МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

САНКТ–ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Н.А. Гатченко, А.С. Исаев, А.Д. Яковлев

Криптографическая защита информации

Учебное пособие

Санкт–Петербург

2012

УДК 003.26
ББК 32.973.26-018 Я73

Гатченко Н.А., Исаев А.С., Яковлев А.Д. «Криптографическая защита информации» – СПб: НИУ ИТМО, 2012. – 142 с.

В учебном пособии изложены и проанализированы основные понятия криптографической защиты информации как в Российской Федерации, так и за рубежом. Рассмотрены понятия блочных шифров, поточных шифров, истории развития криптографии и системы с открытым ключом.

Рекомендовано студентам, аспирантам и специалистам по специальностям: 090103 «Организация и технология защиты информации», 090900 – «Информационная безопасность», которые по роду своей деятельности непосредственно сталкиваются с криптографической защитой информации.

Рекомендовано к печати Ученым советом Института комплексного военного образования СПб НИУ ИТМО протокол №1 от 30 января 2012 г. в качестве учебного пособия для специалистов кафедры мониторинга и прогнозирования информационных угроз.

В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена программа его развития на 2009–2018 годы. В 2011 году Университет получил наименование «Санкт–Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

Предисловие

При современном темпе развития компьютерных и цифровых технологий мы не в состоянии воспринимать свою жизнь вне информационного потока окружающего нас. Процессы обработки, хранения, передачи и использования информации становятся главенствующими в жизни современного общества, любая наша деятельность достаточно тесно связана с этими процессами.

В условиях всеобщей информатизации, вопросы информационной безопасности и защиты информации становятся наиболее актуальными. Наука о тайной передачи информации, недоступной или непонятной для посторонних лиц, произошло и стало развиваться в тот момент, когда человечество осознало необходимость обеспечения защиты информации. Криптография – одна из старейших наук, ее история насчитывает несколько тысяч лет, развиваясь вместе с человеком, она претерпела огромное количество изменений, постоянно совершенствуясь и дополняясь. Криптографическая защита информации является одной из основных подсистем любой системы защиты информации (СЗИ). Все это обусловило необходимость издания предлагаемого вниманию читателей учебного пособия «Криптографическая защита информации».

В учебном пособии по специальностям 090103, 090900, кроме общих и специальных вопросов в области криптографической защиты информации, особое внимание уделено математическому обоснованию, основным направлениям деятельности в современной криптографии, практическому применению полученных знаний, а также опытному применению основополагающих шифров.

Учебное пособие подготовлено в соавторстве трех авторов: Гатченко Н.А., Исаева А.С., Яковлева А.Д., под научной редакцией профессора кафедры «Мониторинга и прогнозирования информационных угроз» кандидата технических наук, доцента Жигулина Г.П.

Введение

Использование криптографии в современных цифровых технологиях становится неотъемлемой частью многих сфер жизни нашего общества. Этот процесс становится все более и более масштабным.

Все чаще в нашей повседневной жизни встречаются такие понятия, как логин и пароль, аутентификация и идентификация, электронная цифровая подпись, шифрование открытым и закрытым ключом, и многие другие.

В учебном пособии рассмотрены общие вопросы обеспечения криптографической защиты информации, история и развитие криптографии, основные алгоритмы и их математическое обоснование.

В доступной форме изложены основные сведения об основных направлениях криптографии, наиболее актуальных новшествах и примерах классической криптографии.

Последовательность изложения обеспечивает наиболее полное и комплексное восприятие предмета для студентов.

Учебное пособие предназначено для подготовки студентов по специальностям: 090103 – «Организация и технология защиты информации», 090900 – «Информационная безопасность».

ГЛАВА 1

ИСТОРИЯ И ОСНОВНЫЕ ПОЛОЖЕНИЯ КРИПТОГРАФИИ

1.1 Криптографические средства с древнего времени

Понятие «безопасность» охватывает широкий круг интересов, как отдельных лиц, так и целых государств. Во все исторические времена существенное внимание уделялось проблеме информационной безопасности, обеспечению защиты конфиденциальной информации от ознакомления с ней конкурирующих групп. Недаром великий психолог Вильям Шекспир в «Короле Лире» изрёк: «Чтоб мысль врага узнать, сердца вскрывают, а не то, что письма».

Существовали три основных способа защиты информации, Первый способ предполагал чисто силовые методы: охрана документа (носителя информации) физическими лицами, его передача специальным курьером и т. д.

Второй способ получил название «стеганография» и заключался в сокрытии самого факта наличия секретной информации. В этом случае, в частности, использовались так называемые «симпатические чернила». При соответствующем проявлении, текст становился видимым. Один из оригинальных примеров сокрытия информации приведён в трудах древнегреческого историка Геродота. На голове раба, которая брилась наголо, записывалось нужное сообщение. И когда волосы его достаточно отрастали, раба отправляли к адресату, который снова брил его голову и считывал полученное сообщение. Идея экзотической защиты секретных текстов (в том числе и с применением симпатических чернил) дошла до наших дней. А. Толстой в известном произведении «Гиперболоид инженера Гарина» описал способ передачи сообщения путем его записи на спине посыльного – мальчика. Во время II Мировой войны таким же образом иногда передавались агентурные сообщения. Секретные послания записывались симпатическими чернилами и на предметах нижнего белья, носовых платках, галстуках и т. д.

Третий способ защиты информации заключался в преобразовании смыслового текста в некий хаотический набор знаков (букв алфавита). Получатель донесения имел возможность преобразовать его в исходное осмысленное сообщение, если обладал «ключом» к его построению. Этот способ защиты информации называется криптографическим. По утверждению ряда специалистов, криптография по возрасту – ровесник египетских пирамид.

В документах древних цивилизаций – Индии, Египта, Месопотамии – есть сведения о системах и способах составления шифрованных писем.

Контрольные вопросы:

1) **Приведите примеры известных вам древних шифров.**
2) **Приведите примеры известных вам способов сокрытия информации.**
3) **Назовите основные способы сокрытия информации в древнем мире.**

1.2 Шифр Гая Юлия Цезаря

В криптографии древних времен использовались два вида шифров: замена и перестановка. Наиболее древним и распространённым примером шифра замены является шифр Цезаря. Гай Юлий Цезарь использовал в своей переписке шифр собственного изобретения. Суть шифра Цезаря состоит в том, что буквы алфавита заменяются буквами того же алфавита, но со сдвигом в право на 3 позиции. Пример ключа применительно к русскому языку будет выглядеть так:

Открытый текст:

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

Зашифрованный:

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

При зашифровывании буква А заменялась буквой Г, Б заменялась на Д, В – Е и так далее. Так, например, «РИМ» превращалось в слово «УЛП». Получатель сообщения «УЛП» искал эти буквы в нижней строке и по буквам над ними восстанавливал исходное слово «РИМ». Ключом в шифре Цезаря является величина сдвига 2-ой нижней строки алфавита, в нашем случае это число 3. Преемник Юлия Цезаря – Цезарь Август использовал тот же шифр, но с ключом – сдвиг 1. Слово «РИМ» в этом случае зашифровывается, как «СЙН».

Развитие шифра Цезаря весьма очевидно, так как в нижней строке двухстрочной записи буквы алфавита могут быть расположены в произвольном порядке. Применительно к русскому языку в нижней строке существует всего 33 варианта ключей (число букв в русском алфавите), при их произвольном расположении число ключей становится огромным. Оно равно 33! (33 факториал), т. е. приблизительно десять в тридцать пятой степени. Это важный момент. Если противник узнал или догадался об используемом шифре, то он может попытаться расшифровать текст путём полного перебора ключей. Даже в современных условиях это занимает большое количество времени и ресурсов, что уж говорить о тех древних временах, когда из-за повсеместной

неграмотности населения, не представлялась возможность за зашифрованным текстом увидеть свой язык. Многие люди, видя шифр, считали, что запись сделана на иностранном языке.

В художественной литературе классическим примером шифра замены является известный шифр «Пляшущие человечки» К. Дойля. В нём буквы заменялись на символические фигурки людей. Ключом такого шифра являлись позы человечков, заменяющих буквы. Фрагмент шифрованного послания представлен на рисунке 1.2.1.

Рисунок 1.2.1 – Фрагмент шифрованного послания.

Для получения соответствующего закрытого текста использован шифр простой замены букв на фигурки людей; флажок в руках означает конец слова, таким образом, эти пляшущие человечки с флажками могли шифровать огромный набор данных. Криптографическая стойкость такого шифра даже в те времена была не особо высокой, ведь пользуясь статистическими знаниями при написании слов, мы можем с легкостью отличить наиболее часто повторяющиеся гласные буквы, что существенно облегчит перебор возможных вариантов.

Контрольные вопросы:

1) **Что такое шифр Цезаря?**
2) **В чем суть работы шифра Цезаря?**
3) **Какое логическое продолжение в работе шифра Цезаря?**
4) **Как выбрать ключ для шифра Цезаря?**
5) **Зашифруйте шифром Цезаря слово «Криптография» использую ключ + 5.**
6) **Зашифруйте шифром Цезаря слово «Университет» использую ключ – 3.**

1.3 Шифр перестановки

При шифровании перестановкой символы шифруемого текста переставляются по определенному правилу в пределах блока этого текста. Шифры перестановки являются самыми простыми и, вероятно, самыми древними шифрами.

Выберем целое положительное число, к примеру, 5; расположим числа от 1 до 5 в верхней строке по порядку, а в нижней строке расположим эти числа в произвольном порядке, так как представлено в таблице 1.3.1.

1	2	3	4	5
2	5	1	4	3

Таблица 1.3.1 – Конструкция подстановки.

Эта конструкция носит название подстановки, а число 5 называется её степенью. Зашифруем фразу «НАУЧНОЕ ОБЩЕСТВО СТУДЕНТОВ». В этой фразе 24 буквы, необходимо дополнить фразу до ближайшего кратного 5 числа. В нашем случае добавляем одну произвольную букву (например, Ы) и получаем число 25. Теперь разбиваем фразу на пятизначные группы:

НАУЧН ОЕОБЩ ЕСТВО СТУДЕ НТОВЫ

Далее буквы каждой группы перестанавливаем в соответствии с нашей таблицей, по следующему правилу: первая буква встаёт на второе место, вторая – на пятое, третья – на первое, четвёртая – на четвёртое, пятая – на третье. Полученный текст выписывается без пропусков:

УННЧАООЩБЕТЕОВСУСЕДТОНЬВТ

При расшифровывании текст также разбивается на пятизначные группы и буквы каждой группы перестанавливаются в обратном порядке: первая буква на третье место, вторая – на первое, третья – на пятое, четвёртая – на четвёртое, пятая – на второе. Ключом шифра является число 5 и порядок расположения чисел в нижнем ряду таблицы.

Контрольные вопросы:

1) **Что такое шифр перестановки?**
2) **На чем основан шифр перестановки?**
3) **Зашифруйте фразу «Я люблю криптографию» шифром перестановки, используя произвольную степень.**

1.4 Шифр перестановки «сцитала»

Известно, что в V веке до нашей эры правители Спарты, наиболее воинственного из греческих государств, имели хорошо отработанную систему секретной военной связи и шифровали свои послания с помощью *сцитала*, первого простейшего криптографического устройства, реализующего метод простой перестановки.

Шифрование выполнялось следующим образом. На стержень цилиндрической формы, который назывался *сцитала*, наматывали спиралью (виток к витку) полоску пергамента и писали на ней вдоль стержня несколько строк текста сообщения, изображенные на рисунке 1.4.1. Затем снимали со стержня полоску пергамента с написанным текстом. Буквы на этой полоске оказывались расположенными хаотично. Такой же результат можно получить, если буквы сообщения писать по кольцу не подряд, а через определенное число позиций до тех пор, пока не будет исчерпан весь текст.

Рисунок 1.4.1 – Шифр «Сцитала».

Сообщение НАСТУПАЙТЕ, при размещении его по окружности стержня по три буквы дает шифротекст НУТАПЕСА_ТЙ. Для расшифровывания такого шифротекста нужно не только знать правило шифрования, но и обладать ключом в виде стержня определенного диаметра. Зная только вид шифра, но, не имея ключа, расшифровать сообщение было непросто. Шифр сцитала многократно совершенствовался в последующие времена. Одним из самых первых шифровальных приспособлений был жезл, применявшийся еще во времена войны Спарты против Афин в V веке до н. э. Это был цилиндр, на который виток к витку наматывалась узкая папирусная лента (без просветов и нахлестов), а затем на этой ленте вдоль его оси записывался необходимый для передачи текст. Лента сматывалась с цилиндра и отправлялась адресату, который, имея цилиндр точно такого же диаметра, наматывал ленту на него и прочитывал сообщение. Ясно, что такой способ шифрования осуществляет перестановку местами букв сообщения.

Шифр «Сцитала», как видно из решения задачи, реализует не более nперестановок, где n длина сообщения. Действительно, этот шифр, как нетрудно видеть, эквивалентен следующему шифру маршрутной перестановки:

в таблицу, состоящую из m столбцов, построчно записывают сообщение, после чего выписывают буквы по столбцам. Число задействованных столбцов таблицы не может превосходить длины сообщения. Имеются еще и чисто физические ограничения, накладываемые реализацией шифра «Сцитала». Естественно предположить, что диаметр жезла не должен превосходить 10 сантиметров. При высоте строки в 1 сантиметр на одном витке такого жезла уместится не более 32 букв ($10\ \pi < 32$). Таким образом, число перестановок, реализуемых этим шифром, вряд ли превосходит 32. Изобретение дешифровального устройства – «антисцитала» – приписывается Аристотелю. Он предложил использовать конусообразные «копьё», на которое наматывался перехваченный ремень; этот ремень передвигался по оси до того положения, пока не появлялся осмысленный текст.

Контрольные вопросы:

1) **Что такое шифр сцитала?**
2) **На чем основывается шифр сцитала?**
3) **Какой максимальный диаметр жезла мог быть использован для шифра сцитала?**
4) **Каким образом предполагается расшифровывать шифр сцитала?**

1.5 Диск Энея

Древнегреческий полководец Эней Тактика в IV веке до н.э. предложил устройство, названное впоследствии «диском Энея». Принцип его был достаточно прост. На диске диаметром 10-15 см и толщиной 1-2 см высверливались отверстия по числу букв алфавита. В центре диска помещалась «катушка» с намотанной на ней ниткой достаточной длины. При зашифровывании нитка «вытягивалась» с катушки и последовательно протягивалась через отверстия, в соответствии с буквами шифруемого текста. Диск и являлся посланием. Получатель послания последовательно вытягивал нитку из отверстий, что позволяло ему получать передаваемое сообщение, но в обратном порядке следования букв. При перехвате диска недоброжелатель имел возможность прочитать сообщение тем же образом, что и получатель. Но Эней предусмотрел возможность легкого уничтожения передаваемого сообщения при угрозе захвата диска. Для этого было достаточно выдернуть «катушку» с закрепленным на ней концом нити до полного выхода всей нити из всех отверстий диска.

Идея Энея была использована в создании и других оригинальных шифров замены. Скажем, в одном из вариантов вместо диска использовалась линейка с числом отверстий, равных количеству букв алфавита. Каждое отверстие

обозначалось своей буквой; буквы по отверстиям располагались в произвольном порядке. К линейке была прикреплена катушка с намотанной на нее ниткой. Рядом с катушкой имелась прорезь. При шифровании нить протягивалась через прорезь, а затем через отверстие, соответствующее первой букве шифруемого текста, при этом на нити завязывался узелок в месте прохождения ее через отверстие; затем нить возвращалась в прорезь и аналогично зашифровывалась вторая буква текста и т.д. После окончания шифрования нить извлекалась и передавалась получателю сообщения. Тот, имея идентичную линейку, протягивал нить через прорезь до отверстий, определяемых узлами, и восстанавливал исходный текст по буквам отверстий. Это устройство получило название «линейка Энея».

Аналогичное «линейке Энея» «узелковое письмо» получило распространение у индейцев Центральной Америки. Свои сообщения они также передавали в виде нитки, на которой завязывались разноцветные узелки, определявшие содержание сообщения.

Заметным вкладом Энея в криптографию является предложенный им, так называемый книжный шифр, описанный в сочинении «Об обороне укрепленных мест». Эней предложил прокалывать малозаметные дырки в книге или в другом документе над буквами секретного сообщения. Интересно отметить, что в первой мировой войне германские шпионы использовали аналогичный шифр, заменив дырки точками, наносимые симпатическими чернилами на буквы газетного текста.

Книжный шифр в современном его виде имеет несколько иной вид. Суть этого шифра состоит в замене букв на номер строки и номер этой буквы в строке в заранее оговоренной странице некоторой книги. Ключом такого шифра является книга и используемая страница в ней. Этот шифр оказался «долгожителем» и применялся даже во время второй мировой войны.

