i-1}) \oplus R_{i-1}$
11	$R_i = E_{M_i \oplus H_{i-1}}(M_i) \oplus R_{i-1}$
12	$R_i = E_{M_i \oplus H_{i-1}}(R_{i-1}) \oplus M_i$

Первые четыре схемы хэширования считаются безопасными при всех атаках, они приведены на рис. 6.4.

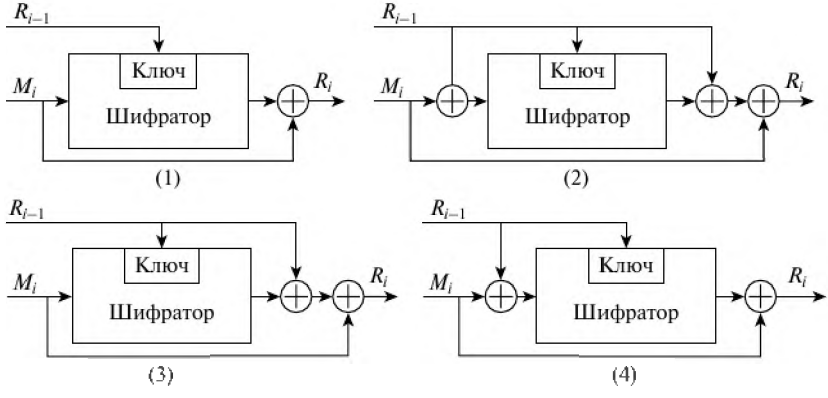


Рис. 6.4. Схемы безопасного хэширования на основе блочных алгоритмов

Отечественный стандарт хэш-функции

Российский криптографический стандарт, определяющий алгоритм и процедуру вычисления хэш-функции, введен 1 января 2013 года ГОСТ Р 34.11—2012 (полное название: «ГОСТ Р 34.11—2012. Инфор-

мационная технология. Криптографическая защита информации. Функция хэширования») взамен ГОСТ Р 34.11—94.

Стандарт разработан Центром защиты информации и специальной связи ФСБ России с участием ОАО «ИнфоТеКС». Данный стандарт определяет две функции хэширования с длинами $n = 256$ бит и $n = 512$ бит.

Преобразования. При вычислении хэш-кода $H(M)$ сообщения M используются следующие преобразования:

$$X[k]: V_{512} \to V_{512}, \quad X[k](a) = k \oplus a,\ k, a \in V_{512};$$
$$S: V_{512} \to V_{512}, \quad S(a) = S(a_{63}\|...\|a_0) = \pi(a_{63}\|...\|\pi a_0),$$

где $a = a_{63}\|...\|a_0 \in V_{512}$, $a_i \in V_8$, $i = 0, ..., 63$;

$$P: V_{512} \to V_{512}, \quad P(a) = P(a_{63}\|...\|a_0) = a_{\tau(63)}\|...\|a_{\tau(0)},$$

где $a = a_{63}\|...\|a_0 \in V_{512}$, $a_i \in V_8$, $i = 0, ..., 63$;

$$L: V_{512} \to V_{512}, \quad L(a) = L(a_7\|...\|a_0) = l(a_7)\|...\|l(a_0),$$

где $a = a_7\|...\|a_0 \in V_{512}$, $a_i \in V_{64}$, $i = 0, ..., 7$.

Функция сжатия. Значение хэш-кода сообщения M вычисляется с использованием итерационной процедуры. На каждой итерации вычисления хэш-кода используется функция сжатия:

$$g_N: V_{512} \times V_{512} \to V_{512}, \quad N \in V_{512},$$

значение которой вычисляется по формуле

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

где $E(K, m) = X[K13]LPSX[K12]...LPSX[K2]LPSX[K1](m)$.

Значения

$$K_i \in V_{512}, \quad i = 1, ..., 13$$

вычисляются следующим образом:

$$K_1 = K;$$
$$K_i = LPS(K_{i-1} \oplus C_{i-1}), \quad i = 2, ..., 13.$$

Процедура вычисления хэш-функции. Исходными данными для процедуры вычисления хэш-кода $H(M)$ является подлежащее хэшированию сообщение — двоичный вектор $|M| < 512$

$$M \in V^*$$

и инициализационный вектор хэширования:

$$IV \in V_{512},$$

где V^* — множество всех двоичных векторов-строк конечной размерности, включая пустую строку; V_n — множество всех n-мерных двоичных векторов, где n — целое неотрицательное число, нумерация подвекторов и компонент вектора осуществляется справа налево, начиная с нуля.

Значение инициализационного вектора IV для функции хэширования с длиной хэш-кода 512 бит равно 0^{512}. Значение инициализационного вектора IV для функции хэширования с длиной хэш-кода 256 бит равно $(00000001)^{64}$.

Алгоритм вычисления функции H состоит из следующих этапов.

Этап 1. Присвоить начальные значения текущих величин:

$$h: IV;$$
$$N: 0^{512} \in V_{512};$$
$$\Sigma := 0^{512} \in V_{512}.$$

Перейти к этапу 2.

Этап 2. Проверить условие $|M| < 512$. При положительном исходе перейти к этапу 3. В противном случае выполнить последовательность вычислений:

$$h: g_N(h, m);$$
$$N: \mathrm{Vec}_{512}(\mathrm{Int}_{512}(N) \boxplus 512);$$
$$\Sigma: \mathrm{Vec}_{512}(\mathrm{Int}_{512}(\Sigma) \boxplus \mathrm{Int}_{512}(m));$$
$$M: = M'.$$

Вновь вернуться к проверке условия $|M| < 512$.

Этап 3.

$$m := 0^{511-|M|} \| 1 \| M;$$
$$h := g_N(h, m);$$
$$N := \mathrm{Vec}_{512}(\mathrm{Int}_{512}(N) \boxplus |M|);$$
$$\Sigma: \mathrm{Vec}_{512}(\mathrm{Int}_{512}(\Sigma) \boxplus \mathrm{Int}_{512}(m));$$
$$h := g_0(h, N);$$
$$h := \begin{cases} g_0(h, N), & \text{для функции хэширования с длиной хэш-кода 512 бит;} \\ MSB_{256}(g_0(h, \Sigma)) & \text{для функции хэширования с длиной хэш-кода 256 бит.} \end{cases}$$

Значение величины h, полученное на последнем шаге, и является значением функции хэширования $H(M)$.

Примечание. В настоящем стандарте в целях сохранения терминологической преемственности по отношению к действующим отечественным нормативным документам и опубликованным научно-техническим изданиям установлено, что термины «электронная подпись», «цифровая подпись» и «электронная цифровая подпись» являются синонимами.

6.3. Алгоритм цифровой подписи RSA

Первой и наиболее известной во всем мире конкретной системой электронной подписи стала система RSA (подробное описание асимметричного алгоритма шифрования RSA (см. раздел 4.6).

Сначала необходимо вычислить пару ключей (секретный ключ и открытый ключ). Для этого отправитель электронных документов вычисляет два больших простых числа P и Q, затем находит их произведение

$$N = P \times Q$$

и значение функции Эйлера

$$\varphi(N) = (P - 1)(Q - 1).$$

Далее отправитель вычисляет число E из условий:

$$E \le \varphi(N), \text{ НОД}(E, \varphi(N)) = 1$$

и число D из условий:

$$D < N, E \times D \equiv 1 (\text{mod}\, \varphi(N)).$$

Пара чисел (E, N) является открытым ключом. Эту пару чисел автор передает партнерам по переписке для проверки его цифровых подписей. Число D сохраняется отправителем как секретный ключ для подписывания сообщений.

