

**А. П. Алферов, А. Ю. Зубов, А. С. Кузьмин,
А. В. Черемушкин**

ОСНОВЫ КРИПТОГРАФИИ

*2-е издание
исправленное и дополненное*

Допущено Министерством образования
Российской Федерации в качестве учебного пособия
для студентов высших учебных заведений,
обучающихся по группе специальностей
в области информационной безопасности

Москва
“Гелиос АРВ”
2002

ББК 32 81в6
А50

Рецензенты

Марков А И, главный ученый секретарь Академии криптографии РФ, доктор физ -
мат наук, профессор,

Малюк А А, декан факультета информационной безопасности МИФИ (ТУ), кандидат
техн наук, доцент,

Тювин Ю Д, проректор МИРЭА, кандидат техн наук, профессор

**Алферов А. П., Зубов А. Ю., Кузьмин А. С.,
Черемушкин А. В.**

А50 Основы криптографии Учебное пособие, 2-е изд., испр. и
доп. — М.: Гелиос АРВ, 2002. — 480 с., ил.

ISBN 5-85438-025-0

Написано ведущими специалистами в области криптографии, имеющими
многолетний опыт разработки криптографических средств защиты и препода-
вания дисциплин криптографического цикла в ведущих вузах страны

Излагаются основные понятия и разделы, позволяющие получить пред-
ставление о задачах и проблемах современной криптографии В пособие вошли
как традиционные вопросы классификации и оценки надежности шифров, так
и системные вопросы использования криптографических методов защиты ин-
формации

Для студентов, аспирантов, изучающих дисциплины по криптографии и
компьютерной безопасности, преподавателей, а также широкого круга специа-
листов, задачами которых являются квалифицированный выбор и организация
использования криптографических средств защиты информации

ББК 32.81в6

Учебное издание

Алферов Александр Павлович, Зубов Анатолий Юрьевич,
Кузьмин Алексей Сергеевич, Черемушкин Александр Васильевич

ОСНОВЫ КРИПТОГРАФИИ

Заведующая редакцией *Т А Денисова*

Корректор *Л М Лисицына*

Компьютерная верстка *С Н Авилкина*

Лицензия ЛР № 066255 от 29 12 98

Формат 84x108/32 Объем 15 п л Печать офсетная Тираж 3000 экз Заказ № 598

Издательство «Гелиос АРВ»

107140, г Москва Верхняя Красносельская ул., 16 Тел /факс (095) 264-44-39,

e-mail info@gelios-arv.ru, www gelios-arv.ru

© Алферов А. П., Зубов А. Ю.

Кузьмин А. С., Черемушкин А. В. 2002

ISBN 5-85438-025-0

© Обложка Авилкин С. Н., Резник Е. Н. 2002

Вступительное слово

Без использования криптографии сегодня немыслимо решение задач по обеспечению безопасности информации, связанных с конфиденциальностью и целостностью, аутентификацией и невозможностью отказа сторон от авторства. Если до 1990 г. криптография обеспечивала закрытие исключительно государственных линий связи, то в наши дни использование криптографических методов получило широкое распространение благодаря развитию компьютерных сетей и электронного обмена данными в различных областях: финансах, банковском деле, торговле и т. п. Представляется, что значение криптографических методов в указанных областях будет возрастать и далее.

Разработка и использование современных приложений криптографии невозможны без изучения теоретических основ криптографии, которые систематически и доступно излагаются в этой книге.

Данное учебное пособие написано высококвалифицированными специалистами-криптографами, имеющими большой опыт преподавания в Институте криптографии, связи и информатики Академии ФСБ России, который осуществляет подготовку криптографов с 1949 г. Содержание полностью соответствует Государственным образовательным стандартам высшего профессионального образования по специальностям: “Компьютерная безопасность”, “Комплексное обеспечение информационной безопасности автоматизированных систем”, а также “Информационная безопасность телекоммуникационных систем”. Эти специальности в 2000 г. вошли в сформированную новую группу специальностей “Информационная безопасность”. В настоящее время более 80 вузов страны начали или развертывают подготовку соответствующих специа-

листов. Поэтому выход в свет данного учебного пособия весьма актуален.

Предлагаемая книга дополняет серию изданий, начатую в 1998 г. Учебно-методическим объединением вузов по образованию в области информационной безопасности, в которую вошли, в частности, следующие учебные пособия:

Белкин П.Ю., Михальский О.О., Першаков А.С., Правиков Д.И., Проскурин В.Г., Фоменков Г.В., Щербаков А.Ю. Программно-аппаратные средства обеспечения информационной безопасности. Защита программ и данных. — М.: Радио и связь, 2000;

Проскурин В.Г., Крутов С.В., Мацкевич И.В. Программно-аппаратные средства обеспечения информационной безопасности. Защита в операционных системах. — М.: Радио и связь, 2000;

Девянин П.Н., Михальский О.О., Правиков Д.И., Щербаков А.Ю. Теоретические основы компьютерной безопасности. — М.: Радио и связь, 2000.

Книга “Основы криптографии” в отличие от многих книг на русском языке по криптографии, вышедших в 90-е годы, изначально была задумана как учебное пособие. Она характеризуется цельностью и последовательностью изложения, четкостью определения основных понятий, математической строгостью и хорошей методической проработкой. Уверен, что ее издание будет способствовать улучшению подготовки специалистов по защите информации.

Б. А. Погорелов,
председатель Учебно-методического объединения вузов
по образованию в области информационной безопасности,
действительный член Академии криптографии РФ,
доктор физ.-мат. наук, профессор

Введение

Еще 20 лет назад криптография использовалась почти исключительно для обеспечения безопасности военной и дипломатической связи, а также для целей разведывательной и контрразведывательной спецслужб. Вместе с тем начавшееся в 80-е годы бурное развитие информационных технологий и внедрение автоматизированных методов и средств обработки информации практически во все сферы деятельности людей привели к необходимости более широкого использования криптографических средств защиты информации. При этом использование таких средств невозможно без знания основных принципов, лежащих в основе их функционирования и определяющих возможности этих средств по защите информации.

Таким образом, к настоящему времени появилась необходимость ознакомления с основными понятиями криптографии широкого круга специалистов, задачей которых является не собственно разработка средств защиты информации, а скорее квалифицированный выбор этих средств и организация их использования в системах закрытой связи, не принадлежащих силовым структурам или специальным службам. Имеющиеся в настоящее время многочисленные публикации по вопросам криптографии не могут, по нашему мнению, обеспечить в полной мере указанные потребности. Эти публикации адресованы либо специалистам-криптографам, и поэтому оставляют без должного внимания многие практические и системные вопросы, не связанные непосредственно с математическими аспектами указанной науки, либо посвящены, хотя и важным, но отдельным аспектам обеспечения защиты информации криптографическими методами, либо носят научно-популярный характер в ущерб научной глубине излагаемых результатов.

Пособие рассчитано, в основном, на студентов, в чей круг обязанностей после окончания учебы должны входить вопросы выбора и эксплуатации средств криптографической защиты информации, используемых в системах засекреченной связи. При этом опущен материал, связанный с детальным обоснованием методов криптографического анализа, представляющих, на наш взгляд, интерес только для специалистов в области криптоанализа. Приводятся лишь общие описания этих методов, позволяющие оценить их опасность и выработать адекватные меры по их нейтрализации с привлечением, при необходимости, специалистов в области криптографии.

Предлагаемое пособие представляет собой переработку курсов, которые читались авторами в нескольких вариантах в Институте криптографии, связи и информатики Академии ФСБ России и Московском государственном институте радиотехники, электроники и автоматики.

Учитывая возможность использования данного пособия для изучения основ криптографии студентами-нематематиками, авторы сознательно пошли на сохранение общего описательного стиля изложения с четкой формулировкой основных понятий и свойств, дополнив его включением теорем и утверждений, для понимания доказательств которых необходима соответствующая подготовка и которые при первоначальном знакомстве рекомендуется опустить. Изложение материала в необходимых случаях сопровождается примерами. После глав приводятся вопросы для самоконтроля.

В пособии принята следующая организация текста: главы разбиты на параграфы, которые имеют двойную нумерацию; основные понятия выделяются в тексте курсивом; наиболее важные определения имеют свой номер; теоремы, леммы, утверждения и формулы имеют независимую нумерацию внутри каждой главы. Для списка литературы выбрана символьная индексация, составленная из трех букв фамилии первого автора и года издания.

Обозначения

A^*	— множество слов в алфавите A ;
$ A $	— мощность множества A ;
$ a $	— абсолютная величина числа a ;
$\lfloor a \rfloor$	— целая часть числа a ;
$\lceil a \rceil$	— ближайшее целое число, не меньшее a ;
$a \equiv b \pmod{n}$	— сравнимость чисел a и b по модулю n ;
$a = b \pmod{n}$	— остаток от деления числа b на число n ;
$(G, *)$	— множество G с бинарной операцией “*”;
$E_k (D_k)$	— правило зашифрования (расшифрования) на ключе k ;
$E_k(x, y, z)$	— правило зашифрования на ключе k сообщения, образованного приписыванием сообщений x, y и z друг к другу;
$h(x, y, z)$	— значение хэш-функции h для последовательности, образованной приписыванием последовательностей x, y и z друг к другу;
$GF(q)$	— конечное поле из q элементов;
$S(M) (S_n)$	— симметрическая группа подстановок на множестве $M (M = \{1, 2, \dots, n\})$,
$tr_q^{q^m}(a)$	— функция “след” из $GF(q^m)$ в $GF(q)$, $tr_q^{q^m}(a) = a + a^q + a^{q^2} + \dots + a^{q^{m-1}}$;
Z_n	— множество $\{0, 1, \dots, n-1\}$, образующее кольцо относительно сложения и умножения элементов по модулю n ;
Z_n^*	— множество обратимых элементов кольца Z_n .

Глава 1

Исторический очерк развития криптографии

История криптографии насчитывает не одно тысячелетие. Уже в исторических документах древних цивилизаций — Индии, Египте, Китае, Месопотамии — имеются сведения о системах и способах составления шифрованного письма. Видимо, первые системы шифрования появились одновременно с письменностью в четвертом тысячелетии до нашей эры.

В древнеиндийских рукописях приводится более шестидесяти способов письма, среди которых есть и такие, которые можно рассматривать как криптографические. Имеется описание системы замены гласных букв согласными, и наоборот. Один из сохранившихся шифрованных текстов Месопотамии представляет собой табличку, написанную клинописью и содержащую рецепт изготовления глазури для гончарных изделий. В этом тексте использовались редко употребляемые значки, игнорировались некоторые буквы, употреблялись цифры вместо имен. В рукописях Древнего Египта шифровались религиозные тексты и медицинские рецепты. Шифрование использовалось в Библии. Некоторые фрагменты библейских текстов зашифрованы с помощью шифра, который назывался *атбааш*. Правило зашифрования состояло в замене i -й буквы алфавита ($i = \overline{1, n}$) буквой с номером $n - i + 1$, где n — число букв алфавита. Происхождение слова *атбааш* объясняется принципом замены букв. Это слово составлено из букв Алеф, Тае, Бет и Шин, то есть первой и последней, второй и предпоследней букв древнесемитского алфавита.

Развитию криптографии способствовал переход от идеографического письма, основанного на использовании огромно-

го числа иероглифов, к фонетическому письму. В древнем семитском алфавите во втором тысячелетии до нашей эры было уже 30 знаков. Ими обозначались согласные звуки, а также некоторые гласные и слоги. Упрощение письма стимулировало развитие криптографии.

В Древней Греции криптография уже широко использовалась в разных областях деятельности, в особенности в государственной сфере. Плутарх сообщает, что жрецы, например, хранили в форме тайнописи свои прорицания. В Спарте в V — IV вв. до н. э. использовалось одно из первых шифровальных приспособлений — *Сцитала*. Это был жезл цилиндрической формы, на который наматывалась лента пергамента. Кроме жезла могли использоваться рукоятки мечей, кинжалов копий и т.д. Вдоль оси цилиндра на пергамент построчно записывался текст, предназначенный для передачи. После записи текста лента сматывалась с жезла и передавалась адресату, который имел точно такую же Сциталу. Ясно, что такой способ шифрования осуществлял перестановку букв сообщения. Ключом шифра служит диаметр Сциталы. Известен также и метод вскрытия такого шифра, приписываемый Аристотелю. Предлагалось заточить на конус длинный брус и, обернув вокруг него ленту, начать сдвигать ее по конусу от малого диаметра до самого большого. В том месте, где диаметр конуса совпадал с диаметром Сциталы, буквы текста сочетались в слоги и слова. После этого оставалось лишь изготовить цилиндр нужного диаметра.

Другим шифровальным приспособлением времен Спарты была *табличка Энея*. На небольшой табличке горизонтально располагался алфавит, а по ее боковым сторонам имелись выемки для наматывания нити. При зашифровании нить закреплялась у одной из сторон таблички и наматывалась на нее. На нити делались отметки (например, узелки) в местах, которые находились напротив букв данного текста. По алфавиту можно было двигаться лишь в одну сторону, то есть делать по одной отметке на каждом витке. После зашифрования

нить сматывалась и передавалась адресату. Этот шифр представляет собой шифр замены букв открытого текста знаками, которые означали расстояния между отметками на нити. Ключом являлись геометрические размеры таблички и порядок расположения букв алфавита. Это был довольно надежный шифр: история не сохранила документов, подтверждающих сведения о методах его вскрытия.

Греческий писатель Полибий использовал систему сигнализации, которая была широко принята как метод шифрования. Он записывал буквы алфавита в квадратную таблицу и заменял их координатами: парами чисел (i, j) , где i — номер строки, j — номер столбца. Применительно к латинскому алфавиту *квадрат Полибия* имеет следующий вид (см. табл. 1).

Таблица 1. Квадрат Полибия

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

Пары (i, j) передавались с помощью факелов. Например, для передачи буквы O нужно было взять 3 факела в правую руку и 4 факела — в левую.

Подобные шифровальные приспособления с небольшими изменениями просуществовали до эпохи военных походов Юлия Цезаря. Положение меняется в эпоху расцвета Рима, который первоначально представлял собой лишь небольшую гражданскую общину, со временем он разросся, подчинив себе сначала Италию, а затем и все Средиземноморье. Чтобы управлять наместниками в многочисленных провинциях,

шифрованная связь для римских органов власти стала жизненно необходимой. Особую роль в сохранении тайны сыграл способ шифрования, предложенный Юлием Цезарем и изложенный им в “Записках о галльской войне” (I в. до н.э.). Вот что пишет о нем Гай Светоний: “...существуют и его письма к Цицерону и письма к близким о домашних делах: в них, если нужно было сообщить что-нибудь негласно, он пользовался тайнописью, то есть менял буквы так, чтобы из них не складывалось ни одного слова. Чтобы разобрать и прочитать их, нужно читать всякий раз четвертую букву вместо первой, например, *D* вместо *A* и так далее” [Кан67]. Таким образом, Цезарь заменял буквы в соответствии с подстановкой, нижняя строка которой представляет собой алфавит открытого текста, сдвинутый циклически на три буквы влево.

Со времен Цезаря до XV в. шифровальное дело претерпело много изменений, однако нам мало известно о методах и системах шифрования, применяемых в этот период времени. В мрачные годы средневековья практика шифрования сохранялась в строжайшей тайне. Так, в годы крестовых походов шифровальщики, служившие у Папы Римского, после года работы подлежали физическому уничтожению.

В эпоху Возрождения в итальянских городах-государствах параллельно с расцветом культуры и науки активно развивается криптография. Нередко ученые зашифровывали научные гипотезы, чтобы не прослыть еретиками и не подвергнуться преследованиям инквизиции.

Научные методы в криптографии впервые появились, по видимому, в арабских странах. Арабского происхождения и само слово *шифр*. О тайнописи и ее значении говорится даже в сказках “Тысячи и одной ночи”. Первая книга, специально посвященная описанию некоторых шифров, появилась в 855 г., она называлась “Книга о большом стремлении человека разгадать загадки древней письменности”. В 1412 г. издается 14-томная энциклопедия, содержащая систематический обзор всех важнейших областей человеческого знания, —

“Шауба аль-Аща”. Ее автор — Шехаб аль-Кашканди. В этой энциклопедии есть раздел о криптографии под заголовком “Относительно сокрытия в буквах тайных сообщений”, в котором приводятся семь способов шифрования. Там же дается перечень букв в порядке частоты их употребления в арабском языке на основе изучения текста Корана, а также приводятся примеры раскрытия шифров *методом частотного анализа* встречаемости букв.

В XIV в. появилась книга о системах тайнописи, написанная сотрудником тайной канцелярии Папы Римского Чикко Симонетти. В этой книге приводятся шифры замены, в которых гласным буквам соответствуют несколько значковых выражений. Такие шифры позже стали называться шифрами *многозначной замены* или *омофонами*. Они получили развитие в XV в. Так, в книге “Трактат о шифрах” Габриеля де Лавинды — секретаря папы Климентия XII — приводится описание шифра *пропорциональной замены*, в котором каждой букве ставится в соответствие несколько эквивалентов, число которых пропорционально частоте встречаемости буквы в открытом тексте. В 1469 г. был предложен подобный же шифр, получивший название “миланский ключ”. Появление омофонов свидетельствовало о том, что к этому времени уже хорошо осознавали слабости шифров простой замены. Такая модификация шифра разрушала статистику букв открытого сообщения, что явилось заметным шагом в развитии криптографии.

Еще один значительный шаг вперед криптография сделала благодаря труду Леона Альберти. Известный философ, живописец, архитектор, он в 1466 г. написал труд о шифрах. В этой работе был предложен шифр, основанный на использовании *шифровального диска*. Сам Альберти назвал его шифром, “достойным королей”.

Шифровальный диск представлял собой пару соосных дисков разного диаметра (см. рис. 1). Большой из них — неподвижный, его окружность разделена на 24 равных сектора,

в которые вписаны 20 букв латинского алфавита в их естественном порядке и 4 цифры (от 1 до 4). При этом из 24-буквенного алфавита были удалены 4 буквы, без которых можно было обойтись, подобно тому, как в русском языке обходятся без Ъ, Ё, Й. Меньший диск — подвижный, по его окружности, разбитой также на 24 сектора, были вписаны все буквы смешанного латинского алфавита.

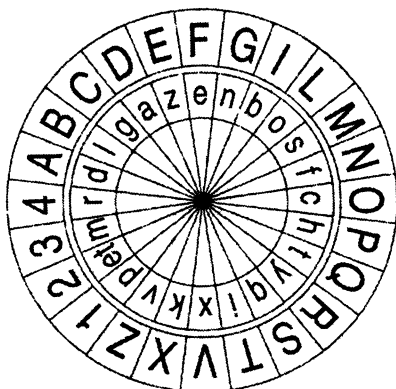


Рис. 1. Диск Альберти

Имея два таких прибора, корреспонденты договаривались о первой индексной букве на подвижном диске. При шифровании сообщения отправитель ставил индексную букву против любой буквы большего диска. Он информировал корреспондента о таком положении диска, записывая эту букву внешнего диска в качестве первой буквы шифртекста. Очередная буква открытого текста отыскивалась на неподвижном диске и стоящая против нее буква меньшего диска являлась результатом ее зашифрования. После того как были зашифрованы несколько букв текста, положение индексной буквы изменялось, о чем также каким-либо образом передавалось корреспонденту.

Такой шифр имел две особенности, которые делают изобретение Альберти событием в истории криптографии. Во-первых, в отличие от шифров простой замены шифровальный диск использовал не один, а несколько алфавитов для зашифрования. Такие шифры получили название *многоалфавитных*. Во-вторых, шифровальный диск позволял использовать так называемые *коды с перешифрованием*, которые получили широкое распространение лишь в конце XIX в., то есть спустя четыре столетия после изобретения Альберти. Для этой цели на внешнем диске имелись цифры. Альберти составил код, состоящий из 336 кодовых групп, занумерованных от 11 до 4444. Каждому кодовому обозначению соответствовала некоторая законченная фраза. Когда такая фраза встречалась в открытом сообщении, она заменялась соответствующим кодовым обозначением, а с помощью диска цифры зашифровывались как обычные знаки открытого текста, превращаясь в буквы.

Богатым на новые идеи в криптографии оказался XVI в. Многоалфавитные шифры получили развитие в вышедшей в 1518 г. первой печатной книге по криптографии под названием “Полиграфия”. Автором книги был один из самых знаменитых ученых того времени аббат Иоганнес Тритемий. В этой книге впервые в криптографии появляется квадратная таблица. Шифралфавиты записаны в строки таблицы один под другим, причем каждый из них сдвинут на одну позицию влево по сравнению с предыдущим (см. табл. 2).

Тритемий предлагал использовать эту таблицу для многоалфавитного зашифрования самым простым из возможных способов: первая буква текста шифруется первым алфавитом, вторая буква — вторым и т. д. В этой таблице не было отдельного алфавита открытого текста, для этой цели служил алфавит первой строки. Таким образом, открытый текст, начинающийся со слов HUNC CAVETO VIRUM ... , приобретал вид HXPF GFBMCZ FUEIB

Преимущество этого метода шифрования по сравнению с методом Альберти состоит в том, что с каждой буквой задействуется новый алфавит. Альберти менял алфавиты лишь после трех или четырех слов. Поэтому его шифртекст состоял из отрезков, каждый из которых обладал закономерностями открытого текста, которые помогали вскрыть криптограмму. Побуквенное зашифрование не дает такого преимущества. *Шифр Тритемия* является также первым нетривиальным примером *периодического шифра*. Так называется многоалфавитный шифр, правило зашифрования которого состоит в использовании периодически повторяющейся последовательности простых замен.

Таблица 2. Таблица Тритемия

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

В 1553 г. Джованни Баттиста Белазо предложил использовать для многоалфавитного шифра буквенный, легко запоминаемый ключ, который он назвал *паролем*. Паролем могло служить слово или фраза. Пароль периодически записывался над открытым текстом. Буква пароля, расположенная над буквой текста, указывала на алфавит таблицы, который использовался для зашифрования этой буквы. Например, это мог быть алфавит из таблицы Тритемия, первой буквой которого являлась буква пароля. Однако Белазо, как и Тритемий, использовал в качестве шифралфавитов обычные алфавиты.

Воскресить смешанные алфавиты, которые применял Альберти, и объединить идеи Альберти с идеями Тритемия и Белазо в современную концепцию многоалфавитной замены выпало на долю итальянца Джованни де ла Порта. Ему было 28 лет, когда он в 1563 г. опубликовал книгу “О тайной переписке”. По сути, эта книга являлась учебником по криптографии, содержащим криптографические познания того времени. Порта предложил использовать квадратную таблицу с периодически сдвигаемым смешанным алфавитом и паролем. Он советовал выбирать длинный ключ. Впервые им был предложен *шифр простой биграммной замены*, в котором пары букв представлялись одним специальным графическим символом. Они заполняли квадратную таблицу размером 20 x 20, строки и столбцы которой занумерованы буквами алфавита

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

Например, биграмма EA заменялась символом \triangle , биграмма LF — символом \ominus и т. д. В своей книге Порта ввел многоалфавитный шифр, определяемый табл. 3.

Таблица 3. Таблица Порты

A	a	b	c	d	e	f	g	h	i	k	l	m
B	n	o	p	q	r	s	t	u	x	y	z	w
C	a	b	c	d	e	f	g	h	i	k	l	m
D	o	p	q	r	s	t	u	x	y	z	w	n
E	a	b	c	d	e	f	g	h	i	k	l	m
F	p	q	r	s	t	u	x	y	z	w	n	o
G	a	b	c	d	e	f	g	h	i	k	l	m
H	q	r	s	t	u	x	y	z	w	n	o	p
I	a	b	c	d	e	f	g	h	i	k	l	m
K	r	s	t	u	x	y	z	w	n	o	p	q
L	a	b	c	d	e	f	g	h	i	k	l	m
M	s	t	u	x	y	z	w	n	o	p	q	r
N	a	b	c	d	e	f	g	h	i	k	l	m
O	t	u	x	y	z	w	n	o	p	q	r	s
P	a	b	c	d	e	f	g	h	i	k	l	m
Q	u	x	y	z	w	n	o	p	q	r	s	t
R	a	b	c	d	e	f	g	h	i	k	l	m
S	x	y	z	w	n	o	p	q	r	s	t	u
T	a	b	c	d	e	f	g	h	i	k	l	m
U	y	z	w	n	o	p	q	r	s	t	u	x
X	a	b	c	d	e	f	g	h	i	k	l	m
Y	z	w	n	o	p	q	r	s	t	u	x	y
Z	a	b	c	d	e	f	g	h	i	k	l	m
W	w	n	o	p	q	r	s	t	u	x	y	z

Шифрование осуществляется при помощи лозунга, который пишется над открытым текстом. Буква лозунга определяет алфавит (заглавные буквы первого столбца), расположенная под ней буква открытого текста ищется в верхнем или нижнем полуалфавите и заменяется соответствующей ей буквой второго полуалфавита. Например, фраза, начинающаяся словами HUNC CAVETO VIRUM..., будет зашифрована при помощи лозунга DE LA PORTA в XFHP YTMOGA FQEAS.

Еще одно важное усовершенствование многоалфавитных систем, состоящее в идее использования в качестве ключа

текста самого сообщения или же зашифрованного текста, принадлежит Джероламо Кардано и Блезу де Виженеру. Такой шифр был назван *самоключом*. В книге Виженера “Трактат о шифрах” самоключ представлен следующим образом. В простейшем случае за основу бралась таблица Гротеция с добавленными к ней в качестве первой строки и первого столбца алфавитами в их естественном порядке. Позже такая таблица стала называться *таблицей Виженера*. Подчеркнем, что в общем случае таблица Виженера состоит из циклически сдвигаемых алфавитов, причем первая строка может быть произвольным смешанным алфавитом (см. табл. 4).

Таблица 4. *Таблица Виженера*

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

Первая строка служит алфавитом открытого текста, а первый столбец — алфавитом ключа. Для зашифрования открытого сообщения ($T_o = t_1 t_2 \dots$) Виженер предлагал в качестве *ключевой последовательности* (Γ) использовать само сообщение (T_o) с добавленной к нему в качестве первой буквы (t_0), известной отправителю и получателю (этим идея Виженера отличалась от идеи Кардано, у которого не было начальной буквы и система которого не обеспечивала однозначности расшифрования). Последовательности букв подписывались друг под другом:

$$\Gamma = t_0 t_1 t_2 \dots t_{i-1} \dots$$

$$\underline{T_o = t_1 t_2 t_3 \dots t_i \dots}$$

$$T_{ш} = s_1 s_2 s_3 \dots s_i \dots$$

При этом пара букв, стоящих друг под другом в Γ и T_o , указывала, соответственно, номера строк и столбцов таблицы, на пересечении которых находится знак s_i зашифрованного текста ($T_{ш}$). Например, фраза HUNC CAVETO VIRUM ..., использованная в предыдущих примерах, и начальная буква P дают шифртекст YCHP ECUWZH IDAMG.

Во втором варианте Виженер предлагал в качестве *ключевой последовательности* использовать зашифрованный текст:

$$\Gamma = s_0 s_1 s_2 \dots s_{i-1} \dots$$

$$\underline{T_o = t_1 t_2 t_3 \dots t_i \dots}$$

$$T_{ш} = s_1 s_2 s_3 \dots s_i \dots$$

Самоключ Виженера был незаслуженно забыт на долгое время, а под *шифром Виженера* до сих пор понимают самый простой вариант с коротким ключевым словом и с таблицей, состоящей из обычных алфавитов.

Кардано принадлежит также идея *поворотной решетки* как средства шифрования. Изначально обычная решетка представляла собой лист из твердого материала, в котором через неправильные интервалы сделаны прямоугольные вырезы высотой для одной строчки и различной длины. Накладывая эту решетку на лист писчей бумаги, можно было записывать в вырезы секретное сообщение. После этого, сняв решетку, нужно было заполнить оставшиеся свободные места на листе бумаги неким текстом, маскирующим секретное сообщение. Подобным стеганографическим методом маскировки сообщения пользовались многие известные исторические лица, например кардинал Ришелье во Франции и русский дипломат и писатель А. Грибоедов. Так, Ришелье использовал прямоугольник размера 7×10 . Для длинных сообщений прямоугольник использовался несколько раз. Прорезы трафарета размещались в позициях:

(1,8), (2,9), (3,6), (4,5), (4,6), (5,1), (5,6),
(5,7), (5,9), (6,2), (6,10), (7,9), (7,10).

	1	2	3	4	5	6	7	8	9	10
1								■		
2									■	
3						■	■			
4					■	■				
5	■					■			■	
6		■								■
7									■	■

Рис. 2. Пример решетки (заимствован из [Сал96])

Следующий текст выглядит как невинное любовное письмо (см. рис. 3).

	1	2	3	4	5	6	7	8	9	10
1	I		L	O	V	E		Y	O	U
2	I		H	A	V	E		Y	O	U
3	D	E	E	P		U	N	D	E	R
4	M	Y		S	K		N		M	Y
5	L	O	V	E		L	A	S	T	S
6	F	O	R	E	V	E	R		I	N
7	H	Y	P	E	R	S	P	A	C	E

Рис. 3

Однако используя трафарет Ришелье, получим зловещую команду:

YOU KILL AT ONES

Кардано использовал квадратную решетку, которая своими вырезами однократно покрывает всю площадь квадрата при ее самосовмещении. На основе такой решетки он построил шифр перестановки.

Нельзя не упомянуть в историческом обзоре имени Матео Ардженти, работавшего в области криптографии в начале XVII в. Он составил руководство по криптографии на 135 листах, изданное в переплете из телячьей кожи. В этой книге впервые предложено использовать некоторое слово в качестве мнемонического ключа для смешанного алфавита. Началом смешанного алфавита служило ключевое слово (как правило, без повторяющихся букв), за которым следовали остальные буквы в их естественном порядке. Например, ключевое слово PIETRO дает смешанный латинский алфавит

PIETROABCDEFGHIJKLMNQSUZ

Такие смешанные алфавиты часто использовались в качестве алфавитов шифртекста в шифрах простой замены.

С целью усложнения шифра простой замены Ардженти вводил *пустышки*, которые добавлялись в зашифрованное сообщение, использовал шифробозначения разной значности, для некоторых частых сочетаний букв текста вводил отдельные обозначения, придавал частым буквам несколько обозначений. Позже подобные идеи получили широкое распространение. Приведем пример *шифра Ардженти* (см. табл. 5).

Таблица 5. Шифр Ардженти

A	B	C	D	E	F	G	H	I	L	M	N	O
1	86	02	20	62	22	06	60	3	24	26	84	9
				82								

P	Q	R	S	T	U	Z	ET	CON	NON	CHE	∅
66	68	28	42	80	04	88	08	64	00	44	5
					40						7

Слово ARGENTI может быть зашифровано многими способами, например так:

5128068285480377

или же так:

172850675628455803

Наибольшим достижением Ардженти считается разработанный им *буквенный код* — один из шифров замены, в котором буквы, слоги, слова и целые фразы заменялись группами букв. Необходимым количеством словарных величин в коде в то время считалось 1200.

В истории криптографии XVII — XVIII в. называют эрой “*черных кабинетов*”. В этот период во многих государствах Европы, в первую очередь во Франции, получили развитие дешифровальные подразделения, названные “*черными кабинетами*”. Первый из них образован по инициативе кардинала Ришелье при дворе короля Людовика XIII. Его возглавил первый профессиональный криптограф Франции Антуан Рос-22

синьоль. Следует отметить, что некоторые оригинальные идеи, возникшие в криптографии в этот период, связаны с именем самого Ришелье, который использовал, например, для секретной переписки с королем оригинальный шифр перестановки с переменным ключом. Его использование становится понятным из следующего примера:

Шифр Ришелье

Ключ: 2741635; 15243; 671852493; 07; 28615; 943; ...

Открытый текст:

LETTER SENT TO THE EMPEROR GIVING FULL DETAIL

Ключ:

(2741635) (15243) (671852493) (07) (28615) (943) (27 41635)

Шифртекст:

TLRTSEE ETOTN EPOEMTHER NI LUGIG VFR TLIE SAD

Известно, что Ришелье пользовался также кодами. Попутно отметим, что свой несложный код был и у знаменитого Наполеона:

Малый шифр Наполеона (Petit Chiffre)

(реконструирован Базери)

A – 15, ar – 25, al – 39

B – 37, bu – 3, bo – 35, bi – 29

C – 6, ca – 32, ce – 20

D – 23, de – 52

E – 53, es – 82, et – 50, en – 68

F – 55, fa – 69, fe – 58, fo – 71

G – 81, ga – 51

H – 85, hi – 77

I – 119, jai – 122

J – 87, jai – 123

K – ?

L – 96, lu – 103, le – 117, la – 106

M – 114, ma – 107

N – 115, ne – 94, ni – 116

O – 90, ot – 153
P – 137, po – 152
Q – 173, que – 136
R – 169, ra – 146, re – 126, ri – 148,
S – 167, sa – 171, se – 177, si – 134, so – 168, su – 174
T – 176, ti – 145, to – 157
U – 138
V – 164, ve – 132, vi – 161, vo – 175
W, X, Y – ?
Z – 166

В то время в Европе получили широкое распространение шифры, называемые *номенклаторами*, объединявшие в себе простую замену и код. В простейших номенклаторах код состоял из нескольких десятков слов или фраз с двухбуквенными кодовыми обозначениями. Со временем списки заменяемых слов в номенклаторах увеличились до двух или трех тысяч эквивалентов слогов и слов. В царской России XVIII в. закодированное открытое сообщение шифровалось далее простой заменой.

Кстати, несколько слов о русской криптографии. Уже с XIV в. в Новгороде существовала техника тайного письма. Использовались в основном шифры простой замены. Благодаря торговым связям Новгорода с Германией в России становятся известными многие западные разработки, в том числе новые системы шифрования. Учреждение постоянной почтовой связи России с Европой дало возможность развитию шифрованной переписки. Благодаря привлечению Петром I для разработки проектов развития образования и государственного устройства России знаменитого Готфрида Вильгельма Лейбница, который известен и как криптограф, в Петербурге появилась цифирная палата, задачами которой было развитие и использование систем шифрования.

Когда Россиньоль начинал свою карьеру, в номенклаторах как элементы открытого текста, так и элементы кода рас-

полагались в алфавитном порядке (или в алфавитном и числовом порядке, если код был цифровой). Россиньоль заметил, что такой “параллелизм” открытого текста и кода облегчал восстановление открытого текста. Если, например, он устанавливал, что в английской депеше 137 заменяет FOR, а 168 — IN, то он уже знал, что 21 не может заменять TO, так как цифровые кодовые обозначения для слов, начинающихся с T, должны быть больше, нежели для слов, начинающихся с I. Обнаружив такую слабость, Россиньоль перемешивал кодовые элементы по отношению к открытому тексту. На одном листе он располагал элементы открытого текста в алфавитном порядке, а кодовые элементы — вразброс, на другом листе для облегчения расшифровки кодовые элементы стояли в алфавитном порядке, тогда как их открытые эквиваленты были разбросаны. Это явилось значительным усовершенствованием подобных шифрсистем. Однако составление неалфавитных номенклаторов обходилось очень дорого, и, таким образом, по соображениям экономии и в ущерб надежности многие номенклаторы регрессировали к упрощенному алфавитному типу.

В Англии тоже был свой “черный кабинет”. В его работе в XVII в. заметное место занимал Джон Валлис, известный как крупнейший английский математик до Исаака Ньютона. Работы по вскрытию шифров для парламента привели к назначению Валлиса в 1649 г. в Оксфорд профессором геометрии в возрасте 32 лет. В своем труде “Арифметика бесконечного” он сделал выводы, которые послужили Ньютону стартовой площадкой для разработки интегрального исчисления. Валлис ввел знак ∞ для бесконечности и первый путем интерполяции вычислил число π . Кстати, само это обозначение также принадлежит ему.

В Германии начальником первого дешифровального отделения был граф Гронсфельд, создавший один из вариантов усовершенствования шифра Виженера. Он взял числовой, легко запоминаемый лозунг. Вместо таблицы Виженера ис-

пользовался один несмешанный алфавит. При шифровании знаки открытого текста выписывались под цифрами лозунга. Очередная буква открытого текста заменялась буквой алфавита, отстоящей от нее вправо на количество букв, равное соответствующей цифре лозунга.

Шифр Гронсфельда

Открытый текст: GERMANY

Лозунг: 1 3 5 7 9

Алфавит: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Для удобства выпишем алфавит с порядковыми номерами букв:

A	B	C	D	E	F	G	H	I	J	K	L	M
1	2	3	4	5	6	7	8	9	10	11	12	13

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
14	15	16	17	18	19	20	21	22	23	24	25	26

и лозунг над текстом:

1 3 5 7 9 1 3
G E R M A N Y

Теперь легко получить зашифрованный текст:

H N W T J O B

Любопытен опыт использования криптографии при составлении *астрономических анаграмм*. Одно из таких применений связано с открытием колец Сатурна [Пер66]:

В годы жизни Галилео Галилея существовал обычай закреплять за собой право на первенство в каком-либо открытии своеобразным способом. Напав на открытие, которое нуждается в дальнейшем подтверждении, ученый из опасения, чтобы его не опередили другие, прибегал к помощи *анаграммы* (перестановке букв); он кратко объявлял о сущности своего

открытия в форме анаграммы, истинный смысл которой был известен лишь ему одному. Это давало ученому возможность не спеша проверить свое открытие, а в случае появления другого претендента — доказать свое первенство. Когда же он окончательно убеждался в правильности первоначальной догадки, он раскрывал секрет анаграммы. Заметив в свою несовершенную подозрительную трубу, что Сатурн имеет по бокам какие-то придатки, Галилей поспешил сделать заявку на это открытие и опубликовал следующий набор букв:

SMAISMRMIELMEPOETALEUMIBUVNEUGTTAVIRAS

Задача восстановления открытого текста (без какой-либо дополнительной информации об использованном преобразовании) требует перебора

39!

3!5!4!4!2!2!5!3!3!2!2!

возможных перестановок букв криптограммы (это — число перестановок с повторениями). Приведенное число имеет в своей записи примерно 35 цифр.

Современник итальянского ученого Иоганн Кеплер с присущим ему беспримерным терпением затратил немало труда на то, чтобы проникнуть в сокровенный смысл заявки Галилея, и ему казалось, что он добился этого, когда из опубликованных букв (опустив две из них) составил такую латинскую фразу:

SALVE, UMBISTINEUM GEMINAUM MARTIA PROLES

(Привет вам, близнецы, Марса порождение)

Кеплер был убежден, что Галилей открыл те два спутника Марса, существование которых подозревал он сам (они в действительности и были открыты, но спустя два с половиной века). Однако остроумие Кеплера на этот раз не привело к

цели. Когда Галилей раскрыл, наконец, секрет своей заявки, оказалось, что фраза (если двумя буквами пренебречь) такова:

ALTISSIMUM PLANETAM TERGEMINUM OBSERVAVI
(Высочайшую планету тройною наблюдал)

Из-за слабости своей трубы Галилей не мог понять истинного значения этого “тройного” образа Сатурна, а когда спустя несколько лет боковые придатки планеты совершенно “исчезли”, Галилей решил, что ошибся, и никаких придатков у Сатурна нет. Открыть кольца Сатурна удалось только через полвека Гюйгенсу. Подобно Галилею, он не сразу опубликовал свое открытие, а скрыл догадку под тайнописью:

AAAAAAACCCCCDEEEEGHIIIIILLLLMMNNNNNNNNNN
OOOOPPQRRSTTTTTUUUUU

Спустя три года, убедившись в правильности своей догадки, Гюйгенс обнаружил смысл заявки:

Annulo cingitur, tenui, plano, nusquam cohaerente,
ad eclipticam inclinato

(Кольцом окружен тонким, плоским, нигде
не прикасающимся, к эклиптике наклоненном)

В целом можно сказать, что XVII и XVIII вв не дали новых идей в криптографии. Эра “черных кабинетов” закончилась в 40-х годах XIX в. в период революционного подъема.

Много новых идей в криптографии принес XIX в. Изобретение в середине XIX в. телеграфа и других технических видов связи дало новый толчок развитию криптографии. Информация передавалась в виде токовых и бестоковых посылок, то есть представлялась в двоичном виде. Поэтому возникла проблема “рационального” представления информа-

ции, которая решалась с помощью кодов. Коды позволяли передать длинное слово или целую фразу двумя-тремя знаками. Появилась потребность в высокоскоростных способах шифрования и в корректирующих кодах, необходимых в связи с неизбежными ошибками при передаче сообщений.

Однако еще до изобретения телеграфа появился ряд интересных шифровальных устройств. Приблизительно в 1800 г. была создана одна шифровальная система, занимающая особое место в истории криптографии. Речь идет о “дисковом шифре” Т. Джефферсона — первого государственного секретаря США, ставшего позже третьим президентом.

Дисковый шифратор Т. Джефферсона состоял из 25 – 36 деревянных дисков одинакового размера, насаженных на общую ось (см. рис. 4)

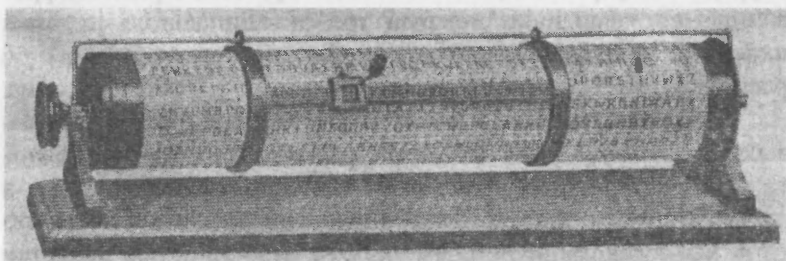


Рис. 4. Дисковый шифратор Т. Джефферсона

На одном конце оси имелась неподвижная головка, на другом — резьба и гайка, с помощью которой все диски фиксировались в любом нужном угловом положении. Имелась также прямолинейная рейка, способная вращаться на оси и позволяющая выделить строку букв на дисках, параллельную оси. На боковой поверхности каждого диска, разделенной на 26 равных частей, наносились буквы смешанных английских алфавитов. Для зашифрования части сообщения (длина которой равнялась числу дисков на оси) под рейку, находящуюся в фиксированном угловом положении, подводилась первая бук-

ва сообщения, найденная на первом диске, затем — вторая буква сообщения, найденная на втором диске, и т. д., так, чтобы все подобранные буквы оказались в одной строке. Положение дисков фиксировалось гайкой, после чего рейка подводилась под любую другую строку цилиндра, буквы которой составляли зашифрованный текст. При расшифровке буквы зашифрованного текста, набранные на последовательных дисках, подводились аналогичным образом под рейку, положение дисков фиксировалось гайкой, после чего с помощью рейки просматривались образовавшиеся строки цилиндра, среди которых несложно было найти открытое сообщение.

Кажущаяся некорректность, связанная с возможностью неоднозначности расшифровки, устраняется достаточно большим числом используемых дисков. Это замечание относится, конечно, лишь к осмысленным текстам. При зашифровке неосмысленных текстов требовалась дополнительная информация о величине сдвига рейки, без чего однозначное расшифрование невозможно.

Такая шифрсистема имеет огромное количество ключевых элементов. К ним относятся: расположение букв алфавита на дисках, расстановка дисков на оси, выбор набора дисков из имеющегося запаса. Дисковый шифр можно отнести по типу к многоалфавитной замене. Его особенностью является блочный характер зашифрования, при котором каждый участок текста (блок) шифруется независимо от других. Позже такие шифры стали называться *блочными шифрами*.

Вместо того чтобы (пользуясь служебным положением) внедрить свое замечательное изобретение в практику, Джефферсон, по-видимому, отложил его в архив и предал забвению. Шифр был обнаружен в его бумагах в библиотеке конгресса лишь в 1922 г., по иронии судьбы именно в том году, когда в армии США начали применять почти аналогичную систему, изобретенную независимо от Джефферсона.

В 1817 г. другой американец Десиус Уодсворт сконструировал шифровальное устройство, которое также внесло но-

вый принцип в криптографию. Его нововведение состояло в том, что он сделал алфавиты открытого и зашифрованного текстов различных длин. Устройство, с помощью которого он это осуществил, представляло собой диск, на котором были расположены два подвижных кольца с алфавитами. Внешний алфавит состоял из 26 букв и 7 цифр (от 2 до 8). Внутренний алфавит состоял лишь из 26 букв. Диск имел подобие неподвижной часовой стрелки, в двух прорезях которой появлялись расположенные друг под другом буквы алфавитов. На внутреннем кольце указывалась буква открытого текста, на внешнем кольце — соответствующая буква шифртекста. Оба кольца могли вращаться и были связаны друг с другом с помощью двух шестерен, одна из которых имела 33 зубца, а другая — 26. Буквы и цифры внешнего кольца были съемными и могли быть собраны в любом порядке. Перед зашифрованием корреспонденты договаривались относительно взаимного начального положения обоих колец. Для установки дисков в такое положение шестерни можно было разъединить. Проследим на примере слова “введение” процесс зашифрования.

Сначала внутреннее кольцо поворачивалось до тех пор, пока в прорези стрелки не показывалась буква “в”. Стоящая в другой прорези буква внешнего кольца записывалась в качестве первой буквы шифртекста. Затем внутреннее кольцо вращалось до тех пор, пока буква “в” вновь не показывалась в прорези. Это вращение посредством шестерен передавалось на внешнее кольцо, но из-за различия в числе букв алфавитов оно совершало лишь $\frac{26}{33}$ полного оборота, в то время как внутреннее кольцо совершало полный оборот. Значит, второй знак шифртекста располагался во внешнем алфавите на расстоянии семи мест вперед от первого знака, несмотря на то, что оба знака представляли одну и ту же букву открытого текста. Если этот процесс зашифрования осуществлять дальше, то эквиваленты шифртекста для буквы “в” начнут повторяться лишь после того, как будут использованы все 33 буквы

и цифры внешнего алфавита. Это объясняется тем, что числа 26 и 33 не имеют общих делителей, благодаря которым такое повторение могло бы произойти раньше. Следующие буквы открытого текста зашифровались аналогично.

Такая шифрсистема реализует периодическую многоалфавитную замену. Различие чисел букв алфавитов открытого и зашифрованного текстов приводит к существенным отличиям этой системы от предыдущих многоалфавитных систем. Так, в устройстве Уодсворда используется 33 шифралфавита, а не 24 или 26, как в системах Тритемия или Виженера. Важнее то, что эти алфавиты используются не непосредственно один за другим, а в произвольном порядке, который зависит от букв открытого текста. Этот произвольный порядок служит гораздо более надежной защитой шифра, чем правильная последовательность использования алфавитов, как в системе Тритемия.

Идея Уодсворда была незаслуженно забыта. Славу открытия приписывают английскому ученому Чарлзу Уитстону, который значительно позже и независимо изобрел свое устройство на том же принципе (см. рис. 5). Основное отличие заключалось в том, что в устройстве Уитстона алфавиты были неподвижными, но зато имелась пара подвижных стрелок, соединенных шестеренками.

Уитстон более известен как ученый, предложивший идею электрического телеграфа, изобретатель концертино, автор первых стереоскопических рисунков. Он высказал гипотезу о создании говорящих машин, разработал метод точного измерения электрического сопротивления, который называется “мостик Уитстона”.

Впервые свое устройство Уитстон продемонстрировал на Всемирной выставке в Париже в 1876 г. На внешнем кольце находился алфавит открытого текста, состоящий из 27 элементов: 26 букв, расположенных в обычном порядке, и знака пробела между словами. Внутренний алфавит состоял из 26 букв, расположенных в произвольном порядке.

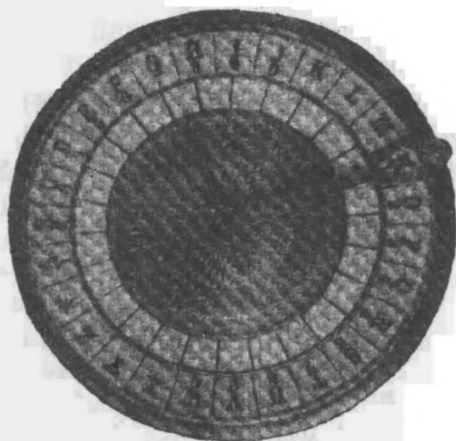


Рис. 5

Уитстон изобрел шифр, который позже стали называть *шифром Плейфера*. Дело в том, что Лион Плейфер, заместитель председателя Палаты общин, министр почт, председатель Британской ассоциации развития науки, был другом Уитстона, был похож на него, так что их часто путали. В 1854 г. Плейфер продемонстрировал систему шифрования, которую он назвал “недавно открытый симметричный шифр Уитстона”. Это был первый из известных биграммных буквенных шифров (напомним, что биграммный шифр Порта был значковым). То обстоятельство, что Плейфер популяризировал изобретение Уитстона, сохранило его имя в названии шифра. Этот шифр использовался англичанами в период первой мировой войны. Описание и обсуждение этого шифра будет приведено в гл. 5.

Во второй половине XIX в. появился весьма устойчивый способ усложнения числовых кодов — *гаммирование*. Он заключался в перешифровании закодированного сообщения с

помощью некоторого ключевого числа, которое и называлось *гаммой*. Шифрование с помощью гаммы состояло в сложении всех кодированных групп сообщения с одним и тем же ключевым числом. Эту операцию стали называть “*наложением гаммы*”. Например, результатом наложения гаммы 6413 на кодированный текст 3425 7102 8139 являлась числовая последовательность 9838 3515 4552:

$$\begin{array}{r}
 3425 \quad 7102 \quad 8139 \\
 + \quad \underline{6413 \quad 6413 \quad 6413} \\
 9838 \quad 3515 \quad 4552
 \end{array}$$

Единицы переноса, появляющиеся при сложении между кодовыми группами, опускались. “*Снятие гаммы*” являлось обратной операцией:

$$\begin{array}{r}
 9838 \quad 3515 \quad 4552 \\
 - \quad \underline{6413 \quad 6413 \quad 6413} \\
 3425 \quad 7102 \quad 8139
 \end{array}$$

В 1888 г. француз маркиз де Виари в одной из своих научных статей, посвященных криптографии, обозначил греческой буквой X любую букву шифрованного текста, греческой буквой Γ любую букву гаммы и строчной буквой c любую букву открытого текста. Он, по сути, *доказал*, что алгебраическая формула

$$X = (c + \Gamma) \bmod 26 \quad (1)$$

воспроизводит зашифрование по Виженеру при замене букв алфавита числами согласно следующей таблице:

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Тем самым была заложена алгебраическая основа для исследования шифров замены типа шифра Виженера. Используя уравнение шифрования, можно было отказаться от громоздкой таблицы Виженера.

Позже лозунговая гамма стала произвольной последовательностью, а шифр с уравнением шифрования (1) стал называться *шифром гаммирования*.

Еще одним известным криптографом того времени был голландец Керкгоффс. Его полным именем было Жан-Вильгельм-Губерт-Виктор-Франсуа-Александр-Огюст Керкгоффс ван Ньювенгоф. Разносторонний ученый, преподававший 6 иностранных языков, историю и математику, он в возрасте 47 лет написал книгу “Военная криптография”. В ней сформулированы 6 конкретных требований к шифрам, два из которых относятся к стойкости шифрования, а остальные — к эксплуатационным качествам. Одно из них (“компрометация системы не должна причинять неудобств корреспондентам”) стало называться “*правилом Керкгоффса*”. Суть этого правила состоит в том, что *стойкость* (или надежность) шифра определяется лишь секретностью ключа. Другими словами, оценка качества шифра (на основе некоторого зашифрованного текста) должна проводиться при условии, что о данном шифре известно все, кроме использованного ключа.

XX в. “прославился” двумя мировыми войнами. Эти войны оставили свой отпечаток на всех процессах, происходивших в человеческом обществе. Они не могли не сказаться и на развитии криптографии.

В период первой мировой войны в качестве полевых шифров широко использовались ручные шифры, в первую очередь шифры перестановки с различными усложнениями. Это были *вертикальные перестановки* (см. гл. 4), усложнен-

ные перекодировкой исходного алфавита, а также *двойные вертикальные перестановки*.

Первая мировая война явилась поворотным пунктом в истории криптографии: если до войны криптография представляла собой достаточно узкую область, то после войны она стала широким полем деятельности. Причина этого состояла в необычайном росте объема шифрпереписки, передаваемой по различным каналам связи. Криптоанализ стал важнейшим элементом разведки.

Прогресс этой области криптографии характеризовался и изменениями в самом криптоанализе. Эта наука переросла методы индивидуальной работы криптоаналитика над криптограммой. Системы секретной связи перестали быть настолько малочисленными и однородными, что один специалист мог овладеть всеми специализациями. Характер используемых шифров потребовал для их вскрытия скрупулезного анализа переписки, поиска ситуаций, благоприятствующих успешному криптоанализу, знания соответствующей обстановки. Кроме того, криптоанализ обогатился большим опытом использования в годы войны ошибок неопытных или ленивых шифровальщиков. Еще Ф. Бэкон писал, что “в результате неловкости и неискусности тех рук, через которые проходят величайшие секреты, эти секреты во многих случаях оказывались обеспеченными слабейшими шифрами” [Кан67]. Этот печальный опыт привел к необходимости введения строгой дисциплины среди шифровальщиков.

Несмотря на указанные последствия, первая мировая война не породила никаких новых научных идей в криптографии. Наоборот, полностью исчерпали свои возможности ручное шифрование, с одной стороны, и техническая сторона криптоанализа, состоявшая в подсчете частот встречаемости знаков, с другой.

В тот период проявились таланты целого ряда ставших впоследствии известными криптографов. В их числе был Г. О. Ярдли, который вскоре после вступления США в войну

в 1917 г. убедил военное министерство в необходимости создания криптографической службы. В 27 лет он был назначен начальником криптографического отдела (MI-8) разведки военного министерства. При отделе было создано учебное отделение по подготовке криптоаналитиков для американской армии. Отдел MI-8 добился больших успехов в дешифровании дипломатической переписки многих развитых стран. В 1919 г. отдел был преобразован в “черный кабинет” с совместным финансированием от военного министерства и госдепартамента в объеме 100 тыс. долларов в год. Одной из главных задач “черного кабинета” было раскрытие японских кодов, некоторые из которых содержали до 25 тысяч кодовых величин. В период с 1917 по 1929 г. специалистам “черного кабинета” удалось дешифровать более 45 тысяч криптограмм различных стран, в том числе и Японии.

Ярдли, желая упрочить успехи, подготовил докладную записку Президенту США о мерах по укреплению своей службы. Однако ставший в то время Государственным секретарем Г. Стимсон был шокирован, узнав о существовании “черного кабинета”, и полностью осудил его деятельность. Ему принадлежит знаменитая фраза: “Джентльмены не читают писем друг друга”. Финансирование “черного кабинета” было прекращено, и Ярдли лишился работы. Он написал книгу “Американский черный кабинет” [Yar31], в которой рассказал о многих успехах по дешифрованию. Книга была издана большими тиражами в ряде стран и произвела эффект разорвавшейся бомбы. Позже он написал книгу “Японские дипломатические секреты”, в которой приводились многие японские телеграммы. Рукопись этой книги была конфискована по решению суда. Последние годы жизни Ярдли не занимался криптографией. Он умер в 1958 г. и был похоронен с воинскими почестями на Арлингтонском национальном кладбище. В некрологе он был назван “отцом американской криптографии”.

Значительный успех в криптографии связан с еще одним американцем — Г. Вернамом. В 1917 г. он, будучи сотрудником телеграфной компании, предложил идею автоматического шифрования телеграфных сообщений. Речь шла о своеобразном наложении гаммы на знаки алфавита, представленные в соответствии с телетайпным кодом Бодо пятизначными “импульсными комбинациями”. Например, буква *a* представлялась комбинацией (+ + — — —), а комбинация (+ + — + +) представляла символ перехода от букв к цифрам. На бумажной ленте, используемой при работе телетайпа, знаку “+” отвечало наличие отверстия, а знаку “—” — его отсутствие. При считывании с ленты металлические щупы проходили через отверстия, замыкали электрическую цепь и тем самым посылали в линию импульс тока.

Вернам предложил электромеханически покоординатно складывать “импульсы” знаков открытого текста с “импульсами” гаммы, предварительно нанесенными на ленту. Сложение проводилось “по модулю 2”. Имеется в виду, что если “+” отождествить с 1, а “—” с 0, то сложение определяется двоичной арифметикой:

+	0	1
0	0	1
1	1	0

Например, наложение на знак открытого текста (11001) знака гаммы (01111) давало знак шифртекста (10110). При расшифровании нужно было произвести ту же операцию со знаком шифртекста: $(10110) \oplus (01111) = (11001)$.

Вернам сконструировал и устройство для такого сложения. Замечательно то, что процесс шифрования оказывался полностью автоматизированным, в предложенной схеме исключался шифровальщик. Кроме того, оказывались слитыми воедино процессы шифрования-расшифрования и передачи по каналу связи. Тем самым наряду с традиционной схемой

предварительного шифрования, когда по каналу передается предварительно зашифрованное сообщение, положено начало *линейному шифрованию*.

В 1918 г. два комплекта соответствующей аппаратуры были изготовлены и испытаны. Испытания дали положительные результаты. Единственное неудовлетворение специалистов-криптографов было связано с гаммой. Дело в том, что первоначально гамма была нанесена на ленту, склеенную в кольцо. Несмотря на то, что знаки гаммы на ленте выбирались случайно, при зашифровании длинных сообщений гамма регулярно повторялась. Этот недостаток так же отчетливо осознавался, как и для шифра Виженера. Уже тогда хорошо понимали, что повторное использование гаммы недопустимо даже в пределах одного сообщения. Хотя сам Вернам не был математиком, он, может и неосознанно, предлагал однократное использование гаммы. Попытки удлинить гамму приводили к неудобствам в работе с длинным кольцом. Тогда был предложен вариант с двумя лентами, одна из которых шифровала другую, в результате чего получалась гамма, имеющая длину периода, равную произведению длин исходных периодов.

Несмотря на то, что *шифр Вернама* обладал целым рядом достоинств, он не получил широкого распространения. Трудности, связанные с изготовлением, рассылкой и учетом использованной гаммы, особенно в условиях военной связи, при передаче больших объемов сообщений, стали непреодолимыми. Вспомнили о шифре Вернама лишь накануне второй мировой войны.

Почти половина XX в. была связана с использованием *колесных шифраторов*. Различные их конструкции были запатентованы примерно в одно и то же время (в период 1917 — 1919 гг.) в разных странах: американцем Э. Х. Хеберном, голландцем Х. Ф. Кохом, немцем А. Шербиусом и шведом А. Г. Даммом.

Чертежи своей схемы на основе *шифрующего диска* Хеберн представил в 1917 г., и уже в следующем году был по-

строен первый дисковый аппарат, получивший одобрение ВМС США. В 1921 г. Хеберн основал первую в США компанию по производству шифрмашин, которую через десять лет ждал бесславный конец, связанный с финансовыми трудностями.

Что представлял собой шифрующий диск? Корпус диска (имевшего размеры хоккейной шайбы) состоял из изоляционного материала, например твердой резины. По окружностям каждой из его сторон были вмонтированы на равном расстоянии друг от друга 26 электрических контактов (см. рис. 6). Каждый контакт был соединен внутри корпуса с некоторым контактом на другой стороне. Контакты на *входной стороне* представляли буквы открытого текста, контакты на *выходной стороне* — буквы шифртекста.

Диск устанавливался на оси между двумя неподвижными пластинами (*розетками*), каждая из которых также была изготовлена из изолятора и имела 26 контактов, соответствующих расположению контактов на диске. Контакты входной розетки соединялись с клавиатурой пишущей машинки, печатающей буквы открытого текста. Контакты выходной розетки соединялись с выходным устройством, указывающим буквы шифртекста, например, с помощью лампочек. При фиксированном угловом положении диска электрические цепи, соединяющие входные и выходные контакты, реализовывали одноалфавитную замену. При повороте же диска (на углы $2\pi k/26$) схема реализовывала многоалфавитную замену (с 26 простыми заменами).

Рядом с одним диском можно было установить и другие диски. Тем самым схема токопрохождения удлинялась и число возможных простых замен, реализуемых многодисковой схемой значительно возрастало. При движении k дисков по простейшей схеме одометра получался период, равный 26^k , который можно было сделать астрономическим числом.

Подобные шифрмашины обслуживали значительную часть линий связи высшего командования ВМС США, начиная с 20-х годов.

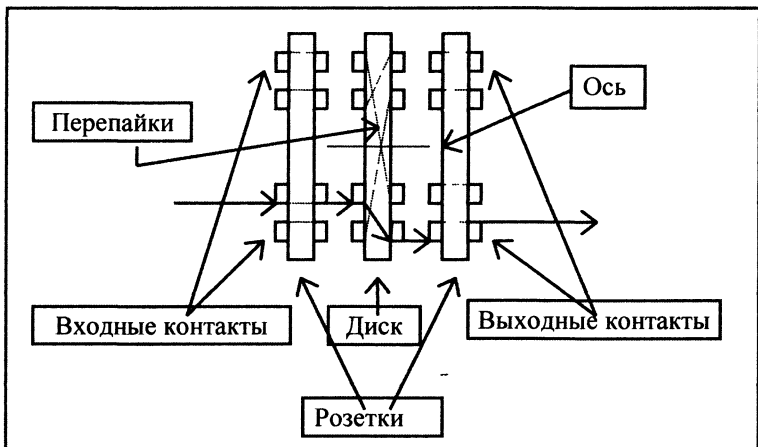


Рис. 6. Шифрующий диск

Х. Ф. Кох предлагал конструкцию шифрующего диска, в котором роль электричества выполняла пневматика. Речь идет о каналах, соединяющих входные и выходные контакты, по которым может проходить поток воздуха, водная или масляная струя и т. п. Любопытно, что подобные дисковые системы на основе пневматики были реально изготовлены и использовались на практике.

Принцип шифрующего диска использовали и шифрмашины, разработанные А. Шербиусом. Самой знаменитой из них была «Энигма», которая в двух отношениях отличалась от других дисковых машин. Во-первых, после блока дисков была расположена неподвижная *обратимая розетка*, контакты которой были попарно соединены друг с другом. Импульс тока, приходивший на этот контакт, заворачивался и вновь проходил через блок дисков в противоположном направле-

нии. Это давало двойное шифрование каждой буквы. Другая особенность “Энигмы” заключалась в неравномерном движении дисков, которое управлялось зубчатыми колесами.

В 1923 г. “Энигма” выставлялась на конгрессе международного почтового союза, однако это не способствовало ее коммерческому успеху: она не раскупалась. За десять лет фирма Шербиуса, производившая “Энигму”, не получила прибыли и в 1934 г. была ликвидирована и передала свои активы другой фирме. После прихода к власти в Германии Гитлера началось серьезное перевооружение армии, и немецкие специалисты сочли “Энигму” достаточно удобной и надежной шифрмашинной. В довоенный период и во время второй мировой войны “Энигма” широко использовалась в германской армии, ВМС и ВВС. Она была портативной (размером с пишущую машинку), работала от батареи, имела деревянный футляр. Ее серьезный недостаток состоял в том, что она не печатала шифртекст (а имела лишь загорающиеся лампочки, отвечающие буквам), и для быстрой работы требовались три или даже четыре человека — для чтения и набора на клавиатуре текста сообщения, диктовки высвечивающихся букв шифртекста и их записи.

С “Энигмой” теснейшим образом связан ход многих событий периода второй мировой войны. Дело в том, что она являлась источником ценнейших сведений для английских спецслужб, читавших переписку “Энигмы” (в рамках *операции “Ультра”*). Эта информация стоила так дорого, что У. Черчилль пожертвовал городом Ковентри, когда ему стал известен план германской бомбардировки этого английского города. С “Энигмой” связано также появление первой в истории вычислительной машины, сконструированной в 1942 г. для перебора ключевых элементов группой специалистов-криптографов под руководством известного математика А. Тьюринга.

Еще один патент на дисковую машину был выдан А. Г. Дамму в 1919 г. Устройство этой машины было настоль-

ко сложным, что никогда не было реализовано. Но его автор основал компанию по производству шифрмашин, которая впоследствии стала прибыльной. Среди вкладчиков капитала были Э. Нобель, племянник знаменитого А. Нобеля, и Ц. Хагелин, управляющий нефтедобывающей компанией братьев Нобелей в России и некоторое время бывший генеральным консулом Швеции в Санкт-Петербурге. До 1927 г. эта компания не имела больших успехов. Их появление было связано с именем сына Ц. Хагелина — Б. Хагелина, родившегося на Кавказе, проучившегося несколько лет в Петербургском университете и получившего позже диплом инженера-механика в Швеции.

В 1925 г. Б. Хагелину удалось модернизировать одну из машин Дамма, снабдив ее клавиатурой и индикаторными лампочками, как у “Энигмы”. Это была также колесная машина, работающая, однако, по иному, чем дисковые машины, принципу. Она получила название В-21. Ее работа была основана на матричном *коммутаторе*, в котором электрически изменялось соединение строк и столбцов для преобразования буквы открытого текста в букву шифртекста. Эти изменения определялись группой ключевых колес, каждое из которых имело по ободу выдвинутые или вдвинутые *штифты*. Колеса имели различные числа штифтов, так что период многоалфавитного шифра, реализуемого машиной, был равен произведению чисел штифтов на всех колесах. В 1926 г. Б. Хагелин предложил В-21 шведской армии, которая сделала на нее большой заказ.

В 1927 г. Б. Хагелин возглавил фирму, выкупленную семьей Хагелин. Свою следующую машину В-211 он снабдил печатающим устройством, работавшим со скоростью около 200 знаков в минуту. Она была самой портативной печатающей шифрмашинной в 1934 г.

В том же году французский генштаб заказал Б. Хагелину карманную печатающую машину, которая могла бы обслуживаться одним человеком. Через некоторое время такая машина

была изготовлена. Она реализовывала шифр гаммирования, причем для выработки гаммы была использована идея суммирующего устройства, состоящего из комбинационных линеек, расположенных в цилиндрическом барабане. На линейках рядами были расположены так называемые *рейтеры*. При повороте барабана на 360° рейтеры, вступая во взаимодействие с другими элементами схемы, могли выдвигать некоторые линейки влево, причем число выдвинутых линеек и определяло значение знака гаммы (от 0 до 25) в данный такт шифрования. Во взаимодействие с рейтерами вступали штифты, расположенные на колесах блока дисков, составляющего вторую основную часть машины. Размеры и схема движения дисков обеспечивали период, приблизительно равный $1,01 \cdot 10^8$. Как расположение рейтеров, так и расположение штифтов могло легко меняться, они являлись ключевыми элементами. Это была машина С-36, ставшая впоследствии знаменитой. По размерам она была меньше телефонного аппарата, весила вместе с футляром около двух с половиной килограммов. Французы сразу же сделали заказ на 5000 машин. Позднее машина была существенно усовершенствована, ею заинтересовались в США. В 1939 г. она была взята на вооружение армии США. Под военным наименованием М-209 она использовалась в качестве полевого шифра на протяжении всей второй мировой войны. Всего было произведено около 140 000 таких машин. Позже фирма Хагелин стала производить широко известные машины С-48, С-52, Т-55 и многие другие.

Среди заметных фигур в криптографии первой половины XX в. выделяется У. Фридман, получивший серьезные теоретические результаты в криптоанализе и ставший известным благодаря своим заслугам по вскрытию военных шифров Японии и Германии.

У. Фридман родился в 1891 г. в Кишиневе, в семье переводчика, работавшего в русском почтовом ведомстве. В 1892 г. его семья эмигрировала в США, где отец стал заниматься швейными машинами. У. Фридман в 1914 г. окончил

Корнельский университет по специальности генетика. В городе Итака, где проживала семья Фридмана, крупный бизнесмен Д. Фабиан имел собственные лаборатории по акустике, генетике и криптографии. Любопытно, что криптографией Д. Фабиан увлекся, пытаясь доказать, что автором пьес У. Шекспира являлся Ф. Бэкон.

В 1915 г. Д. Фабиан нанял на работу в свое поместье Ривербэнк специалиста по генетике. Им стал У. Фридман. Вскоре он увлекся криптографией и проявил себя в этом деле. Через некоторое время У. Фридман уже возглавлял в Ривербэнкских лабораториях два отдела — генетики и шифров.

Помимо криптоаналитической работы У. Фридман занимался преподаванием в классе, состоявшем из армейских офицеров, присланных в Ривербэнк для изучения криптографии. До 1918 г. им был подготовлен цикл из семи лекций, восьмую он написал после возвращения со службы в качестве дешифровальщика в американских экспедиционных силах (шла первая мировая война). Известные все вместе как *Ривербэнкские публикации*, эти работы являются серьезным вкладом в теоретическую криптографию.

Наибольший интерес с точки зрения современной криптографии представляют лекции “Методы раскрытия шифров с длинной связной гаммой” и “Индекс совпадения и его применения в криптографии” [Fri20]. В первой из них предлагается бесключевой метод чтения при использовании неравновероятной гаммы. Во второй излагается так называемый *k-тест*, позволяющий выяснить, можно ли подписать друг под другом две (или более) криптограммы (или отрезки криптограмм) так, чтобы буквы в каждой колонке оказались бы зашифрованы одинаковыми знаками гаммы.

Поступив в 1921 г. на службу в войска связи, У. Фридман успешно применял свои методы для вскрытия машинных шифров. Когда была создана служба радиоразведки, У. Фридман стал ее главой и продолжил свои разработки, самой значимой из которых было вскрытие японской *пурпурной шифрмашинны*. В 1929 г.

он стал широко известен как один из ведущих криптографов мира, когда “Британская энциклопедия” поместила его статью “О кодах и шифрах”. С основными результатами У. Фридмана можно познакомиться в четырехтомнике “Военная криптография” [Fri85].

Выдающиеся результаты в применении математических методов в криптографии принадлежат Клоду Шеннону. К. Шеннон получил образование по электронике и математике в Мичиганском университете, где и начал проявлять интерес к теории связи и теории шифров. В 1940 г. он получил степень доктора по математике, в течение года обучался в Принстонском институте усовершенствования, после чего был принят на службу в лабораторию компании “Bell Telephone”.

К 1944 г. К. Шеннон завершил разработку теории секретной связи. В 1945 г. им был подготовлен секретный доклад “Математическая теория криптографии”, который был рассекречен в 1949 г. и издан [Шен63].

В данной работе излагается теория так называемых *секретных систем*, служащих фактически математической моделью шифров. Помимо основных алгебраических (или функциональных) свойств шифров, постулируемых в модели, множества сообщений и ключей наделяются соответствующими априорными вероятностными свойствами, что позволяет формализовать многие постановки задач синтеза и анализа шифров. Так, и сегодня при разработке новых классов шифров широко используется принцип Шеннона *рассеивания и перемешивания*, состоящий в использовании при шифровании многих итераций “рассеивающих” и “перемешивающих” преобразований.

Разработанные К. Шенноном концепции *теоретической и практической секретности* (или стойкости) позволяют количественно оценивать криптографические качества шифров и пытаться строить в некотором смысле *идеальные* или *совершенные шифры*. Моделируется также и язык открытых сообщений. А именно, предлагается рассматривать язык как *вероятностный*

процесс, который создает дискретную последовательность символов в соответствии с некоторой вероятностной схемой.

Центральной в работах К. Шеннона является концепция *избыточной информации*, содержащейся в текстовых сообщениях. Избыточность означает, что в сообщении содержится больше символов, чем в действительности требуется для передачи содержащейся в нем информации. Например, всего лишь десять английских слов — **the, of, and, to, a, in, that, it, is, i** — составляют более 25% любого (английского) текста. Легко понять, что их можно изъять из текста без потери информации, так как их легко восстановить по смыслу (или по контексту). Фактически К. Шеннон показал, что успех криптоанализа определяется тем, насколько избыточность, имеющаяся в сообщении, “переносится” в зашифрованный текст. Если шифрование “стирает” избыточность, то восстановить текст сообщения по криптограмме становится принципиально невозможно.

Задачу дешифрования К. Шеннон рассматривает как задачу вычисления апостериорных знаний противника о шифре после перехвата криптограммы. Дело в том, что вероятности сообщений и ключей составляют априорные знания противника, которыми он располагает в соответствии с правилом Керкгоффа. После перехвата криптограммы он может (по крайней мере, в принципе, поскольку множества сообщений и ключей *конечны*) вычислить апостериорные вероятности возможных ключей и сообщений, которые могли быть использованы при составлении данной криптограммы. Вот эти вероятности и составляют апостериорные знания противника. С этой точки зрения показателен следующий пример.

Пусть для зашифрования нормативного английского языка применяется шифр простой замены, в котором каждый из 26! ключей может быть выбран с равной вероятностью. Пусть противник знает об источнике сообщений лишь то, что он создает английский текст. Тогда априорными вероятностями различных сообщений из N букв являются их относительные частоты в нормативном тексте. Если же противник перехватил крипто-

грамму из N букв, то он может вычислить условные вероятности открытых текстов и ключей, которые могут создать такую криптограмму. Если N достаточно велико, скажем $N = 50$, то *обычно* имеется единственное сообщение (и единственный ключ) с условной вероятностью, близкой к единице (это — само сообщение, подвергнутое шифрованию), в то время как все другие сообщения имеют суммарную вероятность, близкую к нулю. Таким образом, имеется, по существу, единственное “решение” такой криптограммы. Для меньших значений N , скажем $N = 10$, *обычно* найдется несколько пар сообщений и ключей, вероятности которых сравнимы друг с другом, то есть нет ни одного сообщения (и ключа) с вероятностью, близкой к единице. В этом случае “решение” криптограммы неоднозначно.

Понятие *совершенной секретности* К. Шеннон определяет требованием, чтобы апостериорные знания противника в точности совпадали бы с априорными знаниями. Он приводит пример *совершенного шифра*, которым является шифр Вернама (со случайной равновероятной гаммой). Следует подчеркнуть, что все рассуждения о стойкости шифров К. Шеннон проводит лишь для одной постановки задачи криптоанализа: когда противник располагает лишь *одной криптограммой* и требуется найти текст сообщения. Для других постановок задач требуются отдельные исследования.

Теоретической мерой секретности (или стойкости) по К.Шеннону является энтропийная характеристика — *неопределенность шифра по открытому сообщению*, которая измеряет (в статистическом смысле), насколько “близка” *средняя* криптограмма из N букв к единственному “решению”. Он выводит формулу для приближенного вычисления минимального N , при котором находится единственное “решение”. Такая величина получила название *расстояния единственности*. Формула для расстояния единственности связывает между собой неопределенность шифра по открытому тексту и избыточность текста. Чем большим оказывается расстояние единственности,

тем более шифр приближается к совершенному шифру, для которого формально расстояние единственности равно ∞ .

Наконец, К. Шеннон вводит понятие *рабочей характеристики* шифра, подходя к практической оценке стойкости. Он формулирует также основные критерии оценки качества секретных систем с позиций практики их использования.

Как видим, К. Шеннону удалось решить фундаментальные проблемы в теоретической криптографии. Его работы стимулировали бурный рост научных исследований по теории информации и криптографии.

В работах К. Шеннона по исследованию свойств языка важную роль играет величина удельной энтропии H на букву текста, другими словами, среднее количество информации, передаваемой буквой открытого текста. Предложенный им метод экспериментов с угадыванием очередной буквы английского текста по предыдущим буквам оказался неэффективным при получении оценок величины H для других языков. Метод “отгадывания” развил в своих работах А. Н. Колмогоров. Достаточно точные приближения параметра H для русского и французского языков получил Б. Б. Пиотровский. Он указал на существенную разницу между значениями H для текстов различного характера (литературных, деловых, разговорной речи).

Понятие “количества информации”, содержащейся в тексте, базировалось, по К. Шеннону, лишь на частотных характеристиках. В своих фундаментальных работах 60-х годов А. Н. Колмогоров подошел к определению количества информации с учетом *смыслового содержания* текста, что позволило уточнить приближение величины H для литературных текстов. Необходимо также отметить, что еще задолго до К. Шеннона частотные характеристики языка изучал выдающийся русский ученый А. А. Марков. Сегодня часто используются так называемые *марковские модели* открытых текстов, учитывающие зависимости букв текста от предыдущих букв.

Следующая страница в истории криптографии XX в. посвящена телефонным шифраторам, которые были разработаны в 30-х

годах и стали широко использоваться во время второй мировой войны. В России разработка телефонного шифратора велась под руководством В.А.Котельникова, ставшего впоследствии академиком, ученым с мировым именем. Ему принадлежит знаменитая *теорема дискретизации* (или *теорема отсчетов*), лежащая в основе *теории цифровой обработки сигналов*.

Согласно [Каб67], идея телефонного шифратора была запатентована Д.Х.Роджерсом еще в 1881 г., спустя пять лет после изобретения Беллом телефона. Идея состояла в передаче телефонного сообщения по нескольким (в простейшем случае — по двум) цепям поочередными импульсами в некоторой быстро изменяющейся последовательности. Предлагалось разнести такие линии на значительное расстояние друг от друга с тем, чтобы устранить возможность подключения сразу ко всем одновременно. Подключение же к одной из них позволяло бы слышать лишь отдельные неразборчивые сигналы.

В более поздних разработках предлагались различные преобразования непосредственно самой речи. Звуки речи преобразуются телефоном в непрерывный электрический сигнал, который с помощью соответствующих устройств изменяется шифратором по законам электричества. К числу возможных изменений относятся: *инверсия*, *смещение*, или *деление диапазона частот*, *шумовые маскировки*, *временные перестановки* частей сигнала, а также различные комбинации перечисленных преобразований. Естественно, каждое из указанных преобразований производится под управлением ключа, который имеется у отправителя и получателя. Наиболее просто реализуемым являлось преобразование инверсии. Сложнее реализовались временные перестановки. Для их осуществления речевой сигнал в некоторый промежуток времени предварительно записывался на магнитофонной ленте. Запись делилась на отрезки длительностью в доли секунд. Отрезки с помощью нескольких магнитных головок разносились и перемешивались, в результате чего в канале слышалась хаотическая последовательность звуков. Использовалась также движущаяся магнитная головка, которая в зависимости от направ-

ления движения считывала сигналы быстрее или медленнее, чем они были записаны на ленте. В результате тон сигналов становился выше или ниже обычного, в канале быстро чередовались высокие и низкие звуки, не воспринимаемые ухом. Следует отметить, что одной из самых сложных проблем, которые возникали при разработке телефонных шифраторов, была *проблема узнавания* восстановленной после расшифрования речи.

В США первый *телефонный шифратор*, под названием АЗ, был принят в эксплуатацию в 1937 г. Именно он доставил президенту Рузвельту известие о начале второй мировой войны утром 1 сентября 1939 г. по вызову американского посла в Париже. АЗ осуществлял инверсию и перестановку 5 поддиапазона частот. Из 3840 возможных комбинаций ($5! \cdot 2^5$) фактически использовались лишь 6, которые менялись 36 раз за каждые 20 секунд. Слабость используемой криптографии компенсировалась регулярным изменением частот передачи.

В настоящее время аналоговая телефония уступает место цифровой телефонии. Тем самым и многие технические проблемы, связанные с криптографическими преобразованиями аналоговых сигналов, отпадают за ненадобностью. Дело в том, что *оцифрованный* сигнал является дискретным и, следовательно, к нему можно применить хорошо разработанную надежную “дискретную криптографию”.

Во второй половине XX в., вслед за развитием элементной базы вычислительной техники, появились *электронные шифраторы*, разработка которых потребовала серьезных теоретических исследований во многих областях прикладной и фундаментальной математики, в первую очередь алгебре, теории вероятностей и математической статистике. Сегодня именно электронные шифраторы составляют подавляющую долю средств шифрования. Они удовлетворяют все возрастающим требованиям по надежности и скорости шифрования. Прогресс в развитии вычислительной техники сделал возможными *программные реализации* криптографических алгоритмов, которые все увереннее

вытесняют во многих сферах традиционные аппаратные средства.

В семидесятых годах произошло два события, серьезно повлиявших на дальнейшее развитие криптографии. Во-первых, был принят (и опубликован!) первый *стандарт шифрования данных* (DES), “легализовавший” принцип Керкгоффса в криптографии. Во-вторых, после работы американских математиков У.Диффи и М.Хеллмана [Dif76] родилась “новая криптография” — *криптография с открытым ключом*. Оба этих события были рождены потребностями бурно развивающихся средств коммуникаций, в том числе локальных и глобальных компьютерных сетей, для защиты которых потребовались легко доступные и достаточно надежные криптографические средства. Криптография стала широко востребоваться не только в военной, дипломатической, государственной сферах, но также в коммерческой, банковской и других сферах.

Вслед за идеей Диффи и Хеллмана, связанной с гипотетическим понятием *однопавленной (или односторонней) функции с секретом*, появились “кандидат” на такую функцию и реально осуществленная шифрсистема RSA с открытым ключом. Такая система была предложена в 1978 г. Райвестом, Шамиром и Адлеманом. Парадоксальным казалось то, что в RSA для зашифрования и расшифрования используются разные ключи, причем ключ зашифрования может быть *открытым*, то есть всем известным. Вслед за RSA появился целый ряд других систем. В связи с несимметричным использованием ключей стал использоваться термин *асимметричная шифрсистема*, в то время как традиционные шифрсистемы стали называться *симметричными*.

Наряду с идеей открытого шифрования Диффи и Хеллман предложили идею *открытого распределения ключей*, позволяющую избавиться от защищенного канала связи при рассылке криптографических ключей. Их идея основывалась на сложности решения задачи *дискретного логарифмирования*,

то есть задачи, являющейся обратной для задачи возведения в степень в конечном поле большого порядка.

Позволим себе остановиться в нашем кратком историческом обзоре на этом этапе развития криптографии, поскольку рассказ о ее современных проблемах и приложениях заставил бы нас приступить к изложению основного содержания книги.

Контрольные вопросы

1. Приведите примеры шифров, применявшихся еще до нашей эры.
2. Приведите пример шифра, для которого сам открытый текст является ключом.
3. Какие шифры называются омофонами? В чем их преимущество перед шифрами простой замены?
4. Что является ключом шифра Виженера?
5. Являлись ли трафареты, которые использовали А.Грибоедов и Ришелье для передачи тайных сообщений, средствами шифрования?
6. Приведите пример шифра, допускающего неоднозначное зашифрование.
7. Какими шифрами пользовались Цезарь, Галилей, Наполеон, Ришелье?
8. В чем состоит правило Керкгоффа? Почему это правило является общепринятым в криптографии?
9. Чем отличаются принципы шифрования в аналоговой телефонии от принципов шифрования телеграфных сообщений?
10. Чем отличаются симметричные шифрсистемы от асимметричных?
11. Когда родилась криптография с открытыми ключами и первая реальная система шифрования?
12. Каких выдающихся криптографов XX в. Вы знаете?

Глава 2

Основные понятия

Целью настоящего раздела является определение основных понятий и задач криптографии.

§ 2.1. Криптография

В переводе с греческого языка слово *криптография* означает тайнопись. Смысл этого термина выражает основное предназначение криптографии — защитить или сохранить в тайне необходимую информацию.

Криптография дает средства для защиты информации, и поэтому она является частью деятельности по обеспечению безопасности информации.

Существуют различные методы *защиты информации*. Можно, например, физически ограничить доступ к информации путем хранения ее в надежном сейфе или строго охраняемом помещении. При хранении информации такой метод удобен, однако при ее передаче приходится использовать другие средства.

Можно воспользоваться одним из известных методов сокрытия информации:

- скрыть канал передачи информации, используя нестандартный способ передачи сообщений;

- замаскировать канал передачи закрытой информации в открытом канале связи, например спрятав информацию в безобидном “контейнере” с использованием тех или других стеганографических способов либо обмениваясь открытыми сообщениями, смысл которых согласован заранее;

- существенно затруднить возможность перехвата противником передаваемых сообщений, используя специальные методы передачи по широкополосным каналам, сигнала под

уровнем шумов, либо с использованием “прыгающих” несущих частот и т. п.

В отличие от перечисленных методов криптография не “прячет” передаваемые сообщения, а преобразует их в форму, недоступную для понимания противником. При этом обычно исходят из предположения о полном контроле противником канала связи. Это означает, что противник может не только пассивно перехватывать передаваемые сообщения для последующего их анализа, но и активно изменять их, а также отправлять поддельные сообщения от имени одного из абонентов.

Помимо сокрытия существуют и другие проблемы защиты передаваемой информации. Например, при полностью открытом информационном обмене возникает проблема достоверности полученной информации. Для ее решения необходимо обеспечить:

— проверку и подтверждение подлинности содержания и источника сообщения,
а также

— предотвращение и обнаружение обмана и других умышленных нарушений со стороны самих участников информационного обмена.

Для решения этой проблемы обычные средства, применяемые при построении систем передачи информации, подходят далеко не всегда. Именно криптография дает средства для обнаружения обмана в виде подлога или отказа от ранее совершенных действий, а также других неправомерных действий.

Поэтому можно сказать, что современная *криптография* является областью знаний, связанной с решением таких проблем безопасности информации, как конфиденциальность, целостность, аутентификация и невозможность отказа сторон от авторства. Достижение этих требований безопасности информационного взаимодействия и составляет основные цели криптографии. Они определяются следующим образом.

Обеспечение *конфиденциальности* — решение проблемы защиты информации от ознакомления с ее содержанием со стороны лиц, не имеющих права доступа к ней. В зависимости от контекста вместо термина “конфиденциальная” информация могут выступать термины “секретная”, “частная”, “ограниченного доступа” информация.

Обеспечение *целостности* — гарантирование невозможности несанкционированного изменения информации. Для гарантии целостности необходим простой и надежный критерий обнаружения любых манипуляций с данными. Манипуляции с данными включают вставку, удаление и замену.

Обеспечение *аутентификации* — разработка методов подтверждения подлинности сторон (*идентификация*) и самой информации в процессе информационного взаимодействия. Информация, передаваемая по каналу связи, должна быть аутентифицирована по источнику, времени создания, содержанию данных, времени пересылки и т. д.

Обеспечение *невозможности отказа от авторства* — предотвращение возможности отказа субъектов от некоторых из совершенных ими действий. Рассмотрим средства для достижения этих целей более подробно.

Конфиденциальность

Традиционной задачей криптографии является проблема обеспечения конфиденциальности информации при передаче сообщений по контролируемому противником каналу связи. В простейшем случае эта задача описывается взаимодействием трех субъектов (сторон). Владелец информации, называемый обычно *отправителем*, осуществляет преобразование исходной (*открытой*) информации (сам процесс преобразования называется *шифрованием*) в форму передаваемых *получателю* по открытому каналу связи *шифрованных* сообщений с целью ее защиты от противника.

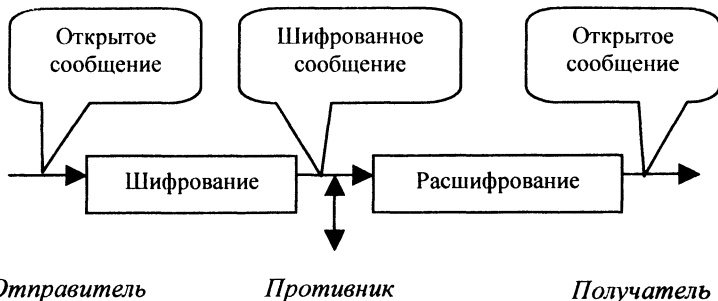


Рис. 7. Передача зашифрованной информации

Под *противником* понимается любой субъект, не имеющий права ознакомления с содержанием передаваемой информации. В качестве противника может выступать *криптоаналитик*, владеющий методами раскрытия шифров. Законный получатель информации осуществляет *расшифрование* полученных сообщений. Противник пытается овладеть защищаемой информацией (его действия обычно называют *атаками*). При этом он может совершать как пассивные, так и активные действия. *Пассивные* атаки связаны с прослушиванием, анализом трафика, перехватом, записью передаваемых зашифрованных сообщений, *дешифрованием*, т. е. попытками “взломать” защиту с целью овладения информацией.

При проведении *активных* атак противник может прерывать процесс передачи сообщений, создавать поддельные (сфабрикованные) или модифицировать передаваемые зашифрованные сообщения. Эти активные действия называют попытками *имитации* и *подмены* соответственно.

Под *шифром* обычно понимается семейство обратимых преобразований, каждое из которых определяется некоторым параметром, называемым *ключом*, а также порядком применения данного преобразования, называемым *режимом шифрования*¹.

¹ Формальное определение шифра будет дано ниже.

Ключ — это важнейший компонент шифра, отвечающий за выбор преобразования, применяемого для зашифрования конкретного сообщения. Обычно ключ представляет собой некоторую буквенную или числовую последовательность. Эта последовательность как бы “настраивает” алгоритм шифрования.

Каждое преобразование однозначно определяется ключом и описывается некоторым *криптографическим алгоритмом*. Один и тот же криптографический алгоритм может применяться для шифрования в различных режимах. Тем самым реализуются различные способы шифрования (простая замена, гаммирование² и т. п.). Каждый режим шифрования имеет как свои преимущества, так и недостатки. Поэтому выбор режима зависит от конкретной ситуации. При расшифровании используется криптографический алгоритм, который в общем случае может отличаться от алгоритма, применяемого для зашифрования сообщения. Соответственно могут различаться ключи зашифрования и расшифрования. Пару алгоритмов зашифрования и расшифрования обычно называют *криптосистемой (шифрсистемой)*, а реализующие их устройства — *шифртехникой*.

Если обозначить через M открытое, а через C шифрованное сообщения, то процессы зашифрования и расшифрования можно записать в виде равенств

$$E_{k_1}(M) = C,$$

$$D_{k_2}(C) = M,$$

в которых алгоритмы зашифрования E и расшифрования D должны удовлетворять равенству

$$D_{k_2}(E_{k_1}(M)) = M.$$

² Эти способы шифрования будут рассмотрены ниже.

Различают *симметричные* и *асимметричные* криптосистемы. В симметричных системах знание ключа зашифрования k_1 позволяет легко найти ключ расшифрования k_2 (в большинстве случаев эти ключи просто совпадают). В асимметричных криптосистемах знание ключа k_1 не позволяет определить ключ k_2 . Поэтому для симметричных криптосистем оба ключа должны сохраняться в секрете, а для асимметричных — только один — ключ расшифрования k_2 , а ключ k_1 можно сделать открытым (общедоступным). В связи с этим их называют еще *шифрами с открытым ключом*.

Симметричные криптосистемы принято подразделять на *поточные* и *блочные* системы. Поточные системы осуществляют зашифрование отдельных символов открытого сообщения. Блочные же системы производят зашифрование блоков фиксированной длины, составленных из подряд идущих символов сообщения.

Асимметричные криптосистемы, как правило, являются блочными. При их использовании можно легко организовать передачу конфиденциальной информации в сети с большим числом пользователей. В самом деле, для того чтобы послать сообщение, отправитель открыто связывается с получателем, который либо передает свой ключ отправителю, либо помещает его на общедоступный сервер. Отправитель зашифровывает сообщение на открытом ключе получателя и отправляет его получателю. При этом никто, кроме получателя, обладающего ключом расшифрования, не сможет ознакомиться с содержанием передаваемой информации. В результате такая система шифрования с общедоступным ключом позволяет существенно сократить объем хранимой каждым абонентом секретной ключевой информации.

Возможна и другая симметричная ситуация, когда открытый и секретный ключи меняются местами. Предположим, например, что для проведения контроля соблюдения выполнения каждой стороной договора об ограничении испытаний ядерного оружия создаются пункты контроля, которые ведут

запись и конфиденциальную передачу сторонам, участвующим в договоре, сейсмологической информации. Поскольку на каждом таком пункте контролируемая сторона одна, а участников договора может быть очень много, то необходимо обеспечить такое шифрование информации, при котором зашифровать сообщение мог бы только один отправитель, а расшифровать мог бы каждый.

Не существует единого шифра, подходящего для всех случаев жизни. Выбор способа шифрования (то есть криптографического алгоритма и режима его использования) зависит от особенностей передаваемой информации (ее ценности, объема, способа представления, необходимой скорости передачи и т. д.), а также возможностей владельцев по защите своей информации (стоимость применяемых технических устройств, удобство использования, надежность функционирования и т. п.). Имеется большое разнообразие видов защищаемой информации: текстовая, телефонная, телевизионная, компьютерная и т. д., причем у каждого вида информации имеются свои существенные особенности, которые надо учитывать при выборе способа шифрования. Большое значение имеют объемы и требуемая скорость передачи зашифрованной информации, а также помехозащищенность используемого канала связи. Все это существенным образом влияет на выбор криптографического алгоритма и организацию защищенной связи.

Наличие надежного криптографического алгоритма и правильный выбор режима еще не гарантируют владельцу защищенность передаваемой информации. Немаловажную роль играет правильность их использования. Поскольку даже самые стойкие шифры при неправильном использовании существенно теряют свои качества, то конфиденциальность передаваемой информации во многом зависит от того, какие ошибки допускает ее владелец при использовании криптографической защиты. А то, что все пользователи допускают ошибки, — неизбежно и является непреложным и важным

(для криптоаналитика) фактом, поскольку любые криптографические средства, какими бы они ни были удобными и прозрачными, всегда мешают пользователям в работе, а различные тонкости известны только криптоаналитикам и, как правило, непонятны пользователям этих средств. Более того, в качестве субъектов взаимодействия могут выступать не только люди, но и различные процессы, осуществляющие обработку информации в автоматизированной системе без участия человека. Поэтому защищенность информации в системе существенно зависит от того, насколько правильно там реализована криптографическая подсистема, отвечающая за выполнение криптографических функций. Одно наличие такой подсистемы еще ничего не гарантирует.