Контрольные вопросы:

1) **Что такое диск Энея?**
2) **На чем основывается принцип работы диска Энея?**
3) **Что такое узелковое письмо?**
4) **Каким образом диск Энея защищался от перехвата?**
5) **Что такое книжный шифр и каков принцип его работы?**

1.6 Квадрат Полибия

Квадрат Полибия появился в древней Греции во втором веке до нашей эры. Этот метод шифрования представляет собой, квадрат, разделённый на 5*5 клеток (применительно к латинскому алфавиту), в каждую клетку вписываются все буквы алфавита, при этом буквы I, J не различаются (J=I), как это представлено в таблице 1.6.1.

	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

Таблица 1.6.1 – Квадрат Полибия.

Каждая шифруемая буква заменялась на координаты квадрата, в котором она записана. Так, A заменялась на AA, Q на AD и т.д. При расшифровывании каждая такая пара букв определяла соответствующую букву сообщения. Интересно, что в данном способе шифрования ключ отсутствует, так как используется фиксированный алфавитный порядок следования букв, а секретом является сам способ замены букв. Усложненный вариант шифра квадрат Полибия заключается в том, что запись букв в квадрат производится в произвольном порядке и этот произвольный порядок является ключом.

Но произвольный порядок сложно запомнить, поэтому пользователю шифра необходимо было постоянно иметь при себе ключ, то есть квадрат. Что приводило к возможности несанкционированного ознакомления с ключом посторонних лиц. Чтобы решить эту проблему был предложен ключ – пароль. Легко запоминаемый пароль выписывался без повторов букв в квадрат; в оставшиеся клетки, в алфавитном порядке, выписывались буквы алфавита, отсутствующие в пароле. Например, пусть паролем является слово «THEWINDOW». Тогда квадрат иметь вид, изображенный в таблице 1.6.2 .

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

Таблица 1.6.2 – Усложненный квадрат Полибия.

Такой квадрат уже не нужно иметь при себе, достаточно запомнить ключ – пароль. В 19 веке был предложен ещё более усложненный вариант квадрата Полибия. Этот вариант шифра имеет вид, изображенный в таблице 1.6.3.

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

Таблица 1.6.3 – Более усложненный квадрат Полибия.

Зашифруем слово «THEWINDOW». Получим зашифрованный текст:

13.21.11.45.22.55.24.51.45.

На этом историческое шифрование по Полибию заканчивалось. Это был шифр простой замены типа шифр Цезаря, в котором каждая буква открытого текста заменялась некоторым двухзначным числом, и эта замена не менялась по всему тексту. Количество ключей этого шифра равно 25!. Усложненный вариант заключался в следующем. Полученный первичный шифротекст (*) шифруется вторично. При этом он выписывается без разбиения на пары:

132111452255245145 (**)

Полученная последовательность цифр сдвигается циклически влево на один шаг: 321114522552451451.Эта последовательность вновь разбивается в группы на два: 32.11.14.52.25.52.45.14.51. И по таблице заменяется на окончательный шифротекст: SELYUYWLO. Количество ключей в этом шифре останется тем же, но он уже значительно более стоек. Заметим, что этот шифр уже не является шифром простой замены. Был подмечен и негативный момент. Если в шифре простой замены шифротекст будет написан с одной ошибкой, например, в тексте вместо седьмой буквы 24 будет написано 42, то расшифрованный текст будет содержать лишь одну ошибку: THEWINPOW, что легко исправляется получателем сообщения. Но если, же в тексте будет искажена седьмая буква (буква W заменена, например, на A), то в расшифрованном тексте будет уже два искажения: THEWINIHW, что уже затрудняет восстановление исходного сообщения.

Контрольные вопросы:

1) **Что такое квадрат Полибия?**
2) **Каков принцип работы квадрата Полибия?**

3) **Опишите усложненный квадрат Полибия и его принцип работы.**

4) **Опишите более усложненный квадрат Полибия и его принцип работы.**

5) **Зашифруйте слово «Криптография» усложненным и простым квадратом Полибия.**

6) **К чему приведет нарушение в порядке записи зашифрованного шифротекста?**

1.7 Шифр Чейза

В середине 19 века американец П.Э.Чейз предложил следующую модификацию шифра Полибия. Выписывается прямоугольник размера 3×10; буквы латинского алфавита дополняются знаком @ и греческими буквами λ, ω, φ. Пример изображен на рисунке 1.7.1 .Ключом шифра является порядок расположения букв в таблице. При шифровании координаты букв выписываются вертикально. Например, слово ALICE приобретает вид: 11312 38946 , здесь первый ряд определяет строку, а второй ряд чисел определяет столбец.

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

Рисунок 1.7.1 – Таблица шифра Чейза.

Чейз предложил ввести ещё один ключ: заранее оговоренное правило преобразования нижнего ряда цифр. Например, число, определяемое этим рядом, умножается на 9:Получаем новую двухстороннюю запись: 11312 350514. Эта двухстрочная запись вновь переводится в буквы согласно таблице; при этом первое число нижнего ряда определяет букву первой строки. Шифротекст приобретает следующий вид: AOφHXM. Могут быть использованы и другие преобразования координат. Этот шифр значительно сильнее шифра Полибия, он уже не является шифром простой замены. При расшифровывании полученная последовательность переводится в двухстрочную запись: 11312 350514, второй ряд разделим на 9, получим 38946. Образуется двухстрочная запись (1) и по ней согласно таблице читается открытый текст.

Однако предложение Чейза не нашло поддержки. Причины: заметное усложнение процесса шифрования – расшифровывания, а также особая чувствительность шифра к ошибкам. В этот вы сможете убедиться самостоятельно. Кроме того, использование в качестве ключа любого числа (кроме 9) может порождать недоразумения, как при шифровании, так и при расшифровывании.

Контрольные вопросы:

1) **Что такое шифр Чейза?**
2) **Опишите принцип работы шифра Чейза.**
3) **Зашифруйте шифром Чейза слово «Криптография».**
4) **Зашифруйте усложненным шифром Чейза слово «Криптография».**
5) **Почему использование любого другого числа кроме 9, влечет за собой нестыковки при шифровании?**

1.8 Тюремный шифр

Интересно отметить, что в несколько измененном виде шифр Полибия дошёл до наших дней и получил своеобразное название «тюремный шифр». Для его использования нужно только знать естественный порядок расположения букв алфавита (как в указанном выше примере квадрат Полибия для английского языка). Стороны квадрата обозначаются не буквами (ABCDE), а числами (12345). Число 3, например, передаётся путем тройного стука. При передаче буквы сначала «отстукивается» число, соответствующее строке, в которой находится буква, а затем номер соответствующего столбца. Например, буква «F» передается двойным стуком (вторая строка) и затем одинарным (первый столбец).

С применением этого шифра связаны некоторые исторические казусы. Так, декабристы, посаженные в тюрьму после неудавшегося восстания, не смогли установить связь с находившимся в «одиночке» князем Одоевским. Оказалось, что князь (хорошо образованный по тем временам человек) не помнил естественного порядка расположения букв в русском и французском алфавитах (другими языками он не владел) декабристы для русского алфавита использовали прямоугольник размера 5×6 (5строк и 6 столбцов) и редуцированный до 30 букв алфавит.

Тюремный шифр, строго говоря, не шифр, а способ перекодирования сообщения с целью его приведения к виду, удобному для передач по каналу связи (через стену). Дело в том, что в таблице использовался естественный

порядок расположения букв алфавита. Так что секретом является сам шифр (а не ключ), как у Полибия.

Контрольные вопросы:

1) **Что такое тюремный шифр и в чем принцип его работы?**
2) **В чем отличие тюремного шифра от шифра Полибия?**
3) **Какой порядок букв используется в тюремном шифре?**
4) **Зашифруйте тюремным шифром слово «Криптография».**

1.9 Магические квадраты

Во времена средневековья европейская криптография приобрела сомнительную славу, отголоски которой слышатся и в наши дни. Криптографию стали отождествлять с черной магией, с некоторой формой оккультизма, астрологией, алхимией, еврейской каббалой. К шифрованию информации призывались мистические силы. Так, например, рекомендовалось использовать «магические квадраты».

В квадрат размером 4×4 (размеры могли быть и другими) вписывались числа от 1 до 16. его магия состояла в том, что сумма чисел по строкам, столбцам и полным диагоналям равнялась одному и тому же значению числу – 34. Впервые эти квадраты появились в Китае, где им и была приписана некоторая «магическая сила». Пример изображен в таблице 1.9.1.

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

Таблица 1.9.1 – Пример магического квадрата.

Шифрование по магическому квадрату производилось следующим образом. Например, требуется зашифровать фразу: «приезжаю сегодня». Буквы этой фразы вписываются последовательно в квадрат согласно записанным в них числах, а в пустые клетки ставятся произвольные буквы. Пример приведен в таблице 1.9.2.

16У	3И	2Р	13Д

5З	10Е	11Г	8Ю
9С	6Ж	7А	12О
4Е	15Я	14Н	1П

Таблица 1.9.2 – Пример шифрования магическим квадратом.

После этого шифрованный текст записывается в строку:

УИРДЗЕГЮСЖАОЕЯНП

При расшифровывании текста вписывается в квадрат, и открытый текст читается в последовательности чисел «магического квадрата». Данный шифр обычный шифр перестановки, но считалось, что особую стойкость ему придает волшебство «магического квадрата».

Контрольные вопросы:

1) **Что такое шифр магического квадрата, и каков принцип его действия?**
2) **Зашифруйте шифром магического квадрата слово «Криптография».**
3) **Зашифруйте шифром магического квадрата слово «Аккумулятор».**
4) **Что произойдет если изменить число сторон в квадрате?**

1.10 Шифр Аве Мария

В Испании первые системы шифрования – преобразование открытого текста в римские цифры – появились в XV в. Христофор Колумб, будучи в Новом Свете, использовал такой шифр в письме к брату, где предлагал выгнать присланного из Испании губернатора. Письмо было перехвачено, дешифровано, и в Испанию отправился сам Колумб – в кандалах и под стражей.

В Германии XV-XVI вв. значительный вклад в криптографию внес Иоганнес Тритемий, аббат монастыря Шпонхейм. Это был образованный по меркам своего времени человек, находившийся под личным покровительством императора Максимилиана I. Ученый богослов, натурализатор и историк, он заслужил славу чернокнижника своим сочинением «Полиграфия» – в те времена так называлась тайнопись, – написанным в 1499 году. Это была первая печатная книга по криптографии.

Богатым на новые идеи в криптографии оказался XVI век. Многоалфавитные шифры получили развитие в вышедшей в 1518 г. первой

печатной книге по криптографии под названием «Полиграфия». Автором книги был один из самых знаменитых ученых того времени аббат Иоганнес Тритемий. В этой книге впервые в криптографии появляется квадратная таблица. Шифралфавиты записаны в строки таблицы один под другим, причем каждый из них сдвинут на одну позицию влево по сравнению с предыдущим. В XV-XVI вв. немецкий аббат Иоганнес Тритемий сделал значительный вклад, в криптографию предложив два новаторских способа шифрования. Это шифр «Аве Мария» и шифр, построенный на основе периодически сдвигаемого ключа. Шифр «Аве Мария» основан на принципе замены букв шифруемого текста на заранее оговоренные слова. Из этих сообщений составлялось внешне «невинное» сообщение. Заменим буквы Е, Н, Т на следующие слова:

Е = «ЗЕЛЁНЫЙ», «ЖДУ», «МОЙ»; Н = «И», «Я», «ЗДЕСЬ»;

Т = «ДОМА». «ВЕЧЕРОМ», «ОКОЛО», «КЛЮЧ»

Тогда отрицательный секретный ответ «нет» на заданный вопрос может иметь несколько «невинных» вариантов: «Я жду дома», «Я жду вечером», «Здесь мой ключ».

Контрольные вопросы:

1) **На чем основывается принцип работы шифра Аве Мария?**
2) **Как называлась первая печатная книга по криптографии?**
3) **Назовите год издания первой печатной книги по криптографии.**
4) **Зашифруйте слово «Криптография» шифром Аве Мария.**

1.11 Таблица Тритемия

Другой шифр, придуманный Тритемием – это так называемая «таблица Тритемия». В первой строке был вписан сам алфавит. Пример таблицы Тритемия изображен на рисунке 1.11.1.

Следующая строка писалась со смещением на одну буквы, следующая ещё на одну и так до конца алфавита. Первая строка являлась одновременно и строкой букв открытого текста. Первая буква текста шифруется по первой строке, вторая по второй и так далее после использования последней строки вновь возвращаются к первой. Так слово apple приобретает вид «aqroi».

В дальнейшем усложнения шифра пошли по двум направлениям:

- введение произвольного порядка расположения букв исходного алфавита шифрованного текста вместо упорядоченного;

– применение усложненного порядка выбора строк таблицы при шифровании.

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

Рисунок 1.11.1 – Таблица Тритемия для английского алфавита.

Шифр Тритемия является также первым нетривиальным примером периодического шифра. Так называется многоалфавитный шифр, правило зашифровывания, которого состоит в использовании периодически повторяющейся последовательности простых замен.

Затем в 1553 году итальянец Жовано Белазо предлагает улучшенный вариант шифра Тритемия. В этом шифре ключом являлся пароль – легко запоминаемая фраза или слово. Пароль записывался периодически над буквами открытого текста. Буква пароля, стоящая над соответствующей буквой открытого текста, указывала номер строки в таблице Тритемия, по которой следует проводить замену (шифрование) этой буквы. Так, если паролем является слово ROI, то при зашифровывании слова FIGHT получаем WWOYH. Аналогичные идеи шифрования используются и сегодня.

Контрольные вопросы:

1) **На чем основывается принцип работы шифра Тритемия?**
2) **Что такое таблица Тритемия?**
3) **Составьте таблицу Тритемия для русского алфавита.**

4) **Зашифруйте таблицей Тритемия слово «Information».**

5) **В чем отличие шифра Белазо от шифра Тритемия?**

6) **Как изменится зашифрованное слово «Information»при введении пароля «SUN»в шифре Белазо?**

1.12 Шифр Бэкона

Крупнейший английский философ и учёный 17 века лорд-канцлер Френсис Бэкон предложил идею стенографической защиты информации. По сути, это двоичное кодирование букв открытого текста буквами «А» и «В». Для кодирования сообщений каждая буква текста заменялась на группу из 5 букв «А» и «В», по алфавиту Бэкона. Пример алфавита Бэкона изображен на рисунке 1.12.1.

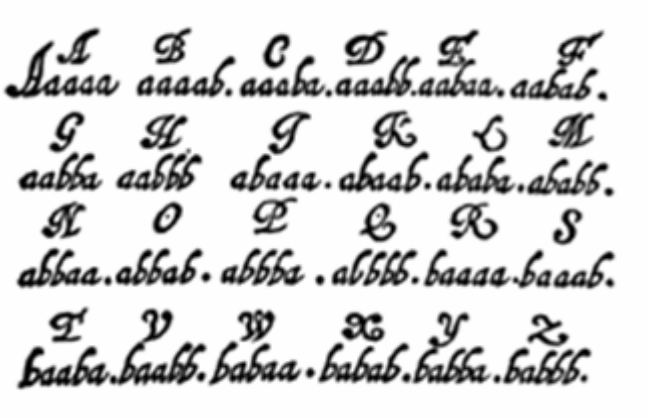

Рисунок 1.12.1 – Алфавит Бэкона.

Рассмотрим пример. Для начала закодируем слово ROME получим:

BAAAA ABBAB ABABB AABAA

Для зашифровывания этого слова использовалось произвольное невинное сообщение, например: «Солнце СВеТиТ нА ГолОву».

Здесь большим буквам соответствует буква «В», а маленьким буква «А».

BAAAA ABBAB ABABB AABAA

Солнц еСВеТ иТнАГ олОву

Большие и маленькие буквы очень отличаются, поэтому у Бэкона было два похожих шрифта, простой читатель не заметил бы ничего странного. Для расшифровывания сообщения, нужно было искать мелкие детали, и закономерность появления их в тексте. Недостатком подобного механизма хранения и передачи шифротекста является чувствительность материала к

мельчайшим изменениям в своей структуре, так простая чернильная клякса могла существенно изменить порядок, расшифровывая закрытого текста.

Контрольные вопросы:

1) **На чем основывается принцип работы шифра Бэкона?**
2) **В чем основная проблема при работе с шифром Бэкона?**
3) **Зашифруйте слово «Криптография» шифром Бэкона.**
4) **Расшифруйте фразу «вот и Наступила долГОжДаННая зима», зашифрованную шифром Бэкона.**

1.13 Шифровальный диск Альберти

Ещё один значительный шаг вперед в криптографии сделал известный итальянский философ, живописец, архитектор Леон Альберти. Он предложил шифр, основанный на использовании шифровального диска, сам он называл его шифром, «достойным королёй».

