ФЕДЕРАЛЬНОЕ АГЕНТСТВО ВОЗДУШНОГО ТРАНСПОРТА
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ГРАЖДАНСКОЙ АВИАЦИИ»

Кафедра основ радиотехники и защиты информации
Э. А. Болелов

КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ
Часть 1. Симметричные криптосистемы

Москва – 2011

Данное учебное пособие издается в соответствии с рабочей программой учебной дисциплины «Криптографические методы защиты информации» по Учебному плану специальности 090106 для студентов дневного обучения.

В учебном пособии изложены основные подходы, математические модели и методы современной симметричной криптографии для решения задач, возникающих при обработке, хранении и передаче информации. Раскрыты методы криптоанализа симметричных криптосистем, рассмотрены основные положения теории стойкости криптосистем.

Рассмотрено и одобрено на заседаниях кафедры 10.02.2011 г. и методического совета 24.02.2011 г.

Содержание

1. Введение в криптографию

Исторический процесс развития средств и методов защиты информации выработал три основных способа защиты.

Первый способ защиты информации – физическая защита от противника материального носителя информации (пергамента, бумаги, магнитной ленты и т.д.), например, передача информации специальным курьером с охраной, перстень с контейнером для тайного послания и т.п.

Второй способ защиты информации – стеганография. Применение стеганографии обеспечивает сокрытие от противника самого факта передачи информации. Стеганографическая защита информации обеспечивается различными способами, например:

- использованием «невидимых» носителей информации (микропленок);
- применением симпатических чернил, которые становятся видимыми при соответствующей химической обработки носителя информации;
- маскированием секретной информации обычным посланием и т.д.

В современной стеганографии имеется достаточно широкий спектр методов защиты информации [1,12].

Третий, наиболее надежный и распространенный способ защиты информации – криптографический. Именно криптографическим методам защиты информации и посвящено данное учебное пособие.

1.1. Основные понятия и определения криптографии

Рассмотрим основные понятия, принятые в криптографии [1,2,8-11], и вначале определим, что такое криптография.

Криптография - это раздел прикладной математики (криптологии), изучающий модели, методы, алгоритмы, программные и аппаратные средства преобразования информации (шифрования) в целях сокрытия ее содержания, предотвращения видоизменения или несанкционированного использования. На решение взаимообратных задач нацелен криптоанализ. **Криптоанализ** - это раздел прикладной математики (криптологии), изучающий модели, методы, алгоритмы, программные и аппаратные средства анализа криптосистем или их входных и выходных сигналов с целью извлечения конфиденциальных параметров, включая открытый текст. Таким образом, криптография и криптоанализ составляют единое целое и образуют науку - **криптологию,** которая с самого начала развивалась как двуединая наука.

Исторически центральным понятием криптографии является понятие шифра. **Шифром** называется совокупность обратимых криптографических преобразований множества открытых текстов на множество зашифрованных текстов, проводимых с целью их защиты. Конкретный вид криптографического преобразования открытого текста определяется с помощью **ключа**

шифрования. **Открытым текстом** называют исходное сообщение, которое подлежит зашифрованию. Под **зашифрованием** понимается процесс применения обратимого криптографического преобразования к открытому тексту, а результат этого преобразования называется **шифртекстом** или **криптограммой**. Соответственно, процесс обратного криптографического преобразования криптограммы в открытый текст называется **расшифрованием.**

Расшифрование нельзя путать с дешифрованием. **Дешифрование** (**дешифровка, взлом**) - процесс извлечения открытого текста без знания криптографического ключа на основе перехваченных криптограмм. Таким образом, расшифрование проводится законным пользователем, знающим ключ шифра, а дешифрование - криптоаналитиком.

Криптографическая система - семейство преобразований шифра и совокупность ключей. Само по себе описание криптографического алгоритма не является криптосистемой. Только дополненное схемами распределения и управления ключами оно становится системой.

Классификация криптосистем представлена на рис. 1.1.

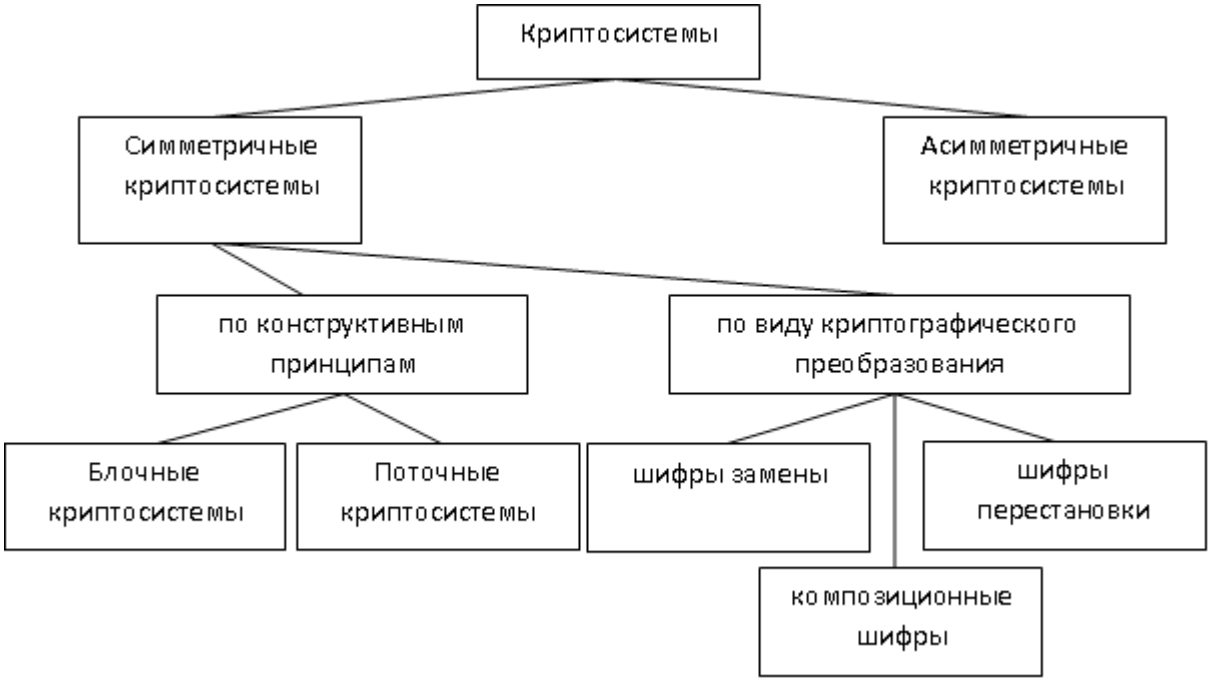

Рис. 1.1. Классификация криптосистем

Более полная классификация криптосистем приведена, например в [1,10].

Симметричные криптосистемы (криптосистемы с секретным ключом) построены на принципе сохранения в тайне ключа шифрования. На рис. 1.2 представлена упрощенная структурная схема симметричной криптосистемы. Перед использованием симметричной криптосистемы пользователи должны получить общий секретный ключ k и исключить доступ

к нему злоумышленника. Открытое сообщение X подвергается криптографическому преобразованию $f_k(X)$ и полученная криптограмма Y по открытому каналу связи передается получателю, где осуществляется обратное преобразование $f_k^{-1}(Y)$ с целью выделения исходного открытого сообщения X.

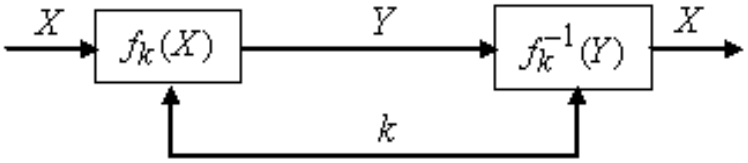

Рис. 1.2. Упрощенная структурная схема симметричной криптосистемы

Симметричные криптосистемы классифицируются по различным признакам [1,2,10]: по виду криптографического преобразования; по конструктивным принципам; по виду защищаемой информации; по криптографической стойкости и т.д. Чаще всего используются первые два признака классификации. В связи с этим множество симметричных криптосистем делится:

- по виду криптографического преобразования – на шифры перестановки, шифры замены и композиционные шифры;
- по конструктивным принципам – на поточные криптосистемы и блочные криптосистемы.

Под **шифром перестановки** понимается переупорядочение букв исходного сообщения, в результате которого он становиться нечитаемым. Под **шифром замены** понимается преобразование, которое заключается в замене букв исходного сообщения на другие буквы по более или менее сложному правилу. **Композиционные шифры** строятся на основе шифров замены и перестановки. **Блочные симметричные криптосистемы** (БСК) представляют собой семейство обратимых криптографических преобразований блоков исходного сообщения. **Поточные криптосистемы** (ПСК) преобразуют посимвольно исходное сообщение в криптограмму.

Отличительной особенностью **асимметричных криптосистем** (**криптосистем с открытым ключом**) является то, что для зашифрования и расшифрования информации используются разные ключи. На рис. 1.3 представлена упрощенная структурная схема асимметричной криптосистемы. Криптосистема с открытым ключом определяется тремя алгоритмами: генерации ключей, шифрования и расшифрования. Алгоритм генерации ключей позволяет получить пару ключей $(k_o, k_з)$, причем $k_o \neq k_з$. Один из ключей k_o публикуется, он называется **открытым**, а второй $k_з$, называется **закрытым** (или секретным) и храниться в тайне. Алгоритмы шифрования $f_{k_o}(\cdot)$ и

расшифрования $f_{k_з}^{-1}(\cdot)$ таковы, что для любого открытого текста X выполняется равенство $f_{k_з}^{-1}(f_{k_o}(X)) = X$.

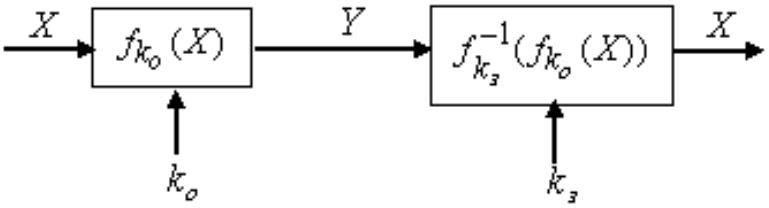

Рис. 1.3. Упрощенная структурная схема асимметричной криптосистемы

1.2. Из истории криптографии

По мнению ряда специалистов, криптография по возрасту – ровесник египетских пирамид. В документах древних цивилизаций (Индии, Египта, Месопотамии) есть сведения о системах и способах составления шифрованных писем.

В криптографии с древних времен использовались два вида шифров: замены (подстановки) и перестановки. Историческим примером шифра замены является **шифр Цезаря** (I век до н.э.), описанный историком Древнего Рима Светонием. Гай Юлий Цезарь использовал в своей переписке шифр собственного изобретения. Применительно к русскому языку он состоит в следующем. Выписывается алфавит, а затем под ним выписывается тот же алфавит, но с циклическим сдвигом на три буквы влево:

А Б В Г Д Е … Э Ю Я
Г Д Е Ё Ж З … А Б В

Зашифрование заключается в выборе буквы из первой строки и замену ее на букву второй строки, расшифрование представляет собой обратную операцию. Например, РИМ – УЛП. Ключом шифра Цезаря является величина циклического сдвига. Гай Юлий Цезарь всю жизнь использовал один и тот же ключ – сдвиг на 3 буквы. Приемник Юлия Цезаря – Цезарь Август использовал тот же шифр, но со сдвигом на одну букву. Светоний не приводит фактов дешифрования шифра Цезаря, однако в те времена, когда царила всеобщая неграмотность, даже обычное открытое послание могло остаться непрочитанным.

Одним из первых физических приборов, реализующих шифр перестановки является **скитала.** Он был изобретен в древней Спарте (V век до н.э.). Кроме Древней Греции прибор скитала использовался широко и в Древнем Риме. Скитала (в переводе - «жезл») представляет собой цилиндр заданного диаметра. На цилиндр наматывался ремень из пергамента, на который наносился текст сообщения вдоль оси цилиндра. Затем ремень сматывался и отправлялся получателю сообщения. Последний, имея

аналогичный цилиндр, расшифровывал сообщение. Ключом шифра является диаметр скитала. Изобретение дешифровального устройства приписывается Аристотелю. Он предложил использовать для дешифрования конусообразное «копье», на которое наматывался перехваченный ремень, до тех пор, пока не появлялся осмысленный текст.

Одним из первых исторических имен, которое упоминается в связи с криптографией, это имя Энея - легендарного полководца, защитника Трои. В области тайнописи Энею принадлежат два изобретения. Первое из них – так называемый **диск Энея**. Его принцип прост. На диске размером 10-15 см и толщиной 1-2 см высверливались отверстия по числу букв алфавита. В центре диска закреплена катушка с нитью. При зашифровании нитка последовательно протягивалась через отверстия соответствующие буквам послания. Диск отсылался получателю, который вытягивал нитку из отверстий и получал сообщение в обратном порядке. Другим устройством является **линейка Энея.** Здесь вместо диска использовалась линейка с числом отверстий, равным числу букв в алфавите. Буквы по отверстиям располагались в произвольном порядке. К линейке прикреплялась катушка с нитью. При шифровании нить протягивалась через отверстие, соответствующее букве шифруемого послания, при этом на нити в месте прохождения отверстия завязывался узелок. Таким образом, зашифрованное послание представляло собой нить с узелками, в которой каждой букве ставилось в соответствие расстояние между узелками нити. Ключом шифра являлся порядок следования букв по отверстиям линейки. Аналогичное линейке Энея **кипу** (узелковое письмо) получило широкое распространение у индейцев Центральной Америки.

Еще оно изобретение древних греков – квадрат Полибия (Полибий – греческий государственный деятель, полководец, историк III века до н.э):

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

Применительно к современному латинскому алфавиту шифрование по этому квадрату заключалось в следующем. Шифруемая буква заменялась на координаты квадрата, в котором она записана. Так буква R заменяется на DB. При расшифровании каждая пара букв определяет соответствующую букву сообщения. Например, TABLE – DDAAABCAAE. Ключом этого шифра является сам квадрат. Усложненный вариант квадрата Полибия заключается в произвольном порядке записи букв в квадрате. При этом для запоминания такого произвольного порядка использовался лозунг, который представлял собой слово, записываемое без повтора букв в квадрат, а оставшиеся клетки

квадрата заполнялись по порядку их следования остальными буквами алфавита. Например, THE APPLE соответствует THEAPL.

Интересно отметить, что в несколько измененном виде квадрат Полибия дошел до наших дней и получил название «тюремный шифр». Для его использования достаточно знать только естественный порядок букв в алфавите. Стороны квадрата обозначаются не буквами, а цифрами. Каждая цифра кодируется определенным количеством стуков. При передаче сообщения сначала «отстукивается» номер строки, а затем номер столбца. «Тюремный шифр» строго говоря, не является шифром, это способ кодировки сообщения с целью его приведения к виду удобному для передачи по каналу связи (тюремная стена).

Во времена средневековья европейская криптография приобрела сомнительную славу, отголоски которой слышаться и в наши дни. Дело в том, что криптографию стали отождествлять с черной магией, астрологией, алхимией, к шифрованию призывались мистические силы. Для шифрования сообщений рекомендовалось использовать **«магические квадраты»**. Магия этих квадратов заключалась в том, что сумма чисел по строкам, столбцам и полным диагоналям равнялась одному числу. Шифрование по «магическому квадрату» заключалось в следующем. Буквы сообщения вписывались в квадрат согласно записанным в них числам, а в пустые клетки вставлялись произвольные буквы. Шифртекст выписывался в оговоренном заранее порядке. Например, сообщение ПРИЕЗЖАЮ СЕГОДНЯ зашифрованное с помощь «магического квадрата»:

16У	**3**И	**2**Р	**13**Д
5З	**10**Е	**11**Г	**8**Ю
9С	**6**Ж	**7**А	**12**О
4Е	**15**Я	**14**Н	**1**П

имеет вид УИРДЗЕГЮСЖАОЕЯНП. Данный шифр – обычный шифр перестановки, однако считалось, что особую стойкость ему придает волшебство «магического квадрата».

В XV веке аббат Тритемий сделал два новаторских предложения в области криптографии: он предложил шифр «Аве Мария» и шифр, основанный на периодически сдвигаемом ключе. Наиболее серьезное предложение Тритемия, дошедшее до наших дней, заключается в придуманной им таблице:

A	B	C	D	…	W	X	Y	Z
B	C	D	E	…	X	Y	Z	A
C	D	E	F	…	Y	Z	A	B
…	…	…	…	…	…	…	…	…
Y	Z	A	B	…	U	V	W	X
Z	A	B	C	…	V	W	X	Y

Первая буква текста шифруется по первой строке, вторая буква по второй строке и так далее. Первая строка одновременно является строкой букв

открытого текста. Например, FIGHT – FJIKX. В первоначальном варианте в шифре Тритемия отсутствовал ключ. Секретом являлся сам способ шифрования. Дальнейшее усложнение шифра шло двумя путями: введением произвольного порядка расположения букв в таблице; усложнением порядка выбора строк таблицы при шифровании. Следует сказать, что шифр Цезаря является частным случаем шифра Тритемия.

Шифр «Аве Мария» основан на принципе замены букв шифруемого текста на целые слова, из которых составлялись внешне невинные сообщения. Например, Н – «Я», «ЗДЕСЬ»; Е – «ЖДУ», «БУДУ»; Т – «ДОМА», «ВЕЧЕРОМ». Тогда открытому сообщению НЕТ могут соответствовать послания «Я ЖДУ ДОМА», «ЗДЕСЬ БУДУ ВЕЧЕРОМ».

В XVII веке английский философ и ученый лорд-канцлер Френсис Бэкон выдвинул главные требования к шифрам: «Они не должны поддаваться дешифрованию, не должны требовать много времени для написания и чтения, не должны возбуждать никаких подозрений». Эти требования актуальны и сегодня.

Широко использовали шифры и братства «вольных каменщиков» (масонов). **Шифр «вольных каменщиков»** является шифром замены и вопреки распространенному мнению не является стойким, но представляет определенный интерес. Шифрование заключается в замене букв открытого текста символами по правилу:

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

Например, APPLE соответствует криптограмме вида:

:	.	.	.	:

При походе на Россию Наполеон использовал шифр «вольных каменщиков» в низших звеньях своей связи, однако шифр достаточно быстро был раскрыт русскими дешифровальщиками.

В XVI веке итальянец Альберти впервые выдвинул идею двойного шифрования, т.е. текст после первого шифрования подвергался повторному шифрованию. Альберти также принадлежит шифр, который он называл «шифром, достойным королей». Реализация шифра осуществлялась с помощью шифровального диска. На внешний неподвижный диск наносились буквы и цифры под которыми располагались буквы и цифры внутреннего подвижного диска. Процесс шифрования прост – буквам и цифрам открытого текста ставились в соответствие буквы и цифры внутреннего диска. После зашифровывания слова послания внутренний диск сдвигался на один шаг. Начальное положение дисков заранее оговаривалось. Диск Альберти с незначительными изменениями использовался вплоть до начала XX века.

В XVI веке заметный вклад в развитие криптографии внесли Матео Ардженти, Жовани Батиста Белазо, Джовани Батиста Порта, Кордано и др.

Матео Ардженти был криптографом папы римского, именно ему принадлежит идея использования слова-лозунга для придания алфавиту легко запоминаемого смешанного вида. Ардженти также предложил вставлять в шифртекст большое количество букв «пустышек», устранять пунктуацию, не вставлять в шифртекст открытые слова («клер»), заменять буквы шифртекста на цифры. Белазо и Порта развили идеи Ардженти в своих трудах «Шифр сеньора Белазо» и «О тайной переписке».

Существенный вклад в развитие криптографии внес математик, врач и философ Кордано. Предложенный им шифр вошел в историю под названием «решетка Кордано». **«Решетка Кордано»** - это шифр перестановки, суть которого заключается в следующем. Брался лист плотного материала (картон, пергамент), представляющий собой квадрат в котором вырезаны «окна». При шифровании квадрат накладывался на лист бумаги и сообщение вписывалось в «окна», затем квадрат поворачивался на 90 градусов и сообщение продолжали записывать в «окна» повернутого квадрата. Такая процедура продолжалась до полного поворота квадрата на 360 градусов. Главное требование «решетки Кордано» - при всех поворотах «окна» не должны попадать на дно и тоже место, а при полном повороте квадрата все места в шифртексте оказываются занятыми. Шифртекст считывался по строкам из полученной таблицы. Предложенный Кордано шифр лежит в основе знаменитого **шифра Ришелье**, в котором шифрованный текст внешне имел вид обычного послания. Накладывая на лист с таким посланием прямоугольник прорезанными с окнами можно было прочесть сообщение. Шифр Ришелье не относиться ни к шифрам замены, ни к шифрам перестановки, он представлял собой стеганографический способ защиты информации. Такого рода шифром пользовался русский писатель и государственный деятель А.С. Грибоедов будучи послом в Персии.

Кордано выдвинул, но не успел целиком реализовать идею «самоключа». Суть ее заключается в использовании в качестве ключа части открытого сообщения.

Познакомившись в трудами Тритемия, Белазо, Кордано и Альберти французский государственный деятель Блез де Виженер разработал собственный шифр, который получил название **шифр Виженера**. Суть шифра заключалась в том, что выбирался секретное слово, которое являлось ключом шифра. Это слово выписывалось под открытым сообщением периодически. Верхняя буква открытого текста соответствовала столбцу таблицы Тритемия, а нижняя буква ключа – строке таблицы Тритемия, буква, стоящая на пересечение строки и столбца являлась буквой шифртекста. Шифр Виженера представляет собой шифр замены. В последующем этот шифр был несколько упрощен для практического использования начальником первого в Германии государственного дешифровального отдела графом Гронсфельдом. Шифр Виженера и шифр Гронсфельда являются по сути дела родоначальниками широко используемого в настоящее время шифра гаммирования. Шифр

Виженера использовался в различных вариантах вплоть до XIX века. Одним из наиболее известных модификаций шифра Виженера является шифр английского адмирала Бофора. Достоинство шифра Бофора заключается в том, что правило зашифрования сообщений и их расшифрования совпадают.

Широкое развитие криптографии в XVI веке было связано в развитие естественных наук, математики. В это же время в Европе появляются первые специальные органы дипломатической службы, которые занимались вопросами шифрования собственной корреспонденции и дешифрования перехваченной корреспонденции. XVII-XVIII века вошли в историю криптографии как эра «черных кабинетов». **«Черные кабинеты»** - специальный государственный орган по перехвату, перлюстрации и дешифрованию переписки, в первую очередь дипломатической. В штат «черных кабинетов» входили дешифровальщики, агенты по перехвату почты, писцы-копировальщики, переводчики, специалисты по подделке печатей, химики, специалисты по подделке почерков и т.д. Эти специалисты ценились весьма высоко и находились под особым покровительством властей, предательство очень сурово наказывалось.

В XIX веке появляются первые механические шифровальные устройства. Наиболее известными являются изобретения полковника американской армии Д. Уодсворта и английского инженера Ч. Уитстона. Устройство Уодсворта (1817 г.) представляло механический шифратор основными элементами которого были два шифровальных диска, на торце дного располагались буквы английского алфавита, а на торце второго буквы и цифры от 2 до 8. Литеры на втором диске били съемные, что позволяло менять алфавит шифрованного текста. Диски помещались в футляр с прорезанными в нем окнами. При вращении первого диска в верхнем окне выставлялась буква открытого сообщения. Диски были соединены шестеренчатой передачей, поэтому в нижнем окне появлялась соответствующая буква шифртекста. Устройство было снабжено специальной кнопкой для разъединения дисков. Это требовалось для того, чтобы обеспечивать установку устройства в заданное начальное положение. В устройстве Уодсворта просматриваются идеи Альберти, Тритемия, Виженера. Несмотря на то, что устройство было достаточно громоздким, к тому же в это время господствовали «ручные» шифры, которые не требовали специальных приспособлений, оно послужило толчком к развитию механических устройств для шифрования и расшифрования сообщений.

Интересное предложение по созданию механического устройства шифрования сделал Ч. Уитстон во второй половине XX века. В устройстве Уитстона просматриваются идеи Альберти, а также Уодсворта. Внешне устройство Уитстона напоминает диск Альберти, однако в нем реализована парадоксальная идея – алфавит открытого текста содержит большее количество знаков, чем шифрованного. Проблема неоднозначности в определении букв

открытого сообщения решена Уитстоном блестяще. На рис. 1.4 представлен внешний вид устройства Уитстона.

Внешний диск, диск алфавита открытого текста, состоял из 27 знаков (26 букв английского алфавита и специального знака "+", означающего пробел). Внутренний алфавит определяет алфавит открытого текста и состоит из обычных 26 букв, расположенных в произвольном ключевом порядке. На той же оси, что и диски (алфавиты) устройства, соединенные шестернями размером 27×26 соответственно, расположены две стрелки, как в современных часах.