Подчеркнем разницу между терминами “*расшифрование*” и “*дешифрование*”. При расшифровании действующий ключ считается известным, в то время как при дешифровании ключ неизвестен. Тем самым расшифрование должно осуществляться столь же просто, как и зашифрование; дешифрование представляет собой значительно более сложную задачу. Именно в этом и состоит смысл шифрования.

Для разных шифров задача дешифрования имеет различную сложность. Уровень сложности этой задачи и определяет главное свойство шифра — способность противостоять попыткам противника завладеть защищаемой информацией. В связи с этим говорят о *криптографической стойкости* шифра (или просто *стойкости*), различая более стойкие и менее стойкие шифры. Методы вскрытия шифров разрабатывает наука, носящая название *криптоанализ*. Согласно [Кан67], термин криптоанализ ввел У. Фридман в 1920 г.

В иностранной литературе часто используется термин *криптология* в качестве области знаний, объединяющей *криптографию* и *криптоанализ*, при этом криптография понимается как наука о создании шифров. Если следовать такому толкованию, то можно прийти к тезису о том, что задачи создания криптографической защиты и ее преодоления прин-

ципально различаются. На самом деле было бы нелогично считать защиту надежной без проведения детального криптоанализа. В связи с этим, применительно к задаче обеспечения конфиденциальности, мы будем считать термины криптография и криптология синонимами, понимая их как название науки о синтезе шифров и “взломе” шифров.

Целостность

Наряду с конфиденциальностью не менее важной задачей является обеспечение целостности информации, другими словами, — неизменности ее в процессе передачи или хранения. Решение этой задачи предполагает разработку средств, позволяющих обнаруживать не столько случайные искажения (для этой цели вполне подходят методы теории кодирования с обнаружением и исправлением ошибок), сколько целенаправленное навязывание противником ложной информации. Для этого в передаваемую информацию вносится избыточность. Как правило, это достигается добавлением к сообщению некоторой проверочной комбинации, вычисляемой с помощью специального алгоритма и играющей роль контрольной суммы для проверки целостности полученного сообщения. Главное отличие такого метода от методов теории кодирования состоит в том, что алгоритм выработки проверочной комбинации является “криптографическим”, то есть зависящим от секретного ключа. Без знания секретного ключа вероятность успешного навязывания противником искаженной или ложной информации мала. Такая вероятность служит мерой *имитостойкости* шифра, то есть способности самого шифра противостоять активным атакам со стороны противника.

Итак, для проверки целостности к сообщению M добавляется проверочная комбинация S , называемая *кодом аутентификации сообщения* (сокращенно — КАС) или *имитовставкой*. В этом случае по каналу связи передается пара $C = (M, S)$. При получении сообщения M пользователь вычисляет значение проверочной комбинации и сравнивает его

с полученным контрольным значением S . Несовпадение говорит о том, что данные были изменены.

Как правило, код аутентификации является значением некоторой (зависящей от секретного ключа) криптографической *хэш-функции*³ от данного сообщения: $h_k(M) = S$. К кодам аутентификации предъявляются определенные требования. К ним относятся:

— невозможность вычисления значения $h_k(M) = S$ для заданного сообщения M без знания ключа k ;

— невозможность подбора для заданного сообщения M с известным значением $h_k(M) = S$ другого сообщения M_1 с известным значением $h_k(M_1) = S_1$ без знания ключа k .

Первое требование направлено против создания поддельных (сфабрикованных) сообщений при атаках типа *имитация*; второе — против модификации передаваемых сообщений при атаках типа *подмена*.

Аутентификация

Аутентификация — установление подлинности. В общем случае этот термин может относиться ко всем аспектам информационного взаимодействия: сеансу связи, сторонам, передаваемым сообщениям и т. д.

Установление подлинности (то есть проверка и подтверждение) всех аспектов информационного взаимодействия является важной составной частью проблемы обеспечения достоверности получаемой информации. Особенно остро эта проблема стоит в случае не доверяющих друг другу сторон, когда источником угроз может служить не только третья сторона (противник), но и сторона, с которой осуществляется взаимодействие.

³ Так обычно называется функция, принимающая значения некоторой фиксированной размерности.

Рассмотрим эти вопросы более подробно.

Применительно к сеансу связи (транзакции) аутентификация означает проверку: целостности соединения, невозможности повторной передачи данных противником и своевременности передачи данных. Для этого, как правило, используют дополнительные параметры, позволяющие “сцепить” передаваемые данные в легко проверяемую последовательность. Это достигается, например, путем вставки в сообщения некоторых специальных чисел или *меток времени*. Они позволяют предотвратить попытки повторной передачи, изменения порядка следования или обратной отсылки части переданных сообщений. При этом такие вставки в передаваемом сообщении необходимо защищать (например, с помощью шифрования) от возможных подделок и искажений.

Применительно к сторонам взаимодействия аутентификация означает проверку одной из сторон того, что взаимодействующая с ней сторона — именно та, за которую она себя выдает. Часто аутентификацию сторон называют также *идентификацией*⁴.

Основным средством для проведения идентификации являются *протоколы идентификации*, позволяющие осуществлять идентификацию (и аутентификацию) каждой из участвующих во взаимодействии и не доверяющих друг другу сторон. Различают *протоколы односторонней* и *взаимной идентификации*.

Протокол — это распределенный алгоритм, определяющий последовательность действий каждой из сторон. В процессе выполнения протокола идентификации каждая из сто-

⁴ Формально это некорректно, так как под идентификацией понимают процедуру установления присвоенного данной стороне уникального системного имени-идентификатора, которое позволяет отличать ее от других сторон; обычно идентификация заключается в предъявлении этого имени и предшествует процедуре аутентификации, то есть подтверждению правильности идентификации.

рон не передает никакой информации о своем секретном ключе, а хранит его у себя и использует для формирования ответных сообщений на запросы, поступающие при выполнении протокола.

Наконец, применительно к самой информации аутентификация означает проверку того, что информация, передаваемая по каналу, является подлинной по содержанию, источнику, времени создания, времени пересылки и т. д.

Проверка подлинности содержания информации сводится, по сути, к проверке ее неизменности (с момента создания) в процессе передачи или хранения, то есть проверке целостности.

Аутентификация источника данных означает подтверждение того, что исходный документ был создан именно заявленным источником.

Заметим, что если стороны доверяют друг другу и обладают общим секретным ключом, то аутентификацию сторон можно обеспечить применением кода аутентификации. Действительно, каждое успешно декодированное получателем сообщение может быть создано только отправителем, так как только он знает их общий секретный ключ. Для не доверяющих друг другу сторон решение подобных задач с использованием общего секретного ключа становится невозможным. Поэтому при аутентификации источника данных нужен механизм цифровой подписи, который будет рассмотрен ниже.

В целом, аутентификация источника данных выполняет ту же роль, что и протокол идентификации. Отличие заключается только в том, что в первом случае имеется некоторая передаваемая информация, авторство которой требуется установить, а во втором требуется просто установить сторону, с которой осуществляется взаимодействие.

Цифровая подпись

В некоторых ситуациях, например в силу изменившихся обстоятельств, отдельные лица могут отказаться от ранее

принятых обязательств. В связи с этим необходим некоторый механизм, препятствующий подобным попыткам.

Так как в данной ситуации предполагается, что стороны не доверяют друг другу, то использование общего секретного ключа для решения поставленной проблемы становится невозможным. Отправитель может отказаться от факта передачи сообщения, утверждая, что его создал сам получатель (*отказ от авторства*). Получатель легко может модифицировать, подменить или создать новое сообщение, а затем утверждать, что оно получено от отправителя (*приписывание авторства*). Ясно, что в такой ситуации арбитр при решении спора не будет иметь возможность установить истину.

Основным механизмом решения этой проблемы является так называемая *цифровая подпись*.

Хотя цифровая подпись и имеет существенные отличия, связанные с возможностью отделения от документа и независимой передачей, а также возможностью подписывания одной подписью всех копий документа, она во многом аналогична обычной “ручной” подписи.

Схема цифровой подписи включает два алгоритма, один — для вычисления, а второй — для проверки подписи. Вычисление подписи может быть выполнено только автором подписи. Алгоритм проверки должен быть общедоступным, чтобы проверить правильность подписи мог каждый.

Для создания схемы цифровой подписи можно использовать симметричные шифрсистемы. В этом случае подписью может служить само зашифрованное на секретном ключе сообщение. Однако основной недостаток таких подписей состоит в том, что они являются одноразовыми: после каждой проверки секретный ключ становится известным. Единственный выход из этой ситуации в рамках использования симметричных шифрсистем — это введение доверенной третьей стороны, выполняющей функции посредника, которому доверяют обе стороны. В этом случае вся информация пересылается через посредника, он осуществляет перешифрование сообще-

ний с ключа одного из абонентов на ключ другого. Естественно, эта схема является крайне неудобной.

При использовании шифрсистем с открытым ключом возможны два подхода к построению системы цифровой подписи.

Первый подход состоит в преобразовании сообщения в форму, по которой можно восстановить само сообщение и тем самым проверить правильность “подписи”. В данном случае подписанное сообщение имеет, как правило, ту же длину, что и исходное сообщение. Для создания такого “подписанного сообщения” можно, например, произвести зашифрование исходного сообщения на секретном ключе автора подписи. Тогда каждый может проверить правильность подписи путем расшифрования подписанного сообщения на открытом ключе автора подписи.

При втором подходе подпись вычисляется и передается вместе с исходным сообщением. Вычисление подписи заключается в преобразовании исходного сообщения в некоторую цифровую комбинацию (которая и является подписью). Алгоритм вычисления подписи должен зависеть от секретного ключа пользователя. Это необходимо для того, чтобы воспользоваться подписью мог бы только владелец ключа. В свою очередь, алгоритм проверки правильности подписи должен быть доступен каждому. Поэтому, как правило, этот алгоритм зависит от открытого ключа пользователя. В данном случае длина подписи не зависит от длины подписываемого сообщения.

Одновременно с проблемой цифровой подписи возникла проблема построения бесключевых криптографических *хэш-функций*. Дело в том, что при вычислении цифровой подписи оказывается более удобным осуществить сначала хэширование, то есть свертку текста в некоторую комбинацию фиксированной длины, а затем уже подписывать полученную комбинацию с помощью секретного ключа. При этом функция хэширования, хотя и не зависит от ключа и является откры-

той, должна быть “криптографической”. Имеется в виду свойство *односторонности* этой функции: по значению комбинации-свертки никто не должен иметь возможность подобрать соответствующее сообщение.

В настоящее время имеются стандарты на криптографические хэш-функции, утверждаемые независимо от стандартов на криптографические алгоритмы и схемы цифровой подписи.

§ 2.2. Управление секретными ключами

Порядок использования криптографической системы определяется системами установки и управления ключами.

Система установки ключей определяет алгоритмы и процедуры генерации, распределения, передачи и проверки ключей.

Система управления ключами определяет порядок использования, смены, хранения и архивирования, резервного копирования и восстановления, замены или изъятия из обращения скомпрометированных, а также уничтожения старых ключей.

Предварительное распределение ключей

Для надежной защиты информации, передаваемой по открытому каналу связи, применяют криптографические средства. Чтобы воспользоваться ими, необходимо осуществить первоначальный выбор и установку ключей. Для генерации ключей могут применяться различные алгоритмы. Выбранные ключи необходимо как-либо передать взаимодействующим сторонам. Поэтому для первоначального распределения ключей необходим защищенный канал связи.

Самый надежный способ первоначального распределения ключей — это личная встреча всех взаимодействующих сторон. Можно использовать также специальных курьеров, которые будут развозить ключи. Однако при большом числе

взаимодействующих сторон требуется предварительная рассылка значительного объема ключевой информации и последующее ее хранение. Поэтому на практике применяют специальные *системы предварительного распределения ключей*, предусматривающие распределение и хранение не самих ключей, а некоторой меньшей по объему исходной информации, на основе которой в дальнейшем каждая сторона может вычислить ключ для взаимодействия с другой стороной. Система предварительного распределения ключей включает два алгоритма. С помощью первого алгоритма осуществляется генерация исходной информации. Эта информация включает открытую часть, которая будет передана всем сторонам или помещена на общедоступном сервере, а также секретные части каждой стороны. Вторым алгоритмом предназначен для вычисления действующего значения ключа для взаимодействия между абонентами по имеющейся у них секретной и общей открытой части исходной ключевой информации.

Система предварительного распределения ключей должна быть *устойчивой*, то есть учитывать возможность раскрытия части ключей при компрометации, обмане или сговоре абонентов, и *гибкой* — допускать возможность быстрого восстановления путем исключения скомпрометированных и подключения новых абонентов.

Пересылка ключей

После того как предварительное распределение ключей произведено, может потребоваться передача ключей для каждого конкретного сеанса взаимодействия. Передача этих ключей может осуществляться с помощью шифрования с использованием ранее полученных ключей.

Для передачи зашифрованных ключей по открытому каналу связи между не доверяющими друг другу абонентами требуется решение всего комплекса задач по установлению подлинности различных аспектов взаимодействия, начиная от подлинности субъектов взаимодействия, подлинности пере-

даваемых сообщений, подлинности самого сеанса связи и кончая подтверждением правильности (идентичности) полученных абонентами ключей.

Для централизованного управления пересылкой ключей создаются специальные доверенные центры, выполняющие функции центров распределения или перешифрования ключей. Различие между этими центрами заключается в том, что в первом случае генерация ключей осуществляется в центре распределения, а во втором случае — самими абонентами.

Открытое распределение ключей

Наиболее просто распределение ключей осуществляется в *системах открытого распределения (секретных) ключей*. Для сетей связи с большим числом абонентов традиционные подходы к построению системы распределения ключей оказываются очень неудобными. Диффи и Хеллман впервые показали, как можно решить эту задачу, используя незащищенный канал связи.

В предложенной ими системе открытого распределения ключей [Диф79] каждая из сторон изначально имеет свой секретный параметр. Стороны реализуют определенный протокол взаимодействия по открытому каналу связи. При этом они обмениваются некоторыми сообщениями (образованными с помощью своих секретных параметров) и по результатам этого обмена вычисляют общий секретный связной ключ. В более поздних работах такие протоколы стали называть *протоколами выработки общего ключа*, поскольку изначально ни одна из сторон не имеет ключа и как такового распределения или пересылки ключей в нем не происходит.

В исходном виде система Диффи и Хеллмана имела существенные недостатки, связанные с возможностью для третьей стороны по осуществлению активного вхождения в канал связи и проведению полного контроля передаваемой информации. Однако после небольших модификаций и дополнений их протокол уже позволяет осуществлять не только

выработку общего ключа, но и одновременно проверять и подтверждать правильность вычислений, а также проводить взаимную аутентификацию взаимодействующих сторон.

Схема разделения секрета

Еще одной задачей современной криптографии, тесно связанной с проблемой распределения ключей и активно развивающейся в последние годы, является задача построения *схем разделения секрета*. Для многих практически важных приложений, связанных с запуском или активизацией критических процессов или определяющих порядок получения доступа к значимым данным, ответственное лицо должно ввести секретный ключ. Чтобы обезопасить процедуру принятия решения и не отдавать все на волю одного человека, являющегося обладателем ключа, используют метод разделения секрета. Он состоит в назначении определенной группы лиц, которая имеет право принимать решение. Каждый член группы владеет определенной долей секрета (точнее, специально выбранным набором данных), полная совокупность которых позволяет восстановить секретный ключ. При этом схема разделения секрета выбирается с таким условием, что для восстановления секретного ключа требуется обязательное присутствие всех членов группы, так как в случае отсутствия хотя бы одного из участников объединение долей оставшихся членов группы гарантированно не позволяет получить никакой информации о секретном ключе. Таким образом, *схема разделения секрета* определяется двумя алгоритмами, удовлетворяющими сформулированному выше условию: первый алгоритм определяет порядок вычисления значений долей по заданному значению секретного ключа, а второй предназначен для восстановления значения секрета по известным долям.

Задачу построения схемы разделения секрета можно обобщить

— либо путем введения так называемой *структуры доверия*, когда решение может приниматься не одной, а несколькими различными группами, причем часть из участников может наделяться правом “вето”,

— либо путем добавления механизмов, позволяющих обнаружить обман или сговор участников,

— либо введением специального протокола распределения долей между участниками с подтверждением правильности полученной информации и аутентификацией сторон.

§ 2.3. Инфраструктура открытых ключей

Сертификаты

Создание цифровой подписи позволило решить проблему *сертификации открытых ключей*. Она заключается в том, что перед тем как использовать открытый ключ некоторого абонента для отправки ему конфиденциального сообщения, отправитель должен быть уверен, что открытый ключ действительно принадлежит этому абоненту. Открытые ключи необходимо очень тщательно обезопасить, в том смысле, что если сервер, на котором они хранятся, не обеспечивает их целостность и аутентичность, то злоумышленник имеет возможность, подменив открытый ключ одного из абонентов, выступить от его имени. Поэтому для защиты открытых ключей создаются специальные *центры сертификации*, которые играют роль доверенной третьей стороны и заверяют открытые ключи каждого из абонентов своими цифровыми подписями.

Сертификат представляет собой набор данных, заверенный цифровой подписью центра и включающий открытый ключ и список дополнительных атрибутов, принадлежащих абоненту. К таким атрибутам относятся: имена пользователя и центра сертификации, номер сертификата, время действия сертификата, предназначение открытого ключа (цифровая подпись, шифрование) и т. д.

Международный стандарт ISO X.509 определяет общую структуру сертификатов открытых ключей и протоколы их использования для аутентификации в распределенных системах.

Центры сертификации

Центр сертификации предназначен для регистрации абонентов, изготовления сертификатов открытых ключей, хранения изготовленных сертификатов, поддержания в актуальном состоянии справочника действующих сертификатов и выпуска списка досрочно отозванных сертификатов.

Для сетей с большим числом абонентов создается несколько центров сертификации. Центры сертификации объединяются в древовидную структуру, в корне которой находится главный центр сертификации, который выдает сертификаты подчиненным ему отраслевым центрам, тем самым обеспечивая доверие к открытым ключам этих центров. Каждый центр вышестоящего уровня аналогичным образом делегирует право выпуска сертификатов подчиненным ему центрам. В результате доверие сертификату открытого ключа каждого центра основано на заверении его сертификата ключом вышестоящего центра. Сертификаты главного центра подписывает сам главный центр.

Зная иерархию и подчиненность друг другу центров сертификации, можно всегда точно установить, является ли абонент владельцем данного открытого ключа.

Основная трудность при создании центров сертификации заключается в их юридическом статусе и потенциальных финансовых возможностях по выплате компенсаций за ущерб, понесенный в результате невыполнения подписанных цифровыми подписями с использованием сертификатов, выданных этим центром, договоров и контрактов, сорванных по причине отказов от цифровых подписей или их подделки.

§ 2.4. Формальные модели шифров

Для того чтобы иметь возможность доказывать в криптографии точные результаты, нужны математические модели основных исследуемых объектов, к которым относятся в первую очередь шифр и открытый текст. Введем сначала алгебраическую модель шифра (шифрсистемы), предложенную, по сути дела, К. Шенноном [Шен63]. С моделями открытых текстов мы познакомимся ниже.

Как мы уже знаем, криптография защищает информацию с помощью *шифрования* — процедуры, использующей некое обратимое преобразование. При этом преобразование открытого текста в зашифрованный называется зашифрованием, а обратный процесс — расшифрованием. Шифрование предполагает наличие множества обратимых преобразований. Выбор преобразования из указанного множества для зашифрования данного сообщения осуществляется с помощью ключа. Имеется однозначное соответствие между множеством ключей и множеством преобразований.

Выбор ключа естественным образом определяет функцию (вообще говоря, многозначную), отображающую множество возможных открытых текстов в множество возможных зашифрованных текстов. Способ вычисления значения этой функции для произвольного аргумента будем называть *правилом зашифрования*. Выбранный ключ будем называть *ключом зашифрования*. Требование однозначности расшифрования определяет обратную функцию, отображающую множество возможных (при выбранном ключе) зашифрованных текстов в множество возможных открытых текстов. Способ вычисления значения этой функции для произвольного аргумента будем называть *правилом расшифрования*. Ключ, определяющий выбор правила расшифрования, будем называть *ключом расшифрования*.

Формализуем сказанное.

Пусть X, K, Y — конечные множества возможных открытых текстов, ключей и зашифрованных текстов соответственно; $E_k : X \rightarrow Y$ — правило зашифрования на ключе $k \in K$. Множество $\{E_k : k \in K\}$ обозначим через E , а множество $\{E_k(x) : x \in X\}$ — через $E_k(X)$. Пусть $D_k : E_k(X) \rightarrow X$ — правило расшифрования на ключе $k \in K$, и D — множество $\{D_k : k \in K\}$.

Здесь и далее мы будем предполагать, что если $k \in K$ представляется в виде $k = (k_z, k_p)$, где k_z — ключ зашифрования, а k_p — ключ расшифрования (причем $k_z \neq k_p$), то E_k понимается как функция E_{k_z} , а D_k — как функция D_{k_p} .

Определение 1. *Шифром (шифрсистемой) назовем совокупность*

$$\Sigma_A = (X, K, Y, E, D)$$

введенных множеств, для которых выполняются следующие свойства:

- 1) для любых $x \in X$ и $k \in K$ выполняется равенство $D_k(E_k(x)) = x$;
- 2) $Y = \bigcup_{k \in K} E_k(X)$.

Неформально, шифр — это совокупность множеств возможных открытых текстов (то, что шифруется), возможных ключей (то, с помощью чего шифруется), возможных шифртекстов (то, во что шифруется), правил зашифрования и правил расшифрования.

Отметим, что условие 1) отвечает требованию однозначности расшифрования. Условие 2) означает, что любой элемент $y \in Y$ может быть представлен в виде $E_k(x)$ для подходящих элементов $x \in X$ и $k \in K$. Отметим также, что в

общем случае утверждение “для любых $k \in K$ и $y \in E_k(X)$ выполняется равенство $E_k(D_k(y)) = y$ ” является неверным.

Легко проверить, что из условия 1) следует свойство инъективности функции E_k . Другими словами, если $x_1, x_2 \in X$, причем $x_1 \neq x_2$, то при любом $k \in K$ выполняется неравенство $E_k(x_1) \neq E_k(x_2)$.

По сути дела определение 1 вводит математическую модель, отражающую основные свойства реальных шифров. В силу этого мы будем отождествлять реальный шифр с его моделью Σ_A , которую будем называть *алгебраической моделью шифра*. Для подавляющего большинства известных шифров несложно составить такую модель, как это будет видно из дальнейшего.

Введем теперь вероятностную модель шифра. Следуя К. Шеннону [Шен63], определим априорные распределения вероятностей $P(X)$, $P(K)$ на множествах X и K соответственно. Тем самым для любого $x \in X$ определена вероятность $p_X(x) \in P(X)$ и для любого $k \in K$ — вероятность $p_K(k) \in P(K)$, причем выполняются равенства

$$\sum_{x \in X} p_X(x) = 1 \quad \text{и} \quad \sum_{k \in K} p_K(k) = 1.$$

В тех случаях, когда нам требуется знание распределений $P(X)$ и $P(K)$, мы будем пользоваться *вероятностной моделью* Σ_B , состоящей из пяти множеств, связанных условиями 1) и 2) определения 1, и двух вероятностных распределений:

$$\Sigma_B = (X, K, Y, E, D, P(X), P(K)).$$

Забегая вперед, отметим, что вероятностные характеристики шифров используются лишь в *криптоанализе* — разде-

ле криптографии, посвященном решению задач *вскрытия* (или *взлома*) шифров.

В большинстве случаев множества X и Y представляют собой объединения декартовых степеней некоторых множеств A и B соответственно, так что для некоторых натуральных L и L_1

$$X = \bigcup_{i=1}^L A^i, \quad Y = \bigcup_{i=1}^{L_1} B^i.$$

Множества A и B называют соответственно *алфавитом открытого текста* и *алфавитом шифрованного текста*. Другими словами, открытые и шифрованные тексты записываются привычным образом в виде последовательностей букв.

Введем шифр простой замены в алфавите A .

Определение 2. Пусть $X = Y = \bigcup_{i=1}^L A^i$, $K \subseteq S(A)$, где

$S(A)$ — симметрическая группа подстановок множества A . Для любого ключа $k \in K$, открытого текста $x = (x_1, \dots, x_l)$ и шифрованного текста $y = (y_1, \dots, y_l)$ правила зашифрования и расшифрования шифра простой замены в алфавите A определяются формулами:

$$\begin{aligned} E_k(x) &= (k(x_1), \dots, k(x_l)), \\ D_k(y) &= (k^{-1}(y_1), \dots, k^{-1}(y_l)), \end{aligned} \tag{1}$$

где k^{-1} — подстановка, обратная к k .

В более общей ситуации для шифра простой замены

$X = \bigcup_{i=1}^L A^i$, $Y = \bigcup_{i=1}^L B^i$, причем $|A| = |B|$, а K представляет

собой множество биекций множества A на множество B . Правила зашифрования и расшифрования определяются для

$k \in K, x \in X, y \in Y$ (и обратной к k биекции k^{-1}) формулами (1).

Определим еще один шифр, называемый *шифром перестановки*.

Определение 3. Пусть $X = Y = A^L$ и пусть $K \subseteq S_L$, где S_L — симметрическая группа подстановок множества $\{1, 2, \dots, L\}$. Для любого ключа k , открытого текста $x = (x_1, \dots, x_L)$ и зашифрованного текста $y = (y_1, \dots, y_L)$ правила зашифрования и расшифрования шифра перестановки определяются формулами:

$$E_k(x) = (x_{k(1)}, \dots, x_{k(L)}), \quad D_k(y) = (y_{k^{-1}(1)}, \dots, y_{k^{-1}(L)}),$$

где k^{-1} — подстановка, обратная к k .

Шифры, введенные определениями 2 и 3, являются представителями двух наиболее важных классов симметричных шифров, а именно *шифров замены* и *шифров перестановки*, которые будут подробно рассматриваться ниже. Другими симметричными шифрами являются композиции (или последовательные применения) некоторых шифров замены и шифров перестановки.

Приведем пример асимметричного шифра.

В следующем определении *шифра RSA* мы будем пользоваться общепринятыми в алгебре терминологией и обозначениями (см. Приложение 3).

Определение 4. Пусть $n = pq$, где p и q — простые числа. Пусть $X = Y = Z_n$ — кольцо вычетов по модулю n . Положим

$$K = \{(n, p, q, a, b) : a, b \in Z_n, n = pq, ab \equiv 1 \pmod{\phi(n)}\},$$

где ϕ — функция Эйлера. Представим ключ $k \in K$ в виде $k = (k_3, k_p)$, где $k_3 = (n, b)$ и $k_p = (n, p, q, a)$ — ключи зашифрования и расшифрования соответственно. Правила зашифрования и расшифрования шифра RSA определим для $x \in X$ и $y \in Y$ формулами:

$$E_{k_3}(x) = x^b \bmod n, \quad D_{k_p}(y) = y^a \bmod n. \quad (2)$$

Аббревиатура RSA определяется начальными буквами фамилий создателей этого шифра — Rivest, Shamir, Adleman. Корректность формул (2) следует из малой теоремы Ферма (подробнее это сделано в § 11.1).

Введенные определения и термины не исчерпывают полный перечень необходимых нам понятий, которые будут вводиться далее по необходимости.

§ 2.5. Модели открытых текстов

Введенная нами математическая модель шифра Σ_B (см. § 2.4) содержит вероятностные распределения $P(X)$ и $P(K)$ на множествах открытых текстов и ключей соответственно. Если $P(K)$ определяется свойствами устройств, служащих для генерации ключей (которые могут быть случайными или псевдослучайными), то $P(X)$ определяется частотными характеристиками самих текстов, подлежащих шифрованию. Характер таких текстов может быть различный: это могут быть обычные литературные тексты, формализованные данные межмашинного обмена и т. д. Так или иначе, открытые тексты обладают многими закономерностями, некоторые из которых наследуются шифрованными текстами. Именно это является определяющим фактором, влияющим на надежность шифрования.

Математические модели открытого текста

Потребность в математических моделях открытого текста продиктована, прежде всего, следующими соображениями. Во-первых, даже при отсутствии ограничений на временные и материальные затраты по выявлению закономерностей, имеющих место в открытых текстах, нельзя гарантировать того, что такие свойства указаны с достаточной полнотой. Например, хорошо известно, что частотные свойства текстов в значительной степени зависят от их характера. Поэтому при математических исследованиях свойств шифров прибегают к упрощающему моделированию, в частности, реальный открытый текст заменяется его моделью, отражающей наиболее важные его свойства. Во-вторых, при автоматизации методов криптоанализа, связанных с перебором ключей, требуется “научить” ЭВМ отличать открытый текст от случайной последовательности знаков. Ясно, что соответствующий критерий может выявить лишь адекватность последовательности знаков некоторой модели открытого текста.

Один из естественных подходов к моделированию открытых текстов связан с учетом их частотных характеристик, приближения для которых можно вычислить с нужной точностью, исследуя тексты достаточной длины (см. Приложение 1). Основанием для такого подхода является устойчивость частот k -грамм или целых словоформ реальных языков человеческого общения (то есть отдельных букв, слогов, слов и некоторых словосочетаний). Основанием для построения модели может служить также и теоретико-информационный подход, развитый в работах К. Шеннона [Шен63].

Учет частот k -грамм приводит к следующей модели открытого текста. Пусть $P^{(k)}(A)$ представляет собой массив, состоящий из приближений для вероятностей $p(b_1 b_2 \dots b_k)$ появления k -грамм $b_1 b_2 \dots b_k$ в открытом тексте, $k \in \mathbb{N}$, $A = \{a_1, \dots, a_n\}$ — алфавит открытого текста, $b_i \in A$, $i = \overline{1, k}$.

Тогда источник “открытого текста” генерирует последовательность $c_1, c_2, \dots, c_k, c_{k+1}, \dots$ знаков алфавита A , в которой k -грамма $c_1 c_2 \dots c_k$ появляется с вероятностью $p(c_1 c_2 \dots c_k) \in P^{(k)}(A)$, следующая k -грамма $c_2 c_3 \dots c_{k+1}$ появляется с вероятностью $p(c_2 c_3 \dots c_{k+1}) \in P^{(k)}(A)$ и т. д. Назовем построенную модель открытого текста *вероятностной моделью k -го приближения*.

Таким образом, простейшая модель открытого текста – *вероятностная модель первого приближения* – представляет собой последовательность знаков c_1, c_2, \dots , в которой каждый знак $c_i, i = 1, 2, \dots$, появляется с вероятностью $p(c_i) \in P^{(1)}(A)$, независимо от других знаков. Будем называть также эту модель *позначной моделью открытого текста*. В такой модели открытый текст $c_1 c_2 \dots c_l$ имеет вероятность

$$p(c_1 c_2 \dots c_l) = \prod_{i=1}^l p(c_i).$$

В вероятностной модели второго приближения первый знак c_1 имеет вероятность $p(c_1) \in P^{(1)}(A)$, а каждый следующий знак c_i зависит от предыдущего и появляется с вероятностью

$$p(c_i / c_{i-1}) = \frac{p(c_{i-1} c_i)}{p(c_{i-1})},$$

где $p(c_{i-1} c_i) \in P^{(2)}(A)$, $p(c_{i-1}) \in P^{(1)}(A)$, $i = 2, 3, \dots$. Другими словами, модель открытого текста второго приближения представляет собой *простую однородную цепь Маркова*. В такой модели открытый текст $c_1 c_2 \dots c_l$ имеет вероятность

$$p(c_1 c_2 \dots c_l) = p(c_1) \cdot \prod_{i=2}^l p(c_i / c_{i-1}).$$

Модели открытого текста более высоких приближений учитывают зависимость каждого знака от большего числа предыдущих знаков. Ясно, что, чем выше степень приближения, тем более “читаемыми” являются соответствующие модели. Проводились эксперименты по моделированию открытых текстов с помощью ЭВМ.

Приведем примеры “открытых текстов”, выработанных компьютером на основе частотных характеристик (алфавита со знаком пробела) собрания сочинений Р. Желязны объемом 10652970 байтов:

1. (Позначная модель) *ались проситете пригнуть стречи разве возникл;*
2. (Второе приближение) *и умере данного отствию официант простояло его то;*
3. (Третье приближение) *уэт быть как ты хоть а что я смящихся фигурой куда п;*
4. (Четвертое приближение) *ество что ты и мы сдохнуть пересовались ярким сторож;*
5. (Пятое приближение) *луну него словно него словно из ты в его не полагаете помощи я д;*
6. (Шестое приближение) *о разведения которые звенел в тонкостью огнем только.*

Как видим, тексты вполне “читаемы”.

Отметим, что с более общих позиций открытый текст может рассматриваться как реализация *стационарного эргодического случайного процесса с дискретным временем и конечным числом состояний* [Гне88].

Критерии распознавания открытого текста

Заменяв реальный открытый текст его моделью, мы можем теперь построить критерий распознавания открытого текста. При этом можно воспользоваться либо стандартными методами различения статистических гипотез, либо наличием в открытых текстах некоторых запретов, таких, например, как биграмма ЪЪ в русском тексте. Проиллюстрируем первый подход при распознавании позначной модели открытого текста.

Итак, согласно нашей договоренности, открытый текст представляет собой реализацию независимых испытаний случайной величины, значениями которой являются буквы алфавита $A = \{a_1, \dots, a_n\}$, появляющиеся в соответствии с распределением вероятностей $P(A) = (p(a_1), \dots, p(a_n))$. Требуется определить, является ли случайная последовательность $c_1 c_2 \dots c_l$ букв алфавита A открытым текстом или нет.

Пусть H_0 — гипотеза, состоящая в том, что данная последовательность — открытый текст, H_1 — альтернативная гипотеза. В простейшем случае последовательность $c_1 c_2 \dots c_l$ можно рассматривать при гипотезе H_1 как случайную и равновероятную. Эта альтернатива отвечает субъективному представлению о том, что при расшифровании криптограммы с помощью ложного ключа получается “бессмысленная” последовательность знаков. В более общем случае можно считать, что при гипотезе H_1 последовательность $c_1 c_2 \dots c_l$ представляет собой реализацию независимых испытаний некоторой случайной величины, значениями которой являются буквы алфавита $A = \{a_1, \dots, a_n\}$, появляющиеся в соответствии с распределением вероятностей $Q(A) = (q(a_1), \dots, q(a_n))$. При таких договоренностях можно применить, например, *наибо-*

лее мощный критерий различения двух простых гипотез, который дает лемма Неймана—Пирсона [Кра75]

В силу своего вероятностного характера такой критерий может совершать ошибки двух родов. Критерий может принять открытый текст за случайный набор знаков. Такая ошибка обычно называется *ошибкой первого рода*, ее вероятность равна $\alpha = p\{H_1/H_0\}$. Аналогично вводится *ошибка второго рода* и ее вероятность $\beta = p\{H_0/H_1\}$. Эти ошибки определяют качество работы критерия. В криптографических исследованиях естественно минимизировать вероятность ошибки первого рода, чтобы не “пропустить” открытый текст. Лемма Неймана—Пирсона при заданной вероятности первого рода минимизирует также вероятность ошибки второго рода.

Критерии на открытый текст, использующие запретные сочетания знаков, например k -граммы подряд идущих букв, будем называть *критериями запретных k -грамм*. Они устроены чрезвычайно просто. Отбирается некоторое число s редких k -грамм, которые объявляются запретными. Теперь, просматривая последовательно k -грамму за k -граммой анализируемой последовательности $c_1c_2\dots c_l$, мы объявляем ее случайной, как только в ней встретится одна из запретных k -грамм, и открытым текстом в противном случае. Такие критерии также могут совершать ошибки в принятии решения. В простейших случаях их можно рассчитать. Несмотря на свою простоту, критерии запретных k -грамм являются весьма эффективными.

Контрольные вопросы

1. Чем отличаются подходы к обеспечению безопасности информации в криптографии и в методах сокрытия информации?

2. Какими методами обеспечивается конфиденциальность информации?
3. Что такое целостность информации?
4. Для каких аспектов информационного взаимодействия необходима аутентификация?
5. Какие средства используются для обеспечения невозможности отказа от авторства?
6. В чем суть предварительного распределения ключей?
7. В чем разница между обычным и открытым распределениями ключей?
8. Для чего нужны схемы разделения секрета?
9. Что такое сертификат открытого ключа?
10. Каковы функции центра сертификации ключей?
11. Чем отличаются алгебраическая и вероятностная модели шифра?
12. С какими целями в криптографии вводят модели открытых текстов?
13. Как подсчитать вероятность данного открытого текста в модели первого приближения?
14. Какие подходы используются для распознавания открытых текстов?
15. Какая идея воплощена в расположении клавиш на клавиатуре пишущей машинке, компьютера, логотипа?

Глава 3

Классификация шифров по различным признакам

В качестве первичного признака, по которому производится классификация шифров, используется тип преобразования, осуществляемого с открытым текстом при шифровании. Если фрагменты открытого текста (отдельные буквы или группы букв) заменяются некоторыми их эквивалентами в шифртексте, то соответствующий шифр относится к классу *шифров замены*. Если буквы открытого текста при шифровании лишь меняются местами друг с другом, то мы имеем дело с *шифром перестановки*. С целью повышения надежности шифрования шифрованный текст, полученный применением некоторого шифра, может быть еще раз зашифрован с помощью другого шифра. Все возможные такие композиции различных шифров приводят к третьему классу шифров, которые обычно называют *композиционными шифрами*. Заметим, что композиционный шифр может не входить ни в класс шифров замены, ни в класс шифров перестановки. В результате получаем первый уровень классификации шифров (см. рис. 8).

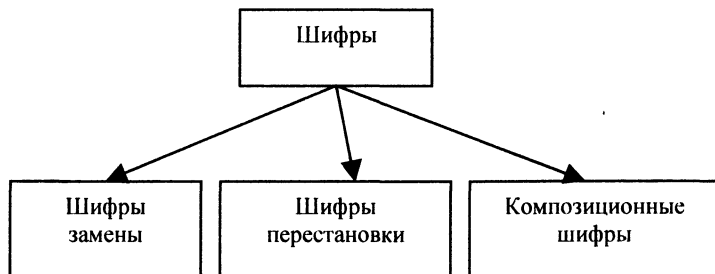


Рис. 8

§ 3.1. Математическая модель шифра замены

Определим модель $\Sigma_A = (X, K, Y, E, D)$ произвольного шифра замены. Будем считать, что открытые и зашифрованные тексты являются словами в алфавитах A и B соответственно: $X \subset A^*$, $Y \subset B^*$, $|A| = n$, $|B| = m$. Здесь и далее C^* обозначает множество слов конечной длины в алфавите C .

Перед зашифрованием открытый текст предварительно представляется в виде последовательности подслов, называемых *шифрвеличинами*. При зашифровании шифрвеличины заменяются некоторыми их эквивалентами в шифртексте, которые назовем *шифробозначениями*. Как шифрвеличины, так и шифробозначения представляют собой слова из A^* и B^* соответственно.

Пусть $U = \{u_1, \dots, u_N\}$ — множество возможных шифрвеличин, $V = \{v_1, \dots, v_M\}$ — множество возможных шифробозначений. Эти множества должны быть такими, чтобы любые тексты $x \in X, y \in Y$ можно было представить словами из U^*, V^* соответственно. Требование однозначности расшифрования влечет неравенства $N \geq n$, $M \geq m$, $M \geq N$.

Для определения правила зашифрования $E_k(x)$ в общем случае нам понадобится ряд обозначений и понятие *распределителя*, который, по сути, и будет выбирать в каждом такте шифрования замену соответствующей шифрвеличине.

Поскольку $M \geq N$, множество V можно представить в виде объединения $V = \bigcup_{i=1}^N V^{(i)}$ непересекающихся непустых подмножеств $V^{(i)}$. Рассмотрим произвольное семейство, состоящее из r таких разбиений множества V :

$$V = \bigcup_{i=1}^N V_{\alpha}^{(i)}, \alpha = \overline{1, r}, r \in \mathbb{N},$$

и соответствующее семейство биекций

$$\varphi_\alpha : U \rightarrow \{V_\alpha^{(1)}, \dots, V_\alpha^{(N)}\},$$

для которых $\varphi_\alpha(u_i) = V_\alpha^{(i)}$, $i = \overline{1, N}$.

Рассмотрим также произвольное отображение $\psi : K \times N \rightarrow N_r^*$, где $N_r = \{1, 2, \dots, r\}$, такое, что для любых $k \in K, l \in N$

$$\psi(k, l) = \alpha_1^{(k)} \dots \alpha_l^{(k)}, \quad \alpha_j^{(k)} \in N_r, \quad j = \overline{1, l}. \quad (1)$$

Назовем последовательность $\psi(k, l)$ *распределителем*, отвечающим данным значениям $k \in K, l \in N$.

Теперь мы сможем определить правило зашифрования произвольного шифра замены. Пусть

$$x \in X, x = x_1 \dots x_l, x \in U, i = \overline{1, l}; k \in K$$

и $\psi(k, l) = \alpha_1^{(k)} \dots \alpha_l^{(k)}$. Тогда $E_k(x) = y$, где $y = y_1 \dots y_l$,

$$y_j \in \varphi_{\alpha_j^{(k)}}(x_j), j = \overline{1, l}. \quad (2)$$

В качестве y_j можно выбрать любой элемент множества $\varphi_{\alpha_j^{(k)}}(x_j)$. Всякий раз при шифровании этот выбор можно производить случайно, например, с помощью некоторого *рандомизатора* типа игровой рулетки. Подчеркнем, что такая многозначность при зашифровании не препятствует расшифрованию, так как $V_\alpha^{(i)} \cap V_\alpha^{(j)} = \emptyset$ при $i \neq j$.

§ 3.2. Классификация шифров замены

Если ключ зашифрования совпадает с ключом расшифрования: $k_z = k_p$, то (как уже указывалось в гл. 2) такие шифры называют *симметричными*, если же $k_z \neq k_p$ — *асимметричными*.

В связи с указанным различием в использовании ключей сделаем еще один шаг в классификации (см. рис. 9).

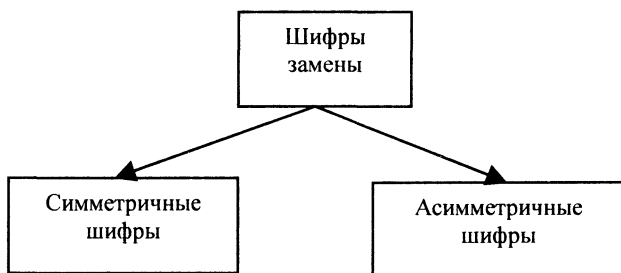


Рис. 9

Отметим также, что в приведенном определении правило зашифрования $E_k(x)$ является, вообще говоря, *многозначной функцией*. Выбор ее значений представляет собой некоторую проблему, которая делает многозначные функции $E_k(x)$ не слишком удобными для использования. Избавиться от этой проблемы позволяет использование однозначных функций, что приводит к естественному разделению всех шифров замены на *однозначные* и *многозначные замены* (называемых также в литературе *омофонами*) (см. рис. 10).

Для однозначных шифров замены справедливо свойство:

$$\forall \alpha, i: |V_{\alpha}^{(i)}| = 1;$$

для многозначных шифров замены:

$$\exists \alpha, i: |V_{\alpha}^{(i)}| > 1.$$

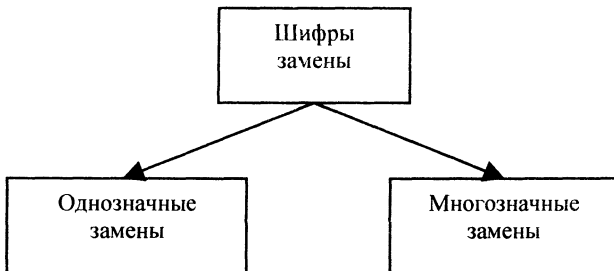


Рис. 10

Исторически известный шифр — *пропорциональной замены* представляет собой пример шифра многозначной замены, *шифр гаммирования* — пример шифра однозначной замены. Далее мы будем заниматься в основном изучением однозначных замен, получивших наибольшее практическое применение. Итак, далее $M = N$ и $\varphi_{\alpha}(u_i) = v_{\alpha}^{(i)}$, $i = \overline{1, M}$.

Заметим, что правило зашифрования E_k естественно “рассматривается как” отображение, $\tilde{E}_k: U^* \rightarrow V^*$.

В силу инъективности (по k) отображения E_k и того, что $|U| = |V|$, введенные в общем случае отображения φ_{α} являются биекциями $\varphi_{\alpha}: U \leftrightarrow V$, определенными равенствами

$\varphi_\alpha(u_i) = v_\alpha^{(i)}$, $i = \overline{1, N}$, $\alpha = \overline{1, r}$. Число таких биекций не превосходит $N!$

Для шифра однозначной замены определение правила зашифрования можно уточнить: в формуле (2) включение следует заменить равенством

$$y_j = \varphi_{\alpha^{(j)}}(x_j), \quad j = \overline{1, l}. \quad (2')$$

Введем еще ряд определений.

Если для некоторого числа $q \in \mathbb{N}$ выполняются включения $v_i \in B^q$, $i = \overline{1, N}$, то соответствующий шифр замены будем называть *шифром равнозначной замены*. В противном случае — *шифром разнзначной замены* (см. рис. 11).

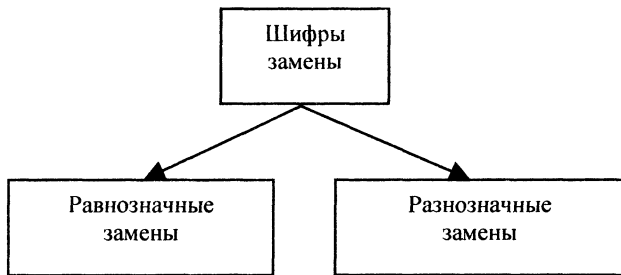


Рис. 11

В подавляющем большинстве случаев используются шифры замены, для которых $U \in A^p$, для некоторого $p \in \mathbb{N}$. При $p = 1$ говорят о *поточных шифрах замены*, при $p > 1$ — о *блочных шифрах замены* (см. рис. 12).

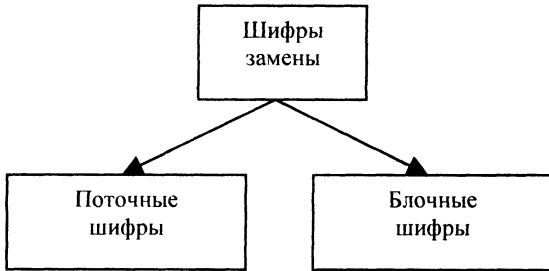


Рис. 12

Относительно деления шифров на поточные и блочные подробнее см. в гл. 8.

Следующее определение. В случае $r = 1$ шифр замены называют *одноалфавитным шифром замены* или *шифром простой замены*. В противном случае — *многоалфавитным шифром замены* (см. рис. 13).

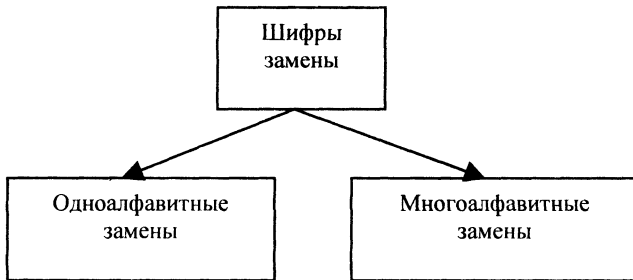


Рис. 13

Ограничиваясь наиболее важными классами шифров замены и исторически известными классами шифров перестановки, сведем результаты классификации в схему, изображенную на рис. 14.

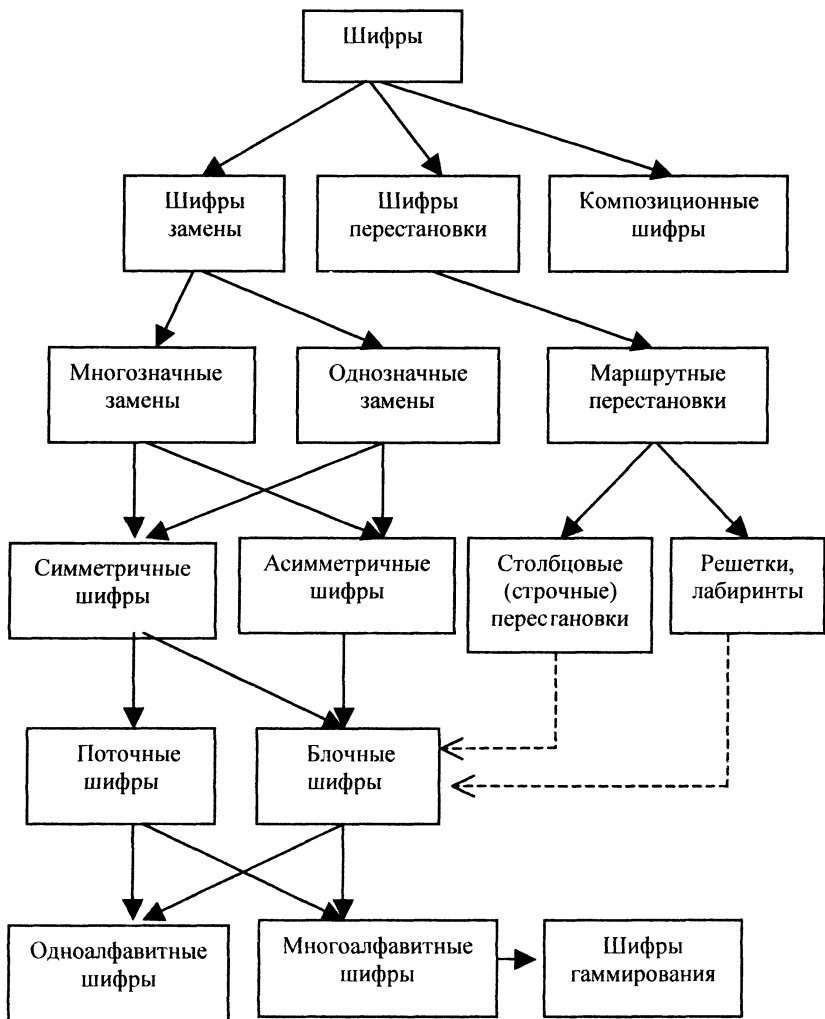


Рис. 14

Прокомментируем приведенную схему. Подчеркнем, что стрелки, выходящие из любого прямоугольника схемы, ука-

зывают лишь на наиболее значимые частные подклассы шифров. Пунктирные стрелки, ведущие из подклассов шифров перестановки, означают, что эти шифры можно рассматривать и как блочные шифры замены в соответствии с тем, что открытый текст делится при шифровании на блоки фиксированной длины, в каждом из которых производится некоторая перестановка букв. Одноалфавитные и многоалфавитные шифры могут быть как поточными, так и блочными. В то же время шифры гаммирования, образующие подкласс многоалфавитных шифров, относятся к поточным, а не к блочным шифрам. Кроме того, они являются симметричными, а не асимметричными шифрами.

Далее мы рассмотрим свойства большинства из указанных на схеме классов шифров.

Контрольные вопросы

1. С какими примерами шифров замены и перестановки Вы познакомились в историческом обзоре?
2. Существуют ли шифры, не являющиеся ни шифрами замены, ни шифрами перестановки?
3. Приведите пример шифра многозначной замены.
4. Может ли блочный шифр быть шифром разнозначной замены?
5. Может ли шифр простой замены быть равнозначным, разнозначным, блочным шифром?
6. В каком случае шифр гаммирования является одноалфавитным шифром?
7. Каково максимальное число простых замен, из которых может состоять многоалфавитный шифр?
8. Можно ли рассматривать множество возможных открытых и шифрованных текстов как множество шифровеличин и шифрообозначений шифра замены?

Глава 4

Шифры перестановки

В историческом обзоре упоминались некоторые типы шифров перестановки. Среди них — шифр *Сцитала*, *атбаи*, *поворотная решетка Кардано*. В самом общем виде шифр перестановки определен в гл. 3. Ключом шифра является перестановка номеров букв открытого текста. Зависимость ключа от длины текста создает значительные неудобства в использовании шифра. В силу этого был предложен ряд частных шифров перестановок, которые можно применять для зашифрования текстов любой длины.

§ 4.1. Маршрутные перестановки

Широкое применение получили так называемые *маршрутные перестановки*, основанные на некоторой геометрической фигуре. Отрезок открытого текста записывается в такую фигуру по некоторой траектории. Шифрованным текстом является последовательность, полученная при выписывании текста по другой траектории. Например, можно записывать сообщение в прямоугольную таблицу, выбрав такой маршрут: будем двигаться по горизонтали, начиная с левого верхнего угла, поочередно слева направо и справа налево. Списывать же сообщение будем по другому маршруту: по вертикалям, начиная с верхнего правого угла и двигаясь поочередно сверху вниз и снизу вверх.

Пример (*маршрутной перестановки*)

Зашифруем указанным выше способом фразу *пример маршрутной перестановки*, используя прямоугольную таблицу размером 4×7 :

п	р	и	м	е	р	м
н	т	у	р	ш	р	а
о	й	п	е	р	е	с
и	к	в	о	н	а	т

Зашифрованная фраза выглядит следующим образом:

мастаерреиршноермвпуиртикионп

Обращение описанных шагов при расшифровании не представляет труда.

Широкое распространение получила разновидность маршрутной перестановки, называемая *вертикальной перестановкой*. В этой системе также используется прямоугольная таблица, в которую сообщение записывается обычным образом (по строкам слева направо). Выписывается же сообщение по вертикалям (сверху вниз), при этом столбцы выбираются в порядке, определяемом *числовым ключом*.

Пример (*вертикальной перестановки*)

Зашифруем фразу *вот пример шифра вертикальной перестановки*, используя прямоугольник размером 6×7 и числовой ключ (5,1,4,7,2,6,3).

5	1	4	7	2	6	3
в	о	т	п	р	и	м
е	р	ш	и	ф	р	а
в	е	р	т	и	к	а
л	ь	н	о	й	п	е
р	е	с	т	а	н	о
в	к	и				

Отметим, что нецелесообразно заполнять последнюю строку прямоугольника “нерабочими” буквами, так как это дало бы противнику, получившему в свое распоряжение данную криптограмму, сведения о длине числового ключа. В самом деле, в этом случае длину ключа следовало бы искать среди делителей длины сообщения.

Теперь, выписывая буквы по столбцам в порядке, указанном числовым ключом, получим такую криптограмму:

орьекрфийамааеотирнсивевлрвиркннитот

При расшифровании, в первую очередь, надо определить число длинных столбцов, то есть число букв в последней строке прямоугольника. Для этого нужно разделить число букв в сообщении на длину числового ключа. Ясно, что остаток от деления и будет искомым числом. Когда это число определено, буквы криптограммы можно водворить на их собственные места, и сообщение будет прочитано естественным образом.

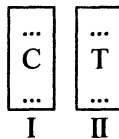
В нашем примере $38 = 7 \cdot 5 + 3$, поэтому в заполненной таблице имеется 3 длинных и 4 коротких столбца. Согласно числовому ключу, начальные буквы криптограммы берутся из второго (по счету слева) столбца, он — длинный (так как первые три столбца — длинные), поэтому первые шесть букв образуют второй столбец. Следующие пять букв образуют пятый столбец (он — короткий). И так далее.

Более сложные маршрутные перестановки могут использовать другие геометрические фигуры и более “хитрые” маршруты, как, например, при обходе шахматной доски “ходом коня”, пути в некотором лабиринте и т.п. Возможные варианты зависят от фантазии составителя системы и, конечно, естественного требования простоты ее использования.

§ 4.2. Элементы криптоанализа шифров перестановки

Укажем сначала основные идеи, используемые при вскрытии вертикальных перестановок.

Заметим прежде всего, что буквы каждого столбца заполненного прямоугольника выписываются в криптограмму подряд, то есть криптограмма разбивается на отрезки, являющиеся столбцами таблицы. Поэтому при дешифровании следует попытаться соединить две группы последовательных букв криптограммы так, чтобы они образовывали хорошие (“читаемые”) с точки зрения обычного текста комбинации. Для этого естественно использовать наиболее частые биграммы открытого текста, которые можно составить из букв рассматриваемого шифрованного текста. Если для первой пробы выбрано, скажем, сочетание СТ (самая частая биграмма русского языка), то мы можем по очереди приписывать к каждой букве С криптограммы каждую букву Т из нее. При этом несколько букв, стоящих до и после данной буквы С, и несколько букв, стоящих до и после данной буквы Т, соединяются в пары, то есть получают два столбца букв, записанные рядом:



Конечно, мы не знаем длины столбцов, но некоторые ограничения на них можно получить, используя положение конкретных букв. Так, столбцы должны иметь одинаковые длины или первый столбец может быть длиннее второго на одну букву, и тогда эта буква — последняя буква сообщения:



Если приписываемые друг к другу буквы разделены, скажем, только двумя буквами, то, как легко видеть, мы можем составить в соседних столбцах не более трех пар, и длина каждого столбца не превышает четырех. Кроме того, ограничением может послужить появление *запретной биграммы* (например, гласная — мягкий знак):



Для выбранного сочетания СТ получается по одной паре столбцов для каждого конкретного выбора букв С и Т из криптограммы, и из них целесообразно отобрать ту пару, которая содержит наиболее частые биграммы.

Заметим, что при автоматизации этого процесса можно приписать каждой биграмме вес, равный частоте ее появления в открытом тексте. Тогда целесообразно отобрать ту пару столбцов, которая имеет наибольший вес. Кстати, появление одной биграммы с низкой частотой может указать на то, что длину столбца надо ограничить.

Выбрав пару столбцов, мы аналогичным образом можем подобрать к ним третий (справа или слева) и т. д. Описанная процедура значительно упрощается при использовании *вероятных слов*, то есть слов, которые могут встретиться в тексте с большой вероятностью.

Рассмотрим также метод, применимый к любым шифрам перестановки. Допустим, что к двум или более сообщениям (или

отрезкам сообщений) одинаковой длины применяется один и тот же шифр перестановки. Тогда очевидно, что буквы, которые находились на одинаковых местах в открытых текстах, окажутся на одинаковых местах и в зашифрованных текстах.

Выпишем зашифрованные сообщения одно под другим так, что первые буквы всех сообщений оказываются в первом столбце, вторые — во втором и т. д. Если предположить, что две конкретные буквы в одном из сообщений идут одна за другой в открытом тексте, то буквы, стоящие на тех же местах в каждом из остальных сообщений, соединяются подобным же образом. Значит, они могут служить проверкой правильности первого предположения, подобно тому как комбинации, которые дают два столбца в системе вертикальной перестановки, позволяют проверить, являются ли соседними две конкретные буквы из этих столбцов. К каждому из указанных двухбуквенных сочетаний можно добавить третью букву для образования триграммы и т. д. Если располагать не менее чем четырьмя сообщениями одинаковой длины, то можно с уверенностью гарантировать их вскрытие подобным образом.

Контрольные вопросы

1. Приведите пример шифра перестановки, который может рассматриваться и как блочный шифр замены.
2. Как определить по криптограмме, полученной с помощью шифра вертикальной перестановки, число коротких столбцов заполненного открытым текстом основного прямоугольника?
3. Какие свойства открытого текста используются при вскрытии шифра вертикальной перестановки?
4. Сколько ключей имеет шифр поворотной решетки (см. главу 1). На основе прямоугольника размером $m \times n$?
5. Каким образом можно использовать вероятные слова для вскрытия ряда криптограмм, полученных на одном ключе шифра перестановки?

Глава 5

Шифры замены

Мы будем рассматривать лишь однозначные замены (см. главу 3), для которых правило зашифрования является обычной однозначной функцией. Одноалфавитные однозначные замены обычно называют *шифрами простой замены*.

§ 5.1. Поточные шифры простой замены

Наибольшее распространение получили поточные шифры простой замены, множества шифрвеличин и шифробозначений которых совпадают с алфавитом открытого текста A . Как указывалось в главе 2, ключом такого шифра является подстановка k на множестве A , верхняя строка которой представляет собой естественную последовательность букв алфавита, а нижняя — систематически перемешанную или случайную последовательность букв из A (см. Приложение 1).

Помимо явного задания (в виде двустрочной записи) ключ может быть задан некоторой формулой, как, например, для определяемого ниже *шифра Цезаря* (который иногда называют также *сдвиговым шифром*) и *аффинного шифра*. При использовании этих шифров буквы алфавита A удобно отождествлять с их порядковыми номерами, так что, например, для латинского алфавита

$$a \equiv 0, b \equiv 1, \dots, z \equiv 25.$$

Шифр Цезаря

$$X = Y = \bigcup_{i=1}^L Z_{26}^i, K = Z_{26}. \quad \text{Для } x = (x_1, \dots, x_l), y = (y_1, \dots, y_l),$$

$k \in K$ полагаем

$$y = E_k(x) = (x_1 + k, \dots, x_l + k),$$

$$x = D_k(y) = (y_1 + (26 - k), \dots, y_l + (26 - k)),$$

где $+$ и \cdot — операции кольца вычетов Z_{26} .

Аффинный шифр

$$X = Y = \bigcup_{i=1}^L Z'_{26}, K = Z_{26}^* \times Z_{26}. \text{ Для } k = (\alpha, \beta) \in K, \alpha \neq 0,$$

$x = (x_1, \dots, x_l), y = (y_1, \dots, y_l)$, полагаем

$$y = E_k(x) = (\alpha \cdot x_1 + \beta, \dots, \alpha \cdot x_l + \beta),$$

$$x = D_k(y) = ((y_1 + (26 - \beta)) \cdot \alpha^{-1}, \dots, (y_l + (26 - \beta)) \cdot \alpha^{-1}),$$

где $+$ и \cdot — операции кольца Z , а α^{-1} — элемент из мультипликативной группы Z_{26}^* , обратный к α .

Пример

Зашифруем слово CRYPTOGRAPHY с помощью аффинного шифра, полагая $k = (3, 5)$. Данный ключ индуцирует следующую подстановку на Z :

0	1	2	3	4	5	6	7	8	9	10	11	12
5	8	11	14	17	20	23	0	3	6	9	12	15

13	14	15	16	17	18	19	20	21	22	23	24	25
18	21	24	1	4	7	10	13	16	19	22	25	2

Если декодировать числа в буквы, то получим следующее соответствие для букв:

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

Слову CRYPTOGRAPHY соответствует числовая последовательность $x = (2, 17, 24, 15, 19, 14, 9, 17, 0, 15, 7, 24)$. Зашифровать открытый текст мы можем двумя способами. Во-первых, можно воспользоваться полученной подстановкой, заменяя каждую букву слова (найденную в верхней строке) ее образом в нижней строке: LEZYKVXEFYAZ. Во-вторых, можно вычислить значение функции зашифрования $E_k(x)$, исходя из ее определения:

$$\begin{aligned}
 y = E_k(x) &= (3 \cdot 2 + 5, 3 \cdot 17 + 5, 3 \cdot 24 + 5, 3 \cdot 15 + 5, \\
 &3 \cdot 19 + 5, 3 \cdot 14 + 5, 3 \cdot 9 + 5, 3 \cdot 17 + 5, \\
 &3 \cdot 0 + 5, 3 \cdot 15 + 5, 3 \cdot 7 + 5, 3 \cdot 24 + 5) = \\
 &= (11, 4, 25, 24, 10, 21, 23, 4, 5, 24, 0, 25).
 \end{aligned}$$

В буквенном эквиваленте y совпадает с полученным ранее зашифрованным текстом.

Для расшифрования y следует вычислить 3^{-1} в группе Z_{26}^* . Очевидно, что $3^{-1} = 9$. Теперь расшифруем y в соответствии с определением правила расшифрования:

$$\begin{aligned}
 x = D_k(y) &= ((11 + 21) \cdot 9, (4 + 21) \cdot 9, (25 + 21) \cdot 9, (24 + 21) \cdot 9, \\
 &(10 + 21) \cdot 9, (21 + 21) \cdot 9, (23 + 21) \cdot 9, (4 + 21) \cdot 9, (5 + 21) \cdot 9, \\
 &(24 + 21) \cdot 9, (0 + 21) \cdot 9, (25 + 21) \cdot 9) = \\
 &= (2, 17, 24, 15, 19, 14, 6, 17, 0, 15, 7, 24).
 \end{aligned}$$

Здесь мы воспользовались определением операций сложения и умножения в кольце Z_{26} , заменяя результат обычных целочисленных вычислений остатком от деления на 26.

В связи с рассмотрением аффинного шифра полезно напомнить один хорошо известный алгебраический результат.

Теорема. *Отображение $f: Z_n \rightarrow Z_n$, определяемое для фиксированных $a, b \in Z_n$ формулой*

$$f(\alpha) = a\alpha + b \pmod{n},$$

является биективным тогда и только тогда, когда $(a, n) = 1$.

До сих пор мы предполагали, что шифробозначениями являются отдельные знаки алфавита. Однако это вовсе не обязательное условие. Как отмечалось в гл. 3, имеются шифры *равнозначной замены* и шифры *разнозначной замены*. В первом случае все шифробозначения имеют одинаковые значности, например один, два и т. д. Во втором — разные значности, например, некоторые шифробозначения могут быть отдельными знаками, другие — состоять из пары или большего числа знаков. По соображениям экономии и скорости шифрования значность шифробозначений не должна быть большой. В большинстве известных примеров разнозначных шифров значность шифробозначений не превосходит двух. Приведем один из таких примеров [Вес97].

Пример (*шифра простой неравнозначной замены*)

Рассматривается прямоугольник размером 4×7 , в который записан систематически перемешанный английский алфавит (расширенный символами “.” и знаком раздела “/”), построенный на основе ключевого слова INCITATUS:

I	N	C	T	A	U	S
0	1	86	3	5	94	6
B	D	E	F	G	H	J
80	83	2	89	91	95	98
K	L	M	O	P	Q	R
81	84	87	4	92	96	7
V	W	X	Y	Z	.	/
82	85	88	90	93	97	99

Нумерация букв алфавита произведена по столбцам (сверху вниз), при этом восемь самых частых букв (A,E,I,N,O,R,S,T) занумерованы числами от 0 до 7, а остальные — двузначными числами от 80 до 99. Такую таблицу легко запомнить. Работать же удобнее с эквивалентной таблицей:

	0	1	2	3	4	5	6	7	8	9
	I	N	E	T	O	A	S	R	-	-
8	B	K	V	D	L	W	C	M	X	F
9	Y	G	P	Z	U	H	Q	.	J	/

При зашифровании открытый текст записывается со знаком пробела между словами. Точка, встретившаяся в тексте, считается отдельным словом. После этого производится замена шифрвеличин на шифробозначения согласно таблице, при этом цифровые данные не изменяются.

§ 5.2. Криптоанализ поточного шифра простой замены

Рассмотрим сначала простейший случай — однобуквенной замены.

Любой метод вскрытия шифра простой однобуквенной замены основан на том обстоятельстве, что с точностью до переобозначений частотные характеристики m -грамм шифртекста и открытого текста одинаковы. При этом существенно используются априорные частотные характеристики предполагаемого открытого текста, получаемые с учетом “характера переписки”. Как отмечено в Приложении 1, такие характеристики являются более “рельефными” для литературных текстов и менее “рельефными” для формализованных электронных текстов. Чем менее рельефно распределение знаков текста, тем сложнее задача вскрытия шифра простой замены. Для открытых текстов с “почти равномерным” распределением знаков эта задача становится практически не решаемой. Это следует учитывать и не питать иллюзий о простоте вскрытия простой замены, о которой часто упоминается в популярных книгах по защите информации. Методы “рандомизации” или “сжатия” открытых текстов, например, с использованием компьютерных архиваторов значительно усложняют задачу вскрытия шифра простой замены.

Как будет ясно из дальнейшего (гл. 7), рельефность диаграммы текста тесно связана с такой его важной теоретико-информационной характеристикой, как *избыточность*. Далее мы будем решать задачу вскрытия простой замены лишь при условии, что предполагаемые открытые тексты — это литературные тексты с “приличной” избыточностью. Кроме того, мы будем считать, что при дешифровании мы располагаем достаточно большим числом знаков шифртекста, чтобы опираться не на “фокусы”, использованные, например, в известных произведениях Э. А. По и А. Конан Дойля, а в большей степени на “статистику”.

Алгоритм вскрытия простой замены по тексту криптограммы достаточно сложно формализовать. При любой попытке формализации теряется какой-нибудь важный нюанс. Поэтому мы укажем лишь основные идеи, лежащие в основе

такого алгоритма. Обычно выделяют следующие этапы алгоритма:

Алгоритм 1

1. Подсчет частот встречаемости шифробозначений, а также некоторых их сочетаний, например биграмм и триграмм подряд идущих знаков.
2. Выявление шифробозначений, заменяющих гласные и согласные буквы.
3. Выдвижение гипотез о значениях шифробозначений и их проверка. Восстановление истинного значения шифробозначений.

Сделаем ряд замечаний и уточнений.

Если длина текста достаточно велика, то найденные на этапе 1 частоты окажутся близкими к табулированным значениям частот знаков (соответственно — биграмм или триграмм). Проведенная на этом этапе работа служит основанием для выдвижения гипотез о значениях шифрвеличин, соответствующих данным шифробозначениям. При этом учитывается, что каждая буква имеет группу предпочтительных связей (см. Приложение 1), которые составляют ее наиболее характерную особенность. Как правило, такие гипотезы подтверждаются не полностью. Хорошим критерием при этом является “читаемость” восстанавливаемого открытого текста. Выделение шифробозначений, отвечающих гласным и согласным, основано на характерных свойствах этих букв, указанных в Приложении 1. Добавим к ним следующие соображения (относимые к большинству европейских языков). Если шифробозначение часто встречается, равномерно располагается по шифртексту, в отдельных местах чередуется через 1, 2 или 3 знака, сочетается со средними и редкими (по частоте) шифробозначениями, то это дает основания полагать, что такое шифробозначение скрывает гласную букву. Удвоение гласных в открытом тексте происходит реже, чем согласных. Если некоторое шифробозначение признано гласной, то бук-

ва, часто сочетающаяся с ней, скорее всего согласная. В открытом тексте чрезвычайно редко встречаются три и более подряд идущие гласные. Четыре и более подряд идущие согласные также редки. Важно учитывать также процентное соотношение чисел гласных и согласных в открытом тексте.

При проверке гипотез о значениях шифробозначений полезен поиск в шифртексте слов с характерной структурой, которые часто встречаются в открытом тексте. Для русского языка — это, например, слова *сколько, которое, что* и т. п. Для английского языка — слова *every, that, look, the* и т. п. Такие слова выделяются в шифртексте посредством интервалов между повторяющимися частыми буквами, характерными сочетаниями гласных и согласных.

Если с помощью приведенных соображений произведено несколько идентификаций шифробозначений, то дальнейшая работа по вскрытию текста криптограммы не представляет особого труда.

Пример вскрытия шифра простой однобуквенной замены приведен в Приложении 2.

Задача дешифрования еще более упрощается, если известно, что использовался сдвиговой или аффинный шифр. Так, для аффинного шифра бывает достаточно идентифицировать лишь пару шифробозначений с тем, чтобы полностью восстановить открытый текст (см., например, [Сал96]).

Из наших рассмотрений становится понятным, что наиболее трудно формализуемым фрагментом алгоритма 1 является проверка выдвигаемых гипотез о значениях шифробозначений. Трудность состоит в формулировке критерия, подтверждающего или отвергающего ту или иную гипотезу. В статье [Jak95], посвященной этому вопросу, был предложен четко сформулированный критерий, основанный на “близости” матрицы биграмм $\Delta(t) = (\Delta_{ij}(t))_{n \times n}$ (где n — число букв алфавита) данного текста t эталонной матрице биграмм

$B = (b_{ij})_{n \times n}$ открытого текста. Приведем эту идею и основанный на ней эвристический алгоритм дешифрования.

Мерой близости служит следующая “целевая функция” $f(t)$, связывающая матрицы $\Delta(t)$ и B :

$$f(t) = \sum_{i,j} |\Delta_{ij}(t) - b_{ij}|. \quad (1)$$

Будем исходить из того естественного предположения, что если y — данная криптограмма и D_k — правило расшифрования на ключе k данного шифра простой замены, то для истинного ключа k_u значение

$$f(D_{k_u}(y)) = \sum_{i,j} |\Delta_{ij}(D_{k_u}(y)) - b_{ij}|$$

должно быть минимальным.

Идея основного шага алгоритма состоит в том, чтобы исходя из некоторого первичного “приближения” k для ключа k_u , основанного, например, на диаграмме частот букв, немного его изменять неким “разумным” способом, уменьшая значение целевой функции $f(t)$.

Приведем теперь формальное описание алгоритма [Jak95].

Алгоритм 2

1. Построить начальный вариант ключа k на основе сравнения частот знаков криптограммы и открытого текста.
2. Положить $v = f(D_k(y))$.
3. Положить $k' = k$.
4. Поменять местами в нижней строке подстановки k' некоторую пару букв, скажем α и β .
5. Положить $v' = f(D_{k'}(y))$.
6. Если $v' < v$, то положить $k = k'$, $v = v'$ и перейти к 4.

7. Перейти к шагу 3.

Алгоритм заканчивается, когда условие $v' < v$ не выполняется в течение некоторого числа итераций, например 100.

Переход на шаге 4 от k к k' , связанный с транспозицией пары символов, имеет под собой следующее основание. На шаге 5 вычисляется величина

$$\begin{aligned} v' &= f(D_{k'}(y)) = \sum_{i,j} |\Delta_{ij}(D_{k'}(y)) - b_{ij}| = \\ &= \sum_{i,j} |\Delta_{ij}^1(D_k(y)) - b_{ij}|, \end{aligned}$$

где $\Delta^1(t)$ — матрица, полученная из матрицы $\Delta(t)$ путем перестановки в ней столбцов с номерами α и β , а также строк с теми же номерами.

В силу отмеченного свойства на шаге 5 алгоритма не нужно проводить трудоемкую операцию вычисления матрицы биграмм $\Delta_y(D_{k'}(y))$ непосредственно по “расшифрованно-му” тексту $D_{k'}(y)$. Достаточно вычислить лишь матрицу $\Delta(D_k(y))$, а на следующих шагах алгоритма производить в ней одноименные перестановки строк и столбцов.

Выбор транспозиции (α, β) на шаге 4 можно производить, например, следующим естественным образом. Пусть $\bar{s} = (s_1, s_2, \dots, s_n)$ — вектор, образованный буквами криптограммы, упорядоченными по убыванию частот. Тогда последовательность транспозиций можно выбрать такой:

$$(s_1, s_2), (s_2, s_3), \dots, (s_{n-2}, s_{n-1}), (s_{n-1}, s_n),$$
$$(s_1, s_3), (s_2, s_4), \dots, (s_{n-2}, s_n),$$

.....

$$(s_1, s_n).$$

В [Jak95] показано, что алгоритм 2 является достаточно эффективным.

Отметим некоторые особенности вскрытия равнозначных и разнозначных шифров простой замены.

Если шифр простой замены не является однобуквенным, то при вскрытии криптограммы необходимо попытаться восстановить множество шифрвеличин. Если эта задача решена, то дальнейшая работа ничем не отличается от той, которую мы проделали для шифра однобуквенной простой замены.

Заметим, что в литературных открытых текстах часто встречаются повторения фрагментов, состоящих из трех и большего числа букв. При применении к тексту шифра простой замены соответствующие повторения остаются и в шифрованном тексте. Если в криптограмме встретилось несколько повторений, то их успешно можно использовать для определения значности шифробозначений.

Очевидно, что для равнозначного шифра простой замены длины повторений и расстояния между ними должны быть кратны значности шифра. Находя наибольший общий делитель этих чисел, мы с большой вероятностью получаем искомую значность. Некоторые сомнения в правильности определения значности помогает устранить подсчет общего числа шифробозначений. Если это число близко к ожидаемому числу шифробозначений (скажем, к числу букв алфавита), и диаграмма их повторяемости близка к табличной, то, скорее всего, значность определена верно.

Для разнозначного шифра дело обстоит несколько сложнее. В этом случае числа, равные длинам повторений и расстояниям между ними, скорее всего, взаимно просты в сово-

купности. Однако и для таких шифров задача определения множества шифробозначений не безнадежна. В этом помогает естественное ограничение, которым обычно пользуются при составлении таблицы шифробозначений. Оно связано с требованием однозначности расшифрования и заключается в том, чтобы ни одно из шифробозначений не являлось началом никакого другого шифробозначения (в теории кодирования в подобной ситуации говорят о *префиксном коде*). Если значность шифробозначений колеблется в незначительных пределах, то перебор сравнительно небольшого числа вариантов приводит (с учетом ограничения) к правильному определению большинства шифробозначений. Некоторые затруднения могут возникать лишь при определении значности шифробозначений, редко встречающихся в тексте. Как правило, эти проблемы решаются вместе с попытками прочтения тех участков криптограммы, для которых восстановленная значность шифробозначений не вызывает сомнений.

Увеличение значности шифробозначений делает шифр неэкономным, поэтому получили распространение шифры, использующие одно- и двузначные шифробозначения, подобные рассмотренному выше в примере цифровому шифру. Понятно, что для таких шифров наибольшую повторяемость в шифртексте имеют цифры, с которых начинаются двузначные шифробозначения. Выдвигая гипотезы о таких цифрах и отмечая в шифртексте соответствующие двузначные шифробозначения, можно восстановить и однозначные шифробозначения, оказавшиеся в шифртексте между некоторыми двузначными шифробозначениями. Дальнейшая работа по вскрытию открытого текста для разнозначного шифра ничем не отличается от уже знакомой нам работы для однобуквенной простой замены.

§ 5.3. Блочные шифры простой замены

Как мы убедились, задача вскрытия простой однобуквенной замены является не слишком сложной. Основная слабость такого шифра состоит в том, что избыточность открытого текста, полностью проникающая в шифртекст, делает (за счет малого числа шифрвеличин, которыми являются буквы алфавита) очень рельефной диаграмму повторяемости знаков криптограммы. Это побудило в свое время криптографов к устранению этой слабости за счет увеличения числа шифрвеличин. Интуитивно понятно, что чем больше разница между числом шифрвеличин и числом букв алфавита, тем более равномерной должна стать диаграмма повторяемости знаков шифртекста. Первым естественным шагом в этом направлении стало увеличение значности шифрвеличин, то есть использование блочных шифров простой замены.

Простейший блочный шифр оперирует с биграммными шифрвеличинами. Одними из первых таких шифров были биграммные шифры Порта и Плейфера (см. гл. 1). Приведем описание шифра Плейфера, нашедшего широкое применение в начале нашего века.

Основой шифра Плейфера является прямоугольная таблица, в которую записан систематически перемешанный алфавит (для удобства запоминания). Правило зашифрования состоит в следующем.

Буквы биграммы (i, j) , $i \neq j$ (являющейся шифрвеличиной) находятся в данной таблице. При зашифровании биграмма (i, j) заменяется биграммой (k, l) , где k и l определяются в соответствии с правилами 1–3.

1. Если i и j не лежат в одной строке или одном столбце, то их позиции образуют противоположные вершины прямоугольника. Тогда k и l — другая пара вершин, причем k — вершина, лежащая в той же строке, что и i .

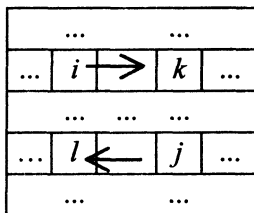
2. Если i и j лежат в одной строке, то k и l — буквы той же строки, расположенные непосредственно справа от i и j

соответственно. При этом если одна из букв — последняя в строке, то считается, что ее “правым соседом” является первая буква той же строки.

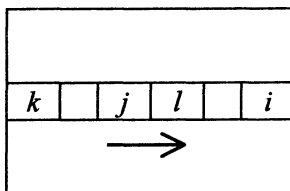
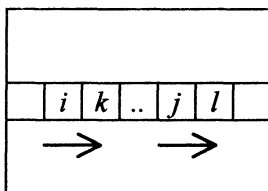
3. Аналогично, если i и j лежат в одном столбце, то они заменяются их “соседями снизу”.

Наглядно правило зашифрования можно проиллюстрировать следующим образом.

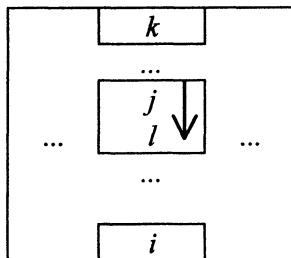
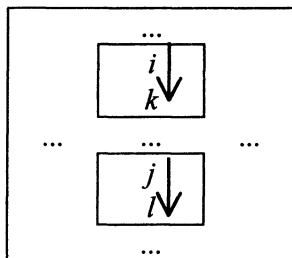
В случае 1:



В случае 2:



В случае 3:



При зашифровании открытый текст представляется в виде последовательности биграмм. Если текст имеет нечетную

длину или содержит биграмму, состоящую из одинаковых букв, то в него добавляются “пустышки” следующим образом. “Пустышкой” является некоторая редкая для данного типа текста буква (или знак), которая вставляется между одинаковыми буквами биграммы или добавляется в текст для того, чтобы его длина стала четной. Такие изменения открытого текста, как правило, не мешают при расшифровании. Проиллюстрируем сказанное следующим примером.

Пример (шифра Плейфера)

Пусть шифр использует прямоугольник размером 5×6 , в который записан систематически перемешанный русский 30-буквенный алфавит на основе ключевого слова *командир*:

к	о	м	а	н	д
и	р	б	в	г	е
ж	з	л	п	с	т
у	ф	х	ц	ч	ш
щ	ь	ы	э	ю	я

Зашифруем фразу “*автором метода является Уитстон*”. В качестве “пустышки” будем использовать редкую букву *ф*.

Представим фразу в виде последовательности биграмм:

АВ ТО РО МФ МЕ ТО ДА ЯВ ЛЯ ЕТ ТЯ УИ ТС ТО НФ

(Нам пришлось дважды вставить “пустышку”).

В соответствии со сформулированными правилами получаем шифртекст:

ВП ЗД ЗР ОХ ДБ ЗД КН ЭЕ ТЫ ТШ ШД ЩЖ ЖТ ЗД ОЧ

или без пробелов

впздрохдбздкнэетытшшдщжжтздох

Криптоанализ шифра Плейфера опирается на частотный анализ биграмм, триграмм и четырехграмм шифртекста и особенности замены шифрвеличин на шифробозначения, связанные с расположением алфавита в прямоугольнике. При этом существенную информацию о заменах дает знание того, что используется систематически перемешанный алфавит. Замечательный пример вскрытия криптограммы, полученной применением шифра Плейфера, приведен в [Sal96].

Шифрвеличинами для другого широко известного блочного шифра — *шифра Хилла* (названного по имени Лестора Хилла) — являются n -граммы открытого текста ($n \geq 2$), представленного некоторым числовым кодом (так что алфавитом открытого текста служит кольцо вычетов по модулю мощности алфавита Z_m).

Правило зашифрования представляет собой линейное преобразование кольца Z_m : если $x = (x_1, \dots, x_n)$ — n -грамма открытого текста, $k = (k_{ij})$ — некоторая обратимая матрица над Z (ключ) и $y = (y_1, \dots, y_n)$ — n -грамма шифртекста, то $y^\downarrow = E_k(x) = k \cdot x^\downarrow$. Соответственно $x^\downarrow = D_k(y) = k^{-1} \cdot y^\downarrow$, где k^{-1} — матрица, обратная к матрице для k .

Подчеркнем, что матричные операции здесь производятся над кольцом Z .

Проиллюстрируем введенное определение примером.

Пример (шифра Хилла)

Положим $n = 4$ и зашифруем фразу:

без труда не вынешь рыбку из пруда

записанную в 30-буквенном русском алфавите. Условимся о числовом кодировании букв в соответствии с таблицей:

А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П
1	18	11	6	0	15	20	5	21	23	13	4	16	8	25

Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я
24	10	19	26	12	2	28	7	17	22	3	29	27	14	9

В качестве ключа выберем матрицу

$$k = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 5 \\ 1 & 3 & 3 & 5 \\ 1 & 3 & 4 & 5 \end{pmatrix},$$

являющуюся обратимой над кольцом Z_{30} . Несложно убедиться в том, что

$$k^{-1} = \begin{pmatrix} 5 & 28 & 1 & 27 \\ 0 & 29 & 1 & 0 \\ 0 & 0 & 29 & 1 \\ 29 & 1 & 0 & 0 \end{pmatrix}.$$

Запишем открытый текст по столбцам матрицы T :

$$T = \begin{pmatrix} 18 & 24 & 16 & 16 & 24 & 26 & 24 \\ 15 & 26 & 15 & 15 & 29 & 21 & 26 \\ 5 & 0 & 11 & 17 & 18 & 5 & 0 \\ 19 & 1 & 29 & 3 & 23 & 25 & 1 \end{pmatrix}$$

и получим шифртекст в виде столбцов матрицы $k \cdot T$:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 5 \\ 1 & 3 & 3 & 5 \\ 1 & 3 & 4 & 5 \end{pmatrix} \cdot \begin{pmatrix} 18 & 24 & 16 & 16 & 24 & 26 & 24 \\ 15 & 26 & 15 & 15 & 29 & 21 & 26 \\ 5 & 0 & 11 & 17 & 18 & 5 & 0 \\ 19 & 1 & 29 & 3 & 23 & 25 & 1 \end{pmatrix} = \\ = \begin{pmatrix} 19 & 20 & 15 & 19 & 18 & 3 & 20 \\ 8 & 21 & 14 & 22 & 11 & 28 & 21 \\ 23 & 17 & 29 & 7 & 10 & 19 & 17 \\ 28 & 17 & 10 & 24 & 28 & 24 & 17 \end{pmatrix}.$$

Теперь осталось воспользоваться числовым кодом, чтобы выписать шифртекст в буквенном виде:

токзжшшшеюыстцчрбвсцьцтржшшш

Замечание. Из соображений удобства в применении получили широкое распространение шифры, для которых правила зашифрования и расшифрования идентичны. Такие шифры называются *обратимыми*. Шифр Хилла является обратимым в том и только том случае, когда $k^{-1} = k$, или иначе: $k^2 = E$, где E — единичная матрица. Матрица, удовлетворяющая этому свойству, называется *инволютивной*.

Из курса алгебры известен критерий обратимости квадратной матрицы над кольцом Z_m .

Теорема. Квадратная матрица M над кольцом Z_m обратима тогда и только тогда, когда $(|M|, m) = 1$, то есть определитель $|M|$ взаимно прост с m .

Заметим, что правило зашифрования E_M (с матрицей $M_{n \times n}$) в шифрсистеме Хилла является обратимым линейным

преобразованием n -мерного модуля Z_m^n . Такая функция является лишь одним из примеров простого задания обратимого преобразования $Z_m^n \rightarrow Z_m^n$, являющегося, очевидно, биекцией на Z_m^n . Любая же такая биекция потенциально может рассматриваться как правило зашифрования блочного шифра простой замены (если n — размер блока), выбираемого некоторым ключом. Поскольку число обратимых линейных преобразований модуля Z_m^n составляет (при $n > 2, m > 2$) лишь незначительную часть общего числа рассматриваемых биекций, то же самое можно сказать и о доле, которую составляют (при данном n) шифрсистемы Хилла в множестве возможных блочных шифров простой замены.

Замечание. Можно проверить, что имеется всего 24 обратимых преобразования $f: Z_2^2 \rightarrow Z_2^2$, причем все они являются аффинными, то есть для любого $\vec{x} \in Z_2^2$ они имеют вид

$$f(\vec{x}) = \vec{x} \cdot A + \vec{b},$$

где A — некоторая обратимая 2×2 -матрица над Z_2 , и $\vec{b} \in Z_2^2$.

Естественным обобщением шифра Хилла является *аффинный блочный шифр*, правило зашифрования $E_{(A, \vec{b})}(\vec{x})$ которого определяется формулой (1). При этом A является обратимой $n \times n$ матрицей, а \vec{b} — фиксированным n -мерным вектором над Z_m .

Как и при рассмотрении предыдущих шифров, приведем некоторые соображения о криптоанализе аффинного блочного шифра и, в частности, шифра Хилла.

Увеличение значности шифрвеличин резко усложняет попытки вскрытия открытого текста по известному тексту криптограммы. Однако свойство линейности, присущее рас-

смаатриваемым шифрам, конечно, является их криптографической слабостью. Чтобы показать это, рассмотрим следующую криптоатаку на аффинный шифр.

Предположим, что известны $n + 1$ пар блоков открытого текста и соответствующих им блоков шифртекста: $(\bar{x}_0, \bar{y}_0), \dots, (\bar{x}_n, \bar{y}_n)$, $\bar{x}_i, \bar{y}_j \in Z_m^n$, полученных на одном ключе (A, \bar{b}) . Требуется этот ключ найти.

Для решения поставленной задачи положим:

$$\bar{x}^{(i)} = \bar{x}_i - \bar{x}_0, \bar{y}^{(i)} = \bar{y}_i - \bar{y}_0, i = \overline{1, n}.$$

Тогда $\bar{x}^{(i)}$ и $\bar{y}^{(i)}$ оказываются связанными линейным соотношением $\bar{y}^{(i)} = \bar{x}^{(i)} \cdot A$.

Аналогичное соотношение имеет место и для матриц

$$\tilde{X} = \begin{pmatrix} \bar{x}^{(1)} \\ \dots \\ \bar{x}^{(n)} \end{pmatrix} \text{ и } \tilde{Y} = \begin{pmatrix} \bar{y}^{(1)} \\ \dots \\ \bar{y}^{(n)} \end{pmatrix} : \tilde{Y} = \tilde{X} \cdot A,$$

из которого находим матрицу A в виде произведения $A = \tilde{X}^{-1} \cdot \tilde{Y}$.

Наконец, вектор \bar{b} находится, например, по формуле $\bar{b} = \bar{y}_0 - \bar{x}_0 \cdot A$. Естественно, что такое решение корректно лишь в том случае, когда матрица \tilde{X} обратима.

Поскольку обращение матрицы и вычисление произведения матриц являются не слишком трудоемкими операциями, то таковой же является и поставленная задача.

Более детальное рассмотрение блочных шифров простой замены будет продолжено в гл. 8.

§ 5.4. Многоалфавитные шифры замены

Напомним, что правило зашифрования многоалфавитного шифра однозначной замены определяется следующим образом. Пусть $x = (x_1, \dots, x_l)$ — открытый текст, представленный последовательностью шифрвеличин $x_i \in U$, $i = \overline{1, l}$, и k — произвольный ключ. Тогда

$$E_k(x) = (\pi_1(x_1), \dots, \pi_l(x_l)), \quad (2)$$

где $\pi_i, i = \overline{1, l}$, — некоторые подстановки на множестве всех шифрвеличин, однозначно определяемые данным ключом. При этом здесь и далее мы ограничимся рассмотрением случая, когда множества шифрвеличин и шифробозначений совпадают друг с другом ($U = V$).

Заметим, что в рассматриваемых условиях любой многоалфавитный шифр представляет собой совокупность шифров простой замены, каждая из которых используется для зашифрования очередной шифрвеличины в соответствии с вспомогательной последовательностью $\psi(k, l)$ (распределителем), определяемой выбранными ключом и открытым текстом по формуле (1), приведенной в гл. 3. Принципиально один многоалфавитный шифр отличается от другого лишь способом образования распределителя.

На практике используются в основном поточные многоалфавитные шифры, среди которых выделяются два больших подкласса — шифры, реализуемые дисковыми шифраторами, и шифры гаммирования. В литературе можно найти упоминание и о других примерах поточных многоалфавитных замен (см., например, [Cal78]). В подобных шифрсистемах в (2) потенциально могут использоваться все возможные подстановки π данного алфавита. Они строятся с помощью произ-

ведений определенного вида из небольшого числа исходных подстановок.

В следующем подпункте мы остановимся на дисковых шифрах, а шифрам гаммирования, в силу их большой значимости, отведем отдельную главу.

§ 5.5. Дисковые многоалфавитные шифры замены

Общая характеристика и принцип действия дискового шифратора были даны в гл. 1. Здесь мы рассмотрим правило зашифрования и некоторые свойства такого шифра.

Прежде всего следует выписать преобразование символов алфавита (в качестве которого, как и ранее, будем рассматривать множество $Z_n = \{0, 1, \dots, n-1\}$), осуществляемое движущимся диском. Для этого рассмотрим два соседних угловых положения диска при его повороте (по часовой стрелке). Пусть в исходном положении диск реализует подстановку

$$X = \begin{pmatrix} 0 & 1 & \dots & n-1 \\ x_0 & x_1 & \dots & x_{n-1} \end{pmatrix}.$$

Для того чтобы выписать подстановку, реализуемую диском после поворота на угол $2\pi/n$, взглянем на соответствующие рисунки (см. рис. 15, 16).

Так как диск сдвигается как твердое тело, символ открытого текста, поступающий на него с входной розетки, проходит затем по имеющимся в диске соединениям, превращаясь в символ шифртекста. Разница между двумя рассматриваемыми положениями диска, как видно из рисунков, состоит лишь в том, что после поворота символы с входной розетки поступают на входные контакты диска, номера которых уменьшаются на единицу (по модулю n). Мы можем перенумеровать входные символы, уменьшив каждый на единицу.