Устройство представляло собой пару дисков разного диаметра. Внешний неподвижный диск, разбивался на 24 сектора, в них были вписаны 20 букв расположенных в естественном порядке и 4 цифры (от 1 до 4). Внутренний подвижный диск также был разделен на 24 сектора, по его окружности были вписаны все буквы смешенного алфавита. Процесс шифрования заключался в нахождении буквы открытого текста на внешнем диске и замену ее на соответствующую (стоящей под ней) букву шифрованного текста. После шифрования нескольких слов диск сдвигался на один шаг. Ключом данного шифра являлся порядок расположения букв на внутреннем диске и его начальное положение относительно внешнего диска.

Рисунок 1.13.1 – диск Альберти.

Пример изображен на рисунке 1.13.1. Имея два таких прибора, корреспонденты договаривались о первой индексной букве на подвижном диске. При шифровании сообщения индексная буква ставилась против любой буквы внешнего диска. Отправитель сообщения информировал корреспондента о таком положении диска, записывая эту букву внешнего диска в качестве первой буквы шифротекста.

Следующая буква открытого текста отыскивалась на внешнем диске и стоящая против нее буква являлась результатом её зашифровывания. После того как несколько букв были зашифрованы, положение индексной буквы менялось, о чем также сообщалось корреспонденту. Такой шифр получил название многоалфавитного шифра.

Контрольные вопросы:

1) **На чем основывается принцип работы диска Альберти?**
2) **Зашифруйте диском Альберте слово «Water».**
3) **Зашифруйте диском Альберте слово «Rainbow».**
4) **Зашифруйте диском Альберте слово «Sunshine».**
5) **Почему данный шифр получил название многоалфавитного?**

1.14 Шифры Порта

Порта предложил шифр биграммной (двухбуквенной) замены, в котором каждому двухбуквенному сочетанию открытого текста в шифрованном тексте соответствовал специально придуманный знак. Знаки шифротекста имели форму символико-геометрических фигур. По сути это был шифр простой замены, но на уровне двухбуквенных сочетаний.

Порта также предложил таблицу, по которой можно было производить процесс шифрования. Он рекомендовал не использовать в переписке стандартных слов и предложений и предлагал делать ошибки в тексте, чтобы затруднит работу дешифровальщика.

Порта предложил некоторую модификацию шифра Белазо. Он представляет собой прямоугольную таблицу из букв алфавита в порядке, представленном на рисунке 1.14.1. Шифрование производится при помощи секретного лозунга. Лозунг периодически выписывается над открытым текстом, по первой букве лозунга отыскивается алфавит (большие буквы в начале строк). В верхней или нижней строке отыскивается первая буква открытого текста и заменяется соответствующей ей буквой из верхней или нижней строки.

1	А Б	а р	б с	в т	г у	д ф	е х	ж ц	з ч	и ш	й щ	к ъ	л ы	м ь	н э	о ю	п я
2	В Г	а с	б т	в у	г ф	д х	е ц	ж ч	з ш	и щ	й ъ	к ы	л ь	м э	н ю	о я	п р
3	Д Е	а т	б у	в ф	г х	д ц	е ч	ж ш	з щ	и ъ	й ы	к ь	л э	м ю	н я	о р	п с
4	Ж З	а у	б ф	в х	г ц	д ч	е ш	ж щ	з ъ	и ы	й ь	к э	л ю	м я	н р	о с	п у
5	И Й	а ф	б х	в ц	г ч	д ш	е щ	ж ъ	з ы	и ь	й э	к ю	л я	м р	н с	о т	п у
6	К Л	а х	б ц	в ч	г ш	д щ	е ъ	ж ы	з ь	и э	й ю	к я	л р	м с	н т	о у	п ф
7	М Н	а ц	б ч	в ш	г щ	д ъ	е ы	ж ь	з э	и ю	й я	к р	л с	м т	н у	о ф	п х
8	О П	а ч	б ш	в щ	г ъ	д ы	е ь	ж э	з ю	и я	й р	к с	л т	м у	н ф	о х	п ц
9	Р С	а ш	б щ	в ъ	г ы	д ь	е э	ж ю	з я	и р	й с	к т	л у	м ф	н х	о ц	п ч
10	Т У	а щ	б ъ	в ы	г ь	д э	е ю	ж я	з р	и с	й т	к у	л ф	м х	н ц	о ч	п ш
11	Ф Х	а ъ	б ы	в ь	г э	д ю	е я	ж р	з с	и т	й у	к ф	л х	м ц	н ч	о ш	п щ
12	Ц Ч	а ы	б ь	в э	г ю	д я	е р	ж с	з т	и у	й ф	к х	л ц	м ч	н ш	о щ	п ъ
13	Ш Щ	а ь	б э	в ю	г я	д р	е с	ж т	з у	и ф	й х	к ц	л ч	м ш	н щ	о ъ	п ы
14	Ъ Ы	а э	б ю	в я	г р	д с	е т	ж у	з ф	и х	й ц	к ч	л ш	м щ	н ъ	о ы	п ь
15	Ь Э	а ю	б я	в р	г с	д т	е у	ж ф	з х	и ц	й ч	к ш	л щ	м ъ	н ы	о ь	п э
16	Ю Я	а я	б р	в с	г т	д у	е ф	ж х	з ц	и ч	й ш	к щ	л ъ	м ы	н ь	о э	п ю

Рисунок 1.14.1 – Шифр Порта.

Аналогично шифруются остальные буквы открытого текста. Рассмотрим пример:

Лозунг:	м	и	р	м	и	р	м	и	р	м	и	р
Открытый текст:	с	и	н	е	е	н	е	б	о			
Шифротекст:	л	ь	х	ы	щ	х	ы	ч	ц			

У этого шифра есть несколько минусов: это необходимость иметь при себе указанную таблицу и сложность процесса шифрования, однако на основе

шифра Порто затем появились другие системы шифра (например, шифр Виженера, который мы рассмотрим чуть позже).

Контрольные вопросы:

1) **На чем основывается принцип работы шифра Порта?**
2) **Для чего необходим лозунг в шифре Порта?**
3) **Зашифруйте шифром Порта слово «Криптография» используя произвольный лозунг.**
4) **Зашифруйте шифром Порта слово «Коалиция» используя произвольный лозунг.**
5) **Зашифруйте шифром Порта слово «Верность» используя произвольный лозунг.**
6) **Зашифруйте шифром Порта слово «Счастье» используя произвольный лозунг.**
7) **Зашифруйте шифром Порта слово «Скорость» используя произвольный лозунг.**

1.15 Шифр Кардано и Решелье

В середине XVI века итальянский математик, врач и философ Дж. Кардано предлагает новый способ шифрования, так называемая «решетка Кардано». Для ее изготовления брался квадратный лист из твердого материала (картон, бумага, металл) в котором были прорезаны «окна». При шифровании решетка накладывалась на лист бумаги и буквы открытого текста вписывались в «окна». При использовании всех «окон» решетка поворачивалась на 90 градусов и процедура повторялась. Так за один «заход» решетка использовалась 4 раза. Если текст зашифровывался не полностью, то решетка ставилась в начальное положение и вся процедура повторялась. По сути, этот шифр являлся шифром перестановки. Если в квадрате после снятия решетки оставались пустые места, то в них вписывались произвольные буквы. Затем буквы квадрата выписывались построчно, что и было зашифрованным текстом. Главное требование к шифру – при всех поворотах «окна» не должны попадать на одно и то же место в квадрате, в котором образуется шифротекст.

В основе шифра Решелье лежит шифр «решетка Кардано». Здесь также из плотного материала вырезался прямоугольник размером 7 на 10, в нем проделывались «окна» различной длины. Решетку накладывали на лист бумаги в вырезанные «окна» вписывалось секретное сообщение, решетку убирали и

заполняли свободные места неким текстом, которое маскировало секретное сообщение. Пример шифра Решелье изображен на рисунке 1.15.1.

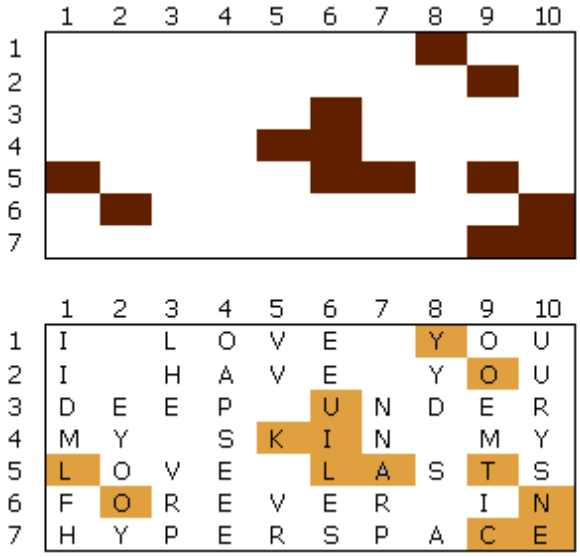

Рисунок 1.15.1 – Шифр Решелье.

Так за невинным любовным письмом скрывается зловещая команда: «YOU KILL AT ONCE». Данный способ шифрования получил широкое применение.

Контрольные вопросы:

1) **На чем основывается принцип работы шифра Кардано?**
2) **В чем отличие шифра Решелье от шифра Кардано?**
3) **Как скрыть факт передачи сообщения шифром Кардано?**
4) **Зашифруйте шифром Решелье слово «Криптография».**
5) **Зашифруйте шифром Решелье слово «Концепция».**
6) **Зашифруйте шифром Решелье слово «Window».**
7) **Зашифруйте шифром Решелье слово «Shine».**
8) **Зашифруйте шифром Решелье слово «Солнце».**

1.16 Шифр Виженера

Посол Франции в Риме Блез де Виженер в 1585 году в своем сочинении «Трактат о шифрах» предлагает идею использования в качестве ключа самого открытого текста, кроме того первая буква ключа берется произвольно из алфавита. Этот шифр имел название «шифр самоключ», за основу бралась

таблица Тритемия, которая в последствии стала называться таблицей Виженера. Авторами идеи был Дж. Кардано и сам Блез де Виженер. Рассмотрим подробней процесс шифрования, для этого составим таблицу, что изображена на рисунке 1.16.1.

В первой строке таблицы алфавит может быть расположение в естественном порядке или в произвольном. Открытый текст записывается в строчку без пробелов, затем для формирования ключа используем этот же открытый текст с добавлением к началу ключа случайного символа, который знают и получатель и отправитель. Пара букв стоящих друг под другом в открытом тексте и в ключе (G и Q), указывали номер столбца и строки, соответственно, на пересечении которых находилась зашифрованная буква.

Открытый текст: GREENTEA

Ключ: QGREENTE

Шифротекст: WXVIRGXE

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

Рисунок 1.16.1 – Таблица Виженера.

Для расшифровывания необходимо было знать шифротекст и ключ. Во втором варианте Виженер предлагал использовать в качестве ключа лозунг. Граф Гронсфельд – начальник первого в Германии государственного дешифровального органа внес изменения в «шифр Виженера». Эти изменения привели к появлению, так называемого, шифра гаммирования – одного из

самых распространённых шифров в современной криптографии. Рассмотрим процесс шифрования.

Выпишем английский алфавит:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

В качестве ключа – лозунга выбирается легко запоминающееся число, например, 123123. Этот ключ – лозунг периодически выписывается над буквами открытого текста:

Ключ – лозунг: 1 2 3 1 2 3 1 2 3 1

Открытый текст: STRAWBERRY

Далее для зашифровывания каждая буква открытого текста заменяется буквой, стоящей от неё справа в алфавите на количество букв, определяемых соответствующей цифрой лозунга. Таким образом, получаем:

Шифротекст: TVUBYEFTUZ

Этот шифр продолжал изменяться, что в последствии привело к шифру модульного гаммирования. Пронумеруем буквы алфавита: A=01, B=02, C=03,…,Z=26. В таком случае слово «STRAWBERRY» приобретает вид:

19.20.18.01.23.02.05.18.18.25.

Зашифровывание проводится по операции модульного сложения (от операции обычного сложения эта операция отличается тем, что в случае если сумма чисел превышает число 26, то из этого числа вычитается 26; при операции вычитания, если в результате получалось отрицательное число, то к нему прибавляли 26). Запишем лозунг над открытым текстом и сложим соответствующие числа:

Ключ – лозунг: 1 2 3 1 2 3 1 2 3 1

Открытый текст: 19.20.18.01.23.02.05.18.18.25.

Шифротекст: 20.22.21.02.25.05.06.20.21.26.

Соответственно: TVUBYEFTUZ

При расшифровывании ключ – лозунг вычитается из букв шифротекста.

Контрольные вопросы:

1) **На чем основывается принцип работы шифра Виженера?**
2) **Что такое шифр гаммирования?**

3) **Кто предложил шифр гаммирования?**

4) **Что выбирается в качестве ключа в шифре гаммирования?**

5) **Что лежит в основе шифра Виженера?**

6) **Зашифруйте слово «Криптография» шифром Виженера, используя произвольный ключ.**

7) **Зашифруйте слово «Криптография» используя шифр гаммирования с произвольным ключом.**

8) **В чем отличие шифра Виженера от шифра гаммирования?**

1.17 Шифр Фальконера

В XVII веке Дж. Фальконер предложил интересную систему шифрования, которая представляла собой сочетание вертикальной перестановки и гаммирования. В этом шифре, как и во многих других, также используется лозунг. Рассмотрим процесс зашифровывания:

Пусть лозунгом будет слово «АВАРИЯ», по этому лозунгу строится так называемые номерной ряд. Но для начала выпишем алфавит по порядку:

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

1 3 4 5 6

2

Далее буквы лозунга заменяются цифрами. Буква лозунга стоящая в алфавитном порядке первее всех будет иметь значение 1, буква, находящаяся в алфавитном порядке правее цифру 2, и так далее. Причем, если буква повторяется, то она получает последовательные номера. Так наш лозунг принимает следующий вид: 132546.

Зашифруем фразу «УЧЕНЬЕ СВЕТ, А НЕУЧЕНЬЕ ТЬМА». При зашифровывании используется прямоугольник, в который вписывается фраза обычный способом (слева направо), слова вписываются без пробела. Столбы отмечаются цифрами номерного ряда. В нашем случае получаем следующую таблицу (1.17.1). Первичный шифротекст получаем путем выписывания букв столбца, а столбцы следуют друг за другом в естественном порядке (от 1 до 6).

1 3 2 5 4 6

У	Ч	Е	Н	Ь	Е
С	В	Е	Т	А	Н

Е	У	Ч	Е	Н	Ь
Е	Т	Ь	М	А	

Таблица 1.17.1 – Шифр Фальконера.

В результате получаем:

У С Е ЕЕЕ Ч Ь Ч В У Т Ь А Н А Н Т Е М Е Н Ь

Затем этот первичный шифротекст зашифровывается вторично шифром модульного гаммирования, о котором мы говорили выше. В результате получает зашифрованный текст.

Контрольные вопросы:

1) **На чем основывается принцип действия шифра Фальконера?**
2) **Зашифруйте шифром Фальконера слово «Криптография».**
3) **Зашифруйте шифром Фальконера слово «Красноречие».**
4) **Зашифруйте шифром Фальконера слово «Механика».**
5) **Зашифруйте шифром Фальконера слово «Философия».**
6) **Зашифруйте шифром Фальконера слово «Физика».**

1.18 Шифр Кеплера и Галилея

Пожалуй, последним примером исторических шифров, который мы разберем, станет шифр Кеплера и Галилея. Шифры Кеплера и Галилея являются осмысленной перестановкой символов или сочетаний исходного текста, так называемое анаграммирование. Так, фразу «ВСТРЕЧАЙ ЗАВТРА И КОЛЮ» Галилей мог бы анаграммировать так: «ЗАВТРАК И ЧАЙ ЮРЕ В СТОЛ». Получение анаграмм «осмысленного» вида было большим искусством. Позже последователи создавали упрощенные анаграммы:

«3А,2В,1Е,1З,1И,1Й,1К,1Л,1О,2Р,1С,2Т,1Ч.1Ю»
«АААВВЕЗИЙКЛОРРСТТЧЮ» и т.д.

Анаграммирование можно рассматривать как упрощённый вариант шифра перестановки, хотя, по сути, эти преобразования не являются шифрованием. Автор анаграммы, отсылая её своим корреспондентам, не предполагал ее расшифровывания. Более того он считал его практически невозможным. Действительно, восстановить открытый текст из анаграммы весьма сложная задача. Цель анаграмм заключалась в защите «авторских прав» без публичного опубликования своих «еретических» идей. Записав обычным

текстом своё открытие, автор прятал эту бумагу и создавал анаграмму, которую и отсылал корреспонденту. В сопроводительном тексте автор сообщал, что удалось открыть удивительно явление, суть которого спрятана за анаграммой. В случае необходимости для подтверждения своего приоритета автор мог предъявить исходный открытый текст. Его истинность подтверждалась ранее отправленной анаграммой. Эта идея в более совершенном виде широко используется и в наши дни. Для подтверждения подлинности посылаемого сообщения и отсутствия его подмены.