Обобщенная схема формирования и проверки электронной подписи RSA показана на рис. 6.5.

Допустим, что отправитель хочет подписать сообщение M перед его отправкой. Сначала сообщение M (блок информации, файл, таблица) сжимают с помощью хэш-функции $H(\cdot)$ в целое число m:

$$m = H(M).$$

Затем вычисляют цифровую подпись S под электронным документом M, используя хэш-значение m и секретный ключ D:

$$S = m^D (\text{mod}\, N).$$

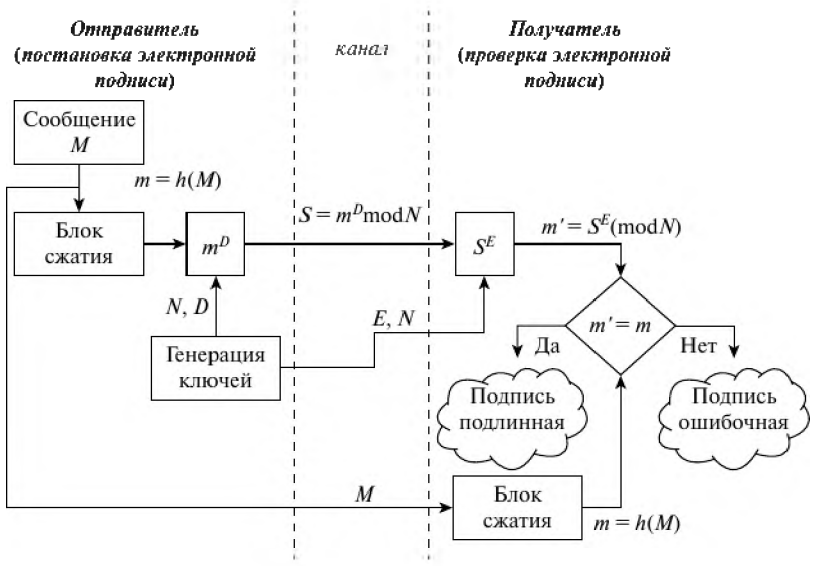


Рис. 6.5. Обобщенная схема цифровой подписи RSA

Пара (M, S) передается партнеру-получателю как электронный документ M, подписанный цифровой подписью S, причем подпись S сформирована обладателем секретного ключа D.

После приема пары (M, S) получатель вычисляет хэш-значение сообщения M двумя разными способами. Прежде всего он восстанавливает хэш-значение m', применяя криптографическое преобразование подписи S с использованием открытого ключа E:

$$m' = S^E (\text{mod } N).$$

Кроме того, он находит результат хэширования принятого сообщения M с помощью такой же хэш-функции $H(\cdot)$:

$$m = H(M).$$

Если соблюдается равенство вычисленных значений, т.е.

$$S^E (\text{mod } N) = H(M),$$

то получатель признает пару (M, S) подлинной. Доказано, что только обладатель секретного ключа D может сформировать цифровую подпись S по документу M, а определить секретное число D по открытому числу E не легче, чем разложить модуль N на множители.

Кроме того, можно строго математически доказать, что результат проверки цифровой подписи S будет положительным только в том

случае, если при вычислении S был использован секретный ключ D, соответствующий открытому ключу E. Поэтому открытый ключ E иногда называют «идентификатором» подписавшего.

Недостатки алгоритма цифровой подписи RSA

1. Можно повторно использовать подписанный документ, так как копирование файлов в компьютерных системах очень простая задача. Чтобы преодолеть этот недостаток, документ должен содержать, например, метку времени, а проверяющий подпись должен создать базу данных, содержащую метки времени получаемых от отправителя документов. Тогда при повторном предъявлении документа легко обнаружить попытку обмана.

2. Отправитель сообщения может подписать сообщение, а затем отказаться от подписи, заявив, что его секретный ключ был скомпрометирован.

3. Стойкость протоколов подписи RSA основана на сложности факторизации большого натурального числа N. В 1978 году, в момент опубликования RSA, считалось, что достаточно взять $N \approx 10^{100}$. В настоящее время выбор $N \approx 10^{150}$ уже не обеспечивает защиту от подделки. Таким образом, лицо, подписавшее документ в 1978 г., может сейчас отказаться от своей подписи под этим документом. Или злоумышленник может подделывать подписи под документами, датировав их 1978 г. Такое свойство электронной подписи приводит к тому, что будущие историки не будут доверять электронным документам нашего времени.

4. При вычислении модуля N, ключей E и D для системы цифровой подписи RSA необходимо проверять большое количество дополнительных условий, что сделать трудно практически. Невыполнение любого из этих условий делает возможным фальсификацию цифровой подписи со стороны того, кто обнаружит такое невыполнение. При подписании важных документов нельзя допускать такую возможность даже теоретически.

5. Для обеспечения криптостойкости цифровой подписи RSA по отношению к попыткам фальсификации необходимо использовать при вычислениях N, D и E целые числа длиной не менее 512 бит, что требует больших вычислительных затрат, превышающих на 20—30% вычислительные затраты других алгоритмов цифровой подписи при сохранении того же уровня криптостойкости.

6. Цифровая подпись RSA уязвима к так называемой мультипликативной атаке. Иначе говоря, алгоритм цифровой подписи RSA позво-

ляет злоумышленнику без знания секретного ключа D сформировать подписи под теми документами, у которых результат хэширования можно вычислить как произведение результатов хэширования уже подписанных документов.

Примеры

Допустим, что злоумышленник может сконструировать три сообщения M_1, M_2 и M_3, у которых хэш-значения

$$m_1 = h(M_1), \quad m_2 = h(M_2), \quad m_3 = h(M_3),$$

причем $m_3 = m_1 m_2 (\mathrm{mod}\, N)$.

Допустим также, что для двух сообщений M_1 и M_2 получены законные подписи

$$S_1 = m_1^D (\mathrm{mod}\, N) \quad \text{и} \quad S_2 = m_2^D (\mathrm{mod}\, N).$$

Тогда злоумышленник может легко вычислить подпись S_3 для документа M_3, даже не зная секретного ключа D:

$$S_3 = S_1 S_2 (\mathrm{mod}\, N).$$

Действительно,

$$S_1 S_2 (\mathrm{mod}\, N) = m_1^D m_2^D (\mathrm{mod}\, N) = (m_1 m_2)^D (\mathrm{mod}\, N) = m_3^D (\mathrm{mod}\, N) = S_3.$$

6.4. Алгоритм цифровой подписи Эль Гамаля

Алгоритм цифровой подписи Эль Гамаля основан на сложной вычислительной задаче дискретного логарифмирования.

Этот протокол, опубликованный в 1985 г., послужил прототипом стандартов электронной подписи США и России.

Зафиксируем конечное поле F_p, где p — простое число. Пусть g — первообразный вычет по $\mathrm{mod}\, p$. Вычет $x (\mathrm{mod}\, p - 1)$ является секретным ключом отправителя A, а вычет $y \equiv g^x (\mathrm{mod}\, p)$ является его открытым ключом. Обозначим через $m \in Z_{p-1}$ сообщение (точнее, его хэш-значение), которое хочет подписать пользователь A и отправить его пользователю B. Последовательность действий следующая.

1) Пользователь A выбирает случайное секретное $k \in \left(Z_{p-1}\right)^*$, вычисляет $a \equiv g^x (\mathrm{mod}\, p)$, $0 < a < p$.