Рис. 1.4. Внешний вид устройства Ч. Уитстона

В начале шифрования большая (длинная) стрелка указывает на знак "+". Малая стрелка, связанная с большой резьбовой шестеренкой, ставилась в то же положение, т.е. "часы" показывали "12.00". Набор букв открытого текста производился поворотом большой стрелки по направлению движения часовой. После такого поворота малая стрелка указывает знак шифрованного текста. Таким образом, при полном повороте большого диска малый диск смещался на единицу по отношению к исходному взаимному состоянию двух дисков, что приводило к сдвиговому изменению алфавита шифрованного текста по отношению к алфавиту открытого текста. По окончании каждого слова большая стрелка становилась на знак "+", буква, на которую при этом указывала короткая стрелка, записывалась как знак шифрованного текста. Во избежание неоднозначности расшифрования, удвоение букв в открытом тексте не допускается. Повторную букву следует либо пропустить, либо ставить вместо нее какую-нибудь редкую букву, например Q. Например, слово THE APPLE при шифровании записывается как +THE+APLE+ или +THE+APQLE+.

Изобретение Уитстона, также как и Уодсворта, не нашло широкого применения. Однако судьба другого его предложения в области криптографии - шифра биграммной замены - сложилась лучше, хотя шифр несправедливо был назван именем друга изобретателя барона Плейфера. Вместе с тем, сам Плейфер вел себя весьма корректно: популяризируя изобретение, он всегда указывал имя автора – Уитстона, но история распорядилась иначе: шифру было

присвоено имя не изобретателя, а популяризатора. Шифр Плейфера будет подробно рассмотрен в следующем разделе.

В начале XX века значительный вклад в развитие криптографии внес американец Г. Вернам. В 1917 году он, будучи сотрудником телеграфной компании, предложил идею автоматического шифрования телеграфных сообщений, суть которой заключается в следующем. Открытый текст представляется в коде Бодо (в виде пятизначных "импульсных комбинаций"). В этом коде, например, буква "А" имела вид (+ + — — —). На бумаге знак "+" означал отверстие, а знак "-" - его отсутствие. При считывании с ленты пятерка металлических щупов "опознавала" отверстия (при наличии отверстия щуп замыкал электрическую цепь). В линию связи посылались импульсы тока. Вернам предложил электромеханически покоординатно складировать импульсы знаков секретного текста с импульсами секретного ключа, представляющего из себя хаотический набор букв того же самого алфавита. Сложение, по современной терминологии, осуществлялось по модулю 2. Г. Вернам создал устройство, производящее операции шифрования автоматически, без участия шифровальщика, тем самым было положено начало так называемому "линейному шифрованию", когда процессы шифрования и передачи сообщения происходят одновременно. До той поры шифрование было предварительным, поэтому линейное шифрование существенно повышало оперативность связи. Шифр Вернама обладает исключительной криптографической стойкостью. В то же время очевиден и недостаток этой системы шифрования - ключ должна иметь ту же длину, что и открытый текст. Для расшифрования на приемном конце связи туда нужно передать (по тайным, защищенным каналам) ключ достаточной длины. При практической реализации это порождает проблемы, причем весьма существенные, что и предопределило скромное распространение шифров Вернама. Сам Вернам не был математиком-криптографом, тем не менее, он настаивал на том, что ключ шифра не должен повторяться при шифровании, и в этом, как показала история криптографии, он был прав. Его идеи породили новые подходы к надежной защите информации при передаче больших объемов сообщений.

Первая половина XX века стала «золотым веком» электромеханических шифровальных машин [11,12]. Наибольшую известность получило семейство немецких электромеханических шифровальных машин Enigma. Различные модификации этой шифровальной машины использовались германскими войсками с конца 1923 года вплоть до 1945 года. В 1943 году союзникам по антигитлеровской коалиции удалось «взломать» машину Enigma, что сыграло большую роль в победе во Второй мировой войне. Для передачи наиболее секретных сообщений во время Второй мировой войне немцами использовалась шифровальная машина Lorenz. В американской армии с 1923 по 1943 год использовалась механическое устройство для шифрования М-94. В основу этого устройства положен диск Альберти. Для защиты дипломатической

переписки в США использовалась машина Хеберна MarkII. Шведский криптограф Б. Хагелин разработал для французской секретной полиции шифровальное устройство CD-57, а для французских спецслужб – шифровальную машину М-209. Модификация этой машины использовалась также и американскими военными во Второй мировой войне. С 1939 года по 1952 год японцы использовали шифровальную машину для защиты дипломатической переписки под названием «Тип 97» и ее модификацию. В США эти машины получили красочное обозначение «Пурпурный код» и «Красный код». В СССР перед войной и в годы Великой Отечественной войны широко использовалась малогабаритная дисковая кодировочная машина К-37 «Кристалл». Только в 1940 году было выпущено 100 комплектов этой машины. После войны были подведены итоги эксплуатации К-37 и проводилась работа по ее дальнейшему совершенствованию.

Уже к началу 1930-х годов сформировались разделы математики (теория чисел, теория вероятностей и математическая статистика) являющиеся основой будущей науки – криптологии. Ключевой вехой в развитии криптографии является фундаментальный труд Клода Шеннона «Теория связи в секретных системах», написанный в форме секретного доклада в 1945 году и опубликованный в 1949 году [5,11]. В этой работе впервые был показан подход к криптографии как к математической науке. Были сформулированы ее теоретические основы и введены понятия, с объяснения которых сегодня начинается изучение криптографии.

Развитие во второй половине XX века компьютерной техники и электроники сделало возможным использование более сложных шифров. В 1960-х годах появляются первые блочные шифры, обладающие большей криптостойкостью, чем электромеханические машины. В 1976 году в США принимается государственный стандарт шифрования – DES (Data Encryption Standart), являющийся первым в мире открыто опубликованным стандартом шифрования. На основе используемой в системе DES сети Хорста Фейстеля разработаны множество других криптосистем: российский стандарт ГОСТ 28147-89, криптосистема TEA (Tiny Encryption Algorithm), Twofish, IDEA (International Data Encryption Algorithm).

В 1975 году публикуется работа Уитфилда Диффи и Мартина Хеллмана «Новые направления в криптографии» [2,8,10,11]. Данная работа открыла новую область криптографии, теперь называемую криптографией с открытым ключом. Хотя работа У. Диффи и М. Хеллмана создала большой теоретический задел для открытой криптографии, первой реальной криптосистемой с открытым ключом считают криптосистему RSA, названную по имени авторов - Рона Ривеста, Ади Шамира и Леонарда Адлемана. В настоящее время существует большое разнообразие криптосистем с открытым ключом, некоторые из которых являются национальными стандартами шифрования.

Таким образом, используя в качестве критерия периодизации технологические характеристики методов шифрования, историю криптографии можно разделить на следующие периоды.

1. Первый период (приблизительно с 3 тысячелетия до н.э. до XV века) характеризуется господством моноалфавитных шифров.

2. Второй период (с XV века по XX век) ознаменован введением многоалфавитных шифров.

3. Третий период (с начала и до середины XX века) характеризуется внедрением электромеханических шифровальных машин.

4. Четвертый период (с середины до 70-х годов XX века) – период перехода к математической криптографии.

5. Современный период (с конца 70-х годов XX века по настоящее время) характеризуется широким использованием криптографии частными лицами коммерческими организациями, что связано с развитием нового направления в криптографии – криптографии с открытым ключом.

1.3. Модели источников открытых текстов

Открытый текст, также как и криптограмма, представляет собой последовательность символов, взятых из конечного набора, называемого **алфавитом** A. Элемент алфавита называется **буквой**, а число символов алфавита определяет его **мощность** $|A| = m$. Например, алфавит простых букв английского языка имеет мощность $|A| = 26$, а алфавит английского языка, состоящий из прописных, строчных букв, цифр, а также знаков препинания и пробела имеет мощность $|A| = 70$. Широко используемыми алфавитами, элементы которых есть двоичные векторы, являются коды ASCII и МТК-2.

Всякий открытый текст, записанный в некотором алфавите, имеет **длину** n, равную числу букв в соответствующей записи. Последовательность k соседних букв текста, при $k \geq 2$ называется k-**граммой**, а при $k = 2$ - **биграммой**. Помимо исходного алфавита, часто рассматриваются производные из него алфавиты, представляющие наборы всевозможных k-грамм исходного алфавита. Таким образом, каждый открытый текст характеризуется: набором используемых алфавитов; длиной текста; тематикой открытых текстов.

Модели открытых текстов разделяются на два класса: детерминированные и вероятностные [11].

1.3.1 Детерминированные модели

Источники открытых сообщений (ИОС) достаточно многообразны. В качестве ИОС рассматривать можно отдельного человека или группу людей, пункты телеграфной и телефонной сети и т.д. Каждый ИОС порождает тексты в соответствии с правилами грамматики используемого языка, что находит отражение и в статистических характеристиках открытых текстов. Всякий ИОС

можно характеризовать разбиением множества k-грамм на **допустимые** (встречающиеся в текстах) и **запрещенные** (не встречающиеся в текстах). Например, в русском языке буквы Ь и Ъ никогда не соседствуют друг с другом, и не следуют за гласными буквами. Разбиение множества k-грамм на допустимые и запрещенные определяет детерминированную модель. В детерминированной модели открытый текст рассматривается как последовательность букв некоторого алфавита, не содержащий запрещенных k-грамм. Необходимо отметить, что разделение на допустимые и запрещенные k-граммы весьма условно в силу динамичности языка, его способности к развитию. Кроме этого, разделение на допустимые и запрещенные k-граммы характеризует не только язык, но и отдельный источник сообщений.

1.3.2 Вероятностные модели

В вероятностных моделях ИОС рассматривается как источник случайных последовательностей. Пусть ИОС генерирует в заданном алфавите A_X текст конечной или бесконечной длины. При этом можно считать, что источник генерирует конечную или бесконечную последовательность случайных букв $x_0, x_1, x_2, ..., x_i, ...$, принимающих значение в A_X.

Вероятность случайного сообщения $a_0, a_1, a_2, ..., a_{n-1}$ определяется как вероятность последовательности событий:

$$P(a_0, a_1, ..., a_{n-1}) = P(x_0 = a_0, x_1 = a_1, ..., x_{n-1} = a_{n-1}). \quad (1.1)$$

Множество случайных текстов образует вероятностное пространство, если выполнены условия:

1) $P(a_0, a_1, a_2, ..., a_{n-1})$ для любого случайного сообщения $a_0, a_1, a_2, ..., a_{n-1}$;

2) $\sum_{a_0, a_1, a_2, ..., a_{n-1}} P(a_0, a_1, a_2, ..., a_{n-1}) = 1$;

3) для любого случайного сообщения $a_0, a_1, a_2, ..., a_{n-1}$ и любого $s > n$ справедливо $P(a_0, a_1, a_2, ..., a_{n-1}) = \sum_{a_s, ..., a_{n-1}} P(a_0, a_1, a_2, ..., a_{s-1})$, то есть вероятность всех продолжений текста длины n есть сумма вероятностей этого сообщения до длины s. Текст, порождаемый таким ИОС, является вероятностным аналогом языка. Задавая определенное вероятностное распределение на множестве открытых текстов задается соответствующая модель ИОС.

Различают стационарные и нестационарные ИОС [11]. Для **стационарных моделей** характерно то, что вероятность появления буквы (k-граммы) не зависит от места в открытом тексте. Рассмотрим наиболее используемые стационарные модели ИОС.

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

$$P(a_0, a_1, ..., a_{n-1}) = \prod_{i=0}^{n-1} P(x_i = a_i), \qquad (1.2)$$

где для всех $i \in \{0,1,...,n-1\}$ и любого $a \in A_X$ $P(x_i = a) > 0$, $\sum_{a \in A_X} P(x_i = a) = 1$.

Открытый текст такого источника является реализацией последовательности независимых испытаний в полиномиальной вероятностной схеме с числом исходов равным m. Множество исходов взаимно-однозначно соответствует множеству всех букв алфавита. Данная модель позволяет разделить буквы алфавита на классы высокой, средней и низкой частоты использования. Например, в русском языке редкими буквами являются Ф, Э, Щ, Ц, Х. В криптографии существует мнемоническое правило, которое позволяет запомнить десять наиболее часто встречаемых букв в русском языке – СЕНОВАЛИТР, причем буквы располагаются не в соответствии с частотой их использования. Необходимо заметить, что частота встречаемости букв в тексте зависит от тематики текста. Так для литературных произведений и научно-технических текстов значение частот встречаемости букв будет различным. Например, в математических текстах частота встречаемости буквы Ф будет выше, чем в литературных произведениях. Таким образом, частотная диаграмма является достаточно устойчивой характеристикой открытого текста.

Рассмотренная модель удобна для практического использования, в то же время некоторые свойства модели противоречат свойствам языка. В частности, согласной этой модели любая k-грамма имеет ненулевую вероятность использования. Вышесказанное не позволяет применять данную модель для дешифрирования широкого класса криптосистем.

Стационарный источник независимых биграмм. Эта модель более громоздка, но точнее отражает свойства языка. Открытый текст такого ИОС является реализацией последовательности независимых испытаний в полиномиальной вероятностной схеме с числом исходов, равным m^2.Множество исходов соответствует множеству всех биграмм алфавита. Модель характеризуется равенством:

$$P(a_0, a_1, ..., a_{2n-1}) = \prod_{i=0}^{n-1} P(x_{2i} = a_{2i}, x_{2i+1} = a_{2i+1}), \qquad (1.3)$$

где для всех $i \in \{0,1,...,n-1\}$ и любых $a,b \in A_X$ $P(x_{2i} = a, x_{2i+1} = b) > 0$, $\sum_{a,b \in A_X} P(x_{2i} = a, x_{2i+1} = b) = 1$.

В качестве оценок вероятностей биграмм используются относительные частоты их появления, которые вычисляются экспериментально на большом текстовом материале. Вероятности биграмм в алфавите A_X могут быть сведены в матрицу размера $m \times m$, где p_{ij}, $i = \overline{1,m}$, $j = \overline{1,m}$, есть вероятность размещения биграммы (i, j) на случайно выбранной позиции в открытом тексте. Например, в английском языке все биграммы $(q,...)$ за исключением (q,u) имеют нулевую вероятность. Данная модель точнее по сравнению с предыдущей отражает особенности языка и ИОС. В то же время моделью игнорируются зависимости между соседними биграммами. В меньшей степени указанный недостаток присущ следующей модели.

Стационарный источник марковски зависимых букв. Открытый текст такого ИОС является реализацией последовательности испытаний, связанных простой однородной цепью Маркова с m состояниями. Данная модель полностью определяется матрицей вероятностей переходов $\mathbf{\Pi} = \|\pi(i \mid j)\|$, $0 \le j$, $s < m$, вектором $\mathbf{P}_0 = \|p_i\|$, $i = \overline{0, m-1}$ начального распределения вероятностей, где p_i - вероятность $i-$й буквы на первой позиции в открытом тексте.

Вероятность случайного сообщения $a_0, a_1, a_2, ..., a_{n-1}$ определяется выражением:

$$P(a_0, a_1, ..., a_{2n-1}) = p(a_0)\pi(a_1 \mid a_0)\pi(a_2 \mid a_1)...\pi(a_{n-1} \mid a_{n-2}). \qquad (1.4)$$

Согласно данной модели всякое сообщение, содержащее на какой-либо позиции запрещенную биграмму, имеет нулевую вероятность. На основании рассмотренных моделей можно предположить их усложнения в направлении увеличения глубины зависимости вероятности очередной буквы текста от значений вероятности нескольких предыдущих букв. Здесь можно выделить два типа моделей ИОС: стационарный источник независимых k-грамм и стационарный источник зависимых k-грамм.

Для моделей первого типа всякое сообщение является реализацией последовательности испытаний в полиномиальной вероятностной схеме с числом исходов, равным m^k. Эти модели адекватно отражают межсимвольные зависимости внутри каждой k-граммы, но игнорируют межсимвольные зависимости соседних k-грамм. Чем больше k, тем, с одной стороны, точнее рассматриваемые модели отражают свойства языков и ИОС и, с дугой стороны, тем они более громоздки и трудоемки и трудоемки в применении.

Модели второго типа способны учесть межсимвольные зависимости соседних k-грамм, однако они еще более громоздки, чем модели первого типа. Так, если модели первого типа описываются m^k-мерной матрицей вероятностей, то модели второго типа – матрицей размерностью $m^k \times m^k$ и m^k–мерным вектором начального распределения.

Выбор конкретной модели носит компромиссный характер и осуществляется с учетом свойств конкретной криптосистемы.

В **нестационарных моделях** вероятности появления k-грамм в тексте зависят от их места в тексте. Нестационарные модели можно рассматривать, как уточнение стационарных, в которых в той или иной мере учтена структура сообщения.

1.4. Модели шифров

Одним из первых ввел и исследовал математическую модель шифра К. Шеннон [1,5,12]. Пусть X, K, Y некоторые конечные множества, соответственно множество открытых текстов, множество ключей и множество криптограмм. Задано отображение:

$$f : X \times K \to Y. \qquad (1.5)$$

Введенная тройка множеств X, K, Y с функцией (1.5) представляет собой **алгебраическую модель шифра**. При этом должны выполняться следующие условия:

- функция (1.5) сюръективна (осуществляет отображение на Y);
- для любого $k \in K$ функция (1.5) инъективна (образы двух различных элементов различны);
- $|X| \le |Y|$.

Выражение (1.5) представляет собой выражение шифрования. Выражение расшифрования имеет вид:

$$f^{-1} : Y \times K \to X. \qquad (1.6)$$

Требование инъективности отображения (1.5) равносильно требованию возможности однозначного расшифрования криптограммы. Требование сюръективности отображения (1.5) не играет существенной роли и оно обычно вводится для устранения некоторых технических, с точки зрения математики, неудобств, т.е. для упрощения изложения. Алгебраическая модель шифра отражает лишь функциональные свойства шифрования и расшифрования в классических (симметричных) системах шифрования. В этой модели открытый текст (криптограмма) является лишь элементом абстрактного множества X (Y), не учитывающий особенностей языка и вообще говоря, не являющийся текстом в его привычном понимании.

Достаточно часто выражения для шифрования и расшифрования записываются в операторной форме:

$$y = T_k x,\ x = T_k^{-1} y,\ x \in X,\ y \in Y,\ k \in K, \qquad (1.7)$$

где T_k, T_k^{-1} - операторы шифрования и расшифрования, соответственно, на ключе $k \in K$.

Использование операторной формы записи позволяет определить операторы комбинирования криптосистем. Наиболее часто на практике используется два типа оператора комбинирования криптосистем: взвешенное суммирование криптосистем; произведение криптосистем [1].

Взвешенной суммой криптосистем называется такая криптосистема, которая образована «взвешенным» суммирование нескольких криптосистем:

$$S = p_1 T_1 + p_2 T_2 + ... + p_m T_m = \sum_{m=1}^{M} p_m T_m , \ \sum_{m=1}^{M} p_m = 1, \qquad (1.8)$$

где T_m - криптографическое преобразование m-й криптосистемы; p_m - вероятность выбора m-й криптосистемы.

Операция, выраженная приведенной выше формулой, состоит, во-первых, из предварительного выбора криптосистем T_m с вероятностями p_m. Этот выбор по сути является частью ключа криптосистемы S. После того как этот выбор сделан, криптосистемы T_m применяются в соответствии с их алгоритмами. Полный ключ криптосистемы S должен указывать, какая из криптосистем T_m выбрана и с каким ключом используется выбранная система.

На рис. 1.5 представлена обобщенная структурная схема комбинированной криптосистемы S, образованной взвешенным суммированием криптосистем T_m.

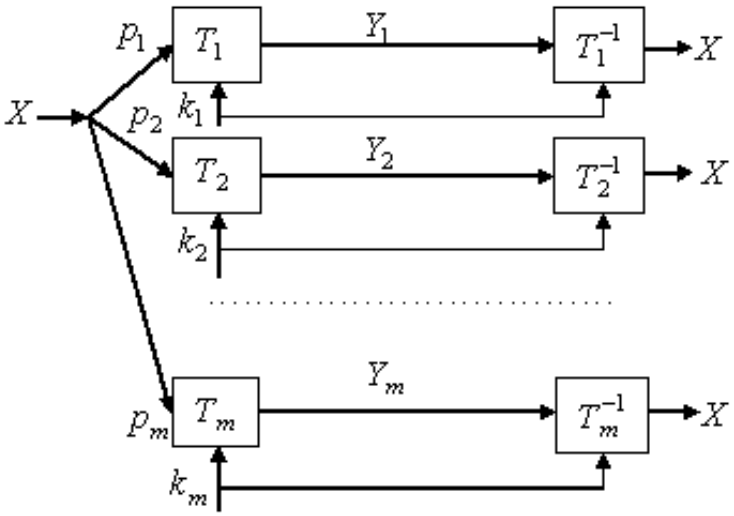

Рис. 1.5. Взвешенная сумма криптосистем

Второй способ комбинирования криптосистем заключается в образовании произведения криптосистем. **Произведением криптосистем** называется такая криптосистема V, для которой выполняется равенство:

$$V = T_1 T_2 ... T_m = \prod_{m=1}^{M} T_m . \qquad (1.9)$$

Строго говоря, данное выражение справедливо только в том случае, если область определения (пространство открытых текстов) криптосистемы T_m отождествляется с областью значений (пространством криптограмм) системы T_{m-1}. Тогда можно применить сначала криптосистему T_{m-1}, а затем криптосистему T_m к результату шифрования (см. рис. 1.6). Ключ криптосистемы V состоит как из ключей криптосистем T_m. Произведение шифров используется достаточно часто; например, после подстановки применяют перестановку или после перестановки – шифр Виженера; или же применяют шифр перестановки к тексту, а затем зашифровывают результат с помощью шифра замены, дробного шифра и т.д.

Необходимо заметить, что произведение криптосистем, вообще говоря, некоммутативно, т.е. не всегда $T_{m-1}T_m = T_mT_{m-1}$, хотя в частных случаях (таких, как подстановка и перестановка) коммутативность имеет место. Произведение криптосистем представляет собой ассоциативную операцию, т.к. оно по определению ассоциативно, т.е. $T_{m-2}(T_{m-1}T_m) = (T_{m-2}T_{m-1})T_m$.

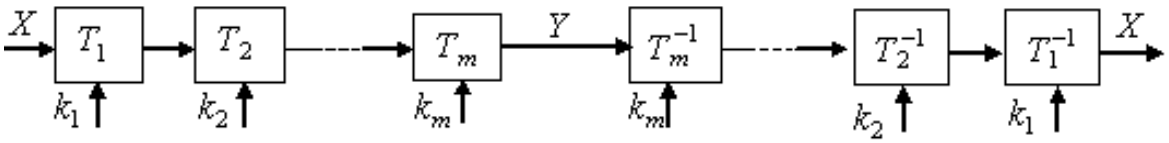

Рис. 1.6. Произведение криптосистем

Важным свойством криптосистем является транзитивность. **Транзитивными криптосистемами** называют криптосистемы, для которых выражение (1.5) разрешимо относительно $k \in K$ при любых парах $(x, y) \in X \times Y$. Транзитивные шифры для которых $|X| = |K|$ называют **минимальными**. Криптосистемы, у которых пространства X и Y можно отождествить (этот случай является очень частым) называются эндоморфными. **Эндоморфная криптосистема** может быть представлена как:

$$V = T^M. \qquad (1.10)$$

Криптосистема V, для которой справедливо соотношение:

$$V = TT = T. \qquad (1.11)$$

называется **идемпотентной**. Например, простая подстановка, шифр Виженера с периодом d являются идемпотентными. Две криптосистемы называются **эквивалентными**, если обе они произвольный открытый текст переводят в одну и ту же криптограмму.

Одно из важнейших предположений К.Шеннона при исследовании криптосистем состояло в том, что каждому возможному передаваемому сообщению (открытому тексту) соответствует априорная вероятность, определяемая вероятностным процессом получения сообщения. Аналогично,

имеется и априорная вероятность использования различных ключей. Эти вероятностные распределения на множестве открытых текстов, и множестве ключей характеризуют априорные знания противника относительно используемого шифра. При этом К.Шеннон предполагал, что сам шифр известен противнику.

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

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

Используя вероятностную модель шифра К.Шеннон впервые сформулировал понятие совершенно стойкого шифра, которое легло в основу теории стойкости криптосистем [5,13].