Контрольные вопросы:

1) **В чем смысл шифров Кеплера и Галилея?**
2) **На чем основывается принцип действия этих шифров?**
3) **Что такое анаграмма?**
4) **Почему шифр анаграмм сложен для восприятия?**
5) **Для чего использовались анаграммы?**
6) **Зашифруйте анаграммой словосочетание «Криптостойкий Алгоритм», допустив замену Й на И.**

1.19 Основные понятия криптографии

Криптография тесно связанна с различными областями технических дисциплин таких как: математика (алгебра, теория вероятности, теория сложности, теория чисел, вычислительная математика и т.д.), теория связи, теория кодирования. Существует техника защиты передаваемых данных, основанная на их сокрытии (стенография), например, акростихи или использование невидимых чернил. В последние годы методы стенографии широко используются для защиты от нелегального копирования (то есть для контроля не повторяемости). Например, если каждый экземпляр программного продукта пометить индивидуальным номером, спрятанным в тексте программ, то можно соотнести владельца данного экземпляра программы с владельцем лицензионного экземпляра, который приобрел его на законных основаниях. Иногда задачи стенографии могут решаться с использованием криптографических средств, например, возможен скрытый канал передачи зашифрованных данных в протоколе цифровой подписи. В предыдущей части главы мы знакомились с историей развития криптографии, и теперь подошло время узнать что, же такое криптография, криптосистема и многое другое.

Криптографическая защита – защита данных при помощи криптографического преобразования данных.

Криптография – это раздел прикладной математики, изучающий модели, методы, алгоритмы, программные и аппаратные средства преобразования информации (шифрования) в целях сокрытия ее содержания, предотвращения видоизменения или несанкционированного использования.

Криптосистема – это система, реализованная программно, аппаратно или программно – аппаратно и осуществляющая криптографическое преобразование информации.

Криптоанализ – это раздел прикладной математики, изучающий модели, методы, алгоритмы, программные и аппаратные средства анализа криптосистемы или ее входных и выходных сигналов с целью извлечения конфиденциальных параметров, включая открытый текст.

Криптология – наука, объединяющая криптографию и криптоанализ.

Криптографическим алгоритмом защиты информации называется последовательность действий, обеспечивающая преобразование защищаемой информации по правилу, заданному ключом.

Криптоалгоритмы могут выполнять шифрование (дешифрование) сообщений, формирование и проверку аутентификаторов сообщений, генерацию ключей, необходимых для выполнения других криптоалгоритмов, и т.п.

Шифрование – это преобразование данных в вид, недоступный для чтения без соответствующей информации (ключа шифрования). Шифрование обеспечивает криптографическую защиту данных от несанкционированного доступа.

Открытый текст (сообщение) – это текст, подлежащий криптографической защите.

Шифром будем называть обратимое преобразование множества текстов с целью обеспечения их конфиденциальности, как правило, выполняемое с помощью ключа. Устройство (или программу), реализующее шифр, назовем шифратором.

Зашифрованный текст (шифротекст) – это образ открытого текста, полученный в результате действия шифра. Если множества открытых и зашифрованных текстов совпадают, то шифр называется эндоморфным. Эндоморфный шифр реализует подстановку, действующую на множестве текстов.

Под шифрованием информации будем понимать процесс перевода открытого текста в зашифрованный (зашифровывание), и обратно (расшифровывание). Основная цель шифрования – обеспечить конфиденциальность открытого текста.

Ключ – конкретное секретное состояние некоторых параметров алгоритма криптографического преобразования данных, обеспечивающее выбор одного преобразования из совокупности всевозможных для данного алгоритма преобразования. Все криптосистемы должны иметь средства управления ключами. Создание, передача и хранение ключей называется управление ключами.

Дешифрованием (вскрытием ключа) шифра назовем несанкционированное нарушение конфиденциальности, достигнутое методами криптоанализа.

Согласно принципу Керхгоффса, надёжность криптографической системы должна определяться сокрытием секретных ключей, но не сокрытием используемых алгоритмов или их особенностей. Классификация ключей приведена на рисунке 1.19.1.

Подключи – ключевая информация, вырабатываемая в процессе работы криптографического алгоритма на основе ключа. Зачастую подключи вырабатываются на основе специальной процедуры развёртывания ключа.

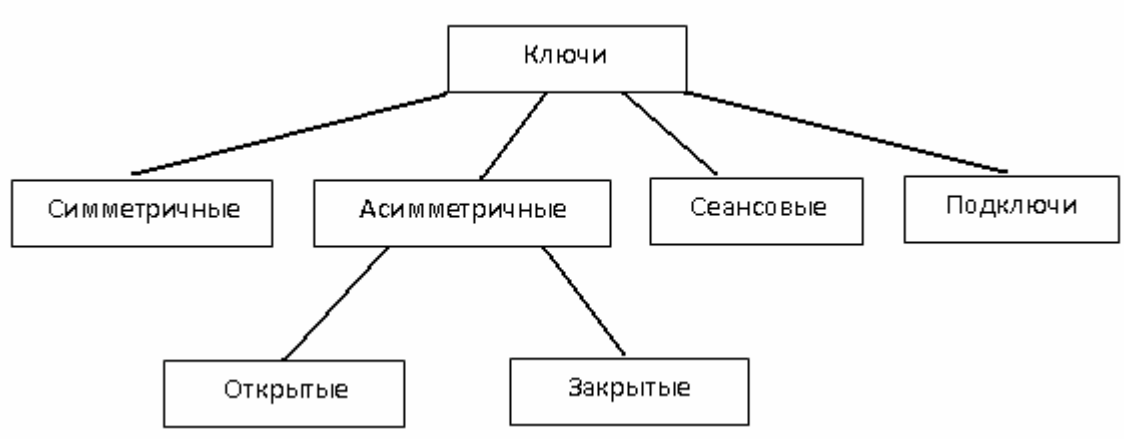

Рисунок 1.19.1 – Классификация ключей.

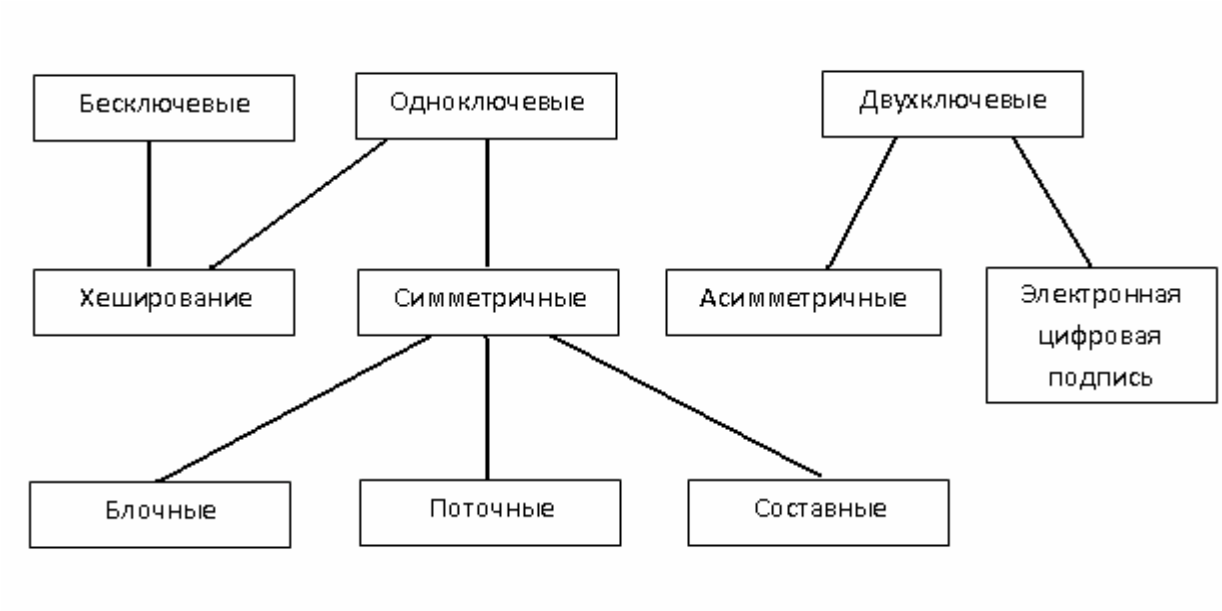

Рисунок 1.19.2 – Классификация криптографических алгоритмов.

Открытый ключ (публичный ключ) – несекретный набор параметров асимметричной криптографической системы, необходимый и достаточный для выполнения отдельных криптографических преобразований.

Закрытый ключ (секретный ключ) – набор секретных параметров одного из алгоритмов асимметричной криптосистемы.

Рассмотрим классификацию криптографических алгоритмов приведенных на рисунке 1.19.2 и коротко охарактеризуем их.

Бесключевые криптосистемы не используют какие–либо ключи в процесс криптографических преобразований. К бесключевым криптосистемам относятся хеш–функции и генераторы псевдослучайных чисел.

Хеш-функции – это математические или иные функции, принимающие на входе строку переменной длины (называемую прообразом) и преобразующие её в выводную сроку фиксированной (обычно меньшей) длины, называемую значением хеш-функции.

Хеш-функции используются для контроля целостности сообщений, передаваемых по каналу связи, и генерации ЭЦП. Пусть $g:\{0,1\}\rightarrow\{0,1\}^{q(n)}$ – функция, вычислимая за полиномиальное от n время, q(n) – некоторый полином. Такая функция называется генератором. Генератор g псевдослучаен, если порождаемые им последовательности неотличимы никаким полиномиальным вероятностным алгоритмом от случайных последовательностей той же длины q(n). Качественный ключ, для использования в рамках симметричной криптосистемы, представляет собой случайный двоичный набор. Если требуется ключ разрядностью n, в процессе его генерации с одинаковой вероятностью должен получаться любой из 2 возможных вариантов. Генерация ключей для асимметричных криптосистем –

процедура более сложная, т.к.ключи, применяемые в таких системах, должны обладать определёнными математическими свойствами. Например, в случае системы RSA модуль шифрования представляет собой произведение двух больших простых чисел.

Хеширование – процесс обработки хеш-функции. Хеш-функцией называется функция, отображающая аргумент произвольной конечной длины в образ фиксированной длины. Подробней о хеш-функции мы поговорим в 4 главе данного учебного пособия.

Одноключевые криптосистемы используют в своих вычислениях некий секретный ключ. С помощью ключевой хеш-функции может обеспечиваться имитозащита (частный случай аутентификации). Говорят, что криптоалгоритм реализует имитозащиту данных, если он обеспечивает защиту от навязывания ложной информации. Иногда в понятие имитозащиты включают и контроль того, что данные предназначены именно тому адресату, который их получил.

Если для имитозащиты используется ключевая хеш-функция, то аргументом хеш-функции являются защищаемые данные, а значение хеш-функции передается вместе с данными. Контроль подлинности и целостности осуществляется путем сравнения вычисленного и полученного значения хеш-функции. Если эти значения совпадают, то при некоторых условиях можно утверждать, что источником данных является владелец секретного ключа и данные в ходе передачи не были искажены.

Симметричное шифрование – алгоритм шифрования, использующий один и тот же ключ для шифрования и расшифровывания. Симметричная криптография позволяет не только шифровать, но и устанавливать подлинность (например, с помощью MAC). В симметричных криптографических системах отправить и получатель сообщения используют один и тот же ключ, называемый секретный. Отправить зашифровывает сообщение секретным ключом, а получатель этим же ключом расшифровывает полученное сообщение. Этот способ шифрования также называется криптографией секретного ключа. Основная трудность при этом состоит в том, что получателю и отправителю необходимо согласовать секретный ключ таким образом, чтобы он не стал известен возможным злоумышленникам. В случае если получатель и отправить находятся на большом расстоянии друг от друга, то возникает необходимость защиты ключа от перехвата при передаче ключа по каналу связи. Так как в противном случае злоумышленник, перехватив ключ при передаче, сможет читать, изменять и подделывать сообщения этим ключом. Преимущество криптографии секретного ключа состоит в том, что основанные на ней алгоритмы выполняются быстрее, чем алгоритмы основанные на криптографии открытого ключа (ассиметричная криптография). Криптография секретного ключа подразделяется на:

– блочные шифры;

– поточные шифры.

Подробней о блочных и поточных шифрах мы поговорим в следующих главах.

MAC (Message Authentication Code) – код установления подлинности сообщения. MAC обычно применяется для обеспечения целостности и защиты от фальсификации передаваемой информации. Эта функция на основе входа переменной длинны и ключа производит выход фиксированной длинны. Код установления подлинности сообщения – опознавательный признак (также называемый контрольной суммой). Рассчитываемый на основе сообщения при помощи опознавательной схемы и секретного ключа. В отличие от цифровой подписи, коды MAC вычисляются и проверяются одним и тем же ключом, и поэтому их может проверить только получатель, имеющий этот ключ. Существуют четыре типа кодов аутентификации сообщения. Абсолютно защищенный. Симмонс и Стинсон предложили абсолютно защищённый MAC, основанный на шифровании одноразовым ключом. Зашифрованный текст подтверждает собственную подлинность, поскольку доступ к одноразовому ключу закрыт. Однако при этом сообщение должно иметь некоторую избыточность. Абсолютно защищенный MAC можно создать также с помощью одноразового секретного ключа, на основе хеш-функции. Для создания MAC на основе хеш-функции (так называемого HMAC) вместе с хеш-функцией используется один или несколько ключей, созданная контрольная сумма добавляется в конец сообщения.

Двухключевые криптосистемы (криптосистемы открытого ключа) используют на различных этапах два вида ключей: закрытый и открытый. Такие системы применяются для шифрования и цифровой подписи. В ассиметричных криптосистемах для шифрования и расшифровывания информации используются разные ключи. В таких системах каждый пользователь получает пару ключей: открытый ключ, которым информация зашифровывается, и закрытый ключ, которым информации расшифровывается. Причем открытый ключ публикуется открыто, а закрытый ключ сохраняется владельцем в секрете. Таким образом, необходимость передачи секретной информации исчезает. Однако, более подробней о криптосистемах открытого ключа мы поговорим в 4 главе нашей книги.

Электронно-цифровая подпись (ЭЦП) используется физическими и юридическими лицами в качестве аналога собственноручной подписи для придания электронному документу юридической силы, равной юридической силе документа на бумажном носителе, подписанного собственноручной подписью правомочного лица и скрепленного печатью.

Электронный документ – это любой документ, созданный и хранящийся на компьютере, будь то письмо, контракт или финансовый документ, схема, чертеж, рисунок или фотография. ЭЦП обеспечивает:

– проверку целостности документов;

– конфиденциальность документов;

– установление лица, отправившего документ.

При формировании электронной подписи под сообщением, отправляемым неким абонентом в той же информационной системе, отправитель подписывает послание своим секретным ключом. На самом деле пользователь вычисляет контрольную сумму сообщения, шифрует ее секретным ключом и присоединяет шифрограмму к сообщению.

При создании электронной подписи под документом в нее закладывается достаточно информации, чтобы любой получатель мог удостовериться с помощью открытого ключа отправителя, что только он мог подписать, а, следовательно, и отправить это сообщение. Но при этом в подписи не должно быть достаточно информации, чтобы извлечь из нее сам секретный ключ отправителя – иначе после первого подписания любой перехвативший письмо мог бы подписывать свои послания, т. е. технология ЭЦП очень напоминает асимметричный шифр, только наоборот. Следует отметить, что асимметричное шифрование и ЭЦП решают совершенно различные задачи: первое – обеспечение конфиденциальности послания, второе – аутентичность отправителя и целостность сообщения. В основе любой схемы асимметричного шифрования либо инверсной ей схемы цифровой подписи лежит определенная трудноразрешимая математическая задача. При этом к данной задаче на самом деле есть способ найти решение, но для этого нужно обладать некоторой дополнительной информацией – в англоязычной литературе ее называют trapdoor – потайная дверь.

В качестве открытого ключа в ассиметричной криптографии выбирается какое-либо частное уравнение, которое и является этой трудноразрешимой задачей. Но при составлении этого уравнения оно разрабатывалось так, что лицо, знающее некоторую дополнительную информацию об этом уравнении, может решить его за разумный временной интервал. Эта дополнительная информация и является закрытым ключом.

В идеальном случае при наличии закрытого ключа и при соответствующем выборе всех параметров задачи процедура шифрования или подписания документа длятся доли секунды, а вот на взлом их без знания секретного ключа требуются десятилетия. При собственно асимметричном шифровании получатель сообщения публикует в качестве открытого ключа часть параметров уравнения, которое только он, зная закрытый ключ, сможет разрешить. Отправитель сообщения делит сообщение на блоки необходимой длины, преобразует их в большие натуральные числа и тем самым заканчивает формирование уравнения. В качестве шифровки посылаются некоторые параметры уравнения или его значения относительно какого–либо вектора, чего недостаточно для злоумышленника, чтобы восстановить исходное послание.