2) Отправитель сообщения A, применяя расширенный алгоритм Евклида, вычисляет с помощью секретного ключа x целое число $b \equiv (m - xa) \cdot k^{-1} (\mathrm{mod}\, p - 1)$ из уравнения $m = x \times a + k \times b\ (\mathrm{mod}\, (p - 1))$

и формирует подписанное сообщение (m, a, b), где пара, (a, b) и есть подпись сообщения m.

3) Для проверки подписи, получатель B вычисляет

$$g^m \pmod p \quad \text{и} \quad y^a a^b \pmod p.$$

Если вычисленные значения совпадают, то подпись принимается, в противном случае отвергается. Нетрудно проверить, что если подпись правильная, то выполняется сравнение

$$g^m \equiv y^a a^b \pmod p.$$

Следует отметить, что выполнение каждой подписи по методу Эль Гамаля требует нового значения k, причем это значение должно выбираться случайным образом. Если нарушитель раскроет когда-либо значение k, повторно используемое отправителем, то он сможет раскрыть секретный ключ x отправителя.

Примеры

Выберем: числа $p = 11$, $g = 2$ и секретный ключ $x = 8$.

Вычисляем значение открытого ключа:

$$y = g^x \bmod p = 2^8 \bmod 11 = 3.$$

Предположим, что исходное сообщение M характеризуется хэш-значением $m = 5$.

Для того чтобы вычислить цифровую подпись для сообщения M, имеющего хэш-значение $m = 5$, сначала выберем случайное целое число $k = 9$.

Убедимся, что числа k и $(p - 1)$ являются взаимно простыми. Действительно,

$$\text{НОД}(9, 10) = 1.$$

Далее вычисляем элементы a и b подписи:

$$a = g^k \bmod p = 2^9 \bmod 11 = 6,$$

элемент b определяем, используя расширенный алгоритм Евклида:

$$m = x \times a + k \times b \pmod{(p - 1)}.$$

При $m = 5$, $a = 6$, $x = 8$, $k = 9$, $p = 11$ получаем $5 = (6 \times 8 + 9 \times b) \pmod{10}$.

Или $9 \times b \equiv -43 \pmod{10}$.

Решение: $b = 3$. Цифровая подпись представляет собой пару: $a = 6$, $b = 3$.

Далее отправитель передаст подписанное сообщение.

Приняв подписанное сообщение и открытый ключ $y = 3$, получатель вычисляет хэш-значение $m = 5$ для сообщения M, а затем вычисляет два числа:

1) $y^a a^b \pmod p = 3^6 \cdot 6^3 \pmod{11} = 10 \pmod{11}$;

2) $g^m \pmod p = 2^5 \pmod{11} = 10 \pmod{11}$.

Так как эти два целых числа равны, принятое получателем сообщение признается подлинным.

Следует отметить, что схема Эль Гамаля является характерным примером подхода, который допускает пересылку сообщения M в открытой форме вместе с присоединенным аутентификатором (a, b). В таких случаях процедура установления подлинности принятого сообщения состоит в проверке соответствия аутентификатора сообщению.

6.5. Алгоритм цифровой подписи DSA

Американский стандарт цифровой подписи состоит из трех частей: алгоритм хэширования SHA (*Secure Hash Algorithm*); алгоритм порождения параметров p, q и алгоритм подписи DSA (*Digital Signature Algorithm*). Алгоритм цифровой подписи DSA используется в стандарте цифровой подписи DSS (*Digital Signature Standard*).

Отправитель и получатель электронного документа используют при вычислении большие простые числа: g и p по L бит каждое ($512 \le L \le 1024$); q — простой делитель числа $(p-1)$, число q имеет длину 160 бит. Числа g, p, q являются открытыми и могут быть общими для всех пользователей сети. Ниже используются обозначения $\frac{1}{a}$ для обозначения обратного элемента для a относительно умножения по заданному модулю, короче: $\frac{1}{a} = a^{-1}$.

Отправитель выбирает случайное целое число x, $1 < x < q$. Число x является секретным ключом отправителя для формирования электронной подписи.

Затем отправитель вычисляет значение

$$y = g^x \bmod p.$$

Число y является открытым ключом для проверки подписи отправителя. Число y передается всем получателям документов.

Этот алгоритм также предусматривает использование односторонней функции хэширования $h(\cdot)$. В стандарте DSS определен так называемый *алгоритм безопасного хэширования* SHA (*Secure Hash Algorithm*).

Для того чтобы подписать документ M, отправитель хэширует его в целое хэш-значение m:

$$m = H(M), \quad 1 < m < q,$$

затем генерирует случайное целое число k, $1 < k < q$, и вычисляет число r:

$$r = (g^k \bmod p) \bmod q.$$

Затем отправитель вычисляет с помощью секретного ключа x целое число s:

$$s = \frac{m + rx}{k} \bmod q.$$

Пара чисел r и s образует цифровую подпись (r, s) под документом M.

Таким образом, подписанное сообщение представляет собой тройку чисел $[M, r, s]$.

Получатель подписанного сообщения $[M, r, s]$ проверяет выполнение условий

$$0 < r < q,$$
$$0 < s < q$$

и отвергает подпись, если хотя бы одно из этих условий не выполнено.

Затем получатель вычисляет значение

$$w = \frac{1}{s} \bmod q,$$

и хэш-значение

$$m = H(M),$$

а затем числа

$$u_1 = (m \times w) \bmod q,$$
$$u_2 = (r \times w) \bmod q.$$

Далее получатель с помощью открытого ключа y вычисляет значение

$$v = ((g^{u_1} y^{u_2}) \bmod p) \bmod q$$

и проверяет выполнение условия

$$v = r.$$

Если условие $v = r$ выполняется, тогда подпись (r, s) под документом M признается получателем подлинной.

Доказано, что последнее равенство будет выполняться тогда, и только тогда, когда подпись (r, s) под документом M получена с помощью именно с того секретного ключа x, из которого был получен открытый ключ y. Таким образом, можно надежно удостовериться, что отправитель сообщения владеет именно данным секретным ключом x (не раскрывая при этом значения ключа x) и что данный документ M подписал именно отправитель.

По сравнению с алгоритмом цифровой подписи Эль Гамаля алгоритм *DSA* имеет следующие основные преимущества.

1. При любом допустимом уровне стойкости, т.е. при любой паре чисел g и p (от 512 до 1024 бит), числа q, x, r, s имеют длину по 160 бит, сокращая длину подписи до 320 бит.

2. Большинство операций с числами k, r, s, x при вычислении подписи производится по модулю числа q длиной 160 бит, что сокращает время вычисления подписи.

3. При проверке подписи большинство операций с числами u_1, u_2, v, w также производится по модулю числа q длиной 160 бит, что сокращает объем памяти и время вычисления.

Недостатком алгоритма DSA является то, что при подписывании и при проверке подписи приходится выполнять сложные операции деления по модулю q:

$$s = \frac{m + rx}{k} (\mathrm{mod}\, q), \qquad w = \frac{1}{s} (\mathrm{mod}\, q),$$

что не позволяет получать максимальное быстродействие.

Следует отметить, что реальное исполнение алгоритма DSA может быть ускорено с помощью выполнения предварительных вычислений. Заметим, что значение r не зависит от сообщения M и его хэш-значения m. Можно заранее создать строку случайных значений k и затем для каждого из этих значений вычислить значения r. Можно также заранее вычислить обратные значения k^{-1} для каждого из значений k. Затем при поступлении сообщения M можно вычислить значение s для данных значений r и k^{-1}. Эти предварительные вычисления значительно ускоряют работу алгоритма DSA.