2. Симметричные криптосистемы и их свойства

2.1. Шифры замены

Пусть имеется открытое сообщение $X = x_0 x_1 ... x_{n-1}$ длины n в алфавите A_X и правило замены $f_з = (f_0, f_1, .., f_{n-1})$, тогда применение этого криптографического преобразования к открытому сообщению дает криптограмму $Y = y_0 y_1 ... y_{n-1} = f_0(x_0), f_1(x_1), ..., f_{n-1}(x_{n-1})$. Семейство криптографических преобразований $f_з = (f_0, f_1, .., f_{n-1})$ называется **шифром замены.**

В зависимости от вида криптографической функции $f_з(\cdot)$ шифры замены делятся на шифры моноалфавитной замены и шифры многоалфавитной замены. **Моноалфавитные замены** – наиболее простой вид преобразований, заключающийся в замене по определенному правилу букв исходного сообщения на другие буквы из этого же алфавита, т.е. каждая буква исходного текста преобразуется в букву криптограммы по одному и тому же закону. В случае **многоалфавитной замены** закон преобразования меняется от буквы к букве. Необходимо заметить, что один и тот же шифр может рассматриваться и как моно-, и как многоалфавитная замена в зависимости от определяемого алфавита. Например, замена биграмм с точки зрения обычного алфавита является моноалфавитной заменой, а с точки зрения алфавита биграмм - многоалфавитным. Рассмотрим наиболее известные шифры замены [1,3,4,8-11].

Шифр Цезаря. Процесс зашифрования исходного текста определяется выражением:

$$y_i = (x_i + k) \bmod m,\ i = \overline{1,n}, \qquad (2.1)$$

где y_i - буква криптограммы, x_i - буква открытого сообщения, k - ключ шифра, n - длина криптограммы (открытого текста), $m = |A_X|$ - мощность алфавита A_X. Очевидно, что выражение:

$$x_j = (y_j - k) \bmod m,\ j = \overline{1,n}, \qquad (2.2)$$

определяет процесс расшифрования криптограммы.

Обобщением шифра Цезаря является **аффинный шифр Цезаря**, определяемый выражением:

$$y_i = (ax_i + k) \bmod m,\ i = \overline{1,n}. \qquad (2.3)$$

Аффинный шифр Цезаря определяется двумя целыми числами a и k, где $0 \le a,k \le m-1,$. Числа a и n должны быть взаимно простыми. Взаимная простота a и n необходима, т.к. в противном случае возможны отображения различных символов в один и, как следствие, неоднозначность расшифрования. Процесс расшифрования определяется выражением:

$$x_j = (y_j - k)a^{-1} \bmod m,\ j = \overline{1,n}. \qquad (2.4)$$

Число a^{-1} является инверсией числа a по модулю m при этом должно выполняться равенство $aa^{-1} \bmod m = 1$.

Шифр Виженера. В шифре Виженера ключ k задается набором из d символов. Такие наборы подписываются под буквами открытого текста $X = x_1, x_2, ..., x_n,\ x_i \in A_X$, до получения периодической ключевой последовательности $\tilde{k} = k_1, k_2, ..., k_n,\ n = sd + r$, где s - число полных периодов k, $r = n \bmod d$, а значение d определяет период ключевой последовательности. Процесс шифрования определяется выражением:

$$y_i = (x_i + \tilde{k}_i) \bmod m,\ i = \overline{1,n}. \qquad (2.5)$$

При повторных операциях шифрования открытого теста шифром Виженера получаем **составной шифр Виженера**, который описывается выражением:

$$y_i = (x_i + \tilde{k}_i + \tilde{v}_i + ... + \tilde{w}_i) \bmod m,\ i = \overline{1,n}, \qquad (2.6)$$

где $\tilde{k}_i, \tilde{v}_i, ..., \tilde{w}_i$ - ключевые последовательности, имеющие, как правило, различные периоды. Период суммы этих ключевых последовательностей равен наименьшему общему кратному отдельных периодов. Иногда используется **усложненный шифр Виженера**. Усложнение заключается в «перемешивании» исходного алфавита и получении нового алфавита A'_X, причем $|A_X| = |A'_X| = m$. Перемешивание обычно проводится при помощи **лозунга**, который представляет собой слово или фразу, неповторяющиеся символы которого образуют начало алфавита. Заметим, что шифр Виженера с периодом $d = 1$

представляет собой шифр Цезаря. Если же криптосистема Виженера имеет период $d = n$, то получаем **шифр гаммирования**:

$$y_i = (x_i + \gamma_i) \bmod m,\ i = \overline{1,n}. \qquad (2.7)$$

В шифре гаммирования ключевая последовательность носит название гамма-последовательности γ. Частным случаем шифра гаммирования является **шифр Вернама**, который определен на алфавите $A = \{0,1\}$.

Разновидностью шифра Виженера является **шифр Бофора**, который определяется выражением:

$$y_i = (x_i - \tilde{k}_i) \bmod m,\ i = \overline{1,n}, \qquad (2.8)$$

т.е. шифр Бофора представляет собой «расшифрование» шифра Виженера. Шифр Бофора с периодом $d = 1$ представляет собой обратный шифр Цезаря. Шифр, в котором само сообщение (или его часть) или результирующая криптограмма используется в качестве ключа, называется **шифром с автоключом**. Шифрование в этом случае начинается с помощью «первичного ключа» и продолжается с помощью сообщения или криптограммы

Примером отечественного шифра замены является **шифр простой литореи** или «тарабарская грамота». Суть криптографического преобразования заключается в следующем. В таблицу из двух строк в определенном порядке записываются согласные буквы алфавита. Замена осуществляется по столбцам «сверху-вниз» или «снизу-вверх». Гласные буквы записываются без зашифрования или как говорят без «затаивания».

Шифр Плейфера. Шифр является примером шифра замены биграмм. Ключом шифра является таблица Θ, с вписанным в нее алфавитом, размера $n_1 \times n_2$, причем $n_1 n_2 = m$. В соответствии с алгоритмом шифра Плейфера открытое сообщение разбивается на биграммы и одновременно видоизменяется так, чтобы не встречались биграммы вида $(x_i = a, x_{i+1} = a)$. Криптографическое преобразование определяется следующим образом:

$$f_з(\theta_{i,j}, \theta_{s,t}) = \begin{cases} (\theta_{i,t}, \theta_{s,j}), i \neq s,\ j \neq t; \\ (\theta_{i,j+1}, \theta_{i,t+1}), i = s,\ j \neq t; \\ (\theta_{i+1,t}, \theta_{s+1,t}), i \neq s,\ j = t, \end{cases} \qquad (2.9)$$

где $\theta_{i,j}$ - символ таблицы шифра Θ.

2.2. Шифры перестановки

Пусть имеется открытое сообщение $X = x_0 x_1 ... x_{n-1}$ длины n в алфавите A_X и правило перестановки $f_n = (f(0), f(1), .., f(n-1))$, тогда применение этого криптографического преобразования к открытому сообщению дает криптограмму $Y = y_0 y_1 ... y_{n-1} = x_{f(0)}, x_{f(1)}, ..., x_{f(n-1)}$. Семейство

криптографических преобразований $f_n = (f(0), f(1),.., f(n-1))$ называется **шифром перестановки**. Таким образом, перестановка заключается в переупорядочении букв открытого текста, в результате чего он становиться нечитаемым. Ключом шифра является правило перестановки. На практике при применении шифров перестановки длины открытого текста и ключа не совпадают, так как, как правило, ключ имеет фиксированную длину l. В этом случае открытый текст разбивается на $n_l = \frac{N}{l}$ отрезков длины l, к каждому из которых применяется перестановка. В случае, когда открытое сообщение не может быть разделено на n_l равных отрезков, т.е. $N = l \cdot n_l + r$, где r - остаток, необходимо дополнить открытый текст $l - r$ произвольными символами, называемыми «пустышками». Рассмотрим наиболее известные шифры перестановки [1,3,4,8-11].

Шифр простой перестановки заключается в следующем. В соответствии с заданным правилом осуществляется перестановка букв открытого текста. Правило перестановки является ключом шифра. Как правило, длина ключа соответствует длине открытого сообщения.

Шифр перестановки с фиксированным периодом относиться к простым шифрам перестановки. Процесс шифрования заключается в следующем. Сообщение делится на блоки соответствующие заданному периоду. К каждому блоку применяется одна и та же перестановка.

Шифры маршрутной перестановки используют прямоугольную таблицу, в которую текст записывается, например, по строкам, а криптограмма считывается по определенному маршруту (по столбцам, по диагонали и т.п.). Расшифрование состоит в обратной действии, сначала по заданному маршруту заполняется таблица, а затем, например, по строкам, считывается исходный текст. Ключом таких шифров являются размеры таблицы и маршрут записи и считывания символов. Наиболее сложные маршрутные перестановки реализуются с применением гамильтоновых путей на графе (см. рис. 2.1).

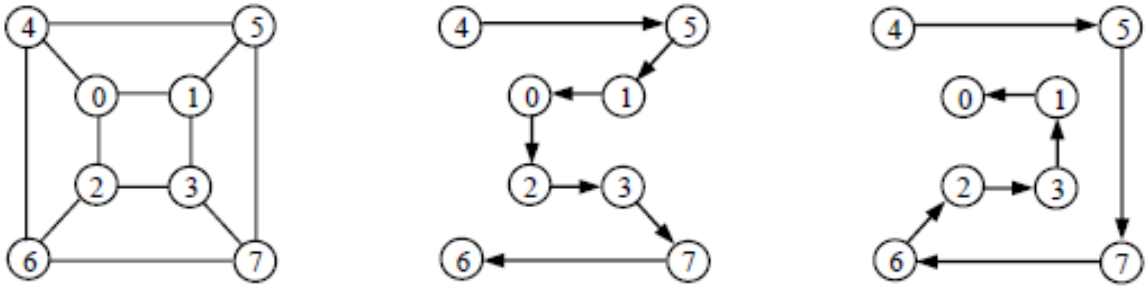

Рис. 2.1. Гамильтоновы пути на графе

Шифр вертикальной перестановки является частным случаем шифра маршрутной перестановки. Шифрование заключается в записи по строкам открытого текста в таблицу, а считывание криптограммы осуществляется по столбцам.

Шифр «магический квадрат» является частным случаем шифра маршрутной перестановки и использует в качестве таблицы «магический квадрат». «Магическим квадратом» является таблица размером $m \times m$, в которую вписываются числа от 1 до m^2 так, чтобы сумма чисел по строкам, столбцам и полным диагоналям равнялась одному и тому же числу. Процесс шифрования сводился к тому, что символы открытого текста в соответствии с их порядковым номером вписывались с соответствующую этому номеру ячейку квадрата, а считывание криптограммы осуществлялось по строкам. Если в квадрате оставались пустые ячейки, то их заполняли произвольными символами. Ключом данного шифра является «магический квадрат».

Наиболее стойким из шифров перестановки с помощью таблиц является шифрование с помощью **шифра Кордано**, который часто называют «решеткой Кордано». Решетка используется при шифровании как ключ и представляет собой квадратную таблицу размера $2n \times 2n$ (n - натуральное число), в которой n^2 всех позиций выделены для записи символов открытого текста. Эти n^2 позиций случайным образом более или менее равномерно распределены по площади квадрата и выбраны таким образом, что при последовательном повороте квадрата вокруг геометрического центра на 90, 180, 270 и 360 градусов вырезанные клетки последовательно покрывают все $4n^2$ клеток квадрата. При шифровании решетка Кордано накладывается на лист бумаги того же размера. В начальном угловом положении, соответствующем углу поворота 0 градусов, в клетки выделенные «окнами» решетки Кордано последовательно (по строкам слева направо, или в другом оговоренном заранее порядке) записываются первые n^2 символов открытого сообщения. Затем решетка Кордано поворачивается в угловое положение 90 градусов и вновь в клетки выделенные «окнами» последовательно записываются следующие n^2 символов открытого сообщения. Данная операция повторяется до полного поворота решетки вокруг своего геометрического центра. Если длина сообщения $N > 4n^2$, то решетка Кордано используется многократно для зашифрования всего сообщения. Криптограмма считывается с листа бумаги в оговоренном заранее порядке. Главное требование к решетке Кордано – при всех поворотах «окна» не должны попадать на одно и тоже место в квадрате, в котором образуется криптограмма. В качестве решетки может быть использован не только квадрат, но и прямоугольник. В этом случае поворот осуществляются на 180 градусов, затем решетка переворачивается оборотной стороной и после записи символов снова поворачивается на 180 градусов.

2.3. Поточные криптосистемы

Поточные криптосистемы относятся к шифрам замены, преобразующим посимвольно открытый текст в криптограмму [2,10,11]. Традиционно шифры замены строились по принципу поточного шифрования. В современных поточных криптосистемах в качестве шифруемых символов фигурируют биты или даже байты. Поточные криптосистемы разделяются на **синхронные** (СПК) и **асинхронные** или **самосинхронизирующиеся** (ССПК). Упрощенная структурная схема СПК представлена на рис. 2.2.

Схема СПК состоит из управляющего и шифрующего блоков. Управляющий блок генерирует управляющую последовательность $\gamma = \overline{\gamma_i}$, $i = \overline{1,n}$, которая используется для формирования шифрующих криптопреобразований $f_\gamma(\cdot)$. Управляющую последовательность часто называют управляющей гаммой, а управляющий блок – генератором гаммы.

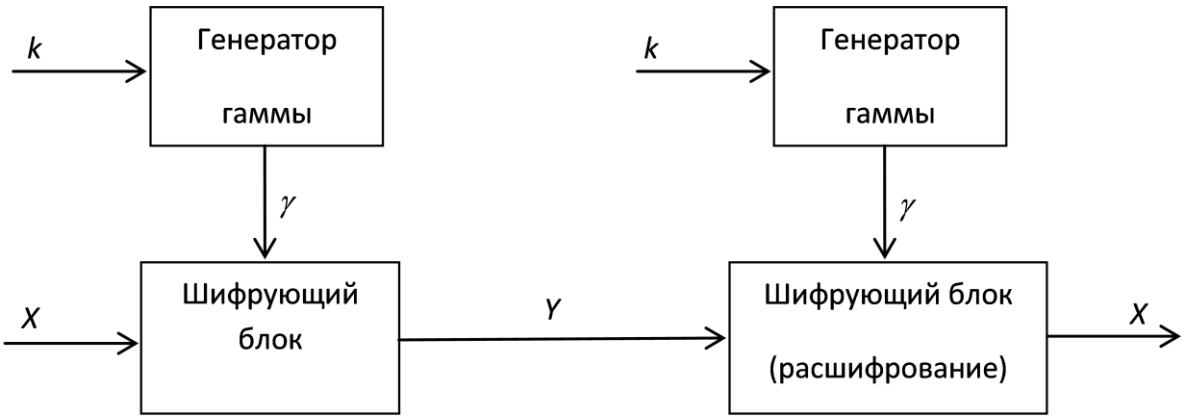

Рис. 2.2. Упрощенная структурная схема СПК

Шифрующий блок зашифровывает символ открытого текста x_i в символ криптограммы y_i с использованием криптографического преобразования $f_\gamma(\cdot)$. Отправитель сообщения устанавливает заранее оговоренный ключ k генератора и, вычислив криптограмму Y, отправляет ее получателю. Для расшифрования получатель использует идентичный генератор гаммы, в который устанавливается тот же ключ k. Шифрующий блок получателя в режиме расшифрования вычисляет открытый текст X по криптограмме Y используя обратное криптографическое преобразование $f_\gamma^{-1}(\cdot)$. В СПК генерируемая гамма не зависит от открытого текста, т.к. генератор гаммы автономен. В связи с этим СПК функционирует исправно до тех пор, пока устройства, реализующие шифрование и расшифрование на концах линии связи, работают синхронно, то есть не имеет места расшифрование символа криптограммы y_i с использованием символа гаммы γ_j, $i \neq j$. Такие

нежелательные сбои, называемые рассинхронизацией, могут наступить из-за различных скоростей работы аппаратуры на приемной и передающем концах, удалении символов при передаче в канале связи и т.д. Сбои могут повлечь неправильное расшифрование всего последующего отрезка сообщения. Если такое случается отправитель и получатель должны восстановить синхронизм работы генераторов гаммы прежде, чем продолжить сеанс связи. Обычно проблемы восстановления синхронизма решаются либо с помощью повторного шифрования с реинициализацией ключа обоими абонентами (повторное использование гаммы крайне нежелательно, а в некоторых криптосистемах недопустимо), либо с помощью разбиения текста на блоки, начала и окончания которых снабжены специальными маркерами. Во втором случае рассинхронизация приводит к некорректному расшифрованию лишь до тех пор, пока получателем не будет принят один из маркеров.

К достоинствам СПК следует отнести то, что они не размножают искажений символов текста, которые довольно часто имеют место при передаче по каналам связи. Если при отправлении сообщения был искажен символ x_i или передаче по каналу связи был искажен символ y_i, то при синхронной работе генераторов эти искажения не повлияют на правильность расшифрования всех символов, кроме $i-$го. Также СПК защищают передаваемое сообщение от несанкционированных вставок и удаления отрезков текста, так как в этих случаях произойдет рассинхронизация и «вмешательство» злоумышленника будет немедленно обнаружено. В то же время СПК не вполне защищают от умышленной подмены отрезка сообщения на другой отрезок такой же длины. Если злоумышленнику известен отрезок открытого текста, то ему не составляет труда подменить его таким отрезком, который расшифруется в требуемый фрагмент текста.

Схема ССПК также состоит из управляющего и шифрующего блоков с аналогичным функциональным назначением. Однако имеются отличия в построении управляющего блока и в схеме взаимодействия блоков. Как видно из рис. 2.3, ССПК имеет обратную связь по криптограмме, что является важным отличием ССПК. Генерируемая гамма зависит от предшествующих битов криптограммы:

$$\gamma_{i+1} = f(y_{i-n+1}, y_{i-n+2}, \ldots, y_i, k), \; i \geq n. \qquad (2.10)$$

Каждое внутренне состояние управляющего блока СППК (за исключением первых n состояний) заполняется n предыдущими знаками криптограммы. Поэтому если n следующих подряд знаков криптограммы не подвергаются искажению при передаче по линии связи, то ССПК на приемном и передающем концах устанавливаются в одинаковые внутренние состояния и, следовательно, вырабатывают при этом одинаковые символы гаммы. Т.е. происходит самосинхронизация ССПК. Как правило, каждое шифруемое сообщение начинается не с содержательного текста, а со случайного отрезка из n символов, который шифруется, передается и затем расшифровывается. И

хотя расшифрование этого отрезка реализуется некорректно в силу несовпадения начальных состояний генераторов, после передачи n начальных знаков генераторы синхронизируются. Для затруднения криптоанализа по первым n символам криптограммы начальное состояние ССПК выбирается случайным образом для каждого сообщения.

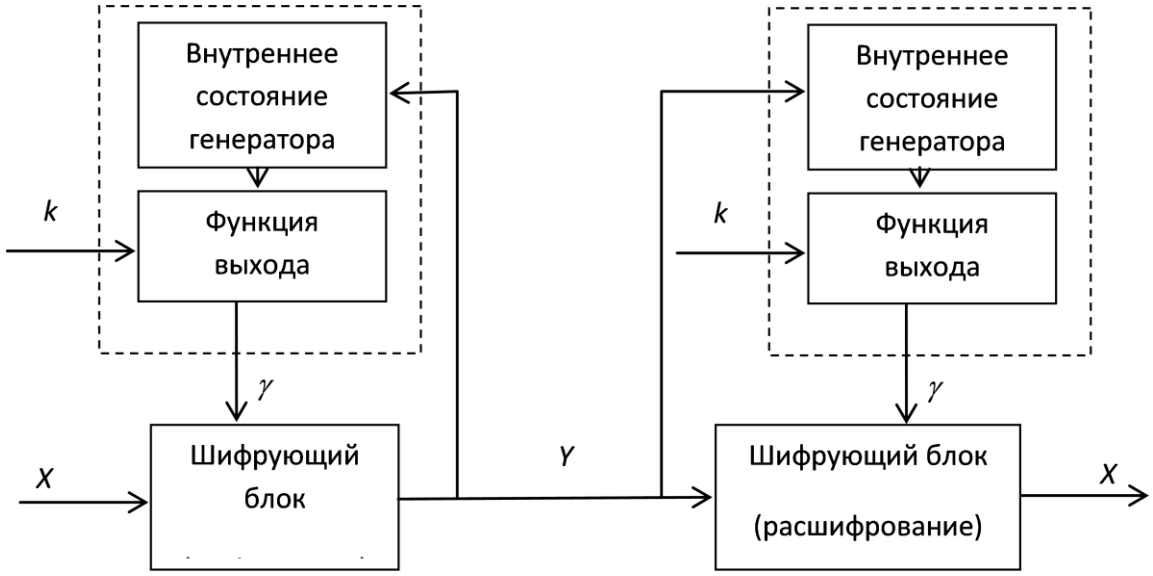

Рис. 2.3 Упрощенная структурная схема ССПК

Основным недостатком ССПК является размножение ошибок. Единичная ошибка в криптограмме порождает n ошибок в открытом тексте. Также ССПК уязвимы к имитации сообщения. Злоумышленник может записать некоторый перехваченный отрезок криптограммы и позже отправить его в адрес. После нескольких нестыковок в начале сообщения (до n символов) посланный отрезок расшифруется верно и получатель не сможет определить, что принято устаревшее сообщение. Подобная имитация невозможна, если использовать метки времени, а также менять ключи при каждом новом сообщении.

Поточные криптосистемы, как правило, являются строятся на основе шифров гаммирования. При этом различаю табличное и модульное гаммирование. **Табличное гаммирование** заключается в том, что криптографическое преобразование $f(x,\gamma)$ представляется в виде латинского квадрата в алфавите A_X. При этом, как и в случае шифра Полибия, символ криптограммы определяется на пересечении строки и столбца, которые задаются символами открытого текста и гаммы. Таким образом, шифр Полибия является примером шифра табличного гаммирования. Выражение (2.7) определяет шифр **модульного гаммирования**. Наиболее удобными с точки зрения практического применения являются шифры модульного гаммирования, которым соответствует уравнение:

$$y_i = x_i \oplus \gamma_i,\ i = \overline{1,n}, \qquad (2.11)$$

где символ $\oplus$ определяет операцию сложения по модулю 2 (операция XOR). Удобство шифров модульного гаммирования заключается в их обратимости:

$$x_i = y_i \oplus \gamma_i = (x_i \oplus \gamma_i) \oplus \gamma_i = x_i,\ i = \overline{1,n}. \qquad (2.12)$$

Для обеспечения высокой криптографической стойкости при применении шифров гаммирования не допускается [2,11]:

- повторное использование гаммы;
- использование неравновероятной гаммы.

Рассмотрим поточные криптосистемы, которые в настоящее время находят достаточно широкое применение [2,10,11].

Поточная криптосистема RC4 (Rivest Cipher 4) был разработана Роном Ривестом в 1987 году. Эта криптосистема позволяет использовать ключи размером от 8 до 2048 бит. В RC4 для зашифрования и расшифрования применяются одни и те же действия: генерируется гамма, которая накладывается на шифруемое сообщение путем сложения по модулю 2. Криптосистема RC4 является собственностью компании RSA Data Security Inc, а ее описание никогда не было опубликовано и предоставлялось партнерам только после подписания соглашения о неразглашении. Однако в сентябре 1994 года алгоритм RC4 был анонимно опубликован. С тех пор сама криптосистема перестала быть секретом, но название RC4 остается торговой маркой. То есть, чтобы получить право заявлять, что в коммерческом программном продукте используется RC4, необходимо приобрести лицензию на этот алгоритм у RSA Data Security. Без лицензии можно утверждать лишь то, что используется алгоритм, похожий на RC4 и совпадающий с ним на всем известном множестве тестов. В связи с этим, некоторые компании не имеющие лицензии на RC4 предпочитают называть ее ARC4 (Alleged RC4). Основные преимущества криптосистемы RC4 - высокая скорость работы и переменный размер ключа. Главным фактором, способствовавшим широкому применению RC4, была простота ее аппаратной и программной реализации. Вместе с тем, RC4 довольно уязвима, если используются не случайные или связанные ключи или один ключевой поток используется дважды. Криптосистема RC4 применяется в таких продуктах, как Microsoft Office, Lotus Notes, Adobe Acrobat и др.

Поточная криптосистема VMPC (Variably Modified Permutation Composition), применяется в системах защиты информации в компьютерных сетях. Криптосистема разработана криптографом Бартошем Зольтаком в качестве улучшенного варианта криптосистемы RC4. Алгоритм VMPC строится как и любой потоковый шифр на основе параметризованного ключом генератора псевдослучайных чисел, базой которого является односторонняя необратимая функция VMPC. Основные преимущества шифра, как и RC4, - высокая скорость работы, переменный размер ключа и вектора инициализации (от 128 до 512 бит включительно), а также простота реализации.