Тогда входные контакты диска будут совпадать с входными символами, которые, пройдя по своим траекториям через диск, попадут на контакты выходной розетки. Чтобы вернуться к исходной нумерации символов, следует их увеличить на единицу.

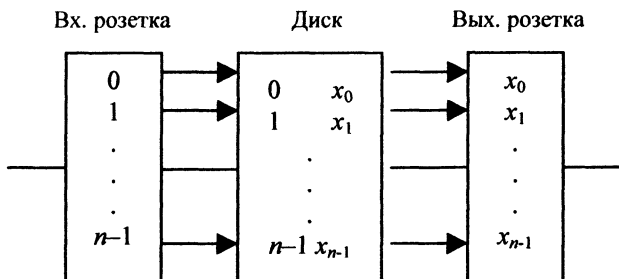


Рис. 15. Начальное расположение диска

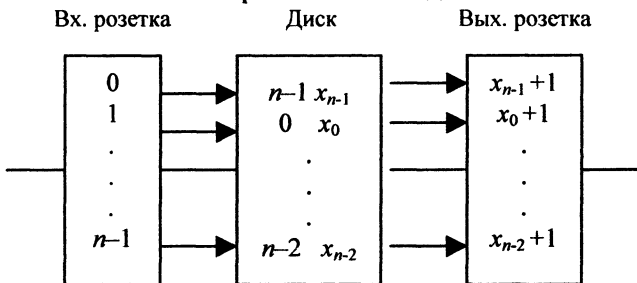


Рис. 16. Положение диска после поворота

Если ввести в рассмотрение подстановку

$$T = \begin{pmatrix} 0 & 1 & \dots & n-2 & n-1 \\ 1 & 2 & \dots & n-1 & 0 \end{pmatrix},$$

то из сказанного выше следует, что после поворота диск реализует подстановку, представимую в виде произведения подстановок:

$$T^{-1} \cdot X \cdot T = \begin{pmatrix} i \\ i-1 \end{pmatrix} \cdot \begin{pmatrix} i-1 \\ x_{i-1} \end{pmatrix} \begin{pmatrix} x_{i-1} \\ x_{i-1} + 1 \end{pmatrix} = \begin{pmatrix} i \\ x_{i-1} + 1 \end{pmatrix}.$$

Теперь очевидно, что при повороте диска на угол $\frac{2m\pi}{n}$, $m = \overline{1, n-1}$, диск будет реализовать подстановку

$$T^{-m} \cdot X \cdot T^m.$$

Рассмотрим теперь дисковый шифратор, состоящий из нескольких насаженных на общую ось дисков, так что символы с входной розетки, попадая на блок дисков, последовательно проходят перепайки каждого из дисков, попадая на контакты выходной розетки. Обычно при работе такого шифратора диски при шифровании очередного знака открытого текста сдвигаются (по определенному правилу) на некоторые угловые положения (кратные $\frac{2\pi}{n}$). Схема движения дисков является ключевым элементом шифратора. Получим правило зашифрования текущего знака открытого текста такого шифратора.

Пусть в начальных угловых положениях рассматриваемые диски реализуют подстановки X_1, \dots, X_N из симметрической группы S_n (они также являются ключевыми элементами) и в данный такт шифрования данные диски находятся в соответствующих угловых положениях $\gamma_1, \dots, \gamma_N$, $\gamma_i \in \overline{0, n-1}$. Это означает, что i -й диск реализует подстановку $T^{-\gamma_i} \cdot X_i \cdot T^{\gamma_i}$. Тогда очередная буква открытого текста x будет зашифрована в букву $y = E_k(x)$, где

$$y = T^{-\gamma_1} \cdot X_1 \cdot T^{\gamma_1 - \gamma_2} \cdot X_2 \cdot T^{\gamma_2 - \gamma_3} \cdot \dots \cdot T^{\gamma_{N-1} - \gamma_N} \cdot X_N \cdot T^{\gamma_N}(x).$$

Формально определить правило зашифрования любого открытого текста для дискового шифратора чрезвычайно сложно (в связи с обилием различных ключевых элементов). Для поточных

шифров, как правило, бывает достаточно знания правила зашифрования буквы текста.

Число простых замен, из которых “состоит” многоалфавитный шифр, реализуемый дисковым шифратором, может быть чрезвычайно большим. Чем больше это число, тем сложнее криптоанализ такого шифра. В связи с этим параметры дисковых схем (число дисков, реализуемые ими подстановки, схемы движения дисков и т. д.) должны быть тщательно продуманы.

Схемы токопрохождения электрических импульсов в дисковом шифраторе могут усложняться за счет введения “отражающего экрана”, вместо выходной розетки. В результате этого импульс тока вторично проходит через блок дисков, только в противоположную сторону. Такая “обратимая” схема токопрохождения была использована в знаменитой “Энигме”.

Криптоанализ дисковых шифраторов является весьма сложной задачей, выходящей за рамки данной книги.

Контрольные вопросы

1. Какие шифры называются шифрами простой замены?
2. Что является ключом шифра простой замены? Каково максимально возможное число ключей шифра простой замены?
3. Что более целесообразно для надежной защиты информации: архивация открытого текста с последующим шифрованием или шифрование открытого текста с последующей архивацией?
4. Имеет ли шифр Плейфера эквивалентные ключи, то есть такие ключи, на которых любые открытые тексты шифруются одинаково? Сколько различных неэквивалентных ключей имеет шифр Плейфера?
5. Предположим, что матричный шифр Хилла используется для зашифрования открытого текста, представленного в виде двоячной последовательности. Сколько ключей имеет такой шифр?

Глава 6

Шифры гаммирования

Напомним, что в основе рассматриваемых систем шифрования лежит метод “наложения” ключевой последовательности — гаммы — на открытый текст. “Наложение” заключается в позначном (побуквенном) сложении или вычитании по тому или иному модулю. Хотя мы уже отмечали выше, что данные шифрсистемы относятся к многоалфавитным системам замены, шифры гаммирования имеют целый ряд особенностей и заслуживают отдельного рассмотрения. В силу простоты своей технической реализации и высоких криптографических качеств эти шифры получили широкое распространение.

Исторически первый шифр гаммирования совпадал, по сути, с шифром Виженера, однако без использования самой таблицы Виженера. Заметим, что таблица Виженера представляет собой квадрат, каждая строка и каждый столбец которого — некоторая перестановка знаков данного алфавита. Произвольная такая таблица называется *латинским квадратом*. Идя по пути обобщения, введем понятие шифра табличного гаммирования.

§ 6.1. Табличное гаммирование

Шифр табличного гаммирования в алфавите $A = \{a_1, \dots, a_n\}$ определяется произвольным латинским квадратом L на A и способом получения последовательности букв из A , называемой *гаммой шифра* (см. рис. 17). Буква a_i открытого текста под действием знака гаммы a_j переходит в букву a_k шифрованного текста, содержащуюся в j -й строке и

i -м столбце квадрата L (подразумевается, что строки и столбцы в L занумерованы в соответствии с порядком следования букв в алфавите A).

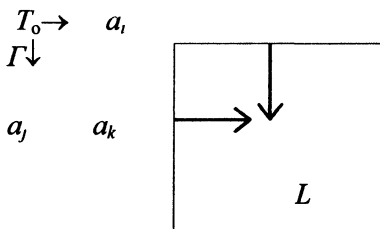


Рис. 17

С алгебраической точки зрения буква a_k есть результат применения к буквам a_i и a_j квазигрупповой операции $*$, табличным заданием которой является латинский квадрат L :

$$a_k = a_i * a_j .$$

В случае шифра Виженера квазигруппа $(A,*)$ является группой $(Z_n,+)$. При этом уравнение шифрования имеет вид

$$b_i = (a_i + \gamma_i) \bmod n , \quad (1)$$

а $\{\gamma_i\}$ представляет собой периодическую последовательность, образованную повторением некоторого ключевого слова.

Наряду со сложением используется и вычитание знаков гаммы. Соответствующие уравнения шифрования принимают вид

$$b_i = (a_i - \gamma_i) \bmod n \quad (2)$$

или

$$b_i = (\gamma_i - a_i) \bmod n . \quad (3)$$

Шифры гаммирования с уравнениями шифрования (1) — (3) обычно называют *шифрами модульного гаммирования*.

Если в качестве квазигрупповой операции $*$ на множестве 5-мерных двоичных векторов используется операция покоординатного сложения по модулю 2:

$$b_i = a_i \oplus \gamma_i, \quad (4)$$

то получаем *шифр Вернама*.

Шифры гаммирования, определяемые уравнениями (3) и (4), замечательны тем, что при их применении для зашифрования и расшифрования требуется лишь один узел. В самом деле, знаки открытого текста находятся из тех же уравнений при взаимной замене a_i на b_i . Такие шифры обычно называют *обратимыми* (см. замечание после примера шифра Хилла).

Сделаем следующее замечание. Как шифр замены, произвольный шифр гаммирования имеет следующую интерпретацию.

С j -й строкой латинского квадрата L ($j = \overline{1, n}$) можно связать подстановку g_j (“сдвиг” на a_j):

$$g_j = \begin{pmatrix} a_1 & a_2 & \dots & a_n \\ a_1 * a_j & a_2 * a_j & \dots & a_n * a_j \end{pmatrix}$$

из симметрической группы $S(A)$. Пусть

$$R(A) = \{g_j : j = \overline{1, n}\}.$$

Тогда в каждом такте шифрования знак открытого текста заменяется по одной из подстановок из $R(A)$. Распределителем такого n -алфавитного шифра замены является сама гамма шифра.

Иногда $R(A)$ — это группа или смежный класс по некоторой подгруппе из $S(A)$. В таких случаях мы будем называть шифр табличного гаммирования *групповым*. Произвольный шифр табличного гаммирования не слишком удобен для практической реализации. Наиболее удобны именно групповые шифры, к которым относятся шифры модульного гаммирования.

§ 6.2. О возможности восстановления вероятностей знаков гаммы

Криптоанализ произвольного шифра табличного гаммирования во многом схож с криптоанализом шифра модульного гаммирования. Рассмотрим основные идеи анализа на примере шифра с уравнением (1).

Занумеруем буквы алфавита A числами от 0 до $n-1$ и воспользуемся формальными моделями рассматриваемых последовательностей (см. гл. 2). Пусть p_i, r_i и s_i — вероятности появления знака i в открытом тексте, гамме и в шифрованном тексте соответственно. Тогда задание вероятностных распределений на знаках открытого текста и гаммы (которые естественно считать независимыми) индуцирует распределение вероятностей знаков шифртекста по формуле:

$$s_j = \sum_{i=0}^{n-1} p_{j-i} \cdot r_i, \quad (5)$$

в которой разность $j-i$ берется по модулю n . (Достаточно заметить, что $b = j \Leftrightarrow a = j-i, \gamma = i$, и воспользоваться формулой полной вероятности.) Легко проверить, что

$$\sum_{j=0}^{n-1} s_j = 1.$$

Из формулы (5) следует, что если $r_i = 1/n$ при всех $i = \overline{0, n-1}$, то и $s_j = 1/n$ при всех $j = \overline{0, n-1}$. Это означает, что при зашифровании открытого текста равновероятной гаммой получается шифртекст, вероятностные свойства которого не отличаются от самой равновероятной гаммы. Это обстоятельство не оставляет шансов криптоаналитику использовать диаграмму повторяемости букв открытого текста, поскольку при наложении гаммы эта информация как бы стирается. Поэтому на практике стремятся к тому, чтобы по своим вероятностным свойствам гамма была близка к случайной равновероятной последовательности.

Возникает естественный вопрос о том, можно ли при использовании неравновероятной гаммы восстановить ее вероятностные характеристики непосредственно по шифртексту и можно ли эту информацию использовать при криптоанализе шифра гаммирования.

Попытаемся сначала оценить вероятности r_i непосредственно по шифртексту. При этом мы должны располагать достаточно точными приближениями распределений

$$\vec{p} = (p_0, \dots, p_{n-1}), \quad \vec{s} = (s_0, \dots, s_{n-1})$$

(получаемыми с помощью подсчета частот встречаемости знаков).

Рассмотрим соотношение (5) как систему линейных уравнений относительно неизвестных $r_i, i = \overline{0, n-1}$. Нетрудно заметить, что матрица рассматриваемой системы имеет вид

$$P = \begin{pmatrix} p_0 & p_{n-1} & \dots & p_1 \\ p_1 & p_0 & \dots & p_2 \\ \dots & \dots & \dots & \dots \\ p_{n-1} & p_{n-2} & \dots & p_0 \end{pmatrix}.$$

Такая матрица называется *циркулянт* [Кос87]. В ней каждая строка получается циклическим сдвигом предыдущей строки. Известно [Кос87], что определитель $|P|$ циркулянта равен произведению

$$f(\varepsilon_0) \cdot f(\varepsilon_1) \cdot \dots \cdot f(\varepsilon_{n-1}),$$

где $\{\varepsilon_0, \dots, \varepsilon_{n-1}\}$ — множество всех корней степени n из 1 (в поле комплексных чисел), причем

$$f(x) = p_0 + p_{n-1} \cdot x + \dots + p_1 \cdot x^{n-1}.$$

В том случае, когда $|P| \neq 0$, вектор r^\downarrow однозначно определяется из соотношения

$$r^\downarrow = P^{-1} \cdot s^\downarrow. \quad (6)$$

Приведем (без доказательства) формулу для P^{-1} :

$$P^{-1} = \begin{pmatrix} x_0 & x_1 & \dots & x_{n-1} \\ x_{n-1} & x_0 & \dots & x_{n-2} \\ \dots & \dots & \dots & \dots \\ x_1 & x_2 & \dots & x_0 \end{pmatrix}, \quad (7)$$

где

$$x_t = \frac{1}{n} \cdot \sum_{m=0}^{n-1} \frac{\varepsilon_m^t}{B_m(\bar{p})}, \quad t = \overline{0, n-1}, \quad B_m(\bar{p}) = \sum_{j=0}^{n-1} p_j \cdot \varepsilon_m^j.$$

Условие $|P| \neq 0$ можно проверить непосредственно по данному распределению вероятностей букв открытого текста.

Пользуясь (6) и (7), можно вычислить приближение r^{\downarrow} для r^{\downarrow} , подставляя вместо s^{\downarrow} в (6) вектор $v^{\downarrow} = \frac{1}{l} \begin{pmatrix} v_0 \\ \dots \\ v_{n-1} \end{pmatrix}$, где

v_i — число вхождений символа i в зашифрованный текст:

$$r^{\downarrow} = \frac{1}{l} \cdot (P^{-1} \cdot v^{\downarrow}),$$

откуда

$$r^{\downarrow}_{\gamma} = \frac{1}{l} \cdot \sum_{j=0}^{n-1} x_j \cdot v_{j-\gamma}, \quad \gamma = \overline{0, n-1}.$$

§ 6.3. Восстановление текстов, зашифрованных неравновероятной гаммой

Пример (использования шифра модульного гаммирования)

Рассмотрим следующую постановку задачи. Пусть при использовании шифра модульного гаммирования в результате, например, некоторой неисправности гаммообразующего

устройства, т.е. устройства, вырабатывающего гамму, в ней встречаются не все знаки. Предположим, далее, что гамма состоит лишь из знаков $\gamma_1, \dots, \gamma_m$, $m < n$, которые встречаются с вероятностями $r_{\gamma_1}, \dots, r_{\gamma_m}$ соответственно. Будем также предполагать, что исходный открытый текст является обычным литературным текстом. В этих условиях требуется дешифровать полученную криптограмму.

Заметим, что к подобной постановке задачи можно было прийти иначе. Выделив несколько знаков гаммы, имеющих достаточную суммарную вероятность, например 0,8, предположим, что лишь они использовались при шифровании. Это предположение может привести к потере истинного варианта при построении некоторого метода вскрытия. Вероятность потери можно оценить при этом стандартными методами статистики.

При решении поставленной задачи примем во внимание, что в i -м такте шифрованию подлежала одна из следующих букв открытого текста:

$$t_i^{(1)} = s_i - \gamma_1, \dots, t_i^{(m)} = s_i - \gamma_m, \quad (8)$$

где s_i — буква шифртекста.

Поэтому знаки открытого текста следует искать в колонках таблицы, изображенной на рис. 18:

	s_1	s_2	...	s_i	...
γ_1	$t_1^{(1)}$	$t_2^{(1)}$...	$t_i^{(1)}$...
γ_2	$t_1^{(2)}$	$t_2^{(2)}$...	$t_i^{(2)}$...
...
γ_m	$t_1^{(m)}$	$t_2^{(m)}$...	$t_i^{(m)}$...

Рис.18

Отбирая по одному знаку из каждой колонки так, чтобы получился “читаемый” текст, мы получим возможность восстановить открытый текст.

Описанный метод относится к классу так называемых *методов бесключевого чтения* (когда открытый текст восстанавливается без предварительного определения ключа) и называется *методом чтения в колонках*.

Метод чтения в колонках можно усовершенствовать за счет упорядочения букв в колонках. В самом деле, в каждом такте возможные знаки открытого текста

$$t^{(1)} = s - \gamma_1, \dots, t^{(m)} = s - \gamma_m$$

имеют априорные вероятности $p_{t^{(1)}}, \dots, p_{t^{(m)}}$, которые считаются известными. В нашем случае имеется также дополнительная информация, а именно, известно, что произошло событие “ $s_i = s$ ”. При этом

$$p\{s_i = s / t_i = t^{(k)}\} = r_{\gamma_k}, \quad k = \overline{1, m}.$$

Отсюда по формуле Байеса получаем

$$p\{t_i = t^{(k)} / s_i = s\} = \frac{p_{t^{(k)}} \cdot r_{\gamma_k}}{\sum_{j=1}^m p_{t^{(j)}} \cdot r_{\gamma_j}}, \quad k = \overline{1, m}. \quad (9)$$

Теперь можно упорядочить вероятности (8) знаков открытого текста в каждой колонке таблицы в соответствии с убыванием вычисленных апостериорных вероятностей. Поступив таким образом, мы поместим наиболее вероятные знаки открытого текста в начало таблицы, чем облегчим чтение в колонках.

С ростом m чтение в колонках становится затруднительным, а при $m = n$ и при условии, что при шифровании использовалась случайная равновероятная гамма, каждая колонка содержит все знаки алфавита, ни одному из которых нельзя отдать предпочтения. Поэтому в последовательности колонок можно прочесть любой текст, то есть нет возможности получить информацию об истинном сообщении.

Пример (использования неисправности в реализации шифра Вернама)

Рассмотрим шифр гаммирования, определяемый уравнением (4), называемый шифром Вернама. Узел реализации такого шифра можно представить схемой, изображенной на рис.19. На этой схеме кружочками обозначены узлы сложения по модулю 2 битов открытого текста с соответствующими битами гаммы. Знаки открытого текста и знаки гаммы представляются при этом 5-мерными двоичными векторами.

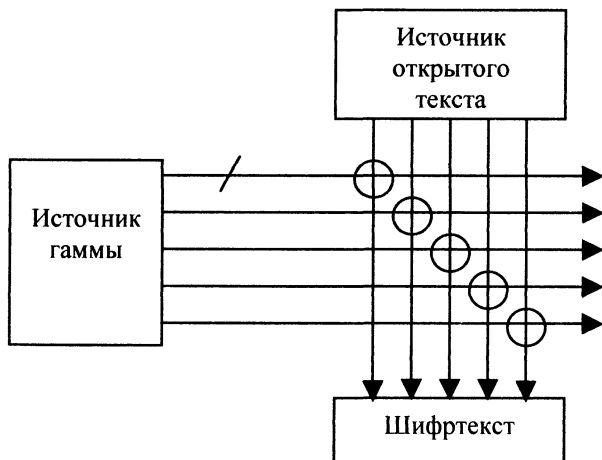


Рис. 19

В случае обрыва одного из “проводков”, идущих от источника гаммы, последовательность знаков гаммы будет со-

держат лишь половину возможных своих значений. Соответствующая координата в любом 5-мерном векторе гаммы будет равна нулю. В случае обрыва двух или большего числа “проводков” векторы гаммы будут содержать два или большее число нулевых координат. Число возможных знаков гаммы будет сокращаться вдвое при каждом обрыве. Таким образом, подобная неисправность схемы приводит к постановке задачи, указанной в предыдущем пункте. В рассматриваемом случае подобную неисправность можно обнаружить по шифртексту.

Покажем, как это сделать при условии, что “исправная” гамма является *случайной и равновероятной*. Будем при этом рассматривать позначные модели открытого текста, гаммы и шифртекста, то есть считать, что они являются реализациями случайных независимых испытаний полиномиальных схем с соответствующими распределениями вероятностей $\vec{p}(A)$, $\vec{r}(A)$, $\vec{s}(A)$ на знаках открытого текста, гаммы и шифртекста. Естественно также условиться, что распределения $\vec{p}(A)$ и $\vec{r}(A)$ являются независимыми. При этом распределение $\vec{s}(A)$ определяется формулой

$$s(y) = \sum_{\substack{(x,\gamma) \\ y=x\oplus\gamma}} p(x) \cdot r(\gamma), \quad (10)$$

где x — знак открытого текста, γ — знак гаммы, y — знак шифртекста.

Итак, в нашем случае алфавитом открытого текста, шифрованного текста и гаммы является множество

$$A = \{(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5) : \alpha_i \in Z_2, i = \overline{1,5}\},$$

образующее абелеву группу относительно операции \oplus поординатного сложения векторов по модулю 2. При обрыве,

например, первого соединения (на рис. 19 обрыв указан символом “/”) возможные знаки гаммы образуют подмножество

$$B = \{(0, \beta_2, \beta_3, \beta_4, \beta_5) : \beta_j \in Z_2, j = \overline{2,5}\},$$

являющееся подгруппой группы (A, \oplus) . Точно так же и при любых других обрывах множество знаков гаммы образует подгруппу B группы (A, \oplus) .

Теорема. При использовании равновероятной гаммы векторы, принадлежащие одному смежному классу группы A по подгруппе B , встречаются в шифртексте с равными вероятностями.

Доказательство. Разложим группу $A = \{a_1, \dots, a_n\}$ в левые смежные классы по подгруппе $B = \{b_1, \dots, b_m\}$:

$$A = B \cup (g_2 \oplus B) \cup \dots \cup (g_r \oplus B), \quad (11)$$

где $g_i \in A$, $i = \overline{2, t}$, — представители соответствующих смежных классов, и рассмотрим один из смежных классов $H = \{h_1, \dots, h_m\}$ в этом разложении. Пусть для удобства $H = g \oplus B$. Ясно, что числа t, m, n связаны равенством $n = t \cdot m$.

Выберем нумерацию элементов множеств B и H в соответствии с равенством

$$h_k = g \oplus b_k \quad (12)$$

при $k = \overline{1, m}$.

Так как B само является группой, при любых $i, k \in \overline{1, m}$ найдется такое $j \in \overline{1, m}$, для которого выполняется равенство

$$b_i = b_k \oplus b_j. \quad (13)$$

Заметим, попутно, что в группах A и B каждый элемент является обратным для самого себя.

Легко видеть, что если в (13) индекс i пробегает все значения от 1 до m , то (при фиксированном k) и индекс j пробегает то же множество значений.

Вычислим вероятность $s(h_k)$ появления знака h_k смежного класса H в шифртексте. В связи с условием равновероятности знаков гаммы, а также в соответствии с (10) – (13) имеем:

$$\begin{aligned} s(h_k) &= \sum_{(10) i=1}^m p(h_k \oplus b_i) \cdot r(b_i) = \frac{1}{(13) m} \sum_{j=1}^m p(h_k \oplus (b_k \oplus b_j)) = \\ &= \frac{1}{m} \cdot \sum_{j=1}^m p((h_k \oplus b_k) \oplus b_j) = \\ &= \frac{1}{(12) m} \cdot \sum_{j=1}^m p(g \oplus b_j) = \frac{1}{m} \cdot \sum_{j=1}^m p(h_j). \end{aligned}$$

При этом мы воспользовались тем очевидным свойством, что в случае обрыва знаки гаммы по-прежнему будут иметь одинаковые вероятности.

Отсюда следует, что вероятность $s(h_k)$ не зависит от k и совпадает с $s(h_l)$ для любого $l = \overline{1, m}$, что и требуется.

Теорема доказана.

Доказанная теорема позволяет определить по шифртексту характер произошедшей неисправности. Для этого с использованием частотных свойств используемого кода (например, МТК-2 или др.), аналогичных частотам букв в открытом тек-

сте, можно подсчитать значения вероятностей знаков смежных классов $g_i \oplus B$, $i = \overline{1, t}$, из разложения (11) по формуле

$$s(\alpha) = \frac{1}{m} \cdot \sum_{\beta \in (g_i \oplus B)} p(\beta),$$

где $\alpha \in g_i \oplus B$. Этот подсчет может быть проведен для любых комбинаций обрывов. Теперь остается определить частоты символов шифртекста и сравнить их с рассчитанными заранее эталонными диаграммами. Сравнение выявит характер неисправности, и задача восстановления открытого текста будет сведена к чтению в колонках.

Заметим, что вместо 5-мерных можно было рассматривать и n -мерные векторы, $n \geq 5$. Предложенный метод работает и в этом, более общем случае.

§ 6.4. Повторное использование гаммы

Как и раньше, мы предполагаем, что алфавит A открытых текстов, гаммы и шифртекстов представляет собой множество чисел $Z_n = \{0, 1, \dots, n-1\}$.

Пусть в распоряжении криптоаналитика оказались две криптограммы, полученные наложением одной и той же гаммы на два разных открытых текста:

$$S_1 = T_1 + \Gamma \pmod{n},$$

$$S_2 = T_2 + \Gamma \pmod{n},$$

где

$$S_1 = \{s'_i\}_{i=1,2,\dots}, \quad S_2 = \{s''_i\}_{i=1,2,\dots}, \quad T_1 = \{t'_i\}_{i=1,2,\dots},$$

$$T_2 = \{t''_i\}_{i=1,2,\dots}, \quad \Gamma = \{\gamma_i\}_{i=1,2,\dots}.$$

Рассмотрим возможности криптоаналитика по восстановлению исходных открытых текстов.

Прежде всего можно найти позначную разность

$$S = S_1 - S_2 = T_1 - T_2 \pmod{n}.$$

Пусть $S = \{s_i\}_{i=1,2,\dots}$. Тогда поставленная задача сводится к попытке подобрать пару открытых текстов, разность которых совпадает с известной последовательностью S . Будем в связи с этим говорить о *разложении* S на два составляющих открытых текста. В случае когда данные тексты являются нормативными текстами, например, на русском, английском или другом языке, для решения последней задачи используется ряд подходов. Интуитивно понятно, что при достаточной длине текстов маловероятна возможность множественного представления данной последовательности S в виде разности $T_1 - T_2$. Как правило, такое разложение бывает единственным. Здесь имеет место приблизительно такая же ситуация, как и при рассмотрении вопроса о расстоянии единственности (см. гл. 7).

Один из таких подходов (хорошо известных из истории криптографии) связан с использованием некоторого запаса слов или словоформ, часто встречающихся в открытых текстах. Это могут быть, например, стандарты переписки, частые k -граммы и т. п.

Предположим сначала, что одно из вероятных слов встретилось в начале первого сообщения:

$$T_1 = \underbrace{t'_1 t'_2 \dots t'_l}_{\text{вероятное слово}} t'_{l+1} \dots .$$

В таком случае можно вычислить начало второго сообщения:

$$T_2 = t'_1 - s_1 t'_2 - s_2 \dots t'_l - s_l \dots .$$

Если $l \geq 4$, легко определить, является ли начало T_2 “читаемым” или нет. В первом случае нужно попытаться продлить начало T_2 по смыслу. Во втором случае нужно сдвинуть начало вероятного слова в T_1 и проделать то же самое.

Если удалось развить T_2 до m знаков ($m > l$): $T_2 = t_1'' t_2'' \dots t_m'' \dots$, то можно вычислить и соответствующие $m - l$ знаки

$$T_1 = t_1' \dots t_l' t_{l+1}'' + s_{l+1} \dots t_m'' + s_m,$$

и попытаться, в свою очередь, развить по смыслу T_1 .

Продолжая этот процесс далее, мы частично или полностью восстановим оба текста или убедимся в том, что опробуемого вероятного слова данные тексты не содержат. В последнем случае следует попытаться ту же процедуру проделать для следующего вероятного слова.

Может оказаться так, что при опробовании некоторого слова удастся восстановить лишь часть каждого из текстов, а дальнейшее развитие их по смыслу бесперспективно. В таком случае следует продолжить работу с другим вероятным словом.

Конечно, данный метод далеко не всегда приводит к успеху. Но нельзя пренебрегать шансом, который он дает.

Пример

Возьмем два текста на английском языке, содержащих наиболее часто встречающуюся триграмму THE:

$$T_1 = \text{THE APPLE}, \quad T_2 = \text{TELL THEM},$$

и зашифруем их одной и той же гаммой $\Gamma = \text{ONETWOTHRE}$. При этом будем пользоваться числовыми значениями букв согласно следующей таблице:

00	01	02	03	04	05	06	07	08	09	10	11	12
A	B	C	D	E	F	G	H	I	J	K	L	M

13	14	15	16	17	18	19	20	21	22	23	24	25
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

В результате зашифрования получаем:

T_1	19	07	04	00	15	15	11	04
T_2	19	04	11	11	19	07	04	12
Γ	14	13	04	19	22	14	19	07
S_1	07	20	08	19	11	03	04	11
S_2	07	17	15	04	15	21	23	19
S	00	03	19	15	22	08	07	18

Предположим теперь, что триграмма ТНЕ находится в начале T_1 , тогда можно вычислить начало T_2 :

$T_1 = T_2 + S =$	Т	Н	Е	А
	19	07	04	00
$T_2 = T_1 - S =$	Т	Е	Л	Л
	19	04	11	11

Дальнейшие попытки продолжить по смыслу T_1 или T_2 к успеху не приводят. Поэтому предположим, что T_2 также содержит ТНЕ. С учетом полученного результата, мы получаем два варианта расположения триграммы ТНЕ в тексте T_2 . В первом из них триграмма ТНЕ расположена, начиная с пятой позиции, во втором — начиная с шестой позиции.

Рассмотрим первый вариант:

T_2	19	04	11	11	19	07	04
	Т	Е	Л	Л	Т	Н	Е
T_2	19	07	04	00	15	15	11
	Т	Н	Е	А	Р	Р	Л

Теперь ясно, что $T_2 = \text{THE APPLE}$, откуда получаем $T_1 = \text{TELL THEM}$.

Идея другого способа разложения разности открытых текстов состоит в упорядочении возможных вариантов пар букв (t_i', t_i'') по убыванию апостериорных вероятностей

$$p(t_i' = v, t_i'' = v - u / s_i = u), \quad u, v \in A, \quad i = 1, 2, \dots,$$

построении для каждого i упорядоченных колонок, состоящих из таких пар, и попытке чтения (аналогичной изложенной выше) в колонках сразу двух открытых текстов. При этом (как и ранее) используются позначные модели рассматриваемых последовательностей и аналог формулы (9).

§ 6.5. Криптоанализ шифра Виженера

Рассмотрим шифр модульного гаммирования с уравнением (1), для которого гамма является периодической последовательностью знаков алфавита. Как указывалось в историческом экскурсе, такая гамма обычно получалась периодическим повторением некоторого ключевого слова. Например, ключевое слово KEY дает гамму $\text{KEYKEYKEY} \dots$. Рассмотрим задачу вскрытия такого шифра по тексту одной криптограммы достаточной длины.

Пусть μ — длина ключевого слова. Обычно криптоанализ шифра Виженера проводится в два этапа. На первом этапе

определяется число μ , на втором этапе — само ключевое слово.

Для определения числа μ применяется так называемый *тест Казиски*, названный в честь Ф. Казиски, применившего его в 1863 г. Тест основан на простом наблюдении того, что два одинаковых отрезка открытого текста, отстоящих друг от друга на расстоянии, кратном μ , будут одинаково зашифрованы. В силу этого в шифртексте ищутся повторения длины, не меньшей трех, и расстояния между ними. Обратим внимание на то, что случайно такие одинаковые отрезки могут появиться в тексте с достаточно малой вероятностью.

Пусть d_1, d_2, \dots — найденные расстояния между повторениями и d — наибольший общий делитель этих чисел. Тогда μ должно делить d . Чем больше повторений имеет текст, тем более вероятно, что μ совпадает с d . Для уточнения значения μ можно использовать так называемый *индекс совпадения*, введенный в практику У. Фридманом в 1920 г.

Для строки $x = (x_1, \dots, x_m)$ длины m , составленной из букв алфавита A , *индексом совпадения в x* , обозначаемым $I_c(x)$, будем (следуя [Sti95]) называть вероятность того, что две случайно выбранные буквы из x совпадают.

Пусть $A = \{a_1, \dots, a_n\}$. Будем отождествлять буквы алфавита с числами, так что $a_1 \equiv 0, \dots, a_{n-1} \equiv n-2, a_n \equiv n-1$.

Теорема. *Индекс совпадения в x вычисляется по формуле*

$$I_c(x) = \frac{\sum_{i=0}^{n-1} f_i(f_i - 1)}{m(m-1)}, \quad (14)$$

где f_i — число вхождений буквы a_i в x , $i \in Z_n$.

Доказательство. Будем вычислять $I_c(x)$ как отношение числа благоприятных исходов к общему числу исходов. Благоприятным является исход, при котором на выбранных двух позициях в x расположены одинаковые буквы. Общее число исходов равно, очевидно, C_m^2 . Число благоприятных исходов есть

$$\sum_{i=0}^{m-1} C_{i,1}^2. \quad (15)$$

В самом деле, переупорядочим буквы в x таким образом, чтобы сначала шли f_{a_1} букв a_1 , затем — f_{a_2} букв a_2 и т.д.:

$$\underbrace{a_1, \dots, a_1}_{f_{a_1}} \dots \underbrace{a_n, \dots, a_n}_{f_{a_n}}. \quad (16)$$

Теперь заметим, что при случайном выборе мест (i и j) в строке x благоприятными являются следующие исходы:

$$(a_1) \begin{cases} 0 \dots i \dots j \dots m-1 \\ \dots a_1 \dots a_1 \dots \end{cases}$$

$$(a_2) \begin{cases} 0 \dots i \dots j \dots m-1 \\ \dots a_2 \dots a_2 \dots \end{cases}$$

.....

$$(a_n) \begin{cases} 0 \dots i \dots j \dots m-1 \\ \dots a_n \dots a_n \dots \end{cases}$$

В случае (a_1) мы можем выбрать пару букв a_1 из набора (16) $C_{f_{a_1}}^2$ способами, в случае (a_2) пару букв a_2 из (16) — $C_{f_{a_2}}^2$ способами и т. д. Таким образом, общее число благо-

приятных исходов выражается величиной (15), а индекс совпадения в x — формулой

$$I_c(x) = \frac{\sum_{i=0}^{m-1} C_{f_i}^2}{C_m^2}$$

и, следовательно, формулой (14).

Пусть x — строка осмысленного текста (например, английского). Допустим, как и ранее, что буквы в x появляются на любом месте текста с соответствующими вероятностями p_0, \dots, p_{n-1} независимо друг от друга, где p_i — вероятность появления буквы i в осмысленном тексте, $i \in Z_n$. В такой модели открытого текста вероятность того, что две случайно выбранные буквы из x совпадают с $i \in Z_n$, равна p_i^2 , следовательно,

$$I_c(x) \approx \sum_{i=0}^{n-1} p_i^2. \quad (17)$$

Взяв за основу значения вероятностей p_i , приведенные в Приложении 1 для открытых текстов на английском языке, получаем приближение $\sum_{i=0}^{25} p_i^2 \approx 0,066$. Тем самым для английских текстов x можно пользоваться следующим приближением для индекса совпадения:

$$I_c(x) \approx 0,066.$$

Аналогичные приближения можно получить и для других языков. Так, для русского языка на основании данных из Приложения 1 получаем приближение:

$$I_c(x) \approx 0,053.$$

Приведем значения индексов совпадения для ряда европейских языков, заимствованные из [Fri85]:

Таблица 6. Индексы совпадения европейских языков

Язык	Русский	Англ.	Франц.	Нем.	Итал.	Испан.
$I_c(x) \approx$	0,0529	0,0662	0,0778	0,0762	0,0738	0,0775

Рассуждения, использованные при выводе формулы (17), остаются, очевидно, справедливыми и в случае, когда x — результат зашифрования некоторого открытого текста простой заменой. В этом случае вероятности p_i переставляются местами, но сумма $\sum_{i=0}^{n-1} p_i^2$ остается неизменной.

Предположим, что x — реализация независимых испытаний случайной величины, имеющей равномерное распределение на Z_n . Тогда индекс совпадения вычисляется по формуле

$$I_c(x) = \sum_{i=0}^{n-1} \frac{1}{n^2} = n \cdot \frac{1}{n^2} = \frac{1}{n}.$$

Вернемся к вопросу об определении числа μ .

Пусть $y = y_1 y_2 \dots y_m$ — данный шифртекст. Выпишем его с периодом μ :

Y_1^\downarrow	Y_2^\downarrow	...	Y_μ^\downarrow
y_1	y_2	...	y_μ
$y_{\mu+1}$	$y_{\mu+2}$...	$y_{2\mu}$
$y_{2\mu+1}$	$y_{2\mu+2}$...	$y_{3\mu}$
...

и обозначим столбцы получившейся таблицы через $Y_1^\downarrow, \dots, Y_\mu^\downarrow$. Если μ — это истинная длина ключевого слова, то каждый столбец $Y_i^\downarrow, i \in \overline{1, \mu}$, представляет собой участок открытого текста, зашифрованный простой заменой, определяемой подстановкой

$$\begin{pmatrix} 0 & 1 & 2 & \dots & n-s & \dots & n \\ s & s+1 & s+2 & \dots & 0 & \dots & s-1 \end{pmatrix}, \quad (18)$$

для некоторого $s \in \overline{0, n-1}$ (числа берутся по модулю n).

В силу сказанного выше (для английского языка) $I_c(Y_i^\downarrow) \approx 0,066$ при любом i . С другой стороны, если μ от-лично от длины ключевого слова, то столбцы Y_i^\downarrow будут более “случайными”, поскольку они являются результатом зашиф-рования фрагментов открытого текста некоторым многоалфа-витным шифром. Тогда $I_c(Y_i^\downarrow)$ будет ближе (для английско-го языка) к числу $\frac{1}{26} \approx 0,038$.

Заметная разница значений $I_c(x)$ для осмысленных от-крытых текстов и случайных последовательностей букв (для английского языка — 0,066 и 0,038, для русского языка — 0,053 и 0,030) позволяет в большинстве случаев установить точное значение μ .

Предположим, что на первом этапе мы нашли длину ключевого слова μ . Рассмотрим теперь вопрос о нахождении самого ключевого слова. Для его нахождения можно исполь-зовать так называемый *взаимный индекс совпадения*.

Пусть $x = (x_1, \dots, x_m)$, $y = (y_1, \dots, y_{m'})$ — две строки букв алфавита A . Взаимным индексом совпадения в x и y , обозначаемым $MI_c(x, y)$, называется вероятность того, что случайно выбранная буква из x совпадает со случайно выбранной буквой из y .

Пусть f_0, f_1, \dots, f_{n-1} и $f'_0, f'_1, \dots, f'_{n-1}$ — числа вхождений букв алфавита в x и y соответственно.

Следующее утверждение доказывается точно так же, как и предыдущая теорема.

Теорема. *Взаимный индекс совпадения в x и y вычисляется по формуле*

$$MI_c(x, y) = \frac{\sum_{i=0}^{n-1} f_i \cdot f'_i}{m \cdot m'} \quad (19)$$

Пусть $\bar{k} = (k_1, \dots, k_\mu)$ — истинное ключевое слово.

Попытаемся оценить индексы $MI_c(Y_i^\downarrow, Y_j^\downarrow)$.

Для этого напомним, что Y_s^\downarrow является результатом зашифрования фрагмента открытого текста простой заменой, определяемой подстановкой (18) при некотором s . Вероятность того, что в Y_i^\downarrow и Y_j^\downarrow произвольная пара букв равна 0, имеет, очевидно, вид $p_{n-s_i} \cdot p_{n-s_j}$ (где p_α — вероятность появления буквы α в открытом тексте); вероятность того, что обе буквы есть 1, равна $p_{n-s_i+1} \cdot p_{n-s_j+1}$, и так далее. На основании этого получаем:

$$MI_c(Y_i^\downarrow, Y_j^\downarrow) \approx \sum_{h=0}^{n-1} p_{h-s_i} \cdot p_{h-s_j} = \sum_{h=0}^{n-1} p_h \cdot p_{h+(s_i-s_j)}.$$

Заметим, что сумма в правой части последнего равенства зависит только от разности $(s_i - s_j) \bmod n$, которую назовем *относительным сдвигом* Y_i^\downarrow и Y_j^\downarrow . Заметим также, что

$$\sum_{j=0}^{n-1} P_j \cdot P_{(j+s) \bmod n} = \sum_{j=0}^{n-1} P_j \cdot P_{(j-s) \bmod n}, \quad (20)$$

поэтому Y_i^\downarrow и Y_j^\downarrow с относительными сдвигами s и $n-s$ имеют одинаковые взаимные индексы совпадения. Приведем таблицу значений сумм (20) для английского языка (см. [Sti95]):

Таблица 7. Взаимный индекс совпадения при сдвиге s

Сдвиг s	0	1	2	3	4	5	6
$MI_c(x, y) \approx$	0,066	0,039	0,032	0,034	0,044	0,033	0,036

7	8	9	10	11	12	13
0,039	0,034	0,034	0,038	0,045	0,039	0,043

Аналогичные таблицы можно получить и для других языков.

Обратим внимание на то, что ненулевые “сдвиги” дают взаимные индексы совпадения, изменяющиеся в пределах от 0,032 до 0,045, в то время как при нулевом сдвиге индекс $MI_c(x, y)$ близок к 0,066. Это наблюдение позволяет определить величины относительных сдвигов $s_i - s_j$ столбцов Y_i^\downarrow и Y_j^\downarrow . Для этого заметим, что при некотором значении $s(i, j) \in \overline{0, n-1}$ столбец $Y_j^{s(i, j)^\downarrow}$, полученный из Y_j^\downarrow прибав-

лением к каждому его элементу числа $s(i, j)$ (по модулю n), имеет нулевой относительный сдвиг с Y_i^\downarrow .

Пусть $Y_j^{0\downarrow}, Y_j^{1\downarrow}, \dots, Y_j^{n-1\downarrow}$ — результаты зашифрования Y_j^\downarrow каждой из простых замен (18). Несложно вычислить взаимные индексы

$$MI_c(Y_i^\downarrow, Y_j^{s\downarrow}), \quad 0 \leq s \leq n-1, \quad 1 \leq i < j \leq \mu$$

(всего, таким образом, имеется $C_\mu^2 \cdot n$ значений). Для этого воспользуемся формулой, полученной из (19):

$$MI_c(Y_i^\downarrow, Y_j^{s\downarrow}) = \frac{\sum_{h=0}^{n-1} f_h \cdot f_{h-s}^1}{m \cdot m'}$$

Если s равно $s_i - s_j$ (относительному сдвигу Y_i^\downarrow и Y_j^\downarrow), то взаимный индекс совпадения должен быть (для английского языка) близок к 0,066, так как относительный сдвиг Y_i^\downarrow и $Y_j^{s\downarrow}$ равен нулю. Если же $s \neq s_i - s_j$, то взаимный индекс совпадения должен колебаться в пределах $0,032 \div 0,045$.

Используя изложенный метод, мы сможем связать системой уравнений относительные сдвиги различных пар столбцов Y_i^\downarrow и Y_j^\downarrow . В результате останется 26 вариантов для ключевого слова, из которых можно выбрать наиболее предпочтительный вариант (если ключевое слово является осмысленным).

Следует отметить, что предложенный метод будет эффективным для не слишком больших значений μ . Это объ-

ясняется тем, что для хороших приближений индексов совпадения требуются тексты достаточно большой длины.

§ 6.6. Ошибки шифровальщика

Предположим, что при шифровании текста $T = t_1 t_2 \dots t_l$ модульной гаммой $\Gamma = \gamma_1 \gamma_2 \dots \gamma_l$ произошел сбой и был пропущен (или повторен) некоторый m -значный отрезок гаммы (или открытого текста) $\gamma_k \gamma_{k+1} \dots \gamma_{k+m-1}$ (соответственно $t_k t_{k+1} \dots t_{k+m-1}$). Это могло произойти, например, в случае, когда шифровальщика, производящего работу вручную, вдруг что-то отвлекло, и он допустил ошибку. При расшифровании получатель не сможет полностью восстановить открытый текст. Предположим, что он запросил отправителя повторно передать сообщение, и тот вновь зашифровал тот же текст той же гаммой, но уже без ошибок. Проанализируем последствия таких действий с позиций противника, ведущего перехват.

Рассмотрим случай, когда в результате сбоя оказывается пропущенным отрезок открытого текста. Тогда криптоаналитик противника будет иметь два зашифрованных текста:

$$\begin{aligned} S_1 &= t_1 + \gamma_1, \dots, t_{k-1} + \gamma_{k-1}, t_{k+m} + \gamma_k, \dots, t_l + \gamma_{l-m}, \\ S_2 &= t_1 + \gamma_1, \dots, t_{k-1} + \gamma_{k-1}, t_k + \gamma_k, \dots, t_l + \gamma_l. \end{aligned} \quad (21)$$

Теперь можно выделить следующие два фрагмента в S_1 и S_2 :

$$\begin{aligned} S'_1 &= t_{k+m} + \gamma_k, \dots, t_{l-m} + \gamma_{l-2m}, \\ S'_2 &= t_{k+m} + \gamma_{k+m}, \dots, t_{l-m} + \gamma_{l-m}. \end{aligned} \quad (22)$$

Распишем отрезки открытых текстов в виде последовательности m -грамм:

$$\begin{aligned} \tilde{T} &= (\underbrace{t_k, \dots, t_{k+m-1}}_{\tilde{t}_0}, \underbrace{t_{k+m}, \dots, t_{k+2m-1}}_{\tilde{t}_1}, \dots, \\ \tilde{T}' &= (\underbrace{t_{k+m}, \dots, t_{k+2m-1}}_{\tilde{t}'_1}, \dots \end{aligned}$$

Так же распишем отрезок гаммы:

$$\tilde{\Gamma} = (\underbrace{\gamma_k, \dots, \gamma_{k+m-1}}_{\tilde{\gamma}_0}, \underbrace{\gamma_{k+m}, \dots, \gamma_{k+2m-1}}_{\tilde{\gamma}_1}, \dots$$

Тогда соответствующие отрезки S_1 и S_2 можно представить в виде суммы m -грамм:

$$\begin{aligned} \tilde{S}_1 &= \tilde{t}_1 + \tilde{\gamma}_0, \tilde{t}_2 + \tilde{\gamma}_1, \dots, \\ \tilde{S}_2 &= \tilde{t}'_1 + \tilde{\gamma}_1, \tilde{t}'_2 + \tilde{\gamma}_2, \dots, \end{aligned} \quad (23)$$

где векторы складываются по координатам по модулю n .

Теперь можно выразить разность $\tilde{S}_2 - \tilde{S}_1$:

$$\tilde{S} = \tilde{S}_2 - \tilde{S}_1 = \gamma_1 - \gamma_0, \gamma_2 - \gamma_1, \dots \quad (24)$$

Пусть $\tilde{S} = \bar{s}_1, \bar{s}_2, \dots$. Тогда с помощью (24) образуем последовательность

$$S = 0, s_1, s_1 + s_2, s_1 + s_2 + s_3, \dots,$$

которая, в силу очевидного равенства $\sum_{u=1}^l \bar{s}_u = \bar{\gamma}_l - \bar{\gamma}_0$, имеет вид

$$\bar{S} = \bar{\gamma}_0 - \bar{\gamma}_0, \bar{\gamma}_1 - \bar{\gamma}_0, \bar{\gamma}_2 - \bar{\gamma}_0, \dots \quad (25)$$

Наконец, находим разность $\bar{S}_1 = \tilde{S}_1 - \bar{S}$, которая, с учетом (25), имеет вид

$$\bar{S}_1 = \bar{t}_1 + \bar{\gamma}_0, \bar{t}_2 + \bar{\gamma}_0, \bar{t}_3 + \bar{\gamma}_0, \dots \quad (26)$$

Последовательность (26) представляет собой результат зашифрования отрезка открытого текста $T_1 = t_{k+m}, t_{k+m+1}, \dots$ периодической гаммой

$$\Gamma_1 = \gamma_k, \gamma_{k+1}, \dots, \gamma_{k+m-1}, \gamma_k, \gamma_{k+1}, \dots, \gamma_{k+m-1}, \dots$$

Тем самым мы свели задачу к вскрытию шифра Виженера, решение которой нам уже известно. Поэтому подобная ошибка шифровальщика является недопустимой. Заметим, что при этом число пропущенных знаков m скорее всего не может быть слишком большим.

Нетрудно убедиться в том, что и три другие ошибки (в комбинации: пропуск/повтор отрезка гаммы/открытого текста) также приводят к задаче вскрытия шифра Виженера. Кроме того, легко установить характер такой ошибки непосредственно по текстам двух имеющихся в распоряжении криптоаналитика шифртекстов.

Контрольные вопросы

1. В чем слабость шифра гаммирования с неравновероятной гаммой?
2. Является ли надежным шифрование литературного текста с помощью модульного гаммирования, использующего гамму, два знака которой имеют суммарную вероятность, совпадающую с суммарной вероятностью остальных знаков? Почему?
3. Почему наложение на открытый текст гаммы, представляющей собой периодическую последовательность не-большого периода, не дает надежной защиты?

4. Почему недопустимо использовать дважды одну и ту же гамму (даже случайную и равновероятную!) для зашифрования разных открытых текстов?
5. Почему в качестве гаммы нецелесообразно использовать текст художественного произведения? Можете ли Вы предложить метод вскрытия такого шифра?
6. Можно ли по шифртексту получить приближения для вероятностей знаков гаммы?
7. В чем состоит тест Казисии?
8. Назовите основные этапы работы по вскрытию шифра Виженера.
9. Каким образом рассчитывается индекс совпадения для реального языка?
10. Из каких простых замен «состоит» шифр гаммирования (как многоалфавитный шифр)?

Глава 7

Надежность шифров

§ 7.1. Энтропия и избыточность языка

Рассмотренные нами модели отражают лишь поверхностные свойства открытых текстов. Более глубокие свойства текстов изучаются методами *теории информации*, разработанной К. Шенноном. Речь идет о “*количестве информации*”, содержащейся в сообщении. Для выяснения этого необходимо ввести разумную меру количества информации.

Проследим за рассуждениями К. Шеннона по этому поводу. Они связаны с понятием *энтропии*, определяемой функцией от вероятностного распределения и характеризующей *количество неопределенности* или *информации* в случайном эксперименте. К. Шеннон предложил признать формулу

прирост информации = устраненной неопределенности,

на основании которой неопределенность и информация должны измеряться одной и той же мерой.

К такому выводу можно прийти на примере эксперимента со случайным бросанием монеты. Какова неопределенность того, что в результате очередного бросания монеты выпадет “орел”? Если монета дефектна и при бросании всегда выпадает орлом, никакой неопределенности нет — наоборот, есть полная определенность: обязательно выпадет орел. Максимальной же неопределенность будет, очевидно, в случае, когда монета не имеет дефектов, т. е. с равными вероятностями выпадают обе ее стороны.

Результат бросания монеты можно трактовать иначе. Если монета всегда выпадает орлом, то при проведении очередного эксперимента мы не получим никакой информации: мы

заранее знали об исходе эксперимента. Другими словами, количество информации, извлекаемой из эксперимента, равно нулю. Максимальным количеством получаемой информации будет, очевидно, в случае когда «орел» и «решка» равновероятны.

Пример количественной меры неопределенности случайного эксперимента дает теоретическая физика — такой мерой служит *энтропия*. Применительно к *независимым испытаниям* случайной величины ξ с распределением вероятностей

$$\xi = \begin{pmatrix} a_1 \dots a_n \\ p_1 \dots p_n \end{pmatrix} \quad (1)$$

энтропия $H(\xi)$ определяется формулой

$$H(\xi) = - \sum_{i=1}^n p_i \cdot \log_2 p_i. \quad (2)$$

Единицу измерения энтропии вероятностной схемы предлагает так называемая *теорема кодирования* [Ягл73], утверждающая, что любой исход можно закодировать символами 0 и 1 так, что полученная длина кодового слова будет сколь угодно близка сверху к $H(\xi)$. На основании этого единицей количества информации естественно считать 1 *бит*. Например, количество информации, получаемое при бросании монеты, равно 1 бит, так как орел можно закодировать единицей, а решку — нулем.

Легко видеть, что если $p_i = 1/n$ при всех $i = \overline{1, n}$, то $H_0 = H(\xi) = \log_2 n$. Кроме того, в общем случае имеет место неравенство $H(\xi) \geq 0$, причем $H(\xi) = 0$ в том и только в том случае, когда $p_i = 1$ для некоторого i и $p_j = 0$ для всех $j \neq i$.

Мерой среднего количества информации, приходящейся на одну букву открытого текста языка Λ (рассматриваемого как источник случайных текстов), служит величина H_Λ , называемая *энтропией* языка Λ . Естественно вычислять ее последовательными приближениями: H_0, H_1 , где H_1 — энтропия позначной модели открытого текста, то есть величина (2), в которой p_i совпадает с вероятностью появления буквы a_i в открытом тексте. Для английского языка, $H_0 \approx 4,70$, $H_1 = H(\xi) \approx 4,14$. В качестве следующего, более точного приближения, энтропия вероятностного распределения биграмм, деленная на 2 (нас интересует энтропия на знак). В общем случае следует взять энтропию вероятностной схемы на r -граммах, деленную на r . Соответствующие вычисления для английского языка дают отношения $\frac{H_2}{2} \approx 3,56$,

$\frac{H_3}{3} \approx 3,30$, и так далее. Исследования показывают, что с рос-

том r отношение $\frac{H_r}{r}$ стремится к некоторому пределу. Этот предел и принимается за определение *энтропии* H_Λ языка Λ ⁵:

$$H_\Lambda = \lim_{r \rightarrow \infty} \frac{H_r}{r}. \quad (3)$$

При этом формула

⁵ Существование предела (3) строго доказывается для стационарных эргодических источников сообщений.

$$R_{\Lambda} = 1 - \frac{H_{\Lambda}}{\log_2 n} \quad (4)$$

определяет *избыточность языка* R_{Λ} .

Термин “избыточность языка” возник в связи с тем, что максимальная информация, которую в принципе могла бы нести каждая буква сообщения, равна $H_0 = \log_a n$ (где n — число букв в алфавите). Как было отмечено выше, так было бы в случае, если буквы сообщения появлялись случайно и равновероятно. В то же время средняя энтропия буквы в открытом тексте значительно меньше и, следовательно, буква несет меньше информации, чем $\log_a n$. Величина $\log_a n - H_{\Lambda}$ характеризует, таким образом, неиспользованные возможности в передаче информации с помощью текста, а отношение

$$\frac{\log_a n - H_{\Lambda}}{\log_2 n} = 1 - \frac{H_{\Lambda}}{\log_2 n}$$

в некотором смысле показывает, какую часть букв открытого текста можно опустить без потери содержания. Имеется в виду, что потерянная информация будет восстановлена другими буквами сообщения вследствие закономерностей языка.

К. Шеннон предложил оригинальный метод оценивания отношения H_r/r для осмысленных текстов с позиции меры неопределенности опыта, состоящего в угадывании r -й буквы текста, при условии, что предшествующие его буквы известны [Ягл73]. Эксперимент по угадыванию r -й буквы текста легко может быть поставлен. Для этого достаточно выбрать осмысленный отрезок открытого текста длины $r-1$ и предложить кому-либо угадать следующую букву. Подобный опыт может быть повторен многократно, при этом сложность угадывания r -й буквы может быть оценена с помощью среднего

значения числа попыток F_r , требующихся для нахождения правильного ответа. Ясно, что величины F_r для разных значений r являются определенными характеристиками статистической структуры языка. Очевидно, что среднее число попыток F_r с ростом r может лишь уменьшаться. Прекращение этого уменьшения будет свидетельствовать о том, что соответствующие опыты имеют одинаковую неопределенность, то есть что отвечающая им величина H_r/r практически уже достигла своего предельного значения H_Λ .

Исходя из этих рассуждений, К. Шеннон произвел ряд подобных экспериментов, в которых r принимало значения $\overline{1,15}$ и 100. При этом он обнаружил, что отгадывание сотой буквы по 99 предшествующим заметно более просто, чем угадывание 15-й буквы по 14 предыдущим. Опыты показали, что с ростом r величина H_r/r убывает вплоть до $r \approx 30$, а при дальнейшем росте r она уже практически не меняется.

Согласно исследованиям Б. Б. Пиотровского [Ягл73], имеют место следующие приближения величины H_Λ :

Таблица 8

	H_Λ (бит/букву)		R_Λ (в процентах)	
	Русский язык	Франц. язык	Русский язык	Франц. язык
Язык в целом	1,37	1,40	72,6	70,6
Разговорная речь	1,40	1,50	72,0	68,4
Литературный текст	1,19	1,38	76,2	71,0
Деловой текст	0,83	1,22	83,4	74,4

Из приведенной таблицы видно, что языки имеют весьма большую избыточность. Что означает, например, избыточность, составляющая 75%? Это не означает буквально то, что любые 3 из 4 букв текста можно вычеркнуть без потери информации. Более точно это означает, что при оптимальном кодировании текста (при использовании, например, *кода Хаффмена*, *кода Фано* или другого *оптимального кода* [Шен63]) его можно сжать до четверти длины без потери информации.

Сделаем замечание о другом возможном подходе к определению величины H_Λ для литературных текстов. А. Н. Колмогоров, не согласившись с тем, что теоретико-информационные рассуждения игнорируют вопрос о *смысловом содержании* литературных текстов, предложил так называемый *комбинаторный подход* [Ягл73]. Суть такого подхода к определению энтропии текста состоит в следующем.

Шенноновскую энтропию H_Λ , приходящуюся на букву текста, можно определить тем условием, что для n -буквенного алфавита число текстов длины L , удовлетворяющих заданным статистическим ограничениям, равно (при достаточно больших L) не $n^L = 2^{L \cdot \log_2 n} = 2^{L \cdot H_0}$, как это было бы, если мы имели бы право брать *любые* наборы из L букв, а всего лишь

$$M(L) = 2^{L \cdot H_\Lambda}. \quad (5)$$

По сути, это и есть асимптотика *числа осмысленных открытых текстов* длины L для данного языка Λ . Исходя из этого, можно определить энтропию H_Λ языка формулой

$$H_\Lambda = \lim_{L \rightarrow \infty} \left(\frac{1}{L} \log_2 M(L) \right),$$

не зависящей ни от каких теоретико-вероятностных представлений. Величину $M(L)$ можно оценивать с помощью подсчета числа возможных продолжений литературного текста.

§ 7.2. Расстояние единственности

Попытки определения истинного ключа шифра по данной криптограмме путем ее расшифрования на всех возможных ключах могут привести к тому, что критерий на открытый текст примет несколько претендентов за открытый текст. Это объясняется не только недостатками критерия. При небольших длинах криптограмм результат ее расшифрования может дать несколько осмысленных текстов. Например, криптограмму WNAJW, полученную при использовании *сдвигового шифра* (см. гл. 5) для английского языка, порождают два открытых текста RIVER и ARENA, отвечающих ключам F (=5) и W (=22). При этом один из ключей является *истинным*, а другой — *ложным*. Аналогичная ситуация может иметь место для любого другого шифра.

Найдем оценку для числа ложных ключей. Для этого рассмотрим связь между энтропиями вероятностных распределений $P(X)$, $P(K)$, $P(Y)$, заданных на компонентах X, K, Y произвольного шифра Σ_B (см. гл. 2). Нам понадобятся некоторые дополнительные сведения об *условной энтропии* двух вероятностных распределений.

Пусть имеются дискретные случайные величины ξ и η , заданные вероятностными распределениями $P(\xi)$, $P(\eta)$. Для них можно вычислить совместное распределение $P(\xi, \eta)$ и условные распределения $P(\xi/y)$, $P(\eta/x)$ для любых фиксированных значений $x \in \xi$, $y \in \eta$.

Определим *условную энтропию* $H(\xi/y)$ формулой

$$H(\xi / y) = - \sum_{x \in \xi} p(x/y) \cdot \log_2 p(x/y).$$

Усредненная (по всем $y \in \eta$) величина $H(\xi / y)$ называется *условной энтропией двух вероятностных распределений*:

$$H(\xi / \eta) = - \sum_{y \in \eta} \sum_{x \in \xi} p(y) \cdot p(x/y) \cdot \log_2 p(x/y). \quad (6)$$

(При этом в (6) по определению полагаем $\log_2 p(x/y) = 0$, если $p(x/y) = 0$.)

Величина (6) измеряет среднее количество информации о ξ , обнаруживаемое с помощью η . Известно (см. [Шен63]), что имеет место неравенство $H(\xi / \eta) \leq H(\xi)$, причем равенство $H(\xi / \eta) = H(\xi)$ выполняется тогда и только тогда, когда ξ, η — независимые случайные величины.

Назовем условную энтропию $H(K/Y)$ *неопределенностью шифра* Σ_B по ключу. Она измеряет среднее количество информации о ключе, которую дает шифртекст. Аналогично вводится *неопределенность шифра по открытому тексту* $H(X/Y)$. Эти величины являются мерой теоретической стойкости шифра. В этом можно убедиться на основании следующих рассуждений.

Минимально возможным значением неопределенности шифра по открытому тексту $H(X/Y)$ является 0. Что можно сказать о шифре в этом случае? Ясно, что равенство $H(X/Y) = 0$ выполняется лишь тогда, когда каждое слагаемое в выражении для $H(X/Y)$ равно нулю:

$$p(y) \cdot p(x/y) \cdot \log_2 p(x/y) = 0$$

для всех x, y . Согласно принятой договоренности это возможно лишь в случаях, когда $(\log_2 p(x/y)) = 0$ или когда

$\log_2 p(x/y) = 0$, то есть если $p(x/y) = 1$ при некоторых x, y . Это означает, что по данному y можно получить существенную информацию о x , что свидетельствует о слабости шифра. Чем больше $H(X/Y)$, тем меньше информации получает противник об открытом тексте по криптограмме. Идеальной является ситуация, когда $H(X/Y) = H(X)$. Именно в этом случае шифр можно было бы назвать *идеальным* или *совершенным*. Заметим, что (в силу сделанного выше замечания) такое представление полностью соответствует определению совершенного по К. Шеннону шифра (см. § 7.3).

Связь между энтропиями компонент шифра дает известная [Шен63] *формула для неопределенности шифра по ключу*:

$$H(K/Y) = H(X) + H(K) - H(Y), \quad (7)$$

полученная К. Шенноном. Эта формула позволит нам получить оценку среднего числа ложных ключей.

Рассмотрим произвольный поточный шифр замены, \sum_B для которого множество X открытых текстов представляет собой множество возможных *осмысленных текстов* в данном алфавите A (например, русском, английском или некотором другом), состоящим из n букв. Зафиксируем некоторое число $L \in \mathbb{N}$ и будем интересоваться числом ложных ключей, отвечающих данной криптограмме $y \in A^L$. Предполагается, что A служит также алфавитом шифрованного текста. Введем обозначение

$$K(y) = \{k \in K : \exists x \in X, E_k(x) = y\}.$$

$K(y)$ есть множество ключей, для каждого из которых y является результатом зашифрования некоторого осмысленного открытого текста длины L .

Если мы располагаем криптограммой y , то число ложных ключей равно $|K(y)| - 1$, так как лишь один из допусти-

мых ключей является истинным. Определим *среднее число ложных ключей* κ_L (относительно всех возможных шифртекстов длины L) формулой,

$$\kappa_L = \sum_{y \in Y} p(y) \cdot (|K(y)| - 1),$$

которая легко приводится к виду

$$\kappa_L = \sum_{y \in Y} p(y) \cdot |K(y)| - 1. \quad (8)$$

Теорема. Для любого рассматриваемого шифра Σ_B с равновероятными ключами при достаточно больших значениях L имеет место неравенство

$$\kappa_L \geq \frac{|K|}{n^{L \cdot R_\Lambda}} - 1, \quad (9)$$

где R_Λ — избыточность данного языка.

Доказательство. Согласно формулам (3) и (4), а также условию $X \subseteq A^L$, при достаточно большом L получаем:

$$H(X) = H_I \approx L \cdot H_\Lambda = L \cdot (1 - R_\Lambda) \cdot \log_2 n. \quad (10)$$

Так как $Y \subseteq A^L$, то

$$H(Y) \leq \log_2 |Y| \leq \log_2 |A^L| = \log_2 n^L = L \cdot \log_2 n.$$

Отсюда, а также из формулы (7) для неопределенности шифра по ключу и формулы (10), получаем:

$$\begin{aligned} H(K/Y) &= H(X) + H(K) - H(Y) \approx \\ &\approx H(K) + L \cdot (1 - R_\Lambda) \cdot \log_2 n - H(Y) \geq \\ &\geq H(K) + L \cdot (1 - R_\Lambda) \cdot \log_2 n - L \cdot \log_2 n = \\ &= H(K) - L \cdot R_\Lambda \cdot \log_2 n. \end{aligned} \quad (11)$$

Найдем соотношение между неопределенностью шифра по ключу и числом ложных ключей. Из определения условной энтропии и равенства (8) получаем:

$$\begin{aligned} H(K/Y) &= \sum_{y \in Y} p(y) \cdot H(K/y) \leq \sum_{y \in Y} p(y) \cdot \log_2 |K(y)| \leq \\ &\leq \log_2 \left(\sum_{y \in Y} p(y) \cdot |K(y)| \right) = \log_2 (\kappa_L + 1). \end{aligned}$$

Мы воспользовались известным *неравенством Йенсена* [Бил69] применительно к функции действительного переменного $f(x) = \log_2 x$.

Так называют следующее утверждение. Если $f(t)$ — заданная на отрезке (a, b) вогнутая (то есть выпуклая вверх) функция действительного переменного, то при любых значе-

ниях $t_1, \dots, t_m \in (a, b)$ и любых $p_1, \dots, p_m \geq 0$, $\sum_{i=1}^m p_i = 1$, вы-

полняется неравенство $\sum_{i=1}^m p_i \cdot f(t_i) \geq f\left(\sum_{i=1}^m p_i \cdot t_i\right)$. Для

выпуклых (вниз) функций знак неравенства — противоположный.

Таким образом, получаем неравенство

$$H(K/Y) \leq \log_2 (\kappa_L + 1), \quad (12)$$

которое имеет место при достаточно большом L . Из (11) и (12) следует, что

$$\log_2 (\kappa_L + 1) \geq H(K) - L \cdot R_\Lambda \cdot \log_2 n.$$

Условие равновероятности ключей шифра максимизирует энтропию $H(K)$. Это дает нам неравенство

$$\log_2(\kappa_L + 1) \geq \log_2|K| - L \cdot R_\Lambda \cdot \log_2 n,$$

потенцируя которое получаем искомое неравенство (9).

Назовем *расстоянием единственности* для шифра Σ_B натуральное число (обозначим его L_0), для которого ожидаемое число ложных ключей κ_Λ равно нулю. По сути, расстояние единственности есть средняя длина шифртекста, необходимая для однозначного восстановления истинного ключа (без каких-либо ограничений на время его нахождения).

Для получения оценки расстояния единственности для поточного шифра с равновероятными ключами воспользуемся неравенством (9). Непосредственно из этого неравенства следует, что

$$\frac{|K|}{\kappa_L + 1} \leq n^{L \cdot R_\Lambda},$$

откуда при $\kappa_L = 0$ получаем $|K| \leq n^{L \cdot R_\Lambda}$ и, следовательно,

$$L \geq \frac{\log_2|K|}{R_\Lambda \cdot \log_2 n}.$$

Минимально возможное значение L в этом неравенстве и принимается за L_0 . Таким образом

$$L_0 = \left\lceil \frac{\log_2|K|}{R_\Lambda \cdot \log_2 n} \right\rceil. \quad (13)$$

Несмотря на некоторую некорректность вывода формулы (13), эта формула тем не менее хорошо согласуется с практикой, и поэтому ею пользуются. Например, для шифра простой замены с параметрами $n = 26$, $|K| = 26!$, $R_\Lambda = 0,5$ формула (13) дает оценку

$$L_0 = \left\lceil \frac{88,4}{0,5 \cdot 4,7} \right\rceil = 38.$$

Это значит, что для английского языка в среднем по криптограмме длиной около 40 можно однозначно определить открытый текст, что приблизительно соответствует практическому опыту.

Замечания

1. Следует иметь в виду, что при попытке вскрыть, например, шифр простой замены с использованием *только частот букв* целесообразно оценивать расстояние единственности с учетом оценки энтропии H_Λ величиной H_1 , вычисленной для позначной модели открытого текста. При этом оценка для расстояния единственности может существенно вырасти. При использовании частот биграмм целесообразно H_Λ приблизить величиной H_2 и так далее.
2. В наших выводах о числе ложных ключей и расстоянии единственности для поточного шифра предполагалось, что число ключей $|K|$ рассматриваемого шифра является постоянным. Тем самым из рассмотрения выпадают, например, шифры гаммирования со случайной гаммой. Для них число ключей растет вместе с ростом длины открытого текста и тем самым может потенциально вырасти до бесконечности. Подобные шифры обычно называют *случайными* в отличие от *программных* шифров, для которых число ключей фиксировано и не зависит от длины открытого текста. Таким образом, наши выводы справедливы лишь для программных шифров. Для случайных шифров расстояние единственности потенциально может быть равно $+\infty$. Такие шифры имеют название *идеальных*.
3. Формулу расстояния единственности можно применять и для блочных шифров. В этом случае шифр можно рас-

смагивать как поточный, но в большом алфавите. Естественно, что при этом следует использовать избыточность языка, подсчитанную для n -грамм, где n — длина блока

(грубым приближением для нее служит отношение $\frac{R \wedge}{n}$).

§ 7.3. Стойкость шифров

Надежность или стойкость шифров определяется объемом работы криптоаналитика, необходимой для их вскрытия. Шифрсистема может служить объектом нападения противника, располагающего разного уровня интеллектуальным и вычислительным потенциалом. Нападающий может быть одиночкой-хакером, имеющим персональный компьютер. Задача криптоанализа может интересовать некую фирму, обладающую солидным персоналом и оборудованием. Наконец, речь может идти о работе по добыванию информации мощной государственной организацией типа АНБ США. Возможности потенциального противника определяют требования, предъявляемые к надежности шифрования.

Исходная информация и цели криптоаналитика могут быть разными. Несомненно, основная цель состоит в получении конфиденциальной информации. Целью нападения может служить также примененный секретный ключ, с помощью которого криптоаналитик может вскрывать другие криптограммы. Шифрсистема может быть надежно защищенной от одних угроз и быть уязвимой по отношению к другим. Попытки противника по добыванию зашифрованной информации называют *криптоатаками*.

В криптографии с секретным ключом обычно рассматривают следующие криптоатаки.

Атака на основе шифртекста: криптоаналитик располагает шифртекстами $y_1 = E_{k_1}(x_1), \dots, y_m = E_{k_m}(x_m)$, отвечаю-

щими неизвестным открытым текстам различных сообщений. Требуется определить хотя бы одно из сообщений $x_i, i = 1, m$, (или соответствующий ключ k_i), исходя из необходимого числа m криптограмм, или убедиться в своей неспособности сделать это. В качестве частных случаев возможно совпадение ключей: $k_1 = \dots = k_m$ или совпадение открытых текстов: $x_1 = \dots = x_m$.

Атака на основе известного открытого текста: криптоаналитик располагает парами $(x_1, y_1), \dots, (x_m, y_m)$ открытых и отвечающих им шифрованных текстов. Требуется определить ключ k_i для хотя бы одной из пар. В частном случае, когда $k_1 = \dots = k_m$, требуется определить ключ k или, убедившись в своей неспособности сделать это, определить открытый текст x_{m+1} еще одной криптограммы $y_{m+1} = E_k(x_{m+1})$, зашифрованный на том же ключе.

Атака на основе выбранного открытого текста: эта атака отличается от предыдущей лишь тем, что криптоаналитик имеет возможность выбора открытых текстов x_1, \dots, x_m . Цель атаки та же, что и предыдущей. Подобная атака возможна, например, в случае, когда криптоаналитик имеет доступ к шифратору передающей стороны, или в системах опознавания “свой-чужой”.

Атака на основе выбранного шифртекста: эта атака отличается от второй атаки лишь тем, что криптоаналитик имеет возможность выбора шифртекстов y_1, \dots, y_m . Цель атаки та же, что и во втором случае. Подобная атака возможна, например, в случае, когда криптоаналитик имеет доступ к шифратору принимающей стороны.

Атаки на основе выбранных текстов считаются наиболее опасными. Иногда к указанным атакам добавляют и другие.

Шифр, выдерживающий все возможные атаки, можно признать хорошим или надежным.

Различие в действенности различных криптоатак можно прокомментировать на примере шифра простой замены. Хотя этот шифр легко вскрываем даже при использовании атаки на основе шифртекста, это все-таки требует некоторых усилий. При проведении атаки на основе известного открытого текста задача становится вовсе тривиальной, как только в доступных открытых текстах встретятся все буквы алфавита. Наконец, при атаке на основе выбранного открытого текста ничего не нужно ждать, так как ключ автоматически получается при зашифровании всех букв алфавита.

При проведении любой криптоатаки обычно пользуются общепринятым в криптографии *правилом Керкгоффса*. В книге “Военная криптография”, изданной в 1883 г., он сформулировал шесть следующих требований к системам шифрования:

- 1) система должна быть нераскрываемой, если не теоретически, то практически;
- 2) компрометация системы не должна причинять неудобств ее пользователям;
- 3) секретный ключ должен быть легко запоминаемым без каких-либо записей;
- 4) криптограмма должна быть представлена в такой форме, чтобы ее можно было передать по телеграфу;
- 5) аппаратура шифрования должна быть портативной и такой, чтобы ее мог обслуживать один человек;
- 6) система должна быть простой. Она не должна требовать ни запоминания длинного перечня правил, ни большого умственного напряжения.

Второе из этих правил и стало называться *правилом Керкгоффса*. Суть его состоит в том, что при проведении криптоанализа можно считать известной систему шифрования. Стойкость (или надежность) шифрования должна определяться лишь секретностью ключа шифрования.

Признание всеми этого принципа в криптографии связано с тем, что “шила в мешке не утаишь”. Рано или поздно те или иные сведения об используемой шифрсистеме становятся известными. В военных условиях могут быть захвачены узлы связи с шифртехникой. Могут потерпеть аварию и попасть в руки противника самолет или судно, оборудованные шифр-средствами. Нельзя исключать предательства шифровальщика и т. п.

Тем не менее шифры, используемые специальными службами, всемерно охраняются. Это обусловлено необходимостью дополнительного запаса прочности, поскольку до сих пор создание шифров с *доказуемой стойкостью* является очень сложной проблемой.

Обоснование надежности шифрсистем осуществляется, как правило, экспериментально при моделировании криптоатак с привлечением группы высококвалифицированных специалистов, которым предоставляются благоприятные условия для работы и необходимая техника. На государственном уровне гарантию надежности криптографической защиты дают уполномоченные для этой цели организации. В России такой организацией является ФАПСИ. Любые средства шифрования, используемые государственными организациями, должны иметь сертификат ФАПСИ.

Рассмотрение вопросов надежности шифрования невозможно без введения качественной и количественной мер. В криптографии рассматривают два подхода к стойкости — *теоретическую стойкость* и *практическую* (или *вычислительную*) *стойкость*.

Теоретическая стойкость шифров

При рассмотрении вопроса о теоретической стойкости шифров отвлекаются от реальных временных и сложностных затрат по вскрытию шифра (что определяет подход к практической стойкости). Во главу угла ставится принципиальная

возможность получения некоторой информации об открытом тексте или использованном ключе. Впервые такой подход исследовал К. Шеннон [Шен63]. Он рассматривал уже знакомую нам модель шифра и единственную криптоатаку на основе шифртекста. Проследим за его рассуждениями.

Как мы указывали, конечной целью работы криптоаналитика является текст сообщения или ключ шифрования. Однако весьма полезной может быть даже некоторая вероятностная информация об открытом тексте. Например, уже предположение о том, что открытый текст написан по-английски, предоставляет криптоаналитику определенную априорную информацию об этом сообщении даже до того, как он увидит шифртекст. Так, например, он заранее знает, что слово “hello” является более вероятным началом сообщения, чем, скажем, набор букв “abcde”. Поэтому первая цель криптоанализа состоит в том, чтобы увеличить количество этой априорной информации, относящейся к каждому возможному открытому тексту таким образом, чтобы истинный открытый текст сделать более вероятным после получения шифртекста, хотя, конечно, и не обязательно точным.

Пусть, например, криптоаналитик перехватил текст “abcccd” и знает (или предполагает), что он был зашифрован при помощи шифра простой замены. Этот шифртекст говорит ему о том, что открытый текст состоит из пяти букв, третья и четвертая из которых являются одинаковыми, а остальные отличными от этой буквы и разными. Хотя он не может быть уверенным, что этим словом является “hello” (это может быть еще “lessy” или что-то подобное), тем не менее *апостериорные вероятности* таких открытых текстов возрастают относительно их априорных вероятностей. Криптоаналитик, кроме того, полностью уверен (в предположении, что использовалась именно простая замена) в том, что этот открытый текст не может быть ни словом “after”, ни словом “catch”, и, таким образом, апостериорная вероятность обоих этих открытых

текстов сокращается до нуля, даже вне зависимости от их априорных вероятностей.

Шеннон назвал шифр *совершенным*, если для любого открытого текста знания, которые могут быть получены из соответствующего ему шифртекста, не раскрывают никакой информации об открытом тексте, за исключением, возможно, его длины. Другими словами, для совершенных шифров апостериорные вероятности открытых текстов (вычисленные после получения криптограммы) совпадают с их априорными вероятностями.

Вспомним о введенной нами модели шифра Σ_B , в которой фигурировали распределения вероятностей $P(X), P(K)$. Они как раз и являются наборами априорных вероятностей $p_X(x), x \in X$ и $p_K(k), k \in K$. Будем предполагать, что $p_X(x) > 0, p_K(k) > 0$ для любых $x \in X, k \in K$.

Далее мы будем рассматривать лишь такие шифры, для которых выбор ключа и выбор открытого текста являются независимыми событиями. Это равносильно тому, что распределения $P(X), P(K)$ являются независимыми. Эти распределения естественным образом индуцируют распределение вероятностей $P(Y) = \{p_Y(y), y \in Y\}$ на множестве возможных шифртекстов по формуле

$$p_Y(y) = \sum_{\substack{(x,k) \\ E_k(x)=y}} p_X(x) \cdot p_K(k). \quad (14)$$

Поясним корректность такого определения. Нужно проверить, что

$$\sum_{y \in Y} p_Y(y) = 1.$$

Рассмотрим отображение $f: X \times K \rightarrow Y$, определенное условием $f|_{X \times \{k\}} = E_k$ для любого $k \in K$. Тогда, поскольку

$$f^{-1}(Y) = X \times K,$$

получаем:

$$\begin{aligned} \sum_{y \in Y} p_Y(y) &= \sum_{y \in Y} \sum_{\substack{(x,k) \\ E_k(x)=y}} p_X(x) \cdot p_K(k) = \\ &= \sum_{y \in Y} \sum_{(x,k) \in f^{-1}(y)} p_X(x) \cdot p_K(k) = \sum_{(x,k) \in X \times K} p_X(x) \cdot p_K(k) = \\ &= \sum_{x \in X} \sum_{k \in K} p_X(x) \cdot p_K(k) = \sum_{x \in X} p_X(x) \cdot \sum_{k \in K} p_K(k) = 1. \end{aligned}$$

Естественным образом вводятся и условные вероятности $p_{Y/X}(y/x)$, $p_{Y/K}(y/k)$, определяемые формулами:

$$p_{Y/X}(y/x) = \sum_{\substack{k \in K: \\ E_k(x)=y}} p_K(k), \quad (15)$$

$$p_{Y/K}(y/k) = \sum_{\substack{x \in X: \\ E_k(x)=y}} p_X(x). \quad (16)$$

Несложно проверить, что формулы (15) и (16) задают вероятностные распределения, то есть что при любом $x \in X$

$$\sum_{y \in Y} p_{Y/X}(y/x) = 1,$$

и при любом $k \in K$

$$\sum_{k \in Y} p_{Y/K}(y/k) = 1.$$

С целью упрощения записи нижние индексы в обозначениях $p_{Y/X}(y/x)$, $p_{Y/K}(y/k)$ будем опускать и записывать их в виде $p(y/x)$, $p(y/k)$ соответственно.

Отметим, что с помощью формулы для условной вероятности

$$p(a/b) = \frac{p(a \cdot b)}{p(b)} \quad (17)$$

мы можем вычислить и условные вероятности $p(x/y)$, $p(k/y)$:

$$\begin{aligned} p(x/y) &= \frac{p_X(x) \cdot p(y/x)}{p_Y(y)}, \\ p(k/y) &= \frac{p_K(k) \cdot p(y/k)}{p_Y(y)}. \end{aligned} \quad (18)$$

Следующее определение лишь формализует предложенный выше подход к теоретической стойкости шифра (только по отношению к атаке на основе единственного шифртекста).

Определение. Назовем шифр \sum_B совершенным, если для любых $x \in X$, $y \in Y$ выполняется равенство

$$p(x/y) = p_X(x). \quad (19)$$

Отметим одно очевидное свойство совершенного шифра.

Утверждение 1. Если шифр \sum_B — совершенный, то

$$|X| \leq |Y| \leq |K|. \quad (20)$$

Доказательство. Первое неравенство, очевидно, имеет место для любого шифра. Если шифр — совершенный, то для любых $x \in X$, $y \in Y$ найдется ключ $k \in K$, такой, что $E_k(x) = y$. В самом деле, в противном случае, согласно (15),

мы бы имели $p(y/x) = 0$, а тогда и, согласно (18), $p(x/y) = 0$. Согласно (19), вероятность $p_X(x)$ также оказывается равной нулю, вопреки нашей договоренности о том, что $p_X(x) > 0$ для любого $x \in X$.

Отсюда следует также, что для любого $x \in X$ выполняется равенство $\{E_k(x), k \in K\} = Y$ и, следовательно, $|Y| \leq |K|$. Утверждение доказано.

В большинстве случаев применяемые на практике шифры обладают свойством $X = Y$. Следуя К. Шеннону, назовем такие шифры *эндоморфными*. К. Шеннону удалось полностью описать эндоморфные совершенные шифры с минимально возможным числом ключей. Согласно (20), это минимально возможное число ключей $|K|$ равно $|Y|$. В несколько более общей форме теорема формулируется следующим образом.

Теорема (К. Шеннон). Пусть Σ_B — шифр, для которого $|X| = |Y| = |K|$. Тогда шифр Σ_B — совершенный тогда и только тогда, когда выполняются два условия:

1) для любых $x \in X, y \in Y$ существует единственный ключ $k \in K$, для которого $E_k(x) = y$;

2) распределение вероятностей $P(K)$ — равномерное, то есть для любого ключа $p_K(k) = \frac{1}{|K|}$.

Доказательство. Пусть шифр Σ_s — совершенный. Согласно доказательству утверждения 1,

$$|\{E_k(x) : k \in K\}| = |Y| = |K|.$$

Поэтому из неравенства $k_1 \neq k_2$ следует неравенство $E_{k_1}(x) \neq E_{k_2}(x)$ для любого $x \in X$. Это доказывает необходимость условия 1).

Пусть $X = \{x_1, \dots, x_N\}$. Зафиксируем произвольный элемент $y \in Y$ и занумеруем ключи так, чтобы $E_{k_i}(x_i) = y$, $i = \overline{1, N}$. Тогда

$$p(x_i / y) = \frac{p(y/x_i) \cdot p_X(x_i)}{p_Y(y)} = \frac{p_K(k_i) \cdot p_X(x_i)}{p_Y(y)}. \quad (21)$$

Так как \sum_B — совершенный шифр, то $p(x_i / y) = p_X(x_i)$. Отсюда и из (21) получаем равенство $p_K(k_i) = p_Y(y)$ для любого $i = \overline{1, N}$, которое доказывает необходимость условия 2).

Пусть условия 1) и 2) выполнены. Тогда, пользуясь для фиксированного элемента $y \in Y$ введенной выше нумерацией ключей, имеем цепочку равенств:

$$p_Y(y) = \sum_{\substack{(x_i, k_i) \\ E_{k_i}(x_i) = y}} p_X(x_i) \cdot p_K(k_i) \stackrel{\text{(усл 1)}}{=} \frac{1}{N} \cdot \sum_{i=1}^N p_X(x_i) = \frac{1}{N},$$

из которой

$$p(x_i / y) = \frac{p_X(x_i) \cdot p(y/x_i)}{p_Y(y)} \stackrel{\text{(усл 2)}}{=} p_X(x_i).$$

Достаточность условий теоремы также доказана.

Обратим внимание на то, что *таблица зашифрования* шифра, удовлетворяющего условиям теоремы Шеннона, согласно условию 1) этой теоремы, является *латинским квадратом*. Поэтому в случае, когда $X = Y = K = Z_n$, по сути дела, *шифры табличного гаммирования* со случайными равно-

вероятными ключами, и только они являются единственными совершенными шифрами.

X/K	x_1	...	x_N
k_1	$E_{k_1}(x_1)$...	$E_{k_1}(x_N)$
...
k_N	$E_{k_N}(x_1)$...	$E_{k_N}(x_N)$

Подчеркнем также, что не только указанные шифры являются совершенными. В качестве примера можно указать следующий не эндоморфный совершенный шифр.

Пример

$$X = \{x_0, x_1\}, Y = \{y_0, y_1, y_2\}, K = \{k_0, k_1, k_2\},$$

$$E_{k_i}(x_j) = y_m, \quad m = (i + j) \bmod 3, \quad p_K(k_i) = 1/3, \quad i = \bar{0}, 2.$$

Теорема Шеннона может быть обобщена и для некоторых других криптоатак. Например, в статье [God90] такое обобщение проводится для криптоатак на основе нескольких шифртекстов, полученных на одном ключе, а также для криптоатак на основе ряда открытых и соответствующих им шифрованных текстов, образованных с помощью одного ключа.

Практическая стойкость шифров

В своей работе [Шен63] К. Шеннон, помимо исследований по теоретической стойкости, рассматривал также вопрос о *практической стойкости* шифров. Он рассуждал следующим образом (рассматривая, как и ранее, криптоатаку на основе одного шифртекста на шифр, не являющийся совершенным).

После того как объем перехвата (y) превзойдет расстояние единственности (естественно, предполагается, что для рассматриваемого шифра оно существует!), обычно будет существовать единственное решение (x или k) криптограммы. Задача дешифрования и состоит в нахождении этого единственного решения, имеющего высокую вероятность ($p(x/y)$ или $p(k/y)$). До того как объем перехвата достигнет расстояния единственности, задача состоит в нахождении всех решений, имеющих большую вероятность (по сравнению с остальными решениями), и, конечно, в определении вероятностей этих решений.

Несмотря на то, что эти решения можно в принципе найти, поочередно перебирая, например, все ключи при расшифровании, для различных шифров нужно будет затратить для этого весьма различающиеся объемы работы. Средний объем работы $W(N)$, необходимый для определения ключа по криптограмме, состоящей из N букв, измеренный в удобных *элементарных операциях*, К.Шеннон предложил назвать *рабочей характеристикой шифра*. Это среднее значение берется по всем сообщениям и всем ключам с соответствующими им вероятностями. Функция $W(N)$ характеризует средние затраты (временные и материальные), необходимые для практического дешифрования криптограммы. Подобную характеристику можно рассматривать не только для одной конкретной криптоатаки, но и для других постановок задач криптоанализа.

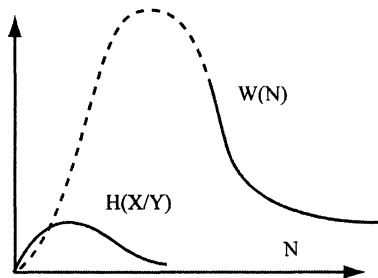


Рис. 20

К. Шеннон привел рабочую характеристику шифра простой замены, сопоставив ее с неопределенностью шифра по открытому тексту (как функцией длины сообщения). Пунктирная линия кривой на рис. 20 относится к области, где имеется несколько возможных решений. По мере увеличения объема перехвата количество необходимой работы быстро уменьшается, стремясь к некоторому асимптотическому значению, которое достигается, когда дополнительные данные уже не уменьшают работы.

Наибольший интерес представляет предельное значение $W(N)$ при $N \rightarrow \infty$ (то есть $W(\infty)$), которое можно рассматривать как среднюю работу по дешифрованию при неограниченном объеме шифртекста. Практическое вычисление $W(\infty)$ представляет собой весьма сложную задачу. В связи с этим практическую стойкость шифра обычно оценивают с помощью величины $W_d(\infty)$, которую можно назвать *достигнутой оценкой* рабочей характеристики. Она определяется средней трудоемкостью наилучшего из известных методов вскрытия данного шифра. Если значение $W_d(\infty)$ вычисляет криптоаналитик, которому известны все имеющиеся в настоящее время методы анализа шифров, то полученная оценка практической стойкости шифра заслуживает доверия, особенно если вычисления сделаны с достаточным “запасом”. Имеется в виду правильный выбор нижнего порога сложности Δ , выделяющий практически стойкие шифры неравенством $W_d(\infty) > \Delta$.

Большое значение имеет математическая модель вычислений, используемая при оценке практической стойкости шифра. По сути, речь идет о модели гипотетической ЭВМ, которой располагает потенциальный противник и которая способна реализовать крупный фрагмент алгоритма вскрытия как одну элементарную операцию. Например, опробование одного ключа $k \in K$ и проверку результата расшифрования по критерию открытого текста (выполняется ли включение

$D_k(y) \in X$?), в принципе, можно принять за одну элементарную операцию.

Более детальное исследование вопросов практической стойкости (выходящее за рамки данной книги) выводит на серьезные проблемы теории сложности алгоритмов и разработки методов криптоанализа.

§ 7.4. Вопросы имитостойкости шифров

Как мы уже отмечали в гл. 2, помимо *пассивных* действий со стороны потенциального противника, состоящих в подслушивании или перехвате передаваемой по каналу связи зашифрованной информации, возможны также его *активные* действия, состоящие в *попытках подмены* или *имитации* сообщения.

Если передается зашифрованное сообщение $y \in Y$ (полученное из открытого текста $x \in X$ на ключе $k \in K$), то противник может его заменить на y' , отличный от y . При этом он будет рассчитывать на то, что на действующем ключе k новая криптограмма при расшифровании будет воспринята как некий осмысленный открытый текст x' , отличный от x . Конечно, это событие может произойти с некоторой вероятностью, и чем больше эта вероятность, тем успешнее будет попытка подмены.

Попытка имитации может быть предпринята противником в том случае, когда линия связи готова к работе (на приеме и передаче установлены действующие ключи), но в рассматриваемый момент никакого сообщения не передается. В таком случае противник может выбрать некий $y \in Y$ и послать его от имени законного отправителя. При этом он будет рассчитывать на то, что на действующем ключе его криптограмма при расшифровании будет воспринята как некий осмысленный открытый текст. Чем больше вероятность этого события, тем успешнее будет попытка имитации.

При теоретическом исследовании активных действий противника обычно ставят его в наиболее благоприятные условия, помещая между отправителем и получателем, как это указано на схеме (см. рис. 21).

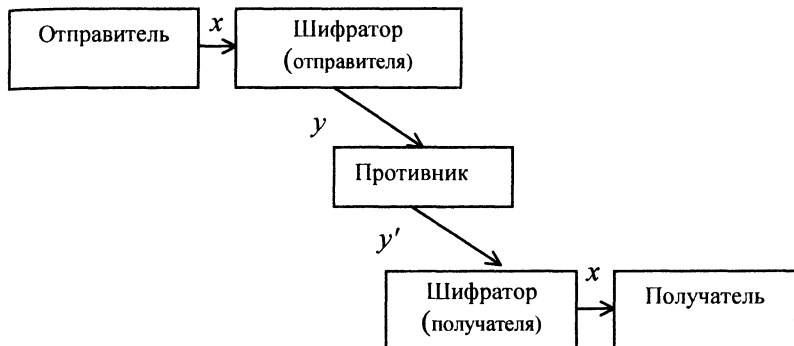


Рис. 21

Имитостойкость шифра определим как его способность противостоять попыткам противника по имитации или подмене. Естественной мерой имитостойкости шифра служит вероятность соответствующего события:

$D_k(y) \in X$ — для попытки имитации сообщения;

$(D_k(y') \in X) \wedge (y' \neq y)$ — для попытки подмены сообщения.