Стойкость алгоритма подписи DSA определяется сложностью решения задачи дискретного логарифмирования в подгруппе $\langle g \rangle < F_p^*$ порядка q.

При анализе алгоритма подписи Эль Гамаля мы видели, что централизованное порождение параметров может быть опасно. Это было учтено при разработке стандарта США. Помимо простых чисел p и q пользователю предоставляют значения S и C параметров алгоритма выработки этих p и q. Зная S, C, пользователь может проверить, действительно ли числа p, q были получены с помощью этого алгоритма. Приведем теперь сам алгоритм.

Входные данные алгоритма. Число L, где $512 \le L \le 1024$ и $L \equiv 0 (\mathrm{mod}\, 64)$. Пусть $L - 1 = 160n + b$, где $0 \le b \le 160$.

Выходные данные алгоритма. Простое число p длины L бит, простое число q длины 160 бит, двоичная последовательность S и число C, $0 \le C < 4095 = 2^{12} - 1$.

Шаг 1. Выбрать произвольную последовательность бит S, длина g которой не менее 160.

Шаг 2. Вычислить

$$U = SHA(S) \oplus SHA((S+1) \bmod 2^g)$$

Таким образом, U — вектор длины 160. Здесь $SHA(S)$ есть результат применения алгоритма *SHA*.

Шаг 3. Образовать q, установив первый и 160-й бит U в 1. Таким образом, q — нечетное число длиной 160 бит.

Шаг 4. Проверить q на простоту.

Шаг 5. Если q составное, то перейти на шаг 1. Если q простое, то перейти на шаг 6.

Шаг 6. Положить $C = 0$, $N = 2$.

Шаг 7. Вычислить

$$V_k = SHA((S + N + k) \bmod 2^g)$$

при всех k, $0 \le k \le n$.

Шаг 8. Пусть W целое число

$$W = V_0 + 2^{160} V_1 + \ldots + 2^{160(n-1)} V_{n-1} + 2^{160n} (V_n \bmod 2^b)$$

длины не более L — 1 бит.

Положить

$$X = W + 2^{L-1}.$$

Таким образом, X — число, длина которого в точности равна L бит.

Шаг 9. Положить

$$p = X - ((X \bmod 2q) - 1).$$

Тогда $p \equiv 1 \pmod{2q}$.

Шаг 10. Если $p < 2^{L-1}$, то перейти к шагу 13.

Шаг 11. Проверить p на простоту.

Шаг 12. Если p простое, то перейти к шагу 15.

В противном случае перейти на шаг 13.

Шаг 13. Положить $C := C + 1$ и $N := N + n + 1$.

Шаг 14. Если $C = 4096$, то перейти на шаг 1.

В противном случае перейти к шагу 7.

Шаг 15. Сохранить числа p, q, C и последовательность S.

Закончить выполнение операций.

Заметим, что для проверки простоты чисел p, q рекомендуется использовать вероятностный тест, то есть тест типа Монте-Карло с вероятностью ошибки не более 2^{-80}.

6.6. Цифровые подписи с дополнительными функциональными свойствами

Рассматриваемые в этом разделе цифровые подписи обладают дополнительными функциональными возможностями, помимо обычных свойств аутентификации сообщения и невозможности отказа подписавшего лица от обязательств, связанных с подписанным текстом. В большинстве случаев они объединяют базовую схему цифровой подписи, например, на основе алгоритма RSA, со специальным протоколом, обеспечивающим достижение тех дополнительных свойств, которыми базовая схема цифровой подписи не обладает.

К схемам цифровой подписи с дополнительными функциональными свойствами относятся:

— схемы слепой (*blind*) подписи;

— схемы неоспоримой (*undeniable*) подписи.

Схемы слепой подписи. В отличие от обычных схем цифровой подписи *схемы слепой подписи* (иногда называемые схемами подписи вслепую) являются двусторонними протоколами между отправителем A и стороной B, подписывающей документ. Основная идея этих схем заключается в следующем. Отправитель A посылает порцию информации стороне B, которую B подписывает и возвращает A. Используя полученную подпись, сторона A может вычислить подпись стороны B на более важном для себя сообщении m. По завершении этого протокола сторона B ничего не знает ни о сообщении m, ни о подписи под этим сообщением.

Цель слепой подписи состоит в том, чтобы воспрепятствовать подписывающему лицу B ознакомиться с сообщением стороны A, которое он подписывает, и с соответствующей подписью под этим сообщением. Поэтому в дальнейшем подписанное сообщение невозможно связать со стороной A.

Приведем пример применения слепой подписи. Схема слепой подписи может найти применение в тех случаях, когда отправитель A (клиент банка) не хочет, чтобы подписывающая сторона B (банк) имела возможность в дальнейшем связать сообщение m и подпись $s_B(m)$ с определенным шагом выполненного ранее протокола.

В частности, это может быть важно при организации анонимных безналичных расчетов, когда сообщение m могло бы представлять денежную сумму, которую A хочет потратить. Когда сообщение m с подписью $s_B(m)$ предъявляется банку B для оплаты, банк B не сможет проследить, кто именно из клиентов предъявляет подписанный документ. Это позволяет пользователю A остаться анонимным.

Для построения протокола слепой подписи необходимы следующие компоненты.

1. Механизм обычной цифровой подписи для подписывающей стороны B. Пусть $s_B(X)$ обозначает подпись стороны B на документе X.

2. Функции $f(\cdot)$ и $g(\cdot)$ (известные только отправителю) такие, что

$$g(s_B(f(m))) = s_m(m),$$

при этом $f(\cdot)$ — маскирующая (*blinding*) функция;
$g(\cdot)$ — демаскирующая (*unblinding*) функция;
$f(m)$ — замаскированное (*blinded*) сообщение m.

При выборе s_B, f и g существует ряд ограничений.

Выберем в качестве алгоритма подписи s_B для стороны B схему цифровой подписи RSA с открытым ключом (N, E) и секретным ключом D, причем $N = P \times Q$ — произведение двух больших случайных простых чисел. Пусть k — некоторое фиксированное целое число, взаимно простое с N, т.е. НОД$(N, k) = 1$.

Маскирующая функция f: $Z_n \to Z_n$ определяется как $f(m) = mk^E \bmod N$, а демаскирующая функция g: $Z_n \to Z_n$ определяется как $g(m) = k^{-1}m \bmod N$. При таком выборе f, g и s получаем $g(s_B(f(m))) = g(s_B(mk^E \bmod N)) = g(m^D k \bmod N) = m^D \bmod N = s_B(m)$, что соответствует требованию 2.

Согласно протоколу слепой подписи, который предложил Дэвид Чом (*David Chaum*)[1], отправитель A сначала получает подпись стороны B на замаскированном сообщении m^*. Используя эту подпись, сторона A вычисляет подпись B на заранее выбранном сообщении m, где $0 \le m \le N - 1$. При этом стороне B ничего неизвестно ни о значении m, ни о подписи, связанной с m.

Пусть сторона B имеет для подписи по схеме RSA открытый ключ (N, E) и секретный ключ D. Пусть k — случайное секретное целое число, выбранное стороной A и удовлетворяющее условиям $0 \le k \le N - 1$ и НОД(N, k).

Протокол слепой подписи Д. Чома включает следующие шаги.

1. Отправитель A вычисляет замаскированное сообщение $m^* = mk^E \bmod N$ и посылает его стороне B.