Поточная криптосистема Rabbit – высокоскоростная криптосистема, впервые представленная в феврале 2003 года. В мае 2005, криптосистема Rabbit была отправлена на конкурс eStream, целью которого было создание европейских стандартов для поточных систем шифрования. Криптосистема Rabbit использует 128-битный ключ и 64-битный инициализирующий вектор. Криптосистема была разработан с целью использования в программном обеспечении, как обладающая высокой скоростью шифрования. Тем не менее, криптосистема также оказалась быстрой и компактной при аппаратной реализации. Основным компонентом криптосистемы является генератор битового потока, который шифрует 128 битов сообщения за итерацию. Достоинство криптосистемы в тщательном перемешивании ее внутренних состояний между двумя последовательными итерациями. Функция перемешивания полностью основана на арифметических операциях, доступных на современных процессорах. Авторы шифра предоставили полный набор технических описаний на домашней странице компании Cryptico, которая обладала патентом на криптосистему, и многие годы для ее использования в коммерческих целях требовалась лицензия. Однако, в 2008 криптосистему Rabbit разрешили использовать в любых целях бесплатно.

Поточная криптосистема Trivium ориентирована, в первую очередь, на аппаратную реализацию с гибким равновесием между скоростью работы и количеством элементов, имеющая также возможность достаточно эффективной программной реализации. Криптосистема была представлена в декабре 2008 года. Авторами криптосистемы являются Кристоф Де Канниэр и Барт Пренил. Криптосистема Trivium генерирует вплоть до 2^{64} бит выходного потока из 80 бит ключа и 80 бит вектора инициализации IV. Это самая простая криптосистема, которая, однако, показывает отличные результаты по криптостойкости. В криптосистеме Trivium, также как и во всех поточных криптосистемах, для зашифрования и расшифрования применяются одни и те же действия: генерируется гамма, которая накладывается на шифруемое сообщение путем сложения по модулю 2.

Поточная криптосистема MICKEY (Mutual Irregular Clocking KEYstream generator) существует в двух вариантах - с длиной ключа 80 бит (MICKEY) и 128 бит (MICKEY-128). Криптосистема был разработана Стивом Бэббиджем и Мэтью Доддом в 2005 году с целью использования в системах с ограниченными ресурсами. Эта криптосистема имеет простую аппаратную реализацию при высокой степени защищенности, в ней используется нерегулярное тактирование сдвиговых регистров, а также новые методы, обеспечивающие достаточно большой период и псевдослучайность ключевой последовательности, а также устойчивость к криптоатакам. Криптосистема имеет входные параметры: ключ длиной 80 бит, вектор инициализации длиной от 0 до 80 бит.

Несколько слов о европейском проекте eSTREAM. Проект eSTREAM имел целью получить поточную криптосистему, которая может быть использована в качестве стандарта поточного шифрования. В ноябре 2004 г. организаторами проекта было объявлено о приеме предложений по алгоритмам поточных криптосистем. Сбор предложений закончился в апреле 2005 г. На конкурс eSTREAM было представлено 34 алгоритма поточного шифрования. В феврале 2006 г. завершился первый этап оценки шифров, в июле начался второй этап, который продлился до сентября 2007 г. В апреле 2008 г. проект был завершен.

Впервые в истории в международном криптографическом конкурсе достигли успехов и российские криптосистемы. Это очень быстрая криптосистема ABC и криптосистема YAMB. Хотя данные криптосистемы в ходе рассмотрения были «взломаны», но вместе с тем оказались одними из самых быстрых. Результаты проекта eSTREAM представлены в табл. 2.1. Поточные криптосистемы объединены в группы: 1 - наиболее перспективные; 2 - другие; 3 - не рассматриваемые далее.

Таблица 2.1. Результаты проекта eSTREAM

Поточные криптосистемы для программной реализации		
1	**2**	**3**
Dragon-128, HC-256	ABC, CryptMT	F-FCSR, Fubuki, MAG
LEX, Phelix	NLS	Frogbit, Hermes,
Salsa20, Sosemanuk	Rabbit, Dicing, Polar Bear	Mir-1
Поточные криптосистемы для аппаратной реализации		
Grain	Achterbahn, Decim, LEX, TCS-3, NLS, VEST	MAG
Mickey-128	Mickey, Mosquito, Edon80, F-FCSR, Hermes	SSS
Phelix	Salsa20, Sfinks, Yamb, Zk-Crypt, WG	TRBDK YAEA
Trivium	Polar Bear, Pomaranch, Rabbit	

Интересным промежуточным результатом проекта eSTREAM можно считать предложенные новые статистические тесты криптографических генераторов. В отличие от предыдущих подходов (криптогенератор - черный ящик), в этих тестах внимание уделяется анализу зависимостей между гаммой и ключом, между гаммой и вектором инициализации IV, между отрезками гаммы, сгенерированными при разных векторах инициализации IV, а также влиянию на свойства гаммы внутреннего состояния генератора. Эти наборы тестов позволяют выявить неочевидные свойства генераторов гаммы.

2.4. Блочные криптосистемы

Блочные симметричные криптосистемы (БСК) представляют собой семейство обратимых криптографических преобразований блоков (частей фиксированной дины) исходного текста [2,4,10,11,14]. Фактически БСК – система подстановки на алфавите блоков (она может быть моно- или многоалфавитной в зависимости от режима блочного шифра).

2.4.1. Принципы построения блочных криптосистем

Первым опытом создания блочной криптосистемы явилась разработанная американской фирмой IBM криптосистема LUCIFER. Блоки открытого и шифрованного текста, обрабатываемые криптосистемой LUCIFER, представляют собой двоичные векторы длиной 128 бит. Криптосистема построена по принципу «сэндвича», составленного из нескольких слоев – преобразований замены S (substitution) блоков и преобразований перестановки P (permutation) элементов блоков. Такие схемы получили название **SP-сетей**, т.е. сетей перестановок и замен. Криптографическая идея SP-сетей заключается в построении сложного криптопреобразования с помощью композиции нескольких относительно простых, удобно реализуемых преобразований (см. рис. 2.4).

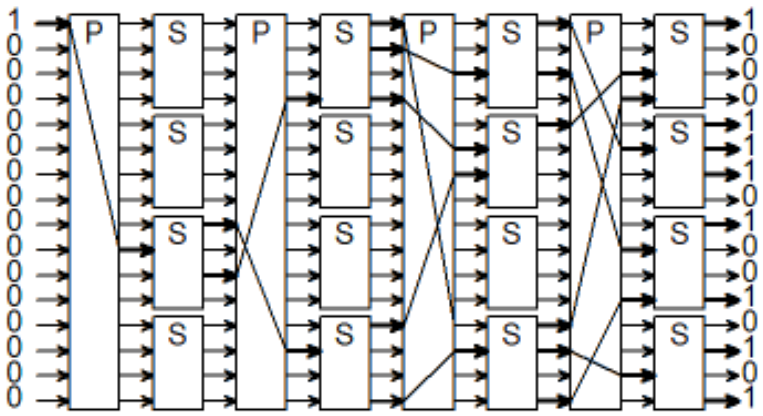

Рис. 2.4. Пример реализации SP-сети

Однако полученная криптосистема LUCIFER получилась достаточно громоздкой и обладала низкой производительностью. Скорость шифрования при программной реализации криптосистемы не превышала 8 кбайт/с, аппаратная реализация давала скорость шифрования не более 97 кбайт/с. К тому же у разработчиков были опасения по поводу криптостойкости, которые впоследствии подтвердились. Вместе с тем, накопленный разработчиками опыт при создании криптосистемы LUCIFER пригодились при разработки последующих блочных криптосистем.

В 1974 году фирмой IBM был разработана криптосистема, получившая название DES (Data Encryption Standart) [2,8,10,11]. Подобно криптосистеме LUCIFER криптосистема DES частично реализует принцип SP-сети и построена по **итеративному** принципу, то есть на основе нескольких однотипных преобразований. В дальнейшем итеративный принцип использовался в подавляющем большинстве разработок блочных криптосистем. Рассмотрим криптографическое преобразование $F(\cdot)$ итеративной блочной криптосистемы. Как правило, блок открытого текста x подвергается предварительному шифрованию (как правило, перестановке) $f_0(x,k_0)$, где k_0 - **ключ входного криптопреобразования**. Затем полученная криптограмма многократно подвергается шифрованию с помощью однотипного криптопреобразования $\varphi_i(y_i',k_i)$, $i=\overline{1,r}$, где k_i - **цикловой (раундовый) ключ**, y_i' - входной блок $i-$го цикла шифрования. Криптопреобразование $\varphi_i(\cdot)$ называется **цикловой функцией**, а переменная r определяет количество **циклов** (**раундов**) шифрования. После реализации всех r раундов шифрования осуществляется еще одно финальное преобразование (как правило, перестановка) $f_{r+1}(y_r',k_{r+1})$, где y_r' - выходной блок последнего раунда шифрования, k_{r+1} - **ключ выходного криптопреобразования.** Таким образом, криптографическое преобразование итеративной блочной криптосистемы имеет вид:

$$y=F(x,k)=f_{r+1}(y_r',k_{r+1})\varphi_r(y_{r-1}',k_r)\cdots\varphi_1(y_0',k_1)f_0(x,k_0). \quad (2.13)$$

Криптопреобразование $f_0(x,k_0)$ называется **входным преобразованием**, а $f_{r+1}(y_r',k_{r+1})$ - **выходным преобразованием**. Обратное криптографическое преобразование определяется равенством:

$$x=F^{-1}(y,k)=f_0^{-1}(y_0',k_0)\varphi_2^{-1}(y_1',k_1)\cdots\varphi_r^{-1}(y_r',k_r)f_{r+1}^{-1}(y,k_{r+1}). \quad (2.14)$$

Многократное использование цикловой функции должно обеспечить следующие свойства криптопреобразования:

- рассеивание (позволяет скрыть статистические зависимости между символами открытого текста и обеспечивает невозможность определения ключа по частям);

- перемешивание (позволяет усложнить зависимость между ключом и криптограммой).

Один из первых способов построения цикловой функции основан на использовании отображения типа регистра сдвига. Конструкция была признана удачной и нашла широкое применение в дальнейших разработках блочных криптосистем (FEAL, Khufu, Khafre, LOKI, Blowfish, ГОСТ 28147-89). Эта конструкция названа в честь разработчика **схемой Фейстеля** [2,12]. Схема Фейстеля представляет собой блочный симметричный шифр, криптографическая функция которого оперирует «половинами» входных блоков и имеет вид:

$$f(x_1, x_2, k) = x_2 \| \psi(x_2, k) \oplus x_1, \qquad (2.15)$$

где x_1 и x_2 - половины входного блока; $\psi(x_2, k)$ - функция усложнения; $\|$ - операция конкатенации. На рис. 2.5 представлена структура схемы Фейстеля. Варианты схемы Фейстеля отличаются конструкцией функции усложнения.

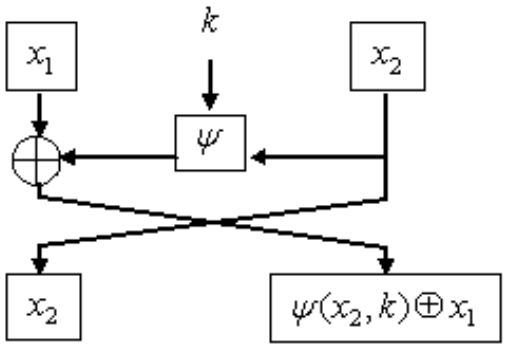

Рис. 2.5. Схема Фейстеля

Для получения криптопреобразования (2.15), обладающего хорошими криптографическими свойствами, функция усложнения реализуется в виде композиции элементарных преобразований, называемых слоями функции усложнения (или цикловой функции). Конструктивные слои функции усложнения имеют следующие назначения: подмешивание раундовых ключей; перемешивание входных блоков; реализацию сложной нелинейной зависимости между знаками ключа, входного и выходного блоков.

Цикловая функция должна удовлетворять ряду условий:

- цикловая функция должна быть обратимой (функция усложнения схемы Фейстеля в принципе может не удовлетворять этому требованию, так как обратимость преобразования обеспечивается за счет использования операции XOR);

- цикловая функция должна быть нелинейной;

- перемешивающие слои цикловой функции должны реализовывать связи между входными и выходными битами S-блоков (блоков замены) таким образом, чтобы каждый S-блок удовлетворял критериям лавинного эффекта, а также совокупность входных битов каждого S-блока зависела от выходов нескольких S-блоков предыдущего цикла;

- цикловая функция должна обладать свойствами, затрудняющими применение методов дифференциального и линейного криптоанализа, т.е. цикловая функция должна иметь минимальную корреляцию между разностью открытых текстов и соответствующих криптограмм.

Для затруднения применения методов криптоанализа блочные криптосистемы должны использовать в качестве входного и выходного преобразований операции XOR. Эти операции получили название отбеливания, а использующий эти операции шифр называют шифром с отбеливанием. Операция отбеливания улучшает криптографические свойства шифра, не нарушая при этом обратимости криптопреобразования.

2.4.2. Режимы шифрования

Для шифрования исходного открытого текста БСК могут использоваться в различных режимах [2,11]. Далее будут рассмотрены четыре режима шифрования наиболее часто встречающиеся на практике:

- режим электронной кодировочной книги - ECB (Electronic Code Book);
- режим сцепления блоков криптограммы - CBC (Cipher Block Chaining);
- режим обратной связи по криптограмме - CFB (Cipher Feed Back);
- режим обратной связи по выходу - OFB (Output Feed Back).

Режим электронной кодировочной книги ECB. Исходный текст разбивается на блоки, равные размеру блока шифра. Затем каждый блок шифруется независимо от других с использованием одного ключа шифрования (см. рис. 2.6). Непосредственно этот режим применяется для шифрования небольших объемов информации, размером не более одного блока или для шифрования ключей. Это связано с тем, что одинаковые блоки открытого текста преобразуются в одинаковые блоки криптограмма, что может дать криптоаналитику определенную информацию о содержании сообщения.

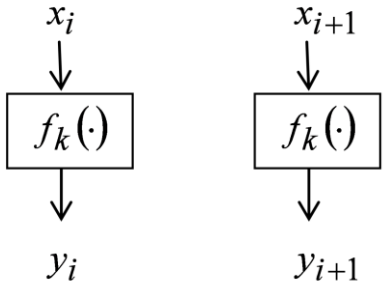

Рис. 2.6. Режим электронной кодировочной книги ECB

Стойкость режима ECB равна стойкости самого шифра, однако, структура исходного текста при этом не скрывается. Скорость шифрования равна скорости блочного шифра. Основным достоинством этого режима является простота реализации.

Режим сцепления блоков криптограммы CBC. В данном режиме каждый блок исходного текста складывается по модулю 2 с предыдущим

блоком криптограммы, а затем шифруются (см. рис. 2.7). Для начала процесса шифрования используется **синхропосылка** (или **начальный вектор**) y_0. Процессы шифрования и расшифрования описывается выражениями:

$$y_i = f_k(x_i \oplus y_{i-1}),\ i = \overline{1,n}. \quad (2.16)$$

$$x_i = f_k^{-1}(y_i) \oplus y_{i-1},\ i = \overline{1,n}. \quad (2.17)$$

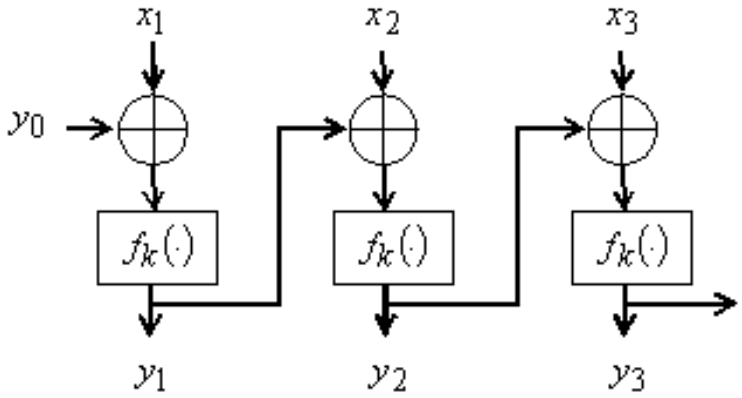

Рис. 2.7. Режим сцепления блоков криптограммы CBC

Стойкость режима CBC равна стойкости блочного шифра, лежащего в его основе. Структура исходного текста скрывается за счет сложения по модулю 2 предыдущего блока криптограммы с очередным блоком открытого текста. Стойкость шифрованного текста увеличивается, поскольку становиться невозможным прямая манипуляция исходным текстом. Скорость шифрования равна скорости работы блочного шифра, однако простого способа распараллеливания процесса шифрования, как для режима ECB, не существует. Начальный вектор y_0 может передаваться по линии связи как в открытом, так и в шифрованном виде. Однако следует избегать повторения начального вектора, это позволит затруднить криптоатаку. Искажение одного бита в блоке открытого текста x_i влечет за собой искажение в среднем половины битов во всех блоках криптограммы, начиная с y_i. Для расшифрования это несущественно, так как восстановленный текст будет содержать ту же единственную ошибку. Искажение бита в блоке y_i влечет за собой искажение около половины битов в блоке x_i, начиная с этого бита, и в блоке x_{i+1} Следующие блоки расшифровываются корректно.

Режим обратной связи по криптограмме CFB. В данном режиме предыдущий блок криптограммы шифруется еще раз, и для получения очередного блока криптограммы результат складывается по модулю 2 с блоком исходного текста (см. рис. 2.8). Для начала процесса шифрования также используется начальный вектор y_0. Процессы шифрования и расшифрования описывается выражениями:

$$y_i = x_i \oplus f_k(y_{i-1}),\ i = \overline{1,n}. \quad (2.18)$$

$$x_i = f_k(y_{i-1}) \oplus x_i,\ i = \overline{1,n}. \qquad (2.19)$$

Особенностью режима является то, что в (2.18) и (2.19) базовый алгоритм используется только для шифрования. Искажение одного бита в блоке x_i влечет за собой искажение одного бита в y_i и в среднем половины битов во всех блоках криптограммы, начиная с y_{i+1}, но при расшифровании получается открытый текст с той же единственной ошибкой.

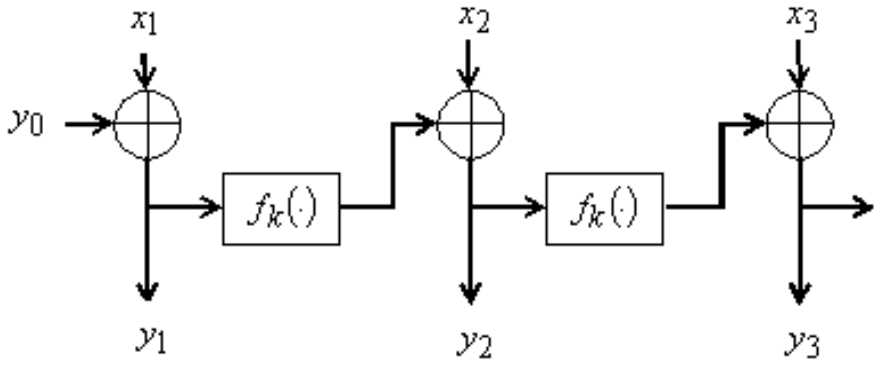

Рис. 2.8. Режим обратной связи по криптограмме CFB

Искажение бита в блоке y_i влечет искажение соответствующего бита в блоке x_i. Затем ошибка искажает в среднем половину битов в каждом из последующих блоков, но в дальнейшем блоки расшифровываются корректно. Данный режим, как и ССПК, самостоятельно восстанавливается после ошибок синхронизации. Стойкость режима равна стойкости блочного шифра, лежащего в его основе, и структура исходного текста скрывается за счет использования операции сложения по модулю 2. Скорость шифрования равна скорости работы блочного шифра, и простого способа распараллеливания процесса шифрования не существует.

Режим обратной связи по выходу OFB. Данный режим подобен режиму CFB, за исключением того, что величины, складываемые по модулю 2 с блоками исходного текста, генерируются независимо от исходного текста и криптограммы (см. рис. 2.9).

Процессы шифрования и расшифрования описывается выражениями:

$$y_i = x_i \oplus s_i,\ s_i = f_k(s_{i-1}),\ i = \overline{1,n}. \qquad (2.20)$$

$$x_i = y_i \oplus s_i,\ i = \overline{1,n}. \qquad (2.21)$$

Для начала процесса шифрования используется начальный вектор s_0.

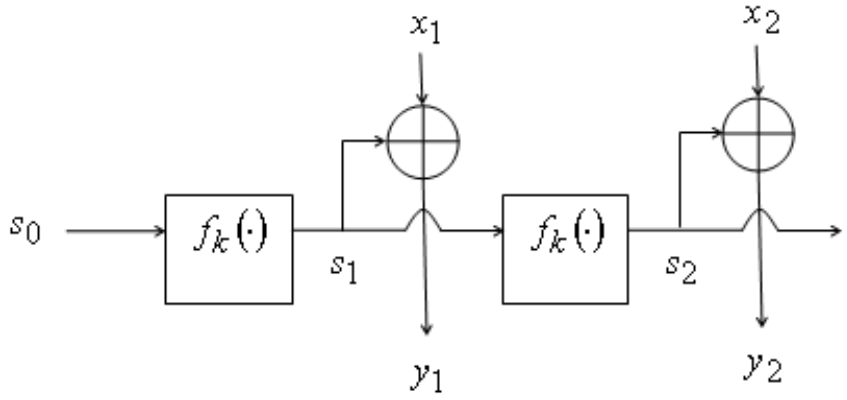

Рис. 2.9. Режим обратной связи по выходу OFB

В данном режиме ошибки не распространяются, что является преимуществом при передаче шифрованных речевых сигналов и видеоизображений. Блочный шифр в данном режиме можно рассматривать как СПК. В связи с этим, при использовании режима OFB чрезвычайно важно сохранять синхронизм.

2.4.3. Усложнение блочных криптосистем

Постоянное развитие методов криптоанализа не позволяет долгое время использовать блочную криптосистему без определенного рода ее усовершенствований, которые усложняют работу криптоаналитика. Наиболее простым методом усложнения блочных криптосистем является увеличение длины ключа, однако этот метод не всегда приемлем и, к тому же, требует существенной перестройки базовой блочной криптосистемы. Другим методом усложнения является многократное шифрование с использованием базовой блочной криптосистемы. Этот метод применим к любой блочной криптосистеме, но его использование снижает скорость шифрования. Рассмотрим различные схемы многократного шифрования.

Простейшая схема кратного шифрования - **двойное шифрование** с использованием независимых ключей:

$$y_i = f_{k_2}\left(f_{k_1}\left(x_i\right)\right),\ i=\overline{1,n}. \qquad (2.22)$$

Эта схема была отвергнута сразу, так как ключи можно определить по открытому тексту и криптограмме методом согласования.

Другой способ двойного шифрования, называемый **методом Дэвиса-Прайса**, построен на идеи режима шифрования CBC:

$$y_i = f_{k_2}\left(x_i \oplus f_{k_1}\left(y_{i-1}\right)\right),\ i=\overline{1,n}. \qquad (2.23)$$

Более стойкие схемы используют тройное шифрование. Схему тройного шифрования Тачмена с парой независимых ключей называют часто **схемой EDE**:

$$y_i = f_{k_1}\left(f_{k_2}^{-1}\left(f_{k_1}\left(x_i\right)\right)\right),\ i=\overline{1,n}. \qquad (2.24)$$

При $k_1 = k_2$ эта схема равносильна однократному шифрованию.

Наиболее надежной схемой тройного шифрования является **схема тройного шифрования с тремя независимыми ключами**:

$$y_i = f_{k_3}\left(f_{k_2}^{-1}\left(f_{k_1}\left(x_i\right)\right)\right),\ i = \overline{1,n}. \qquad (2.25)$$

Еще одна схема усложнения блочный криптосистемы, определяемая выражением:

$$y_i = k_2 \oplus f_k\left(k_1 \oplus x_i\right),\ i = \overline{1,n}, \qquad (2.26)$$

использует «зашумляющие» ключи и называется **схемой Рона Ривеста**. Здесь ключи k_1, k_2 являются не ключами шифрования, а «зашумляющими» ключами.

К методам многократного шифрования относится и **схема двойного гаммирования**:

$$y_i = x_i \oplus \gamma_i^{(1)} \oplus \gamma_i^{(2)},\ i = \overline{1,n}. \qquad (2.27)$$

Гаммы $\gamma_i^{(1)}$ и $\gamma_i^{(2)}$ генерируются с использованием независимых ключей k_1, k_2.Перечисленные схемы кратного шифрования не являются единственными. Существует множество схема с использованием нескольких алгоритмов шифрования, переменным размером ключей и обрабатываемых блоков [11].