В соответствии с этим введем следующие обозначения:

$$p_{\text{им}} = \max_{y \in Y} p(D_k(y) \in X), \quad (22)$$

$$p_{\text{подм}} = \max_{\substack{y, y' \in Y \\ y' \neq y}} p(D_k(y') \in X), \quad (23)$$

которые назовем соответственно *вероятностью имитации* и *вероятностью подмены*⁶. Полагая, что противник выбирает ту попытку, которая с большей вероятностью приводит к успеху, вводят также *вероятность навязывания* формулой $p_n = \max\{p_{\text{им}}, p_{\text{подм}}\}$.

Для шифров с равновероятными ключами (в рамках введенной в гл. 2 модели) можно получить общие оценки введенных вероятностей.

Утверждение 1. Для шифра Σ_B с равновероятными ключами имеет место достижимая оценка $p_{\text{им}} \geq \frac{|X|}{|Y|}$.

Доказательство. Пусть $K(y) = \{k \in K : D_k(y) \in X\}$. Тогда в силу условия равновероятности ключей

$$p\{D_k(y) \in X\} = \frac{|K(y)|}{|K|}.$$

Согласно (22),

$$p_{\text{им}} = \max_{y \in Y} \frac{|K(y)|}{|K|}. \quad (24)$$

Заметим, что имеет место соотношение

$$\sum_{y \in Y} |K(y)| = |X| \cdot |K|. \quad (25)$$

⁶ В [Сим88] $p_{\text{подм}}$ определяется несколько иначе, однако утверждение 2 для него остается справедливым.

В самом деле, рассмотрим *таблицу зашифрования*, строки которой занумерованы ключами, столбцы — открытыми текстами, а на пересечении строки с номером $k \in K$ и столбца с номером $x \in X$ расположен зашифрованный текст $y = E_k(x)$. Легко видеть, что сумма $\sum_{y \in Y} |K(y)|$ совпадает с числом клеток этой таблицы, откуда следует (25).

Из (25) следует очевидное неравенство

$$\max_{y \in Y} |K(y)| \geq \frac{|X| \cdot |K|}{|Y|},$$

из которого, с учетом (24), получаем требуемое неравенство. Остается доказать его достижимость. Приведем пример шифра, для которого $p_{\text{им}} = \frac{|X|}{|Y|}$.

Рассмотрим шифр \sum_B , определенный условием $E_{k_1}(x_1) = E_{k_2}(x_2)$ тогда и только тогда, когда $k_1 = k_2$ и $x_1 = x_2$. Ясно, что для такого шифра выполняются равенства

$$|Y| = |X| \cdot |K|, \quad |K(y)| = 1 \quad \text{и} \quad \frac{|K(y)|}{|K|} = \frac{|X|}{|Y|},$$

что и требуется.

Обратим внимание на то, что для эндоморфного шифра с равновероятными ключами (например, для шифра гаммирования с равновероятной гаммой) $p_{\text{им}} = 1$. Это означает, что такой шифр максимально уязвим к угрозе имитации сообщения. Поэтому, несмотря на их многие положительные качества, эндоморфные шифры нуждаются в *имитозащите*. Утверждение 2 показывает, что имитостойкость шифра растет пропорционально отношению $\frac{|Y|}{|X|}$. Это объясняет широко ис-

пользуемый для имитозащиты способ введения избыточности в передаваемое сообщение, например, дополнительных “добавок” к передаваемому сообщению типа аутентификаторов или имитовставок.

Утверждение 2. Для шифра Σ_B с равновероятными ключами имеет место достижимая оценка

$$P_{\text{подм}} \geq \frac{|X| - 1}{|Y| - 1}. \quad (26)$$

Доказательство. Пусть $K(y, y') = K(y) \cap K(y')$. Тогда, в силу условия равновероятности ключей, из (24) получаем:

$$P_{\text{подм}} = \max_{\substack{y, y' \in Y \\ y \neq y'}} \frac{|K(y, y')|}{|K(y)|}. \quad (27)$$

Несложно заметить, что выполняется соотношение

$$\sum_{y', (y' \neq y)} |K(y, y')| = (|X| - 1) \cdot |K(y)|, \quad (28)$$

из которого (поскольку сумма в левой части равенства содержит $|Y| - 1$ слагаемых), следует неравенство

$$\max_{y' (y' \neq y)} \frac{|K(y, y')|}{|K(y)|} \geq \frac{|X| - 1}{|Y| - 1}. \quad (29)$$

В силу (27)

$$P_{\text{подм}} \geq \max_{y' (y' \neq y)} \frac{|K(y, y')|}{|K(y)|}.$$

Отсюда и из (29) получаем требуемое неравенство. Остается доказать его достижимость. Для этого построим соответствующий пример.

Попытаемся построить шифр Σ_B , удовлетворяющий условиям

$$\begin{cases} |K(y)| = |K(y')|, & \forall y, y' \in Y, \\ |K(y_1, y_2)| = |K(y'_1, y'_2)|, & \forall y'_1, y'_2 \in Y. \end{cases} \quad (30)$$

Согласно (25) и (28) для такого шифра выполняются соотношения

$$\begin{cases} |Y| \cdot |K(y)| = |X| \cdot |K|, & \forall y \in Y, \\ |K(y)| \cdot (|X| - 1) = |K(y, y')| \cdot (|Y| - 1), & \forall y, y' \in Y, \end{cases}$$

из которых сразу получаем искомое равенство

$$P_{\text{подм}} = \frac{|K(y, y')|}{|K(y)|} = \frac{|X| - 1}{|Y| - 1}.$$

Таблицу зашифрования шифра, удовлетворяющего условиям (30), можно построить с помощью так называемых *ортогональных латинских квадратов*.

Два латинских квадрата размером $n \times n$, составленные из элементов множества $1, 2, \dots, n$, называются *ортогональными*, если при их совмещении друг с другом получается таблица, содержащая (в своих клетках) все n^2 возможных (упорядоченных) пар чисел (i, j) , $i, j \in \overline{1, n}$.

Известно [Хол70], что перестановкой строк и столбцов любой латинский квадрат можно привести к виду, в котором в первом столбце числа $1, 2, \dots, n$ идут в возрастающем порядке.

Такой латинский квадрат называется *полуноормализованным*. Известно также [Хол70], что любую пару ортогональных латинских квадратов можно привести перестановкой строк и столбцов к паре полуноормализованных латинских квадратов.

Рассмотрим теперь таблицу чисел размером $2n \times (n-1)$, полученную из пары полуноормализованных латинских квадратов размером $n \times n$ наложением друг на друга и вычеркиванием получившегося первого столбца. При этом $2n$ строк образуются графически из полученного прямоугольника, если каждую составляющую его пару (i, j) записать в виде $\begin{pmatrix} i \\ j \end{pmatrix}$.

Можно заметить, что ее можно использовать в качестве таблицы зашифрования искомого шифра. Так, для $n = 3$ соответствующий пример изображен на рис. 22–24.

1	2	3
2	3	1
3	1	2

1	3	2
2	1	3
3	2	1

Рис. 22. Полуноормализованные ортогональные латинские квадраты

(1,1)	(2,3)	(3,2)
(2,2)	(3,1)	(1,3)
(3,3)	(1,2)	(2,1)

Рис. 23. Наложение латинских квадратов

В данном примере $|K(y)| = 4$, $|K(y, y')| = 2$.

Построенный таким образом шифр \sum_B имеет следующие параметры: $|X| = n - 1$, $|Y| = n$, $|K| = 2n$. Поэтому для него

$$P_{\text{подм}} = \frac{n-2}{n-1}.$$

	x_1	x_2
k_1	2	3
k_2	3	2
k_3	3	1
k_4	1	3
k_5	1	2
k_6	2	1

Рис. 24. Таблица зашифрования

Для доказательства утверждения 2 остается добавить, что имеет место следующее утверждение [Хол70]: для всякого $n > 6$ существует пара ортогональных латинских квадратов порядка n .

Утверждение доказано.

Итак, вероятность навязывания P_n для шифра с равновероятными ключами удовлетворяет неравенству

$$P_n \geq \frac{|X|}{|Y|}.$$

Поэтому для надежной защиты от подмены или навязывания необходимо, чтобы число $|Y|$ значительно превосходило число $|X|$.

Возникает естественный вопрос, как определить *совершенную имитостойкость* (то есть теоретически наилучшую

защиту от имитации или подмены), достижимую при данной величине $|Y|$ множества допустимых криптограмм и при произвольном распределении $P(K)$ на множестве ключей. Чтобы ответить на этот вопрос, приведем более тонкую оценку для вероятности $p_{\text{им}}$, известную как *граница Симмонса*.

Обозначим через $I(Y, K)$ *взаимную информацию* между Y и K [Сим88], то есть величину, определяемую формулой

$$I(Y, K) = H(Y) - H(Y/K), \quad (31)$$

тогда справедливо следующее.

Утверждение 3. *Имеет место достижимая оценка*

$$\log p_{\text{им}} \geq -I(Y, K). \quad (32)$$

Доказательство. Обозначим через $\delta(y, k)$ индикатор события $d_k(y) \in X$:

$$\delta(y, k) = \begin{cases} 1, & d_k(y) \in X, \\ 0, & d_k(y) \notin X. \end{cases}$$

Тогда для данного ключа $k \in K$ криптограмма y будет принята как допустимая лишь в том случае, когда $\delta(y, k) = 1$.

Согласно формуле полной вероятности вероятность $p(y - \text{доп})$ того, что y окажется допустимой криптограммой, равна

$$p(y - \text{доп}) = \sum_{k \in K} p(k) \cdot p(y - \text{доп}/k),$$

а поскольку

$$p(y - \text{доп} / k) = \begin{cases} 1, & \delta(y, k) = 1, \\ 0, & \delta(y, k) = 0, \end{cases}$$

то

$$p(y - \text{доп}) = \sum_{k \in K} \delta(y, k) \cdot p(k). \quad (33)$$

Представим вероятность $p(y)$ в виде

$$p(y) = \sum_{k \in K} p(k) \cdot p(y / k) = \sum_{k \in K} p(k) \cdot p(y / k) \cdot \delta(y, k). \quad (34)$$

Второе равенство в (34) следует из того, что если

$$p(k) \cdot p(y / k) \neq 0, \text{ то } \delta(y, k) = 1.$$

Теперь мы можем получить границу Симмонса для вероятности $p_{\text{им}}$. Для этого введем функцию

$$Q_y(k) = \frac{p(k) \cdot \delta(y, k)}{p(y - \text{доп})}. \quad (35)$$

Легко видеть, что $Q_y(k)$ — неотрицательная функция от k , которая при суммировании по k (с учетом (33)) дает 1.

Выражая $p(k) \cdot \delta(y, k)$ из (35) и подставляя его в (34), получаем:

$$p(y) = p(y - \text{доп}) \cdot \sum_{k \in K} Q_y(k) \cdot p(y / k). \quad (36)$$

Прологарифмировав (36) и умножив получившееся равенство на $p(y)$, получим выражение

$$p(y) \log p(y) = p(y) \log p(y - \text{доп}) + \\ + p(y) \log \sum_{k \in K} Q_y(k) p(y/k). \quad (37)$$

Второе слагаемое в правой части (37), с учетом (36), можно представить в виде

$$p(y - \text{доп}) \left[\sum_{k \in K} Q_y(k) p(y/k) \right] \log \left[\sum_{k \in K} Q_y(k) p(y/k) \right].$$

Воспользуемся неравенством Иенсена [Бил69], взяв за основу функцию $f(t) = t \log t$:

$$\sum_{k \in K} Q_y(k) p(y/k) \log p(y/k) \geq \\ \geq \left[\sum_{k \in K} Q_y(k) p(y/k) \right] \log \left[\sum_{k \in K} Q_y(k) p(y/k) \right],$$

откуда

$$p(y) \log \sum_{k \in K} Q_y(k) p(y/k) \leq \\ \leq p(y - \text{доп}) \sum_{k \in K} Q_y(k) p(y/k) \log p(y/k) = \\ = \sum_{k \in K} p(k) p(y/k) \delta(y, k) \log p(y/k) = \\ = \sum_{k \in K} p(k) p(y/k) \log p(y/k). \quad (38)$$

Здесь мы воспользовались определением (35), а также тем, что из неравенства $p(y/k)p(k) \neq 0$ следует равенство $\delta(y, k) = 1$.

Теперь, суммируя (37) по y и пользуясь оценкой (38) для второго слагаемого, получаем неравенство

$$-H(Y) \leq \sum_{y \in Y} [p(y) \log p(y - \text{доп})] - H(Y/K),$$

из которого, согласно (31), следует, что

$$\sum_{y \in Y} p(y) \log p(y - \text{доп}) \geq -I(Y, K). \quad (39)$$

Отсюда, наконец, получаем искомое неравенство:

$$\begin{aligned} \log p_{\text{им}} &= \log[\max_{y \in Y} p(y - \text{доп})] = \\ &= \left[\sum_{y \in Y} p(y) \right] \cdot \log[\max_{y \in Y} p(y - \text{доп})] \geq \\ &\geq \sum_{y \in Y} p(y) \cdot \log p(y - \text{доп}) \geq -I(Y, K). \end{aligned}$$

Утверждение 3 доказано.

Из приведенных рассуждений следует, что равенство в (32) достигается в том и только в том случае, когда одновременно выполняются следующие два условия:

1. Вероятность $p(y - \text{доп})$ не зависит от y (поэтому левые части в (32) и (39) равны);
2. Для каждой криптограммы $y \in Y$ вероятность $p(y/k)$ одинакова при всех k , для которых $\delta(y, k) = 1$.

Из утверждения 3 следует оценка для логарифма вероятности навязывания p_n :

$$\log p_n \geq -I(Y, K), \quad (40)$$

причем необходимыми (но уже не достаточными) условиями достижения равенства в (40) являются условия 1 и 2.

Согласно [Сим88], *совершенная имитостойкость* определяется как равенство в (40). Следует, однако, отметить, что даже при достижении совершенной имитостойкости вероятность навязывания мала лишь при большой величине

$I(Y, K)$, то есть в том случае, когда криптограмма дает значительную информацию о ключе. Информация, которую дает Y относительно K , есть мера того, в какой степени ключ используется для обеспечения имитостойкости.

В общем случае не известно, при каких условиях существуют шифры, обеспечивающие совершенную имитостойкость, хотя и имеются соответствующие примеры [Сим88]. Любопытно отметить, что эти примеры свидетельствуют о том, что криптостойкость и имитостойкость шифра являются независимыми свойствами шифра.

§ 7.5. Шифры, не распространяющие искажений

Помимо целенаправленных искажений передаваемой шифрованной информации возможны также искажения, происходящие за счет наличия помех в канале связи. Такие помехи могут привести к искажениям или даже потере отдельных знаков используемого алфавита. Если искаженный знак не является знаком того же алфавита, то на приеме факт искажения легко установить. В противном случае факт искажения может быть установлен лишь при расшифровании, когда искажение в шифртексте приводит к потере части или даже всего открытого текста. Так же проявляется и потеря знаков шифртекста. Последствия искажений шифртекста при передаче могут быть различными для разных типов шифров. Нас будет интересовать вопрос о свойствах самого шифра, позволяющих не распространять искажений при расшифровании. При этом мы ограничимся рассмотрением лишь эндоморфных шифров и искажений, которые либо заменяют знаки алфавита знаками того же алфавита (искажения типа “замена знаков”), либо приводят к потере знаков или появлению дополнительных знаков алфавита (искажения типа “пропуск-вставка знаков”).

Шифры, не распространяющие искажений типа “замена знаков”

Будем рассматривать шифры, для которых открытые и шифрованные тексты являются словами в некотором алфавите A и которые не изменяют длины сообщений при шифровании. В терминологии, введенной в гл. 2, речь пойдет о шифрах, описываемых моделью $\Sigma_a = (X, K, Y, E, D)$, в которой $X = Y = \bigcup_{i=1}^L A^i$, причем для любых $x \in X$ и $k \in K$ длина $y = E_k(x)$ совпадает с длиной x .

Естественной мерой значительности последствий искажений типа “замена знаков” является метрика на множестве сообщений $X = Y$. По-видимому, простейшей является метрика Хэмминга μ , определяемая формулой

$$\mu(x, y) = \sum_{i=1}^{\lambda} \delta(x_i, y_i), \quad (41)$$

где $x = (x_1, \dots, x_\lambda)$, $y = (y_1, \dots, y_\lambda) \in X$,

$$\delta(x_i, y_i) = \begin{cases} 1, & x_i \neq y_i, \\ 0, & x_i = y_i. \end{cases}$$

Как мы знаем, для эндоморфного шифра каждое правило зашифрования E_k представляет собой биекцию $E_k : X \rightarrow X$. В силу этого часто бывает удобно пользоваться *подстановочной моделью шифра* — $\Sigma_n = (X, E)$, в которой множество $E = \{e_k : k \in K\}$ рассматривается как множество подстановок $e : X \rightarrow X$, $e \in E$. При этом индекс k можно опустить, имея в виду наличие взаимно однозначного соответствия между правилами E_k и подстановками e .

Далее в этом параграфе мы будем рассматривать именно подстановочную модель шифра.

Определение 1. Будем говорить, что шифр $\Sigma_n = (X, E)$ не распространяет искажений типа замены знаков, если для любых $x, y \in A^\lambda$, $1 \leq \lambda \leq L$, и любого $e \in E$ выполняется неравенство

$$\mu(e^{-1}x, e^{-1}y) \leq \mu(x, y). \quad (42)$$

Данное определение естественным образом формализует понятие “не распространения искажений”. Неравенство (42) означает, что число искажений в расшифрованном тексте не превышает числа искажений в передаваемом шифрованном тексте. Иногда говорят также, что шифр, удовлетворяющий условию (42), является *помехоустойчивым* (или *помехостойким*) к искажениям рассматриваемого типа.

Утверждение 1. Шифр Σ_n не распространяет искажений типа замены знаков тогда и только тогда, когда для любого $\lambda \in \overline{1, L}$, любых $x, y \in A^\lambda$ и любого $e \in E$ выполняется равенство

$$\mu(e^{-1}x, e^{-1}y) = \mu(x, y). \quad (43)$$

Доказательство. Действие преобразования e на множестве $\Delta^\lambda = A^\lambda \times A^\lambda$, заданное формулой $e(x, y) = (e^{-1}x, e^{-1}y)$, определяет биекцию $\Delta^\lambda \rightarrow \Delta^\lambda$. Поэтому если (x, y) пробегает Δ^λ , то и $(e^{-1}x, e^{-1}y)$ также пробегает Δ^λ . Отсюда следует, что

$$\sum_{(x,y)} \mu(e^{-1}x, e^{-1}y) = \sum_{(x,y)} \mu(x, y)$$

или

$$\sum_{(x,y)} [\mu(x,y) - \mu(e^{-1}x, e^{-1}y)] = 0. \quad (44)$$

Если шифр не распространяет искажений, то, в силу (42), каждое слагаемое в (44) неотрицательно. Поэтому равенство (44) может выполняться лишь в случае, когда имеет место условие (43), что и требуется.

Определение 2. Подстановки $e \in E$, удовлетворяющие условию (43), называются *изометриями на X* .

Утверждение 1 показывает, что для шифров, не распространяющих искажений типа замены знаков, множество E состоит из изометрий. Критерий того, что подстановка $e \in E$ является изометрией на X , получен А. А. Марковым (см. [Баб97]). Для его формулировки введем следующие два преобразования множества X :

$$\Pi_{j_1, \dots, j_\lambda} (a_1, \dots, a_\lambda) = (a_{j_1}, \dots, a_{j_\lambda}), \quad (45)$$

$$R(a_1, \dots, a_\lambda) = (R_1(a_1), \dots, R_\lambda(a_\lambda)), \quad (46)$$

где (j_1, \dots, j_λ) — перестановка чисел $1, 2, \dots, \lambda$; $R_i \in S(A)$ — некоторые фиксированные подстановки множества A , $a_i \in A$, $\lambda \in \overline{1, L}$, $i \in \overline{1, \lambda}$.

Теорема (А. А. Маркова). Биекция $e \in E$ является изометрией на X тогда и только тогда, когда $e = R \cdot \Pi_{j_1, \dots, j_\lambda}$ для подходящих преобразований, определенных формулами (45), (46).

Доказательство. Достаточность условия теоремы очевидна, поскольку преобразования R и $\Pi_{j_1, \dots, j_\lambda}$ являются изометриями и произведение изометрий также является изометрией. Обратимся поэтому к доказательству необходимости условия теоремы.

Для фиксированного элемента $(a_1, \dots, a_\lambda) \in A^\lambda$ введем обозначение

$$A_i^\lambda = \{(a_1, \dots, a_{i-1}, a, a_{i+1}, \dots, a_\lambda), a \in A\}$$

и покажем, что если $e(a_1, \dots, a_\lambda) = (c_1, \dots, c_\lambda)$, то для любого $i \in \overline{1, \lambda}$ найдется $j \in \overline{1, \lambda}$ такое, что

$$e(A_i^\lambda) = (c_1, \dots, c_{j-1}, A, c_{j+1}, \dots, c_\lambda). \quad (47)$$

Любые элементы множества A_i^λ отличаются лишь в одной позиции. Если бы равенство (47) не выполнялось, то нашлись бы $x, y \in e(A_i^\lambda)$ такие, что $\mu(x, y) \geq 2$. Это противоречит условию о том, что e — изометрия на X . Итак, (47) имеет место.

В (47) e осуществляет биекцию по соответствующим координатам:

$$\begin{aligned} e(a_1, \dots, a_{i-1}, a, a_{i+1}, \dots, a_\lambda) = \\ = (c_1, \dots, c_{j-1}, R_i(a), c_{j+1}, \dots, c_\lambda), \end{aligned} \quad (48)$$

где R_i — некоторая биекция множества A . Меняя значение i от 1 до λ , получим перестановку (j_1, \dots, j_λ) множества $\overline{1, \lambda}$, такую, что

$$e(a_1, \dots, a_\lambda) = (R_{k_1}(a_{k_1}), \dots, R_{k_\lambda}(a_{k_\lambda})), \quad (49)$$

где

$$\begin{pmatrix} 1 & 2 & \dots & \lambda \\ j_1 & j_2 & \dots & j_\lambda \end{pmatrix}^{-1} = \begin{pmatrix} 1 & 2 & \dots & \lambda \\ k_1 & k_2 & \dots & k_\lambda \end{pmatrix}.$$

Покажем, что

$$e = R \cdot \Pi_{j_1 \dots j_\lambda}, \quad (50)$$

где $R = (R_1, \dots, R_\lambda)$ — преобразование, полученное в (49).

Для этого введем в рассмотрение “окрестность радиуса 1” точки (a_1, \dots, a_λ) :

$$O_1(a_1, \dots, a_\lambda) = \bigcup_{i=1}^{\lambda} A_i^\lambda.$$

Эта окрестность представляет собой множество точек из A^λ , отстоящих (по метрике Хэмминга) от данной точки не более чем на единицу. Аналогично можно ввести и окрестности $O_m(a_1, \dots, a_\lambda)$ большего радиуса.

Заметим вначале, что (50) выполняется на $O_1(a_1, \dots, a_\lambda)$. Это следует из (47) и (48). То же самое можно выразить в другой форме:

$$\varphi = e \cdot \Pi_{k_1 \ k_\lambda} \cdot R^{-1} = id_{A^\lambda}, \quad (51)$$

где id_{A^λ} — тождественное преобразование множества A^λ .

Докажем, что равенство (51) выполняется на всем A^λ . Для этого рассмотрим произвольную точку $(b_1, \dots, b_\lambda) \in A^\lambda$ и систему окрестностей $B_i, i = \overline{0, \lambda}$:

$$B_0 = O_1(a_1, \dots, a_\lambda), \quad B_1 = O_1(b_1, a_2, \dots, a_\lambda), \dots,$$

$$B_\lambda = O_1(b_1, \dots, b_\lambda)$$

и индукцией по i покажем, что $\varphi = id_{A_\lambda}$ на каждом из них.

При $i = 0$ справедливость (51) уже проверена. Допустим, что утверждение верно для $i = k$ и рассмотрим $i = k + 1$.

Пусть $x \in B_{j+1} = O_1(b_1, \dots, b_{j+1}, a_{j+2}, \dots, a_\lambda)$ и пусть $d = (b_1, \dots, b_j, a_{j+1}, \dots, a_\lambda)$ — “центр” окрестности B_j . Ясно, что $\mu(x, d) \leq 2$. Если при этом $\mu(x, d) = 1$, то $x \in B_j$ и $\varphi(x) = x$ по предположению индукции. Пусть тогда $\mu(x, d) = 2$. Покажем, что для любого $y \in O_2(d)$ имеет место равенство $\varphi(y) = y$.

Без ограничения общности можно считать, что

$$y = (c_1, c_2, b_3, \dots, b_j, a_{j+1}, \dots, a_\lambda),$$

где $c_1 \neq b_1, c_2 \neq b_2$. Рассмотрим точки

$$u = (c_1, b_2, \dots, b_j, a_{j+1}, \dots, a_\lambda), \quad v = (b_1, c_2, b_3, \dots, b_j, a_{j+1}, \dots, a_\lambda).$$

Поскольку $u, v \in B_j$, то по предположению индукции $\varphi(u) = u$, $\varphi(v) = v$. В силу того что φ , очевидно, изометрия, получаем цепочку равенств

$$\mu(\varphi(y), v) = \mu(\varphi(y), \varphi(v)) = \mu(y, v) = 1.$$

Аналогично получаем равенство $\mu(\varphi(y), u) = 1$. Следовательно, $\varphi(y) \in O_1(u) \cap O_1(v)$.

Легко заметить, что пересечение $O_1(u) \cap O_1(v)$ состоит лишь из двух точек — d и y . Поэтому $\varphi(y) \in \{d, y\}$. По предположению индукции $\varphi(d) = d$, а так как φ — биекция, то $\varphi(y) = y$, что и требуется.

Мы доказали равенство (50). Тем самым теорема доказана.

Согласно теореме А.А.Маркова, в классе эндоморфных шифров, не изменяющих длины сообщений, не распространяют искажений типа замены знаков, например, шифры перестановки, поточные шифры однозначной замены, а также их композиции типа шифр замены — шифр перестановки.

Шифры, не распространяющие искажений типа “пропуск-вставка знаков”

При исследовании свойств шифров, не распространяющих искажений типа “пропуск знаков”, так же, как и в предыдущем случае, ограничимся рассмотрением лишь эндоморфных шифров. Отметим также, что исследование вопроса об искажениях типа “вставка знаков” производится совершенно аналогично, поэтому далее пойдет речь лишь об искажениях типа “пропуск знаков”.

Приведем результат характеристики множества E подстановок на множестве $X = Y$, отвечающего подстановочной модели эндоморфного шифра $\Sigma_n = (X, E)$, не распространяющего искажений типа “пропуск знаков”.

В данном случае $X = Y = \bigcup_{\lambda=1}^L A^\lambda$, где A — алфавит сообщений. Пусть ε — бинарное отношение на множестве X , определенное следующим образом. Пусть $a, a' \in X$, тогда

$$a \varepsilon a' \Leftrightarrow \exists j \in \overline{1, \lambda} : a = (a_1, \dots, a_\lambda), a' = (a_1, \dots, a_{j-1}, a_{j+1}, \dots, a_\lambda).$$

Через ε^k обозначим “степень” отношения ε :

$$a \varepsilon^k a' \Leftrightarrow \exists a^1, \dots, a^{k-1} \in X, a \varepsilon a^1, \dots, a^{k-1} \varepsilon a'.$$

Определение 3. Будем говорить, что шифр $\Sigma_n = (X, E)$ не распространяет искажений типа “пропуск знаков”, если для любых $x, y \in X$, любого k , меньшего длины слова x , а также любого $e \in E$ из условия $x \varepsilon^k y$ следует, что $(e^{-1}x) \varepsilon^k (e^{-1}y)$.

Обозначим через π_L отображение множества X в себя, определенное для любого $a = (a_1, \dots, a_\lambda) \in X$ формулой

$$\pi_L(a_1, \dots, a_\lambda) = (\pi(a_1), \dots, \pi(a_\lambda)),$$

где π — некоторая подстановка множества A , а через f — отображение X в себя, меняющее порядок следования букв любого слова на противоположный: $f(a_1, \dots, a_\lambda) = (a_\lambda, \dots, a_1)$.

Приведем без доказательства следующий результат [Глу98,99].

Теорема. Если $\Sigma_n = (X, E)$ — шифр, не распространяющий искажений типа “пропуск знаков”, то для любого $e \in E$ либо $e = \pi_L$, либо $e = \pi_L \cdot f$ (при подходящем $\pi \in S(A)$).

Таким образом, класс шифров, не распространяющих искажений типа “пропуск-вставка знаков”, более беден,

нежели класс шифров, не распространяющих искажений типа замены знаков. Таковыми являются, например, шифры простой замены (или одноалфавитные шифры замены) или шифры простой замены, усложненные перестановкой, осуществляющей изменение порядка следования знаков любого слова на противоположный.

Контрольные вопросы

1. Как определяются энтропия и избыточность языка?
2. Как можно качественно охарактеризовать избыточность языка?
3. Какие тексты на русском языке имеют большую избыточность: литературные или технические?
4. Почему неопределенность шифра по открытому тексту (или по ключу) можно рассматривать как меру теоретической стойкости шифра?
5. Как зависит расстояние единственности для шифра от энтропии языка?
6. Найдите расстояние единственности для шифра Вижнера, который используется для шифрования русских технических текстов с избыточностью 0,5.
7. Какие атаки используются в криптоанализе?
8. Каким образом априорные вероятностные распределения на множествах открытых текстов и ключей индуцируют вероятностное распределение на множестве шифрованных текстов?
9. Какой шифр называется совершенным (для атаки на основе шифртекста)?
10. В каком случае шифр модульного гаммирования является совершенным (для атаки на основе шифртекста)?
11. Верно ли, что лишь шифры табличного гаммирования являются совершенными?
12. Чем отличаются понятия теоретической и практической стойкости шифра?

13. Что такое имитостойкость шифра? Что может служить мерой имитостойкости шифра? Является ли шифр гаммирования имитостойким?
14. Что такое совершенная имитостойкость шифра?
15. Является ли шифр гаммирования шифром, не размножающим искажений типа “замена знаков”, “пропуск знаков”?

Глава 8

Блочные системы шифрования

Используемые в настоящее время системы шифрования делятся на два класса: блочные и поточные системы. Как отмечалось в § 3.2, основной критерий такого разделения — мощность алфавита, над знаками которого производится операция шифрования. Если открытый текст перед шифрованием разбивается на блоки, состоящие из нескольких знаков, то есть исходное сообщение обрабатывается блоками, то мы имеем дело с блочным шифром. Если каждый знак сообщения шифруется отдельно, то такой шифр — поточный.

Разделение шифров на поточные и блочные связано с алгоритмическими и техническими особенностями реализации шифрующих преобразований, использующими возможности существующей элементной базы (разрядность процессоров, быстродействие микросхем, объем памяти компьютера). При увеличении мощности алфавита необходимо исследовать, прежде всего, вопросы о выборе преобразований, реализуемых криптосхемой, и способе их практической реализации, влияющем на эффективность функционирования криптосхемы с точки зрения эксплуатационных характеристик.

Естественно, что точное значение мощности алфавита, начиная с которого шифр следует считать уже не поточным, а блочным, назвать нельзя. Более того, с развитием техники эта характеристика меняется в сторону увеличения. Например, в настоящее время используются 16- и 32-разрядные процессоры, а перспективная шифровальная техника проектируется уже на 64-разрядных процессорах. Поэтому при построении поточных шифров могут быть использованы алфавиты мощностью 2^{32} и 2^{64} .

Следует отметить, что переход от поточного к блочному шифрованию открывает дополнительные возможности для

повышения стойкости криптографических алгоритмов. Как отмечалось в гл. 7, все естественные языки обладают большой информационной избыточностью. Интегральной характеристикой избыточности служит энтропия текста. При шифровании текстов с малой энтропией имеется возможность применения методов, аналогичных методам вскрытия шифра простой замены. С увеличением мощности алфавита энтропия на один знак в “новом алфавите” также увеличивается. Таким образом, использование статистических закономерностей открытых сообщений при проведении криптографического анализа блочных шифров существенно затрудняется. Кроме того, анализ блочных шифров неразрывно связан с исследованием преобразований алфавитов большой мощности, а как правило, увеличение размеров задачи приводит к нелинейному росту трудоемкости ее решения, что также приводит к снижению эффективности известных методов криптографического анализа.

Оборотной стороной сложности анализа блочных криптосхем является трудность обоснования их криптографических качеств и получения доказуемых оценок стойкости. Приходится разрабатывать методы анализа, учитывающие специфику схем блочного шифрования. К недостаткам блочных шифров следует отнести также сложность реализации преобразований алфавитов большой мощности. Однако этот недостаток удастся преодолеть путем использования преобразований специального вида.

§ 8.1. Принципы построения блочных шифров

Как правило, алфавитом, на котором действует блочный шифр, является множество двоичных векторов-блоков открытого текста одинаковой длины (64, 128 и т. д.). Сама реализация преобразований столь больших алфавитов является слож-

ной задачей, а использование преобразований с целью шифрования требует от них еще ряда специальных качеств.

В [Шен63] К. Шеннон сформулировал общий принцип построения шифрующих преобразований — принцип “*перемешивания*”. Суть его состоит в требовании, чтобы применение шифрующего преобразования к наборам аргументов, отличающихся в незначительном числе позиций, приводило к существенному изменению результата. Обеспечить выполнение этого требования в сочетании с простотой реализации конкретного отображения в общем случае представляется затруднительным. Поэтому К. Шеннон предложил реализовывать сложные преобразования в виде суперпозиции нескольких простых некоммутирующих отображений. Подход К. Шеннона, использующий итеративное построение преобразований, в настоящее время является магистральным путем синтеза блочных шифров.

Блочные шифры реализуются путем многократного применения к блокам открытого текста некоторых базовых преобразований. Базовые преобразования должны удовлетворять ряду требований, обусловленных тем, что они, во-первых, должны быть просто реализуемым, в том числе программным способом на ЭВМ, и, во-вторых, при небольшом числе итераций давать аналитически сложные преобразования.

Обычно используются базовые преобразования двух типов — сложные в криптографическом отношении локальные преобразования над отдельными частями шифруемых блоков и простые преобразования, переставляющие между собой части шифруемых блоков. В криптографической литературе первые преобразования получили название “перемешивающих”, а вторые — “рассеивающих”. Качественно можно сказать, что перемешивание усложняет восстановление взаимосвязи статистических и аналитических свойств открытого и шифрованного текстов, а рассеивание распространяет влияние одного знака открытого текста на большое число знаков шифртекста, что позволяет сгладить влияние статистических

свойств открытого текста на свойства шифртекста. Следует отметить, что в данном случае перемешивание понимается в обычном смысле, а не как качественное свойство шифра, введенное К. Шенноном.

Алгоритм шифрования выполняет некоторое число циклов (итераций). Каждый цикл состоит в применении преобразований первого и второго типов. Такой принцип построения дает возможность реализовать каждый цикл шифрования с использованием однотипных узлов, а также выполнять расшифрование путем обработки данных в обратном направлении.

Удобной моделью для реализации базовых преобразований служат *регистры сдвига*. При этом рассеивающие преобразования определяются функциями обратной связи, а перемешивающие — сдвигами информации в регистре.

Получили распространение алгоритмы, в которых осуществляются преобразования над векторами, представляющими собой левую и правую половины содержимого регистра сдвига. Для построения таких алгоритмов часто используется конструкция, называемая *сетью Фейстеля* (Feistel Network) (см. рис. 25).

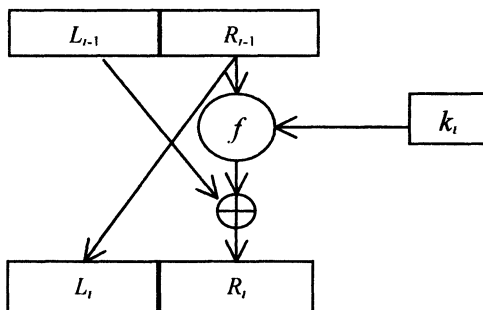


Рис. 25

Преобразование, реализуемое сетью Фейстеля в i -м цикле шифрования, имеет вид

$$\begin{cases} Y_1 = X_2, \\ Y_2 = X_1 \oplus f_i(X_2, k_i), \end{cases} \quad (1)$$

где X — входной блок, разделенный на две половины X_1 и X_2 , а (Y_1, Y_2) — результат шифрования блока X на ключе k_i с помощью функции f_i .

Алгоритм шифрования реализуется несколькими итерациями преобразования сети Фейстеля с использованием ключа k . При этом очередная (i -я) итерация использует в качестве входного блока X результат предыдущей итерации и ключ k_i , вычисляемый определенным образом по ключу k . Функция f_i может зависеть или не зависеть от номера итерации.

Ценность преобразований подобного вида заключается в том, что даже если f_i не является обратимой функцией, преобразование сети Фейстеля обратимо. В самом деле, из (1) сразу следует, что

$$\begin{cases} X_1 = Y_2 \oplus f_i(Y_1, k_i), \\ X_2 = Y_1. \end{cases}$$

§ 8.2. Примеры блочных шифров

Американский стандарт шифрования данных DES

Стандарт шифрования данных DES (*Data Encryption Standard*) опубликован Национальным бюро стандартов США в 1977 г. В 1980 г. этот алгоритм был принят Национальным институтом стандартов и технологий США (НИСТ) в качестве стандарта шифрования данных для защиты от несанкционированного доступа к важной, но не секретной

информации в государственных и коммерческих организациях США.

К достоинствам DES можно отнести простоту ключевой системы, высокую скорость аппаратной и программной реализации, достаточно высокую криптографическую стойкость алгоритма шифрования при заданной длине ключа.

Алгоритм DES, используя комбинацию ряда подстановок и перестановок, осуществляет шифрование 64-битовых блоков данных с помощью 56-битового ключа k . Схема алгоритма DES изображена на рис. 26, 27.

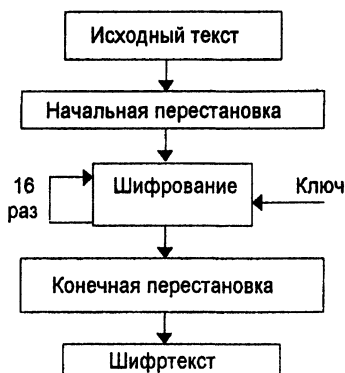


Рис. 26

Процесс шифрования состоит в начальной перестановке битов входного блока, шестнадцати циклах шифрования и, наконец, конечной перестановке битов.

Отметим, что все приводимые ниже таблицы являются стандартными и должны использоваться при реализации алгоритма DES в неизменном виде. Все перестановки и коды в аблицах подобраны разработчиками таким образом, чтобы максимально затруднить процесс вскрытия шифра путем подбора ключа.

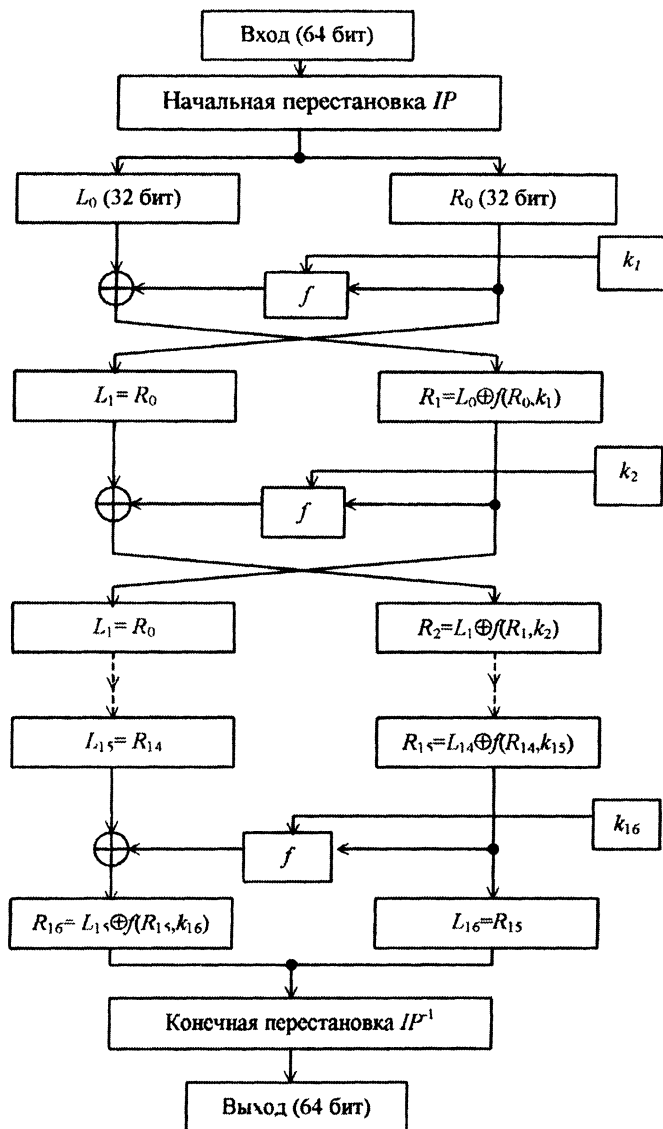


Рис. 27

В приводимом ниже описании алгоритма DES использованы следующие обозначения:

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

\oplus — операция побитового сложения векторов-блоков по модулю 2;

k_i — 48-битовые ключи;

f — функция шифрования;

IP — начальная перестановка степени 64.

При зашифровании очередного блока T (см. рис. 27) его биты подвергаются *начальной перестановке* IP согласно табл. 9.

Таблица 9. Начальная перестановка IP

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

При этом бит 58 блока T становится битом 1, бит 50 — битом 2 и т. д. Полученный после перестановки блок $IP(T)$ разделяется на две половины: L_0 , состоящую из 32 старших бит, и R_0 , состоящую из 32 младших бит.

Затем выполняется итеративный процесс шифрования, состоящий из 16 циклов преобразования Фейстеля. Пусть $T_{i-1} = L_{i-1} R_{i-1}$ — результат $(i-1)$ -й итерации.

Тогда результат i -й итерации $T_i = L_i R_i$ определяется формулами

$$\begin{cases} L_i = R_{i-1}, \\ R_i = L_{i-1} \oplus f(R_{i-1}, k_i), \quad i = \overline{1, 16}. \end{cases} \quad (2)$$

Функция f называется *функцией шифрования*. Ее аргументами являются 32-битовый вектор R_{i-1} и 48-битовый ключ k_i , который является результатом преобразования 56-битового ключа шифра k . Результатом последней итерации является блок $T_{16} = R_{16}L_{16}$. По окончании шифрования осуществляется восстановление позиций битов применением к T_{16} обратной перестановки IP^{-1} .

При расшифровании данных все действия выполняются в обратном порядке, при этом вместо соотношений (2) следует использовать соотношения

$$\begin{cases} R_{i-1} = L_i, \\ L_{i-1} = R_i \oplus f(L_i, k_i), \quad i = \overline{16, 1}, \end{cases}$$

пользуясь которыми можно “спуститься” от L_{16} и R_{16} к L_0 и R_0 .

Схема вычисления значения функции шифрования $f(R_{i-1}, k_i)$ изображена на рис 28.

Для вычисления значения функции f используются: *функция расширения* E ; преобразование S , составленное из восьми преобразований S -блоков S_1, S_2, \dots, S_8 ; перестановка P . Аргументами функции f являются вектор R_{i-1} (32 бита) и вектор k_i (48 бит). Функция E “расширяет” 32-битовый вектор R_{i-1} до 48-битового вектора $E(R_{i-1})$ путем дублирования некоторых битов вектора R_{i-1} , при этом порядок следования битов в $E(R_{i-1})$ указан в табл. 10.

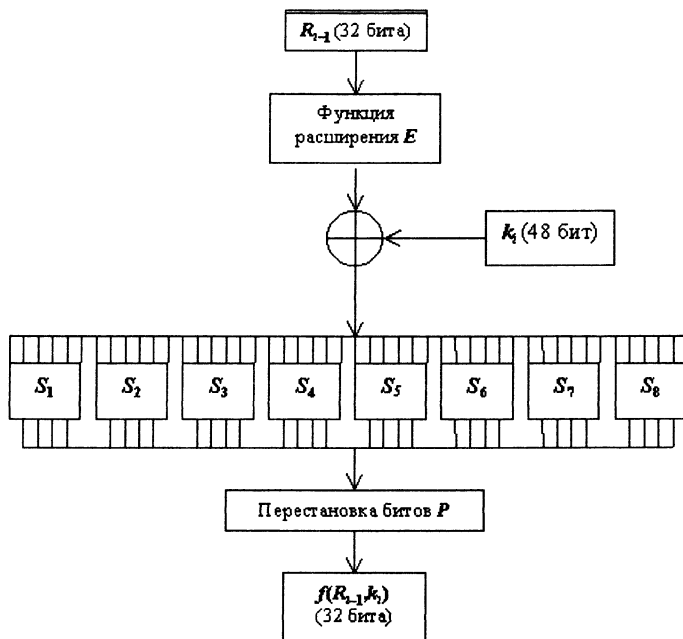


Рис. 28

Таблица 10

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

Первые три бита в $E(R_{i-1})$ — это соответственно биты 32, 1 и 2 вектора R_{i-1} , а последние три бита — это соответственно биты 31, 32, 1 вектора R_{i-1} .

Полученный результат складывается побитно по модулю 2 с текущим значением ключа k_i и затем представляется в виде восьми последовательных 6-битовых блоков B_1, \dots, B_8 :

$$E(R_{i-1}) \oplus k_i = B_1 \dots B_8.$$

Далее каждый из блоков B_j трансформируется в 4-битовый блок B'_j с помощью подходящей таблицы S -блока S_j , список которых приведен в табл. 11.

Преобразование блока B_j в B'_j осуществляется следующим образом. Пусть, например, B_2 равен 111010. Первый и последний разряды B_2 являются двоичной записью числа a , $0 \leq a \leq 3$. Аналогично средние 4 разряда представляют число b , $0 \leq b \leq 15$. В нашем примере $a = 2$, $b = 13$.

Строки и столбцы таблицы S_2 пронумерованы числами a и b . Таким образом, пара (a, b) однозначно определяет число, находящееся на пересечении строки с номером a и столбца с номером b . В данном случае это число равно 3. Записывая его в двоичной форме, получаем B'_2 , равный 0011.

Значение $f(R_{i-1}, k_i)$ теперь получается применением перестановки битов P , заданной таблицей к результирующему 32-битовому блоку $B'_1 B'_2 \dots B'_8$.

Таблица 11

		НОМЕР СТОЛБЦА																
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Н	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	
О	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	
М	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	
Е	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	
Р	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	
С	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	
Т	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	
П	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	

Таблица 12

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

На каждой итерации используется текущее значение ключа k_i (48 бит), получаемое из исходного ключа k следующим образом.

Сначала пользователи выбирают сам ключ k , содержащий 56 случайных значащих битов. Восемь битов, находящихся в позициях 8,16,...,64, добавляются в ключ таким образом, чтобы каждый байт содержал нечетное число единиц. Это используется для обнаружения ошибок при обмене и хранении ключей. Значащие 56 бит ключа подвергаются перестановке, приведенной в табл. 13.

Таблица 13

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

Эта перестановка определяется двумя блоками C_0 и D_0 по 28 бит в каждом (они занимают соответственно верхнюю и нижнюю половины таблицы). Так, первые три бита в C_0 есть соответственно 57, 49, 41 биты ключа. Затем индуктивно определяются блоки C_i и D_i , $i = \overline{1,16}$.

Если уже определены C_{i-1} и D_{i-1} , то C_i и D_i получаются из них одним или двумя левыми циклическими сдвигами согласно табл. 14.

Таблица 14

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

Теперь определим ключи k_i , $1 \leq i \leq 16$. Ключ k_i состоит из 48 битов, выбираемых из битов блока $C_i D_i$ согласно таблице 15. Первыми тремя битами в k_i являются биты 14, 17, 11 из блока $C_i D_i$. Отметим, что 8 из 56 бит (с номерами 9, 18, 22, 25, 35, 38, 43, 54) из $C_i D_i$ отсутствуют в k_i .

Таблица 15

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

Сделаем несколько замечаний.

Нелинейность преобразований, осуществляемых DES, определяется только S -блоками. Их выбор не имеет достаточно обстоятельного обоснования. Высказывались мнения о том, что S -блоки имеют некоторую “лазейку”, позволяющую осуществлять контроль за зашифрованной перепиской. Официальная же версия такова. В 1976 г. АНБ заявило, что выбор S -блоков определен следующими требованиями:

— каждая строка табличного задания каждого S -блока должна являться перестановкой множества $\{0,1,\dots,15\}$;

— S -блоки не должны являться линейными или аффинными функциями своих входов;

— изменение одного бита входа S -блока должно приводить к изменению по крайней мере двух битов выхода;

— для каждого S -блока и любого входа x значение $S(x)$ и $S(x \oplus (0,0,1,1,0,0))$ должны различаться по крайней мере двумя битами.

Криптоанализ DES приводит ко многим нелинейным системам уравнений. Дело в том, что каждый бит блока шифртекста является функцией от всех битов блока открытого текста и ключа. Известные аналитические методы вскрытия DES не дают существенного выигрыша по сравнению с полным перебором всего множества из 2^{56} ключей. К недостаткам алгоритма DES относится небольшое (по современным меркам) число ключей, что дает возможность их полного перебора на быстродействующей вычислительной технике за реальное время.

Официально DES являлся стандартом шифрования данных до 31 декабря 1998 г. В 1997 г. был объявлен конкурс на новый стандарт, который был назван AES (Advanced Encryption Standard). 2 октября 2000 г. Национальный институт стандартов и технологий (НИСТ) США объявил победителя “конкурса AES”. Однако для того, чтобы этот алгоритм завоевал мировое признание, необходимы серьезные исследования его свойств специалистами различных стран.

Стандарт шифрования данных ГОСТ 28147-89

В России установлен единый алгоритм криптографического преобразования данных для систем обработки информации в сетях ЭВМ, отдельных вычислительных комплексах и ЭВМ. Он определяется *ГОСТом 28147-89*. Этот алгоритм предназначен для аппаратной и программной реализации, удовлетворяет необходимым криптографическим требованиям и не накладывает ограничений на степень секретности защищаемой информации. Алгоритм реализует шифрование 64-битовых блоков данных с помощью 256-битового ключа.

Схема алгоритма шифрования представлена на рис. 29.

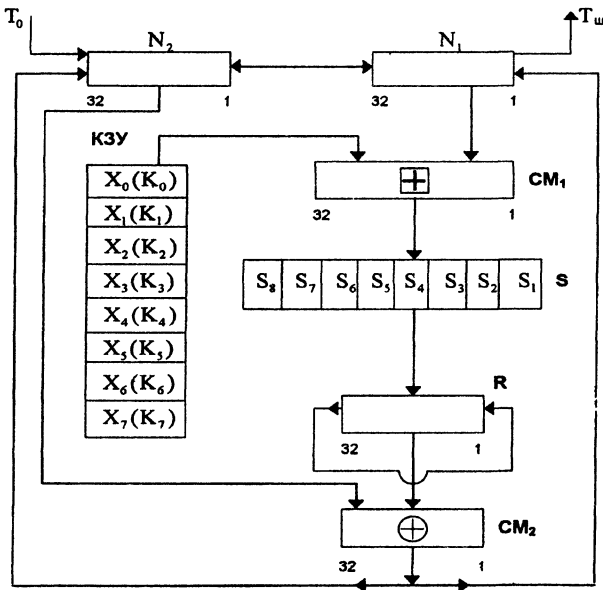


Рис. 29

На приведенной схеме:

N_1, N_2 — 32-разрядные накопители;

CM_1 — сумматор по модулю 2^{32} (операция +);

CM_2 — сумматор по модулю 2 (операция \oplus);

R — 32-разрядный регистр циклического сдвига;

$KЗУ$ — ключевое запоминающее устройство объемом 256 бит, состоящее из восьми 32-разрядных накопителей;

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

Открытые данные, подлежащие зашифрованию, разбивают на 64-разрядные блоки. Процедура зашифрования 64-разрядного блока T_0 включает 32 цикла ($j = \overline{1, 32}$). В ключевое запоминающее устройство вводится 256 бит ключа k , записанного в виде восьми 32-разрядных подключей k_j : $k = k_7 k_6 k_5 k_4 k_3 k_2 k_1$.

Последовательность битов блока

$$T_0 = (a_1(0), \dots, a_{32}(0), b_1(0), \dots, b_{32}(0))$$

разбивается на две половины по 32 бита (взятые в обратном порядке):

$$a(0) = (a_{32}(0), a_{31}(0), \dots, a_1(0)),$$

$$b(0) = (b_{32}(0), b_{31}(0), \dots, b_1(0)).$$

Эти векторы вводятся в накопители N_1 и N_2 перед началом первого цикла зашифрования. В результате начальным заполнением накопителя N_1 является вектор $a(0)$, а начальным заполнением накопителя N_2 — вектор $b(0)$.

Пусть

$$a(j) = (a_{32}(j), \dots, a_1(j)), \quad b(j) = (b_{32}(j), \dots, b_1(j))$$

— заполнения накопителей N_1 и N_2 после j -го цикла зашифрования и f — обозначение функции шифрования. Тогда уравнения шифрования представляются в виде:

$$\begin{cases} a(j) = f(a(j-1) + k_{j-1(\text{mod}8)}) \oplus b(j-1), \\ b(j) = a(j-1), \end{cases}$$

при $j = \overline{1, 24}$,

$$\begin{cases} a(j) = f(a(j-1) + k_{32-j(\text{mod}8)}) \oplus b(j-1), \\ b(j) = a(j-1), \end{cases}$$

при $j = \overline{25, 31}$,

$$\begin{cases} a(32) = a(31), \\ b(32) = f(a(31) + k_0) \oplus b(31). \end{cases}$$

Вычисление значения функции f производится в два этапа. На первом этапе ее 32-битовый аргумент x , поступающий на блок подстановки S , разбивается на восемь последовательных 4-битовых вектора, каждый из которых преобразуется в некоторый 4-битовый вектор соответствующим узлом замены S_i , $i = \overline{1, 8}$. Каждый узел замены можно представить в виде таблицы-перестановки шестнадцати чисел от 0 до 15, представленных в виде двоичных векторов длины 4. Восемь преобразованных S -блоками векторов последовательно соединяются в 32-битовый вектор $S(x)$. На втором этапе с помощью регистра сдвига R производится циклический сдвиг вектора $S(x)$ влево на 11 позиций.

S-блоки представляют собой ключевые элементы, которые являются общими для каждой сети связи. Они должны храниться в секрете.

Результатом зашифрования блока T_0 является блок $T_{ш}$, составленный из заполнений накопителей N_1 и N_2 после 32-го цикла шифрования. При этом биты блока $T_{ш}$ выводятся из накопителей в следующем порядке:

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

Алгоритм шифрования, опеределяемый ГОСТом 28147-89, из-за значительно большей длины ключа является существенно более стойким, нежели алгоритм шифрования DES.

§ 8.3. Режимы использования блочных шифров

Для решения разнообразных криптографических задач блочные шифры используют в нескольких режимах работы. Рассмотрим этот вопрос на примере шифра DES.

Алгоритм DES может использоваться в следующих четырех режимах:

— режим электронной кодовой книги (ECB — *Electronic Code Book*);

— режим сцепления блоков (CBC — *Cipher Block Chaining*);

— режим обратной связи по шифртексту (CFB — *Cipher Feed Back*);

— режим обратной связи по выходу (OFB — *Output Feed Back*).

Режим электронной кодовой книги (ECB) отвечает обычному использованию DES как блочного шифра, осуществляющего некоторую простую замену блоков открытого текста. В режиме сцепления блоков (CBC) каждый блок

$C_i, i \geq 1$, шифртекста перед очередным зашифрованием складывается по модулю 2 со следующим блоком открытого текста M_{i+1} . При этом вектор C_0 полагается равным начальному вектору IV (Initial Vector). Начальный вектор меняется ежедневно и хранится в секрете. Блоки C_1, C_2, \dots вырабатываются по рекуррентной формуле $C_i = DES_k(C_{i-1} \oplus M_i)$.

Схематично этот режим изображен на рис. 30.

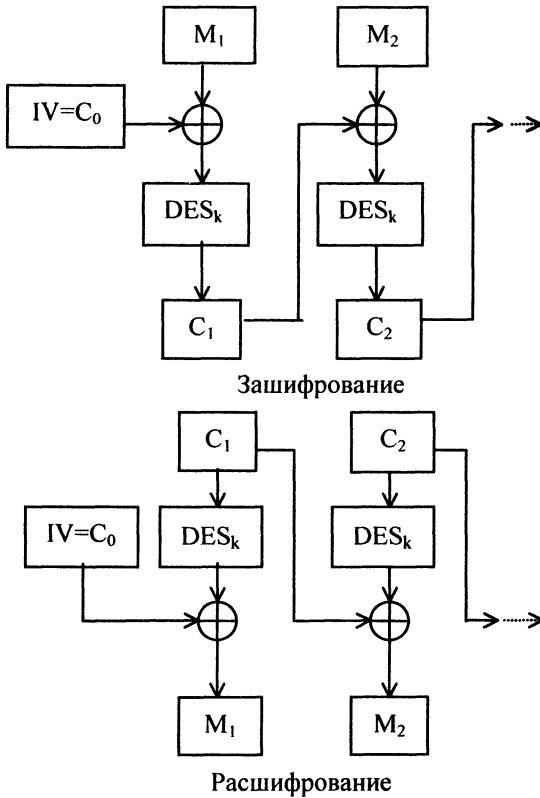


Рис. 30

В режимах CFB и OFB алгоритм DES функционирует по аналогии с “шифром Вернама”, в режиме OFB — как синхронный шифр (см. гл. 9), в режиме CFB — как шифр с самосинхронизацией.

В режиме CFB вырабатывается блочная “гамма” Z_0, Z_1, \dots , причем Z_0 полагается равным начальному вектору IV , а при $i \geq 1$ блоки гаммы удовлетворяют соотношению $Z_i = DES_k(C_{i-1})$. Блоки открытого текста шифруются по правилу $C_i = M_i \oplus Z_i$, $i \geq 1$. Схематично этот режим изображен на рис. 31.

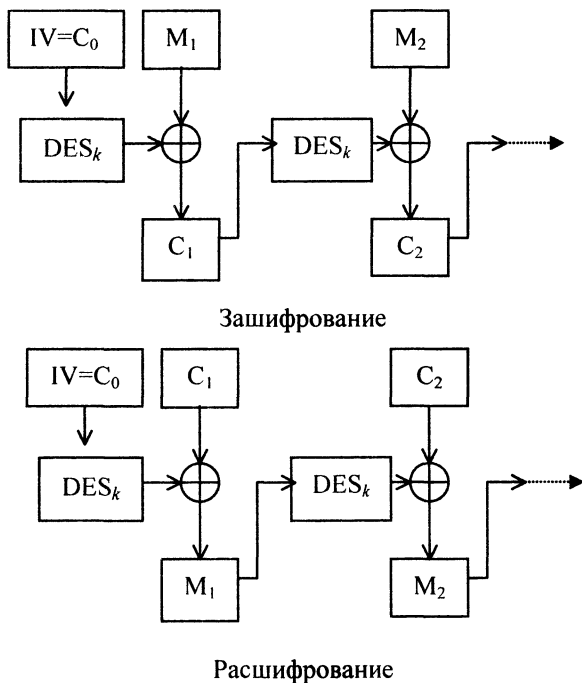


Рис. 31

В режиме OFB вырабатывается блочная “гамма” Z_0, Z_1, \dots , причем Z_0 полагается равным начальному вектору IV , а при $i \geq 1$ блоки гаммы удовлетворяют соотношению $Z_i = DES_k(Z_{i-1})$. Блоки открытого текста шифруются по правилу $C_i = M_i \oplus Z_i, i \geq 1$.

Кроме перечисленных режимов, DES имеет также “режим m -битовой обратной связи”, $1 \leq m \leq 64$. Этот режим оперирует с m -битовыми блоками. Входной блок (64-битовый регистр сдвига) вначале содержит вектор инициализации IV , “выровненный по правому краю” (см. рис. 32).

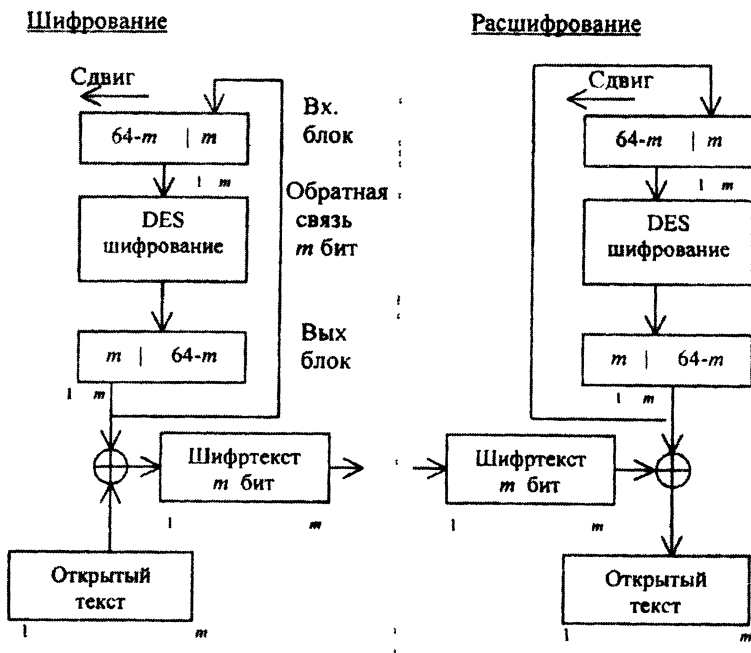


Рис 32

Блоки открытого текста шифруются по правилу $C_i = M_i \oplus P_i$, где P_i — вектор, состоящий из m старших битов блока $DES_k(C_{i-1})$. Обновление заполнения регистра сдвига осуществляется путем отбрасывания старших m битов и дописывания справа вектора P_i .

Указанные режимы имеют свои достоинства и недостатки. Основное достоинство режима ECB — простота реализации. Недостаток — в возможности проведения криптоанализа “со словарем”. Дело в том что вследствие большой избыточности в открытом тексте вполне возможны повторения 64-битовых блоков. Это приводит к тому, что одинаковые блоки открытого текста будут представлены идентичными блоками шифртекста, что дает криптоаналитику возможность при наличии достаточно большого числа пар открытого и шифрованного текста восстанавливать с большой вероятностью блоки открытого текста по шифртексту.

В режимах ECB и OFB искажение при передаче одного 64-битового блока шифртекста C_i приводит к искажению после расшифрования соответствующего блока M_i открытого текста, но не влияет на следующие блоки. Это свойство используется для шифрования информации, предназначенной для передачи по каналам связи с большим числом искажений. Вместе с тем при использовании режима OFB остается открытым вопрос о периоде получаемой выходной гаммы, который в некоторых предположениях может составлять величину порядка 2^{32} .

В режимах CBC и CFB искажение при передаче одного блока шифртекста C_i приводит к искажению на приеме не более двух блоков открытого текста — M_i, M_{i+1} . В то же время изменение блока M_i приводит к тому, что C_i и все последующие блоки шифртекста будут искажены. Это свойство оказывается полезным

для целей аутентификации. Такие режимы применяются для выработки кода аутентификации сообщения (см. гл. 14).

Так, в режиме CBC берется вектор инициализации, состоящий из одних нулей. Затем с помощью ключа k вырабатываются блоки C_1, \dots, C_n шифртекста. Кодом аутентификации сообщения (КАС) служит блок C_n .

Если требуется обеспечить лишь целостность сообщения, отправитель передает блоки M_1, \dots, M_n вместе с C_n . Тогда противнику, желающему изменить сообщение, нужно соответствующим образом изменить и блок C_n . Возможность этого маловероятна, если только противник не располагает секретным ключом k .

Если же нужно обеспечить шифрование и аутентификацию, то отправитель сначала использует ключ k_1 для выработки КАС, затем шифрует последовательность блоков M_1, \dots, M_n , $M_{n+1} = \text{КАС}$ на втором ключе k_2 , и получает последовательность блоков C_1, \dots, C_n, C_{n+1} . Получатель должен сначала расшифровать C_1, \dots, C_n, C_{n+1} на ключе k_2 , а затем проверить (с помощью k), что M_{n+1} — это КАС для M_1, \dots, M_n .

Можно поступить и иначе: сначала использовать k для зашифрования M_1, \dots, M_n , получая C_1, \dots, C_n , а затем k_2 — для получения КАС. Получатель же будет использовать k_2 для проверки КАС, а затем k — для расшифрования C_1, \dots, C_n .

Во всех перечисленных режимах вместо алгоритма DES может быть использован любой алгоритм блочного шифрования, в частности российский стандарт ГОСТ 28147-89.

В российском стандарте также предусмотрено несколько режимов использования: *режим простой замены, режим шифрования с обратной связью и режим гаммирования с обратной связью*, которые являются аналогами соответственно режимов ECB, CBC и CFB. Для того чтобы избавиться от указанной выше проблемы неопределенности длины периода гаммы в режиме OFB, в российском стандарте введен режим гаммирования, при котором блочный шифр используется в качестве узла усложнения некоторой последовательности гарантированного периода. Для выработки этой последовательности обычно применяются линейные регистры сдвига или счетчики по некоторому модулю.

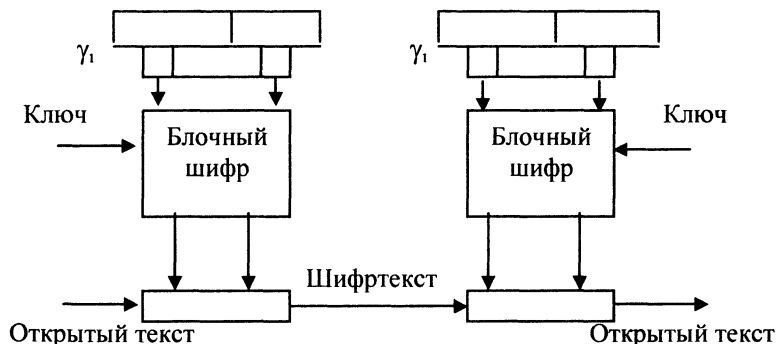


Рис. 33

Уравнение шифрования имеет вид

$$C_i = M_i \oplus F(\gamma_i), \quad i = 1, 2, \dots,$$

где F — преобразование, осуществляемое блочным шифром, γ_i — блоки, сформированные узлом выработки исходной гаммы из начального вектора γ_1 , который передается в начале сообщения в открытом или зашифрованном виде.

Ошибка при передаче, состоящая в искажении символа, приводит при расшифровании к искажению только одного блока, поэтому сохраняются все преимущества шифра гаммирования.

§ 8.4. Комбинирование алгоритмов блочного шифрования

Как было указано выше, алгоритм шифрования DES сегодня уже не является достаточно стойким. Возникает естественный вопрос: нельзя ли использовать его в качестве строительного блока для создания другого алгоритма с более длинным ключом? Уже в 80-х годах была предложена идея многократного шифрования, т.е. использования блочного алгоритма несколько раз с разными ключами для шифрования одного и того же блока открытого текста.

Рассмотрим двукратное шифрование блока открытого текста с помощью двух разных ключей. В этом случае сначала шифруют блок M ключом k_1 , а затем получившийся шифртекст $E_{k_1}(M)$ шифруют ключом k_2 . В результате двукратного шифрования получают криптограмму

$$C = E_{k_2}(E_{k_1}(M)).$$

Если множество преобразований, реализуемых блочным шифром, является группой (относительно операции композиции преобразований), то всегда найдется такой ключ k , что $C = E_k(M)$. В таком случае двукратное шифрование не дает преимуществ по сравнению с однократным шифрованием. В противном случае после двукратного шифрования нужно будет определять оба использованных ключа. Следовательно, трудоемкость перебора ключей по сравнению с однократным шифрованием возводится в квадрат.

Известно, что множество преобразований, реализуемых полной схемой DES, не образует группу [Cam90]. Там же показано, что множество преобразований DES порождает группу подстановок (степени 2^{64}), мощность которой превышает число 10^{2499} . Поэтому многократное шифрование с помощью DES имеет смысл.

Возможны варианты двойного и тройного шифрования с использованием алгоритма DES. В одном из них предлагается шифровать блок M открытого текста три раза с помощью двух ключей k_1 и k_2 . Уравнение шифрования в этом случае имеет вид $C = E_{k_1}(D_{k_2}(E_{k_1}(M)))$. Введение в такую схему операции расшифрования D_{k_2} обеспечивает совместимость схемы со схемой однократного использования DES. Для этого достаточно выбрать одинаковые ключи.

При трехкратном шифровании можно применить три различных ключа. Уравнение шифрования в этом случае принимает вид $C = E_{k_3}(D_{k_2}(E_{k_1}(M)))$. При этом возрастает общая длина результирующего ключа и соответственно возрастает стойкость шифра. Следует отметить, что такое возрастание не может быть безграничным, оно происходит до тех пор, пока суммарное число ключей ($2^{56 \cdot 3}$) не превзойдет общее число преобразований, реализуемых схемой, то есть общее число простых замен, “из которых состоит” данный шифр замены.

§ 8.5. Методы анализа алгоритмов блочного шифрования

Базовым при использовании блочных шифров является режим простой замены. В связи с этим рассмотрим ряд вопросов, связанных с эксплуатацией блочных шифров в этом режиме и влияющих на их криптографическую стойкость.

Как отмечалось выше, серьезным недостатком режима простой замены является то, что зашифрование одинаковых

блоков исходного текста дает идентичные блоки шифротекста. В результате криптоаналитик лишь на основе шифрованных данных может делать выводы о свойствах исходного текста. Примером таких выводов может служить определение факта рассылки писем с одинаковым содержанием в несколько адресов. Если некоторые блоки открытого текста повторились, то во всех зашифрованных сообщениях, независимо от используемых ключей, на одинаковых местах будут встречаться повторяющиеся блоки шифрованных текстов.

Другой пример — передача ключей в зашифрованном виде по линиям связи. Повторение блоков в одном шифрованном тексте показывает, что часть битов в передаваемом ключе повторились и что в дальнейшем трудоемкость перебора ключей при их тотальном опробовании может быть сокращена за счет учета повторений.

Показательно, что при случайном выборе блоков открытого текста их повторение является не столь уж редким событием. Напомним известный *парадокс “дней рождений”*, который заключается в том, что если имеется выборка объема, сравнимого с \sqrt{N} , из множества из N элементов, то вероятность того, что в ней окажутся два одинаковых элемента, сравнима с $1/2$. Этот парадокс показывает, что при случайном выборе блоков открытого текста для получения повтора достаточно взять в среднем порядка \sqrt{N} блоков, где N — общее число блоков, которые теоретически могут встретиться в открытом тексте. Применительно к алгоритмам DES и ГОСТ, которые оперируют с двоичными векторами длины 64, это означает, что в среднем, уже среди 2^{32} блоков открытого текста, будут встречаться повторяющиеся. Следует отметить, что при шифровании осмысленных текстов на естественных языках повторения будут появляться еще чаще, поскольку в осмысленных текстах в силу используемой лексики и грамматических правил встречаются далеко не все сочетания букв и знаков алфавита.

К блочным шифрам, используемым в режиме простой замены, могут быть применены и некоторые методы анализа шифров простой замены в обычных алфавитах. В частности, при достаточно большой длине шифртекста можно применять методы анализа, использующие статистические характеристики открытых текстов. Например, вычисляя частоты появления блоков в зашифрованном тексте и проводя опробование часто повторяющихся блоков, начиная с наиболее вероятных сочетаний знаков в открытом тексте, можно составить словарь соответствия между блоками открытого и зашифрованного текстов. Далее, развивая текст по смыслу с учетом избыточности открытых текстов, найденные блоки открытого текста можно дополнять соседними блоками. При этом одновременно восстанавливается открытый текст и дополняется словарь соответствия. Этот метод эффективен, когда наблюдается стандартность переписки. Например, всегда стандартны заголовки деловых бумаг, юридических и прочих документов.

Еще одна слабость блочных шифров в режиме простой замены при шифровании осмысленных текстов связана с тем фактом, что в открытом тексте могут появляться не все сочетания знаков, что проявляется в фактическом сокращении числа используемых соответствий между блоками открытого и зашифрованного текстов. Однако эта слабость легко устранима, если перед шифрованием применить к открытому тексту процедуру сжатия информации, например использовать стандартные алгоритмы архивации данных.

Следующим моментом, на который следует обратить внимание, является проблема последнего неполного блока данных при шифровании текстов, длины которых не кратны размеру блока. При использовании блочного шифра этот неполный блок должен быть каким-либо образом дополнен до стандартной длины. Если при этом алгоритм дополнения выбран неудачно, например, блок дополняется одними нулями, то при определении соответствующего блока открытого текста у криптоаналитика появляются дополнительные возмож

ности. Эта проблема может показаться надуманной, поскольку относится только к последнему блоку сообщения. Однако именно в конце сообщения обычно ставится подпись и, следовательно, появляются подходы к определению по шифртексту авторства сообщения.

Отдельно остановимся на методах анализа криптографических алгоритмов, основанных на принципе многократного использования блочных шифров. Р. Меркль и М. Хеллман [Mer, Hel] на примере DES показали, как, используя *метод встречи посередине*, можно вскрыть схему двукратного шифрования.

Рассмотрим метод вскрытия блочного шифра при использовании двойного шифрования в общем случае.

Предположим, что известны блок M открытого текста и соответствующий ему блок C шифрованного текста. Алгоритм вскрытия неизвестных ключей k_1 и k_2 состоит из двух этапов.

На первом этапе перебираются все возможные варианты ключа k . Для каждого варианта k ключа k вычисляются значения $ADR(k) = E_k(M)$, после чего значения k помещаются в память по адресу $ADR(k)$.

На втором этапе опробуются возможные варианты ключа k_2 . Для опробуемого варианта k' ключа k_2 вычисляются значения $ADR(k') = D_{k'}(C)$ и производится обращение в память по адресу $ADR(k')$. Если по этому адресу памяти записи отсутствуют, то происходит переход к опробованию следующего варианта k' ключа k_2 . Если же по адресу $ADR(k')$ в памяти хранится ключ k , то образуется допустимая пара ключей (k, k') , удовлетворяющая равенству $C = E_{k'}(E_k(M))$.

Заметим, что в ячейку памяти с номером $ADR(k')$ могут попасть несколько вариантов ключа k (для этого память необходимо соответствующим образом организовать). Для каждого из них пара (k, k') является допустимым ключом.

Несложно заметить, что для реализации данного алгоритма требуется $2|K|$ опробований и столько же операций обращения к памяти, где $|K|$ — общее число ключей шифра.

Таким образом, вместо $|K|^2$ операций, требуемых при полном переборе ключей, затраты метода встречи посередине составляют порядка $4|K|$ операций (операции опробования и обращения к памяти для простоты считают приблизительно равносильными по сложности). Заметим, что такой резкий эффект снижения трудоемкости достигается за счет использования большой (и специальным образом организованной) памяти.

В заключение отметим, что помимо перебора ключей и метода встречи посередине, при исследованиях блочных шифров успешно применяются методы *линейного* и *дифференциального анализа*.

Идея метода линейного анализа состоит в *линеаризации* уравнений шифрования, то есть замене сложных преобразований, описывающих алгоритм шифрования, их приближениями в классе линейных функций. Под приближением в классе линейных функций (или *линейным аналогом*) понимается линейная функция, значения которой для достаточно большого числа наборов аргументов совпадают со значениями данной функции шифрования. Чем выше вероятность совпадения значений линейного аналога со значениями функции шифрования при случайном и равновероятном выборе аргументов, тем лучше качество аналога.

Таким образом, линейный анализ сводит задачу определения ключей к решению системы линейных уравнений, в

которой правые части уравнений известны с некоторой вероятностью. Из общих принципов математической статистики вытекает, что если распределение значений правых частей уравнений системы отлично от равномерного распределения, и имеется достаточно большое число уравнений, то решение такой системы линейных уравнений может быть найдено статистическими методами.

Блочные шифры строятся, как правило, по итеративному принципу. Поэтому даже использование на каждой итерации функций, не имеющих хороших аналогов, не гарантирует отсутствия аналогов в результирующем преобразовании. Проблема построения блочных шифров, для которых удастся доказать отсутствие линейных аналогов, является весьма сложной задачей современной прикладной криптографии.

Методы дифференциального (или иначе, разностного) анализа строятся в предположении, что криптоаналитик имеет для анализа несколько текстов, зашифрованных на одном ключе, и дополнительно предполагается известной информация о том, как различаются между собой открытые тексты (при этом сами открытые тексты могут быть неизвестны). В этом случае криптоаналитик получает информацию о том, как заданные отличия в открытых текстах проявляются в шифр-текстах, или, другими словами, как разность аргументов шифрующего преобразования отражается на изменении его значений. Поскольку шифрующее преобразование однозначно определяется ключом, то информация о зависимостях между разностями значений аргументов и разностями соответствующих значений функции шифрования может быть использована при построении алгебраических и статистических методов вскрытия ключей алгоритма шифрования.

Заметим, что аналогичная ситуация возникает в случае, когда криптоаналитику удастся получить результат зашифрования некоторого сообщения на разных ключах с дополнительной информацией о различиях использованных ключей. В ряде работ некоторые разновидности такого подхода, полу-

чившие общее название *метода дифференциальных искажений*, применялись для вскрытия ключей криптографических алгоритмов, использовавшихся для защиты информации в платежных системах на основе интеллектуальных карт.

§ 8.6. Рекомендации по использованию алгоритмов блочного шифрования

При практическом использовании блочных шифров, помимо чисто криптографических проблем, необходимо учитывать особенности конкретной системы криптографической защиты информации, ее функции и условия эксплуатации. Эти факторы определяют выбор режима шифрования и условий, в которых необходимо оценивать надежность построенной системы защиты.

Основными достоинствами режима простой замены являются простота реализации и тот факт, что изменения одного блока шифртекста вызывают изменения только в одном блоке открытого текста. К недостаткам этого режима, помимо упомянутых выше, можно отнести также неустойчивость системы защиты перед модификацией сообщения, заключающуюся в перестановке блоков шифртекста. Такое нарушение целостности данных может оказаться незамеченным, поскольку при расшифровании каждого блока будет получен осмысленный результат. В особенности это замечание касается передачи формализованных сообщений.

Вследствие отмеченных недостатков блочные шифры редко используются в режиме простой замены для шифрования длинных сообщений. Режим простой замены применяется в основном в системах передачи ключей и в платежных системах, где сообщения состоят из небольшого числа блоков и, следовательно, вероятность шифрования двух одинаковых блоков открытого текста на одном ключе очень мала.

Для практического использования блочных шифров можно предложить модификацию режима простой замены, за-

ключающуюся в замене части битов блоков открытого текста их порядковыми номерами в сообщении. Такой подход несколько снижает скорость передачи информации, поскольку часть блока перестает нести содержательную информацию. Но, с другой стороны, при такой модификации одинаковые блоки открытого текста вследствие различия их порядковых номеров будут представлены различными блоками шифрованного текста.

К достоинствам режима гаммирования следует отнести решение проблемы повторений, возникающих при зашифровании одинаковых блоков сообщения, поскольку в режиме гаммирования одинаковые блоки открытого текста преобразуются в различные блоки шифртекста. Снимается также вопрос о способе дополнения последнего неполного блока данных, так как лишние биты гаммы просто отбрасываются. Перестановка блоков шифртекста также будет обнаружена при расшифровании. Свойство нераспространения ошибок является более выраженным: искажение при передаче одного бита блока приводит при расшифровании в режиме простой замены к искажению всего блока, а в режиме гаммирования — к искажению всего лишь одного бита.

Однако оборотная сторона эффекта нераспространения искажений состоит в том, что появляется возможность целенаправленной модификации шифртекста и открытого текста с точностью до конкретного разряда. Кроме того, необходимо обеспечить уникальность гаммы, что требует для всех сообщений уникальных синхропосылок (иначе возможны повторения используемой гаммы).

Чаще блочные шифры используются в режиме шифрования с обратной связью, когда очередной блок шифртекста зависит не только от ключа, но и от предшествующих блоков шифртекста. Гаммирование с обратной связью устойчиво к перестановкам блоков шифрованного текста и к целенаправленным модификациям шифртекста. В таких системах отсутствует проблема последнего неполного блока данных и одно-

временно снимается острота проблемы обеспечения уникальности синхропосылки.

Контрольные вопросы

1. Каковы с точки зрения криптографии преимущества и недостатки перехода к шифрованию сообщений в алфавитах большой мощности?
2. Как реализуется предложенный К. Шенноном принцип “перемешивания” при практической реализации алгоритмов блочного шифрования?
3. Каковы основные недостатки алгоритма DES и пути их устранения?
4. Как связан “парадокс дней рождений” с криптографическими качествами блочных шифров в режиме простой замены?
5. В каких случаях можно рекомендовать использовать блочный шифр в режиме простой замены?
6. От каких потенциальных слабостей позволяет избавиться использование блочных шифров в режимах шифрования с обратной связью?

Глава 9

Поточные системы шифрования

Как отмечалось в гл. 8, класс блочных шифров наряду с достоинствами имеет определенные недостатки. Один из них связан с эффектом размножения ошибок, снижающим эксплуатационные качества шифра. Еще один недостаток блочных шифров в режиме простой замены связан с возможностью применения *метода анализа "со словарем"*. Такими недостатками не обладают поточные шифры, осуществляющие позначное шифрование в алфавитах небольшой мощности. Кроме того, существующая практика показывает, что пока лишь поточные шифры обеспечивают максимальные скорости шифрования. Это важно при магистральном шифровании больших потоков информации.

Эти, а также многие другие соображения делают поточные шифры в некоторых ситуациях более предпочтительными, чем блочные. Перейдем к подробному изучению вопросов, связанных с поточными шифрами.

§ 9.1. Синхронизация поточных шифрсистем

При использовании поточных шифров простой замены потеря (или искажение) отдельных знаков шифрованного текста при передаче по каналу связи приводит лишь к локальным потерям: все знаки шифртекста, принятые без искажений, будут расшифрованы правильно. Это объясняется тем, что алгоритм шифрования не зависит ни от расположения знаков в тексте, ни от их конкретного вида.

Многоалфавитные поточные шифры также не распространяют ошибок при искажении отдельных знаков шифрованного текста, но оказываются неустойчивыми к пропускам знаков шифрованного текста, поскольку это приводит к не-

правильному расшифрованию всего текста, следующего за пропущенным знаком. Учитывая наличие помех практически во всех каналах передачи данных, в системах криптографической защиты, использующих поточное шифрование, придется заботиться о согласованном порядке применения преобразований при зашифровании и расшифровании, другими словами, решать проблему *синхронизации* процедур зашифрования и расшифрования.

По способу решения этой проблемы поточные шифрсистемы делят на *синхронные* и системы с *самосинхронизацией*.

Для синхронных поточных шифрсистем выбор применяемых шифрующих преобразований однозначно определяется распределителем (см. гл. 3) и зависит только от номера такта шифрования. Каждый знак шифртекста зависит только от соответствующего знака открытого текста и номера такта шифрования и не зависит от того, какие знаки были зашифрованы до или после него. Поэтому применяемое при расшифровании преобразование не зависит от последовательности принятых знаков шифртекста. В этом случае размножение ошибки полностью отсутствует: каждый знак, искаженный при передаче, приведет к появлению только одного ошибочно расшифрованного знака.

Вместе с тем при использовании синхронной системы потеря знака шифртекста приведет к нарушению синхронизации и невозможности расшифрования оставшейся части сообщения. Поэтому для таких систем необходимо предусмотреть специальные процедуры восстановления синхронности работы. Обычно синхронизация достигается вставкой в передаваемое сообщение специальных *маркеров*. В результате этого знак шифртекста, пропущенный в процессе передачи, приводит к неверному расшифрованию лишь до тех пор, пока не будет принят один из маркеров. Другое решение состоит в *реинициализации* состояний, как шифратора отправителя, так и шифратора получателя при некотором предварительно согласованном условии.

Примерами синхронных систем являются регистры сдвига с обратной связью, дисковые шифраторы или шифрмашинка Б. Хагелина С-36 (см. гл. 1).

Поточные шифрсистемы с самосинхронизацией имеют возможность производить правильное расшифрование и в том случае, когда синхронизация передающего и приемного шифраторов нарушается вследствие потери знака шифртекста. Наиболее распространенный режим использования шифрсистем с самосинхронизацией — это (уже знакомый нам) режим обратной связи по шифртексту, при котором текущее состояние системы зависит от некоторого числа N предыдущих знаков шифртекста. В этом режиме потерянный в канале знак влияет на N последовательных состояний. После приема N правильных последовательных знаков из канала связи состояние приемного шифратора становится идентичным состоянию передающего шифратора.

§ 9.2. Принципы построения поточных шифрсистем

В силу ряда естественных причин, связанных с простотой реализации и необходимостью достижения высоких скоростей шифрования, наибольшее распространение получили шифры, осуществляющие побуквенное зашифрование с помощью некоторого множества подстановочных преобразований алфавита открытых сообщений. Другими словами, речь идет об эндоморфных поточных многоалфавитных шифрах замены (см. гл. 3) с множествами шифрвеличин и шифрбозначений, совпадающими с алфавитом открытых сообщений. Далее рассматриваемые поточные шифры мы будем предполагать именно такими.

Для построения многоалфавитного поточного шифра замены необходимо указать его распределитель (см. гл. 3), определяющий порядок использования шифрующих преобразований, и сами эти преобразования, то есть простые замены,

составляющие данный шифр замены. Применительно к рассматриваемому случаю правило зашифрования формулируется следующим образом.

Пусть A — алфавит открытых сообщений, совпадающий с множествами шифрвеличин и шифробозначений, $\{\varphi_\alpha : A \rightarrow A\}$ — совокупность из r биекций множества A , $x = a_1 a_2 \dots a_l$ — произвольный открытый текст, $k \in K$ — выбранный ключ зашифрования. Пусть

$$\psi(k, l) = \alpha_1^{(k)} \dots \alpha_l^{(k)} \quad (\alpha_j^{(k)} \in N_r, j = \overline{1, l}) \quad (1)$$

является распределителем, отвечающим данным значениям $k \in K$, $l \in N$, где $\psi : K \times N \rightarrow N_r^*$ — некоторое отображение в множество $N_r = \{1, 2, \dots, r\}$. Тогда правило зашифрования $E_k(x)$ определяется формулой $E_k(x) = y$, где $y = b_1 b_2 \dots b_l$ и

$$b_j = \varphi_{\alpha_j^{(k)}}(a_j), \quad j = \overline{1, l}. \quad (2)$$

Таким образом, задача построения рассматриваемого шифра сводится к выбору множества шифрующих преобразований $\{\varphi_\alpha\}$ и отображения ψ , задающего распределитель.

В соответствии со сказанным выше, поточная шифрсистема представляется в виде двух основных блоков, отвечающих за выработку распределителя и собственно зашифрование очередного знака открытого текста. Первый блок вырабатывает последовательность номеров шифрующих преобразований, то есть фактически управляет порядком процедуры шифрования. Поэтому этот блок называют *управляющим блоком*, а вырабатываемую им последовательность номеров преобразований — *управляющей последовательностью* (или *управляющей гаммой*).

Второй блок в соответствии со знаком управляющей последовательности реализует собственно алгоритм зашифрования текущего знака. В связи с этим этот блок называют *шифрующим блоком*.

Под номером преобразования следует понимать некий набор символов, достаточный для однозначной идентификации преобразования и удобный с точки зрения практической реализации шифра. Например, номером преобразования может быть двоичный вектор заданной длины.

Достаточно, чтобы в каждом такте шифрующий блок обеспечивал возможность зашифрования лишь текущего знака a_j открытого текста в соответствии с (2). При этом совсем не обязательно строить целиком подстановочное преобразование $\alpha_j^{(k)}$ на всем алфавите A .

Обычно управляющая гамма представляет собой псевдослучайную последовательность, удовлетворяющую некоторой рекуррентной зависимости. В общем случае рекуррентная последовательность (на заданном множестве A) определяется формулой

$$x(i+m) = f(x(i), \dots, x(i+m-1)), \quad i \geq 0,$$

в которой $f: A^m \rightarrow A$ — некоторая функция от m переменных.

Для получения рекуррентных последовательностей используются различные *датчики псевдослучайных чисел*. Наиболее известным таким датчиком с хорошо изученными свойствами является *линейный конгруэнтный генератор* над конечным кольцом или полем. Закон его функционирования представляется в виде

$$x(i+1) = a \cdot x(i) + b, \quad i \geq 0.$$

Обобщением линейного конгруэнтного генератора являются конгруэнтные генераторы, определяемые формулой вида

$$x(i+1) = f(x(i)), \quad i \geq 0, \quad (3)$$

в которой $f: A \rightarrow A$ — произвольное отображение, легко вычисляемое для любого аргумента. Достаточно полно исследованы свойства таких генераторов, задаваемых полиномиальными преобразованиями f .

Изучались также генераторы, определяемые неполиномиальной рекуррентной зависимостью. Примером является целочисленный генератор, основанный на “методе середины квадратов”, для которого вычисление $x(i+1)$ с помощью (3) сводится к отбрасыванию определенного числа знаков из десятичной (или двоичной) записи числа $x(i)^2$.

Исследования подобных, а также других генераторов, определяемых некими алгоритмическими правилами, показывают, что они уступают по своим (необходимым в криптографических приложениях) аналитическим и статистическим качествам рекуррентным последовательностям. Дело в том, что аналитическое представление преобразований позволяет проводить более глубокие исследования и строить последовательности с лучшими криптографическими качествами. В настоящее время большинство датчиков псевдослучайных чисел, в том числе реализованных в программных продуктах ведущих фирм, построены на основе регистров сдвига с линейными функциями обратной связи, или коротко — *линейных регистров сдвига* (ЛРС).

Вид алгоритмов, реализуемых шифрующими блоками поточных шифрсистем, может изменяться в широких пределах. При этом требования к свойствам шифрующего блока в значительной степени зависят от качества управляющей последовательности.

С целью усложнения задачи восстановления управляющей последовательности по виду применяемого шифрующего

преобразования может быть использован способ построения шифрующего блока, при котором многим различным знакам управляющих последовательностей отвечают одинаковые шифрующие преобразования. В таком случае, даже если полностью известно шифрующее преобразование, криптоаналитик не сможет однозначно определить управляющую последовательность и тем самым упростить задачу нахождения ключей криптографического алгоритма.

Если множество шифрующих преобразований $\{\varphi_\alpha\}$ достаточно велико⁷, то можно обеспечить стойкость шифрования даже при повторном использовании ключей. Для этого достаточно, чтобы в множестве $\{\varphi_\alpha\}$ содержались преобразования, переводящие любую пару букв открытого текста в любую пару букв шифрованного текста. Тогда по паре текстов, зашифрованных на одном и том же ключе, нельзя получить информацию об открытых текстах, поскольку любой паре букв шифртекстов может соответствовать произвольная пара букв открытых текстов.

Следует отметить, что указанные качества шифрующего блока повышают криптографическую стойкость, но достигаются за счет избыточности числа простых замен, составляющих данный шифр замены. Действительно, если исключить возможность повторного использования ключей и обеспечить необходимые свойства управляющей последовательности, то можно ограничиться множеством из $|A|$ подстановок $\{\varphi_\alpha\}$, нижние строки которых образуют латинский квадрат. В этом случае мы приходим к введенному ранее шифру табличного гаммирования.

Сформулируем ряд требований, обычно предъявляемых к блокам поточной шифрсистемы, нарушение которых приво-

⁷ Как мы уже знаем, такое множество может совпадать с симметрической группой подстановок на данном алфавите A .

дит к появлению аналитических или статистических слабостей алгоритма шифрования, снижающих его стойкость.

Требования к управляющему блоку:

— период управляющей гаммы должен превышать максимально возможную длину открытых сообщений, подлежащих шифрованию;

— статистические свойства управляющей гаммы должны приближаться к свойствам случайной равновероятной последовательности;

— в управляющей гамме должны отсутствовать простые аналитические зависимости между близко расположенными знаками;

— криптографический алгоритм получения знаков управляющей гаммы должен обеспечивать высокую сложность определения секретного ключа.

Требование к шифрующему блоку:

— применение алгоритма шифрования должно носить универсальный характер и не зависеть от вида шифруемой информации.

Иногда выдвигается дополнительное требование:

— способ построения шифрующего блока должен обеспечивать криптографическую стойкость шифра при перекрытиях управляющей гаммы, в частности при повторном использовании ключей.

Заметим, что выполнение перечисленных требований является необходимым, но не достаточным условием криптографической стойкости поточного шифра.