Однако их достаточно для получателя, чтобы решить с помощью закрытого ключа уравнение и восстановить добавленный отправителем параметр, который и является исходным текстом.

При формировании ЭЦП процесс идет в несколько ином порядке. Отправитель письма добавляет к письму некоторую часть закрытого ключа в таком виде, чтобы по ней невозможно было полностью восстановить закрытый ключ. Однако, этой информации достаточно, чтобы помочь любому желающему (проверяющему ЭЦП) решить то самое уравнение, на базе которого построена данная схема ЭЦП. В качестве других параметров уравнения проверяющий подставляет контрольную сумму полученного письма и значения из открытого ключа отправителя. Если уравнение успешно разрешилось, т. е. при подстановке всех данных оно превратилось в равенство, значит, письмо с данной контрольной суммой отправил именно тот абонент, чья подпись под ним стоит. Если же равенство не получилось, то на каком-то из этапов произошел сбой и необходимо уже более тщательно выяснить, был ли это случайный сбой на канале связи или же было умышленное вмешательство. В асимметричной криптографии и ЭЦП ключ является более сложным компонентом, чем просто 128-битный блок данных симметричной криптографии. Чаще всего и открытый и закрытый ключи состоят из двух или трех очень больших (сотни и даже тысячи бит) натуральных чисел, однако встречаются и случаи, когда ключ состоит из сотен натуральных чисел – определенной последовательности или двумерной матрицы. Поэтому не следует удивляться тому, что в качестве ключа, или цифровой подписи далее встретятся наборы чисел – все это неразделимый ключ, имеющий смысл только в сочетании всех ее компонент. Не стоит удивляться и их размерам (есть схемы с ключами размером сотни килобайт).Далее отметим, что на самом деле процедуры асимметричного шифрования еще и очень медлительны. Этот недостаток, правда, успешно нейтрализуется объединением асимметричного шифрования с блочными шифрами. Весь текст сообщения преобразуется обычным блочным шифром (намного более быстрым), но с использованием случайного ключа получателя и помещается в начало шифрограммы.

Контрольные вопросы:

1) **Что такое криптоанализ?**
2) **Что такое криптосистема?**
3) **Что такое криптология?**
4) **Что такое шифрования?**
5) **Что такое ключ?**
6) **Классифицируйте существующие криптографические ключи.**
7) **Что такое подключ?**
8) **Что такое открытый и закрытый ключ?**

9) **Классифицируйте криптосистемы.**

10) **В чем отличие бесключевых от ключевых систем?**

11) **Что такое хеш–функция?**

12) **Что такое ЭЦП?**

13) **Что такое ключевая хеш–функция?**

14) **Что такое симметричное шифрование?**

15) **Что такое MAC код?**

16) **Что такое ассиметричная криптосистема?**

17) **Для чего используется ЭЦП?**

18) **Что такое «потайная дверь»?**

1.20 Функции, используемые в криптографических системах

Принципы построения криптографических систем защиты информации основаны на использовании математических функций специального вида, которые должны легко вычисляться законными пользователями, знающими «ключ», и очень сложно для всех не обладающих ключом. Подобно любой математической системе в криптографии существует общее описание функций, наиболее часто используемых в различных алгоритмах.

Рассмотрим пример произвольной функции y = *f*(x), которую зададим графически, рисунок 1.20.1. Пусть задано множество X={a,b,c,d,e} и множество Y={1,2,3,4,5}. Напомним, что функция определяется двумя множествами X и Y, и правилом *f*, которое назначает каждому элементу из множества X один элемент из множества Y. Множество X называется областью определения функции, а множество Y областью ее значений.

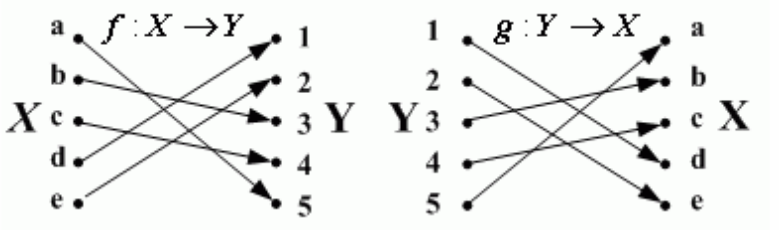

Рисунок 1.20.1 – Биективная функция f и обратная к ней.

Элемент y из множества Y является образом элемента x, а элемент x x является прообразом y. Отображение элементов из множества X в множество Y записывают так: $f: X \to Y$. Множество всех элементов y, имеющих

хотя бы один прообраз, называется образом функции f и обозначается $Im(f)$. Функция называется однозначной (отображением один в один), если каждый элемент из множества Y является образом не более одного элемента из множества X. Функция f называется биекцией, если она является однозначной и $Im(f) = Y$. Функция вида $g = f^{-1}$ называется обратной к f .Среди биективных функций, есть класс функций называемых инволюциями, которые наиболее часто используются для построения симметричных криптографических систем защиты информация.

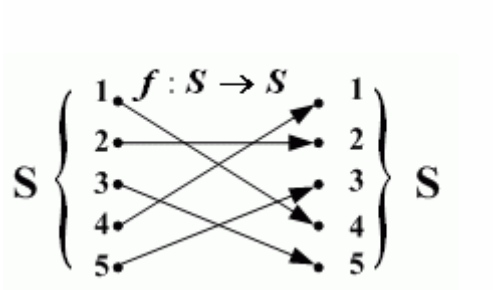

Рисунок 1.20.2 – Инволюция f для множества S={1,2,3,4,5}.

Биективная функция называется инволюцией, если у функции совпадает область определения и область ее значений, т.е. X=Y=S, а также обратная функция с прямой $f = f^{-1}$.Пример инволюции для множества S={1,2,3,4,5} показан на рисунке 1.20.2.

Существование обратной функции является основой построения систем шифрования информации, с помощью которой можно однозначно дешифровать криптограммы в сообщения. Последовательное применение сначала функции шифрования, а затем функции дешифрования к произвольному сообщению $x \in S$ однозначно восстанавливает данное сообщение: $f(f(x)) = x$.

Контрольные вопросы:

1) **Что такое инволюция?**
2) **Что такое обратная функция?**
3) **Что такое биективная функция?**
4) **Объясните рисунок 1.19.1 и 1.19.2 .**
5) **Для чего применяются инволюции в криптографии?**

1.21 Однонаправленные функции

Особую роль в криптографии играют однонаправленные функции, которые в общем случае не являются биективными.

Однонаправленной называется такая функция f, для которой легко определить значение функции y= f (x), но практически невозможно отыскать для заданного y такое x, что y= f (x).

Для построения криптографических систем защиты информации чаще используются однонаправленные функции, для которых обратное преобразование существует и однозначно, но вычислительно нереализуемо. Они называются вычислительно необратимыми функциями. В качестве примера однонаправленной функции y= f (x) рассмотрим функцию дискретного возведения в степень: $y = a^x (mod\ p)$, где x – целое число от 1 дор– 1 включительно, а вычисление производится по модулю p, где p – очень большое простое число; a – целое число (1< $a < p$). Напомним, что простым числом называется целое число, которое не делится ни на какие числа, кроме себя самого и единицы. Для примера возьмем небольшое простое число p=7; тогда для осуществления преобразований можно выбрать примитивный элемент a=3, так как $a^1 (mod\ 7) = 3$,$a^2 (mod\ 7) = 3^2 (mod\ 7) = 9\ (mod\ 7) = 2$,$a^3 (mod\ 7) = 6$,$a^4 (mod\ 7) = 4$,$a^5 (mod\ 7) = 5$, $a^6 (mod\ 7) = 1$.

Функция $y = a^x (mod\ p)$ вычисляется сравнительно просто, а обратная к ней функция $x = log_y p$ является вычислительно сложной практически для всех (1<y<p) при условии, что не только p велико, но и (p–1) имеет большой простой множитель (лучше всего, если это будет другое простое число, умноженное на 2). В связи с этим такую задачу называют задачей нахождения дискретного логарифма или задачей дискретного логарифмирования.

Задача дискретного логарифмирования состоит в том, что для известных целых a, p, y необходимо найти целое число x. Однако алгоритм вычисления дискретного логарифма за приемлемое время пока не найден. Поэтому модульная экспонента считается однонаправленной функцией.

По современным оценкам теории чисел при целых числах $a \approx 2^{664}$ и p$\approx 2^{664}$ решение задачи дискретного логарифмирования потребует около 10^{26} операций, что имеет в 10^{3} раз большую вычислительную сложность, чем задача разложения на множители. При увеличении длины чисел разница в оценках сложности задач возрастает. Следует отметить, что пока не удалось доказать, что не существует эффективного алгоритма вычисления дискретного логарифма за приемлемое время. Исходя из этого, модульная экспонента отнесена к однонаправленным функциям условно, что, однако, не мешает с успехом применять ее на практике.

Одним из первых применений однонаправленных функций было решение задачи обеспечения безопасности и использования пароля, по которому осуществляется доступ пользователя к ресурсам и услугам в автоматизированных системах.

Открытое значение y вместе с именем пользователя может быть помещено в список паролей доступа, хранящихся в ЭВМ. Законный пользователь для получения доступа в автоматизированную систему предъявляет свое число x. ЭВМ вычисляет по этому числу значение однонаправленной функции $y = a^x (mod\ p)$ и сравнивает с хранящимся значением y. При совпадении этих значений пользователь становится идентифицированным и получает требуемый доступ.

Кроме однонаправленных функций, не имеющих вычислительно простого обратного отображения даже для законных пользователей, знающих секретную ключевую информацию, в криптографии широко используются однонаправленные функции, для которых знание секретного ключа дает возможность законному пользователю вычислительно просто находить обратное отображение. Они получили название однонаправленных функций с потайным ходом, иногда их называют однонаправленными функциями с лазейкой.

Быстрое развитие криптографии в последние два десятилетия во многом стало возможным благодаря открытию американскими учеными В. Диффи и М. Хэлманом однонаправленных функций с потайным ходом и их использованием для различных криптосистем защиты информации.

Однонаправленная функция с потайным ходом есть однонаправленная функция f_z с дополнительным свойством, таким, что, зная информацию z потайного хода для каждого $y \in \mathrm{Im}(f)$ вычислительно просто определить $x \in X$, удовлетворяющее уравнению $y = f_z(x)$.

Для нарушителя, не знающего информации z потайного хода, нахождение отображения может быть сделано вычислительно нереализуемым. Поэтому информация z может служить секретным ключом для пользователя функций с потайным ходом. Однонаправленные функции с потайным ходом относятся к вычислительно необратимым функциям. Функция вычислительно необратима, если при попытке формирования алгоритма нахождения обратного отображения к ней противник наталкивается на непреодолимую вычислительную проблему. Оценивая стойкость криптосистем, построенных на основе известных однонаправленных функций с потайным ходом, отметим, что ни одна из них не является безусловно стойкой. Это объясняется тем, что нарушитель с теоретически бесконечными вычислительными ресурсами способен вычислять обратное отображение к таким функциям. На основе однонаправленных функций с потайным ходом можно построить криптосистемы аутентификации информации в условиях взаимного недоверия корреспондентов, системы шифрования информации, в которых отправители сообщений могут пользоваться несекретными ключами шифрования, криптосистемы обмена секретной ключевой информации по открытым каналам связи, а также многие другие криптосистемы.

К настоящему времени предложено большое количество однонаправленных функций с потайным ходом, построенных на основе известных вычислительно сложных математических задач. Наиболее часто для построения однонаправленных функций с потайным ходом используется сложность решения следующих теоретико-числовых задач:

- отыскание дискретного логарифма элемента в большом конечном поле или группе (криптосистема открытого распространения ключей Диффи-Хэллмана, криптосистема шифрования и криптосистема цифровой подписи сообщений Эль-Гамаля, криптосистема цифровой подписи сообщений Шнорра и другие криптосистемы);
- разложение больших чисел на простые множители (криптосистема шифрования и криптосистема цифровой подписи сообщений RSA, криптосистема цифровой подписи сообщений Рабина и другие криптосистемы);
- задача об укладке целочисленного ранца;
- декодирование неизвестных получателю кодов Гоппы.

Контрольные вопросы:

1) **Что такое однонаправленная функция?**
2) **Что такое необратимая функция?**
3) **Для чего впервые использовалась необратимость функции в криптографии?**
4) **Что такое однонаправленная функция с потайным ходом?**
5) **Использование каких задач наиболее часто применяется для создания однонаправленных функций?**

1.22 Имитостойкость

Обозначим через A множество информационных сообщений, передаваемых по каналу связи от передающего устройства к приемному устройству. Предполагается, что противник «контролирует» канал связи и может заменить передаваемое сообщение $a \in A$ (истинное сообщение) на сообщение $a' \in A$ (ложное сообщение). Под обеспечением имитостойкости приемного устройства понимается его защита от навязывания информации, поступающей в результате воздействия противника на передаваемую по каналу связи информацию, а также и на «пустой канал». Ложная информация считается навязанной, если она принята приемным устройством к исполнению (т.е. принята также как принимается истинное сообщение). Воздействие помех

в канале связи на передаваемую информацию также трактуется как попытка навязывания ложной информации.

Для узнавания факта навязывания информации и предусмотрена имитозащита. Существует много способов имитозащиты, позволяющих установить приемному устройству вмешательство противника (или помех) в процесс передачи по каналу связи истинного сообщения a∈ A. Как правило, такие способы имитозащиты основаны на использовании шифров и не обеспечивают стопроцентную гарантию защиты от навязывания ложной информации.

Рассмотрим некоторые варианты навязывания информации способы имитозащиты от них. Пусть A=Y и передаются шифрованные сообщения y∈ Y. Противник перехватывает передаваемое сообщение «y» и через определенный промежуток времени посылает его в канал связи. Если за этот промежуток времени не менялся ключ на приемном и передающем устройстве, то приемное устройство повторно воспримет ранее передаваемую информацию. Для защиты от таких действий противника используют метки времени (проставляя время отправления в открытом сообщении x∈ X). В случае, когда отмеченное время отличается от времени принятия сообщения *y* на величину, превышающую некоторый допустимый предел, приемное устройство делает вывод о принятии ложного сообщения. В некоторых случаях используют секретный алгоритм изменения ключа шифра через короткие промежутки времени. Если задержка в приеме превышает величину этого промежутка, то приемник не расшифрует навязываемое сообщение.

Пусть множество возможных сообщений A является множеством F_2^k двоичных наборов длины *k*. Противник посылает сообщение a’ в канал связи и приемное устройство принимает его. Для обеспечения защиты от навязывания сообщения a’ применяют кодирование векторов F_2^k двоичными векторами длины n>k. В этом случае множеством передаваемых сообщений становится некоторое неизвестное противнику подмножество M мощности 2^k векторов из F_2^n. При случайном и равновероятном выборе противником вектора a’ из F_2^n и его передачи по каналу связи вероятность принятия ложного сообщения a’ совпадает с вероятностью *P* события: a’$_\in$M, очевидно, $P = \frac{2^k}{2^n} = 2^{k-n}$.

Противник знает передаваемое сообщение «a» и может заменить его на a’$_\in$A. Для обеспечения имитостойкости в этом случае используют имитовставки. Предполагается, что приемное и передающее устройства снабжены шифром зашифровывания, позволяющим им на ключах шифра вырабатывать одинаковые гаммы. Через Г обозначим множество всех возможных гамм шифра. На A× Г определяется функция F: A× Г →Z, где Z – некоторое множество называемое множеством проверочных символов (кодов

имитозащиты). Для передачи сообщения $a \in A$ (на передающем устройстве) вырабатывают гамму $Y \in \Gamma$ и по каналу связи передают пару (a,z), где z=F(a, Г). На приемном устройстве с помощью шифра вырабатывается та же самая гамма Г и проверяется для принятой из канала пары (a', z'), совпадение значения F(a', Г) со значением z'. При совпадении значений делается вывод о приеме истинного сообщения. В противном случае, делается вывод о приеме ложного сообщения. Предполагается, что на приемное устройство противник может послать произвольную пару (a',z') из A× Z, для которой a≠a'. В предположении, что выбор гамм $Y \in \Gamma$ проводится случайно и равновероятно, из Г вероятность P((a',z'),(a,z)) навязывания приемному устройству сообщения a' совпадает с вероятностью P(z'=F (a',Y), z=F(a ,Y) события:

z'=F (a', Y), z=F(a ,Y)

Численное значение имтостойкости в рассматриваемом случае можно охарактеризовать величиной:

P(имит)=maxP((a', z'),(a, z))=maxP(z'=F (a',x), z=F(a ,x),

где максимум берется по всем наборам (a', z',a, z), a≠a'.