2.4.4. Блочная криптосистема DES

Криптосистема DES – итеративная 16-раундовая обратимая блочная криптосистема на основе схемы Фейстеля. Размер входного блока - 64 бита. Размер ключа – 64 бита, причем каждый восьмой бит ключа, являющийся двоичной суммой предыдущих семи бит, является служебным и в шифровании не участвует. Раундовые ключи $k_1, k_2, ..., k_{16}$ есть алгоритмически вырабатываемые выборки 48 бит из 56 бит ключа криптосистемы.

Криптосистема DES была принята в качестве национального стандарта шифрования в США и опубликована в 1975 году. Это был беспрецедентный случай в истории криптографии. Открытое опубликование криптосистемы DES привело к тому, что эта криптосистема, как никакая другая, тщательно изучалась криптоаналитиками всего мира.

Процесс криптопреобразования включает три этапа (см. рис. 2.10) [2,11]:

- биты исходного сообщения подвергаются начальной перестановке P;
- полученный блок разбивается на две равные части и подвергается 16-ти раундовому шифрованию по схеме Фейстеля;
- полученный после 16-го раунда блок подвергается конечной перестановке IP^{-1}.

Структурная схема функции усложнения схемы Фейстеля представлена на рис. 2.11. Функция усложнения $\psi(\cdot)$ состоит из следующих слоев:

1) перестановки с расширением (PE) 32 битового вектора до 48 битового вектора;

2) подмешивания 48 битового раундового ключа путем операции сложения по mod2;

3) нелинейной замены с помощью S-блоков 48 битового вектора на 32 битовый вектор;

4) перестановки P координат 32 битового вектора.

Алгоритм формирования раундовых ключей из основного ключа криптосистемы состоит из следующих этапов:

1) из 64 битового основного ключа криптосистемы устраняются 8,16,…,64 служебные биты, а оставшиеся 56 бит подвергаются перестановке P_3;

Рис. 2.10. Схема алгоритма DES

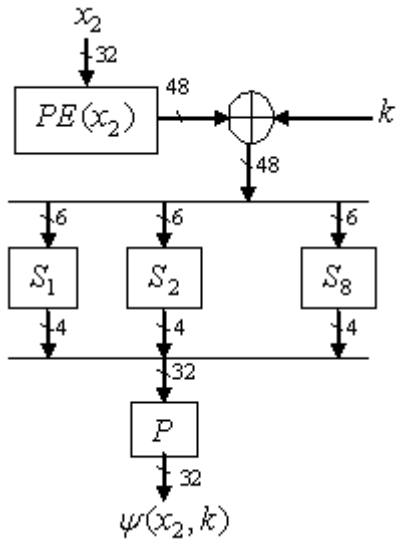

Рис. 2.11. Функция усложнения

2) выходной 56 битовый блок после перестановки разбивается на две равные части по 28 бит, каждая из которых подвергается циклическому сдвигу, причем при $i = 1,2,9,16$ сдвиг осуществляется на одну позицию влево, в остальных случаях – сдвиг на две позиции влево;

3) осуществляется конкатенация 28 битовых блоков;

4) результат конкатенации подвергается перестановке P_4, на выходе которой образуется 48 битовый раундовый ключ.

За последние два десятилетия вычислительная техника развивалась настолько быстро, что временные и стоимостные затраты на реализацию криптоатаки на криптосистему DES постоянно снижались. Это привело к тому, что использование криптосистемы DES не удовлетворяет требованиям скрытности информации. В настоящее время используются варианты усложненной криптосистемы DES. Наиболее широко известна криптосистема 3DES («тройной DES»). Схема усложнения, применяемая в данной

криптосистеме, описывается выражением (2.25). Таким образом, ключ криптосистемы 3DES имеет длину 58·3=168 бит. Криптосистема 3DES примерно в три раза медленнее, чем криптосистема DES. Во многих системах защиты информации такое уменьшение скорости шифрования является неприемлемым. В 1984 году Рон Ривест предложил схему усложнения криптосистемы, которая определяется выражением (2.26). Эта криптосистема получила название DESX (DES eXtended) и оказалась свободной от недостатков 3DES. Ключ криптосистемы DESX состоит из 56+64+64=184 бит и включает основной ключ шифрования и два «зашумляющих» ключа. Дальнейшим развитием криптосистемы DESX стала криптосистема DES-PEP, в схеме усложнения которой операции сложения по модулю 2 были заменены на операции сложения по модулю 2^{32}:

$$y_i = k_2 \lozenge f_k(k_1 \lozenge x_i), \; i = \overline{1,n}, \qquad (2.28)$$

где $\lozenge$ обозначает операцию сложения по модулю 2^{32}.

2.4.5. Блочная криптосистема ГОСТ 28147-89

В Российской Федерации установлен единый стандарт криптографического преобразования для информационных систем. Он носит обязательный характер для государственных организаций, предприятий, банковских и иных учреждений, чья деятельность связана с обеспечением информационной безопасности государства. Для других организаций и частных лиц ГОСТ 28147-89 имеет рекомендательный характер. Данный стандарт формировался с учетом мирового опыта и, в частности, были приняты во внимание недостатки и нереализованные возможности DES. Алгоритм шифрования построен с использованием схемы Фейстеля. **Криптосистема ГОСТ 28147-89** – блочный 32 раундовый итерационный шифр [2,10,11,14]. Размер входного блока 64 бита, размер ключа - 256 бит. Алгоритм шифрования включает в себя следующие этапы:

- 64 битный блок открытого текста разбивается на две равные части;
- 32 битные подблоки подвергаются итеративному процессу шифрования по схеме Фейстеля;
- полученный на 32 раунде 64 битный блок подвергается транспозиции, т.е. левый и правых 32 битные подблоки меняются местами.

Основное отличие схемы Фейстеля криптосистемы ГОСТ от схемы Фейстеля криптосистемы DES – лишь в конструкции функции усложнения. Функция усложнения представленная на рис. 2.12 имеет следующие слои:

1) подмешивание 32 битового раундового ключа путем суммирования по модулю 2^{32};
2) нелинейную замену с помощью S-блоков;
3) перемешивание координат 32 битового вектора с помощью циклического сдвига на 11 бит влево.

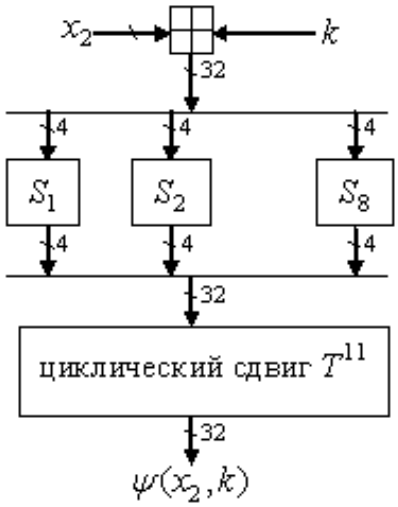

Рис. 2.12. Функция усложнения криптосистемы ГОСТ 28147-89

Отличительной особенностью отечественной криптосистемы является то, что S-блоки выбираются для каждой сети отдельно и, по сути, служат долговременным ключом алгоритма шифрования. Алгоритм выработки раундовых ключей заключается в следующем. Исходный 256 битный ключ делится на восемь 32 битовых подключей, которые используются как раундовые в следующем порядке: 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 8 7 6 5 4 3 2 1. При расшифровании порядок использования ключей меняется на противоположный.

Криптосистема ГОСТ 28147-89 является достаточно стойкой, на сегодняшний день не известно никаких реальных подходов, позволяющих дешифровать криптограммы, не имея ключа. Вместе с тем российский стандарт имеет ряд недостатков, общих с DES. Во-первых, на одном и том же ключе одинаковые 64 битовые блоки перейдут в одинаковые блоки криптограммы. Во-вторых, при использовании простой замены (S-блоки) легко незаметно произвести подмену одной криптограммы, или ее части, другой криптограммой (если они зашифрованы на одном ключе), можно также поменять местами отдельные участки одной криптограммы.

2.4.6. Конкурс AES и блочная криптосистема Rijndael

В 1997 году Национальный институт стандартов и технологий США (NIST) объявил о начале программы AES (Advanced Encryption Standard) по принятию нового стандарта криптографической защиты [2,10,11] взамен устаревшему стандарту DES. Требования к кандидатам: криптоалгоритм должен быть открыто опубликован; криптоалгоритм должен быть симметричным блочным шифром, допускающим размеры ключей в 128, 192 и 256 бит; криптоалгоритм должен быть предназначен как для аппаратной, так и для программной реализации; криптоалгоритм должен быть доступен для открытого использования в любых продуктах; криптоалгоритм подвергается

изучению по следующим параметрам: стойкость, стоимость, гибкость, реализуемость в smart-картах.

На конкурс приняты 15 алгоритмов из 12 стран. В финал конкурса вышли криптосистемы: MARS, TWOFISH, RC6, Rijndael, SERPENT. **Криптосистема MARS** выставлена на конкурс фирмой IBM и по своей структуре может быть отнесена к модифицированным шифрам Фейстеля. Достоинствами криптосистемы является высокий уровень защищенности, потенциальная возможность поддержки ключа размером более 256 бит, высокая эффективность на 32 разрядных платформах. Недостаток криптосистемы состоит в сложности ее конструкции. Это пожалуй самая сложная криптосистема, представленная на конкурс. **Криптосистема TWOFISH** представлена на конкурс Б. Шнайдером. По своей структуре криптосистема является классическим шифром Фейстеля. Главная особенность криптосистемы – меняющиеся в зависимости от ключа таблицы замен. Достоинствами являются: высокий уровень защиты, удобная реализация в smart-картах, высокая эффективность на любых платформах (в том числе и на 64 разрядных), вычисление раундовых ключей «на лету», допускает произвольную длину ключа до 256 бит. К недостаткам можно отнести высокую сложность алгоритма, что затрудняет его аппаратную и программную реализации. **Криптосистема RC6** представлена на конкурс фирмой RSA Lab и по своей структуре может быть также отнесена к модифицированным шифрам Фейстеля. Достоинствами криптосистемы является: простая структура алгоритма, быстрая процедура формирования ключа, потенциальная возможность поддержки ключа размером более 256 бит, длина ключа и число раундов могут быть переменными, высокая эффективность на 32 разрядных платформах. К недостаткам можно отнести: относительно низкий уровень защищенности, невозможность формирования раундовых ключей «на лету». **Криптосистема SERPENT** представлена тремя известными криптоаналитиками Р. Андерсеном, Э. Бихамом, Л. Кнудсеном. Криптосистема является классической SP-сетью. Достоинствами криптосистемы является: высокий уровень защищенности, удобная реализация в smart-картах. К недостаткам относится низкая скорость шифрования. Это самая медленная из всех представленных на конкурс криптосистем. В 2000 году конкурс завершился и победителем была признана криптосистема Rijndael, как имеющая наилучшее сочетание стойкости, стоимости, производительности, эффективности реализации и гибкости. Авторами криптосистемы являются Винсент Райман и Йоан Дамен.

Криптосистема Rijndael, в настоящее время известная больше как **AES,** представляет собой алгоритм шифрования не использующий схему Фейстеля [2,11]. Криптосистема имеет ключи размером 128, 192 и 256 бит, входные блоки могут иметь длину 128, 192 и 256 бит. Количество раундов 10, 12 или 14 в зависимости от длины ключа.

Промежуточные результаты преобразований, выполняемые в рамках криптопреобразования, называют состояниями (State). Состояние можно представить в виде прямоугольного массива байтов. При размере блока, равном 128 бит этот 16-ти байтовый массив имеет 4 строки и 4 столбца (каждая строка или столбец рассматривается как 32 разрядное слово). Входные данные для криптоалгоритма обозначаются как байты состояния. После шифрования выходные данные получаются из байтов состояния в том же порядке. Число столбцов N_b блока данных равно длине блока деленной на 32. Ключ шифрования также представляется в виде прямоугольного массива с четырьмя строками. Число столбцов N_k массива равно длине ключа деленной на 32.

В табл. 2.2 представлены форматы данных блока и ключа шифрования для случая, когда N_b=4 и N_k=4. В таблице s_{ij} - байт массива State, k_{ij} - байт ключа. Число раундов N_r зависит от значений N_b и N_k (см. табл.2.3).

Таблица. 2.2. Форматы данных блока и ключа шифрования

a_{00}	…	a_{03}
…	…	…
a_{30}	…	a_{33}

K_{00}	…	K_{03}
…	…	…
K_{30}	…	K_{33}

Таблица. 2.3. Зависимость числа раундов алгоритма N_r от N_b и N_k.

N_r	N_b=4	N_b=6	N_b=8
N_k=4	10	12	14
N_k=6	12	12	14
N_k=8	14	14	14

Раунд криптоалгоритма (цикловая функция) состоит из четырех различных преобразований:

- замены байтов SubBytes() – побайтовой замены в S-блоках с фиксированной таблицей замен размерностью 8×256;

- сдвига строк ShiftRows() – побайтового сдвига строк массива State на различное количество байт;

- перемешивания столбцов MixColumns() – умножения столбцов состояния, рассматриваемых как многочлены над $GF(2^8)$, на многочлен третьей степени $g($ по модулю x^4+1;

- сложение с раундовым ключом AddRoundKey() – поразрядного XOR с текущим фрагментом развернутого ключа.

Замена байтов SubBytes(). Преобразование SubBytes() представляет собой нелинейную замену байтов, выполняемую независимо с каждым байтом состояния. Таблицы замены S-блока являются инвертируемыми и построены из композиции следующих двух преобразований входного байта:

1) получение обратного элемента относительно умножения в поле $GF(2^8)$, нулевой элемент 00 переходит сам в себя;

2) применение преобразования над $GF(2)$ определяемого уравнением:

$$b_i' = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i, \quad (2.29)$$

где $c_0 = c_1 = c_5 = c_6 = 1$, $c_2 = c_3 = c_4 = c_7 = 0$, b_i и b_i' - соответственно исходное и преобразованное значение $i-$го бита, $i = \overline{0,7}$. На рис. 2.13 иллюстрирует применение преобразования SubBytes() к состоянию.

Сдвиг строк ShiftRows() – последние три строки состояния циклически сдвигаются влево на различное число байтов. Значение сдвигов зависит от длины блока N_b и составляет: для N_b=4 – 10, 12, 14; для N_b=6 – 12, 12, 14; для N_b=8 – 14,14,14. На рис. 2.14 иллюстрируется применение преобразования ShiftRows().

Перемешивание столбцов MixColumns(). В этом преобразовании столбцы состояния рассматриваются как многочлены над $GF(2^8)$ и умножаются по модулю $x^4 + 1$ на многочлен:

$$g(x) = \{03\} x^3 + \{01\} x^2 + \{01\} x + \{02\}. \quad (2.30)$$

Это можно представить в матричном виде:

$$\begin{bmatrix} s_{0c}' \\ s_{1c}' \\ s_{2c}' \\ s_{3c}' \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0c} \\ s_{1c} \\ s_{2c} \\ s_{3c} \end{bmatrix}, \quad (2.3)$$

где $0 \le c \le 3$ - номер столбца массива State. На рис. 2.15 иллюстрирует применение преобразования MixColumns().

Сложение с раундовым ключом AddRoundKey(). В данной операции раундовый ключ добавляется к состоянию посредством простого поразрядного XOR. Длина раундового ключа равна длине блока.

Алгоритм шифрования, таким образом, состоит из начального добавления раундового ключа, реализации N_r-1 раундов цикловой функции и заключительного раунда в котором отсутствует операция MixColumns().

Более подробно криптосистема Rijndael описана в [2,10,11]. Стандарт шифрования FIPS-197, реализующий криптоалгоритм Rijndael, вступил в силу с 2002 года.

Кроме рассмотренных выше криптосистем достаточно широко используются такие криптосистемы, как IDEA (International Data Encryption Algorithm), SAFER+ и SAFER++.

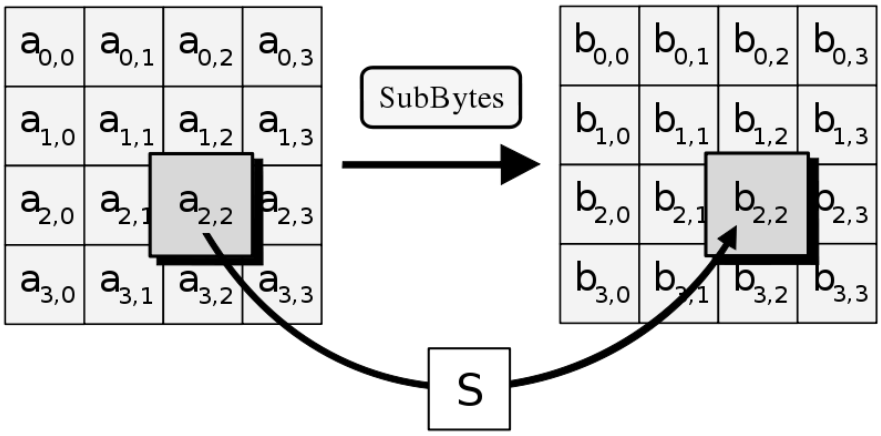

Рис. 2.13 Преобразование SubBytes()

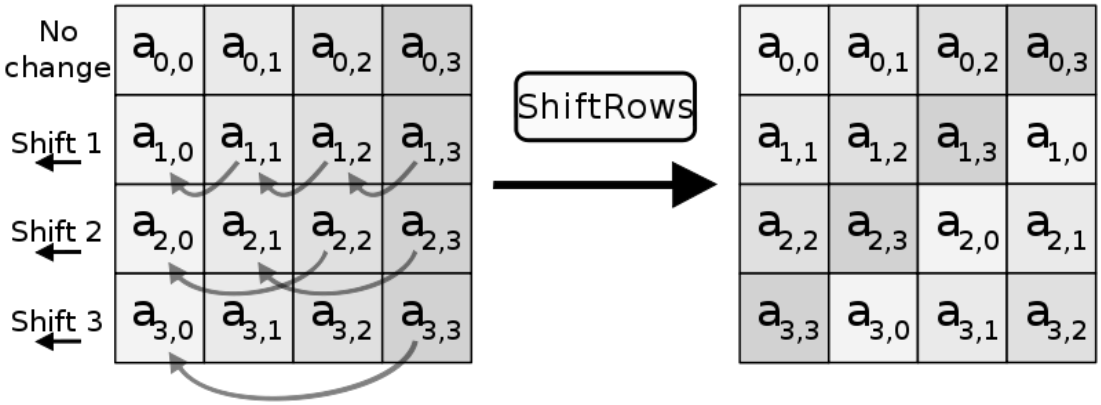

Рис. 2.14. Преобразование ShiftRows()

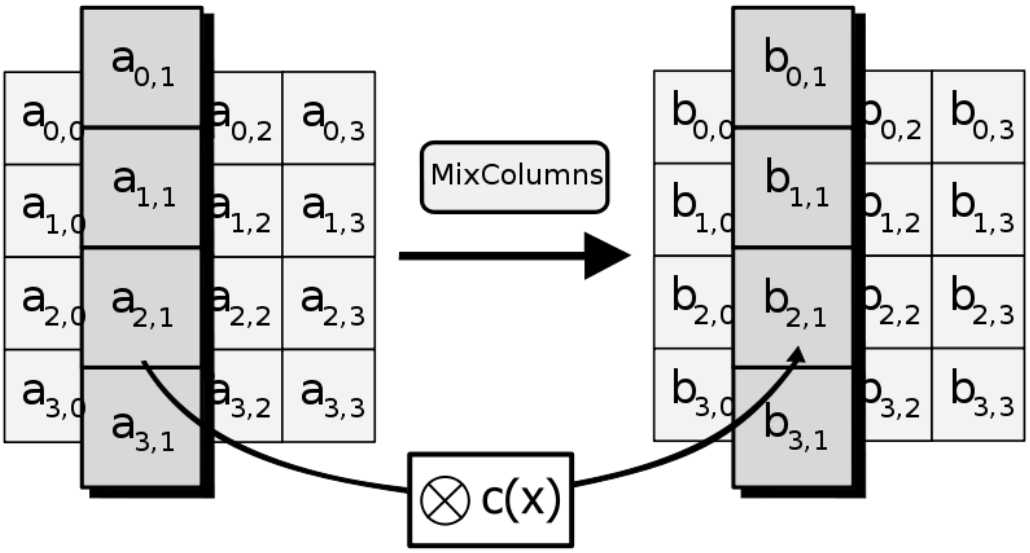

Рис. 2.15 Преобразование MixColumns()

3. Методы криптоанализа симметричных криптосистем

По определению известного американского криптолога У. Фридмана «криптоанализ включает определение используемого языка, типа криптосистемы, ключа и исходного текста; обычно именно в этом порядке». Хотя определение криптоанализа было введено сравнительно недавно, первым известным письменным упоминанием о криптоанализе является «Книга о большом стремлении человека разгадать загадки древней письменности», написанная арабским учёным Абу Вакр бен Али бен Вахшия ал-Набати в средние века. В настоящее время криптоанализ активно развивается, хотя единая математическая теория криптоанализа еще не разработана, и на рынке уже появились пакеты прикладных программ по криптоанализу. В частности, разработкой таких программных продуктов занимается американская фирма Access Data Recovery.

3.1. Задачи и принципы криптоанализа

Следует заметить сразу, что основная цель криптоанализа состоит не столько в получении скрываемой информации, а в оценке стойкости существующих и разрабатываемых криптосистем. Оценка стойкости криптосистем представляется в виде количества операций, необходимых для взлома криптосистемы или в виде времени, которое требуется для взлома.

Приведем основные принципы, которые были «выстраданы» криптологами [5,10]:

1. Принцип Керкхоффа. Только криптоаналитик может судить о криптостойкости системы.

2. Принцип Керкхоффа-Шеннона. Противник знает используемую криптосистему с точностью до ключевой информации.

3. Принцип Жеверже. Поверхностные усложнения криптосистемы могут быть иллюзорны, так как порождают ложные оценки ее криптостойкости.

4. При оценке криптостойкости необходимо учитывать возможные криптографические ошибки и другие нарушения дисциплины безопасности.

Из приведенных выше принципов следует, что основная задача криптоаналитика заключается в оценке ключевой информации, при условии, что сама используемая криптосистема известна. Алгоритм оценки ключевой информации называется **криптоатакой**. В зависимости от условий взаимодействия криптоаналитика с криптосистемой различают следующие основные типы криптоатак:

− криптоатака с использованием только криптограмм (A1);

− криптоатака с использованием открытых текстов и соответствующих им криптограмм (A2);

– криптоатака с использованием выбираемых криптоаналитиком открытых текстов и соответствующих им криптограмм (А3);

– криптоатака с использованием аппаратного воздействия на криптосистему (криптоатака по сторонним каналом) (А4).

Последний тип криптоатак А4 предполагает не исследования теоретического описания криптографического алгоритма, а анализ данных, полученных в результате наблюдения за физическим процессом работы устройства, реализующего криптографический алгоритм. К этому типу криптоатак относятся: криптоатака по времени, криптоатака по энергопотреблению, криптоатака по электромагнитному излучению, криптоатака на основе акустического анализа.

В настоящее время разработано множество методов криптоанализа, реализующих криптоатаки типов А1-А3. В настоящей главе рассмотрим несколько этих типов криптоатак.

3.2. Метод полного перебора

Метод полного перебора или **метод «грубой силы»** (brute force attack) является простейшим методом криптоанализа, и несмотря на свой «солидный возраст» в настоящее время, в связи с интенсивных развитием компьютерной техники, находит достаточно широкое применение. Суть метода заключается в следующем:

1) на основе имеющейся криптограммы y и соответствующего ей открытого текста x составляется система уравнений относительно $k = (k_1, k_2, ..., k_n)$:

$$f^{-1}(y, k_i) = x_i,\ i = \overline{1,n}, \qquad (3.1)$$

полным перебором всех возможных значений ключевого параметра находится некоторое множество решений l;

2) если количество найденных решений $l = 1$, то считается что истинное значение ключевого параметра определено, если $l > 1$ (несколько осмысленных открытых текстов), то определено подмножество значений искомого ключевого параметра и в этом случае целесообразно увеличить количество уравнений в системе.