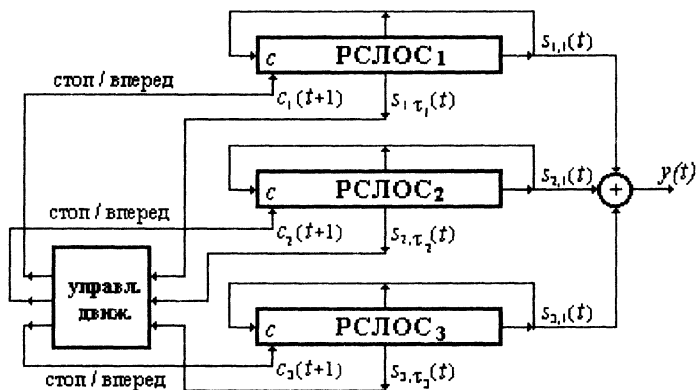
§ 9.3. Примеры поточных шифрсистем

Шифрсистема А5

А5 — шифрсистема гаммирования, применяемая для шифрования телефонных сеансов в европейской системе мобильной цифровой связи GSM (*Group Special Mobile*). В открытой печати криптосхема А5 официально не публикова-

лась. Британская телефонная компания передала всю техническую документацию Брэдфордскому университету. Через некоторое время детали о конструкции А5 стали просачиваться в печать и, в конце концов, появились в INTERNET.

Описание алгоритма приведем по работе [Gol97]. Криптосхема А5 приведена на рис. 34.



Криптосхема А5

Рис. 34

Пусть $f_i(x) = \sum_{l=0}^{r_i} f_{i,l} \cdot x^l$ — функция обратной связи

ЛРС, $i = 1, 2, 3$. Известно, что $r_1 = 19$, $r_2 = 22$, $r_3 = 23$,

$$f_1(x) = x^{19} + x^5 + x^2 + x + 1,$$

$$f_2(x) = x^{22} + x + 1,$$

$$f_3(x) = x^{23} + x^{15} + x^2 + x + 1.$$

Пусть $S_i(0) = (x_i(t))_{t=0}^{r_i-1}$ обозначает начальное заполнение, а $x_i = (x_i(t))_{t=0}^{\infty}$ — соответствующую последовательность максимального периода 2^{r_i-1} , порождаемую ЛРС_{*i*} законом рекурсии

$$x_i(t) = \sum_{l=1}^{r_i} f_{i,l} \cdot x_i(t-l), \quad t \geq r_i.$$

Пусть $S_i(t) = (s_{i,l}(t))_{l=1}^{r_i}$ обозначает состояние ЛРС_{*i*} в момент $t \geq 0$ в схеме с движением “стоп/вперед”, описываемым ниже, и пусть τ_i обозначает средний отвод в ЛРС_{*i*}, используемый для управления движением. Известно, что $\tau_1 = 10$, $\tau_2 = 11$, $\tau_3 = 12$. Тогда управляющая движением регистров последовательность $C(t) = (c(t))_{t=1}^{\infty}$ задается формулой

$$C(t) = g(s_{1,r_1}(t-1), s_{2,r_2}(t-1), s_{3,r_3}(t-1)),$$

где g — это 4-значная функция от трех двоичных переменных, такая, что $g(s_1, s_2, s_3) = (i, j)$, если $s_i = s_j \neq s_k$ при $i < j$ и $k \neq i, j$, и $g(s_1, s_2, s_3) = (1, 2, 3)$, если $s_1 = s_2 = s_3$. Другими словами, знаки управляющей движением последовательности $C(t)$ определяют, какие регистры сдвигаются в t -м такте работы схемы. Ясно, что в каждом такте сдвигаются, по меньшей мере, два регистра. Выходной бит гаммы $y(t)$ вычисляется как сумма

$$y(t) = s_{1,1}(t) + s_{2,1}(t) + s_{3,1}(t), \quad t \geq 1.$$

В системах GSM алгоритм $A5$ используется для защиты информации между абонентом и базовой станцией, так что фактически в сеансе связи двух абонентов шифрование происходит дважды. Это дает возможность использования атаки на основе известного открытого текста. Кроме того, следует отметить, что 64-битовый секретный сеансовый ключ (которым служит совокупность начальных заполнений ЛРС) генерируется с помощью другого алгоритма, исходя из “основного” (master) ключа, специфического для каждого пользователя, и открытого случайного 128-битового ключа, передаваемого в незащищенной форме с базовой станции абоненту. Тем самым успешное вскрытие одного или нескольких сеансовых ключей дает подходы к определению основного ключа пользователя.

Шифрсистема Гиффорда

Д. Гиффорд предложил схему поточного шифра, которая использовалась с 1984 по 1988 г. агентством *Associated Press*. Криптосхема генератора (см. рис. 35) представляет собой 8-байтовый регистр сдвига с линейной функцией обратной связи f и нелинейной функцией выхода h . Ключом являются 64 бита начального заполнения регистра. Схема реализует шифр гаммирования.

В 1994 г. Кейном и Шерманом был предложен метод определения ключа данной криптосхемы, использующий память объема 2^{18} бит и имеющий сложность 2^{27} элементарных операций, что существенно меньше сложности тотального перебора всех 2^{64} начальных состояний. Программа, реализующая данный метод на сети из 8 станций Sparc, находила ключ по одному зашифрованному сообщению достаточной длины в среднем за 4 часа.

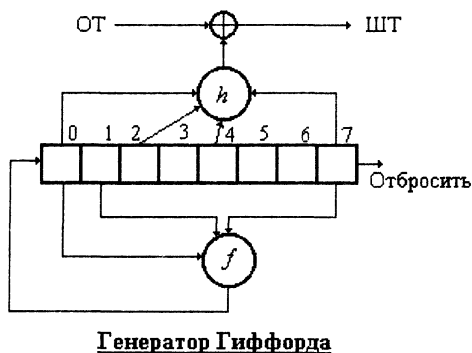


Рис. 35

§ 9.4. Линейные регистры сдвига

Широкое распространение в криптографических приложениях линейных регистров сдвига над конечными полями и кольцами обусловлено целым рядом факторов. Среди них можно отметить:

- использование только простейших операций сложения и умножения, аппаратно реализованных практически на всех вычислительных средствах;

- высокое быстродействие создаваемых на их основе криптографических алгоритмов;

- большое количество теоретических исследований свойств линейных рекуррентных последовательностей (ЛРП), свидетельствующих об их удовлетворительных криптографических свойствах.

Введем ряд определений.

Последовательностью над полем P будем называть любую функцию $u: \mathbb{N}_0 \rightarrow P$, заданную на множестве целых неотрицательных чисел и принимающую значения в поле.

Последовательность u называют *линейной рекуррентной последовательностью* (ЛРП) порядка $m > 0$ над полем P , если существуют константы $f_0, \dots, f_{m-1} \in P$ такие, что

$$u(i+m) = \sum_{j=0}^{m-1} f_j \cdot u(i+j), i \geq 0.$$

ЛРП реализуется схемой линейного регистра сдвига, изображенной на рис. 36.

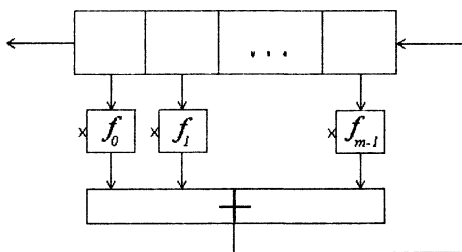


Рис. 36

В очередном такте работы регистра значения, содержащиеся в ячейках его накопителя, умножаются на соответствующие коэффициенты (f_j) и суммируются, после чего происходит (левый) сдвиг информации в регистре, а в освободившуюся крайнюю ячейку записывается вычисленное значение суммы. Заметим при этом, что операции сложения и умножения выполняются в поле P .

Равенство, выражающее зависимость между знаками линейной рекуррентной последовательности, называют *законом рекурсии*, многочлен $F(x) = x^m - \sum_{j=0}^{m-1} f_j \cdot x^j$ — *характеристическим многочленом* ЛРП u , а вектор $u(\overline{0, m-1}) =$

$= (u(0), \dots, u(t-1))$ — начальным вектором ЛРП (или начальным заполнением ЛРС).

Характеристический многочлен ЛРП u , имеющий наименьшую степень, называется ее *минимальным многочленом*, а степень минимального многочлена — *линейной сложностью* ЛРП u .

Линейная сложность ЛРП определяет минимальную длину линейного регистра сдвига, реализующего данную последовательность.

Периодом последовательности u называется наименьшее натуральное число t , для которого существует натуральное число $\lambda > 0$ такое, что для всех $i \geq 0$ справедливо равенство $u(\lambda + i + t) = u(\lambda + i)$.

Из вида закона рекурсии следует, что в случае, когда P — конечное поле из q элементов, максимальное значение периода ЛРП порядка m равно $q^m - 1$. В самом деле, нулевой начальный вектор порождает последовательность, состоящую из одних нулей, а число различных заполнений регистра длины m равно q^m .

Последовательности, имеющие максимально возможный период, получили название линейных рекуррентных последовательностей *максимального периода* или просто *максимальных линейных рекуррентных последовательностей*.

Значения периодов линейных рекуррентных последовательностей определяются свойствами их минимальных многочленов. В частности, для того чтобы линейная рекуррентная последовательность порядка m над полем из q элементов имела максимальный период, необходимо и достаточно, чтобы ее минимальный многочлен был *примитивным многочленом* [Лид88]. Так называется неприводимый многочлен, корень которого имеет в мультипликативной группе поля разложения порядок $q^m - 1$.

Несмотря на то, что имеется достаточно много критериев проверки неприводимости многочленов над конечными полями и методов их построения, доказать то, что заданный неприводимый многочлен примитивен, удается в исключительных случаях.

Для конечного поля из q элементов будем использовать стандартное обозначение $GF(q)$.

Утверждение 1. *Если $F(x)$ — неприводимый многочлен над полем $GF(2)$ степени m , и $2^m - 1$ — простое число, то $F(x)$ — примитивный многочлен.*

Доказательство. Из неприводимости многочлена $F(x)$ следует, что единица поля $GF(2^m)$ (которое является его полем разложения) не является корнем этого многочлена. Так как по условию $(2^m - 1)$ — простое число, все элементы поля $GF(2^m)$, отличные от единицы, имеют в мультипликативной группе этого поля порядок $2^m - 1$. Следовательно, $F(x)$ — примитивный многочлен.

Общий подход к построению примитивных многочленов состоит в построении неприводимых многочленов и непосредственном вычислении их периодов. Проверка максимальности периода неприводимого многочлена может быть осуществлена с помощью следующего утверждения, которое приведем без доказательства.

Утверждение 2. *Неприводимый многочлен $F(x)$ примитивен в том и только в том случае, когда для любого простого числа p , делящего $q^m - 1$, многочлен $x^{\frac{q^m - 1}{p}}$ не сравним с 1 по модулю многочлена $F(x)$.*

Построение примитивных многочленов представляет собой сложную задачу, решение которой даже в частных случаях сопряжено со значительными трудностями вычислительного

характера. На практике используются таблицы неприводимых и примитивных многочленов над конечными полями (см., например, [Неч99]).

Закон рекурсии дает удобный способ вычисления очередного знака ЛРП через предыдущие, но при изучении ее свойств более предпочтительной формой задания является формула общего члена последовательности, представляющая собой аналитическое выражение члена последовательности в виде функции от его номера. Рассмотрим этот способ представления на примере ЛРП над конечными полями с неприводимыми характеристическими многочленами.

Пусть $P = GF(q)$ и Q — поле из q^m элементов, являющееся расширением поля P . Тогда функцией “след” из поля Q в поле P называется отображение $tr_q^{q^m}(a) : Q \rightarrow P$ вида

$$tr_q^{q^m}(a) = a + a^q + a^{q^2} + \dots + a^{q^{m-1}}.$$

В силу свойств конечных полей справедливы равенства

$$(tr_q^{q^m}(x))^q = tr_q^{q^m}(x),$$

$$tr_q^{q^m}(ax + by) = a \cdot tr_q^{q^m}(x) + b \cdot tr_q^{q^m}(y), \quad a, b \in P,$$

означающие, что функция “след” является линейным отображением над полем P .

Лемма 1. Для любого ненулевого $a \in P$ и любого $b \in P$ число решений уравнения $tr_q^{q^m}(a \cdot x) = b$ равно q^{m-1} .

Доказательство. Обозначим через N_b число решений рассматриваемого уравнения. Тогда для любого $b \in P$ выполняется неравенство $N_b \leq q^{m-1}$, так как q^{m-1} — степень мно-

гочлена $tr_q^{q^m}(a \cdot x) - b$, а многочлен над полем не может иметь корней больше, чем его степень. С другой стороны, для любого $x \in P$ значение $tr_q^{q^m}(a \cdot x)$ лежит в поле P , а значит, x является корнем уравнения $tr_q^{q^m}(a \cdot x) = b$ при некотором подходящем значении b . Следовательно,

$$\sum_{b \in Q} N_b = q^m.$$

Полученное равенство (при условии, что $N_b \leq q^{m-1}$ для любого b) может выполняться только в том случае, если для всех значений b имеет место равенство $N_b = q^{m-1}$. Лемма доказана.

Теорема 1. Пусть $F(x) = x^m - \sum_{j=0}^{m-1} f_j \cdot x^j$ — неприводимый многочлен над полем P степени m , θ — корень $F(x)$ в поле Q . Тогда для ЛРП $\{u(i)\}$ с характеристическим многочленом $F(x)$ существует единственная константа $a \in P$, такая, что

$$u(i) = tr_q^{q^m}(a \cdot \theta^i), \quad i \geq 0. \quad (4)$$

Доказательство. То, что последовательность элементов поля, заданная формулой (4), является линейной рекуррентной последовательностью с характеристическим многочленом $F(x)$, следует из равенств

$$\begin{aligned} \sum_{j=0}^{m-1} f_j \cdot u(i+j) &= \sum_{j=0}^{m-1} f_j \cdot \text{tr}_q^{q^m} (a \cdot \theta^{i+j}) = \\ &= \text{tr}_q^{q^m} \left(a \cdot \theta^i \cdot \sum_{j=0}^{m-1} f_j \cdot \theta^j \right) = \\ &= \text{tr}_q^{q^m} (a \cdot \theta^i \cdot \theta^m) = u(i+m). \end{aligned}$$

Заметим, что число всех линейных рекуррентных последовательностей с характеристическим многочленом $F(x)$ равно q^m . Ровно столько же существует и различных констант из поля Q . Из количественных соображений для завершения доказательства теоремы достаточно показать, что при разных константах a из поля Q будут получаться различные последовательности.

Если для двух различных констант a_1 и a_2 выполняется равенство

$$\text{tr}_q^{q^m} (a_1 \cdot \theta^i) = \text{tr}_q^{q^m} (a_2 \cdot \theta^i), \quad i \geq 0,$$

то, пользуясь линейностью функции след, получаем соотношение

$$\text{tr}_q^{q^m} ((a_1 - a_2) \cdot \theta^i) = 0, \quad i \geq 0,$$

которое противоречит лемме 1. Теорема доказана.

Пусть u — ЛРП максимального периода над полем P и $\nu(\alpha_1, \dots, \alpha_k)$ — число решений системы уравнений

$$\begin{cases} u(i+j) = \alpha_j, & j = \overline{1, k}, \\ 0 \leq i < q^m - 1, \end{cases}$$

то есть число появлений мультиграммы $\alpha_1, \dots, \alpha_k$ на периоде последовательности u .

Утверждение 3. Пусть $F(x)$ — примитивный многочлен степени t над полем P и θ — корень $F(x)$ в поле Q . Тогда любая ненулевая мультиграмма $(\alpha_1, \dots, \alpha_k)$ встречается на периоде ЛРП u ровно

$$v(\alpha_1, \dots, \alpha_k) = q^{m-k}$$

раз, а число вхождений нулевой мультиграммы на единицу меньше.

Доказательство. Так как порядок элемента θ равен $q^m - 1$, то множество $\{\theta^i, i = \overline{0, q^m - 2}\}$ содержит все ненулевые элементы поля Q . Поэтому из представления знаков ЛРП u в виде

$$u(i) = \text{tr}_q^{q^m}(a \cdot \theta^i), \quad a \in P,$$

следует, что значение $v(\alpha_1, \dots, \alpha_k)$ равно числу ненулевых решений в поле Q системы уравнений

$$\begin{cases} \text{tr}_q^{q^m}(\theta^j \cdot x) = \alpha_j, \\ j = \overline{1, k}. \end{cases}$$

Пусть $\varepsilon_1, \dots, \varepsilon_m$ — базис поля Q , рассматриваемого как векторное пространство над полем P . Тогда последняя система эквивалентна следующей системе линейных уравнений от m переменных над полем P :

$$\begin{cases} \sum_{s=1}^m x_s \cdot \text{tr}_q^{q^m}(\theta^j \cdot \varepsilon_s) = \alpha_j, \\ j = \overline{1, k}. \end{cases}$$

Покажем, что ранг этой системы линейных уравнений равен k . Предположим, что между строками матрицы системы существует нетривиальное линейное соотношение с коэффициентами c_1, \dots, c_k :

$$\sum_{t=1}^k c_t \cdot \text{tr}_q^{q^m}(\theta^t \cdot \varepsilon_s) = 0, \quad s = \overline{1, m}.$$

Тогда для любого значения $s = \overline{1, m}$ выполняется соотношение

$$\text{tr}_q^{q^m}(\varepsilon_s \cdot \sum_{t=1}^k c_t \cdot \theta^t) = 0.$$

Обозначим $z = \sum_{t=1}^k c_t \cdot \theta^t$. Тогда для любого набора

d_1, \dots, d_m получаем равенство

$$\text{tr}_q^{q^m} \left(z \cdot \sum_{s=1}^m d_s \cdot \varepsilon_s \right) = 0,$$

из которого следует, что $\text{tr}_q^{q^m} (z \cdot x) = 0$ для любого $x \in P$.

Согласно лемме 1, полученное равенство возможно только в случае, когда $z = 0$, что, в силу неприводимости многочлена $F(x)$ и условия $k \leq m$, означает, что $c_1 = \dots = c_k = 0$. Таким образом, ранг системы линейных уравнений равен k , и требуемое утверждение вытекает из известных результатов о числе решений систем линейных уравнений над полем. Утверждение доказано.

Представленные результаты показывают, что линейные рекуррентные последовательности над полем позволяют обеспечить первые два из трех требований к псевдослучайным последовательностям, используемым при построении управляющих блоков поточных шифрсистем. За счет выбора закона рекурсии можно гарантировать достаточную величину периода получаемой псевдослучайной последовательности и хорошие статистические качества. В самом деле, ее период совпадает с числом всех ненулевых векторов длины, равной степени минимального многочлена ЛРП, и каждый из этих векторов встречается на периоде последовательности в точности один раз.

Вместе с тем аналитическое строение ЛРП оказывается очень простым. Для определения начального вектора по некоторому отрезку последовательности достаточно решить несложную систему линейных уравнений. Поэтому при использовании линейных регистров сдвига в криптографических приложениях необходимо предусматривать процедуры, повышающие сложность аналитического строения вырабатываемых ими рекуррентных последовательностей.

§ 9.5. Алгоритм Берлекемпа — Мессе

Основным параметром, характеризующим сложность аналитического строения псевдослучайной последовательности, является ее *линейная сложность*, то есть степень минимального многочлена последовательности. Фактически минимальный многочлен определяет закон рекурсии линейного регистра сдвига минимальной длины, с помощью которого может быть получена данная псевдослучайная последовательность.

Любая периодическая последовательность может рассматриваться как линейная рекуррентная последовательность. Поэтому для любой периодической последовательности существует некоторый линейный регистр сдвига, который ее генерирует. Если элементы, образующие период последовательности, выбираются случайно, то степень минимального многочлена такой последовательности близка к величине ее периода. Поэтому линейная сложность может рассматриваться как характеристика сложности аналитического строения псевдослучайной последовательности. Низкую линейную сложность имеют слабые в криптографическом отношении последовательности. Вместе с тем высокая линейная сложность не гарантирует отсутствия других недостатков. Например, для генерации последовательности, период которой имеет вид $(0,0,\dots,0,1)$, требуется линейный регистр, длина которого совпадает с длиной периода этой последовательности, однако ясно, что считать эту последовательность случайной нельзя.

Эффективным методом нахождения линейного регистра сдвига минимальной длины, генерирующего заданную последовательность, является *алгоритм Берлекемпа — Мессе*. Приведем модификацию этого алгоритма, предложенную А. А. Нечаевым [Kur95].

Пусть P — некоторое поле, e — единица поля P . Обозначим через $u(\overline{0, l-1}) = (u(0), \dots, u(l-1))$ начальный отрезок

произвольной последовательности u элементов поля P . Будем говорить, что многочлен

$$G(x) = x^m - \sum_{j=0}^{m-1} b_j \cdot x^j \in P[x]$$

вырабатывает отрезок $u(\overline{0, l-1})$, если

$$u(i+m) = \sum_{j=0}^{m-1} b_j \cdot u(i+j), \quad i = \overline{0, l-m-1},$$

то есть если данный отрезок последовательности является отрезком некоторой линейной рекуррентной последовательности с характеристическим многочленом $G(x)$.

Алгоритм Берлекемпа — Мессе строит многочлен $G(x)$ наименьшей степени, вырабатывающий отрезок $u(\overline{0, l-1})$. Приведем одну из многочисленных модификаций алгоритма со сложностью, оцениваемой величиной $6m^2(1+o(1))$ операций поля P .

Определим операцию *умножения последовательности на*

многочлен. Для произвольного многочлена $H(x) = \sum_{j=0}^n h_j \cdot x^j$

и последовательности v положим $H(x) \cdot v = w$, где

$$w(i) = \sum_{j=0}^n h_j \cdot v(i+j), \quad i \geq 0.$$

Будем говорить, что многочлен $G(x)$ степени m вырабатывает $l \geq m$ знаков последовательности u , если выполняется равенство

$$G(x) \cdot u = v = (\underbrace{0, \dots, 0}_{l-m}, v(l-m), \dots),$$

то есть если первые $l - m$ знаков последовательности v равны нулю, а следующий за ними отличен от нуля.

Для многочлена $G(x) \in P[x]$ степени m и последовательности u введем параметры $l_u(G)$ и $k_u(G) = l_u(G) - m \in \mathbb{N} \cup \{0, +\infty\}$, где $l_u(G)$ — число знаков последовательности u , вырабатываемых многочленом $G(x)$. Ясно, что $k_u(G)$ — максимальное число первых подряд идущих нулей в последовательности $G(x) \cdot u$ (либо ∞).

Будем индуктивно строить последовательность многочленов $G_0(x), G_1(x), \dots$ неубывающих степеней $0 = m_0 < m_1 \leq m_2 \leq \dots$.

Начальные условия: $G_0(x) = e, m_0 = 0$.

Этап 1. Если

$$G_0(x) \cdot u = u = u_0 = (\underbrace{0, \dots, 0}_{k_0}, \underbrace{u_0(k_0), \dots}_{\neq 0}), \quad k_0 = k_u(G),$$

то полагаем

$$G_1(x) = x^{k_0+1} - u_0(k_0 + 1) \cdot u_0(k_0)^{-1} \cdot x^{k_0} \cdot G_0(x), \quad m_1 = k_0 + 1.$$

Если $G_1(x) \cdot u = 0$, то есть если $k_u(G_1) = \infty$, то $G_1(x)$ — искомый минимальный многочлен ЛРП u . В противном случае строим $G_2(x)$.

Этап $t + 1$. Пусть многочлены $G_0(x), \dots, G_t(x)$ уже построены, и степень $\deg G_j(x)$ многочлена $G_j(x)$ равна m_j , причем $0 = m_0 < m_1 \leq \dots \leq m_t$. Пусть выполняются соотношения

$$G_j(x) \cdot u = u_j = (\underbrace{0, \dots, 0}_{k_j}, \underbrace{u_j(k_j), \dots}_{\neq 0}),$$

$$k_j = k_u(G_j) < \infty,$$

$$l_u(G_j) = k_j + m_j < +\infty.$$

Определим число $s = s(t)$ так, чтобы выполнялись условия $m_t = m_{t-1} = \dots = m_{s+1} > m$, (такое s найдется, так как $m_1 > m_0$). Положим

$$G_{t+1}(x) = \begin{cases} G_t(x) - x^{k_s - k_t} \cdot u_t(k_t) \cdot u_s(k_s)^{-1} \cdot G_s(x), & \text{если } k_t \leq k_s, \\ x^{k_t - k_s} \cdot G_t(x) - u_t(k_t) \cdot u_s(k_s)^{-1} \cdot G_s(x), & \text{если } k_t > k_s. \end{cases}$$

Если $G_{t+1}(x) \cdot u = 0$, то нужный многочлен построен. В противном случае строим $G_{t+2}(x)$.

Теорема 2. Если u — линейная рекуррентная последовательность над полем P с минимальным многочленом $F(x)$ степени m , то $F(x) = G_\tau(x)$ для некоторого подходящего значения $\tau \leq 2m - 1 - k_0$.

Доказательство теоремы носит технический характер и заключается в непосредственной проверке двух условий: каждый следующий многочлен в предложенном алгоритме вырабатывает больше знаков последовательности u , чем предшествующий; любой многочлен, вырабатывающий столько же знаков последовательности u , что и многочлен $G_j(x)$, имеет степень не меньше, чем степень многочлена $G_j(x)$.

§ 9.6. Усложнение линейных рекуррентных последовательностей

Алгоритм Берлекемпа—Мессе позволяет для каждой линейной рекуррентной последовательности построить вырабатывающий ее линейный регистр сдвига минимальной длины. Однако при создании стойких криптографических алгоритмов необходимо гарантировать определенную величину линейной сложности не для одной конкретной ЛРП, а для целого класса псевдослучайных последовательностей, получаемых при различных ключах. При этом, как правило, мощность класса рассматриваемых последовательностей бывает настолько велика, что применить алгоритм Берлекемпа—Мессе к каждой последовательности не представляется возможным. Поэтому важной криптографической задачей является построение целых классов псевдослучайных последовательностей с высокой линейной сложностью.

Описанные выше свойства линейных регистров сдвига показывают, что, несмотря на достаточно большой период и хорошие статистические качества, линейные рекуррентные последовательности имеют очень простое строение. Поэтому в криптографических приложениях используют различные *способы усложнения* аналитического строения линейных рекуррент.

Фильтрующие генераторы

Первый способ заключается в применении к элементальной линейной рекуррентной последовательности некоторой функции f (см. рис. 37) ($F(x)$ — характеристический многочлен определяющий закон рекурсии ЛРС).

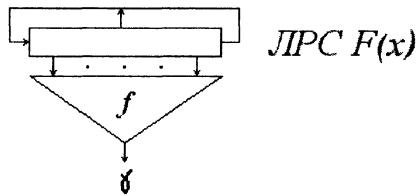


Рис. 37

В литературе подобные узлы усложнения ЛРП получили название *фильтрующих генераторов*. Их результирующей последовательностью является нелинейно “фильтрованное” содержимое регистра сдвига. “Фильтрующая” функция f должна выбираться так, чтобы выходная последовательность имела распределение, близкое к равномерному распределению, и высокую линейную сложность.

Известно [Лид88], что любая функция над конечным полем может быть задана некоторым полиномом. Поэтому при изучении фильтрующих генераторов достаточно рассмотреть только полиномиальные усложнения линейных рекуррент.

Пусть u — линейная рекуррентная последовательность над полем P из q элементов с характеристическим многочленом $F(x)$ степени m , и функция $f(x_1, \dots, x_m)$ имеет вид:

$$f(x_1, \dots, x_m) = \sum_{\substack{0 \leq i_1 < \dots < i_s \leq m \\ 0 \leq k_j \leq q-1 \\ 0 \leq s \leq m}} C_{i_1 \dots i_s}^{k_1 \dots k_s} \cdot x_{i_1}^{k_1} \dots x_{i_s}^{k_s}, \quad C_{i_1 \dots i_s}^{k_1 \dots k_s} \in P.$$

Тогда выходная последовательность фильтрующего генератора определяется следующим равенством:

$$\begin{aligned} \gamma(i) &= f(u(i), \dots, u(i+m)) = \\ &= \sum_{\substack{0 \leq i_1 < \dots < i_s \leq m \\ 0 \leq k_j \leq q-1 \\ 1 \leq s \leq m}} C_{i_1}^{k_1} \dots C_{i_s}^{k_s} u(i+i_1-1)^{k_1} \dots u(i+i_s-1)^{k_s} \end{aligned} \quad (5)$$

Рассмотрим случай, наиболее интересный с практической точки зрения, когда $F(x)$ — примитивный многочлен степени m над полем P . Поскольку, согласно (5), усложненная последовательность является суммой произведений некоторых линейных рекуррентных последовательностей, отличающихся друг от друга сдвигами, продемонстрируем методику изучения строения таких последовательностей на примере доказанной А. А. Нечаевым теоремы о произведении линейной рекуррентной последовательности на ее сдвиг.

Теорема 3. Пусть u — линейная рекуррентная последовательность над полем $P = GF(q)$ характеристики p с примитивным характеристическим многочленом $F(x)$ степени m , элементы которой задаются равенством $u(i) = \text{tr}_q^{q^m}(a \cdot \theta^i)$, где θ — корень $F(x)$ в поле $GF(q^m)$ из q^m элементов, $a \in GF(q^m)$. Тогда элементы последовательности v , задаваемой равенством

$$v(i) = u(i) \cdot u(i+k), \quad i \geq 0,$$

удовлетворяют соотношению

$$v(i) = \text{tr}_q^{q^m} a^2 \theta^k \theta^{2i} + \sum_{s=1}^{\lambda} \text{tr}_q^{q^{m_s}} a^{1+q^s} \theta^{(1+q^s)i} (\theta^k + \theta^{kq^s}), \quad i \geq 0,$$

где $\lambda = \left\lfloor \frac{m}{2} \right\rfloor$, $m_s = \begin{cases} \lambda, & \text{если } m = 2\lambda, s = \lambda; \\ m & \text{в остальных случаях.} \end{cases}$

Доказательство. Непосредственно проверяется справедливость следующих равенств:

$$\begin{aligned} v(i) &= u(i) \cdot u(i+k) = \sum_{t=0}^{m-1} (a\theta^t)^{q^t} \sum_{s=0}^{m-1} (a\theta^k \theta^t)^{q^s} = \\ &= \sum_{s=0}^{m-1} \sum_{t=0}^{m-1} (a\theta^t (a\theta^k \theta^t)^{q^s})^{q^t} = \\ &= \sum_{s=0}^{m-1} \sum_{t=0}^{m-1} ((a\theta^t)^{1+q^s} \cdot \theta^{kq^s})^{q^t} = \\ &= \sum_{s=0}^{m-1} \text{tr}((a\theta^t)^{1+q^s} \cdot \theta^{kq^s}). \end{aligned}$$

Для завершения доказательства теоремы достаточно заметить, что если $m = 2\lambda + 1$, то

$$\text{tr}\left((a\theta^t)^{1+q^{m-s}} \cdot \theta^{kq^{m-s}}\right) = \text{tr}\left((a\theta^t)^{1+q^s} \cdot \theta^k\right), \quad s = \overline{1, \lambda},$$

а если $m = 2\lambda$, то

$$\text{tr}\left((a\theta^t)^{1+q^{m-s}} \cdot \theta^{kq^{m-s}}\right) = \text{tr}\left((a\theta^t)^{1+q^s} \cdot \theta^k\right), \quad s = \overline{1, \lambda-1},$$

при этом справедливо равенство

$$\text{tr}_q^{q^m} (a\theta^i)^{1+q^\lambda} \cdot \theta^{kq^\lambda} = \text{tr}_q^{q^\lambda} (a\theta^i)^{1+q^\lambda} \cdot (\theta^{kq^\lambda} + \theta^k).$$

Теорема доказана.

Следствие. Если в условиях теоремы $0 < k < m$, $m \geq 8$, то

$$M_v(x) = \prod_{s=0}^{\lambda} M_s(x),$$

где $M_s(x)$ — минимальный многочлен элемента θ^{1+q^s} над полем $GF(q)$.

Доказательство. Непосредственно проверяется, что при $k < m$ порядок $\text{ord } \theta^k$ элемента θ^k удовлетворяет неравенствам

$$\text{ord } \theta^k \geq \frac{q^m - 1}{m} > 2(q^{\frac{m}{2}} - 1),$$

и, следовательно, для всех $s = \overline{1, \lambda}$ выполняется условие $\theta^{kq^s} + \theta^k \neq 0$.

Заметим, что в этом случае линейная сложность последовательности v равна $\frac{m(m+1)}{2}$.

В общем случае описание строения минимального многочлена линейной рекуррентной последовательности удобно проводить, исходя из свойств его корней.

Для примитивного многочлена $F(x)$ с корнем θ обозначим через $F^{(r)}(x)$ многочлен вида

$$F^{(r)}(x) = \prod_{W_q(k)=r} (x - \theta^k),$$

где $W_q(k)$ обозначает q -ичный вес числа k , то есть число ненулевых цифр в q -ичной записи числа k . Положим

$$F^{<r>}(x) = \prod_{k=0}^r F^{(r)}(x).$$

Утверждение 4. Пусть u — линейная рекуррентная последовательность над полем $GF(q)$ с примитивным характеристическим многочленом $F(x)$ степени m , θ — корень $F(x)$ в $GF(q^m)$. Тогда ранг последовательности v , полученной из последовательности u усложнением с помощью функции $f(x_1, \dots, x_m)$, $\deg f(\bar{x}) \leq r$, не превосходит $\deg F^{<r>}(x)$, причем многочлен $F^{<r>}(x)$ является характеристическим многочленом последовательности v .

Доказательство. Рассуждая аналогично доказательству теоремы 3 и подставляя выражения членов последовательности v через функцию “след” в произведение $\prod_{k=1}^l u(i + i_k)^{r_k}$, получаем, что каждый знак указанного произведения представляется в виде суммы значений вида

$$c \cdot \theta^{q^{i_1} + q^{i_2} + \dots + q^{i_d}} = c \cdot \theta^{\sum_{t=1}^l q^t \cdot j_t}, \quad (6)$$

где c — некоторая константа из поля $GF(q^m)$,

$$0 \leq i_s < m, \quad s = \overline{1, d}, \quad d = \sum_{k=1}^l r_k \leq r.$$

Следовательно, корни минимального многочлена указанного произведения последовательностей имеют вид:

$$\theta q^{i_1+q i_2+\dots+q^{t-1}i_t} = \theta^{\sum q^t \cdot j_t}, \quad 0 \leq j_t < q,$$

причем $\sum_{t=0}^{m-1} j_t \leq d \leq r$. А поскольку последовательность

$$v(i) = f(u(i), \dots, u(i+m-1)), \quad i \geq 0,$$

представляется в виде суммы последовательностей вида (6), то минимальный многочлен $m_v(x)$ делит $F^{<r>}(x)$. Откуда следует оценка ранга ЛРП ν .

Таким образом, несмотря на то, что фильтрованную последовательность нельзя получить с помощью сдвигового регистра с линейной обратной связью той же длины, что и исходный регистр, ее всегда можно получить с помощью регистра сдвига большей длины.

Заметим, что степень многочлена $F^{(r)}(x)$ равна числу целочисленных наборов (i_0, \dots, i_{m-1}) , таких, что $0 \leq i_j < q$,

$\sum_{j=0}^{m-1} i_j = r$, и представляется в виде

$$\deg F^{(r)}(x) = \sum_{k \geq 0} (-1)^k C_m^k C_{m+r-qk-1}^{m-1}.$$

Данная величина является верхней оценкой линейной сложности класса выходных последовательностей фильтрующего генератора, полученных при различных начальных заполнениях регистра сдвига. Задача получения нижних оценок ранга усложнений линейных рекуррент представляется существенно более сложной. В частности, даже если ограничиться классом последовательностей, полученных с помощью

функций усложнения фиксированной степени нелинейности, то в этом классе будут содержаться рекурренты, линейная сложность которых меняется в диапазоне от m до $\deg F^{<r>}(x)$.

Таким образом, для получения псевдослучайных последовательностей с высоким уровнем линейной сложности функция усложнения фильтрующего генератора должна иметь достаточно высокую степень нелинейности.

Комбинирующие генераторы

Второе направление синтеза псевдослучайных последовательностей с высокой линейной сложностью связано с использованием в одной схеме нескольких линейных регистров сдвига. Генератор псевдослучайных последовательностей, реализующий усложнение нескольких линейных рекуррент с помощью одной общей функции усложнения, получил название *комбинирующего генератора* (см. рис. 38).

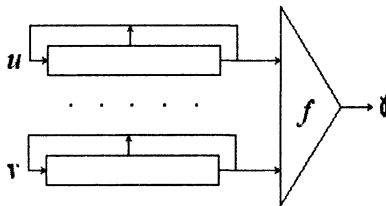


Рис. 38

Рассмотрим один частный случай комбинирующего генератора, когда функция усложнения f имеет степень нелинейности t , и ее представление свободно от квадратов:

$$f(x_1 \dots x_t) = \sum_{k=1}^t \sum_{1 \leq i_1 < \dots < i_k < t} C_{i_1 \dots i_k} \cdot x_{i_1} \cdot \dots \cdot x_{i_k}. \quad (7)$$

Теорема 4. Пусть u_1, \dots, u_t — ЛРП максимального периода с примитивными характеристическими многочленами $F_1(x), \dots, F_t(x)$ над полем $GF(q)$ попарно взаимно простых степеней: $n_s = \deg F_s(x)$, $(n_s, n_j) = 1$ для $s \neq j$, $s, j = \overline{1, t}$. Тогда для функции усложнения вида (7) линейная сложность последовательности v вида

$$v(i) = f(u_1(i), u_2(i), \dots, u_t(i)), \quad i \geq 0$$

равна

$$\sum_{k=1}^t \sum_{1 \leq i_1 < \dots < i_k < t} \bar{C}_{i_1 \dots i_k} \prod_{l=1}^k n_{i_l}, \quad (8)$$

$$\text{где } \bar{C}_{i_1 \dots i_k} = \begin{cases} 0, & \text{если } C_{i_1 \dots i_k} = 0, \\ 1, & \text{если } C_{i_1 \dots i_k} \neq 0. \end{cases}$$

Доказательство. Согласно (7), последовательность v представляется в виде суммы последовательностей вида

$$C_{j_1 \dots j_k} \cdot u_{j_1}(i) \cdot u_{j_2}(i) \cdot \dots \cdot u_{j_k}(i), \quad C_{j_1 \dots j_k} \neq 0, \quad (9)$$

где $1 \leq k \leq t$, $1 \leq j_1 < \dots < j_k \leq t$.

Если в произведение (9) вместо элементов линейных рекуррентных последовательностей подставить их выражения через функцию “след”, то получится представление в виде суммы последовательностей вида $c \cdot (\alpha_{j_1}^{q^{i_1}} \cdot \dots \cdot \alpha_{j_k}^{q^{i_k}})^i$, где α_j — корень многочлена $F_{j_s}(x)$ в поле $GF(q^m)$, $m = n_1 \cdot n_2 \cdot \dots \cdot n_s$, $0 \leq i_s < n_{j_s}$, $s = \overline{1, k}$, c — некоторая константа из поля $GF(q^m)$.

Отсюда следует, что корни минимального многочлена произведения (9) имеют вид $\alpha_{j_1}^{q^{i_1}} \cdot \dots \cdot \alpha_{j_k}^{q^{i_k}}$, $0 \leq i_s < n_{j_s}$, $s = \overline{1, k}$. Непосредственно проверяется, что все такие элементы являются попарно различными и могут быть представлены в виде $(\alpha_{j_1} \cdot \dots \cdot \alpha_{j_k})^{q^r}$ при подходящем значении r . В силу свойств многочленов над конечными полями отсюда следует, что минимальный многочлен произведения (9) является неприводимым многочленом степени $n_{j_1} \cdot n_{j_2} \cdot \dots \cdot n_{j_k}$.

Можно заметить, что при различных наборах индексов $1 \leq j_1 < \dots < j_k \leq t$, $1 \leq k \leq t$, минимальные многочлены для последовательностей вида (9) попарно взаимно просты, поскольку они неприводимы и имеют различные степени. Таким образом, минимальный многочлен последовательности v равен произведению этих неприводимых многочленов, а линейная сложность последовательности v описывается формулой (6). Теорема доказана.

Композиции линейных регистров сдвига

Рассмотрим еще один тип генераторов, представляющий собой композицию линейных регистров сдвига. Так называется схема, в которой выход одного из регистров подается на вход другого регистра (см. рис. 39).

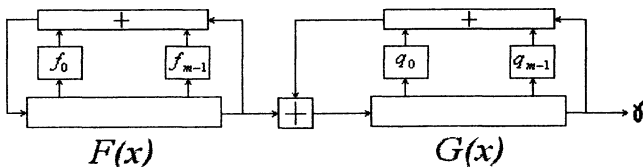


Рис.39

Функционирование такой схемы описывается следующим образом. Пусть v — ЛРП, вырабатываемая первым регистром сдвига, закон рекурсии которого определяется характеристическим многочленом $F(x)$.

Пусть задано начальное состояние второго регистра сдвига, закон рекурсии которого определяется характеристическим многочленом $G(x) = x^m - \sum_{j=0}^{m-1} g_j x^j$. Тогда выходная по-

следовательность композиции регистров сдвига задается соотношением

$$w(i+m) = \sum_{j=0}^{m-1} w(i+j)g_j + v(i), \quad i \geq 0.$$

Непосредственно из определения композиции регистров сдвига вытекает, что минимальный многочлен выходной последовательности делится на минимальный многочлен последовательности v и делит произведение $F(x) \cdot G(x)$.

Схемы с динамическим изменением закона рекурсии

Рассмотренные способы усложнения линейных рекуррент объединяет общая идея повышения линейной сложности исходных последовательностей за счет применения дополнительных функций усложнения. При этом законы функционирования самих регистров остаются неизменными.

Альтернативный способ усложнения ЛРП состоит в изменении закона рекурсии в процессе работы криптографического алгоритма. Привлекательным представляется использование нелинейной логики в цепи обратной связи регистровых преобразований. Однако общая теория подобных схем еще недостаточно разработана, в связи с чем трудно гарантировать необходимые свойства соответствующих последовательностей.

Один из путей построения подобных схем основан на динамическом изменении закона рекурсии линейного регистра сдвига.

Рассмотрим, например, вопрос о строении псевдослучайной последовательности, получаемой с помощью линейного регистра сдвига, закон функционирования которого меняется в зависимости от четности номера вырабатываемого знака.

Рассмотрим два многочлена $A(x) = x^m + \sum a_j x^j$, $B(x) = x^m + \sum b_k x^k$ степени m и последовательность v , удовлетворяющую условиям:

$$v(2t + m) = - \sum_{j=0}^{m-1} a_j \cdot v(2t + j),$$

$$v(2t + 1 + m) = - \sum_{k=0}^{m-1} b_k \cdot v(2t + 1 + k), \quad t \geq 0.$$

Ясно, что в четных тактах закон рекурсии последовательности v определяется характеристическим многочленом $A(x)$, а в нечетных тактах — характеристическим многочленом $B(x)$.

Образуем многочлены $A^{(0)}(x)$, $A^{(1)}(x)$, $B^{(0)}(x)$, $B^{(1)}(x)$ следующим образом:

$$A^{(0)}(x) = \sum_{j \geq 0} a_{2j} x^{2j}, \quad A^{(1)}(x) = \sum_{j \geq 0} a_{2j+1} x^{2j+1},$$

$$B^{(0)}(x) = \sum_{j \geq 0} b_{2j} x^{2j}, \quad B^{(1)}(x) = \sum_{j \geq 0} b_{2j+1} x^{2j+1}.$$

Несложно проверить, что получаемая в результате последовательность v является линейной рекуррентной последовательностью с характеристическим многочленом

$$H(x) = A^{(1)}(x)B^{(1)}(x) - A^{(0)}(x)B^{(0)}(x).$$

К классу генераторов с динамическим изменением закона рекурсии относятся также линейные регистры сдвига с *неравномерным движением информации*. Так называются регистры, для которых число тактов работы до получения $(i+1)$ -го знака выходной последовательности зависит либо от значения числа i , либо от состояния регистра в такте i . Получаемая в результате последовательность будет некоторой выборкой с переменным шагом из исходной ЛРП, вырабатываемой линейным регистром сдвига.

В одном из возможных вариантов значение шага выборки меняется в соответствии с некоторой последовательностью периода t . В этом случае удастся описать минимальные многочлены получаемых псевдослучайных последовательностей [Gol88].

Теорема 5. Пусть d_0, \dots, d_{t-1} -- набор из $t \geq 2$ чисел множества $\overline{0, q^n - 2}$ и $D = \sum_{j=0}^{t-1} d_j$. Пусть при $i = k \cdot t + r$

элементы последовательности v задаются формулой

$$v(i) = \text{tr}_q^{q^n} \left(c\theta^{kD + \sum_{j=0}^{r-1} d_j} \right),$$

где c — ненулевой, а θ — примитивный элемент поля $GF(q^n)$. Пусть, кроме того, простые делители числа t де-

делят $\frac{q^n - 1}{(q^n - 1, D)}$, но не делят $(q^n - 1, D)$. Пусть, наконец, $q^n \equiv 1 \pmod{4}$, если t кратно четырем. Тогда минимальный многочлен $M_\nu(x)$ последовательности ν имеет вид $M_\nu(x) = F(x^t)$, где $F(x)$ — минимальный многочлен элемента θ^D .

Доказательство. Пусть $F(x) = \sum_{s=0}^m f_s x^s$. Тогда

$F(x^t) = \sum_{s=0}^m f_s x^{st}$. Умножая последний многочлен на последовательность ν и полагая $i = kt + r$, получаем:

$$\begin{aligned} [F(x^t)\nu](i) &= [F(x^t)\nu](kt+r) = \\ &= \sum_{s=0}^m f_s \nu(i+st) = \sum_{s=0}^m f_s \nu(kt+r+st) = \\ &= \sum_{s=0}^m f_s \nu((kt+r)+st) = \\ &= \sum_{s=0}^m f_s \cdot \text{tr}_q^{q^n} \left(c\theta^{kD + \sum_{j=0}^{r-1} d_j + sD} \right) = \\ &= \text{tr}_q^{q^n} \left(c\theta^{kD + \sum_{j=0}^{r-1} d_j} \left(\sum_{s=0}^m f_s \theta^{sD} \right) \right) = 0. \end{aligned}$$

Отсюда следует, что многочлен $F(x^t)$ является характеристическим для последовательности ν . Кроме того, в условиях теоремы, он является неприводимым. Это следует из свойств неприводимых многочленов над конечными полями [Лид88]. Таким образом, $F(x^t)$ является минимальным многочленом последовательности ν , что и требовалось доказать.

Схемы с элементами памяти

Дополнительные возможности для усложнения ЛРП открывает использование в криптографических алгоритмах элементов памяти.

Один из наиболее широко известных классов датчиков псевдослучайных чисел, построенных с использованием памяти, составляют *генераторы Макларена—Марсальи*. Принцип работы таких генераторов в самом общем виде можно сформулировать следующим образом.

Пусть имеются три последовательности и массив памяти. Будем записывать элементы первой последовательности в память по адресам, которые определяются элементами второй последовательности. Элементы выходной последовательности получают при считывании значений, хранящихся в массиве памяти, в соответствии с элементами третьей последовательности. Таким образом, первая последовательность определяет, какие знаки заносятся в память, вторая последовательность управляет процессом записи этих элементов в память, а третья — процессом считывания из памяти элементов выходной последовательности.

Рассмотрим частный случай, когда процессами записи и считывания управляет одна и та же последовательность. Схема работы такого генератора изображена на рис. 40.

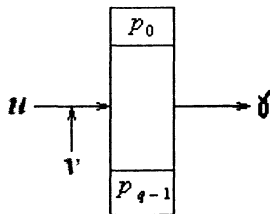


Рис. 40

Пусть u и v — последовательности над полем P , а выходная последовательность γ вырабатывается с использованием q ячеек памяти R_0, \dots, R_{q-1} . Если $R_j(i)$ — заполнение j -й ячейки памяти перед началом i -го такта работы схемы, то преобразование информации в i -м такте описывается формулами

$$\gamma(i) = R_{v(i)}(i),$$

$$R_j(i+1) = \begin{cases} R_j(i), & \text{если } j \neq v(i), \\ u(i), & \text{если } j = v(i). \end{cases}$$

Таким образом, последовательность v определяет адреса, по которым в память записываются элементы последовательности u .

Рассмотрим вопрос о периодах выходных последовательностей генератора Макларена—Марсальи указанного типа.

Пусть последовательность u имеет период τ , а адресная последовательность v принимает значения $0, 1, \dots, q-1$ и имеет период t . Начальное заполнение памяти $R_0(0), \dots, R_{q-1}(0)$ выбирается произвольно.

Пусть i_0 — такое наименьшее натуральное число, что для любого $k \in \{0, 1, \dots, q-1\}$ существует $i \in \overline{0, i_0}$ такое, что

$v(i) = k$. Другими словами, через i_0 тактов работы схемы память будет заполнена только элементами последовательности u .

В дальнейших рассуждениях будем рассматривать работу схемы, начиная с такта $i_0 + 1$.

Теорема 6. Пусть τ — период последовательности u , t — период последовательности v , причем $(\tau, t) = 1$, $t < \tau$. Тогда для значения периода T выходной последовательности γ выполняется равенство $T = \tau s$, где $s = \frac{t}{d}$ и $d \leq e^{\frac{q}{e}}$.

Доказательство. Для $i > i_0$ определим параметры l_i равенствами

$$l_i = \min\{l \in N : v(i-l) = v(i)\}.$$

Тогда $\gamma(i) = u(i - l_i)$. Несложно заметить, что для любых значений i и j выполняются соотношения

$$\gamma(i + jt) = R_{v(i+jt)}(i + jt) = R_{v(i)}(i + jt) = u(i + jt - l_i).$$

Таким образом, каждая регулярная выборка с шагом t из последовательности γ совпадает с некоторой регулярной выборкой с шагом t из последовательности u . Отсюда и из условия $(\tau, t) = 1$ следует, что τ делит T .

С другой стороны, для любого i справедливы равенства

$$\begin{aligned} \gamma(i + \tau t) &= R_{v(i+\tau t)}(i + \tau t) = R_{v(i)}(i + \tau t) = u(i + \tau t - l_i) = \\ &= u(i - l_i) = R_{v(i)}(i) = \gamma(i). \end{aligned}$$

Следовательно, $T = \tau s$, где $s | t$.

Пользуясь условием периодичности и определением величин l_i , для любого $i > i_0$ получаем, что

$$\gamma(i + j\tau s) = \gamma(i) = u(i - l_i),$$

$$\gamma(i + j\tau s) = u(i + j\tau s - l_{i+j\tau s}) = u(i - l_{i+j\tau s}).$$

Так как $(\tau, t) = 1$, то из условия $u(i - l_i) = u(i - l_{i+j\tau s})$, $i \geq 0$, следует, что для любых значений i_1, i_2 , таких, что $i_1 \in \overline{0, t-1}$, $i_2 \in \overline{0, \tau-1}$, выполняется равенство

$$u(i_2 - l_{i_1}) = u(i_2 - l_{i_1 + js}).$$

Отсюда, с учетом неравенства $t < \tau$, получаем, что

$$l_{i_1} = l_{i_1 + js}$$

для $i_1 \in \overline{0, t-1}$.

Это означает, что все регулярные выборки с шагом s из последовательности ν , начинающиеся с одного и того же элемента $c \in \overline{0, q-1}$, совпадают. Тогда период любой выборки из последовательности ν равен числу различных элементов из множества $\{0, 1, \dots, q-1\}$, встречающихся в ней.

Будем говорить о выборке с шагом s из последовательности ν , начинающейся с элемента $\nu(i)$ как (i, s) -выборке.

Обозначим подмножество элементов множества $\{0, 1, \dots, q-1\}$, встречающихся в фиксированной (i, s) -выборке, через $Sp(i, s)$, а его мощность через k_i . Множество $Sp(i, s)$ назовем *носителем* (i, s) -выборки. Заметим, что либо $Sp(i, s) = Sp(j, s)$, либо $Sp(i, s) \cap Sp(j, s) = \emptyset$, $i, j \in \overline{1, s}$.

Будем считать, что первые r выборок ($r \leq s$) имеют различные носители, а любая из оставшихся $s - r$ выборок совпадает с одной из первых r выборок. Тогда

$$q = k_1 + k_2 + \dots + k_r, k_i \geq 1, i \in \overline{1, r}.$$

Так как период t последовательности ν равен $s \cdot \text{НОК}(k_1, k_2, \dots, k_r)$, то для получения нижних оценок периода выходной последовательности γ остается найти верхнюю оценку величины $d = \text{НОК}(k_1, k_2, \dots, k_r)$.

Очевидно, что $\text{НОК}(k_1, k_2, \dots, k_r) \leq k_1 \cdot k_2 \cdot \dots \cdot k_r$, откуда

$$\sqrt[r]{d} = \sqrt[r]{k_1 \cdot k_2 \cdot \dots \cdot k_r} \leq \frac{k_1 + k_2 + \dots + k_r}{r} = \frac{q}{r}.$$

Таким образом, $d \leq \left(\frac{q}{r}\right)^r$. Максимум функции

$f(x) = \left(\frac{q}{x}\right)^x$ действительного переменного на отрезке $[1, s]$

достигается при $x = \frac{q}{e}$ и максимальное значение d равно $e^{q/e}$.

Теорема доказана.

§ 9.7. Методы анализа поточных шифров

Рассмотрим некоторые методы анализа поточных шифр-систем на примере шифров гаммирования.

В первую очередь необходимо исследовать вероятностные характеристики гаммы. Как мы уже знаем (см. гл. 6), имеются подходы к получению оценок вероятностей элементов неравновероятной гаммы по шифртексту, которые можно использовать при бесключевом чтении.

Второй подход связан с попытками *линеаризации* уравнений гаммообразования, то есть сведения задачи нахождения ключей криптографических алгоритмов к решению некоторой системы линейных уравнений.

При таком подходе определяющую роль играет линейная сложность исследуемых последовательностей. Значение линейной сложности определяет размеры системы линейных уравнений, которую надо решить для определения ключа по известной шифрующей гамме. Поэтому линейная сложность определяет эффективность криптоатаки на основе известного открытого текста для шифров гаммирования в классе методов линеаризации. Это обуславливает актуальность разработки методов построения псевдослучайных последовательностей с высокой линейной сложностью.

К вопросу о статистических зависимостях в шифрующей гамме примыкают методы анализа, основанные на наличии у функции усложнения хороших приближений в классе линейных функций. Примером отображений, не имеющих линейных статистических аналогов хорошего качества, является класс бент-функций.

В случае наличия у функции усложнения линейного приближения криптоаналитик может заменить исследуемую схему схемой с линейной функцией усложнения. Если усложнению подвергалась линейная рекуррентная последовательность, то при такой замене результирующая гамма является суммой линейной рекуррентности и некоторой случайной последовательности с “завышенной” вероятностью появления нуля. Тем самым задача сводится фактически к возможности определения ключа по “искаженному” выходу линейного регистра сдвига. Если число искажений невелико, то их появление не оказывает существенного влияния на сложность определения ключа.

Отметим, что функция усложнения может обеспечивать высокий уровень линейной сложности и хорошие статистические качества результирующей гаммы (например, равновероятность появления ее элементов), но при этом она может иметь прибли-

жение в классе линейных функций с большой вероятностью совпадения значений, что сводит на нет перечисленные положительные качества.

При оценке криптографических качеств поточных шифров, помимо алгебраических и статистических свойств шифрующей гаммы, необходимо учитывать также наличие между знаками гаммы зависимостей комбинаторного характера. Например, при использовании в качестве гаммы линейной рекуррентной последовательности с малым числом ненулевых коэффициентов в законе рекурсии может иметь место ситуация, когда значительное число знаков гаммы зависит лишь от небольшого числа знаков ключа. Если такая ситуация имеет место, то криптоаналитик получает возможность проверки гипотез о значениях части ключа, основываясь на статистических свойствах открытых сообщений, что является несомненной слабостью соответствующего алгоритма шифрования.

Таким образом, при создании криптографически стойких поточных шифрсистем необходимо учитывать возможности применения криптоаналитиком всей совокупности статистических, аналитических и комбинаторных свойств используемых преобразований. При этом дополнительные трудности создают постоянно возрастающие возможности вычислительной техники, позволяющие провести экспериментальные исследования тех характеристик поточных шифров, которые не удастся изучить теоретически. В связи с этим необходимо подчеркнуть, что приведенные в данной книге требования к поточным шифрам являются необходимыми, но далеко не достаточными для создания стойких шифров. Вывод о криптографической стойкости конкретного шифра может быть сделан только на основе его комплексных исследований, проведенных с привлечением квалифицированных специалистов-криптографов.

Контрольные вопросы

1. В чем заключаются достоинства и недостатки систем поточного шифрования по сравнению с блочными шифрами?
2. Почему возникает проблема синхронизации поточных шифров?
3. Что с точки зрения криптографического алгоритма определяет управляющий блок?
4. Какой необходимый минимум функциональных возможностей должен быть заложен в шифрующем блоке?
5. За счет чего можно обеспечить стойкость алгоритма шифрования при повторном использовании ключей?
6. Какие причины обусловили широкое использование линейных регистров сдвига в качестве управляющих блоков поточных шифрсистем?
7. Какой период имеет ненулевая линейная рекуррентная последовательность над полем $GF(2)$ с характеристическим многочленом x^5+x^2+1 ? Какова частота появления триграммы $(0,1,1)$ на периоде этой последовательности?
8. Какова длина отрезка, необходимого для восстановления минимального многочлена заданной линейной рекуррентной последовательности с помощью алгоритма Берлекемпа—Месси?
9. Для каких целей применяются усложнения линейных рекуррентных последовательностей?
10. Какую минимальную степень нелинейности должна иметь булева функция от m переменных в фильтрующем генераторе, чтобы после ее применения к знакам линейной рекуррентной последовательности над полем $GF(2)$ с примитивным характеристическим многочленом $F(x)$ степени m линейная сложность выходной последовательности была бы больше m^3 ?
11. Какие существуют типы генераторов Макларена—Марсальи?

Глава 10

Шифрование в аналоговой телефонии

Существуют два класса систем связи: *цифровые* и *аналоговые*. Все наши предыдущие исследования были связаны с цифровыми сигналами, то есть сигналами, имеющими конечное число дискретных уровней. *Аналоговые* сигналы являются непрерывными. Типичным примером такого сигнала является речевой сигнал, передаваемый по обычному телефону. Информацию, передаваемую аналоговыми сигналами, также необходимо защищать, в том числе и криптографическими методами.

Имеются два различных способа шифрования речевого сигнала. Первый состоит в перемешивании (*скремблировании*) сигнала некоторым образом. Это делается путем изменения соотношений между временем, амплитудой и частотой, не выводящих за пределы используемого диапазона. Второй способ состоит в преобразовании сигнала в цифровую форму, к которой применимы обычные методы дискретного шифрования. Зашифрованное сообщение далее передается по каналу с помощью модема. После расшифрования полученной криптограммы вновь восстанавливается аналоговая форма сигнала.

Прежде чем перейти к деталям, необходимо остановиться на некоторых особенностях речевых сигналов.

§ 10.1. Особенности речевых сигналов

Непрерывные сигналы характеризуются своим спектром. *Спектр сигнала* — это эквивалентный ему набор синусоидальных составляющих (называемых также *гармониками* или *частотными составляющими*). Спектр сигнала получается разложением функции, выражающей зависимость формы сиг-

нала от времени, в ряд Фурье. Спектр периодического сигнала — линейчатый (дискретный), он состоит из гармоник с кратными частотами. Спектр непериодического сигнала — непрерывный. Типичный спектр речевого сигнала показан на рис. 41:

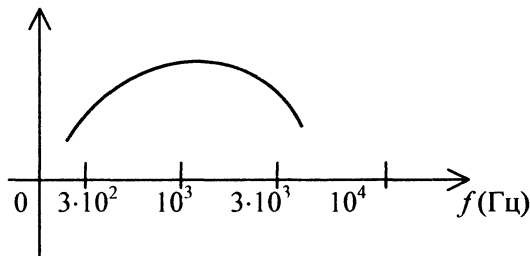


Рис. 41

Частотные составляющие в диапазонах 3÷4 кГц и менее 300 Гц быстро убывают. Таким образом, очень высокие частотные компоненты имеют существенно меньший вклад в сигнал, чем частоты в диапазоне 500÷3000 Гц.

Если ограничиться частотами, не превышающими 3 кГц, и использовать высокочувствительный анализатор, то спектр, производимый некоторыми звуками, имеет вид зубчатой кривой приблизительно следующего вида (см. рис. 42)

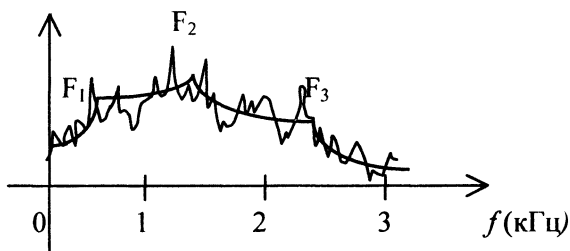


Рис. 42

Мы видим несколько *тиков* графика, называемых *формантами*. Изменение этих частотных компонент во времени можно изобразить на трехмерном графике (при добавлении третьей координаты — времени).

Речевой сигнал является переносчиком смысловой информации. Эта информация при прослушивании речевого сигнала может быть записана в виде текста сообщения. Слуховое восприятие речевого сигнала более богато и несет как основную текстовую информацию, так и дополнительную в виде ударений и интонаций. Элементарными единицами слуховой информации являются элементарные звуки — *фонемы*, а смысловыми единицами — звучащие слоги, слова и фразы. Для каждого языка имеется свой набор фонем. Например, в русском и английском языках имеется около 40 фонем.

Множество фонем разбивается на три класса. Гласные образуют одно семейство, согласные и некоторые другие фонетические звуки (для английского языка — это, например, звуки *ch*, *sh*) образуют два класса, называемые *взрывными* звуками и *фрикативными* звуками. Гласные производятся движением голосовых связок под воздействием потоков воздуха. Проходя через гортань, они превращаются в серию вибраций. Затем воздушный поток проходит через некоторое число резонаторов, главными из которых являются нос, рот и горло, превращаясь в воспринимаемые человеческим ухом фонемы. Возникающие звуки зависят от формы и размеров этих резонаторов, но в значительной степени они характеризуются низкочастотными составляющими.

Гласные звуки производятся в течение длительного времени. Как правило, требуется около 100 мс для достижения его пиковой амплитуды. Взрывные звуки производятся путем “перекрытия” воздушного потока с последующим его выпуском с взрывным эффектом. Блокирование воздушного потока может осуществляться различными способами — языком, нёбом или губами. Например, звук “п” произносится при блокировании воздушного потока губами. Взрывные звуки

характеризуются их высокочастотными составляющими. До 90% их пиков амплитуды имеют длительность, не превышающую 5 мсек. Фрикативные звуки производятся частичным перекрытием воздушного потока, что дает звук, похожий на “белый шум”. Этот звук затем фильтруется резонаторами голосового тракта. Фрикативный звук обычно богат пиками амплитуды длительностью 20÷50 мс и сконцентрирован по частоте от 1 до 3 кГц. Пример фрикатива — звук “ссс...”.

Другой важной характеристикой человеческой речи является *частота основного тона*. Это — частота вибраций голосовых связок. Среднее значение этой частоты колеблется у разных людей, и у каждого говорящего имеется отклонение в пределах октавы выше или ниже этой центральной частоты. Обычно у мужчины частота основного тона колеблется около 1300 Гц, у женщины она выше.

Речевые сигналы не только передают информацию, но и дают сведения о голосовых характеристиках говорящего, что позволяет идентифицировать его по голосу. Можно использовать высоту, форманты, временную диаграмму и другие характеристики речевого сигнала, чтобы попытаться сформировать сигнал, схожий с оригиналом. Это воспроизведение может быть в некоторой степени неестественным и некоторые индивидуальные характеристики говорящего будут утеряны. Такие принципы репродукции лежат в основе *вокодера*, о котором будет сказано далее.

§ 10.2. Скремблирование

Рассмотрим сначала первый способ шифрования речевых сигналов в их аналоговой форме. При оценке стойкости шифрования речевых сигналов в аналоговой форме необходимо, в первую очередь, учитывать возможности человеческого восприятия при прослушивании результирующего сигнала и попытке восстановить какую-либо информацию. Это восприятие очень субъективно: одни люди воспринимают на слух

значительно лучше других. Например, хорошо известно, что родители понимают “речь” своих детей задолго до того, как ее начинают понимать другие люди. В связи с этим говорят о так называемой *остаточной разборчивости сигнала*.

Оценивая надежность шифрования целесообразно несколько раз подряд прослушать скремблированные телефонные сообщения. Дело в том, что человеческий мозг способен адаптироваться к “добыванию” информации и быстро анализировать услышанное. Часто бывает так, что после второго или третьего прослушивания человек начинает распознавать отдельные слова или слоги. Возможно, что смесь различных фрагментов приведет к пониманию смысла сообщения. Имеются даже эксперты для восприятия скремблированных сообщений.

Простейшей формой рассматриваемых преобразований являются преобразования сигнала в частотной области: *инверсии*, *циклические инверсии* и *частотные перестановки*.

§ 10.3. Частотные преобразования сигнала

Простейшим является преобразование инверсии спектра. Оно осуществляется следующим образом. Рассмотрим, например, сигнал, расположенный в диапазоне 300÷3000 Гц (см. рис. 43).

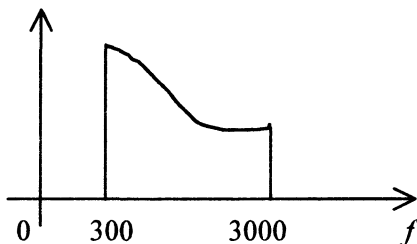


Рис. 43

Попытаемся преобразовать сигнал таким образом, чтобы поменялись местами высокие и низкие частоты. Для этого рассмотрим отдельные гармоники нашего сигнала. Если $V_m \cdot \cos(\omega_m t)$ — одна из гармоник, подаваемая вместе с сигналом $V_c \cdot \cos(\omega_c t)$ на вход устройства, называемого *смесителем*, то его выходом будет сигнал $V_c \cdot V_m \cdot \cos(\omega_c t) \cdot \cos(\omega_m t)$.

Согласно известному равенству

$$\cos A \cdot \cos B = \frac{1}{2} \cos(A + B) + \frac{1}{2} \cos(A - B),$$

выход смесителя выражается в виде

$$\frac{1}{2} V_c V_m \cos(\omega_c + \omega_m)t + \frac{1}{2} V_c V_m \cos(\omega_c - \omega_m)t.$$

Величины V_c и ω_m можно выбирать. Положив $V_c = 1$ и $\omega_c > \omega_m$, мы получим следующий амплитудно-угловой спектр выхода смесителя (см. рис. 44).

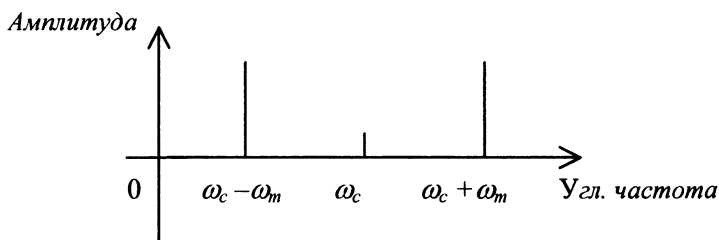


Рис. 44

При рассмотрении каждой гармоники сигнала и соответствующего выхода смесителя получим следующий график (см. рис. 45).

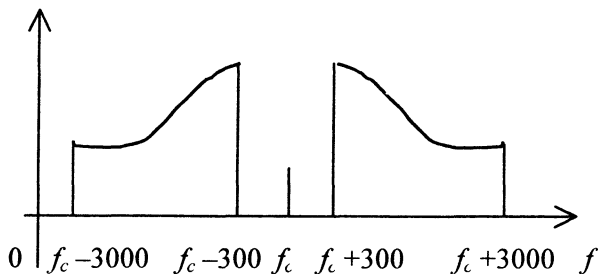


Рис. 45

Между несущей частотой f_c находятся два диапазона, называемые *верхним* и *нижним диапазонами* соответственно. Верхний диапазон аналогичен исходному сигналу, лишь перемещенному вверх (каждая частотная компонента увеличивается на f_c). Нижний диапазон является зеркальным отражением исходного сигнала. Теперь, выбирая подходящую несущую частоту и используя смеситель для перемещения верхнего диапазона, мы можем получить инвертированный речевой сигнал (см. рис. 46).

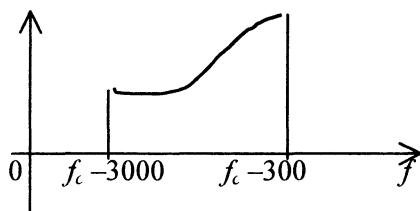


Рис. 46

Выбором несущей частоты для различных сигналов каждый из них может быть перенесен в другой частотный диапазон. Это дает возможность передавать несколько телефонных сигналов по одному каналу.

Преобразование инверсии не зависит от секретного ключа. Это — кодирование, являющееся нестойким против атак противника, обладающего аналогичным оборудованием. Развитие идеи инверсного кода, позволяющее ввести секретный ключ, состоит в использовании так называемой *циклической инверсии*. Суть преобразования циклической инверсии заключается в следующем.

Как мы уже заметили, если инвертированный сигнал находится в том же диапазоне, что и исходный сигнал (300 ÷ 3000 Гц), то несущая частота равна 3300 Гц. Для другой несущей частоты, скажем 4000 Гц, получим инвертированный сигнал со спектром, изображенным на рис. 47.

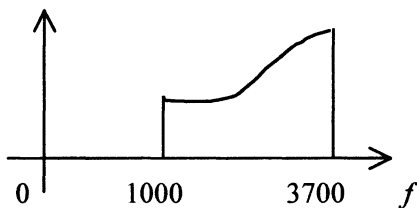


Рис. 47

Этот сигнал не попадает в исходную полосу. Можно договориться переносить часть спектра, превышающую 3000 Гц, в нижнюю часть исходного спектра (см. рис. 48).

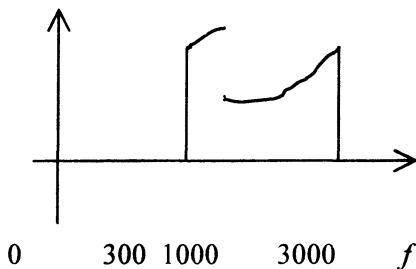


Рис. 48

В таком переносе части спектра и заключается идея циклической инверсии. Типичный инвертор имеет от 4 до 16 различных несущих частот. Это дает такое же число возможных циклических сдвигов. С помощью ключа можно выбирать несущую частоту так, как это делается для шифра простой замены. Можно использовать также генератор псевдослучайных чисел, который выбирает изменяемую несущую частоту. Обычно для этого используют интервал в 10 или 20 мс. Реализующее такой метод устройство называют *циклическим инверсным переключателем диапазона*.

Подобные системы имеют две серьезные слабости. Во-первых, в каждый момент времени имеется лишь небольшое число возможных несущих частот, в силу чего исходный сигнал может быть восстановлен их перебором с помощью сравнительно простого оборудования. Во-вторых, что более важно, остаточная разборчивость выходного сигнала для такого метода неприемлемо высока, что проявляется при непосредственном прослушивании.

Третий способ изменения сигнала в частотной области состоит в делении диапазона. Спектр сигнала делится на некоторое число равных поддиапазонов, которые могут переставляться местами друг с другом. К этому можно добавить также возможность инвертирования для некоторых поддиапазонов. Эту идею проиллюстрируем следующим примером.

Пример

Рассмотрим сигнал, изображенный на рис. 49, а. В нашем примере частотный диапазон разбит на пять равных частей, которые переставляются в соответствии с указанной нумерацией, при этом первая и пятая части инвертированы (см. рис. 49,б).

Всего в нашем примере имеется $5!$ возможных перестановок и 2^5 возможностей для инвертирования. Итого — $5! \cdot 2^5 = 3840$ вариантов преобразований сигнала. Это, конечно, не очень много. Хуже обстоит дело с остаточной разборчивостью. Если использовать лишь перестановки полос, то для большинства из них остаточная разборчивость достигает 10%, что, конечно, не дает гарантии стойкости.

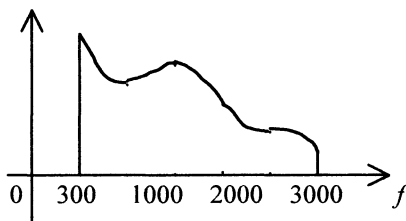


Рис. 49, а

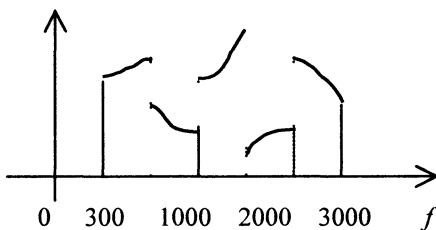


Рис 49, б

Некоторые причины этого легко понять. Так будет, например, если некоторые поддиапазоны остаются неизменными. Кроме того, известно, что обычно более 40% энергии сиг-

нала лежит в первых двух поддиапазонах, соответствующих первой форманте. Как только криптоаналитик найдет правильные позиции первых двух поддиапазонов и переместит их на нужные места, он частично восстановит сигнал и получит неплохой шанс понять фрагмент сообщения.

Можно попытаться улучшить систему защиты за счет использования некоторого числа различных перестановок, которые меняются через короткие промежутки времени с помощью генератора псевдослучайных чисел. Часто для реальных систем лучшие (с точки зрения низкой остаточной разборчивости) перестановки хранятся в ROM (памяти только для чтения), имеющейся внутри устройства.

Несмотря на то, что генератор может вырабатывать последовательность очень большого периода, и размер ключа может быть выбран достаточно большим, даже в этом случае остаточная разборчивость большой доли преобразований так велика, что система не может в полной мере обеспечить необходимую надежность защиты.

То же можно сказать вообще о любом скремблере, использующем лишь действия с частотной областью. Их применение ограничивается лишь ситуациями, когда целью является препятствие пониманию разговора для случайного слушателя или даже противника, не обладающего подходящим оборудованием. Как будет видно из дальнейшего, более совершенные системы или увеличивают ширину спектра сигнала, или вводят временные задержки в передачу. Подобные изменения влекут свои собственные проблемы и поэтому частотные скремблеры могут быть использованы только тогда, когда не требуется гарантированная стойкость.

К сказанному следует добавить замечание о числе поддиапазонов, используемых частотным скремблером. В предыдущем примере их было пять. Ясно, что с ростом этого числа значительно увеличилось бы число возможных перестановок, что привело бы к возрастанию стойкости системы. Однако введение слишком большого числа поддиапазонов связано с

большими практическими трудностями. Дело в том, что на приеме необходимо восстановить исходный сигнал. Фильтры же и другие компоненты схемы вносят шумы и не являются в точности линейными системами. Любые преобразования сигнала, производимые при передаче, являются несовершенными и приводят к ухудшению его качества на выходе. Скремблеры особенно чувствительны к подобным искажениям. Поэтому увеличение числа поддиапазонов делает систему или вообще непригодной, или неэкономичной.

§ 10.4. Временные преобразования сигнала

Рассмотрим теперь скремблеры, воздействующие на временные компоненты сигнала. В их основе лежат следующие принципы.

Сначала аналоговый сигнал делится на равные промежутки времени, называемые *кадрами*. Каждый кадр, в свою очередь, также делится на еще меньшие части, называемые *сегментами*. Входной сигнал преобразуется путем перестановки сегментов внутри каждого кадра. Для этого речевой сигнал на передающем конце записывается на магнитофонную ленту, которая “разрезается” на равные части, пронумерованные по порядку. Затем они перемешиваются и “склеиваются” в некотором другом порядке. Воспроизведенный со склеенной ленты сигнал передается по каналу связи и на приемном конце снова записывается на магнитофонную ленту, которая, в свою очередь, разрезается на те же части, как и при передаче. Части “склеиваются” в порядке возрастания номеров и сигнал восстанавливается.

Проиллюстрируем этот процесс следующим примером.

Пример

На рис. 50 кадр разделен на 8 сегментов. Затем сегменты переставляются в соответствии с перестановкой

(1, 8, 3, 5, 7, 4, 2, 6).

При настройке системы необходимо выбрать длины кадров и сегментов. Так как внутри сегмента сигнал не разрушается, то сегменты желательно выбирать настолько короткими, чтобы в них не содержались целые фрагменты сообщения, например отдельные слова. С другой стороны, длина сегмента серьезно влияет на качество звучания передаваемого сигнала, что объясняется чисто техническими причинами. Чем меньше сегмент, тем ниже качество звучания. Поэтому в выборе длины сегмента необходим разумный компромисс.

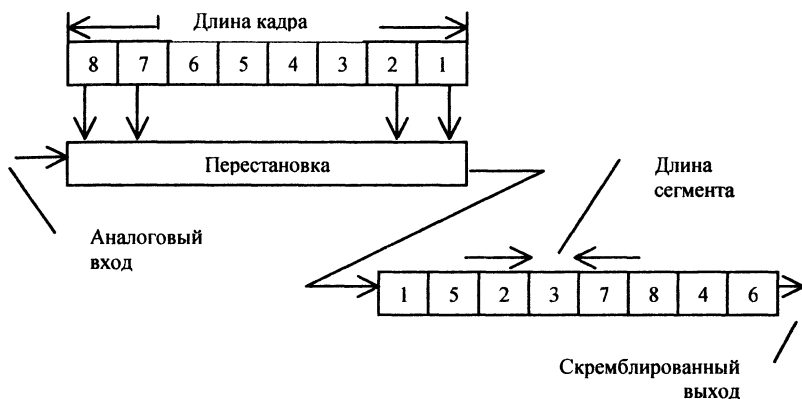


Рис. 50

При выборе длины кадра необходимо учитывать фактор временной задержки между входным аналоговым сигналом, поступающим в аппаратуру, и восстановленным аналоговым сигналом на приеме. Для того чтобы разобраться в этом, вернемся к последнему примеру (см. рис. 50).

Пусть сегмент составляет в нашем примере интервал времени в T с. Тогда ввод восьми речевых сегментов на вход скремблера занимает $8T$ с. Перестановка сегментов могла быть такой, что восьмой сегмент оказался бы первым (то есть подлежащим передаче по каналу связи в первую очередь). В таком случае передача не может начаться прежде, чем в

скремблер не будет введен весь кадр. Для этого потребуется $8T$ с. С начала передачи кадра до ее окончания требуется еще $8T$ с. Поэтому временная задержка неизбежна. Получатель не может начать расшифровку до получения всего кадра. Таким образом, даже если не учитывать время передачи, задержка составляет $16T$ с на каждый сегмент речи. В общем случае для системы, оперирующей с m сегментами на кадр, время задержки может составлять $2mT$ с. С точки зрения пользователя это нежелательно, и подобная задержка должна быть минимизирована. Однако для повышения надежности засекречивания желательны достаточно длинные кадры. Для того чтобы убедиться в этом, заметим следующее.

При обсуждении свойств фонем мы могли заметить, что свойства речевого звука сохраняются в течение достаточно большого интервала времени (структура формант медленно изменяется). Если кадр настолько мал, что он состоит из единственного тона, то независимо от того, как мы его будем скремблировать, результатом будет являться единственный непрерывный тон (с определенной потерей качества звука в результате нашего вмешательства). Нам не удастся добиться достаточного рассеивания сегментов в связи с их небольшим числом. Это может привести к тому, что значительные части слов окажутся неизменными, что позволит слушателю распознать часть сообщения.

Нет также очевидного способа выбора длины сегментов. На практике необходимо экспериментально проверять любой выбор длины сегмента. Обычно неплохим тестом для этого служит попытка воспроизвести на слух результат скремблирования произнесенных в произвольном порядке чисел от 1 до 10. Ясно, что эта задача значительно проще той, когда требуется узнать сообщение, о котором ничего не известно. Эксперименты показывают, что если длина кадра недостаточно велика, то рассматриваемые системы плохо выдерживают такой тест. В большинстве случаев в аппаратуре данного типа кадры делятся на число сегментов, заключенное в пределах от 8 до

16, причем обычно длительность каждого сегмента составляет от 20 до 60 мс.

Помимо выбора длин кадров и сегментов важным параметром является перестановка. Очевидно, что одни перестановки лучше других, и необходимо определить, как их следует выбирать и как управлять их выбором. Как и для преобразований инвертирования и частотных перестановок, имеется несколько вариантов использования базовой системы. Можно выбрать одну фиксированную перестановку для преобразования каждого кадра. Другой вариант связан с выбором (при помощи ключа) нескольких перестановок и периодическим их использованием. Лучшим способом является использование псевдослучайного генератора для выбора перестановки, применяемой для преобразования каждого кадра в отдельности. Для такого варианта актуален вопрос о длине периода соответствующей последовательности перестановок, так как повторное использование одной и той же перестановки нежелательно. В свою очередь, с этим связан выбор числа сегментов в кадре. Например, если это число равно 8 и каждый сегмент имеет длительность в 40 мс, то через 3,6 час. непрерывной работы перестановки начнут повторяться.

Как мы уже отмечали, далеко не все перестановки являются “хорошими” с точки зрения надежности шифрования. Например, если прослушать сигнал после применения каждой из двух подстановок

$$\begin{pmatrix} 12345678 \\ 13245768 \end{pmatrix}, \begin{pmatrix} 12345678 \\ 36258471 \end{pmatrix}, \quad (1)$$

то мы нашли бы в первом случае значительно более высокую остаточную разборчивость, чем во втором.

Замечание. В (1) рассматриваемые перестановки являются нижними строками подстановок, в которых верхние строки

представляют собой исходные порядки сегментов, а нижние — порядки сегментов после перестановки.

Причина указанного различия перестановок (1) состоит в том, что в первой из них символы 1, 4, 5, 8 остаются неподвижными, а остальные смещаются лишь на соседние позиции, тогда как во второй происходит лучшее перемешивание.

Рассмотренный пример приводит к естественной количественной мере “качества” перестановки. Пусть для произвольной подстановки α символ $\alpha(i)$ обозначает позицию, на которую α перемещает i -й сегмент. Тогда смещение символа i после перестановки равно $|i - \alpha(i)|$, а среднее смещение после перестановки характеризуется величиной

$$s(\alpha) = \frac{1}{n} \sum_{i=1}^n |i - \alpha(i)|.$$

Для первой подстановки из (1) среднее смещение $s(\alpha)$ равно 0,5, для второй — 2,5. Величина $s(\alpha)$ называется *сдвиговым фактором* подстановки α . Замечено, что перестановки, приводящие к выходному сигналу с низкой остаточной разборчивостью, имеют большой сдвиговой фактор, хотя обратное может быть неверным. В качестве примера приведем подстановку α восьми элементов со сдвиговым фактором 4, которая плохо выдерживает “тесты на слух”:

$$\begin{pmatrix} 12345678 \\ 57683241 \end{pmatrix}. \quad (2)$$

Помимо своего низкого сдвигового фактора первая подстановка в (1) имеет и другие нежелательные свойства. Рассмотрим, например, соседние сегменты 4 и 5. В скремблированном кадре они расположены в том же порядке, что и в исходном. Если сегменты имеют длительность в 40 мс, то рассматриваемая пара сегментов составляет около 80 мс. Как мы

уже отмечали, большинство фонем могут быть узнаваемы в таком интервале времени. В той же подстановке, а также в подстановке (2), сегменты 6 и 8 являются соседними. Это также нежелательно. Дело в том, что вообще при прослушивании в скремблированном сигнале пары соседних сегментов типа $i, i + 2$ человеческий мозг в состоянии, как правило, восстановить пропущенный сегмент $i + 1$, то есть восстановить соответствующую часть сообщения. Нечто подобное имеет место и в других случаях.

Таким образом, в рассмотренных ситуациях также идет речь о некоторой остаточной разборчивости. Это свидетельствует о сложности формализации определения “хороших” с точки зрения защиты перестановок, а следовательно, и сложности их подсчета. Поэтому имеются существенные различия в подсчете числа “хороших” перестановок, это может зависеть от субъективных предпочтений разработчика.

Теперь нужно решить вопрос о способе выбора перестановок с помощью ключа. Имеются два естественных способа такого выбора. Первый состоит в выборе произвольной перестановки данной степени с последующим ее тестированием. В зависимости от того, подходит она или нет, перестановка используется для преобразования кадра. Другой способ состоит в предварительном отборе всех “хороших” перестановок в ROM (памяти лишь для чтения), имеющейся в самом оборудовании, и их выбором для использования с помощью псевдослучайной последовательности. Рассмотрим оба способа.

Наиболее неблагоприятным для первого способа является фактор времени. В конце промежутка времени, равного длительности кадра, мы должны выбрать следующую подходящую перестановку. При этом нежелательно повторение той же перестановки, что в принципе возможно даже для случайной управляющей последовательности. Поэтому необходим контроль, устраняющий появление неподходящих перестановок. Ожидание подходящей перестановки требует дополнительных временных затрат, что нежелательно.

Второй метод использует лишь те перестановки, которые записаны в ROM. Если их запас не слишком велик, то это улучшает шансы противника. В случае когда кадр состоит из не слишком большого числа сегментов, скажем 8, и имеется возможность хранить все “хорошие” перестановки, второй метод предпочтителен. Чтобы понять еще одно преимущество второго метода, необходимо рассмотреть вопрос о возможностях перехватчика, обладающего той же аппаратурой и полным набором “хороших” перестановок.

Предположим, что одной из перестановок, хранящихся в ROM, является вторая перестановка из (1), и мы ее использовали для перемешивания кадра. Перехватчик, желая определить используемую перестановку, может перебирать перестановки, обратные к хранимому запасу перестановок. Если наша память содержит также нижнюю строку подстановки

$$\begin{pmatrix} 12345678 \\ 36258417 \end{pmatrix}, \quad (3)$$

то перехватчик может опробовать и ее (вместо использованной). Результатом последовательного применения исходной подстановки и обратной к подстановке (3) является подстановка

$$\begin{pmatrix} 12345678 \\ 12345687 \end{pmatrix}.$$

Она так близка к тождественной подстановке, что практически всегда дает возможность противнику восстановить исходный кадр. Помимо (3) есть и другие перестановки, “близкие” к истинной. В случае когда кадр состоит из 8 сегментов, таких пар “близких” перестановок имеется достаточно много и ситуация достаточно опасна (с точки зрения защиты). Дело в том, что мы должны скорректировать определение “хорошей” перестановки и тем самым уменьшить их число в памяти. Надо избегать записывать в память пары перестано-

вок, соответствующих подстановкам α и β , для которых произведение $\alpha \cdot \beta^{-1}$ или $\beta \cdot \alpha^{-1}$ близко к тождественной подстановке. Если ROM заполняется с учетом сделанной коррекции и число хранимых перестановок достаточно велико, то второй метод их выбора для перемешивания кадров становится более предпочтительным.

§ 10.5. Стойкость систем временных перестановок

Рассмотрим сначала вопрос о возможности восстановления информации, содержащейся в скремблированном сигнале, путем непосредственного прослушивания. С точки зрения разработчика необходимо найти баланс между минимальной остаточной разборчивостью и минимальной временной задержкой.

Для уменьшения остаточной разборчивости имеется ряд способов. Один из них состоит в простом реверсировании порядка следования сегментов. Наблюдения показывают, что при использовании такого способа уровень успешного прослушивания уменьшается почти на 10%. Другой метод также имеет отношение к частотной области. Здесь имеется в виду совместное использование частотного и временного перемешивания в одной двумерной системе. Хотя такой метод уменьшает уровень успешного прослушивания почти на 20%, он более дорог в реализации. Отметим при этом, что любые изменения сигнала уменьшают качество воспроизведения и что частотные искажения, в частности, сильно зависят от шумов и нелинейности при передаче.

Используя подобные методы или их комбинацию, можно уменьшить остаточную разборчивость до такого уровня, что будет невозможно воспринимать на слух никакое сообщение. Теперь рассмотрим вопрос о стойкости системы против более изощренных атак.

Одна из них состоит в попытках переупорядочить речевой сигнал кадр за кадром. Эта задача решается с использованием прибора, называемого *сонографом*. Этот прибор воспроизводит сонограмму каждого кадра. *Сонограмма* — это трехмерный график в системе координат время (по горизонтали), частота (по вертикали), амплитуда (по третьей координате), использующий “серую шкалу”. В этой шкале черный цвет представляет максимальную амплитуду и белый цвет — минимальную. Изменения амплитуды представляются изменениями оттенков серого цвета. Светлее оттенок, соответствующий меньшей амплитуде. Таким образом, хотя сонограмма имеет три измерения, она обычно представляется в двумерном виде.

Дескремблирование некоторого числа кадров путем опробования содержания ROM может позволить определить часть псевдослучайной последовательности, достаточной для определения ключа. Для противодействия этому необходим соответствующий генератор псевдослучайной последовательности, стойкий к подобной угрозе.

Предположим, что наша система стойка к описанному подходу. Это означает, что единственный путь, при котором криптоаналитик может получить сообщение, состоит в дескремблировании каждого кадра. Но тогда, очевидно, время, необходимое для восстановления сообщения, прямо пропорционально числу кадров. Криптоаналитик может автоматизировать процесс перебора перестановок, содержащихся в ROM, для проверки критерия того, что полученный сигнал является речевым сигналом (это можно сделать, например, по сонограмме). Для защиты от этой возможности вновь встает вопрос об увеличении числа “хороших” перестановок, что требует увеличения длительности кадра и временной задержки при передаче.

Как мы убедились ранее, временная задержка при передаче преобразованного кадра может быть в два раза больше длительности самого кадра. Это является следствием того, что для некоторых перестановок сегмент мог быть задержан на полную длительность кадра. С целью уменьшения подобной задержки

можно еще более ограничить множество используемых перестановок, добиваясь того, чтобы каждый сегмент задерживался “не слишком долго”. Это достигается при использовании перестановок с относительно небольшими смещениями для каждого символа.

Подытожим рассмотрение скремблеров.

Скремблеры характеризуются аналоговым выходом, лежащим в том же диапазоне, что и исходный сигнал. Кроме того, они обычно имеют характерные спектральные характеристики и выходной сигнал, представляющий собой последовательность фонем открытой речи (с измененным порядком следования). Их стойкость зависит как от типа скремблирования, так и от способа его реализации. В частности, использование зависящего от ключа псевдослучайного генератора для скремблирования может значительно увеличить уровень стойкости. Надежность любого выбранного метода скремблирования зависит в значительной степени от типа и качества канала связи. Скремблеры варьируются от простейших инверторов до сложных частотно-временных систем относительно высокой стойкости. Обычно они применяются как системы шифрования временной стойкости.

§ 10.6. Системы цифровой телефонии

Для преобразования речевого сигнала в цифровую форму берутся *отсчеты*, то есть значения сигнала через равные промежутки времени τ . Интервал τ должен быть настолько мал, чтобы сигнал не успевал намного измениться между отсчетами. Этот интервал часто называют *временным шагом* или *интервалом Найквиста*. Минимальную частоту взятия отсчетов, то есть величину, обратную временному шагу дискретизации, определяет *теорема В.А.Котельникова*, согласно которой частота отсчетов должна быть вдвое больше максимальной частоты звукового спектра. В телефонии такая частота ограничивается 3,4 кГц. Поэтому частота отсчетов должна быть не менее 6800

в секунду, или 6,8 кГц. Процесс взятия отсчетов называют *дискретизацией по времени*.

Для цифровой оценки отсчетов используется *процесс дискретизации по уровню*. Каждый отсчет можно представить числом, соответствующим значению отсчета звукового напряжения. Например, если звуковое напряжение измерять в милливольтгах, то число целых милливольт и будет отсчетом, а 1 мВ — шагом дискретизации по уровню. Отношение максимальной амплитуды звукового напряжения к шагу квантования дает максимальное число, которое нужно получить при отсчетах. Оно определяет *динамический диапазон* передаваемого сигнала. Для передачи речи с удовлетворительным качеством достаточен динамический диапазон 30÷35 дБ, что соответствует числу шагов квантования 30 при отсчетах. Для передачи одного отсчета двоичным кодом в этом случае достаточно $\log_2 30 \approx 5$ разрядов. Для качественной передачи музыки число квантований должно быть не менее 10000, что соответствует динамическому диапазону 80 дБ. В этом случае для передачи одного отсчета потребуется $\log_2 10000 \approx 14$ разрядов.

Переход на цифровую передачу существенно улучшает качество связи. Но не даром. Оценим поток информации при телефонном разговоре.

Полагая полосу звуковых частот равной, как и выше, 3,4 кГц и частоту взятия отсчетов 6,8 кГц, получаем 6800 отсчетов в секунду. При 30 шагах квантования по уровню каждый отсчет занимает 5 разрядов. Следовательно, в секунду передается 34000 двоичных разрядов, или бит информации. Скорость передачи информации, измеренную в бит/с, можно выразить формулой $C = 2F \log_2 N$, где F — максимальная частота звукового спектра, N — число уровней квантования. Чтобы передать цифровой сигнал со скоростью 34 Кбит/с, нужна полоса частот, пропускаемых каналом связи, не менее 34 кГц.

Таким образом, при переходе к цифровому сигналу произошел как бы обмен полосы частот на отношение *сигнал/шум*, но обмен достаточно выгодный. Расширяя полосу частот в 10 раз при переходе к цифровой передаче, мы намного снижаем допустимое отношение *сигнал/шум* или *сигнал/помеха* в канале связи, и это при общем существенном улучшении качества передачи.