Величину P(имит) можно преобразовать к виду

P(имит)=maxP(z'=F (a', Y), z=F(a , Y)) = $\max\limits_{a',a'} \max\limits_{z,z}$ P(z'=F (a', Y), z=F(a , Y)), откуда вытекает, что минимально возможное значение величины P(имит) равно $|\Gamma|/|Z|^2|\Gamma| = 1/|z|^2$,причем оно достигается для тех и только тех функций, для которых при любых a,a',a≠a' число решений Ч (F,a≠a',z,z') (относительно Y) системы уравнений

F(a',Y)=z'

F(a,Y)=z

не зависит от выбора значений (z,z') правой части уравнений, то есть Ч (F,a≠a',z,z')=$|\Gamma|/|z|^2$. Действительно,

P(z'=F(a',Y),z=F(a,Y))=Ч(F,a≠a',z,z')/$|\Gamma|$.

$$\sum_{z,z'} Ч(F, a \neq a', z, z')/|\Gamma|,$$

Откуда вытекает, что величина $\max\limits_{z,z'} P(z' = F(a', Y), z = F(a, Y))$ минимальна при Ч (F,a≠a',z,z')=$|\Gamma|/|z|^2$. Этот вывод справедлив для любых пар (a,a').

Предположим, что канал связи используется для передачи шифрованных сообщений с использованием шифра гаммирования. Противник знает передаваемое открытое сообщение $x \in X$, знает шифротекст $y \in У$ и, узнав по этим данным гамму наложения, может заменить открытое сообщение *x* на другое $x' \in X$ путем наложения гаммы на сообщение *x'*.

Для борьбы с таким навязыванием информации используют узлы имитозащиты, функционирование которых описывается конечными автоматами $A = (I \times G, S, I, \delta, \lambda)$ с выходным алфавитом $I \times G$, где I–алфавит открытого текста, G –алфавит гаммы. Выходные последовательности таких автоматов при заданных входной последовательности и начальном состоянии $s \in S$ являются криптограммами. Основное свойство автоматов состоит в возможности «снять имитозащиту», то есть в возможности легко определить открытый текст по известному шифротексту, начальному состоянию, и гамме.

Приведем несколько примеров узлов имитозащиты. Положим $I = G = F_2$ – поле двух элементов, $S = F_2^n, f : F_2^n \to F_2$ – двоичная функция. При $s = (x1, x2, ..., xn), i \in I, g \in G$, где + означает сложение по модулю 2.

$$\delta(i, g, s) = (x2, ..., xn, i + f(s) + g),$$

$$\lambda(i, g, s) = i + f(s).$$

$$\delta(i, g, s) = (x2, ..., xn, i + f(s) + g),$$

$$\lambda(i, g, s) = i + f(s) + g.$$

$$G = F_2^{n-1}, S = F_{2,}$$

$$\delta(i, g, s) = f(g, s) + i,$$
$$\lambda(i, g, s) = f(g, s) + i.$$

Имитозащита, так же как и имитостойкость одно из основополагающих понятий в современной криптографии, значение которого неоспоримо велико. Современные системы зачастую стараются учесть недостатки предыдущих, избегая проблем с имитозащитой и возможного навязывания ложной информации.

Контрольные вопросы:

1) **Что такое имитостойкость?**
2) **Что такое имитозащита?**
3) **Приведите примеры узлов имитозащиты.**
4) **Что такое имитовставка?**

5) **Что такое навязывание информации и как оно осуществляется?**

6) **Приведите возможные варианты навязывания.**

1.23 Криптографическая стойкость

Стойкость криптоалгоритмов должна быть достаточной для того, чтобы обеспечить безопасность данных в течение некоторого срока определяемого условиями эксплуатации информационной системы, использующей эти алгоритмы.

Стойкость количественно должна отражать трудоемкость нарушения информационной безопасности. Наиболее употребительной единицей измерения стойкости является временная сложность наилучшего известного алгоритма, нарушающего безопасность. Однако иногда используются и другие единицы измерения стойкости:

- длительность вычислений по вскрытию ключа (с учетом развития вычислительной модели нарушителя);
- требуемый объем памяти для вскрытия ключа;
- стоимость вскрытия ключа;
- количество энергии, необходимое для вскрытия ключа;
- физический объем вычислительной модели для вскрытия ключа.

Сложностные характеристики этого алгоритма (число шагов, требуемая память, вероятность успеха) могут существенно различаться в зависимости от вида алгоритма и единиц измерения его сложности. Кроме того, понятие наилучшего алгоритма, решающего данную задачу, постоянно меняется. Часто оказывается, что задача нарушения безопасности сводится к совокупности других задач или является эквивалентной некоторым другим задачам. Длительность сохранения конфиденциальности зашифрованной информации определяется совокупностью различных факторов, в том числе:

- производительностью вычислительной модели, которой может располагать нарушитель;
- объемом памяти вычислительной модели;
- скоростью роста во времени производительности вычислительной модели нарушителя при ее совершенствовании;
- сложностью наилучшего известного (вероятностного) алгоритма, решающего задачу нарушения конфиденциальности;

– развитием вычислительных методов математики, которое ведет к появлению новых, более эффективных алгоритмов, решающих задачу нарушения конфиденциальности;

– вероятностью успешного решения задачи данным алгоритмом;

– возможностью получения дополнительной информации об используемом ключе, например, наличие открытых и соответствующих зашифрованных текстов, возможность зашифровывания или расшифровывания специальным образом подобранных текстов.

Стойкость алгоритма аутентификации данных или источника данных характеризуется, с одной стороны, сложностью создания данных нарушителем, которые будут восприняты как истинные, с другой стороны, вероятностью навязывания ложной информации.

В основу безопасности криптографических методов защиты данных положены те или иные математические или физические задачи. Соответствующие криптографические алгоритмы можно классифицировать по степени доказуемости уровня стойкости:

– безусловно стойкие;

– доказуемо стойкие;

– предположительно стойкие.

Безусловно стойкие криптоалгоритмы гарантированно не позволяют вскрыть ключ. Примером такого криптоалгоритма является алгоритм Вернама. Открытый текст **x** и ключ **к** представляются как n–разрядные векторы над полем F2. Шифрование заключается в вычислении поразрядной суммы векторов; y = x Ф к. Ключ представляет собой случайный по Колмогорову вектор такой, что наилучший способ его вычисления – перебор. Ключ может быть использован только один раз. Использование в качестве ключа рекуррентной последовательности, выработанной с помощью известного детерминированного алгоритма, недопустимо.

Безопасность доказуемо стойких криптоалгоритмов определяется сложностью решения хорошо исследованных многими математиками и криптографами и общепризнанно сложных массовых задач, например, задачи разложения числа на множители или задачи вычисления индекса в конечной группе. При этом следует различать случаи полиномиальной сводимости задачи нарушения безопасности криптосистемы к данной хорошо исследованной массовой задаче и полиномиальной эквивалентности этих задач. Примером доказуемо стойкого криптоалгоритма, взлом которого эквивалентен вычислению квадратного корня в кольце классов вычетов по модулю составного числа, является схема подписи Фиата – Шамира примером доказуемо стойкого криптоалгоритма, взлом которого эквивалентен задаче

разложения составного числа на множители, является протокол аутентификации с нулевым разглашением.

Отличительной особенностью доказуемо стойких криптоалгоритмов является их «жесткость», то есть невозможность модификации (усиления) путем незначительных изменений. Поэтому падение сложности задачи разложения приводит к соответствующему снижению стойкости криптографических алгоритмов, основанных на этой задаче.

Изучение доказуемо стойких криптоалгоритмов обычно относится к области криптографии с открытым ключом.

Безопасность предположительно стойких криптоалгоритмов основана на сложности решения частной задачи, которая исследовалась в течение небольшого (по историческим меркам) промежутка времени небольшим числом математиков или криптографов. Взлом предположительно стойкого шифра сводится с полиномиальной сложностью к задаче о выполнимости булевой формулы.

Предположительно стойкие криптоалгоритмы обычно допускают модификацию, то есть обладают свойством гибкости. Например, безопасность шифра DES в значительной степени определяется тем, что используемые подстановки хорошо аппроксимируются линейными над полем F2 булевыми функциями. Использование подстановок с большей нелинейностью позволило бы усилить этот шифр. Примером усиления DES, позволяющим использовать существующие аппаратные решения, является шифр DESX.

Контрольные вопросы:

1) **Что такое криптографическая стойкость?**
2) **Какова классификация стойкости алгоритмов?**
3) **Что такое безусловно стойкий алгоритм?**
4) **Чем безусловно стойкие алгоритмы отличаются от доказуемо стойких?**
5) **Что такое предположительно стойкий алгоритм?**
6) **Приведите пример безусловно стойкого алгоритма.**

1.24 Практическая криптографическая стойкость

Правила криптоанализа были сформулированы еще в конце XIX века преподавателем немецкого языка в Париже голландцем Керкхофом в книге «LaCryptographiemilitare». Согласно одному из этих правил разработчик шифра

должен оценивать криптографические свойства шифра в предположении, что не только шифртекст известен противнику (криптоаналитику противника), но известен и алгоритм шифрования, а секретным для него является лишь ключ.

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

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

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

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

Второй количественной мерой стойкости шифра относительно метода криптоанализа является надежность метода π(φ) – вероятность дешифрования. Раз метод несет в себе определенную случайность, например, не полное

опробование ключей, то и положительный результат метода возможен с некоторой вероятностью. Блестящим примером является метод дешифрования, заключающийся в случайном отгадывании открытого текста. В ряде случаев представляет интерес и средняя доля информации, определяемая с помощью метода. В методах криптоанализа с предварительным определением ключа можно полагать, что средняя доля информации – это произведение вероятности его определения на объем дешифрованной информации.

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

Под количественной мерой криптографической стойкости шифра понимается наилучшая пара $(T(\varphi), \pi(\varphi))$ из всех возможных методов криптографического анализа шифра. Смысл выбора наилучшей пары состоит в том, чтобы выбрать метод с минимизацией трудоемкости и одновременно максимизацией его надежности.

Криптограф, оценивая стойкость шифра, как правило, имитирует атаку на шифр со стороны криптоаналитика противника. Для этого он строит модель действий и возможностей противника, в которой максимально учитываются интеллектуальные, вычислительные, технические, агентурные и другие возможности противника. Примером такого подхода может служить случай в США в конце 70-х годов, Криптографы не нашли практически приемлемого алгоритма дешифрования «DES-алгоритма». Но небольшой размер ключа DES-алгоритма не позволил прогнозировать его практическую стойкость как достаточную на длительный срок, что привело к решению отказаться от использования DES-алгоритма в государственных учреждениях для защиты информации.

Учет интеллектуальных возможностей противника нередко проявляется в постановках задач криптоанализа шифра. В шифрах гаммирования нередко оценивают трудоемкости и надежности методов определения открытого текста по параметрам эффективности методов определения ключа по известной гамме наложения (или, что-то же самое, по известным открытому и шифрованному текстам). Аналогично иногда поступают и с другими поточными шифрами, например, при анализе шифров поточной замены переходят к решению задачи определения ключа по известной управляющей последовательности шифрующего блока. В задачах чтения открытого текста по шифрованному тексту иногда «добавляют» и другой известной информации, облегчающей нахождение решения задачи.

Таким образом, учет интеллектуальных возможностей противника проводится путем постановки и решения «облегченных» задач криптоанализа. При этом полагают, что криптографическая стойкость шифра, вычисленная по таким «облегченным» задачам, не превышает стойкости шифра,

анализируемого в реальных условиях эксплуатации. Нередко в качестве таких задач выделяют задачи, возникающие на промежуточных этапах анализируемого метода криптоанализа.

Примерами таких задач являются разнообразные математические задачи, к которым сводится метод криптоанализа, например, задача решения систем нелинейных уравнений в разнообразных алгебраических структурах, определение начального состояния автомата по его выходной и входной последовательностям, определение входной последовательности автомата по его начальному состоянию и выходной последовательности и др. Нахождение эффективных алгоритмов решения какой-либо из этих математических задач может значительно понизить криптографическую стойкость многих шифров.

Контрольные вопросы:

1) **В чем отличие практической стойкости от теоретической?**
2) **Что такое количественная мера стойкости алгоритма?**
3) **Кто и когда сформулировал правила криптоанализа?**
4) **Что такое трудоемкость криптоанализа?**
5) **Что такое надежность криптоанализа?**
6) **К чему сводится метод криптоанализа?**
7) **Что может существенно снизить криптографическую стойкость большинства шифров?**
8) **Что дает чет интеллектуальных возможностей противника в криптографии?**

Глава 2

ПОТОЧНЫЕ ШИФРЫ

2.1 Классификация поточных шифров

Прежде чем преступить к детальной классификации шифров, необходимо четко понимать, что в любом шифре используется ключ для шифрования, а вот в зависимости от того какой ключ используется для расшифровывания, они будут делиться на симметричные и асимметричные шифры, таким образом что:

- Симметричные шифры используют один и тот же ключ для шифрования и расшифровывания;
- Асимметричные шифры используют разные ключи для шифрования и расшифровывания.

Таким образом, мы будим классифицировать любой шифр по двум составляющим «блочный \ поточный», «симметричный \ асимметричный».

Поточный шифр – это симметричный шифр, в котором каждый символ открытого текста преобразуется в символ шифрованного текста в зависимости не только от используемого ключа, но и от его расположения в потоке открытого текста. Поточный шифр реализует другой подход к симметричному шифрованию, нежели блочные шифры. Допустим, что в режиме гаммирования для поточных шифров при передаче по каналу связи произошло искажение одного знака шифротекста. Очевидно, что в этом случае все знаки, принятые без искажения, будут расшифрованы правильно. Произойдёт потеря лишь одного знака текста. А теперь представим, что один из знаков шифротекста при передаче по каналу связи был потерян. Это приведёт к неправильному расшифровыванию всего текста, следующего за потерянным знаком.

Практически во всех каналах передачи данных для поточных систем шифрования присутствуют помехи. Поэтому для предотвращения потери информации решают проблему синхронизации шифрования и расшифровывания текста. Таким образом, по способу решения этой проблемы шифросистемы подразделяются на синхронные системы и системы с самосинхронизацией. Стоит заметить, что обе системы довольно часто применяются на практике, однако, по словам экспертов, алгоритмы с самосинхронизирующейся системой наиболее просты и надежны в применении. Классификация поточных шифров приведена на рисунке 2.1.

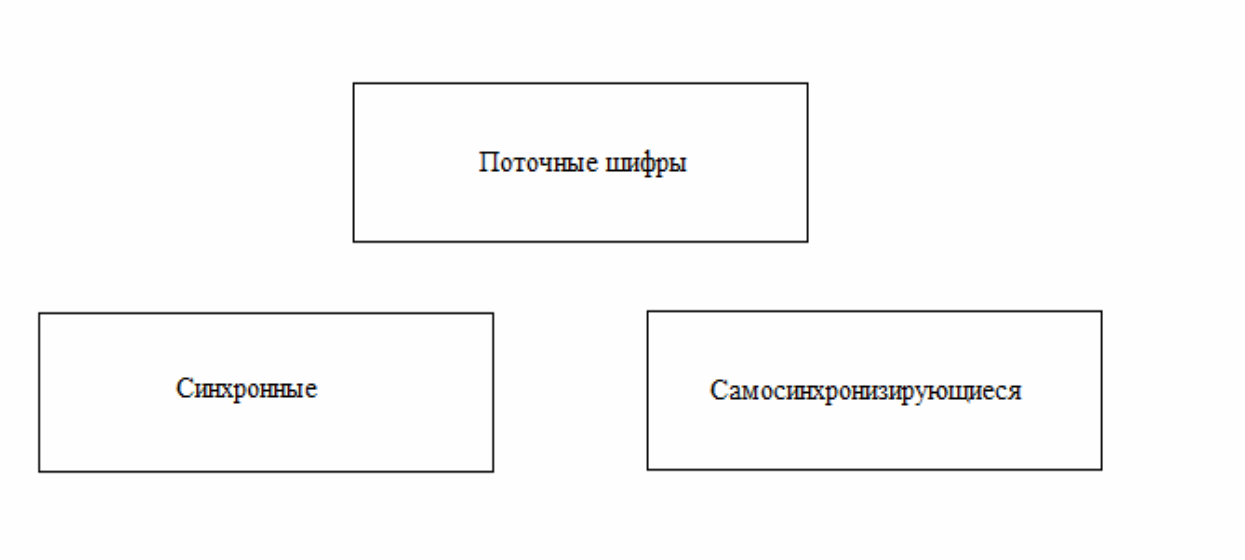

Рисунок 2.1 – Классификация поточных шифров.