2. Подписывающая сторона B вычисляет подпись $s^* = (m^*)^D \bmod N$ и отправляет эту подпись стороне A.

[1] Дэвид Чом — создатель многих криптографических протоколов, включая слепые схемы подписи, и электронных денег. В 1990 году Дэвид Чом получил степень доктора наук по информатике и менеджменту Калифорнийского университета Беркли. Впоследствии он преподавал в Нью-йоркской школе дипломированного специалиста университета менеджмента и в Университете Калифорнии. Чом является профессором католического университета Leuven.

3. Сторона A вычисляет подпись $s = k^{-1}s^* \bmod N$, которая является подписью B на сообщение m.

Нетрудно видеть, что

$$(m^*)^D \equiv (mk^E)^D \equiv m^D k (\bmod N),$$

поэтому

$$k^{-1}s^* \equiv m^D k k^{-1} \equiv m^D (\bmod N).$$

Д. Чом разработал несколько алгоритмов слепой подписи для создания системы анонимных безналичных электронных расчетов *eCash*.

Схемы неоспоримой подписи. Неоспоримая подпись, как и обычная цифровая подпись, зависит от подписанного документа и секретного ключа. Однако в отличие от обычных цифровых подписей неоспоримая подпись не может быть верифицирована без участия лица, поставившего эту подпись. Возможно, более подходящим названием для этих подписей было бы: «подписи, не допускающие подлога».

Рассмотрим два возможных сценария применения неоспоримой подписи.

Сценарий 1. Сторона A (клиент) хочет получить доступ в защищенную зону, контролируемую стороной B (банком). Этой защищенной зоной может быть, например, депозитарий (хранилище ценностей клиентов). Сторона B требует от A поставить на заявке о допуске в защищенную зону подпись, время и дату до предоставления ему доступа. Если A применит неоспоримую подпись, тогда сторона B не сможет впоследствии доказать кому-либо, что A получил допуск, без непосредственного участия A в процессе верификации подписи.

Сценарий 2. Предположим, что известная корпорация A разработала пакет программного обеспечения. Чтобы гарантировать подлинность пакета и отсутствие в нем вирусов, сторона A подписывает этот пакет неоспоримой подписью и продает его стороне B. Сторона B решает сделать копии этого пакета программного обеспечения и перепродать его третьей стороне C. При использовании стороной A неоспоримой подписи сторона C не сможет убедиться в подлинности этого пакета программного обеспечения и отсутствии в нем вирусов без участия стороны A.

Конечно, этот сценарий не препятствует стороне B поставить на пакете свою подпись, но тогда для стороны B будут утрачены все маркетинговые преимущества, связанные с использованием торговой марки корпорации A. Кроме того, будет легче раскрыть мошенническую деятельность стороны B.

Тест к главе 6

1. ***Электронная подпись в соответствии с Федеральным законом от 6 апреля 2011 года № 63-ФЗ «Об электронной подписи» — это*:**
   1) информация в электронной форме, которая присоединена к другой информации в электронной форме (подписываемой информации) или иным образом связана с такой информацией и которая используется для определения лица, подписывающего информацию;
   2) реквизит электронного документа, предназначенный для защиты данного электронного документа от подделки, полученный в результате криптографического преобразования информации с использованием закрытого ключа электронной цифровой подписи и позволяющий идентифицировать владельца сертификата.
2. ***Федеральный закон от 6 апреля 2011 года № 63-ФЗ «Об электронной подписи» регулирует отношения в области использования электронных подписей (несколько верных ответов)*:**
   1) при оказании государственных и муниципальных услуг;
   2) совершении гражданско-правовых сделок;
   3) исполнении государственных и муниципальных функций;
   4) совершении иных юридически значимых действий.
3. ***Ключ электронной подписи — это*:**
   1) уникальная последовательность символов, однозначно связанная с ключом электронной подписи и предназначенная для проверки подлинности электронной подписи (далее — проверка электронной подписи);
   2) уникальная последовательность символов, предназначенная для создания электронной подписи.
4. ***Ключ проверки электронной подписи — это*:**
   1) уникальная последовательность символов, однозначно связанная с ключом электронной подписи и предназначенная для проверки подлинности электронной подписи (далее — проверка электронной подписи);
   2) уникальная последовательность символов, предназначенная для создания электронной подписи.
5. ***Удостоверяющий центр — это*:**
   1) юридическое лицо или индивидуальный предприниматель, осуществляющие функции по созданию и выдаче сертификатов ключей проверки электронных подписей, а также иные функции, предусмотренные настоящим Федеральным законом;

2) осуществляющий обмен информацией в электронной форме государственный орган, орган местного самоуправления или организация;
3) лицо, которому в установленном законом порядке выдан сертификат ключа проверки электронной подписи.

6. ***Сертификат ключа проверки электронной подписи — это*:**
1) уникальная последовательность символов, однозначно связанная с ключом электронной подписи и предназначенная для проверки подлинности электронной подписи (далее — проверка электронной подписи);
2) электронный документ или документ на бумажном носителе, выданные удостоверяющим центром либо доверенным лицом удостоверяющего центра и подтверждающие принадлежность ключа проверки электронной подписи владельцу сертификата ключа проверки электронной подписи.

7. ***Видами электронных подписей в соответствии с Федеральным законом № 63-ФЗ являются* (*несколько верных ответов*):**
1) простая электронная подпись;
2) простая неквалифицированная электронная подпись;
3) усиленная неквалифицированная электронная подпись;
4) усиленная квалифицированная электронная подпись.

8. ***Простой электронной подписью в соответствии с Федеральным законом № 63-ФЗ является*:**
1) электронная подпись, для которой ключ проверки электронной подписи указан в квалифицированном сертификате;
2) электронная подпись, которая посредством использования кодов, паролей или иных средств подтверждает факт формирования электронной подписи определенным лицом;
3) электронная подпись, которая получена в результате криптографического преобразования информации с использованием ключа электронной подписи.

9. ***Неквалифицированной электронной подписью в соответствии с Федеральным законом № 63-ФЗ является электронная подпись, которая* (*несколько верных ответов*):**
1) получена в результате криптографического преобразования информации с использованием ключа электронной подписи;
2) позволяет определить лицо, подписавшее электронный документ;
3) позволяет обнаружить факт внесения изменений в электронный документ после момента его подписания;
4) создается с использованием средств электронной подписи.

10. ***Квалифицированной электронной подписью в соответствии с Федеральным законом № 63-ФЗ является*:**
    1) электронная подпись, которая посредством использования кодов, паролей или иных средств подтверждает факт формирования электронной подписи определенным лицом;
    2) электронная подпись, которая соответствует всем признакам неквалифицированной электронной подписи, а также ключ проверки электронной подписи указан в квалифицированном сертификате;
    3) электронная подпись, которая получена в результате криптографического преобразования информации с использованием ключа электронной подписи.
11. ***В чем отличие рукописной подписи от электронной (несколько верных ответов)*:**
    1) электронную подпись можно подделать, а рукописную подпись нельзя;
    2) с помощью электронной подписи можно определить реальное время подписания документа в отличие от рукописной подписи;
    3) в рукописной подписи содержатся метка времени, а в электронной подписи не содержится;
    4) рукописную подпись можно удостоверить в отличие от электронной подписи;
    5) дополнительное свойство электронной подписи — контроль целостности подписанного документа.
12. ***При отправке по электронной почте подписанного ЭЦП документа будет отправлен*:**
    1) сам файл и подпись файла;
    2) только файл;
    3) только подпись.
13. ***При подписании файла электронной цифровой подписью*:**
    1) создается новая версия файла, в которую добавляется подпись;
    2) все версии файла преобразуются с помощью крипто-алгоритмов ЭЦП;
    3) к файлу добавляется подпись, при этом сам файл не меняется.
14. ***Разрешается ли редактирование файла, подписанного ЭЦП*:**
    1) нет;
    2) да;
    3) разрешается только пользователю с полными правами.