В заключение сделаем одно замечание. Для аналогово-цифровых преобразователей входной сигнал отсчитывается через регулярные интервалы времени и затем передается цифровая “аппроксимация”. Имеется и другой способ передачи информации. Если, например, входным сигналом является синусоида с частотой f , то вместо того, чтобы посылать цифровую аппроксимацию, мы могли бы просто сообщить получателю о параметрах синусоиды и предложить ему самому построить такой сигнал. Этот принцип заложен в основе аппаратов, называемых соответственно *вокодерами* и *липредорами*. С помощью таких аппаратов синтезируются цифровые речевые системы с низкоскоростным выходом ($1,2 \div 4,8$ Кбит/с).

Контрольные вопросы

1. Какие два разных способа шифрования аналоговых сигналов Вы знаете?
2. Какие преобразования используются при скремблировании аналоговых сигналов?
3. В чем (с точки зрения надежности защиты) состоят слабости преобразований сигналов в частотной области, во временной области?
4. Какая фундаментальная теорема лежит в основе цифровой обработки сигналов?
5. Какой метод шифрования аналоговых сигналов обеспечивает гарантированную стойкость?

Глава 11

Системы шифрования с открытыми ключами

Системы шифрования с открытыми ключами называют также *асимметричными системами*. Они следующим образом используются для организации конфиденциальной связи в сети пользователей.

Каждый из корреспондентов системы обладает ключом $k = (k_s, k_p)$, состоящим из *открытого ключа* k_s и *секретного ключа* k_p . Открытый ключ определяет правило зашифрования E_k , а секретный ключ — правило расшифрования D_k (см. гл. 2). Эти правила связаны соотношением $D_k(E_k(M)) = C$ для любого открытого текста M и любого зашифрованного текста C . Знание открытого ключа не позволяет за приемлемое время (или с приемлемой сложностью) определить секретный ключ.

Для удобства записи обозначим правила зашифрования и расшифрования (на выбранном ключе k) произвольного корреспондента A символами E_A и D_A соответственно.

Корреспондент B , желая послать конфиденциальное сообщение M корреспонденту A , получает копию E_A , вычисляет шифртекст $C = E_A(M)$, который направляет по каналу связи корреспонденту A . Получив сообщение C , корреспондент A применяет к нему преобразование D_A , получая открытый текст M .

Открытый ключ не требуется сохранять в тайне. Необходимо лишь обеспечить его аутентичность, что, как правило,

сделать легче, чем обеспечить рассылку и сохранность секретных ключей.

Как мы уже отмечали ранее, системы шифрования с открытыми ключами осуществляют блочное шифрование, поэтому открытый текст перед зашифрованием разбивается на блоки выбранного размера, которые последовательно преобразуются таким же образом, как это происходит при использовании блочного шифра в режиме простой замены (см. § 8.3).

Асимметричные системы шифрования обеспечивают значительно меньшие скорости шифрования, нежели симметричные, в силу чего они обычно используются не столько для шифрования сообщений, сколько для шифрования пересылаемых между корреспондентами ключей, которые затем используются в симметричных системах.

Рассмотрим конкретные примеры систем шифрования с открытыми ключами.

§ 11.1. Шифрсистема RSA

Система RSA была предложена в 1978 г. [Riv78] и в настоящее время является наиболее распространенной системой шифрования с открытым ключом. Напомним ее определение, приведенное в гл. 2 (определение 4).

Пусть $n = p \cdot q$ — целое число, представимое в виде произведения двух больших простых чисел p, q . Выберем числа e и d из условия

$$e \cdot d \equiv 1 \pmod{\varphi(n)}, \quad (1)$$

где $\varphi(n) = (p-1) \cdot (q-1)$ — значение функции Эйлера от числа n . Пусть $k = (n, p, q, e, d)$ — выбранный ключ, состоящий из открытого ключа $k_3 = (n, e)$ и секретного ключа

$k_p = (n, p, q, d)$. Пусть M — блок открытого текста и C — соответствующий блок зашифрованного текста. Тогда правила зашифрования и расшифрования определяются формулами:

$$C = E_k(M) = M^e \pmod{n}, \quad D_k(C) = C^d \pmod{n}. \quad (2)$$

Заметим, что в соответствии с (2) $D_k(C) = M$. Это вытекает из следующих рассуждений. Для любого целого числа M и любого простого p справедливо сравнение

$$M^p \equiv M \pmod{p}. \quad (3)$$

В самом деле, (3) равносильно сравнению

$$M^p - M \equiv 0 \pmod{p}$$

или сравнению

$$M(M^{p-1} - 1) \equiv 0 \pmod{p}. \quad (4)$$

Если $\text{НОД}(M, p) = p$, то p делит M , и поэтому $M \equiv 0 \pmod{p}$, откуда следует (4). Если же $\text{НОД}(M, p) = 1$, то, согласно малой теореме Ферма (см. Приложение 3), $M^{p-1} \equiv 1 \pmod{p}$, откуда также следует (4).

Согласно (1), существует целое число r , такое, что $e \cdot d = r \cdot \varphi(n) + 1$. Отсюда и из (3) получаем следующую цепочку равенств и сравнений:

$$\begin{aligned} C^d &= (M^e)^d = M^{ed} = M^{r\varphi(n)+1} = M^{r(p-1)(q-1)+1} = \\ &= M^{r p(q-1)} \cdot M^{-r q+r+1} = (M^p)^{r(q-1)} \cdot M^{-r q+r+1} \equiv \\ &\equiv M^{r(q-1)} \cdot M^{-r q+r+1} \equiv M \pmod{p}. \end{aligned} \quad (5)$$

Аналогично можно показать, что

$$C^d = M \bmod q. \quad (6)$$

Поскольку p и q — разные простые числа, то на основании известных свойств сравнений из (5), (6) получаем:

$$C^d = M \bmod n.$$

Отсюда и следует корректность определения (2).

Для того чтобы лучше представить себе технические детали, возникающие при зашифровании и расшифровании, приведем пример работы с RSA.

Пример

Зашифруем аббревиатуру RSA, используя $p = 17$, $q = 31$. Для этого вычислим $n = pq = 527$ и $\varphi(n) = (p-1)(q-1) = 480$. Выберем, далее, в качестве e число, взаимно простое с $\varphi(n)$, например $e = 7$. С помощью алгоритма Евклида найдем целые числа u и v , удовлетворяющие соотношению $e \cdot u + \varphi(n) \cdot v = 1$:

$$\begin{aligned} 480 &= 7 \cdot 68 + 4, \\ 7 &= 4 \cdot 1 + 3, \\ 4 &= 3 \cdot 1 + 1, \\ 1 &= 4 - 3 \cdot 1 = 4 - (7 - 4 \cdot 1) \cdot 1 = \\ &= 4 \cdot 2 - 7 \cdot 1 = (480 - 7 \cdot 68) \cdot 2 - 7 \cdot 1 = \\ &= 480 \cdot 2 - 7 \cdot 137, \\ v &= 2, \quad u = -137. \end{aligned}$$

Поскольку $-137 \equiv 343 \pmod{480}$, то $d = 343$. Проверка:

$$7 \cdot 343 = 2401 \equiv 1 \pmod{480}.$$

Теперь представим данное сообщение в виде последовательности чисел, содержащихся в интервале $\overline{0,526}$. Для этого буквы R, S и A закодируем пятимерными двоичными векторами, воспользовавшись двоичной записью их порядковых номеров в английском алфавите:

$$R = 18 = (10010), S = 19 = (10011), A = 1 = (00001).$$

Тогда $RSA = (100101001100001)$. Укладываясь в заданный интервал $\overline{0,526}$, получаем следующее представление:

$$RSA = (100101001), (100001) = (M_1 = 297, M_2 = 33).$$

Далее последовательно шифруем M_1 и M_2 :

$$C_1 = E_k(M_1) = M_1^e \equiv 297^7 \pmod{527} = 474.$$

При этом мы воспользовались тем, что

$$\begin{aligned} 297^7 &= ((297^2)^3 297) \pmod{527} = \\ &= ((200^3 \pmod{527}) 297) \pmod{527}, \\ C_2 &= E_k(M_2) = M_2^e \equiv 33^7 \pmod{527} = 407. \end{aligned}$$

В итоге получаем шифртекст: $y_1 = 474, y_2 = 407$.

При расшифровании нужно выполнить следующую последовательность действий. Во-первых, вычислить

$$D_k(C_1) = (C_1)^{343} \pmod{527}.$$

Отметим, что при возведении в степень удобно воспользоваться тем, что $343 = 256 + 64 + 16 + 4 + 2 + 1$. На основании этого представления получаем:

$$\begin{aligned} 474^2 &\equiv 174 \pmod{527}, & 474^4 &\pmod{527} \equiv 237, \\ 474^8 &\pmod{527} \equiv 307, & 474^{16} &\pmod{527} \equiv 443, \\ 474^{32} &\pmod{527} \equiv 205, & 474^{64} &\pmod{527} \equiv 392, \\ 474^{128} &\pmod{527} \equiv 307, & 474^{256} &\pmod{527} \equiv 443, \end{aligned}$$

в силу чего

$$\begin{aligned} 474^{343} \pmod{527} &\equiv \\ &\equiv (443 \cdot 392 \cdot 443 \cdot 237 \cdot 174 \cdot 474) \pmod{527} \equiv 297. \end{aligned}$$

Аналогично

$$407^{343} \pmod{527} \equiv 33.$$

Возвращаясь к буквенной записи, получаем после расшифрования RSA.

Проанализируем вопрос о стойкости системы RSA.

Нетрудно показать, что сложность нахождения секретного ключа системы RSA определяется сложностью разложения числа n на простые множители. В связи с этим нужно выбирать числа p и q таким образом, чтобы задача разложения числа n была достаточно сложна в вычислительном плане. Для этого рекомендуются следующие требования:

1) числа p и q должны быть достаточно большими, не слишком сильно отличаться друг от друга и в то же время быть не слишком близкими друг другу;

2) числа p и q должны быть такими, чтобы наибольший общий делитель чисел $p-1$ и $q-1$ был небольшим; желательно, чтобы $\text{НОД}(p-1, q-1) = 2$;

3) p и q должны быть сильно простыми числами (*сильно простым* называется такое простое число r , что $r+1$ имеет большой простой делитель, $r-1$ имеет большой простой делитель s , такой, что число $s-1$ также обладает достаточно большим простым делителем).

В случае когда не выполнено хотя бы одно из указанных условий, имеются эффективные алгоритмы разложения n на простые множители (см. [Сал96], [Неч99]).

В настоящее время самые большие простые числа, вида $n = p \cdot q$, которые удается разложить на множители известными методами, содержат в своей записи 140 десятичных знаков. Поэтому, согласно указанным рекомендациям, числа p и q в системе RSA должны содержать не менее 100 десятичных знаков.

Следует подчеркнуть необходимость соблюдения осторожности в выборе модуля RSA (числа n) для каждого из корреспондентов сети. В связи с этим можно сказать следующее.

Читатель может самостоятельно убедиться в том, что, зная одну из трех величин: p , q или $\varphi(n)$, можно легко найти секретный ключ RSA. Известно также, что, зная секретную экспоненту расшифрования d , можно легко разложить модуль n на множители. В этом случае удастся построить вероятностный алгоритм разложения n . Отсюда следует, что каждый корреспондент сети, в которой для шифрования используется система RSA, должен иметь свой уникальный модуль.

В самом деле, если в сети используется единый для всех модуль n , то такая организация связи не обеспечивает конфиденциальности, несмотря на то что базовая система RSA может быть стойкой. Выражаясь другими словами, говорят о *несостоятельности* протокола с общим модулем. Несостоятельность следует из того, что знание произвольной пары экспонент (e_i, d_i) позволяет, как было отмечено, разложить n на множители. Поэтому любой корреспондент данной сети имеет возможность найти секретный ключ любого другого корреспондента. Более того, это можно сделать даже без разложения n на множители [Сал96].

Как отмечалось ранее, системы шифрования с открытыми ключами работают сравнительно медленно. Для повышения скорости шифрования RSA на практике используют малую экспоненту зашифрования.

Если выбрать число e небольшим или таким, чтобы в его двоичной записи было мало единиц, то процедуру шифрования можно значительно ускорить. Например, выбрав $e = 3$ (при этом ни $p-1$, ни $q-1$ не должны делиться на 3), мы сможем реализовать шифрование с помощью одного возведения в квадрат по модулю n и одного перемножения. Выбрав $e = 2^{16} + 1 = 65537$ — число, двоичная запись которого содержит только две 1, мы сможем реализовать шифрование с помощью 16 возведений в квадрат по модулю n и одного перемножения. Если экспонента e выбирается случайно, то реализация шифрования по алгоритму RSA потребует s возведений в квадрат по модулю n и в среднем $s/2$ умножений по тому же модулю, где s — длина двоичной записи числа n . Вместе с тем выбор небольшой экспоненты e может привести к негативным последствиям. Дело в том, что у нескольких корреспондентов могут оказаться одинаковые экспоненты e .

Пусть, например, три корреспондента имеют попарно взаимно простые модули n_1, n_2, n_3 и общую экспоненту $e = 3$. Если еще один пользователь посылает им некое циркулярное сообщение M , то криптоаналитик противника может получить в свое распоряжение три зашифрованных текста $y_i \equiv M^3 \pmod{n_i}$, $i = 1, 2, 3$. Далее он может найти решение у системы сравнений

$$\begin{cases} y \equiv y_1 \pmod{n_1}, \\ y \equiv y_2 \pmod{n_2}, \\ y \equiv y_3 \pmod{n_3}, \end{cases}$$

лежащее в интервале $0 < y < n_1 \cdot n_2 \cdot n_3$. По китайской теореме об остатках (см. Приложение 3) такое решение единственно, а так как $M^3 < n_1 \cdot n_2 \cdot n_3$, то $y = M^3$. Само M можно найти, вычисляя кубический корень: $M = \sqrt[3]{y}$.

Отметим, что выбор малой экспоненты расшифрования d также нежелателен в связи с возможностью определения d простым перебором. Известно также, что если $d < \sqrt[4]{n}$, то экспоненту d легко найти, используя непрерывные дроби.

§ 11.2. Шифрсистема Эль-Гамала

Шифрсистема Эль-Гамала была предложена в 1985 г. [ElG85] и является фактически одним из вариантов метода выработки открытых ключей Диффи-Хеллмана (который будет рассмотрен далее). Криптографическая стойкость данной системы основана на сложности проблемы логарифмирования в мультипликативной группе конечного простого поля.

В соответствии с терминологией, введенной в гл. 2, шифр-система Эль-Гамала (X, K, Y, E, D) определяется следующим образом. Для нее

$$X = Z_p^*, Y = Z_p^* \times Z_p^*, K = \{(p, \alpha, \beta, a) \mid \alpha^a \equiv \beta \pmod{p}\},$$

где p — достаточно большое простое число, α — порождающий элемент группы Z_p^* , a — целое число из интервала $1 \leq a \leq p-2$. Ключ $k = (p, \alpha, \beta, a)$ представляется в виде открытого ключа $k_3 = (p, \alpha, \beta)$ и секретного ключа $k_p = (a)$.

Правило зашифрования на ключе k определяется формулой

$$E_k(M) = (C_1, C_2),$$

где

$$C_1 \equiv \alpha^r \pmod{p}, C_2 \equiv M \cdot \beta^r \pmod{p},$$

а r — случайно выбираемое число (рандомизатор) из интервала $0 \leq r \leq p-2$.

Правило расшифрования на ключе k определяется формулой

$$D_k(C_1, C_2) = C_2 \cdot (C_1^a)^{-1} \pmod{p}.$$

Несложно проверить, что такое определение корректно, то есть что выполняется равенство $D_k(E_k(M)) = M$ при любых $k \in K$ и $M \in X$.

Введение в правило зашифрования рандомизатора r делает шифр Эль-Гамала шифром многозначной замены (см. гл. 5). В связи со случайным характером выбора параметра r подобную схему шифрования называют еще схемой *вероятностного шифрования*. Для нее открытый текст и ключ не определяют шифртекст однозначно.

Для выработки открытого и секретного ключей каждый из абонентов системы осуществляет следующие операции:

1) выбирает большое простое число p и некоторый порождающий элемент α группы Z_p^* ;

2) случайно выбирает целое число a , $1 \leq a \leq p-2$, и вычисляет $\beta \equiv \alpha^a \pmod{p}$;

3) публикует открытый ключ (p, α, β) , оставляя в секрете число a .

Следует отметить, что в приведенной системе необходимо использовать различные значения рандомизатора r для зашифрования различных открытых текстов M и M' . В самом деле, в противном случае соответствующие шифртексты (C_1, C_2) и (C'_1, C'_2) оказываются связанными соотношением $C_2 \cdot (C'_2)^{-1} = M \cdot (M')^{-1}$ и M' может быть легко вычислен, если известен M .

Как уже отмечалось, стойкость системы Эль-Гамала определяется сложностью решения задачи дискретного логарифмирования в Z_p^* . В настоящее время эта задача практически нереализуема для значений p , содержащих не менее 150 десятичных знаков. Рекомендуется также, чтобы число $p-1$ содержало большой простой делитель.

Система Эль-Гамала может быть обобщена для применения в любой конечной циклической группе G . Криптографическая стойкость такой обобщенной схемы определяется сложностью задачи логарифмирования в группе G . При этом групповая операция в G должна быть легко реализуемой. В качестве G чаще всего выбираются следующие три группы:

1) мультипликативная группа Z_p^* целых чисел по модулю простого числа p ;

2) мультипликативная группа $GF(2^m)^*$ конечного поля $GF(2^m)$ характеристики 2;

3) группа точек эллиптической кривой над конечным полем.

Вероятностный характер шифрования можно отнести к достоинствам системы Эль-Гамала, так как схемы вероятностного шифрования обладают, как правило, большей стойкостью по сравнению со схемами с детерминированным процессом шифрования. Недостатком системы является удвоение длины открытого текста при шифровании.

§ 11.3. Шифрсистема Мак-Элиса

Идея, лежащая в основе данной системы, состоит в выборе корректирующего кода, исправляющего определенное число ошибок, для которого существует эффективный алгоритм декодирования. С помощью секретного ключа этот код “маскируется” под общий линейный код, для которого, как известно, задача декодирования не имеет эффективного решения.

В системе Мак-Элиса параметрами системы, общими для всех абонентов, являются целые числа k , n и t . Для получения открытого и соответствующего секретного ключа каждому из абонентов системы следует осуществить следующую последовательность действий:

1) выбрать порождающую матрицу $G = G_{k \times n}$ двоичного (n, k) -линейного кода, исправляющего t ошибок, для которого известен эффективный алгоритм декодирования;

2) случайно выбрать двоичную невырожденную матрицу $S = S_{k \times k}$;

3) случайно выбрать подстановочную матрицу $P = P_{n \times n}$;

4) вычислить произведение матриц $G_1 = S \cdot G \cdot P$.

Открытым ключом является пара (G_1, t) , секретным — тройка (S, G, P) .

Для того чтобы зашифровать сообщение M , предназначенное для абонента A , абоненту B следует выполнить следующие действия:

- 1) представить M в виде двоичного вектора длины k ;
- 2) выбрать случайный бинарный вектор ошибок Z длины n , содержащий не более t единиц;
- 3) вычислить бинарный вектор $C = M \cdot G_A + Z$ и направить его абоненту A .

Получив сообщение C , абонент A вычисляет вектор $C_1 = C \cdot P^{-1}$, с помощью которого, используя алгоритм декодирования кода с порождающей матрицей G , получает далее векторы M_1 и $M = M_1 \cdot S^{-1}$.

Чтобы убедиться в корректности приведенного алгоритма расшифрования, достаточно заметить, что

$$\begin{aligned} C_1 &= C \cdot P^{-1} = (M \cdot G_A + Z) \cdot P^{-1} = \\ &= (M \cdot S \cdot G \cdot P + Z) \cdot P^{-1} = \\ &= (M \cdot S) \cdot G + Z \cdot P^{-1}, \end{aligned}$$

где $Z \cdot P^{-1}$ — вектор, содержащий не более t единиц. Поэтому алгоритм декодирования кода с порождающей матрицей G декодирует C в $M_1 = M \cdot S$.

В качестве кода, исправляющего ошибки в системе Мак-Элиса, можно использовать код Гоппы (см., например, [Пит64]). Известно, что для любого неприводимого полинома $g(x)$ степени t над полем $GF(2^m)$ существует бинарный код Гоппы длины $n = 2^m$ и размерности $k \geq n - mt$, исправляющий до t ошибок включительно, для которого имеется

эффективный алгоритм декодирования. В настоящее время не известны эффективные алгоритмы дешифрования системы Мак-Элиса, использующей код Гоппы, при правильном выборе параметров системы.

Вместе с тем рекомендуемые параметры для этой системы [Men97] — $n = 1024$, $t = 38$, $k > 644$ — приводят к тому, что открытый ключ имеет размер около 2^{19} бит, а длина сообщения увеличивается при шифровании примерно в 1,6 раза, в связи с чем данная система не получила широкого распространения.

§ 11.4. Шифрсистемы на основе “проблемы рюкзака”

“Проблема рюкзака” (или “ранца”) может быть сформулирована следующим образом. Пусть задано множество натуральных чисел $A = \{a_1, a_2, \dots, a_n\}$ и натуральное число S . Требуется установить, имеется ли такое подмножество множества A , сумма элементов которого была бы равна S . Эквивалентной является следующая формулировка: существует ли такой набор чисел $x_i \in \{0, 1\}$, $i \leq n$, для которого

$$\sum_{i=1}^n a_i \cdot x_i = S.$$

Данная проблема получила свое название в связи с тем, что поставленная задача может быть переформулирована также в следующем виде. Имеется набор предметов с известными весами и рюкзак, который может выдержать вес, не превышающий заданной величины. Можно ли выбрать набор предметов для погрузки в рюкзак так, чтобы они в точности имели максимально возможный вес.

“Проблема рюкзака” является весьма сложной, ее решение с полиномиальной сложностью в настоящее время не известно.

Идея построения системы шифрования на основе “проблемы рюкзака” заключается в выделении некоторого подкласса задач об укладке рюкзака, решаемых сравнительно легко, и “маскировки” задач этого класса (с помощью некоторого преобразования параметров) под общий случай. Параметры подкласса определяют секретный ключ, а параметры модифицированной задачи — открытый ключ. В качестве легко решаемой задачи Р. Меркль и М. Хеллман в 1978 г. [Mer81] предложили задачу об укладке “супервозрастающего” рюкзака. Изложим ее суть.

Назовем *супервозрастающей* последовательность натуральных чисел (b_1, b_2, \dots, b_n) , обладающую свойством

$$b_i > \sum_{j=1}^{i-1} b_j, \quad 2 < i < n.$$

Можно убедиться в том, что “проблема рюкзака” для супервозрастающей последовательности может быть решена с помощью процедуры, состоящей в выполнении следующих шагов:

1. Положить $i = n$.
2. Если $i > 1$, то положить x_i равным 1 и S равным $S - b_i$, если $S > b_i$, и положить x_i равным 0 в противном случае.
3. Положить i равным $i - 1$ и возвратиться к шагу 2.

В системе, основанной на проблеме рюкзака, величина n является параметром системы.

Для вычисления открытого и соответствующего секретного ключа каждый из абонентов системы осуществляет следующую последовательность действий.

1. Выбирает супервозрастающую последовательность (b_1, b_2, \dots, b_n) и модуль m , такой, что $m > \sum_{i=1}^n b_i$.
2. Выбирает случайное число W , $1 < W < m - 1$, такое, что $\text{НОД}(W, m) = 1$.
3. Выбирает случайную перестановку π чисел $\{1, 2, \dots, n\}$.
4. Вычисляет $a_i = W \cdot b_{\pi(i)} \bmod m$ для $i = \overline{1, n}$.

Открытым ключом является набор (a_1, a_2, \dots, a_n) , секретным ключом — набор $(\pi, m, W, (b_1, \dots, b_n))$.

Чтобы зашифровать сообщение M , предназначенное для абонента A , абонент B осуществляет следующие шаги с помощью открытого ключа (a_1, a_2, \dots, a_n) абонента A :

1. Представляет M в виде бинарной последовательности $M = M_1 M_2 \dots M_n$ длины n .
2. Вычисляет $C = \sum_{i=1}^n M_i \cdot a_i$ и направляет его к A .

Абонент A , получив C , вычисляет $H = W^{-1} \cdot C \bmod m$, а затем, решая “проблему рюкзака” для супервозрастающей последовательности, находит числа $z_i \in \{0, 1\}$, такие, что

$$H = \sum_{i=1}^n z_i \cdot b_i.$$

Биты последовательности M_i вычисляются по формуле

$$M_i = z_{\pi(i)}, \quad i = \overline{1, n}.$$

Корректность проведенной процедуры расшифрования вытекает из следующих рассуждений. Поскольку

$$H \equiv W^{-1} \cdot C \equiv W^{-1} \cdot \sum_{i=1}^n M_i \cdot a_i \equiv \sum_{i=1}^n M_i \cdot b_{\pi(i)} \pmod{m}$$

и $0 < H < m$, то $H = \sum_{i=1}^n M_i \cdot b_{\pi(i)}$, и, следовательно, алгоритм решения “проблемы рюкзака” действительно находит биты открытого текста, переставленные в соответствии с перестановкой π .

Вместе с тем доказано, что существует алгоритм полиномиальной сложности, который может быть использован противником для получения открытого текста M по шифртексту C . Этот алгоритм, исходя из a_i , находит пару таких целых чисел u_1, m_1 , что отношение u_1/m_1 близко к отношению u/m (где $u = W^{-1} \pmod{m}$, а W, m являются частью секретного ключа). Кроме того, числа $B_i \equiv u_1 \cdot a_i \pmod{m}$, $1 < i < n$, образуют супервозрастающую последовательность. Эта последовательность затем используется противником вместо (b_1, \dots, b_n) для дешифрования сообщения.

Контрольные вопросы

1. В чем состоят преимущества систем с открытыми ключами перед симметричными шифрсистемами?
2. Сложностью какой математической задачи определяется стойкость системы RSA?
3. К какому типу принадлежит схема шифрования, используемая в системе Эль-Гамала? В чем ее преимущества?
4. Чем вызваны трудности в практической реализации системы Мак-Элиса?
5. Придумайте алгоритм вычисления $a^d \pmod{n}$, имеющий сложность $O(\ln n)$.
6. Постройте пример шифра Эль-Гамала для $p = 127$. Зашифруйте и расшифруйте выбранное Вами $m \leq 126$.

Глава 12

Идентификация

Все протоколы идентификации включают двух участников:

1) A — доказывающего — участника, проходящего идентификацию,

и

2) B — проверяющего — участника, проверяющего аутентичность доказывающего.

Целью протокола является проверка того, что проверяемым действительно является A .

С точки зрения проверяющего возможными исходами протокола являются либо принятие решения об идентичности доказывающего A , либо завершение протокола без принятия такого решения.

Протоколы идентификации могут быть разбиты на три большие категории в зависимости от того, на чем основана идентификация.

1. Протоколы, основанные на известной обеим сторонам информации. Такой информацией могут быть пароли, личные идентификационные номера (PIN от английского *personal identification number*), секретные или открытые ключи, знание которых демонстрируется во время выполнения протокола.
2. Протоколы, использующие некоторые физические приборы, с помощью которых и проводится идентификация. Таким прибором может быть магнитная или интеллектуальная пластиковая карта, или прибор, генерирующий меняющиеся со временем пароли.
3. Протоколы, использующие физические параметры, составляющие неотъемлемую принадлежность доказывающего. В качестве таковых могут выступать подписи, отпе-

чатки пальцев, характеристики голоса, геометрия руки и т. д. Протоколы, использующие такие параметры, не носят криптографического характера и не будут далее рассматриваться.

Одной из основных целей идентификации является обеспечение контроля доступа к определенным ресурсам, таким, как банковские счета, телекоммуникационные каналы, компьютерные программы, базы данных, здания, сооружения и т. д. Идентификация также обычно является неотъемлемой частью протокола распределения ключей.

Протоколы идентификации тесно связаны с протоколами цифровой подписи, но проще их. Последние имеют дело с меняющимися по содержанию сообщениями и обычно включают элементы, обеспечивающие невозможность отказа от подписанного сообщения. Для протоколов идентификации содержание сообщения, по существу, фиксировано — это заявление об аутентичности доказывающего A в текущий момент времени.

§ 12.1. Фиксированные пароли (слабая идентификация)

Обычная парольная схема основывается на не зависящих от времени паролях и устроена следующим образом. Для каждого пользователя имеется пароль, обычно представляющий собой последовательность длиной от 6 до 10 знаков алфавита, которую пользователь в состоянии запомнить. Эта последовательность выступает в качестве общего секрета пользователя и системы. Для того чтобы получить доступ к системному ресурсу (база данных, принтер и т. д.), пользователь представляет свой идентификатор и пароль и прямо или косвенно определяет необходимый ресурс. При этом идентификатор пользователя выступает как заявка на идентификацию, а пароль — как подтверждение этой заявки. Различные пароль-

ные схемы отличаются между собой по методам хранения парольной информации в системе и методам ее проверки.

В число угроз данной схеме идентификации, допускающих возможность проникновения в систему, входят раскрытие пароля (вне системы) и перехват информации с линий связи (внутри системы). Угрозой является также опробование паролей, в том числе с использованием словарей.

Остановимся подробнее на методах хранения паролей в системе. Наиболее очевидным из них является хранение паролей в открытом виде в файле, защищенном от записи и считывания системными средствами. При обращении пользователя система сравнивает введенный пароль с паролем данного пользователя, хранимым в файле. Недостаток этого метода состоит в том, что пароли не защищены от привилегированных пользователей системы.

Для устранения этого недостатка используются зашифрованные файлы паролей пользователей. При этом может использоваться либо непосредственное зашифрование паролей с помощью того или иного криптографического алгоритма, либо вычисление значения хэш-функции (см. гл. 13) пароля. Заметим, что использование шифрования перед передачей пароля по незащищенному каналу связи хотя и защищает сам пароль, но не защищает от возможности вхождения противника в систему путем навязывания перехваченного зашифрованного пароля.

Поскольку степень защиты указанной системы определяется сложностью перебора паролей, то на практике используется целый ряд приемов для усложнения противнику этой процедуры.

Правила составления паролей

Типичным требованием, предъявляемым при составлении паролей, является ограничение на минимальную длину паролей. Примером может служить требование, чтобы пароль

имел длину не менее 9 букв или цифр. Другими требованиями могут быть наличие в пароле хотя бы одного символа из каждого регистра (верхнего, числового, не алфавитно-цифрового и т. д.), а также требование, чтобы пароль не мог быть словом из имеющегося словаря или частью идентификационной информации доказывающего.

Очевидно, что сложность проникновения в систему определяется сложностью простейшего из паролей зарегистрированных пользователей. Лучшими из возможных паролей являются те, которые выбираются случайно и равновероятно из всех слов в данном алфавите. Такой выбор может быть реализован с помощью физического датчика случайных чисел.

Усложнение процедуры проверки паролей

Для того чтобы усложнить атаки, включающие опробование большого числа паролей, функция, обеспечивающая проверку паролей (например, однонаправленная функция), может быть усложнена путем применения нескольких итераций более простых функций. Вместе с тем число этих итераций не должно быть слишком большим, чтобы не затруднить доступ в систему законным пользователям. Необходимо также следить за тем, чтобы в результате этих процедур не упростилась задача определения паролей.

“Подсоленные” пароли

Для уменьшения эффективности атак с использованием словаря каждый пароль, при помещении его в словарь, может быть дополнен n -битовой случайной последовательностью, называемой “солью” (она изменяет “вкус” пароля) перед применением однонаправленной функции. Как хэшированный пароль, дополненный “солью”, так и “соль” записываются в файл паролей. Когда в последующем пользователь предъявляет пароль, система находит “соль” и применяет однонаправленную функцию к паролю, дополненному “солью”.

Хотя сложность опробования каждого варианта пароля при дополнении “солью” не изменяется, тем не менее добавление “соли” усложняет атаку с целью получения хотя бы одного пароля из многих. Это происходит потому, что в словаре должно содержаться 2^n вариантов каждого опробуемого слова, что влечет за собой большие требования к памяти, содержащей зашифрованный словарь, и соответственно увеличивает время на его изготовление.

Парольные фразы

Для того чтобы увеличить неопределенность используемых паролей и вместе с тем не нарушить такое их важное качество, как возможность запоминания человеком, часто используются *парольные фразы*. В этом случае в качестве пароля используется целое предложение вместо короткого слова. Парольные фразы хэшируются до фиксированной длины и играют ту же роль, что и обычные пароли. При этом фраза не должна просто сокращаться до фиксированной длины. Идея этого метода состоит в том, что человеку легче запомнить фразу, чем случайную последовательность букв или цифр. Парольные фразы обеспечивают большую безопасность, чем короткие пароли, но требуют большего времени для ввода.

§ 12.2. Атаки на фиксированные пароли

Повторное использование паролей

Принципиальной слабостью схем, неоднократно используемых фиксированные пароли, является возможность получения противником этих паролей одним из следующих способов:

- путем просмотра при введении с клавиатуры,
- путем получения документов, содержащих эти пароли,

— путем перехвата их из каналов связи, используемых пользователями для связи с системой, или из самой системы, поскольку пароли используются в открытом виде.

Все это дает возможность противнику осуществить доступ к системе, имитируя законного пользователя.

Таким образом, фиксированные пароли нельзя использовать в случае их передачи по незащищенным каналам связи, в особенности по эфиру.

Заметим, что при использовании процедуры идентификации на абонентском пункте для получения доступа к ресурсу сервера (например, при идентификации кредитной карточки банкоматом в режиме *on-line*) системный ответ (“идентификация А” или “завершение протокола без идентификации”) должен быть также защищен, как и представляемый пароль. Он должен меняться во времени, чтобы не дать противнику возможность использовать системный ответ “идентификация А”.

Тотальный перебор паролей

Простейшей атакой противника на систему с фиксированными паролями является перебор всех возможных вариантов до тех пор, пока истинный пароль не будет найден. Особенно опасна эта атака в режиме *off-line*, поскольку в этом случае не требуется непосредственного контакта доказывающего с проверяющим, и поэтому число безуспешных попыток проверки пароля в единицу времени не может быть ограничено, как это обычно делается при проверке в режиме *on-line*.

Эффективность указанной атаки напрямую зависит от числа попыток до обнаружения первого пароля, обеспечивающего доступ в систему или к ее ресурсу, а также от временной сложности реализации каждой из таких попыток.

Атаки с помощью словаря

Вместо тотального перебора всех возможных паролей противник может прибегнуть к их перебору в порядке убыва-

ния вероятности их использования. Этот метод обычно эффективен в том случае, когда пароли выбираются пользователем и, как правило, являются неравновероятными. Во многих случаях эти пароли выбираются из словаря. Обычные словари содержат, как правило, не более 150 тыс. слов, что намного меньше числа всех всего лишь 6-буквенных паролей.

В этом случае противником может быть реализована так называемая *атака с использованием словаря*. Для повышения эффективности атаки противником создается и записывается на диск хэшированный файл слов из используемого для выбора паролей словаря. Хэшированные значения из файла системных паролей могут быть отсортированы (используя обычные алгоритмы сортировки) и сравнены со словами из созданного файла. Атаки с использованием словаря, как правило, приводят не к нахождению пароля фиксированного пользователя, а к нахождению одного или нескольких паролей, обеспечивающих доступ в систему.

Личные идентификационные номера

Личные идентификационные номера относятся к категории фиксированных паролей. Они обычно используются в приложениях, связанных с пластиковыми картами, для доказательства того, что данный участник является действительным владельцем карты и как таковой имеет право доступа к определенным системным ресурсам. Ввод личного идентификационного номера требуется во всех случаях, когда используется пластиковая карта. Это обеспечивает дополнительный рубеж безопасности, если карта утеряна или похищена.

В силу исторических обстоятельств (и для удобства пользователей) личный идентификационный номер является цифровым и содержит от 4-х до 8 цифр. Для того чтобы обеспечить защиту от его тотального перебора, применяются дополнительные организационные меры. Например, большинство банкоматов при трехкратном вводе неправильного номера

блокируют кредитную карту. Для ее разблокирования требуется ввести уже более длинный номер. Поскольку люди, как правило, не могут запомнить ключи настолько длинные, чтобы с их помощью можно было бы обеспечить необходимую информационную безопасность системы, часто используется следующая двухступенчатая процедура идентификации пользователя.

Сначала с помощью личного идентификационного номера проверяется личность лица, вводящего пластиковую карту, а затем содержащаяся в пластиковой карте дополнительная ключевая информация используется для идентификации его в системе (как действительного владельца пластиковой карты, имеющего определенные права доступа в системе). Таким образом, пользователь, имеющий пластиковую карту, должен помнить только короткий личный идентификационный номер, в то время как более длинный ключ, содержащийся на карте, обеспечивает необходимый уровень криптографической безопасности при идентификации в системе, использующей незащищенные каналы связи.

Одноразовые пароли

Повышению надежности идентификации служит использование *одноразовых паролей*, то есть паролей, которые могут быть использованы для идентификации только один раз. Такие схемы обеспечивают защиту от противника, использующего перехват паролей из незащищенного канала.

Существуют три схемы использования одноразовых паролей:

- 1) пользователи системы имеют общий список одноразовых паролей, который доставляется по защищенному от перехвата каналу связи;
- 2) первоначально пользователь и система имеют один общий секретный пароль. Во время идентификации, использующей пароль t , пользователь создает и передает в систему

новый пароль $(t + 1)$, зашифрованный на ключе, полученном из пароля t ;

3) пользователи системы используют одноразовые пароли на основе однонаправленной функции.

При использовании третьей схемы пользователь начинает с секретного пароля w и с помощью однонаправленной функции $H(w)$ вырабатывает итерации

$$H(w), H(H(w)), \dots, \underbrace{H(H(\dots(H(w)\dots))}_n = H^n(w).$$

Паролем для i -й идентификации, $1 \leq i \leq n$, является $w_i = H^{n-i}(w)$.

Для того чтобы идентифицировать себя при i -й попытке, A передает B строку (A, i, w_i) . B проверяет соответствие полученного i номеру попытки и равенство $H(w_i) = w_{i-1}$. Если оба равенства выполнены, B идентифицирует A и запоминает (i, w_i) для следующей попытки идентификации.

Заметим, что последний протокол не защищает от активного противника, который перехватывает, сохраняет и блокирует передачу информации от A к B для последующей попытки подмены собой пользователя A . Поэтому этот протокол может быть использован только для идентификации пользователя системой, идентичность которой уже установлена.

§ 12.3. “Запрос-ответ” (сильная идентификация)

Идея построения криптографических *протоколов идентификации типа “запрос-ответ”* состоит в том, что доказывающий убеждает проверяющего в своей аутентичности путем демонстрации своего знания некоторого секрета без предъявления самого секрета. Знание секрета подтверждается выдачей ответов на меняющиеся с течением времени запросы проверяющего. Обычно запрос — это число, выбирае-

мое проверяющим при каждой реализации протокола. Если канал связи контролируется противником, любое допустимое число реализаций протокола идентификации не должно давать противнику возможность извлечения информации, необходимой для последующей ложной идентификации. В таких протоколах обычно используются либо случайные числа, либо числа из неповторяющихся (обычно возрастающих) последовательностей, либо метки времени. Остановимся подробно на последних двух вариантах.

Числа из неповторяющихся последовательностей используются как *уникальные метки* сообщений, обеспечивающие защиту от навязывания ранее переданных сообщений. Такие последовательности чисел используются независимо для каждой пары доказывающего и проверяющего. Кроме того, при передаче информации от A к B и от B к A также используются различные последовательности чисел.

Стороны следуют заранее определенной политике по выработке таких последовательностей. Сообщение принимается только тогда, когда его число (метка сообщения) не было использовано ранее (или в определенный предшествующий период времени) и удовлетворяет согласованной политике. Простейшей политикой является такая: последовательность начинается с нуля и каждое следующее число увеличивается на единицу. Менее жесткая политика состоит в том, что принятые числа должны только монотонно возрастать. Это позволяет работать с учетом возможности потери сообщений из-за ошибок в каналах связи.

Недостатком метода использования последовательностей чисел является необходимость запоминания информации, касающейся каждого доказывающего и каждого проверяющего, а также невозможность обнаружения сообщений, специально задержанных противником.

Метки времени используются для обеспечения гарантий своевременности и единственности сообщений, а также для

обнаружения попыток навязывания ранее переданной информации. Они также могут быть использованы для обнаружения попыток задержки информации со стороны противника.

Протоколы, использующие метки времени, реализуются следующим образом. Сторона, направляющая сообщение, снимает показания своих системных часов и криптографически “привязывает” их к сообщению. Получив такое сообщение, вторая сторона снимает показания своих системных часов и сравнивает их с показаниями, содержащимися в сообщении. Сообщение принимается, если его временная метка находится в пределах приемлемого временного окна — фиксированного временного интервала, выбранного из расчета времени, необходимого для обработки и передачи максимально длинного сообщения и максимально возможной рассинхронизации часов отправителя и получателя. В отдельных случаях для приема сообщения необходимо, кроме указанного выше, обеспечить выполнение условия, чтобы та же самая (или более ранняя) метка времени не приходила ранее от того же самого абонента.

Надежность методов, основанных на метке времени, зависит от надежности и точности синхронизации системных часов, что является главной проблемой в таких системах. Преимуществами указанных систем является меньшее число передаваемых для идентификации сообщений (как правило, одно), а также отсутствие требований по сохранению информации для каждой пары участников (как в числовых последовательностях). Временные метки в протоколах могут быть заменены запросом, включающим случайное число, и ответом.

“Запрос-ответ” с использованием симметричных алгоритмов шифрования

Механизм реализации идентификации с помощью алгоритмов “запрос-ответ” требует, чтобы доказывающий и про-

веряющий имели общий секретный ключ. В небольших системах такими ключами может быть обеспечена каждая пара корреспондентов заблаговременно. В больших системах установление общего ключа может быть обеспечено путем передачи его по защищенному каналу обоим корреспондентам из доверенного центра.

Для описания алгоритмов введем следующие обозначения:

z_A — случайное число, вырабатываемое A ;

t_A — временная метка A ;

E_k — алгоритм шифрования на ключе k ;

$id(B)$ — идентификатор B .

1. Односторонняя идентификация с использованием временной метки.

Доказывающий A передает проверяющему B свою временную метку и идентификатор, зашифрованные на общем ключе:

$$A \rightarrow B : E_k(t_A, id(B)).$$

Проверяющий B , расшифровав данное сообщение, проверяет соответствие допустимому интервалу временной метки и совпадение полученного и собственного идентификаторов. Последнее необходимо для того, чтобы не дать противнику возможности немедленно переадресовать сообщение к A .

2. Односторонняя идентификация с использованием случайных чисел.

Временные метки могут быть заменены случайными числами с помощью дополнительной пересылки:

$$(1) \quad A \leftarrow B : z_B,$$

$$(2) \quad A \rightarrow B : E_k(z_B, id(B)).$$

В данном протоколе проверяющий B , расшифровав сообщение, проверяет, соответствует ли полученное число случайному числу, переданному им на шаге (1). После этого он про-

веряет, соответствует ли его идентификатор полученному идентификатору.

Для предотвращения криптоанализа E_k с помощью специально подобранных открытых текстов A может ввести в (2) свое случайное число так же, как в следующем протоколе.

3. Взаимная идентификация с использованием случайных чисел.

Эта процедура описывается с помощью следующего протокола:

- (1) $A \leftarrow B : z_B,$
- (2) $A \rightarrow B : E_k(z_A, z_B, id(B)),$
- (3) $A \leftarrow B : E_k(z_A, z_B).$

Получив сообщение на шаге (2), проверяющий B расшифровывает его и осуществляет те же проверки, что и выше. После этого он использует z_A на шаге (3). Доказывающий A , расшифровав (3), проверяет соответствие полученных случайных чисел тем, которые использовались ранее в (1) и (2).

“Запрос-ответ” с использованием асимметричных алгоритмов шифрования

В протоколе идентификации, построенном на основе шифрсистемы с открытым ключом, доказывающий может продемонстрировать владение секретным ключом одним из двух способов:

- при расшифровании запроса, зашифрованного на его открытом ключе;
- при проставлении под запросом своей цифровой подписи (см. гл. 15).

Рассмотрим эти два способа более подробно.

Пусть h — некоторая однонаправленная функция и E_A и D_A — алгоритмы зашифрования и расшифрования абонента A . Первый способ основан на следующем протоколе:

$$(1) \quad A \leftarrow B : h(z), id(B), E_A(z, id(B)),$$

$$(2) \quad A \rightarrow B : z.$$

Проверяющий B выбирает случайное число z , вычисляет $h(z)$ и запрос $c = E_A(z, id(B))$. Доказывающий A расшифровывает c и проверяет совпадение значений хэш-функции и идентификаторов. Если получено различие, то A прекращает протокол. В противном случае A посылает z проверяющему B . B идентифицирует A , если полученное от A число z совпадает с имеющимся у него числом.

Использование однонаправленной хэш-функции предотвращает попытки криптоанализа с помощью выбранного открытого текста.

Рассмотрим теперь протокол, использующий цифровую подпись. Пусть z_A, t_A обозначают соответственно случайное число и метку времени доказывающего A , S_A обозначает алгоритм цифровой подписи A . Будем считать, что алгоритм проверки цифровой подписи доказывающего известен проверяющему.

Для идентификации могут быть использованы следующие три протокола.

1. Односторонняя идентификация с использованием временных меток:

$$(1) \quad A \rightarrow B : t_A, id(B), S_A(t_A, id(B)).$$

Получив сообщение, пользователь B проверяет, что временная метка находится в допустимом интервале, $id(B)$ совпадает с его собственным идентификатором, а также то, что цифровая подпись под этими двумя полями верна.

2. Односторонняя идентификация с использованием случайных чисел:

$$(1) \quad A \leftarrow B : z_B,$$

$$(2) \quad A \rightarrow B : z_A, id(B), S_A(z_A, z_B, id(B)).$$

Получив сообщение, пользователь B проверяет, что $id(B)$ соответствует его идентификатору и что цифровая подпись под строкой $(z_A, z_B, id(B))$ верна.

3. Взаимная идентификация с использованием случайных чисел:

$$(1) \quad A \leftarrow B : z_B,$$

$$(2) \quad A \rightarrow B : z_A, id(B), S_A(z_A, z_B, id(B)),$$

$$(3) \quad A \leftarrow B : id(A), S_B(z_B, z_A, id(A)).$$

§ 12.4. Протоколы с нулевым разглашением

Недостатком протоколов с фиксированным паролем является то, что доказывающий A передает проверяющему B свой пароль, вследствие чего B может в последующем выдать себя за A . Протоколы типа “запрос-ответ” ликвидируют этот недостаток. При их выполнении A отвечает на запросы B , меняющиеся во времени, не давая B информации, которую тот может использовать, чтобы выступить от имени A . Тем не менее A может выдать некоторую частичную информацию о своем секрете.

Протоколы с нулевым разглашением призваны решить эту проблему, давая возможность доказывающему продемонстрировать знание секрета, не выдавая о нем никакой информации. Точнее говоря, выдается только один бит информации, обозначающий то, что доказывающий знает секрет.

В протоколах с нулевым разглашением термин “доказательство” имеет смысл, отличный от традиционно принятого в математике. Доказательство имеет вероятностный характер. Это означает, что утверждение имеет место с некоторой вероятностью, которая может быть выбрана сколь угодно близкой к единице.

Примером такого протокола является *протокол Фиата — Шамира*. Он основывается на сложности задачи извлечения квадратного корня по модулю большого составного числа n с неиз-

вестным разложением на множители. Как известно, эта задача эквивалентна задаче разложения числа n на множители.

A доказывает B знание секрета s с помощью t итераций следующего трехшагового протокола.

Доверенный центр T выбирает модуль $n = pq$ и сообщает его всем доказывающим. При этом числа p и q остаются секретными.

Каждый доказывающий A выбирает секрет s , которым является число, взаимно простое с n , $1 \leq s \leq n - 1$, вычисляет значение $v = s^2 \bmod n$ и объявляет v своим открытым ключом.

Следующие три шага производятся независимо t раз, причем B принимает доказательство владения A секретом s , если все эти итерации приводят к положительному ответу.

1. A выбирает случайно z , $1 < z < n - 1$, и посылает B число $x = z^2 \bmod n$.

2. B случайно выбирает бит c и направляет его A .

3. A вычисляет и направляет B число y , равное либо z , если $c = 0$, либо $zs \bmod n$, если $c = 1$.

B дает положительный ответ, если $y \neq 0$ и

$$y^2 \equiv xv^c \pmod{n}.$$

Заметим, что в зависимости от значения бита c выполнено одно из двух условий: $y^2 \equiv x \pmod{n}$ или $y^2 \equiv xv \pmod{n}$, так как $v \equiv s^2 \pmod{n}$. Проверка равенства $y = 0$ исключает случай $z = 0$.

Корректность протокола может быть обоснована следующими рассуждениями.

Наличие запроса c требует, чтобы A был в состоянии ответить на любой из двух вопросов, ответ на один из которых требует знания секрета s , а ответ на другой предотвращает попытку обмана. Противник, выдающий себя за A , может попытаться обмануть проверяющего, выбрав любое число z и передав B число $x = z^2/v$.

Тогда он сможет ответить на запрос “ $c = 1$ ”, направив правильный ответ “ $y = z$ ”, но не сможет ответить на запрос “ $c = 0$ ”, ответ на который требует знания корня квадратного из числа x по модулю n .

Доказывающий A , знаящий s , в состоянии ответить на оба вопроса. Если же он не знает s , то в лучшем случае — на один из двух вопросов. Таким образом, обман удастся с вероятностью, не превышающей $1/2$. При t -кратной итерации протокола вероятность обмана может быть доведена до величины, не превышающей 2^{-t} .

Ответ “ $y = z$ ” не зависит от секрета s доказывающего A , а ответ “ $y = zs \bmod n$ ” также не несет информации о s , так как случайное z не известно проверяющему B .

Идеи, лежащие в основе протоколов с нулевым разглашением, могут быть сформулированы следующим образом.

Доказывающий A выбирает случайный элемент из заранее оговоренного множества, как свой секретный ключ для данной итерации протокола, вычисляет, используя его как аргумент некоторой однонаправленной функции, ее значение, и предъявляет это значение проверяющему. Этим обеспечивается случайность и независимость различных итераций протокола и определяется набор вопросов, на каждый из которых доказывающий готов дать ответ. Протокол построен так, что только доказывающий A , владеющий секретом s , в состоянии ответить на все эти вопросы, и ни один ответ не дает информации о секрете. На следующем этапе B выбирает один из этих вопросов и A дает на него ответ, который затем проверяется B . Осуществляется необходимое число итераций протокола с целью снизить до приемлемого уровня вероятность обмана.

Другими словами, в основе протоколов с нулевым разглашением лежит комбинация идей протоколов типа “режь и выбирай” (этот термин происходит от стандартного метода, которым дети делят кусок пирога: один режет, а другой выбирает) и протоколов типа “запрос-ответ”.

§ 12.5. Атаки на протоколы идентификации

В заключение приведем перечень атак на протоколы идентификации и методов их отражения, часть из которых уже упоминалась выше.

1. *Подмена* — попытка подменить одного пользователя другим.

Методы противодействия состоят в сохранении в тайне от противника информации, определяющей алгоритм идентификации.

2. *Повторное навязывание сообщения (replay)* — подмена или другой метод обмана, использующий информацию ранее проведенного протокола идентификации того же самого или другого пользователя.

Методы противодействия включают использование протоколов типа “запрос-ответ”, использование временных меток, случайных чисел или возрастающих последовательностей чисел.

3. *Комбинированная атака (interleaving attack)* — подмена или другой метод обмана, использующий комбинацию данных из ранее выполненных протоколов, в том числе протоколов, ранее навязанных противником.

Метод противодействия состоит в обеспечении целостности проводимых протоколов и отдельных сообщений.

4. *Атака отражением* — комбинированная атака, использующая посылку части информации только что проведенного протокола доказывающему.

Методы противодействия включают введение в протокол идентификационной информации проверяющего, использование различных ключей для приема и передачи сообщений.

5. *Задержка передачи сообщения (forced delay)* — перехват противником сообщения и навязывание его в более поздний момент времени.

Методы противодействия включают использование случайных чисел совместно с ограничением временного промежутка для ответа, использование временных меток.

6. *Атака с использованием специально подобранных текстов* — атака на протоколы типа “запрос-ответ”, при которой противник по определенному правилу выбирает запросы с целью получить информацию о долговременном ключе доказывающего. Эта атака может включать специально подобранные открытые тексты, если доказывающий должен подписать или зашифровать запрос, и специально подобранные шифрованные тексты, если доказывающий должен расшифровать запрос.

Методы противодействия этой атаке состоят во включении случайных чисел в запросы или ответы, а также в использовании протоколов с нулевым разглашением.

7. *Использование противником своих средств в качестве части телекоммуникационной структуры* — атака, при которой в протоколе идентификации между A и B противник C входит в телекоммуникационный канал и становится его частью при реализации протокола между A и B . При этом противник может подменить информацию, передаваемую между A и B . Эта атака особенно опасна в случае установления A и B общего ключа по протоколу Диффи — Хеллмана.

Противодействие этой атаке состоит в использовании защищенного канала для установления общего ключа между A и B .

В заключение заметим следующее. Идентификация может быть гарантирована только в момент времени после завершения протокола. При этом имеется опасность того, что противник подключится к линии связи после окончания процесса идентификации, выдавая себя за законного пользователя. Для исключения этой возможности следует совместить процесс идентификации с процессом установления общего сеансового ключа, который должен быть использован для защиты передаваемой информации до следующей реализации протокола идентификации.

Контрольные вопросы

1. На какие группы могут быть разбиты алгоритмы идентификации?
2. В чем состоят недостатки систем с фиксированными паролями?
3. За счет чего повышается надежность идентификации при использовании пластиковой карты и личного идентификационного номера?
4. Каковы возможные схемы использования одноразовых паролей?
5. Для каких целей используется временная метка в протоколе типа “запрос-ответ”?
6. Чем могут быть заменены временные метки в протоколах типа “запрос-ответ”?
7. Какая идея лежит в основе протоколов с нулевым разглашением?
8. Какие типы атак могут быть использованы при нападении на протоколы идентификации?

Криптографические хэш-функции

§ 13.1. Функции хэширования и целостность данных

Хэш-функции — это функции, предназначенные для “сжатия” произвольного сообщения или набора данных, записанного, как правило, в двоичном алфавите, в некоторую битовую комбинацию фиксированной длины, называемую *сверткой*. Хэш-функции имеют разнообразные применения при проведении статистических экспериментов, при тестировании логических устройств, при построении алгоритмов быстрого поиска и проверки целостности записей в базах данных (см., например, [Кну99]). Например, для осуществления быстрого поиска нужного сообщения в большом списке сообщений различной длины удобнее сравнивать друг с другом не сами сообщения, а короткие значения их сверток, играющих одновременно роль контрольных сумм. Основным требованием к таким хэш-функциям является равномерность распределения их значений при случайном выборе значений аргументов.

В криптографии хэш-функции применяются для решения следующих задач:

- построения систем контроля целостности данных при их передаче или хранении,
- аутентификации источника данных.

При решении первой задачи для каждого набора данных вычисляется значение хэш-функции (называемое *кодом аутентификации сообщения* или *имитовставкой*), которое передается или хранится вместе с самими данными. При получении данных пользователь вычисляет значение свертки и

сравнивает его с имеющимся контрольным значением. Несовпадение говорит о том, что данные были изменены.

Хэш-функция, служащая для выработки имитовставки, должна позволять (в отличие от обычной контрольной суммы) осуществлять обнаружение не только случайных ошибок в наборах данных, возникающих при их хранении и передаче, но и сигнализировать об активных атаках злоумышленника, пытающегося осуществить навязывание ложной информации. Для того чтобы злоумышленник не смог самостоятельно вычислить контрольное значение свертки и тем самым осуществить успешную имитацию или подмену данных, хэш-функция должна зависеть от секретного, не известного злоумышленнику, параметра — ключа пользователя. Этот ключ должен быть известен передающей и проверяющей сторонам. Такие хэш-функции будем называть *ключевыми*.

Имитовставки, формируемые с помощью ключевых хэш-функций, не должны позволять противнику создавать поддельные (сфабрикованные) сообщения (*fabrication*) при атаках типа *имитация* (*impersonation*) и модифицировать передаваемые сообщения (*modification*) при атаках типа “подмена” (*substitution*).

При решении второй задачи — аутентификации источника данных — мы имеем дело с не доверяющими друг другу сторонами. В связи с этим подход, при котором обе стороны обладают одним и тем же секретным ключом, уже неприменим. В такой ситуации применяют схемы цифровой подписи, позволяющие осуществлять аутентификацию источника данных. Как правило, при этом сообщение, прежде чем быть подписано личной подписью, основанной на секретном ключе пользователя, “сжимается” с помощью хэш-функции, выполняющей функцию кода обнаружения ошибок (см. далее). В данном случае хэш-функция не зависит от секретного ключа и может быть фиксирована и известна всем. Основными требованиями к ней являются гарантии невозможности подмены подписанного документа, а также подбора двух различных

сообщений с одинаковым значением хэш-функции (в этом случае говорят, что такая пара сообщений образует *коллизю*).

Формализуя сказанное, введем следующее определение. Обозначим через X множество, элементы которого будем называть сообщениями. Обычно сообщения представляют собой последовательности символов некоторого алфавита, как правило, двоичного. Пусть Y — множество двоичных векторов фиксированной длины.

Хэш-функцией называется всякая функция $h: X \rightarrow Y$, легко вычисляемая и такая, что для любого сообщения M значение $h(M) = H$ (*свертка*) имеет фиксированную битовую длину.

Обычно число возможных сообщений значительно превосходит число возможных значений свертки, в силу чего для каждого значения свертки имеется большое множество прообразов, то есть сообщений с заданным значением хэш-функции. Заметим, что при случайном и равновероятном выборе сообщений условие равномерности распределения значений хэш-функции эквивалентно наличию одинакового числа прообразов для каждого значения свертки.

Как правило, хэш-функции строят на основе так называемых *одношаговых сжимающих функций* $y = f(x_1, x_2)$ двух переменных, где x_1 и x_2 — двоичные векторы длины m и n соответственно, причем n — длина свертки. Для получения значения $h(M)$ сообщение M сначала разбивается на блоки длины m (при этом если длина сообщения не кратна m , то последний блок неким специальным образом дополняется до полного), а затем к полученным блокам M_1, M_2, \dots, M_N применяют следующую последовательную процедуру вычисления свертки:

$$\begin{aligned} H_0 &= v, \\ H_i &= f(M_i, H_{i-1}), \quad i = 1, \dots, N, \\ h(M) &= H_N. \end{aligned} \quad (1)$$

Здесь v — некоторый фиксированный начальный вектор. Если функция f зависит от ключа, то этот вектор можно положить равным нулевому вектору. Если же функция f не зависит от ключа, то для исключения возможности перебора коротких сообщений (при попытках обращения хэш-функции) этот вектор можно составить из фрагментов, указывающих дату, время, номер сообщения и т. п.

При таком подходе свойства хэш-функции h полностью определяются свойствами одношаговой сжимающей функции f .

Особо выделяют два важных типа криптографических хэш-функций — *ключевые* и *бесключевые*. Первые применяются в системах с симметричными ключами. Ключевые хэш-функции называют *кодами аутентификации сообщений (КАС)* (*message authentication code (MAC)*). Они дают возможность без дополнительных средств гарантировать как правильность источника данных, так и целостность данных в системах с доверяющими друг другу пользователями.

Бесключевые хэш-функции называются *кодами обнаружения ошибок* (*modification detection code (MDC)* или *manipulation detection code, message integrity code (MIC)*). Они дают возможность с помощью дополнительных средств (например, шифрования, использования защищенного канала или цифровой подписи) гарантировать целостность данных. Эти хэш-функции могут применяться в системах как с доверяющими, так и не доверяющими друг другу пользователями. Рассмотрим их более подробно.

§ 13.2. Ключевые функции хэширования

В криптографических приложениях к ключевым функциям хэширования предъявляются следующие основные требования:

- невозможность фабрикаций;
- невозможность модификации.

Первое требование означает высокую сложность подбора сообщения с правильным значением свертки. Второе — высокую сложность подбора для заданного сообщения с известным значением свертки другого сообщения с правильным значением свертки.

Иногда эти свойства объединяют в одно более сильное свойство — свойство *вычислительной устойчивости*. Это требование означает высокую сложность подбора для заданного множества сообщений $\{x_1, \dots, x_t\}$ (быть может, пустого) с известными значениями сверток еще одного сообщения x , $x \neq x_i$, $i = 1, \dots, t$, с правильным значением свертки (возможен случай $h(x) = h(x_i)$, $i \in \{1, \dots, t\}$).

Заметим, что здесь и всюду ниже слова “высокая сложность” означают такую вычислительную сложность задачи, при которой ее решение с использованием вычислительной техники за реальное время невозможно.

Ключевые функции применяются в ситуациях, когда стороны доверяют друг другу и могут иметь общий секретный ключ. Обычно в этих условиях не требуется, чтобы система обеспечивала защиту в случае отказа получателя от факта получения сообщения или его подмены. Поэтому от ключевых хэш-функций не требуется устойчивости к коллизиям.

Обычные атаки на ключевые хэш-функции заключаются в имитации, то есть в передаче сфабрикованных сообщений в пустом канале, а также в подмене передаваемых сообщений с целью навязывания приемной стороне ложных сообщений.

Заметим, что из свойства вычислительной устойчивости вытекает невозможность определения ключа, используемого хэш-функцией, так как знание ключа дает возможность вычислять значение свертки для любого набора данных. В то же время обратное утверждение неверно, так как подбор значения функции хэширования возможен в некоторых случаях без предварительного определения ключа.

В качестве примера рассмотрим широко распространенную хэш-функцию, построенную на основе одношаговой сжимающей функции вида

$$f_k(x, H) = E_k(x \oplus H),$$

где E_k — алгоритм блочного шифрования.

Для вычисления значения $h(M)$ сообщение M представляется в виде последовательности n -битовых блоков M_1, M_2, \dots, M_N . Если при этом длина сообщения не кратна длине блока, то последний блок неким специальным образом дополняется до полного блока. Алгоритм вычисления свертки имеет следующий вид:

$$\begin{aligned} H_0 &= 0, \\ H_i &= E_k(M_i \oplus H_{i-1}), \quad i = 1, \dots, N, \\ h(M) &= H_N. \end{aligned} \quad (2)$$

Данный алгоритм фактически совпадает с режимом шифрования со сцеплением блоков (см. гл. 8), с той лишь разницей, что в качестве результата берется не весь шифртекст H_1, H_2, \dots, H_N , а только его последний блок. Такой режим в ГОСТе 28147-89 называется *режимом выработки имитовставки*.

Еще одной основой для построения ключевых хэш-функций могут служить бесключевые хэш-функции. При этом для вычисления значения свертки ключ приписывается к исходному сообщению.

Заметим, что если ключ просто дописывать в начало или в конец исходного сообщения, то это может приводить к потенциальным слабостям, позволяющим в некоторых случаях осуществлять модификацию сообщений.

Пусть, например, ключ k добавляется к началу сообщения согласно формуле $h_k(x) = h(k, x)$. Если функция h построена на основе одношаговой сжимающей функции по формуле (1), то по известным значениям M и $H = h(k, M)$ можно вычислять значения этой функции для любых сообщений вида (M, M') с дописанным произвольным окончанием M' . Это объясняется итеративностью процедуры вычисления функции, в силу которой для нахождения значения $H' = h(k, M, M')$ не требуется знание ключа k , достаточно воспользоваться уже вычисленным “промежуточным” значением H . Поэтому такая функция не устойчива к модификации.

В случае когда ключ добавляется в конец сообщения согласно формуле $H = h_k(M) = h(M, k)$, знание коллизии для функции h , то есть пары x_1, x_2 , $x_1 \neq x_2$, такой, что $h(x_1) = h(x_2)$, позволяет вычислять значения $h(x_1, k) = h(x_2, k)$ для любого ключа k . Поэтому трудоемкость модификации сообщения $M = x_1$ оценивается не величиной $O(2^n)$, а сравнима с трудоемкостью поиска коллизии, оцениваемой величиной $O(2^{n/2})$, так как в данном случае применима атака, основанная на парадоксе “дней рождений”.

В связи с этим более предпочтительными являются способы введения ключа, при которых ключ вставляется в сообщение не один, а, по крайней мере, два раза. В [Men97] указываются два таких способа:

$$H = h(k, y, M, k),$$

$$H = h(k, y_1, h(k, y_2, M)),$$

где y , y_1 и y_2 — дополнения ключа k до размера, кратного длине блока n . Для определенных бесключевых хэш-функций h такой подход позволяет строить эффективно вычисляемые и устойчивые к атакам ключевые хэш-функции. Недостатком

такого метода является слишком большая длина n свертки. Дело в том, что для целей проверки целостности обычно выбирают длину свертки n в пределах $32 \div 64$, а для аутентификации необходимо условие $n \geq 128$.

В заключение отметим, что существуют ключевые хэш-функции, не использующие какую-либо основу типа блочного шифрования или вычисления бесключевой хэш-функции, а разработанные независимо с учетом эффективной реализации на современных ЭВМ. Примером служит ключевая хэш-функция, используемая в алгоритме МАА (*Message Authenticator Algorithm*), утвержденном стандартом ISO 8731-2.

§ 13.3. Бесключевые функции хэширования

Обычно требуется, чтобы бесключевые хэш-функции обладали следующими свойствами:

- 1) *однонаправленность*,
- 2) *устойчивость к коллизиям*,
- 3) *устойчивость к нахождению второго прообраза*,

означающими соответственно высокую сложность нахождения сообщения с заданным значением свертки; пары сообщений с одинаковыми значениями свертки; второго сообщения с тем же значением свертки для заданного сообщения с известным значением свертки.

Например, хэш-функция CRC-32, представляющая собой контрольную сумму, является линейным отображением и поэтому не удовлетворяет ни одному из этих трех свойств.

Использование в качестве бесключевой хэш-функции рассмотренной выше функции (2), построенной на основе алгоритма блочного шифрования в режиме выработки имитовставки, также нецелесообразно, так как обратимость блочного шифрования позволяет подбирать входное сообщение для любого значения свертки при фиксированном и общеизвестном ключе.

Для построения примера хэш-функции, удовлетворяющей свойству 1), рассмотрим функцию, заданную формулой

$g_k(x) = E_k(x) \oplus x$, где E_k — алгоритм блочного шифрования. Такая функция является однонаправленной по обоим аргументам. Поэтому на ее основе можно построить хэш-функцию по правилу (1), определив одношаговую сжимающую функцию одной из следующих формул:

$$f(x, H) = E_H(x) \oplus x$$

или

$$f(x, H) = E_x(H) \oplus H.$$

Первая из этих функций лежит в основе российского стандарта хэш-функции, а вторая — в основе американского стандарта SHA.

Справедливо следующее

Утверждение 1. *Если функция хэширования h построена на основе одношаговой сжимающей функции f по правилу (1), то из устойчивости к коллизиям функции f следует устойчивость к коллизиям функции h .*

Действительно, если у функции h имеется коллизия, то на некотором шаге i должна существовать коллизия у функции f (при определении коллизий функцию $f(x_1, x_2)$ следует рассматривать как функцию от одного переменного, полученного конкатенацией переменных x_1, x_2 в один входной вектор).

Укажем взаимозависимость между свойствами 1) и 2).

Утверждение 2. *Если хэш-функция устойчива к коллизиям, то она устойчива к нахождению второго прообраза.*

Действительно, если для заданной пары сообщение-свертка можно подобрать второй прообраз, то полученная пара сообщений будет составлять коллизию.

Утверждение 3. *Устойчивая к коллизиям хэш-функция не обязана быть однонаправленной.*

В качестве примера несжимающей функции приведем функцию $h(x) = x$, которая, очевидно, является устойчивой

к коллизиям и к нахождению второго прообраза, но не является однонаправленной.

В качестве примера сжимающей хэш-функции рассмотрим функцию h , определенную условиями

$$\begin{aligned} h(x) &= (1, x), \text{ если битовая длина } x \text{ равна } n, \\ h(x) &= (0, g(x)), \text{ если битовая длина } x \text{ больше } n, \end{aligned}$$

где $g(x)$ — сжимающая n -битовая функция, устойчивая к коллизиям. Функция h также является устойчивой к коллизиям и к нахождению второго прообраза, но, очевидно, не является однонаправленной.

Утверждение 4. Пусть $h: X \rightarrow Y$ — хэш-функция и $|X| > 2|Y|$. Тогда если существует эффективный алгоритм обращения функции h , то существует вероятностный алгоритм нахождения коллизии функции h с вероятностью успеха, большей $1/2$.

Доказательство. Будем случайно и равновероятно выбирать сообщение x , вычислять $y = h(x)$, $x' = h^{-1}(y)$ и сравнивать x с x' . Покажем, что данный алгоритм имеет вероятность успеха $p > 1/2$. Под успехом мы понимаем построение x' , отличного от x .

Пусть $X = X_1 \cup \dots \cup X_m$ — разбиение X на классы, состоящие из сообщений с одинаковыми значениями хэш-функции. Ясно, что $m \leq |Y|$. Легко заметить, что выполняются следующие соотношения:

$$\begin{aligned} p &= \frac{1}{|X|} \sum_{i=1}^m \sum_{x \in X_i} \frac{|X_i| - 1}{|X_i|} = \frac{1}{|X|} \sum_{i=1}^m (|X_i| - 1) = \\ &= \frac{|X| - m}{|X|} \geq \frac{|X| - |Y|}{|X|} > \frac{1}{2}. \end{aligned}$$

Утверждение доказано.

Заметим, что трудоемкость подбора прообраза для однонаправленной функции или трудоемкость поиска второго прообраза оцениваются величиной $O(2^n)$. В то же время трудоемкость поиска коллизии оценивается величиной $O(2^{n/2})$, так как в данной ситуации применима атака, основанная на парадоксе “дней рождений”.

Рассмотрим конкретные примеры хэш-функций, построенных на основе некоторых алгоритмов преобразования блоков.

Пусть E_k — алгоритм блочного шифрования, n — размер блока, l — размер ключа и G — некоторое отображение, ставящее в соответствие вектору длины n вектор длины l . Рассмотрим следующие одношаговые сжимающие функции, построенные на основе алгоритма E_k :

а) $f(x, H) = E_x(H) \oplus H$ (Дэвис — Мейер);

б) $f(x, H) = E_{G(H)}(x) \oplus x$ (Матиас — Мейер — Осеас);

в) $f(x, H) = E_{G(H)}(x) \oplus x \oplus H$ (Миагучи — Принель).

Значением любой из хэш-функций, построенных по правилу (1) из приведенных одношаговых сжимающих функций, является вектор длины n , равной размеру блока. В случае если эта величина оказывается недостаточной, ее можно увеличить, заменив одношаговую функцию f на функцию f' с удвоенной размерностью значений. Это можно сделать, например, путем двукратного применения функции f с последующим перемешиванием полублоков согласно формуле:

$$f'(x, H_1, H_2) = \pi(f(x, H_1), f(x, H_2)),$$

в которой π переставляет произвольные полублоки a, b, c, d по правилу $\pi((a, b), (c, d)) = (a, d, c, b)$. Такой подход, ис-

пользующий схему (б), реализован в конструкции одношаговой функции MDC-2.

Другие примеры бесключевых хэш-функций дают известные алгоритмы MD-4, MD-5 и SHA. Они оперируют с блоками длины n , совпадающей с длиной результирующего значения свертки, причем $n = 128$ для алгоритма MD-4 и $n = 160$ для MD-5 и SHA. Указанные алгоритмы спроектированы специально с учетом эффективной реализации на 32-разрядных ЭВМ.

При их использовании исходное сообщение M разбивается на блоки длиной $m = 512$ бит. Последний блок формируется путем дописывания к концу сообщения комбинации $10\dots 0$ до получения блока размера 448 бит, к которому затем добавляется комбинация из 64 бит, представляющая битовую длину сообщения. Затем вычисляется значение свертки согласно процедуре (1) с использованием одношаговой сжимающей функции, заданной формулой $f(x, H) = E_x(H) \oplus H$, где x — блок сообщения длины $m = 512$ бит, H — блок из n бит, а E_x — некоторое преобразование множества блоков. Значение начального вектора определяется в описании преобразования E_x .

В стандарте хэш-функции ГОСТ Р 34.11-94 приняты значения $n = m = 512$. Одношаговая сжимающая функция $f(x, H)$, используемая для вычисления последовательности значений $H_i = f(x_i, H_{i-1})$, построена на базе четырех параллельно работающих схем блочного шифрования (ГОСТ 28147-89), каждая из которых имеет 256-битовый ключ и оперирует с блоками размера 64 бита. Каждый из ключей вычисляется в соответствии с некоторой линейной функцией от блока исходного сообщения x_i и значения H_{i-1} . Значение H_i является линейной функцией от результата шифрования, блока исходного сообщения x_i и значения H_{i-1} . После вычисления значения H_N для последовательности блоков M_1, M_2, \dots, M_N применяют еще два шага вычисления согласно формуле

$$H = h(M) = f(Z \oplus M_N, f(L, H_N)),$$

где Z — сумма по модулю два всех блоков сообщения, а L — длина сообщения.

§ 13.4. Целостность данных и аутентификация сообщений

Термин “аутентификация” означает установление подлинности. Он может относиться ко всем аспектам взаимодействия: сеансу связи, сторонам, передаваемым сообщениям и т. д. Применительно к самой информации аутентификация означает проверку того, что данные, передаваемые по каналу связи, являются подлинными по своему источнику и содержанию, по времени создания, времени пересылки и т. д.

Рассмотрим эти свойства более подробно.

Целостность данных — свойство, позволяющее убедиться в том, что данные не изменялись неавторизованным способом с тех пор, как они были созданы, переданы или сохранены авторизованным источником. Под изменениями обычно понимают пропуски, вставки, замены и перестановки фрагментов сообщения.

Аутентификация источника данных — получение подтверждения того, что рассматриваемый документ был создан именно указанным соответствующим образом источником информации. Подчеркнем, что при этом не требуется проверка времени создания и единственности документа, важно только то, что он был создан в некоторый (обычно неопределенный) момент времени в прошлом. Нарушение “единственности документа” подразумевает его повторную передачу или повторное использование. Если источник сообщений фиксирован, то вместо термина “аутентификация источника данных” используют термин “аутентификация сообщений”.

Целостность данных и аутентификация источника данных тесно связаны друг с другом. Действительно, если данные подверглись модификации, то у них автоматически изменился источник. Если же не установлен источник, то без ссылки на него нельзя разрешить проблему целостности. В связи с этим

будем считать по определению, что аутентификация источника данных включает проверку целостности данных.

Рассмотрим примеры.

1. Для обеспечения целостности хранимых данных (M) можно добавить к ним значение криптографической хэш-функции ($M, h_k(M)$), зависящей от ключа, известного только владельцу информации. Такой способ автоматически решает проблему аутентификации источника данных.
2. Для обеспечения целостности передаваемого сообщения можно так же, как и в п. 1, использовать хэш-функцию, зависящую от секретного ключа, известного отправителю и получателю. Аутентификация источника гарантируется тем, что секретный ключ известен только двум сторонам. При этом, однако, исключается возможность разрешения споров, связанных с отказом от авторства. Поэтому аутентификация источника с помощью ключевой хэш-функции возможна только при взаимодействии доверяющих друг другу сторон.
3. При аутентификации источника можно использовать симметричное шифрование, добавляя предварительно к исходному сообщению некоторый секретный, известный только отправителю и получателю, идентификатор. Такой способ, однако, не гарантирует целостности получаемых данных. Дело в том, что шифрование может допускать возможность модификации данных при передаче. Кроме того, для его реализации требуется секретный канал для обмена подобными идентификаторами.
4. Аутентификация источника возможна также при совместном использовании бесключевой хэш-функции и симметричного шифрования. Для этого достаточно воспользоваться одной из следующих форм передаваемого сообщения: $E_k(M, h(M))$ или $(M, E_k(h(M)))$. В рассматриваемом случае можно повторить те же комментарии, что и в п. 2. К этому следует добавить, что алгоритм шифрова-

ния E_k должен быть стойким к атакам на основе известного открытого текста.

5. На практике чаще, чем предыдущие, применяются схемы, использующие бесключевые хэш-функции в сочетании с асимметричным шифрованием. Подобным образом на основе схемы $(M, E_k(h(M)))$ строятся алгоритмы вычисления цифровой подписи. Подобные схемы позволяют решать проблему аутентификации источника как при взаимодействии доверяющих друг другу, так и не доверяющих друг другу сторон.
6. Возможно совместное использование ключевой хэш-функции и симметричного шифрования в соответствии с одной из следующих схем:

$$\begin{aligned} & E_{k_1}(M, h_{k_2}(M)), \\ & (E_{k_1}(M), h_{k_2}(M)), \\ & (E_{k_1}(M), h_{k_2}(E_{k_1}(M))), \end{aligned}$$

и т. д. При таком подходе не только ключи шифрования (k_1) и хэш-функции (k_2) должны быть независимыми, но и сами алгоритмы шифрования и вычисления значения хэш-функции также должны иметь существенные различия. В противном случае возникают дополнительные соотношения, которые можно использовать для отбраковки ключей.

В заключение рассмотрим вопрос о дополнительном гарантировании единственности и своевременности передачи сообщений. В этом случае используется термин “аутентификация транзакции”, означающий аутентификацию сообщения с подтверждением единственности и своевременности передачи данных. Такой тип аутентификации предоставляет возможность защиты от повторного использования ранее переданных сообщений, что является необходимым в тех случаях, когда подобная угроза может привести к нежелательным по-

следствиям. Примером таких приложений являются электронные банковские платежи или системы автоматизированного управления подвижными объектами.

Для обеспечения единственности и своевременности передачи сообщений обычно используются дополняющие аутентификацию параметры, которые вставляются в передаваемые сообщения. Это могут быть метки времени или некоторые последовательности чисел. Если метки времени позволяют установить время создания или передачи документа, то последовательность чисел гарантирует правильность порядка получения сообщений. Помимо этого для аутентификации последующих сообщений могут использоваться случайные числа, передаваемые в предыдущих сообщениях. Такой способ позволяет организовать “жесткое сцепление” идущих друг за другом сообщений. Подробно этот подход уже был рассмотрен в гл. 12.

§ 13.5. Возможные атаки на функции хэширования

Простейшая атака с целью создания поддельного сообщения, применимая к любой хэш-функции, состоит в следующем. Злоумышленник может осуществить генерацию некоторого числа (r_1) сообщений, вычислить значения их сверток и сравнить получившиеся значения с известными значениями сверток некоторого множества (из r_2) переданных ранее сообщений. Атака окажется успешной при получении хотя бы одного совпадения. Вероятность успеха P можно оценить на основании парадокса “дней рождений”. Известно, что эта вероятность оценивается по формуле

$$P \approx 1 - e^{-\frac{r_1 r_2}{2^n}},$$

где n — длина свертки, e — основание натуральных логарифмов. Наибольшей эта вероятность становится при $r_1 = r_2 = 2^{\frac{n}{2}}$. В этом случае ее значение приблизительно равно 0,63.

Ранее указывалось, что во многих случаях хэш-функции строятся на основе одношаговых сжимающих функций. Поэтому имеется тесная связь атак на хэш-функцию с атаками на соответствующую одношаговую сжимающую функцию. В частности, последняя должна обладать практически всеми теми же свойствами, которыми обладает и сама хэш-функция.

Итеративный способ построения хэш-функций позволяет иногда при ее обращении или построении коллизий использовать метод “встречи посередине”. Для защиты от этой опасности в конце сообщения обычно дописывают блоки с контрольной суммой и длиной сообщения.

Возможны атаки, использующие слабости тех схем, на базе которых построены хэш-функции. Например, для построения коллизий хэш-функций, основанных на алгоритмах блочного шифрования, можно использовать наличие слабых ключей или свойство дополнения (как это имеет место у алгоритма DES), наличие неподвижных точек (для которых $E_k(x) = x$), коллизии ключей (то есть пар различных ключей, для которых выполняется равенство $E_k(x) = E_{k'}(x)$) и т. п.

Контрольные вопросы

1. Для каких целей применяются хэш-функции?
2. Перечислите основные требования, предъявляемые к хэш-функциям.
3. Почему нельзя использовать в качестве хэш-функций линейные отображения?
4. Сравните требования, предъявляемые к ключевым и бесключевым хэш-функциям.

5. Можно ли использовать в качестве бесключевой хэш-функции ключевую хэш-функцию с фиксированным ключом?
6. Можно ли использовать в качестве ключевой хэш-функции функцию вида $h_k(M) = h(h(M), k)$?

Глава 14

Цифровые подписи

§ 14.1. Общие положения

Цифровая подпись для сообщения является числом, зависящим от самого сообщения и от некоторого секретного, известного только подписывающему субъекту, ключа. При этом предполагается, что она должна быть легко проверяемой и что осуществить проверку подписи должен иметь возможность каждый без получения доступа к секретному ключу. При возникновении спорной ситуации, связанной с отказом подписывающего от факта подписи им некоторого сообщения либо с попыткой подделки подписи, третья сторона должна иметь возможность разрешить спор.

Цифровая подпись позволяет решить следующие три задачи:

- осуществить аутентификацию источника сообщения,
- установить целостность сообщения,
- обеспечить невозможность отказа от факта подписи конкретного сообщения.

Использование термина “подпись” в данном контексте оправдано тем, что цифровая подпись имеет много общего с обычной собственноручной подписью на бумажном документе. Собственноручная подпись также решает три перечисленные задачи, однако между обычной и цифровой подписями имеются существенные различия. Сведем основные различия между обычной и цифровой подписями в таблицу.

<i>Собственноручная подпись</i>	<i>Цифровая подпись</i>
Не зависит от подписываемого текста, всегда одинакова	Зависит от подписываемого текста, практически всегда разная
Неразрывно связана с подписывающим лицом, однозначно определяется его психофизическими свойствами, не может быть утеряна	Определяется секретным ключом, принадлежащим подписывающему лицу, может быть утеряна владельцем
Неотделима от носителя (бумаги), поэтому отдельно подписывается каждый экземпляр документа	Легко отделима от документа, поэтому верна для всех его копий
Не требует для реализации дополнительных механизмов	Требует дополнительных механизмов, реализующих алгоритмы ее вычисления и проверки
Не требует создания поддерживающей инфраструктуры	Требует создания доверенной инфраструктуры сертификатов открытых ключей

Для реализации схемы цифровой подписи необходимы два алгоритма:

— алгоритм вычисления цифровой подписи

и

— алгоритм ее проверки.

Главные требования к этим алгоритмам заключаются в исключении возможности получения подписи без использования секретного ключа и гарантировании возможности проверки подписи без знания какой-либо секретной информации.

Надежность схемы цифровой подписи определяется сложностью следующих трех задач:

— *подделки подписи*, то есть нахождения значения подписи под заданным документом лицом, не являющимся владельцем секретного ключа;

— *создания подписанного сообщения*, то есть нахождения хотя бы одного сообщения с правильным значением подписи;

— *подмены сообщения*, то есть подбора двух различных сообщений с одинаковыми значениями подписи.

Имеется множество различных схем цифровой подписи, обеспечивающих тот или иной уровень стойкости. Основные подходы к их построению будут рассмотрены ниже.

Принципиальной сложностью, возникающей при использовании цифровой подписи на практике, является проблема создания *инфраструктуры открытых ключей*. Дело в том, что для алгоритма проверки подписи необходима дополнительная открытая информация, связанная с обеспечением возможности открытой проверки подписи и зависящая от секретного ключа автора подписи. Эту информацию можно называть открытым ключом цифровой подписи. Для исключения возможности подделки этой информации (открытого ключа) лицами, которые хотят выступить от лица законного владельца подписи (секретного ключа), создается инфраструктура, состоящая из центров сертификации открытых ключей и обеспечивающая возможность своевременного подтверждения достоверности принадлежности данной открытой информации заявленному владельцу и обнаружения подлога.

Создание сертификационных центров с технической точки зрения не представляет большой сложности. Они строятся во многом аналогично центрам сертификации, которые используются в криптографических системах с открытыми ключами. Однако с юридической точки зрения здесь имеется множество проблем. Дело в том что в случае возникновения споров, связанных с отказом от авторства или подделки подписи, такие центры должны нести юридическую ответственность за достоверность выдаваемых сертификатов. В частности, они должны возмещать понесенные убытки в случае

конфликтных ситуаций, когда алгоритм проверки подписи подтверждает ее правильность. В связи с этим сложилась практика заключения договоров между участниками информационного взаимодействия с применением цифровых подписей. В таком договоре должно быть четко указано:

— кто должен нести ответственность в случае, если подписанные сделки не состоятся;

— кто должен нести ответственность в случае, если система окажется ненадежной и будет взломана, то есть будет выявлен факт подделки секретного ключа;

— какова ответственность уполномоченного по сертификатам в случае, если открытый ключ будет сфальсифицирован;

— какова ответственность владельца секретного ключа в случае его утраты;

— кто несет ответственность за плохую реализацию системы в случае повреждения или разглашения секретного ключа;

— каков порядок разрешения споров и т. п.

Поскольку данные проблемы носят юридический, а не технический характер, то для их разрешения нужен юридически правильно заключенный договор, оформленный стандартным образом на бумаге.

В настоящее время предложено несколько принципиально различных подходов к созданию схем цифровой подписи. Их можно разделить на три группы:

1) схемы на основе систем шифрования с открытыми ключами;

2) схемы со специально разработанными алгоритмами вычисления и проверки подписи;

3) схемы на основе симметричных систем шифрования.

Рассмотрим их более подробно.

§ 14.2. Цифровые подписи на основе шифрсистем с открытыми ключами

Идея использования систем шифрования с открытыми ключами для построения систем цифровой подписи как бы заложена в постановке задачи. Действительно, пусть имеется пара преобразований (E, D) , первое из которых зависит от открытого ключа, а второе — от секретного. Для того чтобы вычислить цифровую подпись S для сообщения, владелец секретного ключа может применить к сообщению M второе преобразование D : $S = D(M)$. В таком случае вычислить подпись может только владелец секретного ключа, в то время как проверить равенство $E(S) = M$ может каждый. Основными требованиями к преобразованиям E и D являются:

— выполнение равенства $M = E(D(M))$ для всех сообщений M ;

— невозможность вычисления значения $D(M)$ для заданного сообщения M без знания секретного ключа.

Отличительной особенностью предложенного способа построения цифровой подписи является возможность отказаться от передачи самого подписываемого сообщения M , так как его можно восстановить по значению подписи. В связи с этим подобные системы называют *схемами цифровой подписи с восстановлением текста*.

Заметим, что если при передаче сообщение дополнительно шифруется с помощью асимметричного шифра, то пара преобразований (E, D) , используемая в схеме цифровой подписи, должна отличаться от той, которая используется для шифрования сообщений. В противном случае появляется возможность передачи в качестве шифрованных ранее подписанных сообщений. При этом более целесообразно шифровать подписанные данные, чем делать наоборот, то есть подписывать шифрованные данные, поскольку в первом случае про-

тивник получит только шифртекст, а во втором — и открытый, и шифрованный тексты.

Очевидно, что рассмотренная схема цифровой подписи на основе пары преобразований (E, D) удовлетворяет требованию невозможности подделки, в то время как требование невозможности создания подписанного сообщения не выполняется: для любого значения S каждый может вычислить значение $M = E(S)$ и тем самым получить подписанное сообщение. Требование невозможности подмены сообщения заведомо выполняется, так как преобразование E взаимно однозначно.

Для защиты от создания злоумышленником подписанного сообщения можно применить некоторое взаимно-однозначное отображение $R: M \mapsto \tilde{M}$, вносящее избыточность в представление исходного сообщения, например, путем увеличения его длины, а затем уже вычислять подпись $S = D(\tilde{M})$. В этом случае злоумышленник, подбирая S и вычисляя значения $\tilde{M} = E(S)$, будет сталкиваться с проблемой отыскания таких значений \tilde{M} , для которых существует прообраз M . Если отображение R выбрано таким, что число возможных образов \tilde{M} значительно меньше числа всех возможных последовательностей той же длины, то задача создания подписанного сообщения будет сложной.

Другой подход к построению схем цифровых подписей на основе систем шифрования с открытым ключом состоит в использовании бесключевых хэш-функций. Для заданного сообщения M сначала вычисляется значение хэш-функции $h(M)$, а затем уже значение подписи $S = D(h(M))$. Ясно, что в таком случае по значению подписи уже нельзя восстановить сообщение. Поэтому подписи необходимо передавать вместе с сообщениями. Такие подписи получили название *цифровых подписей с дополнением*. Заметим, что системы подписи, построенные с использованием бесключевых хэш-функций, заведомо удовлетворяют всем требованиям, предъ-

являемым к цифровым подписям. Например, невозможно создание сообщения с известным значением подписи, поскольку бесключевая хэш-функция должна быть однонаправленной.

В качестве системы шифрования с открытыми ключами можно использовать, например, систему RSA.

§ 14.3. Цифровая подпись Фиата — Шамира

Рассмотрим подход к построению схемы цифровой подписи, основанной на сложности задач факторизации больших целых чисел и извлечения квадратного корня в кольце вычетов. Идея построения схемы принадлежит А. Фиату и А. Шамиру (см. гл. 12). Приведем одну из модификаций схемы, предложенную ими совместно с У. Фейджем. В ней реализуется цифровая подпись с дополнением.

Пусть h — некоторая хэш-функция, преобразующая исходное сообщение в битовую строку длины m . Выберем различные простые числа p и q и положим $n = pq$. В качестве секретного ключа каждый абонент должен сгенерировать m различных случайных чисел $a_1, a_2, \dots, a_m \in Z_n$. Открытым ключом объявляется набор чисел $b_1, b_2, \dots, b_m \in Z_n$, где $b_i = (a_i^{-1})^2 \bmod n$, $i = 1, \dots, m$.

Алгоритм вычисления цифровой подписи для сообщения M состоит в выполнении следующих действий:

1. Выбрать случайное число r , $1 \leq r \leq n - 1$.
2. Вычислить $u = r^2 \bmod n$.
3. Вычислить $h(M, u) = s = (s_1, s_2, \dots, s_m)$.

4. Вычислить $t = r \prod_{i=1}^m a_i^{s_i} \bmod n$.

5. Подписью для сообщения M положить пару (s, t) .

Алгоритм проверки подписи состоит в выполнении следующих действий:

1. По открытому ключу $b_1, b_2, \dots, b_m \bmod n$ и значению t вычислить

$$w = t^2 \prod_{i=1}^m b_i^{s_i} \bmod n.$$

2. Вычислить $h(M, w) = s'$.

3. Проверить равенство $s = s'$.

Достоинствами описанной схемы являются возможность выработки цифровых подписей для нескольких различных сообщений с использованием одного секретного ключа, а также сравнительная простота алгоритмов вычисления и проверки подписи. Например, для схемы цифровой подписи, основанной на алгоритме RSA, соответствующие алгоритмы требуют выполнения значительно большего числа умножений. Попытка компрометации этой схемы сталкивается с необходимостью решения сложной задачи нахождения квадратных корней по модулю n .

Недостатком схемы является большая длина ключа, которая определяется числом m . Если двоичная запись числа n содержит l знаков, то длина секретного ключа составляет ml бит, а открытого ключа — $(m+1)l$ бит. При этом необходимо учитывать, что для обеспечения достаточной стойкости данной схемы цифровой подписи числа l и m должны иметь в своей двоичной записи несколько сотен бит.

§ 14.4. Цифровая подпись Эль-Гамала

Схема цифровой подписи Эль-Гамала основана на сложности другой задачи — вычисления значения логарифма в конечном поле (см. также § 11.2).

Пусть p — простое число и α — примитивный элемент поля Z_p . Выберем случайное число a в интервале

$1 \leq a \leq p-2$ и вычислим значение $\beta = \alpha^a \bmod p$. Число a является секретным ключом, а набор (p, α, β) — открытым ключом.

Подпись для сообщения M вычисляется с помощью следующего алгоритма:

1. Выбрать случайное целое число r , $1 \leq r \leq p-2$.
2. Вычислить $\gamma = \alpha^r \bmod p$.
3. Для $x = M$ вычислить $\delta = (x - a\gamma)r^{-1} \bmod (p-1)$.
4. Подписью для сообщения M положить пару (γ, δ) .

Алгоритм проверки подписи заключается в проверке сравнения $\beta^r \gamma^\delta \equiv \alpha^x \pmod{p}$. Если оно верно, то подпись принимается, если нет, то отвергается.

Основным достоинством такой схемы цифровой подписи является возможность выработки цифровых подписей для большого числа сообщений с использованием одного секретного ключа. При этом попытка компрометации схемы сталкивается с необходимостью решения сложной математической задачи, связанной с нахождением решений показательных уравнений, в частности — с нахождением значения логарифма в поле Z_p .

Сделаем два замечания.

Первое касается выбора числа r . Оно должно уничтожаться сразу после вычисления подписи. Действительно, зная число r и значение подписи, легко вычислить секретный ключ a :

$$a = (x - r\delta)\gamma^{-1} \bmod (p-1).$$

Поэтому подпись может быть полностью скомпрометирована. Кроме того, число r должно быть действительно случайным и не должно повторяться для различных подписей, полученных при одном значении секретного ключа. В противном случае также можно вычислить секретный ключ a .