Синхронные поточные шифры (СПШ) – шифры, в которых поток ключей генерируется независимо от открытого текста и шифротекста. При шифровании генератор потока ключей выдаёт биты потока ключей, которые идентичны битам потока ключей при дешифровании. Потеря знака шифротекста приведёт к нарушению синхронизации между этими двумя генераторами и невозможности расшифровывания оставшейся части сообщения. Очевидно, что в этой ситуации отправитель и получатель должны повторно синхронизоваться для продолжения работы. Обычно синхронизация производится вставкой в передаваемое сообщение специальных маркеров. В результате этого пропущенный при передаче знак приводит к неверному расшифровыванию лишь до тех пор, пока не будет принят один из маркеров. Необходимо заметить, что выполняться синхронизация должна так, чтобы ни одна часть потока ключей не была повторена.

Плюсы СПШ:

- отсутствие эффекта распространения ошибок (только искажённый бит будет расшифрован неверно);
- предохраняют от любых вставок и удалений шифротекста, так как они приведут к потере синхронизации и будут обнаружены.

Минусы СПШ:

- уязвимы к изменению отдельных бит шифрованного текста. Если злоумышленнику известен открытый текст, он может изменить эти биты так, чтобы они расшифровывались, как ему надо.

Самосинхронизирующиеся поточные шифры (асинхронные поточные шифры (АПШ)) – шифры, в которых поток ключей создаётся функцией ключа и фиксированного числа знаков шифротекста. Итак, внутреннее состояние генератора потока ключей является функцией предыдущих N битов шифротекста. Поэтому расшифровывающий генератор потока ключей, приняв N битов, автоматически синхронизируется с шифрующим генератором. Реализация этого режима происходит следующим образом: каждое сообщение

начинается случайным заголовком длиной N битов; заголовок шифруется, передаётся и расшифровывается; расшифровка является неправильной, зато после этих N бит оба генератора будут синхронизированы.
Плюсы АПШ:

- Размешивание статистики открытого текста. Так как, каждый знак открытого текста влияет на следующий шифротекст, то статистические свойства открытого текста распространяются на весь шифротекст. Следовательно, АПШ может быть более устойчивым к атакам на основе избыточности открытого текста, чем СПШ.

Минусы АПШ:

- распространение ошибки (каждому неправильному биту шифротекста соответствуют N ошибок в открытом тексте);
- чувствительны к вскрытию повторной передачей.

Контрольные вопросы:

1) **Дайте определение поточному шифру.**
2) **Дайте определение ассиметричному шифру.**
3) **Дайте определение синхронных поточных шифров.**
4) **Дайте определение асинхронных поточных шифров.**
5) **Перечислите плюсы и минусы асинхронных поточных шифров.**
6) **Перечислите плюсы и минусы синхронных поточных шифров.**

2.2 Регистр сдвига с линейной обратной связью

Генерация случайных чисел – один из важнейших аспектов, как в криптографии, так и в защите информации в целом. Огромное количество алгоритмов использует такие генераторы, однако мы должны понимать, что любая последовательность случайных чисел полученных через определенные алгоритмы не может являться случайной. Получившиеся числа называют псевдослучайными.

Регистр сдвига с линейной обратной связью (РСЛОС, англ. Linearfeedbackshiftregister, LFSR) – один из методов генерации псевдослучайных чисел. Регистр сдвига с линейной обратной связью состоит из двух частей: собственно регистра сдвига и функции обратной связи. Регистр состоит из битов, его длина – количество этих бит. Когда нужно извлечь бит, все биты регистра сдвигаются вправо на одну позицию. Новый крайний слева бит определяется функцией остальных битов. На выходе регистра оказывается один, обычно младший, значащий бит. Период регистра сдвига – длина получаемой последовательности до начала её повторения. Для РСЛОС функция обратной связи представляет собой сумму по модулю 2 (xor) некоторых битов

регистра, называемых отводами. Регистр сдвига с линейной обратной связью длины состоит из ячеек, пронумерованных, каждая из которых способна хранить бит и имеет один вход и один выход, а также синхросигнал, который контролирует смещение данных. В течение каждой единицы времени выполняются следующие операции (рисунок 2.2):

- содержимое ячейки $L-1$ формирует часть выходной последовательности;
- содержимое i–й ячейки перемещается в ячейку $i+1$ для любого $i, 0 < i < L-1$;
- новое содержимое ячейки 0 определяется битом обратной связи, который вычисляется сложением по модулю 2 с определёнными коэффициентами битов ячеек $0, 1, 2, \ldots, L-1$.

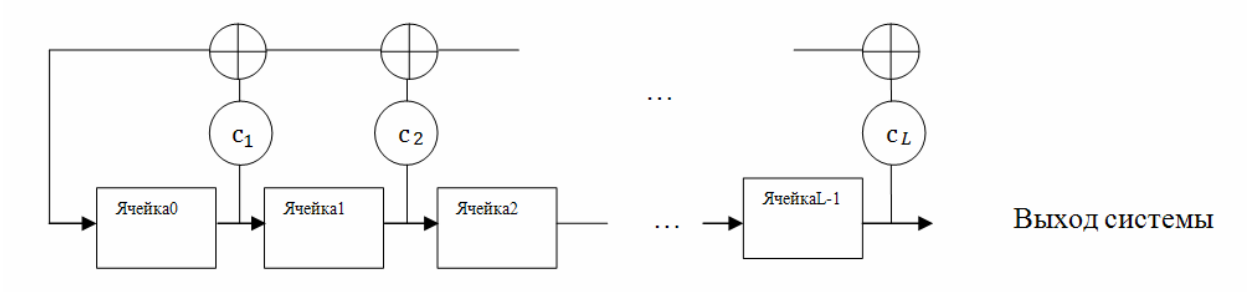

Рисунок 2.2 – Схема операций LFSR.

Таким образом, в качестве функции обратной связи берётся логическая операция XOR (исключающее ИЛИ), то есть:

- на первом шаге: $S_L = (c_1 * S_{L-1}) \oplus (c_2 * S_{L-2}) \oplus \ldots \oplus (c_L * S_0)$
- на втором шаге: $S_{L+1} = (c_1 * S_L) \oplus (c_2 * S_{L-1}) \oplus \ldots \oplus (c_L * S_1)$
- …
- на j – L – 1-м шаге: $S_j = (c_1 * S_{j-1}) \oplus (c_2 * S_{j-2}) \oplus \ldots \oplus (c_L * S_{j-L})$

Свойства:

1. Периодичность
2. Линейная сложность

Свойства выдаваемой РСЛОС последовательности тесно связаны со свойствами ассоциированного многочлена $C(x) = 1 + c_1 x + c_2 x^2 + \ldots + c_L x^L$ над полем $\mathbb{F}_2$ Его ненулевые

коэффициенты называются отводами, как и соответствующие ячейки регистра, поставляющие значения аргументов функции обратной связи.

Периодичность:

Так как существует разных ненулевых состояний регистра, то период последовательности, генерируемой РСЛОС при любом ненулевом начальном состоянии, не превышает. При этом период зависит от ассоциированного многочлена:

- если старший коэффициент ассоциированного многочлена C_L равен нулю, то периодичная часть генерируемой последовательности может проявляться не сразу
- если $C_L = 1$, то соответствующая последовательность называется не особой. Такая последовательность начинается со своей периодичной части.

Наиболее интересны обычные последовательности, соответствующие многочленам со следующими дополнительными свойствами:

- если C(x) неприводим, то при любом ненулевом начальном состоянии регистра период генерируемой последовательности равен наименьшему числу N, при котором многочлен C(x) делит $1 + x^N$. Как следствие, период последовательности будет делить число
- если C(x) примитивен (то есть, является делителем $x^{2^L-1} + 1$, но не является делителем $x^d + 1$ для всех d, делящих), то любое ненулевое начальное состояние регистра дает последовательность с максимально возможным периодом .

Контрольные вопросы:

1) **Дайте определение LFSR.**
2) **Дайте определение псевдослучайной последовательности.**
3) **Назовите свойства LFSR.**
4) **Что подразумевает под собой свойство периодичности?**
5) **Объясните схему операций LFSR.**
6) **Что такое период регистра сдвига?**
7) **Из чего состоит длина регистра сдвига?**

2.3 Линейная сложность

В пункте 2.2 настоящей главы мы рассмотрели регистры сдвига с линейной обратно связью и его свойство периодичность, однако, оставив линейную сложность. Дело в том, что линейная сложность любой бинарной последовательности – одна из самых важных характеристик работы РСЛОС, и для удобства работы с ними нам придется ввести дополнительные обозначения:

- $S = (s_1,\ s_2,\ s_3,\ \ldots)$ – бесконечная последовательность;
- $S_n = (s_1,\ s_2,\ \ldots,\ s_n)$ подпоследовательность длины n последовательности S;
- говорят, что РСЛОС генерирует последовательность S, если существует некоторое исходное состояние, при котором выходная последовательность РСЛОС совпадает S;
- говорят, что РСЛОС генерирует конечную последовательность S_n, если существует некоторое начальное состояние, для которого выходная последовательность РСЛОС имеет в качестве первых n членов члены последовательности S_n.

Тогда линейной сложностью бесконечной двоичной последовательности S называется число L(S), которое определяется следующим образом:

- если $S = (0,\ 0,\ 0, \ldots)$ – нулевая последовательность, то L(S) = 0;
- если не существует РСЛОС, который генерирует S, то $L(S) = \infty$;
- иначе L(S) равна длине самого короткого РСЛОС, который генерирует S.

Таким образом, линейной сложностью конечной двоичной последовательности S_n называется число L(S_n), равное длине самого короткого РСЛОС, который генерирует последовательность, имеющую в качестве первых n членов L(S_n).

Свойства линейной сложности:

Пусть S и K – двоичные последовательности. Тогда:

- для любого n>0 линейная сложность подпоследовательности L(S_n) удовлетворяет неравенствам $0 \le L(S_n) \le n$;
- L(S_n) = 0 тогда и только тогда, когда S_n – нулевая последовательность длины n;
- L(S_n) = n тогда и только тогда, когда $S_n = (0,\ 0,\ \ldots,\ 0,\ 1)$;

– если S периодическая с периодом T, то $L(S_n) \leq T$;

– $L(S \oplus K) \leq L(S) + L(K)$.

Контрольные вопросы:

1) **Дайте определение линейной сложности.**

2) **Назовите свойства линейной сложности.**

2.4 Алгоритм Берлекэмпа-Мэсси

Елвин Ральф Берекэмп (Elwyn Ralph Berlekamp) – выдающийся американский математик и профессор Калифорнийского университета, разработавший алгоритм поиска кратчайшего регистра сдвига с линейной обратной связью, для поданной на вход алгоритма требуемой генерируемой последовательности. Это открытие было обнародовано в 1968 году, а годом позже применение данного алгоритма к линейным кодам было найдено Джеймсом Мэсси. Блок-схема данного алгоритма представлена на рисунке 2.4.

В общем виде для двоичных последовательностей алгоритм примет следующий вид:

– Задать требуемую последовательность битов $s_0, s_1, ..., s_{n-1}$.

– Создать массивы b,t,c длины n, задать начальные значения $b_0 \leftarrow 1, c_0 \leftarrow 1, N \leftarrow 1, L \leftarrow 0, m \leftarrow -1$.

– Пока $N < n$:

1. Вычислить $d \leftarrow s_N \oplus c_1 s_{N-1} \oplus c_2 s_{N-2} \oplus ... \oplus c_L s_{N-L}$
2. Если $d = 0$, то текущая функция генерирует выбранный участок $s_{N-L}, s_{N-L+1}, ..., s_N$последовательности; оставить функцию прежней.
3. Если $d \neq 0$.

– Сохранить копию массива cвt

– $c_{N-m} \leftarrow c_{N-m} \oplus b_0, c_{N-m+1} \leftarrow c_{N-m+1} \oplus b_1, ..., c_{n-1} \leftarrow c_{n-1} \oplus b_{n-N+m-1}$

– Если $2L \leqslant N$, установить значения$L \leftarrow N + 1 - L$, $m \leftarrow N$ и скопировать t в b.

– В результате массив c – функция обратной связи, то есть $c_L s_i \oplus c_{L-1} s_{i+1} \oplus c_{L-2} s_{i+2} \oplus ... \oplus c_0 s_{i+L} = 0$ для любых i.

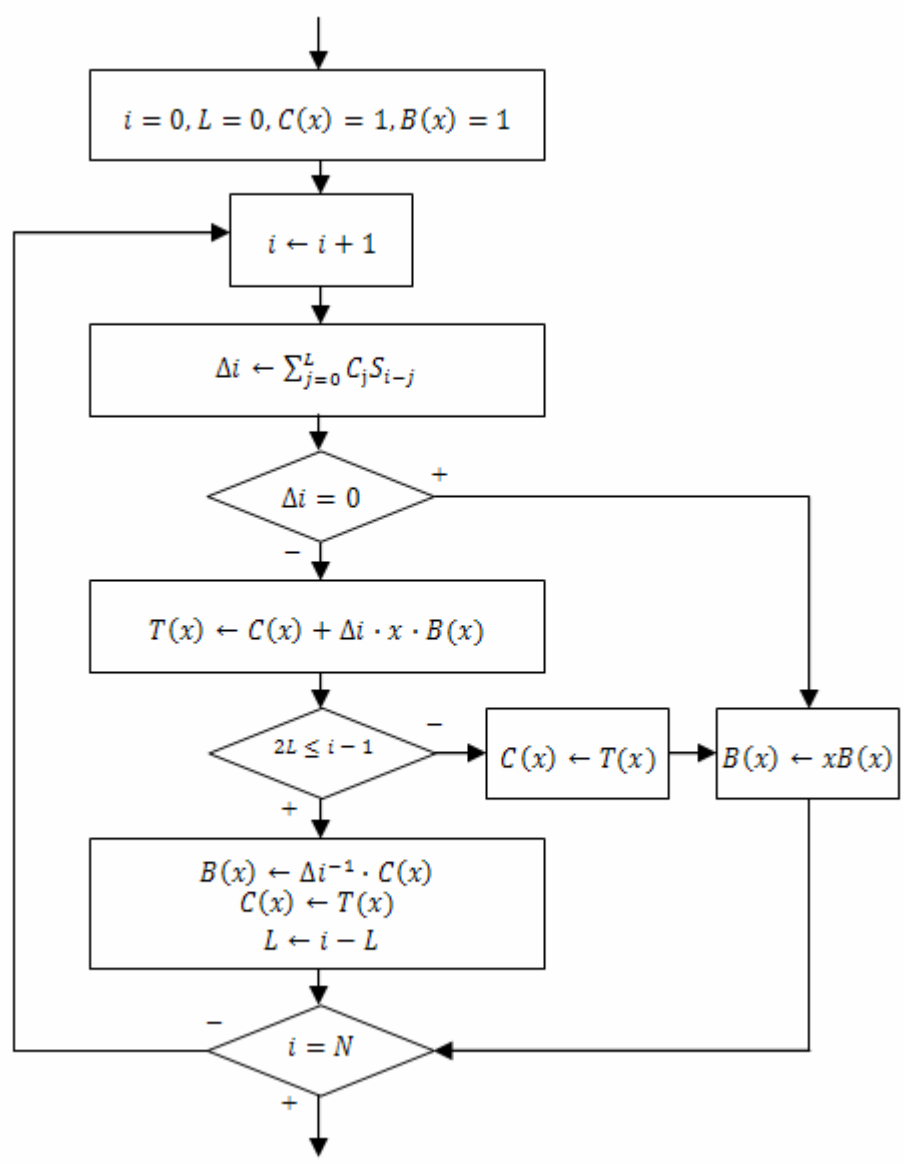

Рисунок 2.4 – Блок – схема алгоритма Берликэмпа-Мэсси.

Контрольные вопросы:

1) **Назовите год создание алгоритма Берликэмпа–Мэсси.**
2) **Объясните блок– схему 2.4 .**

2.5 Метод «одноразовых блокнотов»

С глубокой древности и по настоящее время мы все чаще и чаще сталкиваемся с проблемой защиты окружающей нас информации. Проблема тайной передачи сообщений является одним из приоритетных направлений в развитии существующих систем криптографической защиты информации, большинство из которых, так или иначе, обеспечивают достаточно высокую вероятность сохранения конфиденциальности передаваемой информации.

Однако в сложившихся тенденциях развития вычислительных способностей современных суперкомпьютеров, криптостойкость данных алгоритмов стремительно уменьшается. В связи с этим, становится все более и более актуальным вопрос использования шифра Вернама – единственного алгоритма шифрования для которого доказана абсолютная криптографическая стойкость, что, по сути, означает, что шифр Вернама является самой безопасной криптографической системой из всех существующих.