15. ***Может ли документ одновременно быть зашифрованным и подписанным ЭЦП*:**
    1) да;
    2) нет.
16. ***При подписании документа ЭЦП используется*:**
    1) открытый ключ;
    2) секретный ключ;
    3) сертификат ключа.
17. ***При шифровании файла с использованием асимметричной криптосистемы используется*:**
    1) открытый ключ;
    2) секретный ключ;
    3) открытый и секретный ключи совместно.
18. ***Какой ключ пользователя необходимо использовать при расшифровке файла, зашифрованного с использованием асимметричной криптосистемы*:**
    1) секретный;
    2) открытый.
19. ***Хэш-функции от документов разной длины будут иметь*:**
    1) одинаковую длину, определенную стандартом функции хэширования;
    2) разную длину;
    3) длину, которая определяется алгоритмом постановки электронной подписи.
20. ***Хэш-функция используется* (*несколько верных ответов*):**
    1) для создания сжатого образа сообщения, применяемого в ЭЦП;
    2) быстрой передачи данных;
    3) идентификации отправителя;
    4) построения кода аутентификации сообщений.
21. ***Какую роль выполняет электронная цифровая подпись*:**
    1) роль дополнительной информации о передаваемых данных;
    2) это данные о времени передачи информации;
    3) роль обычной подписи в электронных документах;
    4) роль обратного адреса отправителя.
22. ***При формировании цифровой подписи по классической схеме отправитель* (*последовательность из двух действий*):**
    1) применяет к исходному тексту хэш-функцию;
    2) применяет к исходному тексту идентификатор отправителя;
    3) выполняет максимальное сжатие;
    4) вычисляет ЭЦП по хэш-образу с использованием секретного ключа создания подписи.

23. ***При верификации подписи получатель отделяет цифровую подпись от основного текста и выполняет проверку (последовательность из двух действий)*:**

1) дополнительной информации;
2) применяет к тексту полученного сообщения хэш-функцию;
3) проверяет соответствие хэш-образа сообщения полученной цифровой подписи с использованием открытого ключа проверки подписи;
4) проверяет исходный код.

24. ***Первый ФЗ «Об электронной цифровой подписи» в Российской Федерации был принят*:**

1) в 2000 г.;
2) в 2002 г.;
3) в 2006 г.

Таблица ответов на тест к главе 6

Номер вопроса	1	2	3	4	5	6	7	8
Правильный ответ	1	1, 2, 3, 4	2	1	1	2	1, 3, 4	2
Номер вопроса	9	10	11	12	13	14	15	16
Правильный ответ	1, 2, 3, 4	2	2, 5	1	3	1	1	2
Номер вопроса	17	18	19	20	21	22	23	24
Правильный ответ	1	1	1	1, 4	3	1, 4	2, 3	2

ЛИТЕРАТУРА

1. *Аграновский А.В., Хади Р.А.* Практическая криптография. Алгоритмы и их программирование (+CD-ROM). СПб., СОЛОН-Пресс, 2002.
2. Анин Б. Защита компьютерной информации. Серия «Мастер». — СПб. : БХВ-Петербург, 2002. — 384 с.
3. *Асосков А.В., Иванов М.А., Мирский А.А.* и др. Поточные шифры. КУДИЦ-Образ, 2003.
4. *Бабаш А.В., Шанкин Г.П.* История криптографии. Ч. 1. М. : Гелиос АРВ, 2002. — 240 с.
5. *Бабаш А.В., Шанкин Г.П.* Криптография / под редакцией В.П. Шерстюка, Э.А. Применко. М. : СОЛОН-Р, 2002. — 512 с.
6. *Бабаш А.В.* Криптографические и теоретико-автоматные аспекты современной защиты информации. Том 1. М. : Изд. центр ЕАОИ, 2009. — 414 с.
7. *Бабаш А.В., Баранова Е.К., Мельников Ю.Н.* Информационная безопасность. Лабораторный практикум (+CD) : учебное пособие. М. : КНОРУС, 2012. — 136 с.
8. *Бабенко Л.К., Ищукова Е.А.* Современные алгоритмы блочного шифрования и методы их анализа. М. : Гелиос АРВ, 2006.
9. *Баричев С.Г., Гончаров В.В., Серов Р.Е.* Основы современной криптографии. М. : Горячая линия-Телеком, 2001. — 120 с.
10. *Баранова Е.К., Бабаш А.В.* Криптографические методы защиты информации. Лабораторный практикум : учеб. пособие (+CD-ROM) М. : КНОРУС, 2015. — 200 с.
11. *Бернет С., Пэйн С.* Криптография. Официальное руководство RSA Security. М. : Бином, 2002.
12. *Василенко О.Н.* Теоретико-числовые алгоритмы в криптографии. М. : Московский центр непрерывного математического образования, 2003. — 328 с.
13. *Вельшенбах М.* Криптография на Си и С++ в действии (+CD-ROM). М. : Триумф, 2004.
14. *Гатчин Ю.А., Коробейников А.Г.* Основы криптографических алгоритмов : учебное пособие. СПб. : СПбГИТМО (ТУ), 2002. — 29 с.
15. *Гашков С.Б., Болотов А.А., Фролов А.Б., Часовских А.А.* Элементарное введение в эллиптическую криптографию. М. : КомКнига, 2006.
16. *Зубов А.Ю.* Криптографические методы защиты информации. М. : Гелиос АРВ, 2005. — 192 с.
17. *Коутинхо С.* Введение в теорию чисел. Алгоритм RSA. М. : Постмаркет, 2001. — 328 с.
18. *Корт С.С.* Теоретические основы защиты информации : Учебное пособие. М. : Гелиос АРВ, 2004. — 240 с.
19. *Коблиц Н.* Курс теории чисел и криптографии. М. : ТВП, 2001.
20. *Корченко А.Г.* Построение систем защиты информации на нечетких множествах. М. : МК-Пресс, 2006.