Второе замечание связано с тем, что реально при вычислении подписи на шаге 3 алгоритма в качестве x целесообразнее использовать свертку $x = h(M)$, а не само сообщение M . Это защищает схему подписи от возможности подбора сообщений с известным значением подписи. Существует несколько способов такого подбора. Например, если выбрать случайно числа i, j , удовлетворяющие условиям $0 < i < p-1$, $0 < j < p-1$, $(j, p-1) = 1$, и положить

$$\begin{aligned}\gamma &= \alpha^i \beta^j \pmod{p}, \\ \delta &= -\gamma j^{-1} \pmod{p-1}, \\ x &= -\gamma ij^{-1} \pmod{p-1},\end{aligned}$$

то легко убедиться в том, что пара (γ, δ) является верной цифровой подписью для сообщения $M = x$.

Схема цифровой подписи Эль-Гамала послужила образцом для построения большого семейства во многом сходных по своим свойствам схем подписи. В их основе лежит проверка сравнения вида

$$\alpha^A \beta^B \equiv \gamma^C \pmod{p},$$

в котором тройка (A, B, C) совпадает с одной из перестановок чисел $\pm x$, $\pm \delta$ и $\pm \gamma$ при некотором выборе знаков. Например, исходная схема Эль-Гамала получается при $A = x$, $B = -\gamma$ и $C = \delta$. На базе схем подписи из этого семейства построены и стандарты цифровой подписи США и России. Так, в американском стандарте DSS (*Digital Signature Standard*) используются значения $A = x$, $B = \gamma$ и $C = \delta$, а в российском стандарте — значения $A = -x$, $B = \delta$ и $C = \gamma$.

Еще одним достоинством схемы Эль-Гамала является возможность уменьшения длины подписи путем замены пары

чисел (γ, δ) на пару чисел $(\gamma \bmod q, \delta \bmod q)$, где q является некоторым простым делителем числа $p-1$. При этом проверочное равенство по модулю p следует заменить на модифицированное равенство по модулю q :

$$(\alpha^A \beta^B \bmod p) \bmod q = \gamma^C \bmod q.$$

Именно так сделано в американском стандарте цифровой подписи DSS.

§ 14.5. Одноразовые цифровые подписи

Рассмотренные системы цифровой подписи, основанные на сложности задач разложения целых чисел на множители, вычисления квадратного корня или логарифмирования в конечных полях, имеют один потенциальный недостаток. Он состоит в возможности построения новых эффективных алгоритмов для решения этих математических задач. Поэтому в реальных схемах длину ключа выбирают с определенным превышением необходимой величины для обеспечения достаточного запаса стойкости. Это, в свою очередь, значительно усложняет алгоритмы вычисления и проверки подписи. Поэтому представляется весьма привлекательной задача построения схем цифровой подписи на основе симметричных систем шифрования, свободных от подобных недостатков.

Рассмотрим, например, схему цифровой подписи Диффи — Лампорта на основе симметричных систем шифрования.

Пусть требуется подписать сообщение $M = m_1 m_2 \dots m_n$, $m_i \in \{0, 1\}$, $i = 1, \dots, n$. Согласно схеме Диффи — Лампорта подписывающий сначала выбирает $2n$ случайных секретных ключей

$$K = [(k_{10}, k_{11}), \dots, (k_{n0}, k_{n1})]$$

для используемой им симметричной шифрсистемы, затем n пар случайных чисел

$$S = [(S_{10}, S_{11}), \dots, (S_{n0}, S_{n1})],$$

где $S_{ij} \in \{0,1\}$, $i = 1, \dots, n$, $j = 0, 1$, и вычисляет значения

$$R_{ij} = E_{k_{ij}}(S_{ij}), \quad i = 1, \dots, n, \quad j = 0, 1.$$

Наборы S и $R = [(R_{10}, R_{11}), \dots, (R_{n0}, R_{n1})]$ являются открытыми и помещаются в общедоступном месте так, чтобы каждый мог прочитать их, но записать их туда мог бы только автор подписи.

Подпись для сообщения M имеет вид $(k_{1m_1}, \dots, k_{nm_n})$. Чтобы убедиться в ее правильности, следует проверить равенства

$$R_{ij} = E_{k_{ij}}(S_{ij}), \quad j = m_i, \quad i = 1, \dots, n.$$

Недостатком этой схемы является слишком большой размер подписи, который может превышать размер самого подписываемого сообщения. Имеется несколько способов избавиться от этого недостатка (см. [Бер92]). Рассмотрим некоторые из них.

Во-первых, можно хранить не $2n$ значений секретных ключей, а лишь один секретный ключ k . Для этого можно воспользоваться, например, одной из следующих схем формирования последовательности K :

$$k_{ij} = E_k(i, j), \quad j = 0, 1, \quad i = 1, \dots, n;$$

$$k_{ij} = \begin{cases} E_k(i-1,1), & j = 0, \\ E_k(i,0), & j = 1, \end{cases} \quad k_{10} = E_k(0), \quad i = 1, \dots, n.$$

Во-вторых, можно аналогичным образом свернуть набор открытых значений S . В-третьих, можно подписывать не само сообщение, а его свертку, если воспользоваться какой-либо хэш-функцией. Можно использовать и другие подходы, позволяющие сократить как длину подписи, так и размеры открытого и секретного ключей.

Вместе с тем подобные модификации не устраняют главного недостатка рассматриваемых подписей, состоящего в том, что после проверки подписи либо весь секретный ключ, либо его часть становятся известными проверяющему. Поэтому рассмотренная схема цифровой подписи является по существу одноразовой.

Контрольные вопросы

1. Что общего между обычной и цифровой подписями? Чем они различаются?
2. Какие задачи позволяет решить цифровая подпись?
3. В чем заключается принципиальная сложность в практическом применении систем цифровой подписи?
4. Почему в криптографических системах, основанных на открытых ключах, нельзя использовать одинаковые ключи для шифрования и цифровой подписи?
5. Проверьте, что указанный в тексте способ подбора подписанных сообщений для схемы Эль-Гамала действительно дает верные цифровые подписи.

Глава 15

Протоколы распределения ключей

Различают следующие типы протоколов распределения ключей:

- протоколы передачи (уже сгенерированных) ключей;
- протоколы (совместной) выработки общего ключа (открытое распределение ключей);
- схемы предварительного распределения ключей.

Различают также протоколы распределения ключей между отдельными участниками и между группами участников информационного взаимодействия.

§ 15.1. Передача ключей с использованием симметричного шифрования

Имеются протоколы, в которых стороны осуществляют передачу ключей при непосредственном взаимодействии, то есть двусторонние протоколы или, иначе, протоколы типа “точка-точка”, и протоколы с централизованным распределением ключей, в которых предусмотрена третья сторона, играющая роль доверенного центра.

Двусторонние протоколы

Рассмотрим сначала двусторонние протоколы передачи ключей с использованием симметричного шифрования. Различают протоколы, в которых стороны заранее располагают какой-либо известной им обоим секретной информацией, и протоколы, не требующие этого условия.

Пусть стороны A и B заранее обладают общей секретной информацией. Допустим, что это — секретный ключ k_{AB} .

Тогда для передачи ключа k стороны могут использовать одностороннюю передачу:

$$A \rightarrow B: E_{k_{AB}}(k, t, B),$$

где E — алгоритм шифрования, t — метка времени, B — идентификатор абонента B (для краткости вместо $id(B)$ будем использовать лишь один символ B).

Подчеркнем, что если не передавать метки времени, то злоумышленник может осуществить повторную передачу того же сообщения. Если же не указывать идентификатора адресата, то злоумышленник может вернуть отправителю перехваченное сообщение, что в некоторых ситуациях может быть опасным, поскольку абонент A не сможет установить, что это сообщение получено не от абонента B .

Заметим, что в приведенном протоколе вместо шифрования можно было использовать ключевую хэш-функцию, зависящую от общего ключа:

$$A \rightarrow B: k \oplus h_{k_{AB}}(t, B).$$

Если дополнительно требуется аутентификация сеанса, то можно использовать следующий протокол типа “запрос-ответ”:

$$(1) B \rightarrow A: r_B,$$

$$(2) A \rightarrow B: E_{k_{AB}}(k, r_B, B),$$

где r_B — случайное число, сгенерированное абонентом B и переданное абоненту A в начале сеанса. При использовании хэш-функции подобный протокол может выглядеть так:

- (1) $B \rightarrow A: r_B,$
- (2) $A \rightarrow B: k \oplus h_{k_{AB}}(r_B, B).$

Если требуется двусторонняя аутентификация, то можно модифицировать последний протокол, предоставив возможность стороне A путем генерации своего случайного числа r_A и введения его в сообщение на шаге (2) протокола убедиться в том, что он имеет дело именно с абонентом B .

Исходный протокол можно модифицировать так, чтобы искомый ключ k генерировался не одной стороной, а являлся результатом двустороннего обмена.

Пусть абонентами A и B помимо случайных чисел r_A и r_B генерируют также случайные числа k_A и k_B соответственно. Тогда в результате выполнения протокола

- (1) $B \rightarrow A: r_B,$
- (2) $A \rightarrow B: E_{k_{AB}}(k_A, r_A, r_B, B),$
- (3) $B \rightarrow A: E_{k_{AB}}(k_B, r_B, r_A, A),$

каждая из сторон может вычислить общий ключ с помощью некоторой функции f по правилу $k = f(k_A, k_B)$. Подчеркнем, что в этом протоколе ни одна из сторон не может знать заранее значения ключа.

Приведем теперь “бесключевой” протокол А. Шамира, позволяющий передать ключ без использования какой-либо общей секретной информации.

Пусть имеется некоторое коммутирующее шифрующее преобразование E . Это означает, что при всех сообщениях x и ключах k_1 и k_2 выполняется равенство

$$E_{k_1}(E_{k_2}(x)) = E_{k_2}(E_{k_1}(x)).$$

Тогда пользователи A и B могут реализовать следующий трехшаговый протокол для передачи секретного ключа k от A к B :

- (1) $A \rightarrow B: E_{k_A}(k),$
- (2) $B \rightarrow A: E_{k_B}(E_{k_A}(k)),$
- (3) $A \rightarrow B: D_{k_A}(E_{k_B}(E_{k_A}(k))).$

Заметим, что в этом протоколе можно использовать не каждое коммутирующее преобразование E . Например, легко заметить, что для преобразования $E_{k_A}(k) = k \oplus \Gamma$ протокол оказывается заведомо нестойким. Поэтому в протоколе Шамира рекомендуется использовать преобразование вида $E_{k_A}(k) = k^a \bmod p$, в котором константа a определяется ключом k_A .

Трехсторонние протоколы

Рассмотрим протоколы распределения ключей между парами участников с использованием третьей стороны T , называемой центром. В этом качестве обычно выступает некоторый выделенный узел сети, или сервер, которому доверяют все участники. Центр T хранит ключи всех абонентов сети. Поэтому схема ключевых взаимоотношений графически представляет собой звезду.

Один из первых протоколов такого типа был предложен в работе [Nee78]. Он заключается в выполнении следующих шагов:

- (1) $A \rightarrow T: A, B, r_A,$
- (2) $T \rightarrow A: E_{k_{A1}}(r_A, B, k, E_{k_{B1}}(k, A)) ,$

$$(3) A \rightarrow B: E_{k_B} (k, A) ,$$

$$(4) B \rightarrow A: E_k (r_B) ,$$

$$(5) A \rightarrow B: E_k (r_B - 1) .$$

В результате выполнения трех первых шагов протокола пользователи A и B получают сгенерированный центром T общий ключ k для организации взаимодействия. Четвертый и пятый шаги предназначены для аутентификации пользователя A и подтверждения правильности получения ключа обеими сторонами.

Слабость этого протокола заключается в возможности повторной передачи абоненту B сообщения, переданного на шаге (3). При этом абонент B не имеет возможности установить, что полученный ключ k уже был использован. Поэтому в случае компрометации этого ключа злоумышленник может аутентифицироваться и передавать сообщения от имени A .

Недостаток этого протокола устранен в протоколе *Kerberos*. Рассмотрим сначала базовый протокол, применяемый в протоколе аутентификации и распределения ключей *Kerberos*. Он состоит из следующих шагов:

$$(1) A \rightarrow T: A, B, r_A ,$$

$$(2) T \rightarrow A: E_{k_{AT}} (k, r_A, L, B) , \text{ билет} ,$$

$$(3) A \rightarrow B: \text{ билет} , \text{ аутентификатор} ,$$

$$(4) B \rightarrow A: E_k (t, k_B) .$$

Здесь “билетом” названа величина $E_{k_{BT}} (k, A, L)$, “аутентификатором” — величина $E_k (A, t, k_A)$, t — метка времени; L — период времени действия билета, r_A — случайное число, сгенерированное абонентом A и вставленное в передаваемое сообщение для взаимной аутентификации, а k_A и k_B — случайные числа, сгенерированные абонентами A и B соот-

ответственно, и используемые либо в качестве ключа шифрования информации другой стороне, либо для выработки общего ключа $k_{AB} = f(k_A, k_B)$ с помощью некоторой функции f .

В полном протоколе *Kerberos* описанный выше базовый протокол используется два раза. Дело в том что в нем предусмотрено два сервера (см. рис. 51). Первый — “сервер аутентификации”, обозначаемый *AS*, выдает так называемые “билеты для получения билетов” (*tgt*), содержащие ключи, предназначенные для длительного использования. Вторым сервером, *TGS*, — “сервер выдачи билетов”, выдает обычные билеты для доступа к сетевым ресурсам и обращения к другим пользователям.

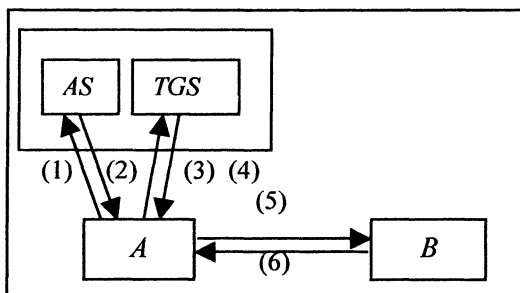


Рис. 51

Сообщения, передаваемые согласно этому протоколу, выглядят следующим образом:

- (1) $A \rightarrow AS: A, TGS, r_A,$
- (2) $AS \rightarrow A: E_{k_{A,AS}}(k_{A,TGS}, r_A, L_1, TGS), tgt,$
- (3) $A \rightarrow TGS: tgt, \text{аутентификатор}_1, B, r_A',$
- (4) $TGS \rightarrow A: E_{K_{A,TGS}}(k, r_A', L_2, B), \text{билет},$
- (5) $A \rightarrow B: \text{билет}, \text{аутентификатор}_2,$

$$(6) B \rightarrow A: E_k(t_2, k_B),$$

где

$$tgt = E_{k_{AS, TGS}}(k_{A, TGS}, A, L_1),$$

$$\text{аутентификатор}_1 = E_{k_{A, TGS}}(A, t_1),$$

$$\text{билет} = E_{k_{B, TGS}}(K, A, L_2),$$

$$\text{аутентификатор}_2 = E_k(A, t_2, k_A).$$

Благодаря введению второго сервера нагрузка на первый сервер уменьшается во много раз. Первый сервер должен быть наиболее защищенным, поскольку он хранит главные ключи всех пользователей. Серверов второго типа может быть несколько, и они могут соответствовать определенной подсети или определенному типу ресурса.

Приведем еще один протокол распределения ключей с использованием сервера (см. [Otw87]), предпочтительный для случая, когда сервер находится в более удобном расположении для второго абонента. Протокол состоит в выполнении следующих действий:

$$(1) A \rightarrow B: r, A, B, E_{k_{AT}}(r_A, r, A, B),$$

$$(2) B \rightarrow T: r, A, B, E_{k_{AT}}(r_A, r, A, B), E_{k_{BT}}(r_B, r, A, B),$$

$$(3) T \rightarrow B: E_{k_{AT}}(r_A, k), E_{k_{BT}}(r_B, k),$$

$$(4) B \rightarrow A: E_{k_{AT}}(r_A, k).$$

Пользователь A генерирует два случайных числа: первое (r_A) используется, как и раньше, для взаимной аутентификации, а второе (r) — для аутентификации сеанса связи (вместо него может быть использована метка времени).

Этот протокол можно дополнить еще одним шагом для обеспечения взаимной аутентификации сторон и подтверждения правильности полученного ключа:

$$(4') B \rightarrow A: E_{k_{Ar}}(r_A, k), E_k(r, r_B),$$

$$(5) A \rightarrow B: E_k(r).$$

§ 15.2. Передача ключей с использованием асимметричного шифрования

Рассмотрим варианты использования асимметричного шифрования для передачи секретных ключей симметричных криптосистем.

Протоколы без использования цифровой подписи

Для передачи ключа k можно использовать следующий одношаговый протокол:

$$A \rightarrow B: E_{k_B}(k, t, A),$$

где E — алгоритм шифрования с открытым ключом, t — метка времени, вставляемая для предотвращения возможности повторного использования ключа.

Для осуществления взаимной аутентификации и подтверждения правильности получения ключа можно воспользоваться протоколом из [Nee78]:

$$(1) A \rightarrow B: E_B(k_1, A),$$

$$(2) B \rightarrow A: E_A(k_1, k_2),$$

$$(3) A \rightarrow B: E_B(k_2).$$

Производя расшифрование полученных сообщений на втором и третьем шагах, стороны убеждаются в том, что они имеют

дело именно с нужной стороной и что другая сторона правильно расшифровала полученное значение ключа.

Протоколы с использованием цифровой подписи

При использовании цифровой подписи аутентифицированный протокол передачи ключей может содержать только одно сообщение и иметь, например, один из следующих трех видов:

$$A \rightarrow B: E_B(k, t, S_A(B, k, t))$$

(шифрование подписанного ключа);

$$A \rightarrow B: E_B(k, t), S_A(B, k, t)$$

(зашифрование и подпись ключа);

$$A \rightarrow B: t, E_B(A, k), S_A(B, t, E_B(A, k))$$

(подпись зашифрованного ключа).

Сертификаты открытых ключей

Как правило, при использовании открытых ключей хранят не сами ключи, а их сертификаты. *Сертификат* представляет собой набор данных

$$C_A = (A, k_A, t, S_{K_{I_1}}(A, k_A, t)),$$

состоящий из идентификатора абонента A , его открытого ключа k_A и, быть может, еще какой-либо дополнительной информации, например, времени t выдачи сертификата и срока его действия, заверенных цифровой подписью доверенного центра TA или заслуживающего доверия лица. Сертификат

предназначен для исключения возможности подмены открытого ключа при его хранении или пересылке.

Получив такой сертификат и проверив цифровую подпись, можно убедиться в том, что открытый ключ действительно принадлежит данному абоненту.

Международный стандарт ССІТТ Х.509 определяет следующий протокол аутентификации с одновременным распределением ключей:

$$(1) A \rightarrow B: C_A, D_A, S_A(D_A),$$

$$(2) B \rightarrow A: C_B, D_B, S_B(D_B),$$

$$(3) A \rightarrow B: r_B, B, S_A(r_B, B),$$

где C_A и C_B — сертификаты сторон, S_A и S_B — цифровые подписи сторон,

$$D_A = (t_A, r_A, B, data_1, E_B(k_1)),$$

$$D_B = (t_B, r_B, A, r_A, data_2, E_A(k_2))$$

— наборы передаваемых и подписываемых данных. В поля *data* заносится дополнительная информация для аутентификации источника. Третий шаг протокола требуется стороне *B* для подтверждения того, что она действительно взаимодействует со стороной *A*.

§ 15.3. Открытое распределение ключей

Открытое распределение ключей позволяет двум абонентам выработать общий секретный ключ путем динамического взаимодействия на основе обмена открытыми сообщениями без какой-либо общей секретной информации, распределяемой заранее. Важным преимуществом открытого распределения является также то, что ни один из абонентов заранее не

может определить значения ключа, так как ключ зависит от сообщений, передаваемых в процессе обмена.

Первый алгоритм открытого распределения ключей был предложен У. Диффи и М. Хеллманом [Dif76]. Для его выполнения стороны должны договориться о значениях большого простого числа p и образующего элемента α мультипликативной группы $Z_p^* = \{1, 2, \dots, p-1\}$. Для выработки общего ключа k они должны сгенерировать случайные числа x , $1 \leq x \leq p-2$, и y , $1 \leq y \leq p-2$, соответственно. Затем они должны обменяться сообщениями в соответствии с протоколом:

$$(1) \quad A \rightarrow B: \alpha^x \bmod p,$$

$$(2) \quad B \rightarrow A: \alpha^y \bmod p.$$

Искомый общий ключ теперь вычисляется по формуле:

$$k = (\alpha^y)^x = (\alpha^x)^y \bmod p.$$

Недостатком этого протокола является возможность атаки типа “злоумышленник в середине”, состоящей в следующем. Предположим, что злоумышленник имеет возможность осуществлять подмену передаваемых абонентами сообщений. Тогда, выбрав числа x^* и y^* и подменив сообщения $\alpha^x \bmod p$ и $\alpha^y \bmod p$ на $\alpha^{x^*} \bmod p$ и $\alpha^{y^*} \bmod p$ соответственно, он может сформировать ключи

$$k_1 = (\alpha^{x^*})^{y^*} \bmod p$$

и

$$k_2 = (\alpha^{y^*})^{x^*} \bmod p$$

для связи с пользователями A и B соответственно. В результате злоумышленник получает возможность полностью кон-

тролировать обмен сообщениями между абонентами A и B . При этом они не смогут обнаружить подмену и будут уверены, что связываются непосредственно друг с другом.

Рассмотрим два протокола, устраняющие этот недостаток. Первый протокол, называемый STS (*station-to-station*), предполагает, что пользователи применяют цифровую подпись, которой подписываются передаваемые по протоколу Диффи—Хеллмана сообщения:

$$(1) \quad A \rightarrow B: \alpha^x \bmod p,$$

$$(2) \quad B \rightarrow A: \alpha^y \bmod p, E_k(S_B(\alpha^y, \alpha^x)),$$

$$(3) \quad A \rightarrow B: E_k(S_A(\alpha^x, \alpha^y)).$$

Здесь S_A и S_B — цифровые подписи пользователей A и B соответственно, k — искомый общий ключ. Они позволяют гарантировать достоверность получения сообщения именно от того пользователя, от которого это сообщение получено. Шифрование значений подписей пользователей введено для того, чтобы обеспечить взаимное подтверждение правильности вычисления значения ключа.

Еще один подход также предполагает наличие у абонентов открытых ключей, но вместо цифровой подписи предлагается использовать модифицированную процедуру выработки общего ключа. Рассмотрим протокол МТІ (названный так в честь его авторов: Т. Мацумото, И. Такашима и Х. Имаи, см. [Mat86]).

Предположим, что пользователи A и B имеют секретные ключи a , $1 \leq a \leq p-2$, и b , $1 \leq b \leq p-2$, соответственно, и публикуют свои открытые ключи $z_A = \alpha^a \bmod p$ и $z_B = \alpha^b \bmod p$. Для выработки общего секретного ключа k они должны сгенерировать случайные числа x , $1 \leq x \leq p-2$,

и y , $1 \leq y \leq p-2$, соответственно, а затем обменяться следующими сообщениями:

$$(1) \quad A \rightarrow B: \alpha^x \bmod p,$$

$$(2) \quad B \rightarrow A: \alpha^y \bmod p.$$

Искомый общий ключ вычисляется по формуле:

$$k = (\alpha^y)^a z_B^x = (\alpha^x)^b z_A^y = \alpha^{xb+ya} \bmod p.$$

Теперь любая подмена сообщений приведет к тому, что все стороны получают различные значения ключа, что, в свою очередь, приведет к невозможности чтения всей передаваемой информации.

§ 15.4. Предварительное распределение ключей

Большинство криптографических систем требуют проведения предварительного распределения секретных ключей. Для предварительного распределения стороны могут обменяться ключами при личной встрече, либо поручить доставку ключей специально назначенному доверенному курьеру, либо использовать для передачи некоторый выделенный защищенный канал. В зависимости от назначения криптографической системы иногда оказывается удобным распределять не сами ключи, а некоторые вспомогательные ключевые материалы, на основании которых каждый участник или группа участников могут самостоятельно вычислить необходимый ключ, используя для этого некоторую установленную заранее процедуру.

Рассмотрим ситуации, в которых необходимо проводить предварительное распределение ключей.

Схемы предварительного распределения ключей в сети связи

Если число абонентов сети засекреченной связи невелико, то и число распределяемых ключей также невелико. Для больших же сетей распределение ключей становится очень серьезной проблемой. Она заключается в том, что для сети, в которой работают n абонентов, необходимо выработать заранее и хранить в дальнейшем $n(n-1)/2$ ключей. Кроме того, каждому абоненту сети необходимо передать ключи для связи с остальными $n-1$ абонентами, которые абонент должен постоянно хранить. Например, для сети со 100 абонентами нужно сгенерировать и хранить почти 5000 ключей, причем каждый абонент при этом должен хранить у себя 99 ключей.

Для уменьшения объема хранимой ключевой информации применяются различные *схемы предварительного распределения ключей* в сети связи. Их суть заключается в том, что в действительности вначале происходит распределение не самих ключей, а некоторых вспомогательных ключевых материалов, занимающих меньшие объемы. На основании этих материалов каждый абонент сети может самостоятельно вычислить по некоторому алгоритму необходимый для связи ключ. Такой подход позволяет уменьшить объемы как хранимой, так и распределяемой секретной информации.

В качестве примера рассмотрим *схему Блома* [Blo83] распределения ключей между n абонентами, для которой процедура вычисления ключа заключается в вычислении значения некоторого симметрического многочлена над конечным полем.

Выберем поле F , имеющее конечное, но достаточно большое число элементов, и зафиксируем n различных элементов $r_1, \dots, r_n \in F$, отличных от нуля. Каждый элемент r_i припишем i -му абоненту сети, $i = \overline{1, n}$. Эти элементы не являются секретными и могут храниться на общедоступном сервере сети. Выберем теперь многочлен над полем F степени $2m$, $1 \leq m < n$, вида

$$f(x, y) = \sum_{i=0}^m \sum_{j=0}^m a_{ij} x^i y^j,$$

где $a_{ij} = a_{ji}$, $i \neq j$, $i, j = \overline{0, m}$. Его коэффициенты являются секретными и должны храниться только в центре распределения ключей. Каждый абонент A получает в качестве ключевых материалов набор $(a_0^{(A)}, a_1^{(A)}, \dots, a_k^{(A)})$, состоящий из коэффициентов многочлена

$$g_A(x) = f(x, r_A) = a_0^{(A)} + a_1^{(A)}x + \dots + a_k^{(A)}x^k.$$

Для связи между абонентами A и B теперь можно использовать общий ключ k_{AB} :

$$k_{AB} = k_{BA} = f(r_A, r_B) = g_B(r_A) = g_A(r_B),$$

который могут легко вычислить оба абонента.

При использовании данной схемы каждый абонент должен хранить $m+1$ секретных значений вместо $n-1$, общее же число секретных коэффициентов многочлена f равно $m(m+1)/2$.

Имеют место следующие результаты.

Лемма. Пусть имеется матрица Вандермонда

$$\begin{pmatrix} 1 & r_1 & r_1^2 & \dots & r_1^m \\ 1 & r_2 & r_2^2 & \dots & r_2^m \\ \dots & \dots & \dots & \dots & \dots \\ 1 & r_{m+1} & r_{m+1}^2 & \dots & r_{m+1}^m \end{pmatrix}.$$

Если элементы r_1, r_2, \dots, r_{m+1} попарно различны и отличны от нуля, тогда обратима над полем F .

Теорема 1. *Схема Блома предварительного распределения ключей между n абонентами, использующая многочлен степени m , $1 \leq m < n$, является стойкой к компрометации ключей t абонентов. Это означает, что при раскрытии ключевых материалов, принадлежащих любым t абонентам, злоумышленник, зная их значения, не сможет определить ни один из ключей для связи между остальными $(n - t)$ абонентами.*

Доказательство. Запишем выражение для общего ключа абонентов A и B , вычисляемого по формуле

$$k_{AB} = f(r_A, r_B) = \sum_{i=0}^m \sum_{j=0}^m a_{ij} r_A^i r_B^j,$$

в матричном виде:

$$k_{AB} = (1, r_A, r_A^2, \dots, r_A^m) \begin{pmatrix} a_{00} & a_{01} & \dots & a_{0m} \\ a_{10} & a_{10} & \dots & a_{1m} \\ \dots & \dots & \dots & \dots \\ a_{m0} & a_{m0} & \dots & a_{mm} \end{pmatrix} \begin{pmatrix} 1 \\ r_B \\ r_B^m \\ \dots \\ r_B^m \end{pmatrix}.$$

Здесь матрица $\Lambda = (a_{ij})_{m \times m}$ составлена из коэффициентов многочлена $f(x, y)$ и является симметричной.

Абонент A имеет секретный набор $(a_0^{(A)}, a_1^{(A)}, \dots, a_m^{(A)})$, представляющий собой коэффициенты многочлена

$$g_A(x) = f(x, r_A) = a_0^{(A)} + a_1^{(A)} x + \dots + a_m^{(A)} x^m,$$

или иначе

$$(a_0^{(A)}, a_1^{(A)}, a_2^{(A)}, \dots, a_m^{(A)}) = (1, r_A, r_A^2, \dots, r_A^m) \cdot \Lambda.$$

Предположим, что известны ключевые материалы, принадлежащие t абонентам. Покажем, что по ним нельзя опре-

делить ни один из ключей для связи между оставшимися ($n - m$) абонентами.

Пусть, например, известны секретные наборы ключевых материалов m абонентов — A, B, \dots, C . Будем искать ключ для связи между абонентами U и W , не входящими в множество $\{A, B, \dots, C\}$. Для этого рассмотрим матричное равенство

$$\begin{pmatrix} k_{AA} & k_{AB} & \dots & k_{AC} & k_{AU} \\ k_{BA} & k_{BB} & \dots & k_{BC} & k_{BU} \\ \dots & \dots & \dots & \dots & \dots \\ k_{CA} & k_{CB} & \dots & k_{CC} & k_{CU} \\ k_{WA} & k_{WB} & \dots & k_{WC} & k_{WU} \end{pmatrix} = \begin{pmatrix} 1 & r_A & r_A^2 & \dots & r_A^m \\ 1 & r_B & r_B^2 & \dots & r_B^m \\ \dots & \dots & \dots & \dots & \dots \\ 1 & r_C & r_C^2 & \dots & r_C^m \\ 1 & r_W & r_W^2 & \dots & r_W^m \end{pmatrix} \cdot \Lambda \cdot \begin{pmatrix} 1 & 1 & \dots & 1 & 1 \\ r_A & r_B & \dots & r_C & r_U \\ r_A^k & r_B^k & \dots & r_C^k & r_U^k \\ \dots & \dots & \dots & \dots & \dots \\ r_A^m & r_B^m & \dots & r_C^m & r_U^m \end{pmatrix}.$$

В матрице из левой части равенства не известен только искомый ключ k_{WU} . В правой части неизвестная матрица Λ умножается слева и справа на обратимые матрицы, так как по лемме их определители не равны нулю.

Выражая теперь матрицу Λ из данного равенства, замечаем, что при произвольно заданном значении ключа k_{WU} из поля F матрица Λ находится как произведение трех матриц. (Проверьте, что данная матрица должна быть симметричной матрицей, составленной из коэффициентов многочлена.)⁸

Таким образом, при каждом значении ключа k_{WU} может быть найден многочлен $f(x, y)$, при котором данный ключ по-

⁸ Указание замените в нижней строке рассматриваемого матричного равенства символ W на U и заметьте, что нижняя строка полученной матрицы будет линейной комбинацией строк исходной матрицы

лучается в качестве решения. Это свидетельствует о невозможности получения информации об этом ключе на основании известных значений.

С другой стороны, если известны ключевые материалы, принадлежащие $m + 1$ абонентам, то, очевидно, из рассматриваемого равенства матрица (a_{ij}) всегда находится однозначно. Теорема доказана.

Теорема 2. *Если схема предварительного распределения ключей между n абонентами является стойкой к компрометации ключей m абонентов, $1 \leq m < n$, и использует l -битовые ключи, то каждый абонент должен хранить не менее $l(m + 1)$ бит ключевых материалов.*

Докажите эту теорему самостоятельно в качестве упражнения.

Из теорем 1 и 2 следует, что для заданного числа m схема Блома дает минимальное по объему количество хранимых у абонента ключевых материалов.

Рассмотрим еще одну схему предварительного распределения ключей, которая также позволяет значительно сократить общее число хранимых и передаваемых секретных ключей. Она называется KDP (*key distribution patterns*) и основана на схеме пересечений множеств (см. [Dye95]).

Пусть имеется n , $n > 2$, абонентов (пользователей) и множество секретных ключей K , $|K| = q$. Будем считать, что все ключи перенумерованы числами $1, 2, \dots, q$. Выберем некоторое семейство $\{S_1, \dots, S_n\}$ подмножеств множества $\{1, 2, \dots, q\}$. Предварительно абоненту i по защищенному каналу передается множество секретных ключей с номерами из подмножества S_i , $i = \overline{1, n}$. Таким образом, семейство $\{S_1, \dots, S_n\}$ представляет собой таблицу с номерами ключей каждого пользователя. Хотя данная таблица является несекретной, она должна быть защищена от модификаций и подделок.

Если абонент i хочет связаться с абонентом j , то он использует для выработки общего ключа множество ключей, номера которых содержатся в пересечении $S_i \cap S_j$. Если каждый ключ представлен некоторой битовой строкой, то для формирования общего связного ключа можно взять, например, их сумму, или значение некоторой хэш-функции от строки, составленной из ключей, номера которых входят в пересечение множеств $S_i \cap S_j$.

Схемой распределения ключей типа KDP, или KDP(n, q)-схемой, назовем всякое семейство $\{S_1, \dots, S_n\}$ подмножеств множества K , удовлетворяющее следующему условию:

если при некоторых $i, j, r \in \{1, \dots, n\}$ выполнено включение $S_i \cap S_j \subseteq S_r$, то либо $i = r$, либо $j = r$.

Это условие означает, что общий ключ двух абонентов не должен быть известным никакому другому абоненту.

Семейство подмножеств называется *семейством Шпернера*, если ни одно из них не содержится в другом.

Теорема 3. Семейство $\{S_1, \dots, S_n\}$ подмножеств множества K , $|K| = q$, образует KDP(n, q)-схему в том и только в том случае, если множество $\{S_i \cap S_j \mid 1 \leq i < j \leq n\}$ образует семейство Шпернера.

Докажите теорему 3 самостоятельно в качестве упражнения.

Теорема 4 (Sperner, 1928). Если подмножества $\{S_1, \dots, S_m\}$ множества K , $|K| = q$, образуют семейство Шпернера, то

$$m \leq C_q^{\lfloor \frac{q}{2} \rfloor}.$$

Равенство достигается только в случае, если множество $\{S_1, \dots, S_m\}$ совпадает с множеством всех w -элементных подмножеств множества K , где $w = q/2$ при четном q и $w = (q+1)/2$ или $(q-1)/2$ при нечетном q .

Доказательство. Пусть семейство $\{S_1, \dots, S_m\}$ подмножеств множества K состоит из максимально возможного числа элементов. Покажем, что существует семейство Шпернера с не меньшим, чем m , числом множеств, каждое из которых имеет мощность $\left\lfloor \frac{q}{2} \right\rfloor$.

Обозначим $w = \max_{i=1, n} |S_i|$. Предположим, что $w > \left\lfloor \frac{q}{2} \right\rfloor$ и что подмножества $S_1, \dots, S_t, t \leq n$, состоят из w элементов. Рассмотрим подмножества R_1, \dots, R_u множества K , каждое из которых состоит из $w-1$ элементов и содержится хотя бы в одном из подмножеств S_1, \dots, S_t . Нетрудно видеть, что семейство

$$R_1, \dots, R_u, S_{t+1}, \dots, S_m$$

также образует семейство Шпернера. Покажем, что $u \geq t$.

Каждое подмножество S_i содержит ровно w подмножеств R_j . С другой стороны, каждое подмножество R_j входит не более чем в $(q-w+1)$ подмножеств S_i . Поэтому, вычисляя двумя способами число ребер в двудольном графе, вершинами которого являются подмножества S_1, \dots, S_t и R_1, \dots, R_u , а ребра соединяют вложенные подмножества $R_j \subseteq S_i$, получаем неравенство $wt \leq (q-w+1)u$. Отсюда следует, что

$$t \left(\left\lfloor \frac{q}{2} \right\rfloor + 1 \right) \leq wt \leq (q-w+1)u \leq \left(q - \left\lfloor \frac{q}{2} \right\rfloor \right) u,$$

или

$$u \geq \frac{\left\lfloor \frac{q}{2} \right\rfloor + 1}{q - \left\lfloor \frac{q}{2} \right\rfloor} t \geq t.$$

Таким образом, можно считать, что в семействе $\{S_1, \dots, S_n\}$ все элементы состоят не более чем из $\left\lfloor \frac{q}{2} \right\rfloor$ элементов.

Если в семействе $\{S_1, \dots, S_n\}$ имеется множество с меньшим, чем $\left\lfloor \frac{q}{2} \right\rfloor$, числом элементов, то, переходя к семейству $\{S'_1, \dots, S'_n\}$, состоящему из подмножеств $S'_i = K \setminus S_i$, которое также составляет семейство Шпернера, и, повторяя приведенные выше рассуждения, получаем семейство, в котором все множества имеют одинаковое число элементов. Теорема доказана.

Теорема 5. *Для любой KDP(n, q)-схемы каждый абонент должен иметь не менее $\log_2 n$ ключей. Если $n \geq 4$, то $q \geq 2 \log_2 n$.*

Доказательство. Если у какого-либо абонента имеется множество, состоящее менее чем из $\log_2 n$ ключей, то с их помощью можно построить не более чем $n - 2$ различных подмножеств ключей. Поэтому он не сможет сформировать различные ключи для связи с $n - 1$ абонентами.

Вторая оценка получается из очевидного неравенства

$$C_q^{\lfloor q/2 \rfloor} \geq C_n^2,$$

которое справедливо в силу предыдущей теоремы.

Схемы разделения секрета

Схема разделения секрета представляет собой схему предварительного распределения ключей между уполномоченными группами пользователей, в которой ключ заранее определен и одинаков для каждой уполномоченной группы. При этом каждый пользователь получает свою *долю* или “часть секрета”. Схема включает два протокола: протокол

формирования долей (разделения секрета) и распределения их между пользователями и протокол восстановления секрета группой пользователей. Схема должна позволять восстанавливать ключ только тем группам пользователей, которые имеют на это полномочия, и никакая другая группа не должна иметь возможности для восстановления ключа или получения о нем какой-либо информации.

Основное назначение схемы разделения секрета — защита ключа от потери. Обычно для защиты от потери делают несколько копий ключа. С возрастанием числа копий ключа возрастает вероятность его компрометации. Если число копий мало, то велик риск потери ключа. Поэтому лучше “разделить” ключ между несколькими лицами так, чтобы допускалась возможность восстановления ключа при различных обстоятельствах несколькими уполномоченными группами с заранее оговоренным составом участников. Тем самым исключается риск безвозвратной потери ключа.

Еще одно положительное качество схем разделения секрета заключается в разделении ответственности за принятие решения, которое автоматически вводится при определении состава уполномоченных групп. Такая коллективная ответственность нужна для многих приложений, включая принятие важных решений, касающихся применения систем оружия, подписания корпоративных чеков или допуска к банковскому хранилищу.

В простейшем случае, когда имеется только одна группа, состоящая из t пользователей, уполномоченная формировать ключ, схему разделения секрета можно построить следующим образом. Предположим, к примеру, что ключ представляет собой двоичный вектор s длины m . Выберем случайным образом t векторов s_1, \dots, s_t так, чтобы их сумма совпадала с вектором s , и распределим их между пользователями. Теперь, собравшись вместе, они могут легко восстановить значение ключа s , в то время как никакая группа, состоящая из меньшего числа пользователей, не сможет этого сделать. Действительно, в данном случае отсутствие хотя бы одной доли при-

водит к полной неопределенности относительно значения секрета, поскольку для каждого значения искомого секрета найдется возможный вариант значения отсутствующей доли.

Заметим, что если бы мы в предыдущем примере просто разбили вектор на t частей, то такая схема не могла быть схемой разделения секрета, так как знание любой доли давало бы частичную информацию о секрете s .

Другой пример схемы разделения секрета дает пороговая схема Шамира [Sha79].

Пусть $1 < t \leq n$. Схема разделения секрета между n пользователями называется (n, t) -пороговой, если любая группа из t пользователей может восстановить секрет, в то время как никакая группа из меньшего числа пользователей не может получить никакой информации о секрете.

Для построения (n, t) -пороговой схемы А. Шамир предложил использовать многочлен степени $t - 1$ над конечным полем с достаточно большим числом элементов. Как известно, многочлен степени $t - 1$ можно однозначно восстановить по его значениям в t различных точках, но при этом меньшее число точек использовать для интерполяции нельзя.

Выберем поле F и зафиксируем n различных несекретных элементов $r_1, \dots, r_n \in F$, отличных от нуля. Каждый элемент r_i приписан i -му абоненту сети, $i = \overline{1, n}$. Выберем также t случайных элементов a_0, \dots, a_{t-1} поля F и составим из них многочлен $f(x)$ над полем F степени $t - 1$, $1 < t \leq n$,

$$f(x) = \sum_{i=0}^{t-1} a_i x^i.$$

Положим $s = f(0) = a_0$. Вычислим теперь значения

$$s_1 = f(r_1), \dots, s_n = f(r_n)$$

и распределим в качестве долей между участниками наборы

$$(r_i, s_i), i = \overline{1, n}.$$

Для восстановления секрета S можно воспользоваться интерполяционной формулой Лагранжа [Лид88]. Пусть имеются t пар (x_i, y_i) , где $y_i = f(x_i)$. Тогда формула Лагранжа имеет вид

$$f(x) = \sum_{i=0}^{t-1} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j} .$$

Так как $s = f(0)$, то из формулы Лагранжа получаем равенства

$$s = \sum_{i=0}^{t-1} y_i c_i, \quad c_i = \prod_{j \neq i} \frac{x_j}{x_j - x_i} ,$$

причем коэффициенты c_i не зависят от коэффициентов многочлена $f(x)$ и могут быть вычислены заранее.

С помощью полученной формулы любая группа из t пользователей может легко восстановить секрет. В то же время можно показать, что никакая группа из меньшего числа пользователей не может получить никакой информации о секрете (докажите это самостоятельно).

Схема Шамира удобна тем, что она позволяет легко увеличивать число пользователей. Для этого не нужно ничего менять, кроме множества $\{r_1, \dots, r_n\}$, к которому следует добавить новые элементы r_{n+1}, \dots, r_{n+w} . Заметим, что компрометация одной доли делает из (n, t) -пороговой схемы $(n-1, t-1)$ -пороговую схему.

§ 15.5. Способы установления ключей для конференц-связи

Еще один тип распределения ключей между группами пользователей дают протоколы распределения ключей для проведения конференц-связи. Несмотря на внешнюю схожесть с протоколами разделения секрета, они имеют несколько принципиальных отличий. Если протоколы разделения

секрета осуществляют предварительное распределение одного и того же ключевого значения (секрета) по секретным каналам между привилегированными группами пользователей, то протоколы конференц-связи осуществляют динамическое распределение ключей по открытым каналам связи между привилегированными группами пользователей. При этом ключи должны быть различными для каждой группы.

Тривиальный пример распределения ключей для проведения конференц-связи дает использование централизованного распределения ключей с помощью одного из трехсторонних протоколов передачи ключей, используемых для симметричных шифрсистем. Для реализации такого подхода нужно выделить одного из пользователей группы и возложить на него функции центра генерации и распределения ключей. Естественно, что при этом возрастают требования к доверенности и безопасности выделенного пользователя, что вносит серьезную асимметрию между участниками конференц-связи.

Другой подход основан на использовании идеи открытого распределения ключей.

Приведем примеры протоколов, в которых все участники группы имеют одинаковые полномочия и выполняют симметричные функции.

Простейший пример такого протокола для группы из трех участников можно получить, слегка модифицировав протокол открытого распределения ключей Диффи — Хеллмана. Участники протокола заранее договариваются о значениях большого простого числа p и образующего элемента α мультипликативной группы $Z_p^* = \{1, 2, \dots, p - 1\}$. Для выработки общего ключа k пользователи A , B и C должны сгенерировать соответственно случайные числа x , y и z , $1 \leq x, y, z \leq p - 2$. Затем они должны обмениваться сообщениями согласно следующему протоколу:

- (1) $A \rightarrow B: X = \alpha^x \bmod p,$
- (2) $B \rightarrow C: Y = \alpha^y \bmod p,$
- (3) $C \rightarrow A: Z = \alpha^z \bmod p,$
- (4) $A \rightarrow B: Z' = Z^x \bmod p,$
- (5) $B \rightarrow C: X' = X^y \bmod p,$
- (6) $C \rightarrow A: Y' = Y^z \bmod p.$

Искомый общий ключ $k = \alpha^{xyz} \bmod p$ теперь вычисляется пользователями A, B и C по формулам:

$$\begin{aligned} k &= (Y')^x \bmod p, \\ k &= (Z')^y \bmod p, \\ k &= (X')^z \bmod p \end{aligned}$$

соответственно.

Рассмотрим теперь протокол формирования общего ключа для конференц-связи группы из t пользователей U_0, \dots, U_{t-1} , предложенный в статье [Brm90]. Как и в предыдущем протоколе, каждый пользователь U_i должен сгенерировать секретное случайное число r_i , $1 \leq r_i \leq p-2$, и вычислить открытую экспоненту $z_i = \alpha^{r_i} \bmod p$. Положим

$$A_i = \alpha^{r_i r_{i+1}} = (z_{i+1})^{r_i}.$$

Тогда общий ключ k имеет вид

$$k = \alpha^{r_0 r_1 + r_1 r_2 + \dots + r_{t-1} r_0} \bmod p = A_0 A_1 \dots A_{t-1} \bmod p.$$

Протокол состоит из следующих шагов:

- (1) каждый пользователь U_i рассылает z_i остальным $t-1$ пользователям;
- (2) каждый пользователь U_i вычисляет значение $X_i = (z_{i+1} / z_{i-1})^{r_i} \bmod p$ и рассылает его остальным $t-1$ пользователям;
- (3) каждый пользователь U_i вычисляет значение общего ключа k по формуле

$$k_i = (z_{i-1})^{tr_i} \cdot X_i^{t-1} \cdot X_{i+1}^{t-2} \cdot \dots \cdot X_{i+t-3}^2 \cdot X_{i+t-2}^1 \bmod p.$$

Покажем, что данное значение является искомым ключом. В самом деле,

$$\begin{aligned} & A_{i-1}^{t-1} \cdot X_i^{t-1} \cdot X_{i+1}^{t-2} \cdot \dots \cdot X_{i+t-3}^2 \cdot X_{i+t-2}^1 = \\ &= A_{i-1} \cdot (A_{i-1} X_i) \cdot (A_{i-1} X_i X_{i+1}) \cdot \dots \cdot (A_{i-1} X_i X_{i+1} \cdot X_{i+t-2}) = \\ &= A_{i-1} A_i A_{i+1} \dots A_{i-2} = A_0 A_1 A_2 \dots A_{i-1}. \end{aligned}$$

Поэтому $k_i = k$.

Протокол требует передачи $2t(t-1)$ сообщений, причем каждый пользователь должен отправлять сообщения всем остальным. Можно модифицировать протокол для случая обмена сообщениями по схеме двунаправленного кольца.

Рассмотренный протокол не решает задачи аутентификации, поскольку в нем не заложено процедур для взаимной аутентификации сторон.

§ 15.6. Возможные атаки на протоколы распределения ключей

При анализе протоколов обычно рассматривают несколько видов атак. Одну из них мы уже упоминали при рассмот-

рении протокола Диффи — Хеллмана. Она называется “*злоумышленник в середине*” и заключается в полной подмене всех сообщений между сторонами. Для защиты от нее необходимо дополнить протокол средствами взаимной аутентификации сторон. Это могут быть либо дополнительные, либо встроенные процедуры взаимной аутентификации. Использование дополнительных процедур в протоколе не всегда удобно, так как злоумышленник может подменять не все сообщения, а только относящиеся к выработке ключа. Более предпочтительно, чтобы аутентификация была заложена в сами процедуры выработки ключа, и в случае активного вмешательства протокол заведомо давал бы различные значения ключа сторонам и злоумышленнику.

Еще один тип атак связан с повтором или обратным отражением ранее переданных сообщений — “атака отражением”. Для защиты от таких атак протоколы специально делают несимметричными, включая в зашифрованные сообщения идентификаторы сторон либо изменяя процедуры так, чтобы стороны должны были выполнять разные действия.

Возможны также атаки, при которых нарушитель, выступая от имени одной из сторон и полностью имитируя ее действия, получает в ответ сообщения определенного формата, необходимые для подделки отдельных шагов протокола. В данном случае успех атаки определяется тем, насколько протокол устойчив к подобным подменам. Поэтому для защиты от таких атак используют различные форматы сообщений, передаваемых на разных шагах протокола, а также вставляют в них специальные идентификационные метки и номера сообщений.

Наконец, в протоколах с использованием третьей стороны возможны атаки, основанные на подмене доверенного сервера. Например, одна из сторон, имеющая доверительные отношения с сервером, выступает от его имени, подменяет его трафик обмена с другими сторонами и в результате получает возможность раскрывать значения генерируемых центром ключей. Эта атака может быть успешной для протоколов, в

которых аутентификация при доступе к серверу основана только на идентификаторах сторон и случайных числах, генерируемых при каждом взаимодействии. Для защиты от таких атак применяют средства привязки ключей не к одной, а к обеим взаимодействующим сторонам путем передачи обоих идентификаторов в зашифрованном виде.

В целом при анализе протоколов распределения ключей применяют различные методы. Это и эвристические подходы, основанные на имеющемся наборе практических приемов и известных атаках, зарекомендовавших себя при анализе других протоколов. Обоснование стойкости протоколов часто проводится путем доказательства различных результатов о *сведении* задачи вскрытия протокола к известным труднорешаемым математическим проблемам либо путем *оценки сложности* задач в рамках некоторой модели вычислений. Многие свойства протоколов можно сформулировать с помощью *теоретико-информационных* понятий, связанных с оценкой количества информации, которая становится известной в результате анализа передаваемых сообщений. Наконец, в [Bur90] предложен формальный метод анализа протоколов, получивший название *ВАН-логики*. В нем все шаги протокола и передаваемые сообщения записываются в некотором стандартном формализованном виде, а затем осуществляется их формальный анализ при некоторых типовых предположениях о возможном поведении сторон. Данные формальные доказательства относятся в основном к корректности протоколов и не всегда могут рассматриваться как доказательство их безопасности.

Контрольные вопросы

1. Каковы преимущества централизованного распределения ключей?
2. Какие шифры нельзя использовать в протоколе Шамира?
3. Каков недостаток протокола Нидхэма — Шредера?

4. С какой целью вводится второй сервер в протоколе Kerberos?
5. Как можно использовать цифровую подпись для защиты протоколов передачи ключей?
6. Каковы назначение и структура сертификата открытого ключа?
7. Каков основной недостаток протокола распределения ключей Диффи — Хеллмана и каковы пути его устранения?
8. Что такое схема разделения секрета?
9. В чем общность и в чем отличия схемы разделения секрета и способов распределения ключей для протокола конференц-связи?
10. Перечислите основные атаки на протоколы распределения ключей.

Глава 16

Управление ключами

Управление ключами состоит из процедур, обеспечивающих:

- включение пользователей в систему;
- выработку, распределение и введение в аппаратуру ключей;
- контроль использования ключей;
- смену и уничтожение ключей;
- архивирование, хранение и восстановление ключей.

Управление ключами играет важнейшую роль в криптографии как основа для обеспечения конфиденциальности обмена информацией, идентификации и целостности данных. Важным свойством хорошо спроектированной системы управления ключами является сведение сложных проблем обеспечения безопасности многочисленных ключей к проблеме обеспечения безопасности нескольких ключей, которая может быть относительно просто решена путем обеспечения их физической изоляции в выделенных помещениях и защищенном от проникновения оборудовании. В случае использования ключей для обеспечения безопасности хранимой информации субъектом может быть единственный пользователь, который осуществляет работу с данными в последовательные промежутки времени. Управление ключами в сетях связи включает, по крайней мере, двух субъектов — отправителя и получателя сообщения.

Целью управления ключами является нейтрализация таких угроз, как:

- 1) компрометация конфиденциальности секретных ключей;

2) компрометация аутентичности секретных или открытых ключей. При этом под аутентичностью понимается знание или возможность проверки идентичности корреспондента, для обеспечения конфиденциальной связи с которым используется данный ключ;

3) несанкционированное использование секретных или открытых ключей, например использование ключа, срок действия которого истек.

Управление ключами обычно осуществляется в контексте определенной политики безопасности. Политика безопасности прямо или косвенно определяет те угрозы, которым должна противостоять система. Кроме того, она определяет:

— правила и процедуры, которыми необходимо руководствоваться и которые необходимо выполнять в процессе автоматического или ручного управления ключами,

— ответственность и подотчетность всех субъектов, участвующих в управлении, а также

— все виды записей, которые должны сохраняться для подготовки необходимых сообщений и проведения проверки действий, связанных с безопасностью ключей.

Одним из инструментов, используемых для обеспечения конфиденциальности ключей, является разделение ключей по уровням следующим образом.

1. *Главный ключ* — высший ключ в иерархии, который не защищается криптографически. Его защита осуществляется с помощью физических или электронных средств.

2. *Ключи для шифрования ключей* — секретные или открытые ключи, используемые для засекречивания перед передачей или при хранении других шифровальных ключей. Эти ключи сами могут быть зашифрованы с помощью других ключей.

3. *Ключи для шифрования данных* — используются для защиты данных пользователей.

Ключи более высоких уровней используются для защиты ключей или данных на более низких уровнях, что уменьшает

ущерб при компрометации ключей и объем ключевой информации, нуждающейся в физической защите.

Одной из важных характеристик системы управления ключами являются сроки действия ключей. Под сроком действия ключа понимается интервал времени, в течение которого он может быть использован законным пользователем.

Сокращение сроков действия ключей необходимо для достижения следующих целей:

- ограничения объема информации, зашифрованной на данном ключе, которая может быть использована для криптоанализа;

- ограничения размера ущерба при компрометации ключей;

- ограничения объема машинного времени, которое может быть использовано для криптоанализа (в случае, если нет требований по длительному сохранению ключа и информации, защищенной на этом ключе).

С учетом срока действия ключей, в дополнение к указанной выше классификации ключей по уровням, может быть введена также следующая классификация.

1. Ключи с длительным сроком действия. К ним относятся главный ключ, часто — ключи для шифрования ключей.

2. Ключи с коротким сроком действия. К ним относятся ключи для шифрования данных.

Как правило, в телекоммуникационных приложениях используются ключи с коротким сроком действия, а для защиты хранимых данных — с длительным сроком действия.

Следует иметь в виду, что термин “короткий срок действия” относится только к сроку действия ключа, а не к промежутку времени, в течение которого ключ должен оставаться в секрете. Например, к ключу, используемому для шифрования в течение только одного сеанса связи, часто предъявляется требование, чтобы зашифрованная на нем информация не могла быть вскрыта на протяжении нескольких десятков лет. В то же время электронная подпись проверяется немедленно

после передачи сообщения, поэтому ключ подписи должен сохраняться в тайне в течение достаточно короткого срока.

§ 16.1. Жизненный цикл ключей

Ключевая информация должна быть сменена до момента истечения срока действия ключа. Для этого может быть использована действующая ключевая информация, протоколы распределения ключей и ключевые уровни (см. выше).

Для того чтобы ограничить ущерб от компрометации ключей, по возможности следует избегать зависимостей между действующей и устанавливаемой ключевой информацией. Например, не рекомендуется защищать очередной сеансовый ключ с помощью действующего сеансового ключа. При хранении секретных ключей должны быть приняты меры по обеспечению их конфиденциальности и аутентичности. При хранении открытых ключей должны быть приняты меры, позволяющие проверить их аутентичность. Конфиденциальность и аутентичность могут быть обеспечены криптографическими, организационными и техническими мерами.

В зависимости от конкретных приложений могут выдвигаться различные требования к необходимости и длительности хранения используемых ключей. Так, например, открытые ключи, используемые для проверки цифровой подписи, в ряде случаев необходимо хранить длительное время, даже возможно дольше, чем соответствующие секретные ключи, чтобы была возможность проверки отдельных подписей. В то же время во многих приложениях секретные ключи цифровой подписи не следует хранить длительное время, а тем более архивировать их, поскольку попадание их к посторонним лицам может повлечь отказ от подписанного документа. Проблема отсутствия секретного ключа цифровой подписи из-за его преждевременного уничтожения (без компрометации) сравнительно легко может быть решена путем генерации нового ключа, поскольку уничтоженный ключ не требуется для

проверки ранее произведенных подписей. Аналогично этому открытые ключи, используемые для засекречивания информации, не нуждаются в архивировании. С другой стороны, секретные ключи, используемые для расшифрования, должны определенное время храниться, поскольку в противном случае засекреченная информация будет утрачена.

Все криптосистемы, за исключением простейших, в которых используемые ключи зафиксированы раз и навсегда, нуждаются в периодической замене ключей. Эта замена проводится с помощью определенных процедур и протоколов, в ряде которых используются и протоколы взаимодействия с третьей стороной. Последовательность стадий, которые проходят ключи от момента установления до следующей замены, называется *жизненным циклом ключей* и приведена ниже.

1. *Регистрация пользователей.* Эта стадия включает обмен первоначальной ключевой информацией, такой, как общие пароли или PIN-коды, путем личного общения или пересылки через доверенного курьера.
2. *Инициализация.* На этой стадии пользователь устанавливает аппаратное оборудование и/или программные средства в соответствии с установленными рекомендациями и правилами.
3. *Генерация ключей.* При генерации ключей должны быть приняты меры по обеспечению их необходимых криптографических качеств. Ключи могут генерироваться как самостоятельно пользователем, так и специальным защищенным элементом системы, а затем передаваться пользователю по защищенному каналу.
4. *Установка ключей.* Ключи устанавливаются в оборудование тем или иным способом. При этом первоначальная ключевая информация, полученная на стадии регистрации пользователей, может либо непосредственно вводиться в оборудование, либо использоваться для установления защищенного канала, по которому передается ключевая ин-

формация. Эта же стадия используется в последующем для смены ключевой информации.

5. *Регистрация ключей.* Ключевая информация связывается регистрационным центром с именем пользователя и сообщается другим пользователям ключевой сети. При этом для открытых ключей создаются сертификационным центром ключевые сертификаты, и эта информация публикуется тем или иным способом.
6. *Обычный режим работы.* На этой стадии ключи используются для защиты информации в обычном режиме.
7. *Хранение ключа.* Эта стадия включает процедуры, необходимые для хранения ключа в надлежащих условиях, обеспечивающих его безопасность до момента его замены.
8. *Замена ключа.* Замена ключа осуществляется до истечения его срока действия и включает процедуры, связанные с генерацией ключей, протоколами обмена ключевой информацией между корреспондентами, а также с доверенной третьей стороной. Для открытых ключей эта стадия обычно включает обмен информацией по защищенному каналу с сертификационным центром.
9. *Архивирование.* В отдельных случаях ключевая информация после ее использования для защиты информации может быть подвергнута архивированию для ее извлечения со специальными целями (например, рассмотрения вопросов, связанных с отказами от цифровой подписи).
10. *Уничтожение ключей.* После окончания сроков действия ключей они выводятся из обращения, и все имеющиеся их копии уничтожаются. При этом необходимо следить, чтобы в случае уничтожения секретных ключей тщательно уничтожалась и вся информация, по которой возможно их частичное восстановление.
11. *Восстановление ключей.* Если ключевая информация уничтожена, но не скомпрометирована (например, из-за неисправности оборудования или из-за того, что оператор

забыл пароль) должны быть предусмотрены меры, дающие возможность восстановить ключ из хранимой в соответствующих условиях его копии.

12. *Отмена ключей.* В случае компрометации ключевой информации возникает необходимость прекращения использования ключей до окончания срока их действия. При этом должны быть предусмотрены необходимые меры оповещения абонентов сети. При отмене открытых ключей, снабженных сертификатами, одновременно производится прекращение действия сертификатов.

§ 16.2. Услуги, предоставляемые доверенной третьей стороной

В жизненном цикле управления ключами важную роль играет так называемая доверенная третья сторона, на функциях которой мы ниже остановимся подробнее. Эти функции могут быть определены следующим образом.

1. *Сервер имен абонентов* — обеспечивает придание каждому из абонентов индивидуального имени.
2. *Регистрационный центр* — обеспечивает включение каждого из абонентов в данную сеть засекреченной связи и выдачу ему соответствующей ключевой информации.
3. *Центр производства ключей.*
4. *Ключевой (идентификационный) сервер* — обеспечивает установку общего сеансового ключа между двумя абонентами путем передачи этого ключа по защищенному каналу, образуемому сервером с каждым из абонентов. При этом может осуществляться и идентификация абонентов.
5. *Центр управления ключами* — обеспечивает хранение, архивацию, замену и отмену ключей, а также аудит действий, связанных с жизненным циклом ключей.
6. *Сертификационный центр* — обеспечивает аутентичность открытых ключей путем придания им сертификатов, заверенных цифровой подписью.

7. *Центр установки временных меток* — обеспечивает привязку временной метки к электронному сообщению или транзакции, заверяя тем самым их наличие в определенный момент времени.
8. *Центр нотариализации* — обеспечивает невозможность отказа от сделанного в определенный момент заявления, зафиксированного в электронной форме.
Остановимся подробнее на последних двух функциях.

Установка временных меток

Центр установки временных меток обеспечивает пользователей заверенными временными метками (в момент представления этому центру документа в электронной форме), которые могут служить подтверждением наличия документа в определенный момент времени. Такие метки могут быть использованы для проверки наличия подписанных контрактов в определенный момент времени или подтверждения наличия лабораторных записей в делах, связанных с патентами, а также для обеспечения невозможности отказа от сделанной ранее электронной подписи.

Основная идея такого подхода состоит в следующем. Доверенная третья сторона T добавляет временную метку t_1 к представленному электронному документу или файлу данных, подписывает составленный таким образом документ (тем самым ручаясь за то, что он существовал в определенное временной меткой время) и возвращает подписанный документ вместе с меткой t_1 абоненту, представившему первоначальный документ. Последующая проверка состоит в проверке правильности подписи третьей стороны T .

Если в качестве документа представлять доверенной третьей стороне T не сам документ, а значение вычисленной от него хэш-функции, то будет обеспечена конфиденциальность этого документа при передаче по незащищенным каналам связи, а также снизятся требования к пропускной способ-

ности используемых каналов и объему памяти, необходимому для хранения подписанных документов.

Нотаризация цифровых подписей

Термин “нотаризация” происходит от слова нотариус, который, как известно, должен подтверждать правильность копий представленных документов для представления их третьим лицам — арбитрам.

Центр нотаризации служит для предотвращения опасности отказа подписавшего документ от своей цифровой подписи в последующем, при изменении обстоятельств. При использовании нотаризационного центра предполагается наличие третьего лица — арбитра, который не входит в центр нотаризации, но доверяет нотаризационному центру.

Заметим, что имеется существенная разница между следующими двумя внешне сходными задачами.

1. Сторона A убеждается, что цифровая подпись S была сделана в момент времени t_0 .

2. Сторона A пытается убедить других в момент времени $t_1 \geq t_0$, что подпись S была действительна в момент времени t_0 .

Первая задача решается стандартным способом с использованием симметричной шифрсистемы в отсутствие не доверяющих друг другу сторон.

Особенностью второй задачи как раз и является наличие не доверяющих друг другу сторон. При этом, поскольку цифровой подписи можно доверять только при условии неразглашения секретного ключа, возникает опасность, что подписавший сообщение может намеренно разгласить свой секретный ключ, а затем объявить цифровую подпись поддельной.

Для предотвращения этой угрозы используется центр нотаризации, который функционирует следующим образом. Абонент, получивший подпись на документе (или в значении вычисленной от него хэш-функции), представляет его в центр нотаризации. Центр нотаризации проверяет подпись, готовит

составной документ, состоящий из исходного документа, информации, подтверждающей правильность его цифровой подписи, и временной метки. Затем подписывает этот составной документ своей цифровой подписью. Через определенный промежуток времени, который может быть определен для оповещения в случае компрометации секретных ключей, документ, подписанный нотариационным центром, должен признаваться истинным всеми сторонами, доверяющими данному центру, даже в случае объявления о компрометации секретных ключей.

Контрольные вопросы

1. В чем состоят цели управления ключами?
2. В чем состоит политика безопасности?
3. Как могут быть классифицированы ключи в зависимости от предназначения и сроков их действия?
4. Какие стадии включает в себя жизненный цикл ключей?
5. Какие услуги могут быть предоставлены доверенной третьей стороной?

Глава 17

Некоторые практические аспекты использования шифрсистем

§ 17.1. Анализ потока сообщений

В ряде случаев, когда противник не может дешифровать информацию, передаваемую между абонентами системы шифрованной связи, он тем не менее в состоянии извлечь полезную информацию из анализа интенсивности потока зашифрованных сообщений между различными абонентами этой системы.

Так, например, усиление потока сообщений в военной системе связи может служить одним из признаков подготовки наступательной операции, внезапное увеличение потока сообщений между двумя компаниями является признаком появления объекта, являющегося предметом их общего интереса и т. д.

В некоторых практических ситуациях возникает необходимость скрытия потока сообщений между абонентами системы. Такое скрытие может быть осуществлено либо путем сокращения (и, в идеале, прекращения передачи информации) по каналам, доступным для перехвата противником, либо путем добавления сообщений, не имеющих какого-либо смысла (пустые сообщения). Идеальным же средством скрытия потока сообщений является использование непрерывного линейного шифрования канала связи между каждой парой абонентов сети. Вместе с тем эта мера приводит к значительному удорожанию связи, поскольку требует круглосуточного использования каналов.

§ 17.2. Ошибки операторов

Практика криптоанализа свидетельствует о том, что многие системы шифрованной связи, обеспечивающие надежную защиту информации при исправной работе, не смогли противостоять усилиям криптоаналитиков из-за ошибок шифровальщиков или неисправностей оборудования этих систем.

Простейшим примером ошибки шифровальщика является ситуация, описанная в гл. 6. Приведем еще один пример из этой области. Предположим, что оператор дисковой шифр-машины ошибочно установил угловое положение первого диска, зашифровал телеграмму и отправил ее получателю. Получатель, получив телеграмму и установив правильные угловые положения дисков, естественно, не смог расшифровать эту телеграмму, о чем и послал уведомление отправителю. Если, получив такое уведомление, отправитель установит диски в правильное положение и снова направит зашифрованную телеграмму получателю, то он даст возможность противнику, ведущему перехват, легко прочитать эту телеграмму и, кроме того, установить часть ключа используемой шифр-машины. Противник может распознать такое событие, наблюдая за потоком сообщений, зная шифр-машину и определяя путем анализа те ошибки шифровальщиков, появление которых наиболее вероятно. При этом следует иметь в виду, что слабая обученность шифровальщиков и стрессовая ситуация значительно увеличивают вероятности ошибок.

При эксплуатации ручных и электромеханических систем для избежания таких ошибок предусматривались методы самопроверки шифровальщика. Так, он должен был зашифровать (но не передавать) некоторое тестовое сообщение (например, состоящее из определенного числа символов) после установки ключа и перед началом передачи реального сообщения. Полученный результат сравнивался с правильно зашифрованной версией того же сообщения, рассылаемой вместе с ключом.

В современных электронных шифраторах подобные процедуры проверки проводятся автоматически, так же как и блокировки всех опасных с криптографической точки зрения ошибочных действий шифровальщика.

§ 17.3. Физические и организационные меры при использовании шифрсистем

Для того чтобы криптографические средства защиты обеспечивали надежную защиту информации, необходимо, помимо их высокой криптографической стойкости, позаботиться о том, чтобы противник не мог обойти эти средства путем анализа физических процессов, сопровождающих их работу, или получения шифровальных ключей через операторов, обслуживающих эти устройства.

Выходной сигнал электронного шифратора гаммирования представляет собой последовательность импульсов двух видов, один из которых соответствует нулю, а другой — единице. В силу особенностей реализации электронных схем импульсы, соответствующие одному и тому же знаку, обычно несколько отличаются друг от друга. Эти отличия не должны превосходить определенных пределов для того, чтобы другие части системы могли их распознать. Однако для шифраторов одного этого требования мало. Например, при реализации схемы сложения по модулю 2 может сложиться ситуация, когда из-за несовершенства электронной схемы противник сможет различать четыре типа сигналов, соответствующих результатам операций

$$\begin{array}{ll} (0 + 0) \bmod 2, & (0 + 1) \bmod 2, \\ (1 + 0) \bmod 2, & (1 + 1) \bmod 2. \end{array}$$

Хотя в результате операции $(0 + 1) \bmod 2$ должен появляться такой же сигнал, как и в результате операции $(1 + 0) \bmod 2$,

различия в форме этих сигналов могут привести к тому, что противник сможет дешифровать систему, даже в случае использования одноразовой гаммы. Поэтому к электронным шифраторам должно предъявляться требование, согласно которому импульсы, отображающие одинаковые величины, были бы настолько одинаковыми, насколько это возможно.

Работа криптографической аппаратуры, как и работа других электронных устройств, сопровождается электромагнитным излучением. Это излучение может быть перехвачено и подвергнуто анализу с целью получения информации о шифруемом тексте или используемых ключах. Поэтому необходимо принимать меры для того, чтобы в том месте, где может находиться приемная аппаратура противника, уровень сигнала, излучаемого криптографической аппаратурой, не позволял бы противнику извлечь из него полезную информацию. Этого можно добиться путем снижения уровня сигналов, излучаемых аппаратурой, а также экранированием, искусственным зашумлением и т. д. Подобное же внимание необходимо уделять устранению утечки звука, света и других потенциальных носителей информации.

Если к криптографическому оборудованию возможен доступ посторонних лиц, то оно должно помещаться в специальный контейнер, защищающий его от физического воздействия. Такой контейнер должен защищать от тайного проникновения, не препятствуя, однако, охлаждению и обеспечивая необходимый уровень экранирования. Для предотвращения тайного считывания ключа должны быть предусмотрены специальные устройства, устанавливающие ключ в “нулевое” состояние при попытке открыть контейнер или при обнаружении других физических воздействий.

Таким образом, задача создания электронного шифрующего устройства, обеспечивающего надежную защиту информации, является сложной многопрофильной задачей, требующей разносторонних знаний разработчика в области

криптографии, физики, радиотехники, конструирования электронных приборов и т. д.

Криптографическая система не сможет обеспечить надежной защиты информации, если лица, имеющие доступ к засекречиваемой информации или шифровальным ключам, передают эту информацию противнику. Поэтому соблюдение требований режима секретности персоналом шифровальной службы является важнейшим условием надежной защиты информации. При этом должна быть обеспечена регистрация всех действий персонала, связанных с доступом к защищенной информации, и периодически осуществляться проверка правильности осуществления этого доступа.

Контрольные вопросы

1. Какая информация может быть получена из анализа потока сообщений?
2. Какие средства могут быть использованы для скрытия потока сообщений?
3. Какие меры для защиты от ошибок операторов могут применяться при использовании ручных и электронных систем?
4. Какие физические и организационные меры необходимо применять для защиты криптографического оборудования?

Глава 18

Квантово-криптографический протокол открытого распределения ключей

Квантовый канал и его свойства

Применение в системах передачи информации таких элементарных квантовых систем, как поляризованные фотоны, позволяет получить совершенно новый криптографический эффект, который нельзя достичь при использовании обычных средств передачи информации. Используя принцип неопределенности Гейзенберга, можно получить канал связи, в котором невозможно какое бы то ни было прослушивание без наличия нарушений при передаче, которые не были бы обнаружены с очень большой вероятностью. Такой канал позволяет осуществить безопасное распределение ключей между законными пользователями канала связи.

В криптографических исследованиях традиционно считается, что каналы связи доступны злоумышленнику, и он может беспрепятственно осуществлять перехват и копирование информации без обнаружения пользователями факта вмешательства. В противоположность этому, когда передаваемая информация кодируется неортогональными квантовыми состояниями, например одиночными фотонами с направлениями поляризации 0° , 45° , 90° и 135° , получается такой канал связи, что передаваемые по нему данные даже теоретически не могут ни читаться, ни копироваться нарушителем. Нарушитель не может извлечь никакой, даже частичной, информации об этих данных таким способом, который не поддавался

бы контролю, и который не смогли бы обнаружить законные пользователи канала.

Несмотря на то, что направление поляризации является величиной непрерывной, принцип неопределенности Гейзенберга не допускает такого измерения состояния любого одиночного фотона, которое раскрывало бы более одного бита информации (в вероятностном смысле) об угле его поляризации. Например, если луч света с осью поляризации, направленной под углом α , попадает на фильтр, ориентированный под углом β , то все отдельно взятые фотоны ведут себя независимо и совершенно непредсказуемым образом, проходя через такой фильтр с вероятностью $\cos^2(\alpha-\beta)$, и поглощаясь, соответственно, с вероятностью $\sin^2(\alpha-\beta)$.

Детерминировано фотоны ведут себя только тогда, когда направляющие поляризации фотона и фильтра либо параллельны друг другу (тогда все фотоны проходят через фильтр), либо перпендикулярны (в этом случае все фотоны поглощаются). При этом применить клонирование фотона с целью получить ансамбль из одинаково поляризованных фотонов так, чтобы впоследствии можно было выполнить над ними различные измерения, невозможно. Существование такого ансамбля не согласуется с основными положениями квантовой механики.

С практической же точки зрения достаточно понимать, что имеются два простых прибора. Один из этих приборов может отличать горизонтально поляризованные фотоны от вертикально поляризованных фотонов, а другой может отличать фотоны с разной диагональной поляризацией. Однако если первый прибор используется для определения состояния диагонально поляризованного фотона (а второй — для прямоугольно поляризованного), то в такой ситуации фотон поведет себя совершенно случайным и непредсказуемым образом, и подобное измерение угла его поляризации с одинаковой вероятностью может дать любое из двух возможных значений.

Протокол открытого распределения ключей

Цель открытого распределения ключей по квантовому каналу заключается в том, чтобы, используя квантовый канал, обеспечить передачу последовательности случайных битов между двумя пользователями, которые до этого не имели никакой совместно используемой секретной информации. Если квантовая передача не нарушалась, то пользователи могут с уверенностью применять эту согласованную секретную последовательность в качестве секретного ключа в любой традиционной криптосистеме. С другой стороны, если обнаружится, что передача была нарушена, то пользователи могут не принимать во внимание полученную двоичную последовательность и должны попытаться произвести квантовую передачу еще раз.

Рассмотрим более подробно, каким образом два пользователя (обозначим их A и B) могут осуществить открытое распределение ключей с использованием квантового канала. Мы предполагаем наличие нарушителя C .

В качестве первого шага выбирает произвольную битовую строку и произвольную последовательность поляризационных базисов (прямоугольных и диагональных). Затем A посылает B ряд фотонов, каждый из которых является носителем одного бита информации этой выбранной двоичной строки. Значение каждого бита определяется поляризацией соответствующего ему фотона в базисе, номер позиции которого в выбранной последовательности (поляризационных базисов) совпадает с порядковым номером бита в строке. Так, например, горизонтально или под углом в 45° поляризованные фотоны могут использоваться в качестве носителя двоичного нуля, в то время как вертикально или под углом в 135° поляризованные фотоны будут определять двоичную единицу.

Если C захочет измерить поляризацию тех фотонов, которые A посылает B , то он не будет знать, в каких базисах это

необходимо делать. С другой стороны, B также не знает, какие базисы нужно использовать. Поэтому при получении фотонов пользователь B для каждого фотона случайным образом решает, в каком базисе (прямоугольном или диагональном) проводить измерения. После этого он интерпретирует результат либо как двоичный ноль, либо как двоичную единицу, в зависимости от исхода соответствующего измерения. Получающийся в результате ответ, вообще говоря, носит элемент случайности, так как при попытке измерить в прямоугольном базисе угол поляризации диагонально поляризованного фотона (или наоборот) вся заключенная в нем информация будет потеряна. Таким образом, в итоге B получает корректные данные лишь приблизительно для половины фотонов, которые он измеряет (а именно для тех, для которых он правильно угадал базис поляризации).

Последующие шаги протокола выполняются в обычном открытом канале связи. Будем считать, что этот канал восприимчив только к прослушиванию, но не к введению новых или изменению порядка поступающих сообщений. Сначала A и B определяют посредством открытого обмена сообщениями, какие из фотонов были получены в действительности и какие из них измерялись B в том базисе, в котором их и нужно было измерять. Если квантовая передача не была искажена, то A и B смогут в результате определить биты, закодированные этими фотонами, несмотря на то, что никакая информация о значениях этих битов никогда не передавалась по открытому каналу связи.

Таким образом, порядок открытого распределения ключей по квантовому каналу представляется в виде следующего протокола.

Передача по квантовому каналу

Пользователь A :

1. Случайно выбирает битовую строку.
2. Случайно выбирает последовательность базисов поляризации передаваемых фотонов.

3. Передает поляризованные фотоны пользователю B .
Пользователь B :
4. Случайно выбирает последовательность базисов поляризации фотонов для измерения принимаемых фотонов.
5. Получает последовательность битов при измерении принятых фотонов.

Обсуждение по открытому каналу

- Пользователь B :
6. Сообщает пользователю A базисы измерений полученных фотонов.
Пользователь A :
 7. Отмечает, какие базисы были угаданы правильно.
 8. Посылает пользователю B информацию о номерах битов, которые можно использовать.
Пользователь B :
 9. Указывает пользователю A некоторые, выбранные наугад, биты ключа.
Пользователь A :
 10. Подтверждает эти биты.

Из-за того что прямоугольно и диагонально поляризованные фотоны чередуются в квантовой передаче случайным образом, любой нарушитель рискует при перехвате изменить передачу таким способом, что это приведет к расхождению между A и B . Часть битов, значения которых при отсутствии вмешательства со стороны C должны были совпадать, на самом деле будут отличаться. Отметим, что при пересылке любого фотона никакое измерение его состояния нарушителем, которому исходный базис поляризации этого фотона не известен заранее, не может определить соответствующее значение ключевого бита с вероятностью большей, чем $1/2$.

Остается выяснить, как A и B смогут определить, являются ли получившиеся у них в результате описанного выше протокола битовые строки идентичными. Идентичность битовых

строк показывает с высокой вероятностью, что в квантовом канале нарушения не произошло или что это нарушение имело место на очень малом числе фотонов. Различие строк означает, что квантовый канал подвергнут прослушиванию.

Простое решение этой проблемы заключается в том, что A и B могут открыто сравнить некоторые из битов, значения которых, по их мнению, должны совпадать. Позиции таких “особо проверяемых” битов должны быть выбраны случайно, причем уже после того, как квантовая передача будет завершена. Это лишит противника информации о том, какие фотоны он может измерять без опаски. Недостаток подобного способа заключается в рассекречивании части полученных битов. Если совокупность позиций битов, используемых при этом сравнении, является произвольным подмножеством (скажем, одной трети) всех правильно полученных битов, то перехват более десятка фотонов, позволяющий избежать обнаружения, маловероятен. Если все сравнения подтверждаются, то A и B могут заключить, что квантовая передача прошла без перехвата. Следовательно, оставшиеся биты, которые были посланы и получены в одном и том же базисе, могут использоваться в качестве ключа в системах защиты информации.

Приложение 1

Открытые сообщения и их характеристики

Как уже отмечалось во введении, криптография занимается защитой сообщений, содержащихся на некотором материальном носителе. При этом сами сообщения представляют собой последовательности знаков (или *слова*) некоторого *алфавита*. Различают *естественные* алфавиты, например русский или английский, и *специальные* алфавиты (цифровые, буквенно-цифровые), например двоичный алфавит, состоящий из символов 0 и 1. В свою очередь, естественные алфавиты также могут отличаться друг от друга даже для данного языка.

Алфавиты открытых сообщений

Наиболее привычны буквенные алфавиты, например русский, английский и т. д. Приведем сведения об алфавитах некоторых естественных европейских языков.

Полный русский алфавит состоит из 33 букв:

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

Вместе с тем используются и сокращенные русские алфавиты, содержащие 32, 31 или 30 букв. Можно отождествить буквы Е и Ё, И и Й, Ь и Ъ. Часто бывает удобно включить в алфавит знак пробела между словами, в качестве которого можно взять, например, символ “-”.

Английский алфавит состоит из 26 букв:

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

Иногда используется сокращенный 25-буквенный алфавит, в котором отождествлены буквы I и J.

Во французском языке практически не используются буквы K и W. Эти буквы встречаются только в некоторых словах иностранного происхождения, например *tramway*, *kilo*, *wagon*, *weekend*. Поэтому часто используют 24-буквенный алфавит (без K и W).

В немецком языке исключительно редки буквы Q, X и Y. Буква Q появляется в виде начальной буквы лишь в некоторых малоупотребительных словах, большей частью иностранного происхождения, например *quelle*, *quarta*, *quiting*. Помимо латинских букв немецкий язык использует еще три буквы: ö, ä, ё, которые часто заменяют эквивалентами OE, AE и UE соответственно, например в словах *kaempfen*, *moebel*, *glueck*. Буквы X и Y практически не используются, поэтому часто обходятся лишь 24 буквами (без X и Y).

Испанский язык содержит некоторые особые буквы: CH, LL, Ñ, которые можно заменить эквивалентами: C,H; L,L; N. Как и во французском языке, буквы K и W исключительно редки, и поэтому часто используется 24-буквенный алфавит.

В итальянском языке крайне редки буквы J, K, W, X и Y. Поэтому используется либо 21-буквенный алфавит, либо 22-буквенный (без J, K, W, Y).

Реже встречается греческий алфавит. Он состоит из 24 букв (см. табл. 1).

Любопытно отметить, что полинезийский язык Самоа имеет алфавит, содержащий всего 16 букв, из которых около 60% — гласных. В арабском языке и иврите согласные вообще не используются. Они опускаются в письменном тексте и восстанавливаются читателем по смыслу.

Таблица 1. Греческий алфавит

A(α)	B(β)	Г(γ)	Δ(δ)	Ε(ϵ)	Z(ζ)
альфа	бетта	гамма	дельта	эпсилон	дзета
H(η)	Θ(θ)	I(ι)	K(κ)	Λ(λ)	M(μ)
эта	тэта	йота	каппа	лямбда	мю
N(ν)	Ξ(ξ)	O(\omicron)	Π(π)	P(ρ)	Σ(σ)
ню	кси	омикрон	пи	ро	сигма
T(τ)	Υ(υ)	Φ(ϕ)	X(χ)	Ψ(ψ)	Ω(ω)
тау	эпсилон	фи	хи	пси	омега

Код Бодо, применяемый для передачи сообщений с помощью телетайпов и телексов, использует 32-значный алфавит:

Таблица 2. Упрощенный Бодо-подобный алфавит

00000	00001	00010	00011
A	B	C	D
00100	00101	00110	00111
E	F	G	H
01000	01001	01010	01011
I	J	K	L
01100	01101	01110	01111
M	N	O	P
10000	10001	10010	10011
Q	R	S	T
10100	10101	10110	10111
U	V	W	X
11000	11001	11010	11011
Y	Z		.
11100	11101	11110	11111
?	+	()

Первым идею двоичного кодирования букв алфавита использовал современник В. Шекспира Фрэнсис Бэкон. Он предложил двухбуквенное кодирование.

Таблица 3. Двухбуквенный алфавит Ф. Бэкона

A – aaaaa	I, J – abaaa	R – baaaa
B – aaaab	K – abaab	S – baaab
C – aaaba	L – ababa	T – baaba
D – aaabb	M – ababb	U, V – baabb
E – aabaa	N – abbaa	W – babaa
F – aabaa	O – abbab	X – babab
G – aabba	P – abbba	Y – babba
H – aabbb	Q – abbbb	Z – babbb

Цифровое кодирование букв применял упоминаемый в историческом очерке И. Тритемий.

Таблица 4. Трехзначный алфавит Тритемия

A – 111	J – 211	S – 311
B – 112	K – 212	T – 312
C – 113	L – 213	U – 313
D – 121	M – 221	V – 321
E – 122	N – 222	W – 322
F – 123	O – 223	X – 323
G – 131	P – 231	Y – 331
H – 132	Q – 232	Z – 332
I – 133	R – 233	& – 333

В вычислительной технике распространены 128-битовые и 256-битовые алфавиты, использующие представление знаков алфавита в виде 7- или 8-значных двоичных комбинаций.

Наиболее известен код ASCII (American Standart Code for Information Interchange) — американский стандартный код информационного обмена. Приведем фрагмент этого кода.

Таблица 5. Код ASCII

Символ	Восьмеричная запись	Двоичная запись
A	101	01000001
B	102	01000010
C	103	01000011
D	104	01000100
E	105	01000101
F	106	01000110
1	061	00110001
2	062	00110010
3	063	00110011
4	064	00110100
&	046	00100110
.	056	00101110
#	043	00100011
,	054	00101100

В практике передачи сообщений по техническим каналам связи используется множество других кодов, основанных на двоичном кодировании.

Буквенный алфавит, в котором буквы расположены в их естественном порядке, обычно называют *нормальным алфавитом*. В противном случае говорят о *смешанных алфавитах*. В свою очередь, смешанные алфавиты делят на *системати-*

чески перемешанные алфавиты и случайные алфавиты. К первым относят алфавиты, полученные из нормального на основе некоторого правила, ко вторым — алфавиты, буквы которых следуют друг за другом в хаотическом (или случайном) порядке.

Смешанные алфавиты обычно используются в качестве нижней строки подстановки, представляющей собой ключ шифра простой замены (см. гл. 5). Для запоминания ключа (это надежнее, чем хранение ключа на некотором носителе) применяется несложная процедура перемешивания алфавита, например, основанная на *ключевом слове*. Одним из первых такой способ построения систематически перемешанного алфавита предложил Ардженти (см. гл. 1).

Частотные характеристики текстовых сообщений

Криптоанализ любого шифра невозможен без учета особенностей текстов сообщений, подлежащих шифрованию. Глубинные закономерности текстовых сообщений исследуются в теории информации. Наиболее важной для криптографии характеристикой текстов является избыточность текста, введенная К. Шенноном. Именно избыточность открытого текста, проникающая в шифртекст, является основной слабостью шифра.

Более простыми характеристиками текстов, используемыми в криптоанализе, являются такие характеристики, как *повторяемость* букв, пар букв (*биграмм*) и вообще *m*-ок (*m*-грамм), *сочетаемость* букв друг с другом, чередование гласных и согласных и некоторые другие. Такие характеристики изучаются на основе эмпирических наблюдений текстов достаточно большой длины.

Для установления статистических закономерностей проводилась большая серия экспериментов по оценке вероятно-

стей появления в открытом тексте фиксированных m -грамм (для небольших значений m).

Суть экспериментов состоит в подсчете чисел вхождений каждой из n^m возможных m -грамм в достаточно длинных открытых текстах $T = t_1 t_2 \dots t_l$, составленных из букв алфавита $\{a_1, a_2, \dots, a_n\}$. При этом просматриваются подряд идущие m -граммы текста:

$$t_1 t_2 \dots t_m, t_2 t_3 \dots t_{m+1}, \dots, t_{l-m+1} t_{l-m+2} \dots t_l.$$

Если $\mathcal{G}(a_1 a_2 \dots a_m)$ — число появлений m -граммы $a_1 a_2 \dots a_m$ в тексте T , а L — общее число подсчитанных m -грамм, то опыт показывает, что при достаточно больших L частоты

$$\frac{\mathcal{G}(a_1 a_2 \dots a_m)}{L} \quad (1)$$

для данной m -граммы мало отличаются друг от друга. В силу этого относительную частоту (1) считают приближением вероятности $P(a_1 a_2 \dots a_m)$ появления данной m -граммы в случайно выбранном месте текста (такой подход принят при статистическом определении вероятности). Например, при $m = 1$ хорошее приближение вероятностей появления букв достигается на текстах длиной в несколько тысяч букв.

Ниже приводится таблица частот букв (в процентах) ряда европейских языков. Данные заимствованы из книги [Вау39] (см. табл. 6).

Некоторая разница значений частот в приводимых в различных источниках таблицах объясняется тем обстоятельством, что частоты существенно зависят не только от длины текста, но и от его характера. Так, в технических текстах редкая буква Φ может стать довольно частой в связи с частым

использованием таких слов, как *функция, дифференциал, диффузия, коэффициент* и т. п.

Таблица 6. Частоты букв европейских языков

Буква ал- фавита	Франц. язык	Немец. язык	Англ. язык	Испан. язык	Итал. язык
A	7,68	5,52	7,96	12,90	11,12
B	0,80	1,56	1,60	1,03	1,07
C	3,32	2,94	2,84	4,42	4,11
D	3,60	4,91	4,01	4,67	3,54
E	17,76	19,18	12,86	14,15	11,63
F	1,06	1,96	2,62	0,70	1,15
G	1,10	3,60	1,99	1,00	1,73
H	0,64	5,02	5,39	0,91	0,83
I	7,23	8,21	7,77	7,01	12,04
J	0,19	0,16	0,16	0,24	-
K	-	1,33	0,41	-	-
L	5,89	3,48	3,51	5,52	5,95
M	2,72	1,69	2,43	2,55	2,65
N	7,61	10,20	7,51	6,20	7,68
O	5,34	2,14	6,62	8,84	8,92
P	3,24	0,54	1,81	3,26	2,66
Q	1,34	0,01	0,17	1,55	0,48
R	6,81	7,01	6,83	6,95	6,56
S	8,23	7,07	6,62	7,64	4,81
T	7,30	5,86	9,72	4,36	7,07
U	6,05	4,22	2,48	4,00	3,09
V	1,27	0,84	1,15	0,67	1,67
W	-	1,38	1,80	-	-
X	0,54	-	0,17	0,07	-
Y	0,21	-	1,52	1,05	-
Z	0,07	1,17	0,05	0,31	1,24

Еще большие отклонения от нормы в частоте употребления отдельных букв наблюдаются в некоторых художественных произведениях, особенно в стихах. Поэтому для надежного определения средней частоты буквы желательно иметь набор различных текстов, заимствованных из различных источников. Вместе с тем, как правило, подобные отклонения незначительны, и в первом приближении ими можно пренебречь.

В связи с этим подобные таблицы, используемые в криптографии, должны составляться с учетом *характера переписки*.

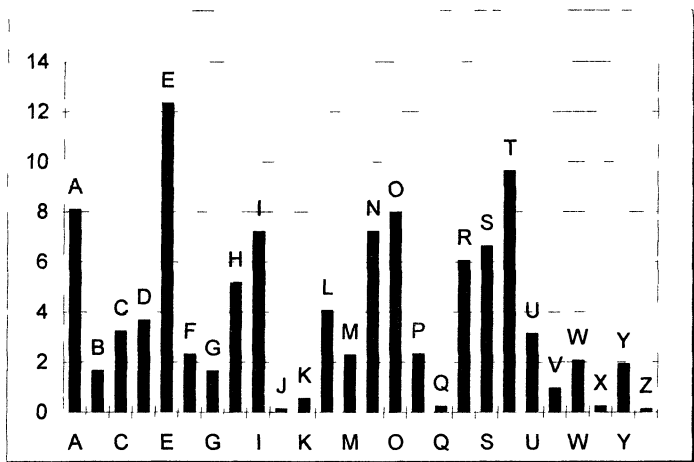


Рис. 1. Частоты букв английского языка (в процентах)

Наглядное представление о частотах букв дает диаграмма встречаемости. Так, для английского языка, в соответствии с табл. 6, такая диаграмма изображена на рис. 1.

Для русского языка частоты (в порядке убывания) знаков алфавита, в котором отождествлены Е с Ё, Ь с Ъ, а также име-

ется знак пробела (-) между словами, приведены в следующей таблице (см. [Ягл.73]):

Таблица 7. Частоты букв русского языка
(в 32-буквенном алфавите со знаком пробела)

-	О	Е,Ё	А
0,175	0,090	0,072	0,062
И	Т	Н	С
0,062	0,053	0,053	0,045
Р	В	Л	К
0,040	0,038	0,035	0,028
М	Д	П	У
0,026	0,025	0,023	0,021
Я	Ы	З	Ь,Ъ
0,018	0,016	0,016	0,014
Б	Г	Ч	Й
0,014	0,013	0,012	0,010
Х	Ж	Ю	Ш
0,009	0,007	0,006	0,006
Ц	Щ	Э	Ф
0,004	0,003	0,003	0,002

Таблица 8. 10 наиболее частых букв европейских языков

Французский язык	E,S,A,N,T,I,R,U,L,O	79,9%
Немецкий язык	E,N,I,S,T,A,H,D,U	77,2%
Английский язык	E,T,A,I,N,R,O,S,H,D	75,3%
Испанский язык	E,A,O,S,I,R,N,L,D,C	78,3%
Итальянский язык	I,E,A,O,N,T,R,L,S,C	79,9%

На основании табл. 7 получаем следующую диаграмму частот (см. рис. 2).