Если считать проверку одного варианта ключа $k_i \in K$ в выражении (3.1) за одну операцию, то полный перебор ключей потребует $|K|$ операций. Пусть ключ в алгоритме шифрования выбирается случайно и равновероятно из множества K. Тогда с вероятностью $\frac{1}{|K|}$ трудоемкость метода полного перебора равна 1. Это происходит в том случае, когда случайно выбран ключ, расположенный в нашем порядке на первом месте. Поэтому естественно в

качестве оценки трудоемкости метода взять математическое ожидание числа шагов в переборе до попадания на использованный ключ. Пусть случайная величина ε - число опробований включительно до момента обнаружения использованного ключа. При $i=\overline{1,|K|}$ случайные величины $\xi_i=1$, если использованный ключ находиться в порядке на месте i и $\xi_i=0$ в противном случае. Тогда:

$$M[\varepsilon]=\sum_{i}^{|K|} iP(\xi_i=1). \qquad (3.2)$$

Если считать, что все ключи расположены в установленном порядке, то процедуру равновероятного выбора ключа можно представить как равновероятный выбор числа i в последовательности натуральных чисел $1,...,|K|$. Тогда $P(\xi_i=1)=\frac{1}{|K|}$ для любого $i=\overline{1,|K|}$. Подставляя полученное значение в (3.2) имеем:

$$M[\varepsilon]=\frac{1}{|K|}\sum_{i}^{|K|} i=\frac{|K|(|K|+1)}{2|K|}. \qquad (3.3)$$

При больших $|K|$ можно приблизительно считать, что $M[\varepsilon]\approx\frac{|K|}{2}$. Например, для криптосистемы DES $|K|=2^{56}$, тогда средняя трудоемкость полного перебора $\frac{|K|}{2}=2^{55}\approx 0{,}5\cdot 10^{17}$. При использовании современных вычислительных систем, среднее время взлома шифра Цезаря составляет $1{,}2\cdot 10^{-11}$ секунды, а шифра Вернама - $2\cdot 10^{162}$ лет.

Метод полного перебора допускает распараллеливание. Один из способов распараллеливания заключается в том, что множество натуральных чисел $1,...,|K|$ разбивается на непересекающиеся подмножества $B_1,...,B_R$. Система из R вычислительных машин перебирает ключи так, что i-ая машина осуществляет перебор ключей из множества B_i, $i=\overline{1,R}$. Система прекращает работу, если одна из вычислительных машин определила ключ. Например, если одна вычислительная машина опробует ключ за 10^{-6} секунд, то для того чтобы найти ключ криптосистемы DES полным перебором за 24 часа требуется $R=\frac{10^{17}}{2\cdot 864\cdot 10^{8}}=5{,}787\cdot 10^{5}$ вычислительных машин.

3.3. Методы бесключевого чтения

При рассмотрении поточных криптосистем гаммирования были определены требования к гамме. Нарушение этих требований дает противнику возможность реализовать криптоатаку на криптосистему. Рассмотрим некоторые методы криптоанализа криптосистем гаммирования не требующие определения ключа [11].

Метод чтения в колонках реализует способ восстановления текста, зашифрованного неравновероятной гаммой. Суть метода состоит в следующем. Пусть p_i есть вероятность использования знака i в гамме, причем полагаем, что не все вероятности p_i равны $\frac{1}{n}$, где n - длина гаммы. Без ущерба для общности положим $p_0 \geq p_1 \geq ... \geq p_{r-1} > 0$, $p_r = p_{r+1} = ... = p_{m-1} = 0$, $r < m$. В i-м такте шифрованию подверглась одна из r следующих букв открытого текста, $i = 1,2,...,N$:

$$x_i^{(0)} = y_i,\ x_i^{(1)} = (y_i - 1) \bmod m,\ \ldots,\ x_i^{(r-1)} = (y_i - r + 1) \bmod m. \qquad (3.4)$$

Открытый текст может быть получен подбором i-го знака в i-й колонке табл. 3.1 (элементы таблицы рассматриваются по $\bmod m$, где m - мощность алфавита).

Таблица 3.1. Таблица дешифрования

$\gamma \setminus y$	y_1	…	y_i	…	y_N
0	y_1	…	y_i	…	y_N
1	y_1-1	…	y_i-1	…	y_N-1
…	…	…	…	…	…
$r-1$	$y_1 - r + 1$	…	$y_i - r + 1$	…	$y_N - r + 1$

Знаки в каждой колонке упорядочены по вероятности их использования в открытом тексте. Это облегчает чтение в колонках, т.к. «читаемый текст» с повышенной вероятностью расположен в верхних строках таблицы. В случае, когда все $p_i > 0$, но при этом имеют разные значения при составлении таблицы необходимо исключить из рассмотрения $m - r$ наименее вероятных знаков гаммы. В этом случае существует ненулевая вероятность потери истинного решения, т.е. исходного открытого текста.

Применение данного метода тем успешнее, чем меньше высота колонок r, характеризующая многозначность решения задачи. При r близких к m, чтение в колонках теряет смысл, т. к. табл. 3.1 может содержать большое количество осмысленных, но и взаимно противоречивых открытых текстов. Данный метод может быть использован и в случае, когда текст зашифрован периодической гаммой. Пусть дана криптограмма $Y = y_1, y_2, ..., y_n$. Известен

период ключевой последовательности d. Сформируем две подпоследовательности исходной криптограммы:

$y_1, y_2, \ldots, y_i, \ldots y_{(k-1)d+r}$;

$y_{1+d}, y_{2+d}, \ldots, y_{i+d}, \ldots, y_{kd+r}$, $n = kd + r$.

Будем полагать, что открытыми текстами, подлежащими шифрованию, являются содержательные тексты с известными вероятностями букв алфавита $P(a_j) = p_j$, $j = \overline{1,m}$, где j - номер буквы алфавита. Также будем считать, что на множестве K задано равномерное распределение, т.е. ключом является реализация выборки объемом n из равномерного распределения K. Тогда вероятность того, что i-я и $(i+d)$-я буквы открытого текста были равны соответственно, $x_i = s$ и $x_{i+d} = l$, при условии, что i-я и $(i+d)$-я буквы криптограммы равны соответственно, y_i и y_{i+d} определяется выражением:

$$P(x_i = s, x_{i+d} = l \mid y_i, y_{i+d}) = \frac{P(x_i = s, x_{i+d} = l; y_i, y_{i+d})}{P(y_i, y_{i+d})}. \qquad (3.5)$$

Так как в рассматриваемой криптосистеме множество ключей задано латинским квадратом, следовательно, используемый ключ однозначно определен любым переходом в этой замене. Если числитель (3.5) не равен нулю, то справедливо равенство:

$$P(x_i = s, x_{i+d} = l \mid y_i, y_{i+d}) = \frac{p_s p_l}{\sum\limits_{f \in K} p_{f^{-1}(y_i)} p_{f^{-1}(y_{i+d})}} = \tilde{p}_{sl}. \qquad (3.6)$$

Для каждой пары y_i и y_{i+d} букв исходной криптограммы упорядочим в соответствии с убыванием полученных значения условных вероятностей (3.6) пары букв открытого текста s и l.

Построив такие колонки для каждого i, в результате получаем таблицу (см. табл. 3.2), в которой верхние пары имеют большую условную вероятность, чем нижние.

Буквы искомого содержательного текста будут находится вероятнее всего в первых строках и задача сводится к подбору таких пар букв, чтобы в результате получался читаемый текст.

Таблица 3.2. Таблица дешифрования

$i=1$	…	$i=j$	…	$i=n$
$\frac{y_1}{y_{1+d}}$	…	$\frac{y_j}{y_{j+d}}$	…	$\frac{y_{(k-1)d+r}}{y_{kd+r}}$
….	…	$\left(\frac{x_j = s}{x_{j+d} = l}, \tilde{p}_{sl}\right)$	…	…

Период ключевой последовательности d может быть определен с помощью **методов Фридмана** [1]. Рассмотрим эти методы. Основаны методы на введенном У. Фридманом понятии индекса совпадения. **Индексом совпадения** последовательности $X = a_1, a_2, ..., a_n$ называется величина

$$\Im(X) = \sum_{i=1}^{m} \frac{F_i(F_i - 1)}{n(n-1)}, \qquad (3.7)$$

где $X \in A_X$, $|A_X| = m$ - некоторая последовательность; F_i - частота встречаемости (число мест в тексте) i - буквы в последовательности X.

Индекс совпадения $\Im$ последовательности равен вероятности $P_X(j = a_{j'})$ совпадения символов данной последовательности на случайно и равновероятно выбранных местах j и j', причем $j \neq j'$ и $j, j' \in \overline{1,n}$.

При $n \to \infty$

$$\Im(X) = \sum_{i=1}^{m} \frac{F_i(F_i - 1)}{n(n-1)} \to \sum_{i=1}^{m} p_i^2,$$

где p_i - вероятность $i-$го символа из алфавита A_X в содержательных текстах.

При случайном выборе последовательности X индекс совпадения $\Im$ является случайной величиной, и представляет интерес подсчет его математического ожидания. Математическое ожидание индекса совпадения для открытых текстов определяется выражением:

$$M[\Im(X)] = \sum_i p_i^2. \qquad (3.8)$$

Если реализация $X = a_1, a_2, ..., a_n$ представляет собой выборку из равномерного вероятностного распределения, то справедливо выражение:

$$M[\Im(X)] = \frac{1}{m}. \qquad (3.9)$$

Для любой последовательности $X = a_1, a_2, ..., a_n$ вероятность совпадения $P(a_j = a_{j'})$ двух случайно и равновероятно выбранных букв последовательности приблизительно равна:

$$P_X(a_j = a_{j'}) \approx \frac{n}{n-1} \sum_i p_i^2 - \frac{1}{n-1}. \qquad (3.10)$$

Пусть $Y = b_1, b_2, ..., b_n$ - криптограмма, полученная в результате шифрования случайного текста при случайном и равновероятно выбранном ключе. Тогда математическое ожидание индекса совпадения для криптограммы равно:

$$M[\Im(Y)] = P_Y(b_j = b_{j'}) = \frac{1}{m}. \qquad (3.11)$$

Для криптосистемы гаммирования с периодической гаммой, получаемого шифрованием открытого текста $X = a_1, a_2, ..., a_n$ с помощью равновероятного выбора ключа $\tilde{k}$ из множества всех локально-периодических последовательностей K_n^d периода d и $n = sd + r$ справедливо:

$$M[\mathfrak{I}(Y)] = \frac{(s+1)sr + s(s-1)(d-r)}{n(n-1)} \sum_i p_i^2 + \left(1 - \frac{(s+1)sr + s(s-1)(d-r)}{n(n-1)}\right)\frac{1}{m}. \quad (3.12)$$

При $r = 0$ (т.е. $n = sd$):

$$M[\mathfrak{I}(Y)] = \frac{s(s-1)d}{n(n-1)} \sum_i p_i^2 + \left(1 - \frac{s(s-1)d}{n(n-1)}\right)\frac{1}{m} = \frac{n-d}{d(n-1)} \sum_i p_i^2 + \frac{n(d-1)}{(n-1)d}\frac{1}{m}. \quad (3.13)$$

Первый метод Фридмана состоит в том, что вычисляется индекс совпадения $\mathfrak{I}(Y)$ для имеющейся криптограммы в соответствии с выражением (3.7) и затем его значение сравнивается с (3.13) при $d = 1,2,3,....$ При достаточной близости индекса совпадения к одному из значений (3.13), при некотором d, предполагают, что период равен этому значению d.

Первый метод Фридмана эффективен для $d \le 5$, т.к. значение $M[\mathfrak{I}(Y)]$ для фиксированного периода d совпадает со значениями целого ряда различных периодов ключевой последовательности.

Второй метод Фридмана также основан на вычислении индекса совпадения. Суть второго метода Фридмана состоит в опробовании возможных периодов d по следующей схеме. Из исходной криптограммы $Y = b_1, b_2, ..., b_n$ для предполагаемого периода d ключевой последовательности выписывается d подпоследовательностей:

$$\begin{aligned} &1)\ y_1, y_{1+d}, y_{1+2d}, \cdots \\ &2)\ y_2, y_{2+d}, y_{2+2d}, \cdots \\ &\cdots\cdots\cdots \\ &d)\ y_d, y_{d+d}, y_{d+2d}, \cdots \end{aligned} \quad (3.14)$$

Для каждой подпоследовательности подсчитывается ее индекс совпадения $\mathfrak{I}(Y_d)$. Если все индексы совпадения в среднем близки к значению $\frac{1}{d}\sum_i p_i^2$ (среднее значение индекса совпадения случайных криптограмм, полученных с помощью гамм периода 1), то принимают величину d за истинный период, в противном случае опробуется следующая величина

периода. Второй метод Фридмана позволяет эффективно определять периоды $d \le 30$.

Метод восстановления текстов, основанный на атаке с помощью вставки символа (insertion attack). Рассмотрим существо метода. Пусть открытый текст $X = x_1, x_2, x_3, ...$ с помощью гаммы $\gamma = \gamma_1, \gamma_2, \gamma_3, ...$ преобразуется в соответствии с уравнением шифрования:

$$y_i = (x_i + \gamma_i) \bmod m, \qquad (3.15)$$

в криптограмму $Y = y_1, y_2, y_3, ...$ Криптоаналитику известна криптограмма, но не известен открытый текст и гамма. Предположим, что криптоаналитик располагает еще одной криптограммой, полученной зашифрованием той же гаммой видоизмененного открытого текста, полученного из оригинального текста вставкой в некоторой позиции известного бита x'. Пусть видоизмененный открытый текст имеет вид $X' = x_1, x', x_2, x_3, ...,$ а соответствующая криптограмма $Y' = y_1, y'_2, y'_3, ...$

По двум криптограммам, начиная с момента вставки, можно определить и гамму и открытый текст:

$\gamma_2 = y'_2 - x'$, $x_2 = y_2 - \gamma_2$;

$\gamma_3 = y'_3 - x_2$, $x_3 = y_3 - \gamma_3$;

$\gamma_4 = y'_4 - x_4$, $x_4 = y_4 - \gamma_4$, ….

Все вычисления выполняются по $\bmod m$. Момент вставки символа x' можно определить, сравнивая видоизмененную и оригинальную криптограммы. Если значение вставленного символа неизвестны, то можно реализовать подбор вариантов его значения. Для защиты от такого рода атак достаточно никогда не использовать одинаковые отрезки гаммы для повторного шифрования.

3.4. Методы криптоанализа с использованием теории статистических решений

Рассмотрим симметричную криптосистему, криптоаналитику априорно известно:

1) криптографическое преобразование $f(\cdot)$ с точностью до ключа $k \in K$;

2) криптограмма Y длинной n.

Задача криптоаналитика состоит в оценке неизвестного параметра $k \in K$ криптографического преобразования $f(\cdot)$.

Допущения (предположения) [10]:

1) исходный текст X представляет собой случайную последовательность с некоторым заданным дискретным распределением вероятностей $P\{x = a\} \equiv P\{x_1 = a_1, ..., x_n = a_n\} \equiv p_n(a)$, $0 \le p(a) \le 1$, $\sum_{a \in A} p_n(a) = 1$.

2) $k \in K$ - является детерминированной неизвестной величиной.

Метод частотного криптоанализа базируется на реализации методов теории статистических решений, а именно, на методе максимального правдоподобия [10]. В соответствии с этим методом формируется функция правдоподобия:

$$L(k) = q_n(y_1, y_2, ..., y_n \mid k), \tag{3.16}$$

где $q_n(b \mid k) = P\{Y = b \mid k\}$ - дискретное распределение криптограммы Y в ситуации, когда значение ключа k фиксировано:

$$q_n(b \mid k) = P\{Y = b \mid k\} = P\{f(X, k) = b\} = P\{X = f^{-1}(b, k)\} = p_n(f^{-1}(b, K)) = p_n(a). \tag{3.17}$$

Тогда:

$$L(k) = p_n(f^{-1}(Y, k)). \tag{3.18}$$

Функция правдоподобия $L(k)$ позволяет судить, насколько правдоподобно получить криптограмму Y при условии, что ключ равен k. На практике оперируют не самой функцией правдоподобия, а ее монотонным преобразованием, в частности логарифмом функции правдоподобия:

$$l(k) = \log L(k) = \log p_n(f^{-1}(Y, k)). \tag{3.19}$$

Для получения оценочного значения ключа требуется отыскать такое его значение, которое максимизирует значение функционала:

$$k^* = \arg\max_{k \in K} L(k) = \arg\max_{k \in K} l(k). \tag{3.20}$$

Рассмотрим применение частотного метода криптоанализа к криптосистеме Цезаря для двух случаев:

1) источник открытых сообщений представляет собой стационарный источник дискретных сообщений без памяти;

2) источник открытых сообщений представляет собой однородную цепь Маркова.

Случай 1. Стационарный источник дискретных сообщений без памяти. Открытый текст представляется в виде:

$$X(k) = (x_1(k), x_2(k), ..., x_n(k)) = f^{-1}(Y, k) = (f_1^{-1}(Y, k), ..., f_n^{-1}(Y, k)). \tag{3.21}$$

Введем величину, имеющую смысл частоты встречаемости символа $j \in A_N$ в $X(k)$:

$$\nu_j(Y, k) = \sum_{i=1}^{n} \delta_{j, X_i(k)} = \sum_{i=1}^{n} \delta_{j, f^{-1}(Y, k)}, \tag{3.22}$$

где $\delta_{j,k}$ - символ Кронекера, $\sum_{j=0}^{N-1} \nu_j(Y, k) = n$ - условие нормировки.

Совместное n-мерное дискретное распределение вероятностей исходного текста имеет вид:

$$p_n(a) = p(a_1, a_2, ..., a_n) = \prod_{i=1}^{n} p_i(a_i) = \prod_{i=1}^{n} p_1(f^{-1}(Y, K)), \qquad (3.23)$$

где $p_1(\cdot)$ - распределение вероятностей для одного символа алфавита A.

Тогда логарифм функционала правдоподобия имеет вид:

$$l(k) = \log \prod_{i=1}^{n} p_1(f^{-1}(Y,k)) = \sum_{i=1}^{n} \log(p_1(f^{-1}(Y,k)) = \\ = \sum_{j=0}^{N-1} \nu_1(Y,k) \log(p_1(j)) = \sum_{j=0}^{N-1} \nu_j(Y,k) \log p_1(j). \qquad (3.24)$$

Тогда:

$$k^* = \arg\max_{k \in K} \sum_{j=0}^{N-1} \nu_j(Y,k) \log p_1(j). \qquad (3.25)$$

Например, для криптосистемы Цезаря выражение (3.20) будет иметь вид:

$$k^* = \arg\max_{k \in K} \sum_{j=0}^{N-1} \nu_{(j+k) \bmod N} \log p_1(j). \qquad (3.26)$$

Случай 2. Марковский источник открытых сообщений (однородная цепь Маркова). Открытый текст представляет собой реализацию однородной цепи Маркова, которая задана:

- матрицей вероятностей переходов $p_{js} = P\{x_{i+1} = s \mid x_i = j\}$;

- вектором начального распределения вероятностей $p_1(u)$.

Введем в рассмотрение выражение для частот встречаемости биграмм (j, s):

$$\nu_{js}(Y,k) = \sum_{i=1}^{n-1} \delta_{j, f_i^{-1}(Y,k)} \delta_{s, f_{i+1}^{-1}(Y,k)}, \qquad (3.27)$$

При этом должно выполняться условие нормировки $\sum_{j,s=0}^{N-1} \nu_{js}(Y,k) = n-1$.

Оценка ключа будет определяться в соответствии с выражением:

$$k^* = \arg\max_{k \in K} \left\{ \sum_{j=0}^{N-1} \delta_{j, f_i^{-1}(Y,k)} \log p_1(j) + \sum_{j,s=0}^{N-1} \nu_{js}(Y,k) \log p_{js} \right\}. \qquad (3.28)$$

Например, для криптосистемы Цезаря оценка ключа будет иметь вид:

$$k^* = \arg\max_{k \in K} \left\{ \sum_{j=0}^{N-1} \delta_{y_1, (j+k) \bmod N} \log p_1(j) + \sum_{j,s=0}^{N-1} \nu_{(j+k) \bmod N, (s+k) \bmod N}(Y) \log p_{js} \right\}$$

Точность решения задачи криптоанализа частотным методом, т.е. точность оценки k^* характеризуется вероятностью ошибки:

$$r = P_k\{k^*(Y) \neq k\} = r(n,k) = 1 - \sum_{b \in A_N} p_n(f^{-1}(b,k))\delta_{k^*,k}. \qquad (3.29)$$

Алгоритм криптоанализа является состоятельным, если при увеличении длины криптограммы $n \to \infty$ вероятность ошибки стремиться к нулю $r(n,K) \to 0$. Более сложные выражения для оценки ключа k^* получаются, если снять допущение о детерминированности ключа. Алгоритм криптоанализа в этом случае базируется на байесовском методе теории статистических решений и подробно описан в [10].

3.5. Линейный криптоанализ

Метод линейного криптоанализа разработан в 1993 году японским криптологом Митсуру Матсуи. В первоначальном виде этот метод сформулирован применительно к криптоанализу криптосистемы DES. В настоящее время создаются новые модификации этого метода [10].

Идея метода линейного криптоанализа основана на том, что существует возможность заменить нелинейную функцию криптографического преобразования ее линейным аналогом. Линейный криптоанализ базируется на знании криптоаналитиком пар «открытый текст - криптограмма», а также алгоритма шифрования. Будем считать, что при генерации исходного текста X случайные биты независимы и равновероятны $P(x_i = 1) = p$, $P(x_i = 0) = 1 - p$, $p = 0{,}5$. **Линейным статистически аналогом** (или приближенным линейным аналогом) называется:

$$\lambda(X,Y) = \sum_{i=1}^{n} a_i x_i \oplus \sum_{i=1}^{n} b_i y_j = \sum_{k=1}^{L} c_k k_k, \qquad (3.30)$$

если вероятность $P\left\{\lambda(X, f(X,K)) = \sum_{k=1}^{L} c_k k_k\right\} = 0{,}5 + \Delta$. Величина $\Delta = |1 - 2p|$ называется **эффективностью линейного аналога**, а коэффициенты $a_i = \overline{0,1}$, $b_i = \overline{0,1}$, $c_k = \overline{0,1}$ - параметрами линейного аналога. По существу Δ характеризует степень линейности функции криптографического преобразования и имеет максимальное значение $\Delta_{\max} = 0{,}5$.

При применении метода линейного криптоанализа решаются две взаимосвязанные задачи [4,10]:

1) нахождение эффективного линейного статистического аналога и вычисление его вероятности;

2) определение ключа (или нескольких бит ключа) с использованием эффективного линейного статистического аналога.

Практическая реализация метода линейного криптоанализа связана с реализацией следующих последовательных шагов.

1. Тщательно анализируется криптографическая функция и определяется множество линейных статистических аналогов. На этом шаге в первую очередь анализируются S-блоки функции усложнения ψ. Для этого необходимо для каждого S-блока сформировать таблицы значений $Q_t(i, j)$, где: t - номер S-блока. Значение $Q_t(i, j)$ представляет собой количество совпадений суммы по mod2 некоторых битов входных данных с суммой по mod2 некоторых битов выходных данных. В ходе анализа прослеживаются все возможные комбинации двоичных векторов i, j. Каждая пара векторов используется в качестве маски, которая накладывается на возможные пары «вход-выход» S-блока. Эти маски указывают на биты входа и выхода, которые необходимо сложить по mod2, а затем сравнить полученные результаты. Далее проводится анализ полученных таблиц $Q_t(i, j)$ и отыскивается такие значения i^*, j^*, для которых выполняется условие

$$Q_t(i^*, j^*) : \max\left|Q_t(i, j) - n_X\right|, \qquad (3.31)$$

где n_X - длина подблока.

В соответствии с полученной парой i^*, j^*, и учитывая в схеме алгоритма шифрования перестановки и сложение по mod2, формируется эффективный линейный статистический аналог

$$\lambda^*(X,Y) = \sum_{i=1}^{n} a_i^* x_i \oplus \sum_{i=1}^{n} b_i^* y_j = \sum_{k=1}^{L} c_k^* k_k \text{, при } P_{эа} = \frac{Q(i^*, j^*)}{2n_X}. \qquad (3.32)$$

Как правило, формируются несколько линейных аналогов (один из них эффективный) для которых значения вероятностей близки.

2. Генерируется множество независимых исходных текстов $X^{(1)}, X^{(2)}, ..., X^{(M)}$ и регистрируются соответствующие им криптограммы $Y^{(1)}, Y^{(2)}, ..., Y^{(M)}$.

3. Для каждой пары $X^{(m)}, Y^{(m)}$, $m = \overline{1, M}$ вычисляется значение левой части эффективного линейного статистического аналога:

$$\lambda^*(X^{(m)}, Y^{(M)}) = \sum_{i=1}^{n} a_i^* x_i^m \oplus \sum_{i=1}^{n} b_i^* y_i^m . \qquad (3.33)$$

4. Определяется частота получения «1» при вычислении M значений (3.33):

$$\nu = \frac{1}{M} \sum_{m=1}^{M} \lambda^*(X^{(m)}, Y^{(m)}), \qquad (3.34)$$

и строится оценка максимального правдоподобия в соответствии с правилом:

$$d = \begin{cases} 1, & \nu \geq 0{,}5, \\ 0, & \nu < 0{,}5. \end{cases} \qquad (3.35)$$

Вычисления на этапах 3 и 4 выполняются для всех сформированных эффективных линейных статистических аналогов.