21. *Масленников М.М.* Практическая криптография (+CD-ROM). СПб., BHV — Санкт — Петербург, 2003.
22. *Молдовян А.А.*, *Молдовян Н.А.*, *Советов Б.Я.* Криптография : учебник для вузов. М. : Лань, 2005.
23. *Молдовян А.А.*, *Молдовян Н.А.*, *Гуц Н.Д.*, *Изотов Б.В.* Криптография. Скоростные шифры. СПб., БХВ-Петербург, 2002.
24. *Мао В.* Современная криптография: теория и практика. М. : Издательский дом «Вильямс», 2005.
25. *Маховенко Е.Б.* Теоретико-числовые методы в криптографии. М. : Гелиос АРВ, 2006.
26. *Новиков Ф.А.* Дискретная математика для программистов. СПб. : Питер, 2001. — 304 с.
27. *Осипян В.О.*, *Осипян К.В.* Криптография в задачах и упражнениях. М. : Гелиос АРВ, 2004.
28. *Ростовцев А.Г.*, *Маховенко Е.Б.* Введение в криптографию с открытым ключом. СПб. : «Мир и Семья», 2001.
29. *Смарт Н.* Криптография. М. : Техносфера, 2006. — 528 с.
30. *Утешев А.Ю.*, *Черкасов Т.М.*, *Шапошников А.А.* Цифры и шифры. СПб. : Изд-во СПбГУ, 2001.
31. *Черемушкин А.В.* Лекции по арифметическим алгоритмам в криптографии. М. : МЦНМО, 2002. — 104 с.
32. *Чмора А.* Современная прикладная криптография. М. : Гелиос, 2002.
33. *Шеннон К.* Работы по теории информации и кибернетике. М. ИЛ, 1963. — 830 с. (Раздел — Теория связи в секретных системах.)
34. *Шнайер Брюс.* Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М. : Триумф, 2002.
35. *Щербаков А.Ю.*, *Домашев А.В.* Прикладная криптография. Использование и синтез криптографических интерфейсов. М. : Издательско-торговый дом «Русская редакция», 2003. — 416 с.
36. *Фомичев В.М.* Дискретная математика и криптология. Курс лекций. М. : Диалог-МИФИ, 2003.
37. *Харин Ю.С.*, *Берник В.И.*, *Матвеев Г.В.*, *Агиевич С.В.* Математические и компьютерные основы криптологии: учебное пособие. М. : Новое издание, 2003.

ПРИЛОЖЕНИЕ 1

Контрольные задания к главе 1

Вариант 1

1. Найдите произведение подстановок:
$G1 \cdot G2 = ?$

$$G1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 1 & 5 & 4 & 6 & 2 \end{pmatrix}; \quad G2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 6 & 1 & 5 & 3 \end{pmatrix}.$$

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

2. Найдите обратную подстановку к данной:

$$G = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 5 & 3 & 1 & 4 \end{pmatrix}; \quad G^{-1} = ?$$

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

3. Определите порядок подстановки:

$$G = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 1 & 2 & 5 & 3 \end{pmatrix}.$$

Ответ. 5.

4. Вычислить сумму 4 и 5 по модулю n. Множество {1, 2, 3, 4, 5, 6, 7}.

Ответ. 2.

5. Вычислить произведение 6 и 7 по модулю n. Множество {1, 2, 3, 4, 5, 6, 7}.

Ответ. 6.

6. Множество {0, 1, 2, 3, 4, 5, 6}. Является ли элемент «5» обратимым по сложению? Если да, то какой элемент обратный?

Ответ. Является. Обратный — «2».

7. Множество {0, 1, 2, 3, 4, 5, 6}. Является ли элемент «0» обратимым по умножению? Если да, то какой элемент обратный?

Ответ. Необратим.

Вариант 2

1. Найдите произведение подстановок:
$G1 \cdot G2 = ?$

$$G1=\begin{pmatrix}1&2&3&4&5\\4&2&1&3&5\end{pmatrix};\quad G2=\begin{pmatrix}1&2&3&4&5\\3&5&1&2&4\end{pmatrix}.$$

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

2. Найдите обратную подстановкук данной:

$$G=\begin{pmatrix}1&2&3&4&5\\2&5&4&3&1\end{pmatrix};\quad G^{-1}=?$$

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

3. Определите порядок подстановки:

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

Ответ. 5.

4. Вычислить сумму 1 и 5 по модулю *n*. Множество {1, 2, 3, 4, 5, 6, 7}.

Ответ. 6.

5. Вычислить произведение 2 и 7 по модулю *n*. Множество {1, 2, 3, 4, 5, 6, 7}.

Ответ. 0.

6. Множество {0, 1, 2, 3, 4, 5, 6}. Является ли элемент «2» обратимым по сложению? Если да, то какой элемент обратный?

Ответ. Является. Обратный — «5».

7. Множество {0, 1, 2, 3, 4}. Является ли элемент «3» обратимым по умножению? Если да, то какой элемент обратный?

Ответ. Является. Обратный — «2».

Вариант 3

1. Найдите произведение подстановок:

$G1 \cdot G2 = ?$

$$G1=\begin{pmatrix}1&2&3&4&5\\4&1&2&5&3\end{pmatrix};\quad G2=\begin{pmatrix}1&2&3&4&5\\5&4&1&2&3\end{pmatrix}.$$

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

2. Найдите обратную подстановку к данной:

$$G=\begin{pmatrix}1&2&3&4&5&6\\1&3&4&5&6&2\end{pmatrix};\quad G^{-1}=?$$

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

3. Определите порядок подстановки:

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

Ответ. 6.

4. Вычислить сумму 1 и 2 по модулю *n*. Множество {0, 1, 2, 3, 4, 5, 6}.

Ответ. 3.

5. Вычислить произведение 0 и 5 по модулю *n*. Множество {0, 1, 2, 3, 4, 5, 6}.

Ответ. 0.

6. Множество {1, 2, 3, 4, 5, 6, 7}. Является ли элемент «4» обратимым по сложению? Если да, то какой элемент обратный?

Ответ. Является. Обратный — «3».

7. Множество {0, 1, 2, 3, 4}. Является ли элемент «4» обратимым по умножению? Если да, то какой элемент обратный?

Ответ. Является. Обратный — «4».

Вариант 4

1. Найдите произведение подстановок:

$G1 \cdot G2 = ?$

$$G1=\begin{pmatrix}1&2&3&4&5\\5&4&3&2&1\end{pmatrix};\quad G2=\begin{pmatrix}1&2&3&4&5\\3&5&1&2&4\end{pmatrix}.$$

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

2. Найдите обратную подстановку к данной:

$$G=\begin{pmatrix}1&2&3&4&5&6\\6&2&5&1&3&4\end{pmatrix};\quad G^{-1}=?$$

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

3. Определите порядок подстановки:

$$G=\begin{pmatrix}1&2&3&4&5\\2&5&3&1&4\end{pmatrix}.$$

Ответ. 4.

4. Вычислить сумму 1 и 4 по модулю *n*. Множество {0, 1, 2, 3, 4, 5, 6}.

Ответ. 5.

5. Вычислить произведение 4 и 5 по модулю n. Множество {0, 1, 2, 3, **4**, 5, 6}.

Ответ. 6.

6. Множество {0, 1, 2, 3, 4}. Является ли элемент «2» обратимым по сложению? Если да, то какой элемент обратный?

Ответ. Является. Обратный — «3».

7. Множество {0, 1, 2, 3, 4, 5, 6, 7, 8}. Является ли элемент «8» обратимым по умножению? Если да, то какой элемент обратный?

Ответ. Является. Обратный — «8».

Вариант 5

1. Найдите произведение подстановок:

$G1 \cdot G2 = ?$

$$G1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 2 & 3 & 1 & 6 & 5 \end{pmatrix}; \quad G2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 5 & 3 & 4 & 1 & 2 & 6 \end{pmatrix}.$$

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

2. Найдите обратную подстановку к данной:

$$G = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 1 & 7 & 6 & 3 & 5 & 4 \end{pmatrix}; \quad G^{-1} = ?$$

Ответ. $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 7 & 3 & 4 & 6 & 5 & 1 \end{pmatrix}$.

3. Определите порядок подстановки:

$$G = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 1 & 3 & 2 & 4 \end{pmatrix}.$$

Ответ. 4.

4. Вычислить сумму 1 и 4 по модулю n. Множество {0, 1, 2, 3, 4}.

Ответ. 0.

5. Вычислить произведение 3 и 4 по модулю n. Множество {0, 1, 2, 3, **4**}.

Ответ. 2.