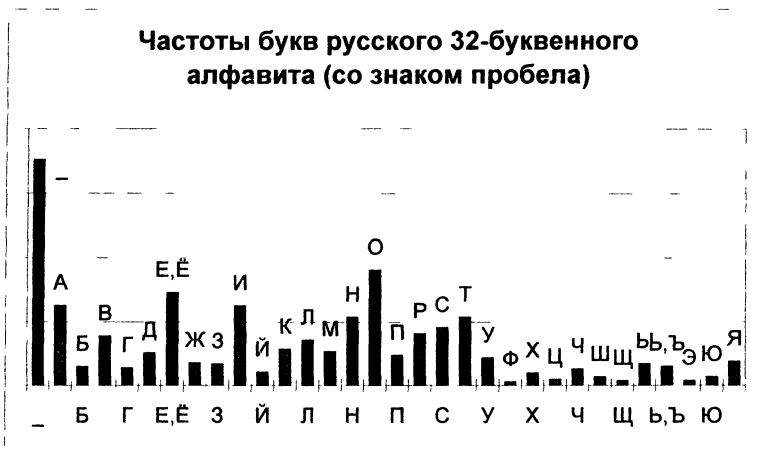


Рис. 2. Диаграмма частот букв русского языка

Имеется мнемоническое правило запоминания десяти наиболее частых букв русского алфавита. Эти буквы составляют нелепое слово СЕНОВАЛИТР. Можно также предложить аналогичный способ запоминания частых букв английского языка, например, с помощью слова TETRIS-HONDA (см. табл. 8).

Устойчивыми являются также частотные характеристики биграмм, триграмм и четырехграмм осмысленных текстов.

Приведем таблицы частот биграмм для русского (табл. 9) и английского языков (табл. 10) (таблицы заимствованы из книги [Fri85]). Для удобства они разбиты на четыре части по следующей схеме:

ЧАСТЬ 1	ЧАСТЬ 2
ЧАСТЬ 3	ЧАСТЬ 4

Хорошие таблицы k -грамм легко получить, используя тексты электронных версий многих книг, содержащихся на CD-дисках.

Для получения более точных сведений об открытых текстах можно строить и анализировать таблицы k -грамм при $k > 2$, однако для учебных целей вполне достаточно ограничиться биграммами. Неравновероятность k -грамм (и даже слов) тесно связана с характерной особенностью открытого текста — наличием в нем большого числа повторений отдельных фрагментов текста: корней, окончаний, суффиксов, слов и фраз. Так, для русского языка такими привычными фрагментами являются наиболее частые биграммы и триграммы:

СТ, НО, ЕН, ТО, НА, ОВ, НИ, РА, ВО, КО,
СТО, ЕНО, НОВ, ТОВ, ОВО, ОВА

Полезной является информация о сочетаемости букв, то есть о предпочтительных связях букв друг с другом, которую легко извлечь из таблиц частот биграмм.

Имеется в виду таблица, в которой слева и справа от каждой буквы расположены наиболее предпочтительные “соседи” (в порядке убывания частоты соответствующих биграмм). В таких таблицах обычно указывается также доля гласных и согласных букв (в процентах), предшествующих (или следующих за) данной букве.

Таблица 9. Таблица частот биграмм русского языка

ЧАСТЬ 1

	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
А	2	12	35	8	14	7	6	15	7	7	19	27	19	45	5	11
Б	5					9	1		6			6		2	21	
В	35	1	5	3	3	32		2	17		7	10	3	9	58	6
Г	7				3	3			5		1	5		1	50	
Д	25		3	1	1	29	1	1	13		1	5	1	13	22	3
Е	2	9	18	11	27	7	5	10	6	15	13	35	24	63	7	16
Ж	5	1			6	12			5					6		
З	35	1	7	1	5	3			4		2	1	2	9	9	1
И	4	6	22	5	10	21	2	23	19	11	19	21	20	32	8	13
Й	1	1	4	1	3		1	2	4		5	1	2	7	9	7
К	24	1	4	1		4	1	1	26		1	4	1	2	66	2
Л	25	1	1	1	1	33	2	1	36		1	2	1	8	30	2
М	18	2	4	1	1	21	1	2	23		3	1	3	7	19	5
Н	54	1	2	3	3	34			58		3		1	24	67	2
О	1	28	84	32	47	15	7	18	12	29	19	41	38	30	9	18
П	7					15			4			9		1	46	

ЧАСТЬ 2

	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ы	Ь	Э	Ю	Я
А	26	31	27	3	1	10	6	7	10	1			2	6	9
Б	8	1		6						1	11				2
В	6	19	6	7		1	1	2	4	1	18	1	2		3
Г	7			2											
Д	6	8	1	10			1	1	1		5	1			1
Е	39	37	33	3	1	8	3	7	3	3			1	1	2
Ж		1													
З	3	1		2							4				4
И	11	29	29	3	1	17	3	11	1	1			1	3	17
Й	3	10	2				1	3	2						
К	10	3	7	10			1								
Л		3	1	6		4		1			2	30		4	9
М	2	5	3	9	1			2			5	1	1		3
Н	1	9	9	7	1		5	2			36	3			5
О	43	50	39	3	2	5	2	12	4	3			2	3	2
П	41	1		6							2				2

ЧАСТЬ 3

	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
Р	55	1	4	4	3	37	3	1	24		3	1	3	7	56	2
С	8	1	7	1	2	25			6		40	13	3	9	27	11
Т	35	1	27	1	3	31		1	28		5	1	1	11	56	4
У	1	4	4	4	11	2	6	3	2		8	5	5	5	1	5
Ф	2				2				2						1	
Х	4	1	4	1	3	1		2	3		4	3	3	4	18	5
Ц	3				7				10		2				1	
Ч	12				23				13		2			6		
Ш	5				11				14		1	2		2	2	
Щ	3				8				6					1		
Ы		1	9	1	3	12		2	4	7	3	6	6	3	2	10
Ь		2	4	1	1	2		2	2		6		3	13	2	4
Э											1			1		
Ю		2	1	2	1			3	1		1		1	1	1	3
Я	1	3	9	1	3	3	1	5	3	2	3	3	4	6	3	6

ЧАСТЬ 4

	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ы	Ь	Э	Ю	Я
Р	1	5	9	16		1	1	1	2		8	3			5
С	4	11	82	6		1	1	2	2		1	8			17
Т	26	18	2	10				1			11	21			4
У	7	14	7			1		8	3	2				9	1
Ф	1	1													
Х	3	4	2	2	1			1							
Ц				1							1				
Ч			7	1				1				1			
Ш				1								1			
Щ				1											
Ы	3	9	4	1		16		1	2						
Ь	1	11	3					1	4				1	3	1
Э		1	9												
Ю	1	1	7			1	1		4						
Я	3	6	10			2	1	4	1	1			1	1	1

Таблица 10. Таблица частот биграмм английского языка

ЧАСТЬ 1

	A	B	C	D	E	F	G	H	I	J	K	L	M
A	4	20	28	52	2	11	28	4	32	4	6	62	23
B	13	0	0	0	55	0	0	0	8	2	0	22	0
C	32	0	7	1	69	0	0	33	17	0	10	9	1
D	40	16	9	5	65	18	3	9	56	0	1	4	15
E	84	20	55	125	51	40	19	16	50	1	4	55	54
F	19	3	5	1	19	21	1	3	30	2	0	11	1
G	20	4	3	2	35	1	3	15	18	0	0	5	1
H	101	1	3	0	270	5	1	6	57	0	0	0	3
I	40	7	51	23	25	9	11	3	0	0	2	38	25
J	3	0	0	0	5	0	0	0	1	0	0	0	0
K	1	0	0	0	11	0	0	0	13	0	0	0	0
L	44	2	5	12	62	7	5	2	42	1	1	53	2
M	52	14	1	0	64	0	0	3	37	0	0	0	7

ЧАСТЬ 2

	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	167	2	14	0	83	76	127	7	25	8	1	9	1
B	0	11	0	0	15	4	2	13	0	0	0	15	0
C	0	50	3	0	10	0	28	11	0	0	0	3	0
D	6	16	4	0	21	18	53	19	5	15	0	3	0
E	146	35	37	6	191	149	65	9	26	31	12	5	0
F	0	51	0	0	26	8	47	6	3	3	0	2	0
G	4	21	1	1	20	9	21	9	0	5	0	1	0
H	2	44	1	0	3	10	18	6	0	5	0	3	0
I	202	56	12	1	46	79	117	1	22	0	4	0	3
J	0	4	0	0	0	0	0	3	0	0	0	0	0
K	2	2	0	0	0	6	2	1	0	2	0	1	0
L	2	25	1	1	2	16	23	9	0	1	0	33	0
M	1	17	18	1	2	12	3	8	0	1	0	2	0

ЧАСТЬ 3

	A	B	C	D	E	F	G	H	I	J	K	L	M
N	42	10	47	122	63	19	106	12	30	1	6	6	9
O	7	12	14	17	5	95	3	5	14	0	0	19	41
P	19	1	0	0	37	0	0	4	8	0	0	15	1
Q	0	0	0	0	0	0	0	0	0	0	0	0	0
R	83	8	16	23	169	4	8	8	77	1	10	5	26
S	65	9	17	9	73	13	1	47	75	3	0	7	11
T	57	22	7	1	76	5	2	330	126	1	0	14	10
U	11	5	9	6	9	1	6	0	9	0	1	19	5
V	7	0	0	0	72	0	0	0	28	0	0	0	0
W	36	1	1	0	38	0	0	33	36	0	0	4	1
X	1	0	2	0	0	1	0	0	3	0	0	0	0
Y	14	5	4	2	7	12	2	6	10	0	0	3	7
Z	1	0	0	0	4	0	0	0	0	0	0	0	0

ЧАСТЬ 4

	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
N	7	54	7	1	7	44	124	6	1	15	0	12	0
O	134	13	23	0	91	23	42	55	16	28	0	4	1
P	0	27	9	0	33	14	7	6	0	0	0	0	0
Q	0	0	0	0	0	0	0	17	0	0	0	0	0
R	16	60	4	0	24	37	55	6	11	4	0	28	0
S	12	56	17	6	9	48	116	35	1	28	0	4	0
T	6	79	7	0	49	50	56	21	2	27	0	24	0
U	31	1	15	0	47	39	31	0	3	0	0	0	0
V	0	5	0	0	0	0	0	0	0	0	0	3	0
W	8	15	0	0	0	4	2	0	0	1	0	0	0
X	0	1	5	0	0	0	3	0	0	1	0	0	0
Y	5	17	3	0	4	16	30	0	0	5	0	0	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0

Таблица 11. Сочетаемость букв русского языка

Г	С	Слева		Справа	Г	С
3	97	л, д, к, т, в, р, н	А	л, н, с, т, р, в, к, м	12	88
80	20	я, е, у, и, а, о	Б	о, ы, е, а, р, у	81	19
68	32	я, т, а, е, и, о	В	о, а, и, ы, с, н, л, р	60	40
78	22	р, у, а, и, е, о	Г	о, а, р, л, и, в	69	31
72	28	р, я, у, а, и, е, о	Д	е, а, и, о, н, у, р, в	68	32
19	81	м, и, л, д, т, р, н	Е	н, т, р, с, л, в, м, и	12	88
83	17	р, е, и, а, у, о	Ж	е, и, д, а, н	71	29
89	11	о, е, а, и	З	а, н, в, о, м, д	51	49
27	73	р, т, м, и, о, л, н	И	с, н, в, и, е, м, к, з	25	75
55	45	ь, в, е, о, а, и, с	К	о, а, и, р, у, т, л, е	73	27
77	23	г, в, ы, и, е, о, а	Л	и, е, о, а, ь, я, ю, у	75	25
80	20	я, ы, а, и, е, о	М	и, е, о, у, а, н, п, ы	73	27
55	45	д, ь, н, о, а, и, е	Н	о, а, и, е, ы, н, у	80	20
11	89	р, п, к, в, т, н	О	в, с, т, р, и, д, н, м	15	85
65	35	в, с, у, а, и, е, о	П	о, р, е, а, у, и, л	68	32
55	45	и, к, т, а, п, о, е	Р	а, е, о, и, у, я, ы, н	80	20
69	31	с, т, в, а, е, и, о	С	т, к, о, я, е, ь, с, н	32	68
57	43	ч, у, и, а, е, о, с	Т	о, а, е, и, ь, в, р, с	63	37
15	85	п, т, к, д, н, м, р	У	т, п, с, д, н, ю, ж	16	84
70	30	н, а, е, о, и	Ф	и, е, о, а, е, о, а	81	19
90	10	у, е, о, а, ы, и	Х	о, и, с, н, в, п, р	43	57
69	31	е, ю, н, а, и	Ц	и, е, а, ы	93	7
82	18	е, а, у, и, о	Ч	е, и, т, н	66	34
67	33	ь, у, ы, е, о, а, и, в	Ш	е, и, н, а, о, л	68	32
84	16	е, б, а, я, ю	Щ	е, и, а	97	3
0	100	м, р, т, с, б, в, н	Ы	л, х, е, м, и, в, с, н	56	44
0	100	н, с, т, л	Ь	н, к, в, п, с, е, о, и	24	76
14	86	с, ы, м, л, д, т, р, н	Э	н, т, р, с, к	0	100
58	42	ь, о, а, и, л, у	Ю	д, т, щ, ц, н, п	11	89
43	57	о, н, р, л, а, и, с	Я	в, с, т, п, д, к, м, л	16	84

Таблица 12. Сочетаемость букв английского языка

Г	С	Слева		Справа	Г	С
19	81	l,c,d,m,n,s,w,t,r,e,h	A	n,t,s,r,l,d,c,m	6	94
55	45	y,b,n,t,u,d,o,s,a,e	B	e,l,u,o,a,y,b,r	70	30
61	39	u,o,s,n,a,i,l,e	C	h,o,e,a,i,t,r,l,k	59	41
52	48	r,i,l,a,n,e	D	e,i,t,a,o,u	54	46
8	92	c,b,e,m,v,d,s,l,n,t,r,h	E	r,d,s,n,a,t,m,e,c,o	21	79
69	31	s,n,f,d,a,i,e,o	F	t,o,e,i,a,r,f,u	52	48
36	64	o,d,u,r,i,e,a,n	G	e,h,o,r,a,t,f,w,i,s	42	58
7	93	g,e,w,s,c,t	H	e,a,i,o	90	10
13	87	f,m,w,e,n,l,d,s,r,h,t	I	n,t,s,o,c,r,e,m,a,l	17	83
28	72	y,w,t,s,n,e,c,b,a,c	J	u,o,a,e,m,w	88	12
53	47	y,u,i,n,a,r,o,c	K	e,i,n,a,t,s	68	32
52	48	m,p,t,i,b,u,o,e,l,a	L	e,i,y,o,a,d,u	65	35
69	31	s,d,m,r,i,a,o,e	M	e,a,o,i,p,m	71	29
89	11	U,e,o,a,i	N	d,t,g,e,a,s,o,i,c	32	68
21	79	o,d,l,p,h,n,e,c,f,s,i,r,t	O	n,f,r,u,t,m,l,s,w,o	18	82
47	53	r,l,t,n,i,p,m,a,o,u,e,s	P	o,e,a,r,l,u,p,t,i,s	59	41
20	80	o,n,l,e,d,r,s	Q	u	100	0
70	30	p,i,u,t,a,o,e	R	e,o,a,t,i,s,y	61	39
48	52	d,t,o,u,r,n,s,i,a,e	S	t,e,o,i,s,a,h,p,u	41	59
43	57	u,o,d,t,f,e,i,n,s,a	T	h,i,o,e,a,t,r	38	62
35	65	p,f,t,l,b,d,s,o	U	n,s,t,r,l,p,b,c	8	92
88	12	r,u,o,a,i,e	V	e,i,o,a	99	1
48	52	g,d,y,n,s,t,o,e	W	a,h,i,e,o,n	80	20
95	5	u,n,i,e	X	p,t,i,a,u,c,k,o	38	62
24	76	b,n,a,t,e,r,l	Y	a,o,s,t,w,h,i,e,d,m	38	62
88	12	o,n,a,i	Z	e,i,w	86	14

При анализе сочетаемости букв друг с другом следует иметь в виду зависимость появления букв в открытом тексте от значительного числа предшествующих букв. Для анализа этих закономерностей используют понятие *условной вероятности*.

Наблюдения над открытыми текстами показывают, что для условных вероятностей выполняются неравенства

$$p(a_{i_1}) \neq p(a_{i_1}/a_{i_2}), p(a_{i_1}/a_{i_2}) \neq p(a_{i_1}/a_{i_2} a_{i_3}), \dots$$

Систематически вопрос о зависимости букв алфавита в открытом тексте от предыдущих букв исследовался известным русским математиком А. А. Марковым (1856 — 1922). Он доказал, что появления букв в открытом тексте нельзя считать независимыми друг от друга. В связи с этим А. А. Марковым отмечена еще одна устойчивая закономерность открытых текстов, связанная с чередованием гласных и согласных букв. Им были подсчитаны частоты встречаемости биграмм вида гласная-гласная (*г, г*), гласная-согласная (*г, с*), согласная-гласная (*с, г*), согласная-согласная (*с, с*) в русском тексте длиной в 10^5 знаков. Результаты подсчета отражены в следующей таблице:

Таблица 13. *Чередование гласных и согласных*

	Г	С	Всего
Г	6588	38310	44898
С	38296	16806	55102

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

$$p(\Gamma \text{ с}) \approx 0,663, \quad p(\text{с}/\Gamma) \approx 0,872,$$

$$p(\Gamma) \approx 0,432. \quad p(\text{с}) \approx 0,568.$$

После А. А. Маркова зависимость появления букв текста вслед за несколькими предыдущими исследовал методами теории информации К. Шеннон. Фактически им было показано, в частности, что такая зависимость ощутима на глубину приблизительно в 30 знаков, после чего она практически отсутствует.

Таблица 14. Доля гласных букв в литературном тексте

Французский язык	44,27%
Немецкий язык	39,27%
Английский язык	39,21%
Испанский язык	47,95%
Итальянский язык	46,80%

Приведенные выше закономерности имеют место для обычных “читаемых” открытых текстов, используемых при общении людей. Как уже отмечалось ранее, эти закономерности играют большую роль в криптоанализе. В частности, они используются при построении формализованных критериев на открытый текст, позволяющих применять методы математической статистики в задаче распознавания открытого текста в потоке сообщений. При использовании же специальных алфавитов требуются аналогичные исследования частотных характеристик “открытых текстов”, возникающих, например, при межмашинном обмене информацией или в системах передачи данных. В этих случаях построение формализованных критериев на “открытый текст” — задача значительно более сложная.

В качестве примера приведем частотные характеристики букв английского алфавита, входящих в состав кода ASCII.

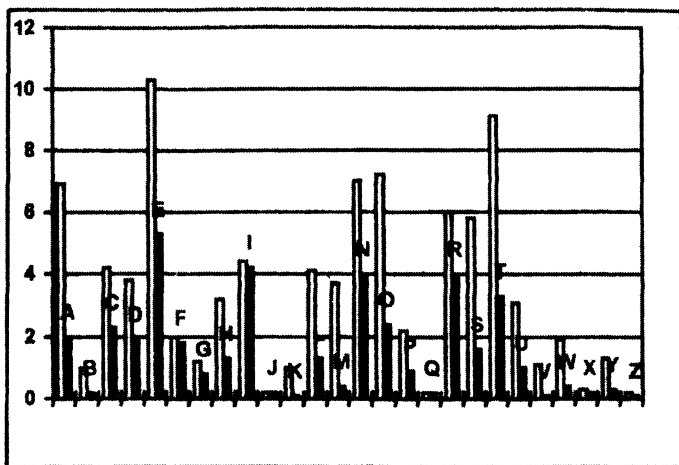


Рис. 3. Частоты символов ASCII (буквы)
 (светлым — в статье по компьютерной тематике;
 темным — в тексте программы на языке Паскаль)

Помимо криптографии частотные характеристики открытых сообщений существенно используются и в других сферах. Например, клавиатура компьютера, пишущей машинки или линотипа — это замечательное воплощение идеи ускорения набора текста, связанное с оптимизацией расположения букв алфавита относительно друг друга в зависимости от частоты их применения.

Приложение 2

Пример

Рассмотрим криптограмму, текст которой расположен в строках табл. 1 (см. стр. 454).

Для успешного решения задачи найдем частотные характеристики криптограммы, построив соответствующую матрицу биграмм (см. табл. 2 на стр. 455). Обратим при этом внимание на то, что в данном тексте используется обычный русский алфавит, состоящий из 30 букв. Поэтому матрица будет иметь размеры 30×30 , строки и столбцы которой занумерованы буквами 30-буквенного алфавита.

В табл. 1 текст криптограммы выписан построчно с интервалом в две строки. Промежуточная строка потребуется далее для записи букв открытого текста под соответствующими буквами шифрованного текста.

Построим диаграмму встречаемости букв криптограммы.

Подсчет дает следующие значения относительных частот букв (в процентах):

А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О
1,80	7,54	6,9	11	2,1	8,9	4,1	6,1	5,1	1,5	3,6	0,2	9,5	2

П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ы	Ь	Э	Ю	Я
0	6,4	3,4	4,6	1,2	1	0	1,5	0,5	3,8	0	1	3,8	0,5	1,3	1

На основании этих данных получаем такую картину (рис. 1)

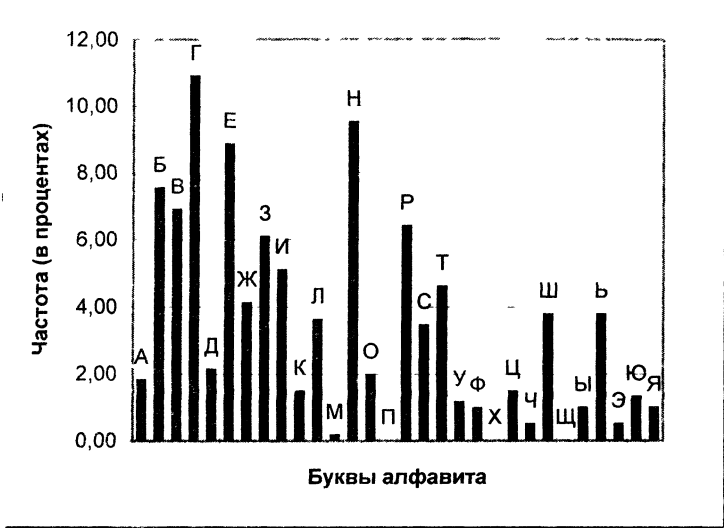


Рис. 1. Диаграмма встречаемости букв криптограммы

Построенная диаграмма очень близка по характеру к диаграмме открытого текста (см. рис. 1 Приложения 1). Кроме того, матрица биграмм является достаточно рельефной. Это свидетельствует о том, что скорее всего использован шифр простой замены. Нам ничто не мешает проверить эту гипотезу.

В матрице биграмм имеется ярко выраженный лидер: биграмма ЗИ встретилась 11 раз. Естественно предположить, что она заменяет биграмму СТ открытого текста (наиболее часто встречающуюся в русском литературном тексте). Явным лидером по частоте встречаемости является буква Г. Предположим, что она заменяет букву О.

Проверим эти гипотезы, сделав соответствующие замены букв криптограммы. В результате получим картину, отраженную в табл. 3 (см. стр. 456).

Обратим внимание на некоторые получившиеся фрагменты открытого текста: О?О?ОТТ и СТ? (15-я строка), ?СТО (6-я строка), СОС? (26-я строка), на основании которых можно выдвинуть гипотезы о том, что вторая по частоте буква шифр-текста Н — гласная (причем совпадающая с одной из букв И, Е, А), а пятая по частоте буква шифртекста В — согласная (и поэтому она, скорее всего Н, так как С и Т уже задействованы).

Обратим также внимание на два частых удвоения: ЕЕ (4 раза) и ВВ (6 раз). Самыми частыми в открытом тексте являются удвоения ИИ, НН, ОО, СС. Это дает основание полагать, что В заменяет букву Н открытого текста.

Буква Е — третья по частоте в криптограмме, поэтому вполне вероятно, что она заменяет одну из букв И, Е или А открытого текста. Учитывая сделанное замечание об удвоениях, сделаем предположение о том, что Е заменяет букву И.

Дополним последнюю таблицу новыми предположениями о заменах (см. табл. 4 на стр. 457).

Обратим внимание на то, что биграмма ВВ встречается в криптограмме 9 раз. По нашему предположению В заменяет в криптограмме букву открытого текста Н. Согласно таблице биграмм открытого текста (см. табл. 11 Приложения 1), самыми частыми биграммами с первой буквой Н являются НО, НА, НИ. Буквы О и И уже задействованы, поэтому оправдана гипотеза о том, что ВВ заменяет биграмму открытого текста НА.

Мы уже замечали, что Н заменяет одну из букв А, Е, И открытого текста. С учетом предыдущего остается лишь одна возможность: Н заменяет букву Е.

Внесем полученные результаты в табл. 5 (см. стр. 458).

Здесь лавинообразно определяются многие другие буквы. Так, из второй строки легко заметить, что Ш заменяет букву Д открытого текста, Т — букву Л. Из рассмотрения четвертой строки следует, что Ь заменяет М, из восьмой строки — что Р

заменяет В, а С — букву К, из восемнадцатой — что Ж заменяет Р и т. д.

Теперь легко закончить работу, убедившись в том, что наши гипотезы оправдались (см. табл. 6 на стр. 459).

В рассмотренном примере достаточно точно сработала статистика. Это объясняется большой длиной сообщения, содержащего в нашем случае 611 знаков. Нам не понадобились многие стандартные приемы, применяющиеся при решении подобных задач для текстов меньшей длины, о которых было упомянуто в комментариях к алгоритму 1.

Ключом использованного в примере шифра является следующий подстановочный алфавит:

А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П
Б	А	Р	Ы	Ш	Н	Я	К	Е	С	Т	Ь	В	Г	Д

Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ы	Ь	Э	Ю	Я
Ж	З	И	Л	М	О	П	У	Ф	Х	Ц	Ч	Щ	Э	Ю

составленный на основе ключевой фразы **БАРЫШНЯ КРЕСТЬЯНКА**, совпадающей с названием одноименной повести А.С. Пушкина. Заметим, что особенности таких (систематически перемешанных) алфавитов существенно помогают в восстановлении текстов. Соответствующая технология подробно изложена в [Кah67]. Там же содержится пример вскрытия простой замены, использующий значительно более тонкие рассуждения, чем приведенные в нашем примере.

Таблица 1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	Р	Г	Ш	В	І	Е	Г	К	Г	И	Ш	Б	Т	Н	В	В	Ц	О	В	Ь	Ф	Ф	О	Ы	Л
2																									
3	А	Н	Ж	В	Е	Е	В	Б	О	І	Ш	Е	Г	Г	З	Ч	Е	Ь	Н	В	Е	Н	Ф	Р	Б
4																									
5	В	Б	Д	Н	И	Ж	Г	Р	Е	У	Б	А	Н	Ж	Н	З	И	Г	Р	Б	Р	Ь	Г	Т	Г
6																									
7	Ш	Г	З	И	Е	З	Р	Г	Н	Ф	З	Т	Л	Я	Е	Т	І	В	Р	Ы	Р	Б	Ж	Ш	Е
8																									
9	Е	Р	Ц	Ф	Н	Т	Р	Г	И	З	И	Б	Р	С	Л	Р	В	Б	У	Б	Т	Н	Ы	Г	Ш
10																									
11	Б	Л	Н	О	Б	Т	Р	З	Р	Г	Э	Ш	Н	Ж	Н	Р	В	Э	Е	З	И	Н	О	Д	Г
12																									
13	Ж	Г	В	Г	И	И	Л	Ш	Б	В	Н	Р	Ц	Н	К	Я	Б	Т	Г	В	А	Ц	Т	Я	Н
14																									
15	В	Ь	И	В	Б	А	Н	Ш	В	І	Е	Ш	Р	Г	Ж	Ю	В	С	Н	С	Г	И	Г	Ж	Б
16																									
17	Ю	Л	Ь	Н	Ж	Т	Б	Р	Ж	Г	Ш	Б	О	Р	И	Г	Р	Ж	Н	Ь	Ю	С	Б	С	Г
18																									
19	В	В	Б	О	Г	Ш	Е	Т	З	Ю	Р	Г	И	Ч	Н	К	Я	Н	Ь	Д	Г	Т	Н	О	Г
20																									
21	К	Ю	Е	З	И	Р	Н	В	В	Ц	Н	Л	Д	Ж	Б	Я	В	Н	В	Ф	Ю	З	С	Г	Ж
22																									
23	Г	Н	Ы	Г	Л	И	Н	Ф	Т	Е	Г	В	Р	Ц	З	И	Ж	Г	Ф	Т	Ш	Г	Ь	Д	
24																									
25	Г	З	Г	А	З	И	Р	Н	В	В	Г	Ь	Л	Д	Т	Б	В	Л	К	Б	Р	Н	Т	Л	З
26																									
27	Н	А	Ю	З	Л	С	Г	В	В	Л	Э	М	Б	А	Ж	Е	С	Л	Л	И	Ж	Г	Е	Т	Ш
28																									
29	Г	О	Г	Ш	Ц	Е	З	И	Б	Т	Д	Г	У	Е	И	Б	И	Ч	З	Н	А	Ю	Л	Ь	В
30																									
31	Н	Ф	Ф	Е	Ь	У	Н	Т	Г	Р	Н	С	Г	Ь	Р	І	Р	З	Н	Ь	Г	С	Г	Т	Г
32																									
33	И	С	Н	Р	У	Н	Ь	Е	В	Н	Д	Ж	Н	С	Г	З	Т	Г	Р	Е	Т	Е	Н	Ь	Л
34																									
35	З	Г	З	Н	Ш	Е	Д	Ж	Е	Н	К	Я	Б	Р	Ф	Е	Н	С	В	Н	Ь	Л	Ы	Г	З
36																									
37	И	Е	И	Ч	З	З	Р	Г	Е	Ь	Ф	З	Н	Ь	Н	Е	З	И	Р	Б	Ь	Е	Е	З	Г
38																									
39	А	Б	С	Б	Ь	Е	Р	А	Л	Ш	В	Е	Г	В	О	Г	Ш	Е	Т	Р	Д	Т	Е	З	Г
40																									
41	Р	Г	Е	С	Л	Ж	И	С	Н	Д	І	Д	Ж	Б	К	Ш	В	Е	С	Б	Ь	В	Б	Ш	Н
42																									
43	Р	Б	Т	З	Н	Ж	И	Л	С	Е	К	З	Л	С	В	Б	Ш	Г	Ь	Б	Ф	В	Н	Е	Ж
44																									
45	Б	А	Г	И	Ц	З	Б	Ь	К	Б	Д	Ф	З	Ц	Р	Б	Т	Ж	Б	З	О	Г	Ш	Е	В
46																									
47	Е	У	Н	Ы	І	В	Н	У	Е	И	Б	Т	С	Ж	Г	Ь	Н	З	Н	В	Б	И	З	С	Е
48																									
49	О	Р	Н	Ш	Г	Ь	Г	З	И	Н	Ф														

Таблица 2

	А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ы	Ь	Э	Ю	Я	Σ
А	1		1			1	1			1	3									1							2		11		
Б	4	4	2		1	1	3	1	1				3	5	2	8	1	2			2					4	1	1	46		
В	1	9	6	4	6				2		7	1	2	1					2			2				1			42		
Г	2		8	1	6	4	7	7	1		2	1	7	1	3	1							9			6	1		67		
Д			5		1	4					1				2														13		
Е			3	2	1	4	1	10	3	2			4	2	3	3	8	2	1				1			3	1		54		
Ж	5	1	7		2			2	1		4					1						1					1		25		
З	1		4				1	11		2	7	1	3	2	2				1	1							1		37		
И	4	1	3		2	3	2	1		2	3		3	3	3					1	2	1							31		
К	2		1				1																1				1	3	9		
Л	1			2	1	2	2	1	1		1			1	3								2		1	2	1	1	22		
М		1																											1		
Н	2		7	2	6	5	2	1	3	1			3	5	3	3	1	1				3		3	7				58		
О		1	1	6	1										2										1				12		
П																													0		
Р	1	6	2	9	1	2	2	2	1			5			1	1	1		3					1	1				39		
С		3	2	7	2	1				3	3																		21		
Т	2		7	1	3	1	2		2	3		3	1									2					1		28		
У	2				2						3																		7		
Ф			1		4					1																			6		
Х																													0		
Ц							3				2	1	1	1	1														9		
Ч					1	1				1																			2		
Ш	6	2	5		6						2		1									1							23		
Щ																													0		
Ы				3		1				1				1															6		
Ь	1	2	3	2	4			1	3	4		1		1												1			23		
Э					1					1													1						3		
Ю		1			1	2		2						1	1														8		
Я	2	1			1						2																		6		

Таблица 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
1	Р	Г	Ш	В	Г	Е	Е	К	Г	И	Ш	Б	Т	Н	В	В	Ц	О	В	Б	Ф	Е	О	Ы	Л	
2		О		О				О	Т																	
3	А	Н	Ж	В	Ф	Е	В	Б	О	Г	Ш	Е	Т	Г	З	Ч	Е	Ь	Н	В	Е	Н	Е	Р	Б	
4									О					О	С											
5	В	Б	Д	Н	И	Ж	Г	Р	Е	У	Б	А	Н	Ж	Н	З	И	Г	Р	Б	Р	Ь	Г	Т	Г	
6				/	О										(Т	О					О			О	
7	Ш	Г	З	И	Е	З	Р	Г	Н	Е	З	Т	Л	Я	Е	Т	Г	В	Р	Ы	Р	Б	Ж	Ш	Е	
8	О	(/	(О			((О											
9	Е	Р	Ц	Ф	Н	Т	Р	Г	И	З	И	Б	Р	С	Л	Р	В	Б	У	Б	Т	Н	Ы	Г	Ш	
10							О	Т	С	Т												О				
11	Б	Л	Н	О	Б	Т	Р	З	Р	Г	Э	Ш	Н	Ж	Н	Р	В	Э	Е	З	И	Н	О	Д	Г	
12							С	Р	О											С	Т				О	
13	Ж	Г	В	Г	И	И	Л	Ш	Б	В	Н	Р	Ц	Н	К	Я	Б	Т	Г	В	А	Ц	Т	Я	Н	
14	О		О	Т	Т													О								
15	В	Б	И	В	Б	А	Н	Ш	В	Г	Е	Ш	Р	Г	Ж	Ю	В	С	Н	С	Г	И	Г	Ж	Б	
16			Т					О				О			О						О	Т	О			
17	Ю	Л	Ь	Н	Ж	Т	Б	Р	Ж	Г	Ш	Б	О	Р	И	Г	Р	Ж	Н	Ь	Ю	С	Б	С	Г	
18									О					Т	О										О	
19	В	В	Б	О	Г	Ш	Е	Т	З	Ю	Р	Г	И	Ч	Н	К	Я	Н	Ь	Д	Г	Т	Н	О	Г	
20				О				С			О	Т									О				О	
21	К	Ю	Е	З	И	Р	Н	В	В	Ц	Н	Л	Д	Ж	Б	Я	В	Н	В	Е	Ю	З	С	Г	Ж	
22				С	Т																С				О	
23	Г	Н	Ы	Г	Л	И	Н	Ф	Е	Т	Е	Г	В	Р	Ц	З	И	Ж	Г	Е	Т	Ш	Г	Ь	Д	
24	О			О	Т					О					С	Т			О						О	
25	Г	З	Г	А	З	И	Р	Н	В	В	Г	Ь	Л	Д	Т	Б	В	Л	К	Б	Р	Н	Т	Л	З	
26	О	С	О		С	Т					О														С	
27	Н	А	Ю	З	Л	С	Г	В	В	Л	Э	М	Б	А	Ж	Е	С	Л	Л	И	Ж	Г	Е	Т	Ш	
28				С		О														Т		О				
29	Г	О	Г	Ш	Ц	Е	З	И	Б	Т	Д	Г	У	Е	И	Б	И	Ч	З	Н	А	Ю	Л	Ь	В	
30	О	О			С	Т					О			Т	Т	С										
31	Н	Е	Ф	Е	Ь	У	Н	Т	Г	Н	С	Г	Ь	Р	Г	Р	З	Н	Ь	Г	С	Г	Т	Г		
32							О				О			О		О		С		О		О		О	О	О
33	И	С	Н	Р	У	Н	Ь	Е	В	Н	Д	Ж	Н	С	Г	З	Т	Г	Р	Е	Т	Е	Н	Ь	Л	
34	Т														О	С										
35	З	Г	З	Н	Ш	Е	Д	Ж	Е	Н	К	Я	Б	Р	Ф	Е	Н	С	В	Н	Ь	Л	Ы	Г	З	
36	С	О	С																						О	С
37	И	Е	И	Ч	З	З	Р	Г	Е	Ь	Е	З	Н	Ь	Н	Е	З	И	Р	Б	Ь	Е	Е	З	Г	
38	Т			С	С		О				С						С	Т							С	О
39	А	Б	С	Б	Ь	Е	Р	А	Л	Ш	В	Е	Г	В	О	Г	Ш	Е	Т	Р	Д	Т	Е	З	Г	
40												О			О										С	О
41	Р	Г	Е	С	Л	Ж	И	С	Н	Д	Г	Д	Ж	Б	К	Ш	В	Е	С	Б	Ь	В	Б	Ш	Н	
42	О						Т				О															
43	Р	Б	Т	З	Н	Ж	И	Л	С	Е	К	З	Л	С	В	Б	Ш	Г	Ь	Б	Ф	В	Н	Е	Ж	
44				С		Т					С						О									
45	Б	А	Г	И	Ц	З	Б	Ь	К	Б	Д	Е	З	Ц	Р	Б	Т	Ж	Б	З	О	Г	Ш	Е	В	
46		О	Т		С						С								С		О					
47	Е	У	Н	Ы	Г	В	Н	У	Е	И	Б	Т	С	Ж	Г	Ь	Н	З	Н	В	Б	И	З	С	Е	
48				О					Т						О				С			Т	С			
49	О	Р	Н	Ш	Г	Ь	Г	З	И	Н	Е															
50				О		О	С	Т																		

Таблица 4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25			
1	Р	Г	Ш	В	Е	Е	К	Г	И	Ш	Б	Т	Н	В	В	Ц	О	В	Б	Ф	Е	О	Ы	Л				
2	З	О	Н	В	О	И	В	О	Т	Ш	Е	Т	Г	З	Н	Ч	Ь	Н	В	Ф	И	Е	Р	Б				
3	А	Н	Ж	В	Е	Ф	В	Б	О	Г	Ш	Е	Т	Г	З	Ч	Ь	Н	В	Ф	И	Е	Р	Б				
4				Н	И	И	Н		О	И			О	С		И		Н	И		И							
5	В	Б	Д	Н	И	Ж	Г	Р	Е	У	Б	А	Н	Ж	Н	З	И	Г	Р	Б	Р	Ь	Г	Т	Г			
6	Н			Т			О		И						С		Т	О							О			
7	Ш	Г	З	И	Е	З	Р	Г	Н	Е	З	Т	Л	Я	Е	Т	Г	В	Р	Ы	Р	Б	Ж	Ш	Е			
8	О	С	Т	И	С		О		И	С					И		О	Н							И			
9	Е	Р	Ц	Ф	Н	Т	Р	Г	И	З	И	Б	Р	С	Л	Р	В	Б	У	Б	Т	Н	Ы	Г	Ш			
10	И						О	Т	С	Т						Н									О			
11	Б	Л	Н	О	Б	Т	Р	З	Р	Г	Э	Ш	Н	Ж	Н	Р	В	Э	Е	З	И	Н	О	Д	Г			
12							С	О		О						Н		И	С	В	А	Ц	Т	Я	Н			
13	Ж	Г	В	Г	И	И	Л	Ш	Б	В	Н	Р	Ц	Н	К	Я	Б	Т	Г	О	Н							
14	О	Н	О	Т	Т				Н																			
15	В	Б	И	В	Б	А	Н	Ш	В	Г	Е	Ш	Р	Г	Ж	Ю	В	С	Н	С	Г	И	Г	Ж	Б			
16	Н	Т	Н				Н	О	И				О			Н				О	Т	О						
17	Ю	Л	Ь	Н	Ж	Т	Б	Р	Ж	Г	Ш	Б	О	Р	И	Г	Р	Ж	Н	Ь	Ю	С	Б	С	Г			
18								О							Т	О									О			
19	В	В	Б	О	Ш	Е	Т	З	Ю	Р	Г	И	Ч	Н	К	Я	Н	Ь	Д	Г	Т	Н	О	Г				
20	Н	И		О	И		С		Ю							Н									О			
21	К	Ю	Е	З	И	Р	Н	В	В	Ц	Н	Л	Д	Ж	Б	Я	В	Н	В	Е	Ю	З	С	Г	Ж			
22			И	С	Т		Н	Н								Н		Н	И		С				О			
23	Г	Н	Ы	Г	Л	И	Н	Ф	Е	Т	Е	Г	В	Р	Ц	З	И	Ж	Г	О	И	Т	Ш	Б	Д			
24	О			О	Т			И		И	О	Н				С	Т						Г	О				
25	Г	З	Г	А	З	И	Р	Н	В	В	Г	Ь	Л	Д	Т	Б	В	Л	К	Б	Р	Н	Т	Л	З			
26	О	С	О	С	Т			Н	Н	О						Н									С			
27	Н	А	Ю	З	Л	С	Г	В	Л	Э	М	Б	А	Ж	Е	С	Л	Л	И	Ж	Г	Е	Т	Ш				
28			С		О	Н	Н								И				Т		Ж	О	И					
29	О	О	Г	Ш	Ц	Е	З	И	Б	Т	Д	Г	У	Е	И	Б	И	Ч	З	Н	А	Ю	Л	Ь	В			
30	О	И	О		И	С	Т			О				И	Т		Т		С						Н			
31	Н	Е	Ф	Е	Ь	У	Н	Т	Г	Р	Н	С	Г	Ь	Р	Г	Р	З	Н	Ь	Г	С	Г	Т	Г			
32	И	И	И				О		О			О			О		С		С		О		О		О			
33	И	С	Н	Р	У	Н	Ь	Е	В	Н	Д	Ж	Н	С	Г	З	Т	Г	О	Р	Е	Т	Е	Н	Ь	Л		
34	Т														О	С	Г	З	Т	Г	О	Р	Е	Т	Е	Н	Ь	Л
35	З	Г	З	Н	Ш	Е	Д	Ж	Е	Н	К	Я	Б	Р	Ф	Е	Н	С	В	Н	Ь	Л	Ы	Г	З			
36	С	О	С		И			И							И										О	С		
37	И	Е	И	Ч	З	Р	Г	Е	Ь	Е	З	Н	Ь	Н	Е	З	И	Р	Б	Ь	Е	Е	З	Г	О			
38	Т	И	Т	С		О	И		И	С					И	С	Т				И	И	С		О			
39	А	Б	С	Б	Ь	Е	Р	А	Л	Ш	В	Е	Г	В	О	Г	Ш	Е	Т	Р	Д	Т	Е	З	Г			
40					И					Н	И	О	Н		О		И						И	С	О			
41	Р	Г	Е	С	Л	Ж	И	С	Н	Д	Г	Д	Ж	Б	К	Ш	В	Е	С	Б	Ь	В	Б	Ш	Н			
42	О	И			Т					О						Н	И					Н						
43	Р	Б	Т	З	Н	Ж	И	Л	С	Е	К	З	Л	С	В	Б	Ш	Г	Ь	Б	Ф	В	Н	Е	Ж			
44			С		Т				И		С				Н		О					Н						
45	Б	А	Г	И	Ц	З	Б	Ь	К	Б	Д	Е	З	Ц	Р	Б	Т	Ж	Б	З	О	Г	Ш	И	В			
46		О	Т		С							И	С						С		О	Г	Ш	И	Н			
47	Е	У	Н	Ы	Г	В	Н	У	Е	И	Т	Б	Т	С	Ж	Г	Ь	Н	З	Н	В	Б	И	З	С	Е		
48	И				О				И	И					О						И	Т	С		И			
49	О	Р	Н	Ш	Г	Ь	Г	З	И	Н	Е																	
50				О		О		С	Т	И																		

Таблица 5

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	Р	О	Ш	В	О	Е	Е	К	О	И	Ш	Б	Т	Н	В	В	Ц	О	В	Б	Ф	Е	О	Ы	Л
2	Г	О	Ш	Н	О	И	И	К	О	Т	Ш	Б	Т	Н	В	В	Ц	О	В	Б	Ф	Е	О	Ы	Л
3	А	Н	Ж	В	Е	Е	В	Б	О	Г	Ш	Е	Т	Г	З	Ч	Е	Ь	Н	В	Е	Н	Е	Р	Б
4	Е	Б	Д	Н	И	Ж	Г	Р	Е	У	Б	А	Н	Ж	Е	Н	З	И	Г	О	Р	Ь	Р	Б	А
5	В	Б	Д	Н	И	Ж	Г	Р	Е	У	Б	А	Н	Ж	Е	Н	З	И	Г	О	Р	Ь	Р	Б	А
6	Н	А	Е	Т	И	С	О	Е	И	С	А	Е	Я	И	Е	Т	Г	О	В	Н	Ы	Р	Б	Ж	Ш
7	Ш	Г	З	И	Е	З	Р	Г	Н	Е	З	Т	Л	Я	И	Е	Т	Г	О	В	Н	Ы	Р	Б	Ж
8	Т	И	С	О	Е	И	С	А	Е	Я	И	Е	Т	Г	О	В	Н	Ы	Р	Б	Ж	Ш	Е	И	
9	Е	Р	Ц	Ф	Н	Т	Р	Г	И	З	И	Ь	Р	С	Л	Р	В	Б	А	У	Б	Т	Н	Ы	Г
10	И	Л	Н	О	Б	Т	Р	З	Р	Г	Э	Ш	Н	Ж	Н	Р	В	Э	Е	З	И	Н	О	Д	Г
11	Б	Л	Н	О	Б	Т	Р	З	Р	Г	Э	Ш	Н	Ж	Н	Р	В	Э	Е	З	И	Н	О	Д	Г
12	А	Е	Г	А	И	Л	Ш	Б	В	Н	Р	Ц	Н	К	Я	Б	А	Т	Г	О	В	А	Ц	Т	Я
13	Ж	О	Н	О	Т	И	Л	Ш	Б	В	Н	Р	Ц	Н	К	Я	Б	А	Т	Г	О	В	А	Ц	Т
14	О	Н	О	Т	И	Л	Ш	Б	В	Н	Р	Ц	Н	К	Я	Б	А	Т	Г	О	В	А	Ц	Т	Я
15	В	Б	И	В	Б	А	Н	Ш	В	Г	Е	Ш	Р	Г	Ж	Ю	В	С	Н	Г	И	Г	Ж	Б	А
16	Н	А	Т	Н	А	Е	Н	О	И	О	И	О	О	О	О	О	О	О	О	О	О	О	О	О	О
17	Ю	Л	Ь	Н	Ж	Т	Б	Р	Ж	Г	Ш	Б	О	Р	И	Г	Р	Ж	Н	Ь	Ю	С	Б	С	Г
18	О	Е	Н	Е	Т	Е	И	О	И	О	И	О	И	О	И	О	И	О	И	О	И	О	И	О	И
19	В	Н	Б	О	Г	Ш	Е	Т	З	Ю	Р	Г	И	Ч	Н	К	Я	Н	Ь	Д	Г	Т	Н	О	Г
20	Н	А	Ю	Е	З	И	Р	Н	В	Ц	Н	Л	Д	Ж	Б	Я	В	Н	Е	Н	И	О	З	С	Г
21	К	Ю	Е	З	И	Р	Н	В	Ц	Н	Л	Д	Ж	Б	Я	В	Н	Е	Н	И	О	З	С	Г	Ж
22	О	Н	И	С	Т	И	Е	Н	И	Е	Н	И	Е	Н	И	Е	Н	И	Е	Н	И	Е	Н	И	Е
23	Г	Н	Ы	Г	Л	И	Ф	Е	Т	Е	И	О	И	О	И	О	И	О	И	О	И	О	И	О	И
24	О	Е	Г	О	Л	Т	Е	И	О	И	О	И	О	И	О	И	О	И	О	И	О	И	О	И	О
25	Г	З	Г	А	З	И	Р	Н	В	В	Г	Ь	Л	Д	Т	Б	В	Л	К	Б	Р	Н	Т	Л	З
26	О	С	О	С	Т	С	Г	В	Е	Н	О	О	О	О	О	О	О	О	О	О	О	О	О	О	О
27	Н	А	Ю	З	Л	С	Г	В	Л	Э	М	Б	А	Ж	Е	С	Л	Л	И	Ж	Г	Е	Т	Ш	
28	Е	А	Ю	З	Л	С	Г	В	Л	Э	М	Б	А	Ж	Е	С	Л	Л	И	Ж	Г	Е	Т	Ш	
29	Г	О	Г	Ш	Ц	Е	З	И	Б	Т	Д	Г	О	У	Е	И	А	И	Ч	З	Н	А	Ю	Л	Ь
30	О	Г	Ш	Ц	Е	З	И	Б	Т	Д	Г	О	У	Е	И	А	И	Ч	З	Н	А	Ю	Л	Ь	
31	Н	Е	Ф	Е	Ь	У	Н	Т	Г	Р	Н	С	Г	Ь	Р	Г	Р	З	Н	Ь	Г	С	Г	Т	Г
32	Е	Н	И	Е	У	Н	Т	Г	Р	Н	С	Г	Ь	Р	Г	Р	З	Н	Ь	Г	С	Г	Т	Г	О
33	И	С	Н	Р	У	Н	Ь	Е	В	Н	Д	Ж	Н	С	Г	З	Т	Г	Р	Е	Т	Е	Н	Ь	Л
34	Т	Е	Н	У	Н	Ь	Е	В	Н	Д	Ж	Н	С	Г	З	Т	Г	Р	Е	Т	Е	Н	Ь	Л	
35	З	О	С	Е	Н	Ш	Е	Д	Ж	Е	Н	К	Я	Б	Р	Ф	Е	Н	С	В	Н	Ь	Л	Ы	Г
36	С	О	С	Е	Н	Ш	Е	Д	Ж	Е	Н	К	Я	Б	Р	Ф	Е	Н	С	В	Н	Ь	Л	Ы	Г
37	И	Е	И	Ч	З	З	Р	Г	Е	Ь	Е	З	Н	Ь	Н	Е	З	И	Р	Б	Ь	Е	Е	З	Г
38	Т	И	Т	С	С	О	И	И	С	Е	И	С	Е	И	С	Т	Г	А	Б	И	И	С	О	И	С
39	А	Б	С	Б	Ь	Е	Р	А	Л	Ш	В	Е	Г	Н	О	Г	Ш	Е	Т	Р	Д	Т	Е	З	Г
40	Б	А	С	Б	Ь	Е	Р	А	Л	Ш	В	Е	Г	Н	О	Г	Ш	Е	Т	Р	Д	Т	Е	З	Г
41	Р	Г	Е	С	Л	Ж	И	С	Н	Д	Г	Д	Ж	Б	К	Ш	В	Н	И	С	Б	Ь	В	Б	Ш
42	О	И	Т	Л	Ж	И	С	Н	Д	Г	Д	Ж	Б	К	Ш	В	Н	И	С	Б	Ь	В	Б	Ш	
43	Р	Б	Т	З	Н	Ж	И	Л	С	Е	К	З	Л	С	В	Б	Ш	Г	А	Б	Ф	В	Н	Е	Ж
44	Б	А	Г	Т	Ц	З	А	Б	К	Б	Д	Е	З	Ц	Р	Б	Т	Ж	Б	З	О	Г	Ш	Е	В
45	А	Г	Т	Ц	З	А	Б	К	Б	Д	Е	З	Ц	Р	Б	Т	Ж	Б	З	О	Г	Ш	Е	В	
46	Е	У	Н	Ы	Г	В	Н	У	Е	И	Б	Т	С	Ж	Г	Ь	Н	З	Н	В	Б	И	З	С	Е
47	И	У	Н	Ы	Г	В	Н	У	Е	И	Б	Т	С	Ж	Г	Ь	Н	З	Н	В	Б	И	З	С	Е
48	О	Р	Н	Ш	Г	Ь	Г	З	И	Н	Е	И	Т	А											
49	О	Р	Н	Ш	Г	Ь	Г	З	И	Н	Е	И	Т	А											
50	Е	О	И	С	Т	И	Е	О	С	Т	Е	И													

Таблица 6

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
1	В	О	Д	Н	О	И	И	З	О	Т	Д	Ш	Г	Л	Т	Н	В	В	В	У	Н	А	Ш	О	Ы	П
2	А	Н	Ж	В	Е	Е	В	Б	О	У	Г	Ш	Л	Л	Г	З	Ч	Е	М	Н	В	Г	Н	Р	Б	А
3	Б	Е	Р	Н	И	И	Н	А	У	О	Д	И	Л	О	С	Ь	И	М	Е	Н	И	Е	И	В	Г	А
4	В	Б	Д	Н	И	Ж	Р	Г	Ф	У	Ч	А	Б	Е	Р	Ф	С	Г	О	В	А	В	М	О	Л	О
5	Н	А	П	Е	Т	Р	О	В	И	Ч	А	Б	Е	Р	Ф	С	Г	О	В	А	В	М	О	Л	О	Г
6	Ш	Г	З	И	Е	З	Р	Г	Н	Ч	А	Б	Е	Р	Я	И	Г	О	В	Р	Ы	Р	Б	Ж	Ш	И
7	Д	О	С	Т	И	С	В	О	Л	И	С	Т	У	Ж	И	Л	О	В	Г	В	А	Р	В	А	Ж	Д
8	Е	Р	Ц	Ф	Н	Т	Р	Г	И	З	И	Б	Р	С	Л	Р	В	Ь	У	Ь	І	Н	Ы	І	Ш	І
9	И	В	Ы	Ш	Е	Л	В	О	Т	С	Г	А	В	К	У	В	Н	А	Ч	А	Л	Е	Г	О	Д	І
10	Б	Л	Н	О	Б	Т	Р	З	Р	Г	Э	Ш	Н	Ж	Н	Р	В	Э	И	З	І	Н	О	Д	Г	О
11	А	У	Е	А	А	Л	В	С	В	О	Ю	Д	Ф	Р	Ф	В	Р	Ю	И	С	І	Г	У	П	Г	О
12	Ж	Г	В	Г	И	И	Л	Ш	Б	В	Н	Р	Ц	Н	К	Я	Б	Т	Г	В	А	Ц	І	Я	Н	Е
13	Р	О	Н	О	Т	Т	У	Д	А	Н	Е	В	Ы	Е	З	Ж	А	І	О	Н	Ь	Ы	І	Ж	Б	Е
14	В	Б	И	В	Б	А	Н	Ш	В	Г	О	И	Д	В	Р	Г	Ю	В	С	Н	С	Г	И	Г	Ж	Е
15	Н	А	Т	Н	А	Б	Ф	Л	В	О	И	Ш	Д	В	Р	Г	Ю	В	С	Н	С	Г	И	Г	Ж	Е
16	Н	А	Т	Н	А	Б	Ф	Л	В	О	И	Ш	Д	В	Р	Г	Ю	В	С	Н	С	Г	И	Г	Ж	Е
17	Ю	Л	Ь	Н	Ж	Т	Б	Р	Ж	Г	Ш	Б	О	В	И	Т	О	В	Р	Ж	Н	Ь	Ю	С	Б	С
18	Я	У	М	Е	Р	Л	А	В	Р	О	Д	А	У	В	Т	О	В	Р	Ф	М	Я	К	А	К	О	Г
19	В	В	Б	О	Г	Ш	Е	Т	Ю	Р	Г	И	Ч	Н	К	Я	Н	Ь	Д	Г	Т	Н	О	Г	О	Г
20	Н	Н	А	У	О	Д	И	Л	С	Я	В	О	Т	Ь	Ф	З	Ж	Ф	М	П	О	Л	Е	Х	О	Ж
21	К	Ю	Е	З	И	Р	Н	В	Ц	Н	Л	Д	П	Ж	Б	Я	В	Н	В	Е	Ю	З	С	Г	Р	Ж
22	З	Я	И	С	Т	В	Е	Н	Н	Ы	Е	У	П	Р	А	Ж	Н	Е	Н	И	Я	С	К	О	Г	О
23	Г	Н	Ы	Г	Л	И	Н	Ф	Е	Т	Е	Г	В	Р	Ц	З	И	Ж	Г	Е	Т	Ш	І	Ь	Д	П
24	О	Е	І	О	У	Т	Е	Ш	И	Л	И	О	Н	В	Ы	С	Т	Р	О	И	Л	Д	О	М	П	І
25	Г	З	Г	А	З	И	Р	Н	В	В	Г	Ь	Л	Д	Т	Б	В	Ч	К	Б	Р	Н	Т	Л	З	С
26	О	С	О	Б	С	Т	В	Е	Н	Н	О	М	У	П	Л	Б	Н	У	З	А	В	Ф	Л	У	С	О
27	Н	А	Ю	З	Л	С	Г	В	В	Л	Э	М	Б	А	Б	Ж	Е	С	У	Л	У	Т	И	Ж	Г	Е
28	Е	Б	Я	С	У	К	О	Н	Н	У	Ю	Ф	А	Б	Е	Р	И	К	У	У	Т	Р	О	И	Т	Д
29	Г	О	Я	Г	Ш	Ц	Е	З	И	Ь	Т	Д	Г	У	Е	И	Б	И	Ч	З	Н	А	Ю	Л	Ь	В
30	О	У	О	Д	Ы	И	С	Т	А	Л	П	О	Ч	И	Т	А	Т	Ь	С	Е	Б	Я	У	М	Н	І
31	Н	Е	Ф	Е	Ь	У	Н	Т	Г	Р	Н	С	Г	М	Р	Г	В	З	Н	Ь	Г	С	Г	Т	Г	О
32	Е	И	Ш	И	М	Ч	Ф	Л	О	В	Е	К	О	М	В	О	Р	С	Е	М	О	К	О	Л	О	Г
33	И	С	Н	Р	У	Н	Ь	Р	В	Н	Д	Ж	Н	С	Г	З	І	Г	Е	Р	Е	Т	Е	Н	Ь	Л
34	Т	К	Е	В	Ч	Е	М	И	Н	Ь	П	Р	Е	К	Ф	О	С	Л	О	В	И	Л	И	Ф	М	У
35	З	Г	З	Н	Ш	Е	Д	Ж	Е	Н	К	Я	Б	Р	Ф	Е	Н	С	В	Н	Ь	У	Ы	Г	О	З
36	С	О	С	Е	Д	И	П	Р	И	Е	З	Ж	А	В	Ш	И	Е	К	Н	Е	М	У	Л	Г	О	С
37	И	Е	И	Ч	З	З	Р	Г	Е	Ь	Е	З	Н	Ь	Н	Е	З	И	Р	Б	Ь	Е	Е	З	Г	О
38	Т	И	Т	Ь	С	С	В	О	И	М	И	С	Е	М	Е	И	С	Т	В	А	М	И	И	С	О	Г
39	А	Б	С	Б	Ь	Е	Р	А	Ц	Ш	В	Г	Е	В	О	Г	Ш	Ф	Т	Р	Д	Т	Р	Д	И	С
40	Б	А	К	А	М	И	В	Б	У	Д	Н	И	О	Н	Х	О	Д	И	Д	В	П	Л	И	С	О	Г
41	Р	Г	Е	С	Л	Ж	И	С	Н	Д	Г	Д	Ж	Б	К	Д	Ш	Н	И	С	Б	М	В	Б	Ш	Е
42	В	О	И	К	У	Р	Т	К	Е	П	О	П	Р	А	З	Д	В	Н	И	С	Б	М	Н	А	Д	Е
43	Р	Б	Т	З	Н	Ж	И	Ч	С	Е	К	З	Л	С	В	Б	Ш	Г	М	А	Б	Ф	В	Н	Е	Ж
44	В	А	Л	С	Е	Р	Т	У	К	И	З	С	У	К	Н	А	Д	О	М	А	Ш	Н	Е	И	Р	І
45	Б	А	Г	И	Ц	З	Б	Ь	К	Б	Д	Е	З	Ц	Р	В	А	Т	Ж	Б	С	О	Х	О	Ш	І
46	А	Б	О	Т	Ы	С	А	М	З	А	П	И	С	Ы	Р	В	А	Д	Ж	Р	А	З	О	Г	Д	И
47	Е	У	Н	Ы	Г	В	Н	У	Е	И	Б	Г	С	Ж	Г	Ь	Н	З	Н	В	Б	И	З	С	И	Е
48	И	Ч	Е	Г	О	Н	Е	Ч	И	Т	А	Л	К	Ж	Р	О	М	Е	С	Н	А	Т	С	К	И	І
49	О	Р	Н	Ш	Г	Б	Г	З	И	Н	Е															
50	Х	В	Е	Д	О	М	О	С	Т	Е	И															

Приложение 3

Элементы алгебры и теории чисел

Модулярная арифметика

Пусть a и n — натуральные числа. “Разделить число a на число n с остатком” — это значит найти целые числа q и r , удовлетворяющие условию

$$a = q \cdot n + r, \text{ где } 0 \leq r < n.$$

При этом число q называют неполным частным, а r — остатком от деления числа a на число n .

Если остаток r равен нулю, то говорят, что число n делит число a , или, по-другому, n является делителем числа a ,

Целые числа a и b называют *сравнимыми по модулю n* , если их остатки при делении на n совпадают. Обычно для обозначения этого факта используется запись

$$a \equiv b \pmod{n}.$$

Отсюда, в частности, следует, что число n делит разность чисел a и b . Для обозначения остатка часто используют бесскобочную запись

$$b = a \bmod n.$$

Операцию нахождения числа $b = a \bmod n$ называют *приведением числа a по модулю n* .

Множество целых чисел a_0, \dots, a_{n-1} , таких, что для любого целого числа b найдется $k \in \{0, \dots, n-1\}$ со свойством $a_k \equiv b \pmod{n}$, называется *полной системой вычетов по модулю n* .

дулю n . Обычно используется полная система вычетов $\{0, 1, \dots, n-1\}$.

Разложение чисел на простые множители

Натуральное число a , большее 1, называется *простым*, если оно не имеет натуральных делителей, отличных от 1 и самого числа a .

Любое натуральное число, отличное от 1, либо является простым, либо может быть представлено в виде произведения простых чисел. Это представление определено однозначно с точностью до порядка сомножителей в произведении.

Следует отметить, что в настоящее время нет достаточно эффективных алгоритмов разложения произвольного целого числа на простые множители даже в случае, когда известно, что оно разлагается в произведение двух простых чисел. Отсутствие эффективных алгоритмов с доказуемыми оценками сложности позволяет использовать задачу разложения натуральных чисел на простые множители при обосновании стойкости некоторых криптографических алгоритмов.

Алгоритм Евклида нахождения наибольшего общего делителя

Наибольшее целое число, делящее одновременно целые числа a и b , называется их наибольшим делителем и обозначается $\text{НОД}(a, b)$ или просто (a, b) . Если $(a, b)=1$, то a и b называются взаимно простыми числами.

Алгоритм Евклида нахождения наибольшего общего делителя двух целых чисел заключается в проведении следующей последовательности операций деления с остатком:

$$a = q \cdot b + r, \text{ где } 0 \leq r < b,$$

$$b = q_1 \cdot r + r_1, \text{ где } 0 \leq r_1 < r,$$

$$r = q_2 \cdot r_1 + r_2, \text{ где } 0 \leq r_2 < r_1,$$

$$r_1 = q_3 \cdot r_2 + r_3, \text{ где } 0 \leq r_3 < r_2,$$

$$\dots$$

$$r_k = q_{k+2} \cdot r_{k+1} + r_{k+2}, \text{ где } 0 < r_{k+2} < r_{k+1},$$

$$\dots$$

$$r_1 = q_3 \cdot r_2 + r_3, \text{ где } 0 \leq r_3 < r_2.$$

Корректное завершение алгоритма гарантируется тем, что остатки от делений образуют строго убывающую последовательность натуральных чисел. Из приведенных равенств следует, что

$$(a, b) = (b, r) = (r, r_1) = \dots = (r_{n-1}, r_n) = r_n.$$

Поэтому наибольший делитель чисел a и b совпадает с r_n .

Как следствие из алгоритма Евклида, можно получить утверждение, что наибольший делитель целых чисел a и b может быть представлен в виде линейной комбинации этих чисел, т. е. существуют целые числа u и v такие, что справедливо равенство

$$a \cdot u + b \cdot v = r_n.$$

Вычисление обратного элемента по заданному модулю

Если целые числа a и n взаимно просты, то существует число a' , удовлетворяющее сравнению $a \cdot a' \equiv 1 \pmod{n}$. Число a' называют *обратным к a по модулю n* и используют обозначение $a^{-1} \pmod{n}$. Вычислить a' можно, например, воспользовавшись представлением наибольшего общего делителя чисел a и n в виде их линейной комбинации: $a \cdot u + b \cdot n = 1$. Взяв наименьшие неотрицательные вычеты обеих частей этого равенства по модулю n , получим, что искомое значение a' удовлетворяет сравнению $a' \equiv u \pmod{n}$.

Вычисление обратных величин по некоторому модулю может быть выполнено более просто, если использовать некоторые факты из теории чисел.

Функцией Эйлера называется функция $\varphi(n)$, определенная на множестве натуральных чисел и равная количеству целых чисел в промежутке $[1, n]$, взаимно простых с n .

Если известно, что $n = p_1^{k_1} \cdot \dots \cdot p_l^{k_l}$, где p_1, \dots, p_l — различные простые числа, то формула вычисления функции Эйлера имеет вид

$$\varphi(n) = \prod_{j=1}^l (p_j - 1) \cdot p_j^{k_j - 1}.$$

Справедлива малая теорема Ферма: если n — простое число и $\text{НОД}(a, n) = 1$, то

$$a^{n-1} \equiv 1 \pmod{n}.$$

Обобщение малой теоремы Ферма, полученное Эйлером, утверждает: если $\text{НОД}(a, n) = 1$, то

$$a^{n-1} \equiv 1 \pmod{n}.$$

С учетом приведенных фактов получаем, что наиболее просто значение $a^{-1} \pmod{n}$ находится из соотношения

$$a^{-1} \equiv a^{\varphi(n)-1} \pmod{n}.$$

Китайская теорема об остатках

Любое целое неотрицательное число, не превосходящее произведения натуральных чисел m_1, m_2, \dots, m_l , можно однозначно восстановить, если известны его вычеты по этим модулям. Этот результат, полученной в I веке китайским мате-

матиком Сун Це, носит название китайской теоремы об остатках.

Математическая формулировка этого результата такова:

Теорема. Пусть m_1, m_2, \dots, m_t — попарно взаимно простые числа. Тогда для любых целых чисел a_1, a_2, \dots, a_t сравнения

$$x \equiv a_1 \pmod{m_1},$$

...

$$x \equiv a_t \pmod{m_t}$$

имеют в интервале $[0, M-1]$, $M = m_1 \cdot m_2 \cdot \dots \cdot m_t$ единственное общее решение вида

$$x = \sum_{j=1}^t a_j \cdot N_j \cdot M_j \pmod{M},$$

где $M_j = \frac{M}{m_j}$, $a_j N_j = M_j^{-1} \pmod{m_j}$, $j = 1, \dots, t$.

Алгебраические структуры

Множество элементов G с заданной на нем бинарной операцией “ \cdot ” называется *группой*, если выполнены три условия:

- 1) операция “ \cdot ” ассоциативна, то есть $(a \cdot b) \cdot c = a \cdot (b \cdot c)$,
- 2) существует элемент e из G , такой, что для любого g из G выполняются равенства $e \cdot g = g \cdot e = g$,
- 3) для любого g из G существует элемент g' из G со свойством $g \cdot g' = g' \cdot g = e$.

Обычно используется обозначение (G, \cdot) .

Элемент e из G называют *нейтральным* элементом группы, а элемент g' — *обратным* элементом к g . Для обратного элемента обычно используется обозначение $g' = g^{-1}$. Следует

отметить, что в группе G нейтральный элемент и элемент, обратный к элементу g , определены однозначно.

С точки зрения решения уравнений основное свойство группы состоит в том, что в ней однозначно разрешимы уравнения вида

$$\begin{aligned} a \cdot x &= b, \\ y \cdot a &= b \end{aligned}$$

при любых $a, b \in G$.

Заметим, что если при всех a, b и c эти уравнения однозначно разрешимы относительно x и y , то (G, \cdot) называется *квазигруппой*. Для квазигруппы не обязательно выполняются условия 1)–3) из определения группы. Вместе с тем ассоциативная квазигруппа всегда является группой.

Операция “ \cdot ” называется коммутативной, если для любых двух элементов a и b из G выполнено равенство $a \cdot b = b \cdot a$. В этом случае группа G называется *коммутативной* или *абелевой*.

Примером группы является множество комплексных корней степени n из 1 с операцией умножения корней как комплексных чисел.

Множество R с двумя бинарными ассоциативными операциями сложения “+” и умножения “ \cdot ” называется *кольцом*, если выполнены следующие условия:

— множество R с бинарной операцией сложения “+” является абелевой группой,

— операция “ \cdot ” удовлетворяет условию дистрибутивности относительно операции “+”, т. е. $(a + b) \cdot c = a \cdot c + b \cdot c$ и $a \cdot (b + c) = a \cdot b + a \cdot c$.

Если операция “ \cdot ” коммутативна, то кольцо называется *коммутативным*.

Примером кольца является множество Z_n , образующее полную систему вычетов целых чисел по модулю n с операциями сложения и умножения по модулю n , причем это кольцо является коммутативным.

Нейтральный элемент кольца относительно операции “+” называют нулем кольца и обозначают через 0. При умножении на 0 любого элемента кольца будет получаться 0.

Можно отдельно рассмотреть множество всех ненулевых элементов кольца с операцией умножения “·”. Для этого множества можно ввести понятия нейтрального и обратного элементов относительно операции умножения “·”. Нейтральный элемент кольца относительно операции “·” называют единицей кольца и обозначают через 1.

Единица существует не в любом кольце. Например, в кольце четных целых чисел единица отсутствует. Если в кольце существует единица, то в общем случае обратные элементы определены не для всех элементов кольца.

Поле называется коммутативное кольцо с единицей, отличной от нуля, в котором любой ненулевой элемент обратим.

Кольцо вычетов целых чисел по модулю n является полем в том и только в том случае, когда n — простое число. Другими примерами полей являются хорошо известные множества рациональных, действительных и комплексных чисел с обычными операциями сложения и умножения.

Группы подстановок

Подстановкой непустого множества M называют любое биективное (взаимно однозначное) отображение множества M в себя. Множество всех подстановок на множестве M обозначают через $S(M)$.

Множество $S(M)$ относительно операции суперпозиции отображений образует группу. Если M — конечное множество мощности n , то говорят, что $S(M)$ — *симметрическая группа подстановок степени n* .

Группа $S(M)$ является коммутативной только в случае $n \leq 2$.

Заметим, что, перенумеровав элементы множества M некоторым фиксированным образом: $M = \{x_1, x_2, \dots, x_n\}$ и ото-

ждествив элементы x_i с их номерами i , вместо группы $S(M)$ можно рассматривать группу $S(\Omega)$, где $\Omega = \{1, 2, \dots, n\}$. Обычно группу $S(\Omega)$ обозначают через S_n .

Любая подгруппа G группы S_n (подмножество группы S_n , которое само является группой) называется *группой подстановок степени n* .

Конечные поля

Число элементов конечного поля равно p^t для некоторого простого числа p и натурального числа t . Обычно поле из p^t элементов обозначается $GF(p^t)$. Пусть $q = p^t$.

Сформулируем основные свойства конечных полей.

1. Все элементы поля $GF(q)$ являются корнями многочлена $x^q - x$.
2. Любой многочлен $f(x)$ степени m , неприводимый над полем $GF(q)$, является делителем многочлена $x^{q^m} - x$. (В частности, все корни многочлена $f(x)$ содержатся в поле $GF(q^m)$, то есть оно является *полем разложения* этого многочлена.)
3. Любой делитель многочлена $x^{q^m-1} - 1$, неприводимый над полем $GF(q)$, имеет степень, делящую значение m .
4. В поле $GF(q)$ существует примитивный элемент α такой, что все ненулевые элементы поля представляются в виде его степеней, то есть мультипликативная группа конечного поля является циклической.

В прикладных целях обычно используются задания конечных полей в виде кольца вычетов целых чисел по простому модулю p (поле $GF(p)$) либо в виде фактор-кольца кольца многочленов над полем $GF(p)$ по модулю неприводимого многочлена (поля $GF(p^t)$, $t > 1$).

В последнем случае для задания поля $GF(p^t)$ рассматриваются многочлены над полем $GF(p)$. Для них вводится поня-

тие деления с остатком: разделить многочлен $a(x)$ на многочлен $b(x)$ степени s — это значит представить многочлен $a(x)$ в виде

$$a(x) = q(x) \cdot b(x) + r(x),$$

где степень многочлена $r(x)$ строго меньше s .

По аналогии с целыми числами вводятся понятия вычета по модулю многочлена $b(x)$, сравнимости многочленов и операции сложения и умножения по модулю многочлена.

Роль полной системы вычетов по модулю многочлена $p(x)$ выполняет множество всех возможных остатков от деления многочленов над полем $GF(p)$ на $p(x)$. Другими словами, полную систему вычетов по модулю многочлена $p(x)$ степени s образует множество многочленов

$$\{r(x) = r_0 + r_1x + \dots + r_{s-1}x^{s-1}, \quad r_0, \dots, r_{s-1} \in GF(p)\}.$$

Множество вычетов по модулю фиксированного многочлена $p(x)$ степени s с операциями сложения и умножения образует коммутативное кольцо. Это кольцо является полем в том и только в том случае, когда многочлен $p(x)$ неприводим (т.е. не раскладывается над $GF(p)$ в произведение многочленов меньших степеней).

Наличие в конечном поле примитивного элемента α позволяет ввести понятие логарифма для ненулевых элементов этого поля. Логарифм элемента β по основанию α определяется как наименьшее целое неотрицательное число k , удовлетворяющее равенству $\beta = \alpha^k$. В настоящее время задача вычисления логарифма в конечном поле в общем случае не имеет достаточно эффективных алгоритмов решения и по этой причине, наряду с задачей разложения на множители, используется при построении стойких криптографических алгоритмов и протоколов.

Литература

- [Ано97] *Анохин М. И., Варновский Н. П., Сидельников В. М., Яценко В. В.* Криптография в банковском деле. — М.: Изд-во МИФИ, 1997.
- [Арш83] *Аршинов М. Н., Садовский Л. Е.* Коды и математика. — М.: Наука, 1983.
- [Баб97] *Бабаиш А. В., Глухов М. М., Шанкин Г. П.* О преобразованиях множества слов в конечном алфавите, не размножающих искажений // Дискретная математика. — 1997. — Т. 9. — № 3.
- [Бер92] *Березин Б. В., Дорошкевич П. В.* Цифровая подпись на основе традиционной криптографии // Защита информации. — 1992. — Вып. 2. — С. 148—167.
- [Бил69] *Биллингслей П.* Эргодическая теория и информация. — М.: Мир, 1969.
- [Бол86] *Болл У., Коксетер Г.* Математические эссе и развлечения (криптография и криптографический анализ). — М.: Мир, 1986.
- [Бра99] *Брассар Ж.* Современная криптология. — М.: Полимед, 1999.
- [Вар95] *Варфоломеев А. А., Пеленицын М. Б.* Методы криптографии и их применение в банковских технологиях. — М.: Изд-во МИФИ, 1995.
- [Вар00] *Варфоломеев А. А., Жуков А. Е., Пудовкина М. А.* Поточные криптосистемы. Основные свойства и методы анализа стойкости. — М.: Изд-во МИФИ, 2000.
- [Вар96] *Варфоломеев А. А., Домнина О. С., Пеленицын М. Б.* Управление ключами в системах криптографической защиты банковской информации. — М.: Изд-во МИФИ, 1996.
- [Гай94] *Гайкович В., Першин А.* Безопасность электронных банковских систем. — М.: Единая Европа, 1994.

- [Глу99] Глухов М. М. Инъективные отображения слов, не размножающие искажений типа пропуск букв // Дискретная математика. — 1999. — Т. 2. — № 2.
- [Глу98] Глухов М. М. Инъективные отображения слов, не размножающие искажений // Математические вопросы кибернетики. — 1998. — Т. 7.
- [Гне88] Гнеденко Б. В. Курс теории вероятностей. — М.: Наука, 1988.
- [Диф79] Диффи У., Хеллман М. Э. Защищенность и имитостойкость. Введение в криптографию // ТИИЭР. — 1979. — Т. 67. — № 3.
- [Жел96] Жельников В. Криптография от папируса до компьютера. — М.: АБФ, 1996.
- [Каб88] Кабатянский Г. А. Математика разделения секрета // Математическое просвещение. — 1998. — Сер. 3. — Вып. 2. — С. 115—126.
- [Кну99] Кнут Д. Искусство программирования. Т. 2. Полнчисленные алгоритмы. Третье издание. — М.: МЦНМО, 1999.
- [Кон87] Конхейм А. Г. Основы криптографии. — М.: Радио и связь, 1987.
- [Кос87] Сборник задач по алгебре / Под ред. А. И. Кострикина — М.: Наука, 1987.
- [Кра75] Крамер Г. Математические методы статистики. — М.: Мир, 1975.
- [Куз98] Кузьминов Т. В. Криптографические методы защиты информации. — Новосибирск: Наука, 1998.
- [Лид88] Лидл Р., Нидеррайтер Г. Конечные поля. Т. 1, 2. — М.: Мир, 1988.
- [Маф93] Мафтик С. Механизмы защиты в сетях ЭВМ. — М.: Мир, 1993.
- [Мэс88] Мэсси Дж. Л. Современная криптология: введение // ТИИЭР. — 1988. — Т. 76. — № 5.

- [Неч99] *Нечаев В. И.* Элементы криптографии. Основы теории защиты информации. — М.: Высшая школа, 1999.
- [Пер66] *Перельман Я. И.* Занимательная астрономия. — М.: Наука, 1966.
- [Пет00] *Петров А. А.* Компьютерная безопасность. Криптографические методы защиты. — М.: ДМК, 2000.
- [Пит64] *Питерсон У.* Коды, исправляющие ошибки. — М.: Мир, 1964.
- [Ром99] *Романец Ю. В., Тимофеев П. А., Шаньгин В. Ф.* Защита информации в компьютерных системах и сетях. — М.: Радио и связь, 1999.
- [Рос00] *Ростовцев А.* Алгебраические основы криптографии. — СПб.: Мир и семья. — Интерлайн, 2000.
- [Рос01] *Ростовцев А., Маховейко Е.* Введение в криптографию с открытым ключом. — СПб.: Мир и семья — Интерлайн, 2001.
- [Сал96] *Саломая А.* Криптография с открытым ключом. — М.: Мир, 1996.
- [Сим88] *Симмонс Г. Дж.* Обзор методов аутентификации информации // ТИИЭР. — 1988. — Т. 76. — № 5.
- [Соб94] *Соболева Т. А.* Тайнопись в истории России. История криптографической службы России XVIII — начала XX в. — М.: Международные отношения, 1994.
- [Сто01] *Столлингс В.* Криптография и защита сетей. Принципы и практика. — М.: Изд. Дом “Вильямс” 2-е изд., 2001.
- [Хол70] *Холл М.* Комбинаторика. — М.: Мир, 1970.
- [Хоф80] *Хоффман Л.* Современные методы защиты информации. — М.: Радио и связь, 1980.
- [Шен63] *Шеннон К.* Теория связи в секретных системах // В кн.: Работы по теории информации и кибернетике. — М.: ИЛ, 1963.

- [Ягл73] Яглом А.М., Яглом И.М. Вероятность и информация. — М.: Наука, 1973.
- [Яще98] Введение в криптографию / Под ред. В. В. Яценко — М.: МЦНМО — ЧеРо, 1998.
- [Вес97] *Becket B.* Introduction to cryptology and PC security. — McGraw-Hill, London, 1997.
- [Bek82] *Beker H., Piper F.* Ciper systems. The Protection of Communications. — Northwood Books, London, 1982.
- [Bih92] *Biham E., Shamir A.* Differential cryptanalysis of the full 16-round DES // Adv. in Cryptology — CRYPTO'92 Proceedings, 1992. — Spr.-Verl., 1992.
- [Bau39] *Baudouin C.* Èlèments de cryptographie / Èd. Pedone A. — Paris, 1939.
- [Blo83] *Blom R.* Nonpublic key distribution // Advances in Cryptology. — Proceedings of EUROCRYPT'82. Plenum. New York. — 1983.— P. 231 — 236.
- [Brm90] *Burmeister M., Desmedt Y.* A secure and efficient conference key distribution system // Advances in Cryptology — EUROCRYPT '89. LNCS 434. — 1990. — P. 122—133.
- [Bur90] *Burrows M., Abadi M., Needham R.* A Logic of authentication. — ACM Trans. on Comp. Systems. — 1990. — V. 8, n. 1, p. 18 — 36.
- [Cal78] *Callas N. P.* An application of computers in cryptography // Cryptologia, October 1978.
- [Cam90] *Campbell K. W., Wiener M. J.* DES is not a group // Adv. In Cryptology-CRYPTO'90. LNCS 740, 1990.
- [Dif76] *Diffie W., Hellman M. E.* New directions in cryptography // IEEE Trans. on Inf. Theory. — 1976. — IT-22.
- [Dye95] *Dyer M., Fenner T., Frieze A., Thomason A.* On key storage in secure networks // J. Cryptology. — 1995. — n. 8. — P. 189 — 200.

- [ElG85] *El Gamal T.* A public-key cryptosystem and a signature scheme based on discrete logarithms // IEEE Trans. Inf. Theory. — 1985. — IT-31. — № 4.
- [Fei88] *Feige U., Fiat A., Shamir A.* Zero-knowledge proofs of identity // J. Cryptology. — 1988. — n. 1. — P. 77 — 94.
- [Fri20] *Friedman W. F.* The index of coincidence and its applications in cryptanalysis. — Aegean Park Press, Laguna Hills CA, 1920.
- [Fri85] *Friedman W. F., Callimahos D.* Military cryptanalysis. Part I. Vol. 2. — Aegean Park Press, Laguna Hills CA, 1985.
- [Gai40] *Gaines H. F.* Elementary cryptanalysis (A study of ciphers and their solution). — Chapman and Hall, Limited, London, 1940.
- [Gol88] *Gollmann D., Chambers W. G.* Lock-in effect in cascades of clock controlled shift registers // Lect. Notes in Computers Science, v. 30, 1998.
- [God90] *Godlewsky P.* Key-minimal cryptosystems for unconditional secrecy // J. Cryptology. — 1990. — №3.
- [Gol97] *Golic J.* Cryptanalysis of alleged A5 stream cipher // Proceedings of EUROCRYPT'97. LNCS 1233, Springer-Verlag, 1997.
- [Jak95] *Jakobsen T.* A Fast method for cryptanalysis of substitution ciphers // J. Cryptologia. — 1995. — № 3.
- [Kah67] *Kahn D.* The codebreakers. The story of secret writing. — Macmillan, N.Y., 1967.
- [Kur95] *Kurakin V. L., Kuzmin A. S., Mikhalev A. V., Nechaev A. A.* Linear Recurring Sequences over Rings and Modules. I. of Math. Science. Contemporary Math. and it's Appl. Thematic surveys, vol. 10, 1994, I. of Math. Sciences, vol. 76, № 6, 1995.
- [Mat86] *Matsumoto T., Takashima Y., Imai H.* On seeking smart public-key-distribution systems // Trans. of the IECE of Japan. — 1986. — E69. — P. 99 — 106.

- [Men97] *Menezes A J, van Oorschot P C, Vanstone S A* Handbook of applied cryptography. — CRC Press, Boca Raton, New York, London, Tokyo, 1997.
- [Mer81] *Merkle R C, Hellman M E* On the security of multiple encryption // Communications of the ACM — 1981. — Vol. 24.
- [Nee78] *Needham R M, Schroeder M D* Using encryption for authentication in large networks of computers // Communications of the ACM. — 1978. — Vol. 21. — P. 993 — 999.
- [Otw87] *Otway D, Rees O* Efficient and timely mutual authentication // Operating Systems Review. — 1987. — Vol. 21. — P. 8 — 10.
- [Pre91] *Preneel B, Govaerts R, Vandewalle J* Computer security and industrial cryptography State of Art and Evolution // ESAT Course, Leuven, Belgium, May 21—23, 1991. Springer-Verlag, Berlin, 1991.
- [Riv78] *Rivest R L, Shamir A, Adleman L* A Method for obtaining didital signatures and public key cryptosystems // Commun. ACM. — 1978. — V. 21. — № 2.
- [Rue87] *Rueppel R A, Stafflbach O J* Products of linear recurring sequences with maximum complexity // IEEE Trans. Inf. Theory. — 1987. — V. 33 — № 1. - - P 126 — 131.
- [Smi43] *Smith D* Cryptography, the science of secret writing. — 1943.
- [Sch96] *Schneier B* Applied cryptology. — John Wiley & Sons, Inc., 1996.
- [Sha79] *Shamir A* How to share a secret // Commun. ACM. — 1979.—V.22.—No.11.—P.612—613.
- [Sti95] *Stinson D R* Cryptography: Theory and practice. — CRC Press, N.Y., 1995.
- [Ver26] *Vernam G S* Cipher printing telegraph systems for secret write and radio telegraphic communications // J. of the American Institute of Electrical Engineers. — 1926. — Vol. XLV.

- [Yar31] *Yardley H O* The american black chamber. — Bobbs Merrill, Indianapolis, IN, 1931.
- [Zie73] *Zierler N, Mills W H* Products of linear recurring sequences // *J. Algebra* — 1973. — V. 27. — №. 1. — P. 147 — 157.

ОГЛАВЛЕНИЕ

<i>Погорелов Б. А.</i> Вступительное слово	3
Введение	5
Обозначения	7
Глава 1. Исторический очерк развития криптографии	8
Глава 2. Основные понятия	54
§ 2.1. Криптография	54
Конфиденциальность	56
Целостность	62
Аутентификация	63
Цифровая подпись	65
§ 2.2. Управление секретными ключами	68
Предварительное распределение ключей	68
Пересылка ключей	69
Открытое распределение ключей	70
Схема разделения секрета	71
§ 2.3. Инфраструктура открытых ключей	72
Сертификаты	72
Центры сертификации	73
§ 2.4. Формальные модели шифров	74
§ 2.5. Модели открытых текстов	79
Математические модели открытого текста	80
Критерии распознавания открытого текста	83
Глава 3. Классификация шифров по различным признакам	86
§ 3.1. Математическая модель шифра замены	87
§ 3.2. Классификация шифров замены	89
Глава 4. Шифры перестановки	95
§ 4.1. Маршрутные перестановки	95
§ 4.2. Элементы криптоанализа шифров перестановки	98
Глава 5. Шифры замены	101
§ 5.1. Поточные шифры простой замены	101
§ 5.2. Криптоанализ поточного шифра простой замены	105
§ 5.3. Блочные шифры простой замены	113
§ 5.4. Многоалфавитные шифры замены	121
§ 5.5. Дисковые многоалфавитные шифры замены	122
Глава 6. Шифры гаммирования	126
§ 6.1. Табличное гаммирование	126

§ 6.2. О возможности восстановления вероятностей знаков гаммы	129
§ 6.3. Восстановление текстов, зашифрованных неравновероятной гаммой	132
§ 6.4. Повторное использование гаммы	139
§ 6.5. Криптоанализ шифра Виженера	143
§ 6.6. Ошибки шифровальщика	152
Глава 7. Надежность шифров... ..	156
§ 7.1. Энтропия и избыточность языка.....	156
§ 7.2. Расстояние единственности.....	162
§ 7.3. Стойкость шифров	169
Теоретическая стойкость шифров	172
Практическая стойкость шифров	179
§ 7.4. Вопросы имитостойкости шифров	182
§ 7.5. Шифры, не распространяющие искажений	194
Шифры, не распространяющие искажений типа “замена знаков”	195
Шифры, не распространяющие искажений типа “пропуск-вставка знаков”	201
Глава 8. Блочные системы шифрования	205
§ 8.1. Принципы построения блочных шифров.....	206
§ 8.2. Примеры блочных шифров	209
Американский стандарт шифрования данных DES.....	209
Стандарт шифрования данных ГОСТ 28147-89.....	220
§ 8.3. Режимы использования блочных шифров	223
§ 8.4. Комбинирование алгоритмов блочного шифрования.....	230
§ 8.5. Методы анализа алгоритмов блочного шифрования	231
§ 8.6. Рекомендации по использованию алгоритмов блочного шифрования	237
Глава 9. Поточные системы шифрования.....	240
§ 9.1. Синхронизация поточных шифрсистем	240
§ 9.2. Принципы построения поточных шифрсистем	242
§ 9.3. Примеры поточных шифрсистем.....	247
Шифрсистема А5	247
Шифрсистема Гиффорда.....	250
§ 9.4. Линейные регистры сдвига	251
§ 9.5. Алгоритм Берлекемпа—Месси	261
§ 9.6. Усложнение линейных рекуррентных последовательностей	265

Фильтрующие генераторы	265
Комбинирующие генераторы	272
Композиции линейных регистров сдвига	274
Схемы с динамическим изменением закона рекурсии	275
Схемы с элементами памяти	279
§ 9 7 Методы анализа поточных шифров	283
Глава 10 Шифрование в аналоговой телефонии	287
§ 10 1 Особенности речевых сигналов	287
§ 10 2 Скремблирование	290
§ 10 3 Частотные преобразования сигнала	291
§ 10 4 Временные преобразования сигнала	298
§ 10 5 Стойкость систем временных перестановок	305
§ 10 6 Системы цифровой телефонии	307
Глава 11 Системы шифрования с открытыми ключами	310
§ 11 1 Шифрсистема RSA	311
§ 11 2 Шифрсистема Эль-Гамала	318
§ 11 3 Шифрсистема Мак-Элиса	321
§ 11 4 Шифрсистемы на основе “проблемы рюкзака”	323
Глава 12 Идентификация	327
§ 12 1 Фиксированные пароли (слабая идентификация)	328
Правила составления паролей	329
Усложнение процедуры проверки паролей	330
“Подсолненные” пароли	330
Парольные фразы	331
§ 12 2 Атаки на фиксированные пароли	331
Повторное использование паролей	331
Тотальный перебор паролей	332
Атаки с помощью словаря	332
Личные идентификационные номера	333
Одноразовые пароли	334
§ 12 3 “Запрос-ответ” (сильная идентификация)	335
“Запрос-ответ” с использованием симметричных алгоритмов шифрования	337
“Запрос-ответ” с использованием асимметричных алгоритмов шифрования	339
§ 12 4 Протоколы с нулевым разглашением	341
§ 12 5 Атаки на протоколы идентификации	344
Глава 13 Криптографические хэш-функции	347
§ 13 1 Функции хэширования и целостность данных	347

§ 13 2	Ключевые функции хэширования	350
§ 13 3	Бесключевые функции хэширования	354
§ 13 4	Целостность данных и аутентификация сообщений	359
§ 13 5	Возможные атаки на функции хэширования	362
Глава 14	Цифровые подписи	365
§ 14 1	Общие положения	365
§ 14 2	Цифровые подписи на основе шифрсистем с открытыми ключами	369
§ 14 3	Цифровая подпись Фиата-Шамира	371
§ 14 4	Цифровая подпись Эль-Гамала	372
§ 14 5	Одноразовые цифровые подписи	375
Глава 15	Протоколы распределения ключей	378
§ 15 1	Передача ключей с использованием симметричного шифрования	378
	Двусторонние протоколы	378
	Трехсторонние протоколы	381
§ 15 2	Передача ключей с использованием асимметричного шифрования	385
	Протоколы без использования цифровой подписи	385
	Протоколы с использованием цифровой подписи	386
	Сертификаты открытых ключей	386
§ 15 3	Открытое распределение ключей	387
§ 15 4	Предварительное распределение ключей	390
	Схемы предварительного распределения ключей в сети связи	391
	Схемы разделения секрета	398
§ 15 5	Способы установления ключей для конференц-связи	401
§ 15 6	Возможные атаки на протоколы распределения ключей	404
Глава 16	Управление ключами	408
§ 16 1	Жизненный цикл ключей	411
§ 16 2	Услуги, предоставляемые доверенной третьей стороной	414
	Установка временных меток	415
	Нотаризация цифровых подписей	416
Глава 17	Некоторые практические аспекты использования шифрсистем	418
§ 17 1	Анализ потока сообщений	418
§ 17 2	Ошибки операторов	419
		479

§ 17.3. Физические и организационные меры при использовании шифрсистем	420
Глава 18. Квантово-криптографический протокол открытого распределения ключей	423
Квантовый канал и его свойства.....	423
Протокол открытого распределения ключей.....	425
Приложение 1. Открытые сообщения и их характеристики	429
Приложение 2. Пример.....	450
Приложение 3. Элементы алгебры и теории чисел.....	460
Литература	469