5. Строится система линейных уравнений, причем каждое уравнение системы представляет собой равенство правой части (3.32) и соответствующего значения (3.35)

$$\sum_{k=1}^{L} c_k^* k_k = d\,. \qquad (3.36)$$

Единственное решение полученной системы (3.36) используется в качестве оценки ключа $K^* = k_1^*, k_2^*, ..., k_L^*$. Таким образом, на шаге 1 решается первая задача линейного криптоанализа, а шаги 2-5 обеспечивают решение второй задачи.

Рассмотрим пример реализации алгоритма линейного криптоанализа на базе криптосистемы S-DES, подробно рассмотренной в [4]. Криптосистема S-DES разработана для изучения свойств криптосистем, основанных на схеме Фейстеля. Алгоритм шифрования по своей структуре аналогичен алгоритму DES. Отличие состоит в следующем:

- алгоритм имеет два раунда шифрования;
- алгоритм оперирует 8 битными входными блоками;
- функция усложнения включает два блока замены;
- основной ключ шифра 10 битный, а раундовые ключи 8 битные.

Таблицы замен и перестановок приведены в [4]. Для простоты, но, не нарушая общности, будем считать, что шифрование производилось одним раундом.

В соответствии с алгоритмом линейного криптоанализа вначале проводиться анализ блоков замены (см. табл. 3.1, табл. 3.2) функции усложнения.

Таблица 3.1. Блок замены S_0

S_0	№ столбца			
№ строки	**0**	**1**	**2**	**3**
0	1	0	2	3
1	3	1	0	2
2	2	0	3	1
3	1	3	2	0

Таблица 3.2. Блок замены S_0

S_1	№ столбца			
№ строки	**0**	**1**	**2**	**3**
0	0	3	1	2
1	3	2	0	1
2	1	0	3	2
3	2	1	3	0

Результаты анализа блока S_0 и блока S_1 приведены в табл. 3.3 и 3.4.

Таблица 3.3. Анализ блока S_0

i	Значение j		
$S(i, j)$	**01**	**10**	**11**
0001	8	8	8
0010	8	6	6
0011	8	6	6
0100	6	10	8
0101	14	10	8
0110	6	8	6
0111	6	8	6
1000	8	8	8
1001	8	8	8
1010	8	10	10
1011	8	10	2
1100	6	10	8
1101	6	10	8
1110	6	4	10
1111	6	12	10

Таблица 3.4. Анализ блока S_1

i	Значение j		
$S(i, j)$	**01**	**10**	**11**
0001	8	8	8
0010	6	8	10
0011	6	12	6
0100	8	8	8
0101	8	12	12
0110	10	8	6
0111	10	8	6
1000	8	8	8
1001	8	8	8
1010	10	12	10
1011	10	8	6
1100	8	4	12
1101	8	8	8
1110	6	8	10
1111	14	8	10

Итак, в соответствии с табл. 3.3, можно выделить четыре эффективных линейных статистических аналога и составить первые четыре наиболее эффективных линейных уравнения, соответствующие парам: (i, j) – $(0101, 01)$, $(1011, 11)$, $(1110, 10)$, $(1111, 10)$. В соответствии с табл. 3.4 можно выделить семь эффективных линейных статистических аналогов и составить семь наиболее эффективных линейных уравнений, соответствующие парам: (i, j) – $(0011, 10)$, $(0101, 10)$, $(0101, 11)$, $(1010, 10)$, $(1100, 10)$, $(1100, 11)$, $(1111, 01)$. Следует отметить, что одно из уравнений будет более эффективным, однако его не достаточно для нахождения битов ключа.

Рассмотрим первую пару $(0101, 01)$ блока S_0. Очевидно, что $X_1' \oplus X_3' = Y_1'$ выполняется с вероятностью $p = 14/16 = 7/8$, а соответственно $\Delta = |1 - 2p| = |1 - 2 \cdot 7/8| = 3/4$. Тогда, в соответствии с (3.32) можно записать первое линейное уравнение для блока замены S_0:

$$X_1' \oplus X_3' \oplus Y_1' = k_1 \oplus k_3. \qquad (3.37)$$

Повторяя подобные рассуждения аналогично можно записать и другие линейные уравнения для блока S_0:

$$X_0' \oplus X_2' \oplus X_3' \oplus Y_0' \oplus Y_1' = k_0 \oplus k_2 \oplus k_3,$$
$$X_0' \oplus X_1' \oplus X_2' \oplus Y_0' = k_0 \oplus k_1 \oplus k_2, \qquad (3.38)$$

$X'_0 \oplus X'_1 \oplus X'_2 \oplus X'_3 \oplus Y'_0 = k_0 \oplus k_1 \oplus k_2 \oplus k_3$.

и блока S_1:

$$X'_6 \oplus X'_7 \oplus Y'_2 = k_6 \oplus k_7,$$
$$X'_5 \oplus X'_7 \oplus Y'_2 = k_5 \oplus k_7,$$
$$X'_5 \oplus X'_7 \oplus Y'_2 \oplus Y'_3 = k_5 \oplus k_7,$$
$$X'_4 \oplus X'_6 \oplus Y'_2 = k_4 \oplus k_6, \qquad (3.39)$$
$$X'_4 \oplus X'_5 \oplus Y'_2 = k_4 \oplus k_5,$$
$$X'_4 \oplus X'_5 \oplus Y'_2 \oplus Y'_3 = k_4 \oplus k_5,$$
$$X'_4 \oplus X'_5 \oplus X'_6 \oplus X'_7 \oplus Y'_3 = k_4 \oplus k_5 \oplus k_6 \oplus k_7.$$

Представим исходный текст, как $X0X1X2X3X4X5X6X7$. Затем выполняется начальная перестановка *IP* [4]. После перестановки получаем $X7X6X4X0X2X5X1X3$, далее текст разбивается на левую часть $X7X6X4X0$ и правую часть $X2X5X1X3$. Правая часть подвергается перестановке с расширением *E* [4]. В результате получается следующий результат - $X3X2X5X1X5X1X3X2$, который складывается с битами раундового ключа $k_0\,k_1\,k_2\,k_3\,k_4\,k_5\,k_6\,k_7$. На блок S_0 подаем $X3X2X5X1 \oplus k_0\,k_1\,k_2\,k_3$, на блок S_1 подаем $X5X1X3X2 \oplus k_4\,k_5\,k_6\,k_7$. Представим входные биты блоков замены в следующем виде. Для блока S_0 справедливы зависимости:

$$X'_0 = X3 \oplus k_0,\ X'_1 = X2 \oplus k_1,\ X'_2 = X5 \oplus k_2,\ X'_3 = X1 \oplus k_3. \qquad (3.40)$$

Для блока S_1 справедливы зависимости:

$$X'_4 = X5 \oplus k_4,\ X'_5 = X1 \oplus k_5,\ X'_6 = X3 \oplus k_6,\ X'_7 = X2 \oplus k_7. \qquad (3.41)$$

Пусть криптограмма имеет вид $Y0Y1Y2Y3Y4Y5Y6Y7$. Вид информационного блока до конечной перестановки (обратной начальной) имеет вид $Y7Y6Y4Y0Y2Y5Y1Y3$. Разбиваем данный блок на левую часть $Y7Y6Y4Y0$ и правую часть $Y2Y5Y1Y3$. Правая часть $Y2Y5Y1Y3$ получается в результате сложения по модулю 2 левой части исходного текста после перестановки *IP* $X7X6X4X0$ с текстом $X2X5X1X3$ прошедшим функцию усложнения. На выходе блоков замены функции усложнения имеем - $Y'_0\,Y'_1\,Y'_2\,Y'_3$. После перестановки *P* получаем $Y'_1\,Y'_0\,Y'_3\,Y'_2$. В результате можно записать:

$$Y'_1 = X7 \oplus Y2,\ Y'_0 = X6 \oplus Y5,\ Y'_3 = X4 \oplus Y1,\ Y'_2 = X0 \oplus Y3. \qquad (3.42)$$

Теперь подставим линейные уравнения (3.40) – (3.42) в уравнения (3.38) и (3.39) и получим уравнения линейных статистических аналогов. Полученные результаты сведем в табл. 3.5.

На этом первый этап алгоритма линейного криптоанализа закончен. Выбрав в качестве ключа шифрования $k = 0010001111$ реализуем второй, третий и четвертый этапы. Сформировав набор пар «открытый текст –

криптограмма», используя $k = 0010001111$, на основании выражения (3.33) вычислим левые части линейных эффективных аналогов.

Таблица 3.5. Линейные статистические аналоги

№ блока	Линейные уравнения	p	$\Delta = \lvert 1-2p \rvert$
S_0	$X1 \oplus X2 \oplus X7 \oplus Y2 = k_1 \oplus k_3$	7/8	3/4
	$X1 \oplus X3 \oplus X5 \oplus X6 \oplus X7 \oplus Y2 \oplus Y5 = k_0 \oplus k_2 \oplus k_3$	1/8	3/4
	$X2 \oplus X3 \oplus X5 \oplus X6 \oplus Y5 = k_0 \oplus k_1 \oplus k_2$	1/4	1/2
	$X1 \oplus X2 \oplus X3 \oplus X5 \oplus X6 \oplus Y5 = k_0 \oplus k_1 \oplus k_2 \oplus k_3$	3/4	1/2
S_1	$X0 \oplus X2 \oplus X3 \oplus Y3 = k_6 \oplus k_7$	3/4	1/2
	$X0 \oplus X1 \oplus X2 \oplus Y3 = k_5 \oplus k_7$	3/4	1/2
	$X0 \oplus X1 \oplus X2 \oplus X4 \oplus Y1 \oplus Y3 = k_5 \oplus k_7$	3/4	1/2
	$X0 \oplus X3 \oplus X5 \oplus Y3 = k_4 \oplus k_6$	3/4	1/2
	$X0 \oplus X1 \oplus X5 \oplus Y3 = k_4 \oplus k_5$	1/4	1/2
	$X0 \oplus X1 \oplus X4 \oplus X5 \oplus Y1 \oplus Y3 = k_4 \oplus k_5$	3/4	1/2
	$X1 \oplus X2 \oplus X3 \oplus X4 \oplus X5 \oplus Y1 = k_4 \oplus k_5 \oplus k_6 \oplus k_7$	7/8	3/4

В результате получаем систему уравнений (пятый этап):

$$k_0 \oplus k_2 \oplus k_3 = 0,\ \ k_0 \oplus k_1 \oplus k_2 = 0,\ k_0 \oplus k_1 \oplus k_2 \oplus k_3 = 0,\ k_6 \oplus k_7 = 1,$$
$$k_5 \oplus k_7 = 1, k_5 \oplus k_7 = 1,\ k_4 \oplus k_6 = 1,\ k_4 \oplus k_5 = 1,\ k_4 \oplus k_5 = 1, \qquad (3.43)$$
$$k_4 \oplus k_5 \oplus k_6 \oplus k_7 = 0.$$

Решение системы уравнений (3.43) имеет вид:

$k_1 = 10101001$, $k_5 = 10100101$, $k_9 = 10101010$, $k_{13} = 10100110$,
$k_2 = 01010101$, $k_6 = 01011001$, $k_{10} = 01010110$, $k_{14} = 01011010$,
$k_3 = 10011001$, $k_7 = 10010101$, $k_{11} = 10011010$, $k_{15} = 10010110$,
$k_4 = 01100101$, $k_8 = 01101001$, $k_{12} = 01100110$, $k_{16} = 01101010$.

Анализ решения системы уравнений (3.43) позволяет определить раундовый ключ $k_1 = 10101001$, а затем используя алгоритм формирования ключей алгоритма S-DES и метод полного перебора – ключ криптосистемы.

3.6 Дифференциальный (разностный) криптоанализ

Метод дифференциального (разностного) криптоанализа предложен А. Бихамом и А. Шамиром, и, по мнению ряда специалистов компании IBM, является общим методом криптоанализа блочно-итерационных криптосистем. Идея заключается в анализе процесса изменения несходства для пары открытых текстов $\Delta X = X \oplus X'$, имеющих определенные исходные различия, в процессе прохождения через циклы шифрования с одним и тем же ключом.

Метод дифференциального криптоанализа будем рассматривать на примере криптосистемы S-DES. Пусть задана пара входов X и X', с несходством $\Delta X = X \oplus X'$. Известны перестановка IP и перестановка с расширением E, а следовательно, известны и несходства ΔA на входе блоков замены S_0 и S_1. Выходы Y и Y' известны, следовательно, известно и несходство $\Delta Y = Y \oplus Y'$, а значит, при известных перестановках IP^{-1} и P известны несходства ΔC на выходе блоков замены S_0 и S_1. Доказано, что для любого заданного ΔA не все значения ΔC равновероятны. Комбинация ΔA и ΔC позволяет предположить значения битов для $E(X) \oplus k_i$ и $E(X') \oplus k_i$. То, что $E(X)$ и $E(X')$ известны, дает информацию о k_i. Несходство различных пар открытых текстов приводит к несходству получаемых криптограмм с определенной вероятностью. Эти вероятности можно определить, построив таблицы для каждого из блоков замены. Таблицы сроятся по следующему принципу: по вертикали располагаются все возможные комбинации ΔA, по горизонтали – все возможные комбинации ΔC, а на пересечении – число соответствий данного ΔC данному ΔA.

Число наибольших совпадений указывает нам пару ΔA и ΔC, с помощью которой можно определить секретный ключ. Пара открытых текстов, соответствующих данным ΔA и ΔC называется **правильной парой**, а пара открытых текстов, не соответствующих данным ΔA и ΔC – **неправильной парой**. Правильная пара подскажет правильный ключ, а неправильная пара – случайный. Чтобы найти правильный ключ, необходимо просто собрать достаточное число предположений. Один из подключей будет встречаться чаще, чем все остальные. Фактически, правильный подключ появляется из всех возможных случайных подключей.

Рассмотрим пример применения дифференциального криптоанализа на практике. Вначале анализируются блоки замены и формируются таблицы зависимостей ΔA от ΔC. Анализ осуществляется следующим образом. Так как на вход каждого блока замены подается по четыре бита, то и размерность их суммы по mod2 не будет превышать четырех бит. Таким образом, диапазон изменения ΔA лежит в пределах 0000 – 1111. Однако, пара анализируемых текстов должна различаться хотя бы одним битом, тогда значение $\Delta A = 0000$ не может использоваться для анализа. Поэтому диапазон изменения ΔA составляет 15 значений от 0001 до 1111. Каждое из значений ΔA может быть получено шестнадцатью возможными комбинациями входных данных блоков замены. Так, например, $\Delta A = 0001$ может быть получено следующими возможными комбинациями:

0000⊕0001, 0001⊕0000, 0010⊕0011, 0011⊕0010, 0100⊕0101, 0101⊕0100, 0110⊕0111, 0111⊕0110, 1000⊕1001, 1001⊕1000, 1010⊕1011, 1011⊕1010, 1100⊕1101, 1101⊕1100, 1110⊕1111, 1111⊕1110.

При этом сумма выходов по mod2, полученных после прохождения любой пары данных входов через конкретный блок замены, не всегда совпадет с суммой выходов того же блока замены по mod 2 другой пары. Рассмотрим пару входов 0011⊕0010, значение 0011 при прохождении через блок S_0 даст 01, а значение 0010 – 00. Сумма этих выходов по mod2 будет равна $\Delta C = 01 \oplus 00 = 01$. Рассмотрим другую пару входов 0101⊕1011. При прохождении через блок S_1 значение 0101 даст нам 00, а 1011 – 11. Таким образом, $\Delta C = 00 \oplus 11 = 11$. Из данного примера наглядно видно, что одному и тому же значению ΔA могут соответствовать различные ΔC. Результаты анализа блоков замены S_0 и S_1 приведен в табл. 3.6 и 3.7.

Таблица 3.6. ΔC(ΔA) в блоке S_0

ΔA	ΔC			
	00	**01**	**10**	**11**
0001	6	6	2	2
0010	0	4	8	4
0011	2	2	6	6
0100	0	12	0	4
0101	6	6	2	2
0110	0	0	8	8
0111	2	2	6	6
1000	6	6	2	2
1001	4	8	4	0
1010	2	2	6	6
1011	0	4	0	12
1100	6	6	2	2
1101	8	4	0	4
1110	2	2	6	6
1111	4	0	12	0

Таблица 3.7. ΔC(ΔA) в блоке S_1

ΔA	ΔC			
	00	**01**	**10**	**11**
0001	2	6	2	6
0010	0	8	0	8
0011	6	2	6	2
0100	0	8	4	4
0101	6	2	6	2
0110	0	0	12	4
0111	2	6	2	6
1000	2	6	2	6
1001	0	0	12	4
1010	6	2	6	2
1011	0	8	4	4
1100	6	2	6	2
1101	4	4	0	8
1110	2	6	2	6
1111	12	4	0	0

После того, как проведен анализ и построены таблицы, можно приступить к выявлению наилучшего ΔA и соответствующего ему ΔC, то есть пары (ΔA, ΔC). Из табл. 3.6 и 3.7 можно выделить несколько равновероятных пар. Для блока S_0 такими парами будут: (0100, 01), (1011, 11), (1111,10), а для блока S_1 - (0110, 10), (1001, 10), (1111, 00). Однако, следует учитывать, что ΔA равно сумме по mod2 переставленных и расширенных входных бит. Тогда можно выделить единственное значение ΔA=11111111, которому соответствует ΔC=1000. Именно эта пара и будет рассматриваться далее.

Зная наилучшее значение пары (ΔA,ΔC), можно приступить к нахождению ключа. Для этого понадобятся несколько пар открытых текстов

($X1$,$X2$), таких, что $\Delta A = E(X1) \oplus E(X2) = 11111111$, а $\Delta C = S(E(X1)) \oplus S(E(X2)) = 1000$. Для того, чтобы из зашифрованного сообщения $X1$ выделить $S(E(X1))$, необходимо к последним четырем битам шифрованного сообщения добавить первые четыре бита, а затем учесть последнюю перестановку. Для удобства работы данные тексты и относящиеся к ним данные занесем в табл. 3.8.

Таблица 3.8. Пары ($X1$, $X2$), соответствующие наилучшим (ΔA, ΔC)

№	$X1$	$E(X1)$	$S(E(X1))$	$Y1$
1	00011001	11000011	1011	01000110
2	11111001	11000011	1011	00001000
№	$X2$	$E(X2)$	$S(E(X2))$	$Y2$
1	01000110	00111100	0011	11010111
2	01000110	00111100	0011	11010111

Рассмотрим приведенные пары открытых текстов, учитывая, что результат ψ складывается по mod2 с левой частью исходного сообщения. Так как на вход блоков замены S_0 и S_1 поступают значения $E(X1)$ и $E(X2)$, то для всех $X1$ и $X2$ имеем следующее.

Для блока S_0:

1. Пара (00011001, 01000110): $1100 \oplus k_1$ даст на выходе 10; $0011 \oplus k_1$ даст на выходе 00. На выходе блока S_0 значение 10 получается в том случае, когда на его вход подается одно из значений 0100, 0101, 1000, 1101, а значение 00 – при входных 0010, 0111, 1010, 1011. Исходя из этого, имеем следующие возможные варианты:

$1100 \oplus k_1 = 0100$, $k_1 = 1000$;
$1100 \oplus k_1 = 0101$, $k_1 = 1001$;
$1100 \oplus k_1 = 1000$, $k_1 = 0100$;
$1100 \oplus k_1 = 1101$, $k_1 = 0001$;
$0011 \oplus k_1 = 0010$, $k_1 = 0001$;
$0011 \oplus k_1 = 0111$, $k_1 = 0100$;
$0011 \oplus k_1 = 1010$, $k_1 = 1001$;
$0011 \oplus k_1 = 1011$, $k_1 = 1000$.

2. Пара (11111001, 01000110). Для блока S_0 данную пару рассматривать нет необходимости, так как первые четыре бита $E(X1)$ и $E(X2)$ будут аналогичны тем же битам предыдущей пары, а, следовательно, дадут такой же результат.

Для блока S_1:

1. Пара (00011001, 01000110): $0011 \oplus k_2$ даст на выходе 11; $1100 \oplus k_2$ даст на выходе 11. На выходе блока S_1 значение 11 получается в том случае, когда на его вход подается одно из значений 0001, 0010, 1100, 1101, а значение 11 – при входных 0001, 0010, 1101. Исходя из этого, имеем следующие возможные варианты:

$0011 \oplus k_2 = 0001$, $k_2 = 0010$;
$0011 \oplus k_2 = 0010$, $k_2 = 0001$;
$0011 \oplus k_2 = 1100$, $k_2 = 1111$;
$0011 \oplus k_2 = 1101$, $k_2 = 1110$;
$1100 \oplus k_2 = 0001$, $k_2 = 1101$;
$1100 \oplus k_2 = 0010$, $k_2 = 1110$;
$1100 \oplus k_2 = 1101$, $k_2 = 0001$.

2. Пара (11111001, 01000110). Для блока S_1 данную пару рассматривать нет необходимости, так как вторые четыре бита $E(X1)$ и $E(X2)$ будут аналогичны тем же битам предыдущей пары, а следовательно дадут такой же результат.

Объединив результаты анализа, получим следующие наиболее вероятные раундовые ключи:

$k_1 = 10000001$, $k_2 = 10010001$, $k_3 = 01000001$,
$k_4 = 00010001$, $k_5 = 10001110$, $k_6 = 10011110$,
$k_7 = 01001110$, $k_8 = 00011110$.

Как показала проверка, из всех возможных комбинаций, $k_7 = 01001110$ является искомым раундовым ключом. Используя алгоритм формирования раундовых ключей алгоритма S-DES, остальные 2 бита ключа криптосистемы можно найти методом полного перебора.

4. Теория стойкости криптосистем

Систематические вопросы теоретической стойкости криптосистем впервые исследовал К.Шеннон в своей фундаментальной работе [5,10,11,13], опубликованной в 1949 г. В этой работе К. Шеннон рассматривал вероятностную модель шифра и криптоатаку на основе криптограммы. Примерно в те же годы концепция совершенных криптосистем разрабатывалась в закрытых работах, выполняемых под руководством В.А. Котельникова.

4.1. Совершенно стойкие криптосистемы

Предположим, что имеется конечное число возможных открытых сообщений $X = \{X_1, X_2, ..., X_m\}$, множество возможных ключей

$K = \{k_1, k_2, ..., k_l\}$ и множество криптограмм $Y = \{Y_1, Y_2, ..., Y_n\}$. Задано криптопреобразование:

$$Y_j = f(X_i, k_l). \quad (4.1)$$

Считаем, что на множестве открытых сообщений $X = \{X_1, X_2, ..., X_m\}$ задано априорное распределение вероятностей, т.е. определены априорные вероятности $P(X_i)$, $i = \overline{1,m}$. Это априорное распределение известно противнику. После того как шифровальщик противника перехватил некоторую криптограмму Y_j, $j = \overline{1,n}$, он может вычислить, по крайней мере в принципе, апостериорные вероятности различных сообщений $P(X_i | Y_j)$.

Криптосистема называется **совершенно стойкой** (**совершенно секретной**), если выполняется условие:

$$P(X_i | Y_j) = P(X_i), \text{ при всех } X_i, Y_j \text{ и } k_l. \quad (4.2)$$

В этом случае перехват криптограммы не дает криптоаналитику противника никакой информации. Он не может корректировать никакие свои действия в зависимости от информации, содержащейся в криптограмме, так как все вероятности, относящиеся к содержанию криптограммы, не изменяются. Смысл выражения (4.2) заключается в том, что открытый текст и криптограмма статистически независимы.

Теорема Шеннона. *Если система является совершенно стойкой (т.е. выполняется условие (4.2)), то справедливо равенство:*

$$P(Y_j | X_i) = P(Y_j), \text{ при всех } i \text{ и } j. \quad (4.3)$$

Верно и обратное утверждение: если (4.3) выполняется, то система совершенно стойкая.

□ Используя определение условной вероятности, при $P(Y_j) \neq 0$, можно записать:

$$P(X_i | Y_j) = \frac{P(X_i Y_j)}{P(Y_j)} = \frac{P(X_i) P(Y_j | X_i)}{P(Y_j)}. \quad (4.4)$$

Принимая во внимание (4.2), получаем:

$$P(X_i | Y_j) = \frac{P(X_i | Y_j) P(Y_j | X_i)}{P(Y_j)}, \text{ то есть } \frac{P(Y_j | X_i)}{P(Y_j)} = 1. \blacksquare$$

Другими словами, полная вероятность всех ключей, переводящих сообщение X_i в данную криптограмму Y_j, равна полной вероятности всех ключей, переводящих сообщение X_k в ту же самую криптограмму Y_j для всех X_i, X_k, и Y_j. К. Шеннон доказал, что совершенно стойкие криптосистемы существуют.