6. Множество {0, 1, 2, 3, 4, 5, 6, 7, 8}. Является ли элемент «7» обратимым по сложению? Если да, то какой элемент обратный?

Ответ. Является. Обратный — «2».

7. Множество {0, 1, 2, 3, 4, 5, 6}. Является ли элемент «2» обратимым по умножению? Если да, то какой элемент обратный?

Ответ. Является. Обратный — «4».

Вариант 6

1. Найдите произведение подстановок:
$G1 \cdot G2 = ?$

$$G1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 3 & 4 & 1 & 2 \end{pmatrix}; \quad G2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 1 & 4 & 5 & 3 \end{pmatrix}.$$

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

2. Найдите обратную подстановку к данной:

$$G = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 \end{pmatrix}; \quad G^{-1} = ?$$

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

3. Определите порядок подстановки:

$$G = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 1 & 7 & 6 & 3 & 5 & 4 \end{pmatrix}.$$

Ответ. 10.

4. Вычислить сумму 1 и 3 по модулю *n*. Множество {0, 1, 2, 3, 4}.

Ответ. 3.

5. Вычислить произведение 3 и 8 по модулю *n*. Множество {0, 1, 2, 3, 4, 5, 6, 7, 8}.

Ответ. 6.

6. Множество {1, 2, 3, 4, 5, 6, 7}. Является ли элемент «1» обратимым по сложению? Если да, то какой элемент обратный?

Ответ. Является. Обратный — «6».

7. Множество {1, 2, 3, 4, 5, 6, 7}. Является ли элемент «7» обратимым по умножению? Если да, то какой элемент обратный?

Ответ. Необратим.

ПРИЛОЖЕНИЕ 2

Контрольные задания к главе 2

Задача 1

Зашифровать текст с помощью шифра простой замены при имеющемся ключе. Пропуски не шифруются.

Текст: «КРИПТОГРАФИЯ ЭТО СПОСОБ ЗАЩИТЫ ИНФОРМАЦИИ».

Ключ:

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

Решение:

С помощью ключа зашифровываем текст. В соответствии с ключом первая буква текста «К» перейдет в «В», «Р» перейдет в «Г» и так далее. В итоге получим **зашифрованный текст:**

«ВГАДЮБКГЖЯАО МЮБ ЕДБЕБЗ ЛЖФАЮТ АЬЯБГЫЖРАА».

Задача 2

Расшифровать текст с помощью шифра простой замены, при имеющемся ключе шифрования.

Текст: «ВГАДЮБКГЖЯАО МЮБ ЕДБЕБЗ ЛЖФАЮТ АЬЯБГЫЖРАА».

Ключ-подстановка:

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

Решение:

Запишем **ключ обратной подстановки** на основе ключа-подстановки:

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

С помощью этого ключа расшифровываем текст. В соответствии с ключом обратной подстановки первая буква зашифрованного текста «В» перейдет в «К», «Г» перейдет в «Р» и так далее. В итоге получим **расшифрованный текст**:

«КРИПТОГРАФИЯ ЭТО СПОСОБ ЗАЩИТЫ ИНФОРМАЦИИ».

Задача 3

Зашифровать текст с помощью шифра перестановки при имеющемся ключе.

Текст: «КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА».

Ключ:

1	2	3	4	5	6
5	3	4	1	6	2

Решение:

Разделим текст в соответствии с длиной ключа и запишем в столбик:

К	Р	И	П	Т	О
Г	Р	А	Ф	И	Ч
Е	С	К	А	Я	
З	А	Щ	И	Т	А

Затем поменяем столбцы местами в соответствии с ключом. Первый столбец станет пятым, второй — третьим и так далее. В итоге получим такую таблицу:

П	О	Р	И	К	Т
Ф	Ч	Р	А	Г	И
А		С	К	Е	Я
И	А	А	Щ	З	Т

Затем выпишем строки по порядку и получим **зашифрованный текст:**

«ПОРИКТФЧРАГИА СКЕЯИААЩЗТ».

Задача 4

Расшифровать текст, зашифрованный шифром перестановки, имея ключ.

Текст: «ПОРИКТФЧРАГИА СКЕЯИААЩЗТ».

Ключ:

1	2	3	4	5	6
5	3	4	1	6	2

Решение:

Составим **ключ обратной подстановки**:

1	2	3	4	5	6
4	6	2	3	1	5

Разобьем текст в соответствии с длиной ключа и запишем в столбик:

П	О	Р	И	К	Т
Ф	Ч	Р	А	Г	И
А		С	К	Е	Я
И	А	А	Щ	З	Т

Переставим столбцы в соответствии с этим ключом. Первый столбец станет четвертым, второй — шестым и так далее. После перестановки получим таблицу:

К	Р	И	П	Т	О
Г	Р	А	Ф	И	Ч
Е	С	К	А	Я	
З	А	Щ	И	Т	А

Затем выпишем строки по порядку и получим **расшифрованный текст:**

«КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА».

Задача 5

Зашифровать текст с помощью шифра случайного гаммирования, считая, что буквы алфавита пронумерованы от 0 до 32 соответственно и зная определенную гамму (ключ):

А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	

Текст: «КРИПТОГРАФИЯ».

Гамма (ключ):

11	1	17	1	14	19	9	14	19	17	15	11

Решение:

X	К	Р	И	П	Т	О	Г	Р	А	Ф	И	Я
	11	17	9	16	19	15	3	17	0	21	9	32
K	11	1	17	1	14	19	9	14	19	17	15	11
Y	22	18	26	17	0	1	12	31	19	5	24	10
	Х	С	Щ	Р	А	Б	Л	Ю	Т	Е	Ч	Й

Первая строка — открытый текст.

Вторая строка — номера соответствующих букв в алфавите.

Третья строка — гамма.

Четвертая строка — номера букв зашифрованного текста.

Пятая строка — зашифрованный текст в соответствии с таблицей подстановки.

Складываем номер буквы и соответствующий компонент ключа, если сумма больше или равна 33, то вычитаем 33. Например, первая буква (11 + 11 = 22), а пятая буква (19 + 14 = 33; 33 – 33 = 0). Затем записываем в **зашифрованный текст:** «ХСЩРАБЛЮТЕЧЙ».

Задача 6

Расшифровать криптограмму, полученную с помощью метода случайного гаммирования, считая, что буквы алфавита пронумерованы от 0 до 32 соответственно, и зная определенную гамму (ключ):

А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	

Текст: «ХСЩРАБЛЮТЕЧЙ».

Гамма (ключ):

11	1	17	1	14	19	9	14	19	17	15	11

Решение:

У	Х	С	Щ	Р	А	Б	Л	Ю	Т	Е	Ч	Й
	22	18	26	17	0	1	12	31	19	5	24	10
К	11	1	17	1	14	19	9	14	19	17	15	11
Х	11	17	9	16	19	15	3	17	0	21	9	32
	К	Р	И	П	Т	О	Г	Р	А	Ф	И	Я

Первая строка — зашифрованный текст.

Вторая строка — номера букв зашифрованного текста.

Третья строка — гамма.

Четвертая строка — номера букв открытого текста в алфавите.

Пятая строка — открытый текст в соответствии с номерами.

Вычитаем из номера буквы алфавита соответствующий компонент ключа, если разность меньше 0, то прибавляем 33. Например, первая буква (22 – 11 = 11), а пятая буква (0 – 14 = –14; –14 + 33 = 19). Затем записываем в **расшифрованный текст** в соответствии с номером в алфавите: «КРИПТОГРАФИЯ».