Теорема о совершенной стойкости шифра Вернама. *Шифр Вернама является совершенно стойкой криптосистемой.*

□ Согласно теореме Шеннона достаточно доказать справедливость (4.3). Имеем:

$$P(Y_j \mid X_i) = P(y^n \mid x^n) = P(k_1 = y_1 \oplus x_1; \ldots, k_n = y_n \oplus x_n) = \\ = P(k_1, k_2, \ldots, k_n) = 2^{-n}. \qquad (4.5)$$

В выражении (4.5) использовано предположение о равновероятности ключей. Найдем $P(Y_j)$. По формуле полной вероятности $P(Y_j) = \sum_{i=1}^{2^n} P(X_i) P(Y_j \mid X_i)$. Учитывая, что $P(Y_j \mid X_i) = 2^{-n}$, получаем:

$$P(Y_j) = 2^{-n} \sum_{i=1}^{2^n} P(X_i) = 2^{-n}, \text{ при } \sum_{i=1}^{2^n} P(X_i) = 1. \blacksquare \qquad (4.6)$$

На рис. 4.1 представлен граф совершенно стойкой криптосистемы.

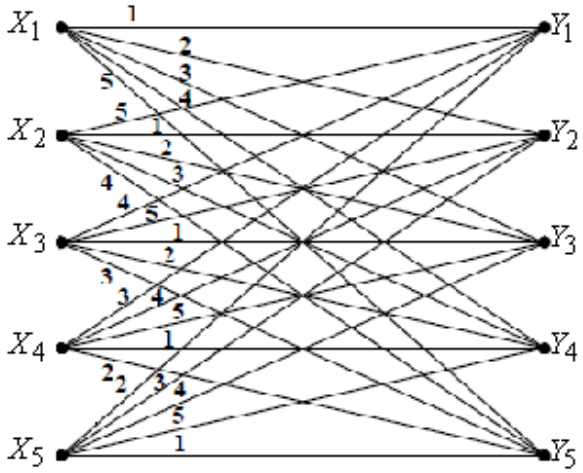

Рис. 4.1. Граф совершенно стойкой криптосистемы

Совершенно стойкие криптосистемы, характеризуются следующими свойствами: каждое открытое сообщение X_i связывается с криптограммой X_j только одной линией; все ключи равновероятны.

4.2. Идеально стойкие криптосистемы

Криптосистема включает в себя два статистических выбора: выбор сообщения и выбор ключа. Можно измерять количество информации, создаваемой при выборе сообщения, через энтропию [5,10,13]:

$$H(X) = -\sum_{X} P(X) \log P(X). \qquad (4.7)$$

Суммирование производиться по всем сообщениям. Аналогично, неопределенность, связанная с выбором ключа определяется выражением:

$$H(K) = -\sum_{K} P(K) \log P(K). \qquad (4.8)$$

В совершенно стойких криптосистемах количество информации в сообщении равно самое большое $\log n$, где n - число открытых сообщений (эта величина достигается для равновероятных сообщений). Эта информация может быть скрыта полностью лишь тогда, когда неопределенность ключа не меньше $\log n$. Это является первым примером общего принципа, который будет часто встречаться ниже: существует предел, которого нельзя превзойти при заданной неопределенности ключа – количество неопределенности, которое может быть введено в решение, не может быть больше, чем неопределенность ключа.

Предположим теперь, что, например, для английского текста используется шифр простой замены и что перехвачено определенное число, скажем N, букв зашифрованного текста. Если N достаточно велико, то почти всегда существует единственное решение задачи криптоанализа, т.е. единственная последовательность, имеющая смысл на английском языке, в которую переводится перехваченный материал с помощью простой подстановки. Для меньших N шансы на не единственность решения увеличиваются; и для определенных малых значений N, вообще говоря, будет существовать некоторое число подходящих отрывков осмысленного английского текста. При N=1, очевидно, возможна любая буква открытого текста и апостериорная вероятность любой буквы будет равна ее априорной вероятности. Для одной буквы система является совершенно стойкой.

В теории связи показано [10,13], что естественной математической мерой неопределенности того, что действительно было передано, при условии, что известен только искаженный шумом вариант – принятый сигнал, является условная энтропия передаваемого сигнала при условии, что принятый сигнал известен. Эта условная энтропия носит название ненадежности.

С криптографической точки зрения криптосистема почти тождественна системе связи при наличии шума. На сообщение действует некоторый статистический элемент (криптосистема к выбранным ключом). В результате получается криптограмма, подлежащая дешифрованию. Основное различие заключается в следующем: во-первых, в том, что преобразование при помощи шифра имеет обычно более сложную природу, чем возникающее за счет шума в канале; и, во-вторых, ключ в секретной системе обычно выбирается из конечного множества, в то время как шум в канале чаще является непрерывным, выбранным по существу из бесконечного множества.

Учитывая эти соображения, естественно использовать ненадежность в качестве теоретической меры секретности. Следует отметить, что имеются две основные ненадежности: **ненадежность ключа** и **ненадежность сообщения.**

Они будут обозначаться через $H(K|Y)$ и $H(X|Y)$, соответственно. Их величины определяются соотношениями:

$$H(K|Y) = -\sum_{K,Y} P(K,Y)\log P(K|Y), \quad (4.9)$$

$$H(X|Y) = -\sum_{X,Y} P(X,Y)\log P(X|Y). \quad (4.10)$$

где $P(K,Y)$, $P(X,Y)$ - совместные вероятности ключа и криптограммы и сообщения и криптограммы, соответственно; $P(K|Y)$, $P(X|Y)$ - апостериорные вероятности ключа и сообщения при перехваченной криптограмме.

В (4.9) и (4.10) суммирование осуществляется по всем возможным ключам и сообщениям, соответственно, а также по криптограммам определенной длины N. Таким образом, $H(K|Y)$ и $H(X|Y)$ являются функциями числа N, т.е. числа перехваченных символов криптограммы. Если ненадежность равна нулю, следует, что одно сообщение (или ключ) имеет единичную вероятность, а все другие - нулевую. Этот случай соответствует полной осведомленности криптоаналитика. Постепенное убывание ненадежности с ростом N соответствует увеличению сведений об исходном ключе или сообщении. В совершенно стойких криптосистемах для сообщений неограниченной длины требуется ключ бесконечного объема. Если использовать ключ конечного объема, то ненадежности ключа и сообщения, вообще говоря, будут стремиться к нулю, хотя это и не обязательно. На самом деле можно удерживать значение $H(K|Y)$ равным ее начальному значению $H(K)$. Тогда, независимо от того, сколько зашифрованного материала перехвачено, единственного решения не будет, а будет много решений со сравнимыми по величине вероятностями. Определим **идеально стойкую криптосистему** как такую, в которой величины $H(K|Y)$ и $H(X|Y)$ не стремятся к нулю при $N \to \infty$. **Строго идеально стойкая криптосистема** - это такая криптосистема, в которой величина $H(K|Y)$ остается равной $H(K)$. Неформально, строгая идеальность означает, что количество решений криптограммы равно количеству различных ключей и все решения равновероятны.

Примером строго идеальной криптосистемы может служить простая подстановка, примененная к искусственному языку, в котором все буквы равновероятны и последовательные буквы выбираются независимо. В этом случае $H(K|Y) = H(K)$ и $H(X|Y)$ растет линейно по прямой с наклоном $\log m$, где m - мощность алфавита, до тех пор, пока она не пересечет линию $H(K)$, после чего она остается равной этой константе. Из сказанного выше очевидно, что одной из важнейших характеристик языка является его избыточность. **Избыточность языка** – это количественная мера взаимной зависимости символов и их неравновероятности. Избыточность языка определяется как:

$$R = \log m - h, \qquad (4.11)$$

где $h = -\sum_{i=1}^{m} P(i) \log P(i)$ - энтропия на букву сообщения.

Таким образом, на основании рассмотренного можно ввести понятие расстояния единственности криптосистемы. Пусть рассматривается криптосистема и $H(K)$ - энтропия ключа. Пусть R - избыточность шифруемого сообщения, а n_p - длина сообщения, такая, что $H(K|Y) \approx 0$, т.е. при этой длине перехваченной криптограммы ключ почти однозначно восстановлен. Тогда справедливо неравенство:

$$n_p \geq \frac{H(K)}{R}. \qquad (4.12)$$

Число n_p называется **расстоянием единственности криптосистемы.** Анализ неравенства (4.12) позволяет сделать следующие выводы: для восстановления ключа с высокой вероятностью достаточно перехватить n_p символов криптограммы; если значение избыточности R=0, то ключ никогда не будет определен, так как $n_p \to \infty$; с практической точки зрения требуется менять ключ криптосистемы задолго до достижения n_p; для уменьшения R требуется преобразовывать исходный текст (например, использовать сжимающее кодирование), так как в этом случае значение R уменьшается, а энтропия преобразованного текста не изменяется. Оценку расстояния единственности криптосистемы можно использовать при ее разработке.

4.3. Практическая стойкость криптосистем

Вопрос о практической стойкости, поставленный К.Шенноном, формулируется так: «Надежна ли криптосистема, если криптоаналитик располагает ограниченным временем и ограниченными вычислительными возможностями для анализа перехваченных криптограмм?». С одной стороны, криптосистема должна обеспечивать надежную защиту информации, с другой стороны, должна быть удобна с точки зрения технической реализации и эксплуатации. Так как криптосистемы, обеспечивающие идеальную стойкость, в большинстве случаев практически неприменимы, то вопрос относиться прежде всего к криптосистемам, использующим ключи ограниченного размера и способным обрабатывать большие объемы информации.

По К.Шеннону, практически стойкая криптосистема по своим свойствам должна быть близка к идеальным криптосистемам. Например, высокая стойкость шифра гаммирования обеспечивается при использовании шифрующей последовательности, близкой по своим свойствам к равномерно распределенной случайной последовательности, поэтому криптографические

свойства шифра гаммирования определяется свойствами используемого генератора гаммы.

Системный подход к оценке стойкости криптосистемы подразумевается определенную детализацию понятия стойкости криптосистемы. В результате этой детализации формируется ряд критериев математического и технического характера, которым должна удовлетворять стойкая криптосистема.

Основной количественной мерой стойкости криптосистемы является **вычислительная сложность** решения задачи дешифрования. Вычислительная сложность определяется несколькими характеристиками. Предположим, перед криптоаналитиком поставлена задача дешифрования криптосистемы $f(\cdot)$ по набору криптограмм Y_j, $j=\overline{1,m}$. Пусть Ψ_Y - класс применимых к криптосистеме $f(\cdot)$ алгоритмов дешифрования, которыми располагает криптоаналитик. При этом криптоаналитик рассматривает как вероятностное пространство W элементарных событий множество пар ключей и открытых текстов, если открытые тексты неизвестны, или множество ключей, если открытые тексты известны. Для алгоритма $\psi \in \Psi_Y$ обозначим через $T(\psi)$ среднюю трудоемкость его реализации, измеряемую в некоторых условных вычислительных операциях. При этом величина трудоемкости обычно усредняется по множеству W.

Одной из основных характеристик практической стойкости криптосистемы $f(\cdot)$ является **средняя трудоемкость** T_Y **дешифрования**, определяемая как

$$T_Y = \min_{\psi \in \Psi_Y} T(\psi). \qquad (4.13)$$

При этом важно отметить следующее [11,13].

1. Существуют алгоритмы дешифрования, определенные не на всем вероятностном пространстве W, а лишь на некоторой его части. Кроме того, некоторые алгоритмы дешифрования устроены так, что их реализация приводит к успеху не на всей области определения, а лишь на некотором ее подмножестве. Поэтому к важнейшим характеристикам алгоритма дешифрования $\psi \in \Psi_Y$ необходимо отнести не только его трудоемкость, но и **надежность алгоритма дешифрования** $\nu(\psi)$, под которой понимается средняя доля информации, дешифруемой с помощью алгоритма ψ.

Если надежность алгоритма дешифрования мала, то с точки зрения криптографа он является неопасным, а с точки зрения криптоаналитика неэффективным. Таким образом, при получении оценки (4.13) целесообразно рассматривать лишь те алгоритмы дешифрования, надежность которых велика. При этом для определения наилучшего алгоритма дешифрования криптосистемы $f(\cdot)$ можно использовать различные критерии в зависимости от конкретных условий. Например, можно считать наилучшим алгоритм

дешифрования ψ, для которого наименьшее значение принимает величина $\frac{T(\psi)}{\nu(\psi)}$. Эту величину можно интерпретировать как **средние трудозатраты,** необходимые для успешного дешифрования криптосистемы.

2. Сложность дешифрования зависит от количественных и качественных характеристик криптограмм, которыми располагает криптоаналитик. Количественные характеристики определяются числом перехваченных криптограмм и их длинами. Качественные характеристики связаны с достоверностью перехваченных криптограмм (наличие искажений, пропусков и т.д.).

По К. Шеннону, каждая криптосистема имеет объективную характеристику $T_Y(n)$ - **среднюю вычислительную сложность дешифрования** (по всем криптограммам длины n и ключам). Величина $\lim\limits_{n\to\infty} T_Y(n)$ характеризует предельные возможности дешифрования криптосистемы при неограниченном количестве шифрматериала и абсолютной квалификации криптоаналитика. Оценивая стойкость криптосистемы, криптоаналитик получает верхние оценки предельной стойкости, так как практическое дешифрование использует ограниченное количество шифрматериала и ограниченный класс так называемых известных методов дешифрирования.

3. Важной характеристикой криптостойкости криптосистемы является **временная сложность** ее дешифрования. Оценка временной сложности дешифрования криптосистемы подразумевается более детальную проработку реализации алгоритмов дешифрования с учетом характеристик вычислительного устройства, используемого для дешифрования. К таким характеристикам вычислительного устройства, реализующего алгоритмы дешифрования, относятся архитектура, быстродействие, объем и структура памяти, быстрота доступа к памяти и др. Следовательно, время дешифрования криптосистемы определяется имеющимся классом алгоритмов дешифрования Ψ_Y и вычислительными возможностями криптоаналитика.

Выбор наилучшего алгоритма осложняется и тем, что различным вычислительным устройствам могут соответствовать различные наилучшие алгоритмы дешифрования. Вопрос о криптостойкости криптосистемы имеет некоторые особенности с точки зрения криптоаналитика и криптографа. Криптоаналитик атакует криптосистему, располагая конкретными интеллектуальными, вычислительными и экономическими ресурсами. Его цель - успешное дешифрование криптосистемы.

Криптограф оценивает стойкость криптосистемы, имитируя атаку на криптосистему со стороны криптоаналитика противника. Для этого криптограф моделирует действия криптоаналитика, оценивая по максимуму

интеллектуальные, вычислительные, технические и другие возможности противника. Цель криптографа – убедиться в высокой криптостойкости разработанной криптосистемы. Используя понятие практической криптостойкости можно классифицировать криптосистемы по величине стойкости, или по продолжительности временного периода, в течение которого криптосистема с высокой надежностью обеспечивает требуемый уровень защиты информации. Кроме рассмотренных подходов к оценке стойкости криптосистем существуют еще ряд подходов [11].

Асимптотический анализ стойкости. Этот подход развивается теорией сложности вычислений. При исследовании криптосистемы оценка его стойкости увязывается с некоторым параметром криптосистемы, обычно это длина ключа, и проводится асимптотический анализ оценок стойкости. Считается, как правило, что криптосистема имеет высокую криптостойкость, если последняя выражается через длину ключа экспоненциально, и криптосистема имеет низкую криптостойкость, если стойкость выражается в виде многочлена от длины ключа.

Оценка количества необходимого шифрматериала. Данный подход основан не на сложности вычислений при реализации дешифрования, а на оценке среднего количества материала, который необходимо проанализировать криптоаналитику для вскрытия криптосистемы. Оценка количества необходимого криптоаналитику шифрматериала представляет интерес с той точки зрения, что является нижней оценкой стойкости криптосистемы в смысле вычислительной сложности дешифрования.

Стоимостный подход. Этот подход предусматривает оценку стоимости дешифрования криптосистемы. Особенно он актуален тогда, когда для дешифрования криптосистемы необходимо разработать и построить новый вычислительный комплекс. Стоимостный подход полезен с точки зрения сопоставления материальных затрат на дешифрования криптосистемы и ценности информации, защищаемой криптосистемой.

В заключении необходимо отметить, что в связи с развитием вычислительных средств, а также прогрессом в области разработки методов дешифрования, требуется пересматривать оценки стойкости криптосистем.

4.4. Имитостойкость и помехоустойчивость криптосистем

В предыдущих пунктах рассмотрены вопросы криптостойкости криптосистем. Криптостойкость, наряду с имитостойкостью и помехоустойчивостью, является составляющей классической триады требований к криптосистемам.

Имитостойкость (imitation resistance) – свойство криптосистемы, характеризующее способность противостоять активным атакам со стороны противника, целью которых является навязывание ложного сообщения,

подмена передаваемого сообщения или изменение данных. Ложная информация считается **навязанной**, если она принята приемным устройством к исполнению. Предположим, что имеется связь между двумя абонентами А и В. Абонент А может в определенный момент времени отправить абоненту В криптограмму $y \in Y$, полученную криптосистемой $f_k(x)$, $x \in X$, на ключе $k \in K$. До момента передачи канал связи «пуст», но абонент В вводит в криптосистему ключ $k \in K$ в ожидании криптограммы от абонента А.

Возможности противника можно сформулировать в виде предположений: 1) противник знает действующий алгоритм криптопреобразования; 2) противник имеет доступ к каналу связи; 3) противник может считывать в канале любое сообщение; 4) противник может формировать и вставлять в канал связи любое сообщение; 5) противник может заменять передаваемое сообщение на любое другое; 6) все действия противник может выполнять мгновенно (противник располагает требуемыми техническими средствами); 7) противник не знает действующего ключа криптопреобразования. Рассмотрим виды имитации [1,2,11].

Имитация на пустом канале. Пусть канал связи «пуст» и противник вставляет в канал связи некоторое ложное сообщение y'. Априори возможны два исхода: абонент В примет ложное сообщение и получит $x' = f_k(y') \notin X$, т.е. абонент В сочтет полученное сообщение за ложное; абонент В примет ложное сообщение и получит $x' = f_k(y') \in X$, т.е. абонент В сочтет полученное сообщение за истинное и предпримет соответствующие действия, нанося себе ущерб.

Заметим, что при таком навязывании ложной информации противник не знает какое именно ложное сообщение абонент В примет за истинное. Такое навязывание ложной информации носит название **навязывание «наугад»**. При случайно выбираемом ключе $k \in K$ событие $x' = f_k(y') \in X$ имеет свою вероятность:

$$P_н(y') = P(x' = f_k(y') \in X), \qquad (4.14)$$

называемую **вероятностью навязывания криптограммы** y'. Противник будет выбирать навязываемые ложные криптограммы таким образом, чтобы обеспечить максимальное значение (4.14), а защищающаяся сторона должна выбрать такую криптосистему, которая обеспечила бы минимальное из всех максимальных значений (4.14), т.е.:

$$P_н^{\text{б}}(y') = \min_{f(\cdot)} \max_{y'} P(x' = f_k(y') \in X). \qquad (4.15)$$

Величина обратная вероятности навязывания криптограммы $\frac{1}{P_н^{\text{б}}(y')}$ носит название **коэффициента имитозащиты в пустом канале**. При навязывании

противником наилучших криптограмм величина $\frac{1}{P_н(y')}$ характеризует среднее число попыток противника до навязывания ложной информации. Противник может подбирать криптограммы y' таким образом, что наносить абоненту В максимальный ущерб. Такое навязывание называется **прицельным.**

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

Имитация при знании открытого текста. Противник может обладать дополнительной информацией, например, может знать открытый текст $x \in X$ который скрывается за криптограммой y в канале связи. В этом случае аналогично может быть определена **вероятность навязывания ложной информации при знании открытого текста:**

$$P_н(y' \mid x) = P(x' = f_k(y') \in X \mid x). \qquad (4.16)$$

Аналогично (4.15) может быть определено значение:

$$P_н(y' \mid x \in X) = \min_{f} \max_{y'} P(x' = f_k(y') \in X \mid x \in X). \qquad (4.15)$$

В широком смысле к имитации относятся и другие действия противника. Во-первых, противник может переадресовать сообщение y адресованное В, абоненту С. Во-вторых, противник может изменить подпись абонента А в сообщении. В-третьих, противник, перехватив сообщение y от абонента А может задержать его и вставить в канал связи в другое время. Для определения факта навязывания информации предусмотрена **имитозащита.**

Рассмотрим способы имитозащиты.

1. Использование меток времени в исходном сообщении $x \in X$. В случае если метки времени в принятом сообщении y превышают некоторый допустимый предел, то делается вывод о принятии ложного сообщения.

2. Изменение ключа шифра через заданные (не всегда постоянные) интервалы времени. Если задержка в приеме сообщения превышает величину этого интервала, то навязанное сообщение не расшифруется.

3. Введение в сообщение дополнительной избыточности, например, кодирования.

4. Использование **имитовставок** или **кодов аутентификации**. К передаваемому сообщению добавляется отрезок информации фиксированной длины, вычисленной по определенному правилу на основе данных и ключа. Обеспечение имитозащиты таким способом предусмотрено в криптосистеме ГОСТ 28147-89.

Помехоустойчивость криптосистемы (noise stability of a cipher) - способность криптосистемы противостоять действию случайных помех (в отличие от целенаправленных действий противника), возникающих при

передаче шифрованного сообщения по каналу связи. Искажение сообщения y при его передаче по каналу связи приводит к замене его на другое $y^* \in Y^*$, причем $Y \le Y^*$. Характер замены $y \in Y$ на $y^* \in Y^*$ определяется физическим состоянием как самого канала связи, так и окружающей среды.

Примером искажений являются:

1) искажения типа замены, при этом передаваемое по каналу связи сообщение $y = b_1, b_2, ..., b_L$ заменяется на $y^* = b_1^*, b_2^*, ..., b_L^*$ (искажению может быть подвигнута любая буква сообщения b_i, $i = \overline{1, L}$);

2) искажения типа вставки или пропуска, при этом передаваемое по каналу связи сообщение $y = b_1, b_2, ..., b_L$ заменяется или на $y^* = b_1, b_2, b_2^*, ..., b_L$ (длина сообщения при этом составляет $L+1$), или на $y^* = b_1, b_3, ..., b_L$ (длина сообщения при этом составляет $L-1$);

Для указанных искажений может оказаться, что при некоторых $f($ может не существовать $f_k^{-1}(y^*)$, т.е. на приемной стороне y^* не будет расшифровываться.

Подробно вопрос помехоустойчивости криптосистем рассмотрен в [1], где определены криптосистемы не размножающие искажений типа замены, пропуска или вставки букв, а также рассмотрены методы обеспечения помехоустойчивости криптосистем.

Литература

1. Бабаш А.В., Шанкин Г.П. Криптгграфия. /Под редакцией В.П. Шерстюка, Э.А. Применко. – М.: СОЛОН-ПРЕСС, 2007.
2. Баричев С.Г, Гончаров В.В., Серов Р.Е. Основы современной криптографии: Учебный курс. – М.: Горячая линия-Телеком, 2002.
3. Болелов Э.А. Криптографические методы защиты информации: Пособие по выполнению практических занятий. – М.: МГТУ ГА, 2010.
4. Болелов Э.А. Криптографические методы защиты информации: Пособие по выполнению лабораторных работ. – М.: МГТУ ГА, 2010.
5. Зубов А.Ю. Криптографические методы защиты информации. Совершенные шифры: Учебное пособие. – М.: Гелиос АРВ, 2005.
6. Нечаев В.И. Элементы криптографии (Основы теории защиты информации): Учеб. пособие для вузов / Под ред. В.А. Садовничего. – М.: Высш. шк., 1999.
7. Плотников А.Д. Дискретная математика: учеб. пособие. – Мн: Новое знание, 2008.
8. Рябко Б.Я., Фионов А.Н. Криптографические методы защиты информации: Учебное пособие для вузов. – М.: Горячая линия-Телеком, 2005.
9. Танова Э.В. Введение в криптографию: как защитить сое письмо от любопытных. Элективный курс: учебное пособие. – М.: БИНОМ. Лаборатория знаний, 2007.
10. Харин Ю.С. и др. Математические и компьютерные основы криптологии: Учеб. пособие. – Мн.: Новое знание, 2003.
11. Фомичев В.М. Дискретная математика и криптология. Курс лекций / Под ред. Н.Д. Подуфалова. – М.: ДИАЛОГ-МИФИ, 2003.
12. Ассоциация историков спецслужб им. А.Х.Артузова. http://www.agentura.ru/dossier/russia/fapsi/stor/.
13. Шеннон К. Теория связи в секретных системах. – М.: ИЛ, 1963.
14. ГОСТ 28147-89. Система обработки информации. Защита криптографическая. Алгоритм криптографического преобразования данных.