Шифр Вернама или «схема одноразовых блокнотов» – это система, в которой для шифрования и расшифровывания применяется один и тот же криптографический ключ (симметричное шифрование), была изобретена сотрудниками компании AT&T Мейджором Джозефом Моборном и Нильбертом Вернамом в 1917 году. Для изготовления шифротекста открытый текст объединяется с ключом при помощи операции «исключающее ИЛИ», результат такой операции будет являться истинным только в случае, если является истинным в точности один из аргументов. При этом ключ или «одноразовый шифроблокнот» должен обладать рядом критически важных свойств:

1) Быть истинно случайным
2) Совпадать по размеру с заданным открытым текстом (или превосходить его)
3) Применяться только один раз
4) После использования ключ должен быть уничтожен

В 1949 году американский инженер и математик Клод Шеннон опубликовал работу, в которой смог доказать абсолютную стойкость шифра Вернама. Не трудно догадаться, что требования при реализации такой схемы достаточно сложны, поскольку необходимо обеспечить не только наложение уникальной гаммы, равной длине сообщения, но и гарантированно ее уничтожить. В связи с этим широкого коммерческого применения шифр Вернама не получил, ведь в большинстве случаев цена утраты сведений, составляющих коммерческую тайну, не превышает затрат на использование данного алгоритма, однако в случае особой важности сведении целесообразность применения шифра значительно увеличивается. В основном используется для передачи секретной информации государственными структурами. Область фактического применения достаточно велика, на практике можно один раз физически передать носитель информации с длинным истинно случайным ключом, а затем по мере необходимости пересылать сообщения. Используя идею шифроблокнотов, шифровальщик при личной встрече получает шифроблокнот, каждая из страниц которого содержит ключ, такой же блокнот есть и у принимающей стороны, использованные страницы уничтожаются.

Проанализировав все области и условия применения шифра Вернама , мы выявили следующие существующие проблемы при использовании данной системы:

1) Проблема непригодность псевдослучайных последовательностей.
2) Проблема тайной передачи последовательности.
3) Проблема надежного уничтожения использованных страниц шифроблокнота.
4) Возможность восстановления ключа, при перехвате сообщения посторонними лицами.
5) Проблема чувствительности системы к малейшему нарушению процесса шифрования.

Дело в том, что для работы шифра Вермана необходима истинно случайная последовательность нулей и единиц, однако по определению последовательность полученная с использованием любого алгоритма, является псевдослучайной, а следовательно необходимо использовать неалгоритмические методы получения ключа. Выход из данной ситуации достаточно прост, необходимо попросить помощи у физической составляющей процесса, предмету исследований квантовой криптографии, которая в отличие от традиционной криптографии использует не математические методы обеспечения секретности информации, а рассматривает процесс передачи информации с помощью объектов квантовой механики. Ведь процесс передачи и приема информации всегда выполняется физическими средствами, такими как электроны в электрическом токе или фотоны в линиях оптоволоконной связи. Более того использование квантовой криптографии позволит нам так же избавиться от проблем пункта 2 и 4, приведенных выше.

Одной из основ квантовой криптографии является принцип неопределённости Гейзенберга, который гласит, что для любой квантовой системы невозможно одновременно получить координаты и импульс частицы, невозможно измерить один параметр фотона не исказив другой, а следовательно используя явления квантовой механики, возможно создать некую схему связи, которая всегда может обнаружить прослушку. Это обуславливается тем, что при каждой попытке измерения взаимосвязанных параметров внутри квантовой системы будет вноситься нарушение, разрушающее исходные сигналы, и по уровню шума в канале легитимные пользователи всегда смогут узнать о наличии третьей стороны и распознать степень активности перехватчика.

Рассмотрим простейший пример генерации секретного ключа. Отправляющая сторона (Алиса), используя случайный базис, передает принимающей стороне (Боб) некую последовательность фотонных импульсов,

каждый из которых случайным образом поляризован в одном из четырех направлений, фотоны могут посылаться один за другим или все вместе, главное чтобы Алиса и Боб смогли однозначно установить взаимное соответствие между принятым и отправленным фотоном. В это же время Боб производит измерение принимаемых фотонов в одном из двух произвольно выбранных базисах. При этом в случае использования одинаковых базисов Алиса и Боб получат абсолютно коррелированные результаты, но в случае использования различных базисов результат будет противоположным. В итоге мы получим строку с 25% ошибок, которая называется «первичным ключом». Далее Боб при помощи открытых каналов связи сообщает Алисе об использованном базисе для каждого из фотонов, не оглашая результат, после чего Алиса передает в каких случаях, базисы совпали. Если базисы совпали – бит оставляют, если нет – его попросту игнорируют. В таком случае примерно 50% данных выбрасывается, а оставшийся набор бит образует новый «просеянный» ключ (таблица №1). В том случае если в канале не было шумов и прослушки обе стороны получают одинаковый набор случайных бит, который в дальнейшем и будут использовать для применения шифра Вернама, в противном случае полученные биты уничтожаются, в канале данных устраняются места утечки информации и процесс повторяется снова. Канал не прослушивается с вероятность:

$1 - 2^{-k}$,где k – число сравненных битов.

Таким образом, мы можем исключить проблемы истинно случайных значений, тайной передачи последовательности и ее конфиденциальность. На настоящий момент не существует канала передачи данных, в которых исключена возможность перехвата данных, в противном случае криптография утратила бы свой смысл. Зачастую частными предприятиями осуществляются передачи ключа системы Вернама с помощью других алгоритмов шифрования, таких как RSA, DES, но в таком случае мы получаем настолько, же защищенный шифр, на сколько защищены эти алгоритмы, что как мы уже выяснили не достаточно надежно. Проблема надежного уничтожения использованных страниц шифроблокнотов, а также других реализованных физических носителей информации является чисто организационной проблемой и может быть устранена лишь ужесточением контроля за уничтожением столь ценных данных.

Одно из основных достоинств шифра Вернама, также является и его главным недостатком – это чувствительность к любому нарушению процедур шифрования. В истории известны случаи, когда из перехваченной агентами АНБ США в 40-х годах прошлого века, были обнаружены сообщения, которые дважды закрыли при помощи одной и той же гаммы. И хотя период этот длился не долго (об успехах американских криптоаналитиков в спецслужбах СССР довольно быстро узнали о серьезных проблемах с надежностью в своей шифрпереписке), такие сообщения были расшифрованы в рамках секретного проекта «Venona», документы которого были не так давно рассекречены и

выложены на всеобщее обозрение на сайте АНБ. Таким образом, мы проанализировали проблемы практического применения шифра Вернама, а также попытались предложить возможные пути их устранения, что в последующем приведет к развитию криптографии в целом. Несмотря на все недостатки и сложности в использовании шифра Вернама, эта схема одноразового блокнота является единственной системой с доказанной абсолютной криптографической стойкостью. А исходя из этого перед вами, как перед специалистами по защите информации, стоит непростой выбор, между системами отличающимися простотой, доступностью, вероятной стойкостью и системой, стойкость (а значит и надежность) которой не вызывает сомнений.

Последовательность фотонов Алисы	|	/	/	–	\	|	|	–	–
Последовательность анализаторов Боба	+	X	+	+	X	X	X	+	+
«Первичный» ключ	0	0	1	1	1	0	1	1	0
Верные анализаторы	+	+		+	+			+	
Ключ	0	0		1	1			1	

Таблица 2.5 – Передача и анализ истинно случайного ключа

Контрольные вопросы:

1) **Назовите год создания метода «одноразовых блокнотов».**
2) **Назовите основные проблемы использования метода Вернама.**
3) **Назовите основные критерии работы шифра.**
4) **Перечислите доказанные, абсолютно стойкие алгоритмы шифрования.**
5) **Перечислите пути решений проблем использования метода Вернама.**
6) **Можем ли мы использовать случайную газету, достаточно большого объема в качестве ключа к шифру Вернама? Почему?**
7) **Объясните таблицу 2.5 .**

2.6 Нелинейные регистры сдвига с обратной связью

Рассмотрев линейные регистры сдвига с обратной связью, мы вплотную подошли к рассмотрению нелинейных регистров. Казалось бы, зачем нам нужны нелинейные сдвиги? Ведь с нелинейностью связано огромное количество проблем, да и математические преобразования с ними становится на порядок сложнее? Дело в том, что линейные сдвиги хороши, и просты в применении, но в чистом виде для криптографии мало пригодны. Существуют достаточно простые алгоритмы, которые способны определить устройство LFSR по данным длинной всего 2L, что делает его крайне уязвимым. Выход из такой ситуации достаточно прост, нам всего-то нужно ввести где-то

нелинейность. Простейший пример нелинейного сдвига приведен на рисунке 2.6.1. Можно поступить еще проще, достаточно простого объединения нескольких LFSR с простейшей нелинейной функцией и на выходе мы получим нелинейный регистр сдвига, представленный на рисунке 2.6.2. Исходя из этих соображений, нелинейные регистры сдвига зачастую строятся на основе линейных, с добавлением какого-либо нелинейного элемента: логического сложения или логического умножения.

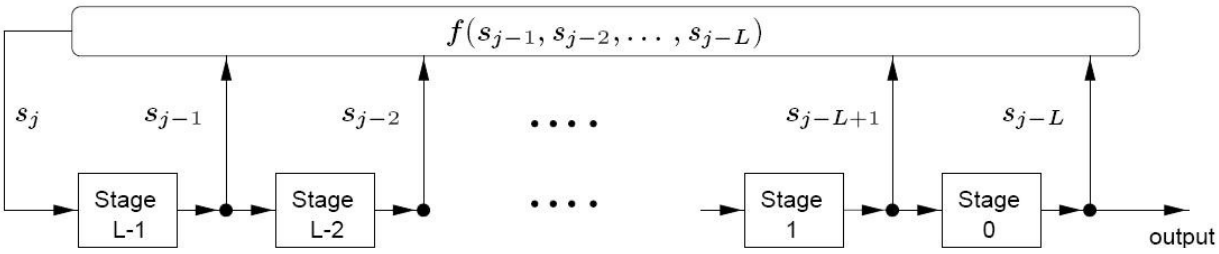

Рисунок 2.6.1 – Пример нелинейного сдвига.

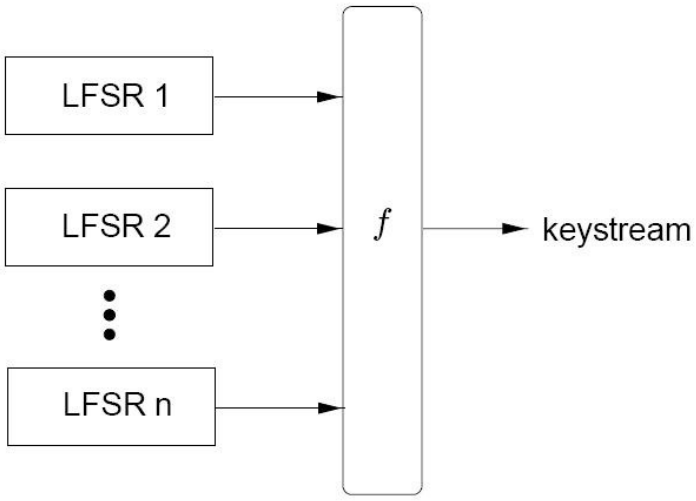

Рисунок 2.6.2 – Объединение нескольких LFSR нелинейной функцией.

Существует несколько классов нелинейных сдвигов, но наиболее актуальными являются следующие:

- фильтрующие;
- комбинирующие;
- динамические.

Фильтрующими нелинейными регистрами сдвига – называются сдвиги, использующие в своей основе дополнительную комбинационную схему – фильтр, на выходах некоторых бит LFSR (рисунок 2.6.3), выход комбинационной схемы и будет являться гаммой.

Комбинирующими нелинейными регистрами сдвига – также называют сдвиги, использующие в своей основе комбинационную схему с нелинейными

преобразованиями бит, но на выходе этой комбинационной схемы подаются выходы нескольких линейных регистров сдвига (рисунок 2.6.4).

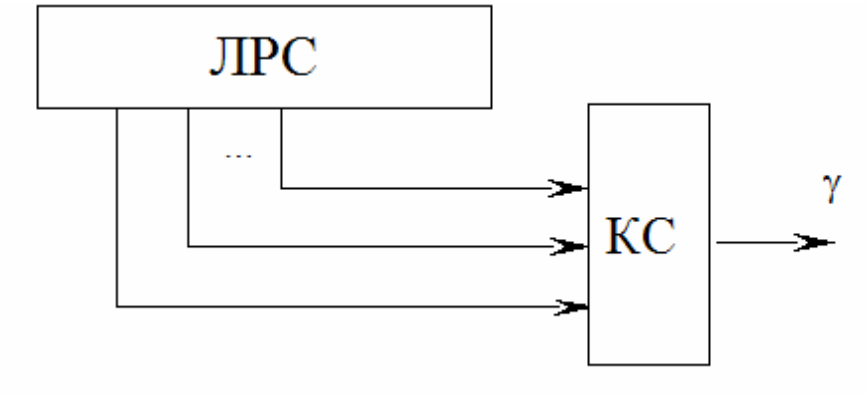

Рисунок 2.6.3 – Фильтрующий нелинейный регистр сдвига.

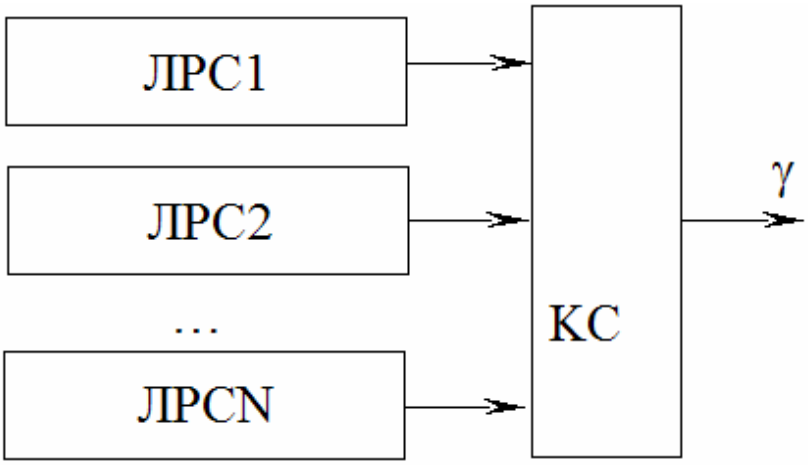

Рисунок 2.6.4 – Комбинирующий нелинейный регистр сдвига.

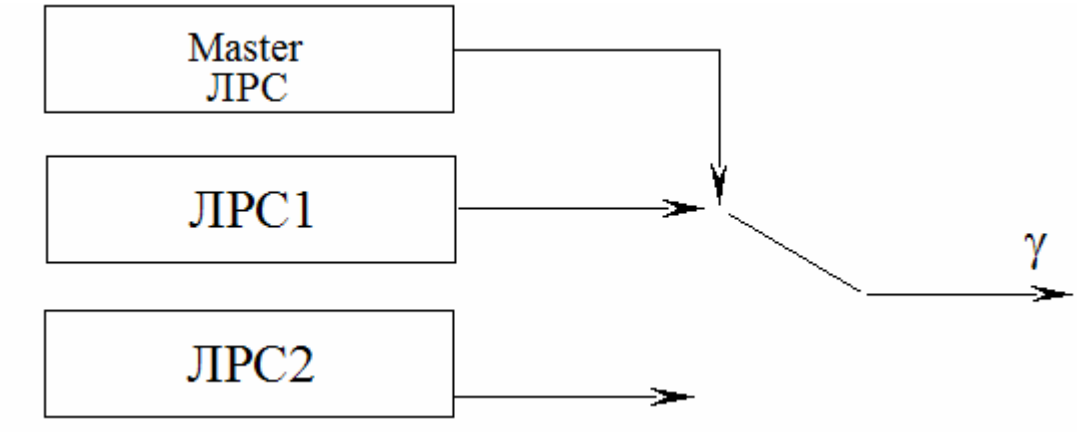

Рисунок 2.6.5 – Динамический нелинейный регистр сдвига.

Динамическими нелинейными регистрами сдвига – называются сдвиги, постоянные на основе нескольких линейных сдвигов, но на основе архитектуры отношений «Master – Slave». В зависимости от выхода Master ЛСР на общий

выход нелинейного регистра сдвига подается либо выход первого ЛРС, либо второго (рисунок 2.6.5).

Таким образом, нелинейные регистры сдвига вводятся из-за основной проблемы ЛСР – уязвимости к атакам на основе известного открытого текста. Даже при неизвестной внутренней структуре ЛСР, криптоанализ на основе алгоритма Бэрликампа-Мэсси по известным 2L битам открытого текста и соответствующего щифротекста, приведет нас к возможности построить ЛСР, порождавшую подобную последовательность, что является просто недопустимым и ведет к неминуемому взлому шифруемой информации.

Контрольные вопросы:

1) **Дайте определение нелинейным регистрам сдвига.**
2) **Перечислите основные типы НРС.**
3) **Дайте определение динамическим НРС.**
4) **Дайте определение комбинирующим НРС.**
5) **Дайте определение фильтрующим НРС.**
6) **Укажите причину введения нелинейности.**
7) **Объясните рисунки 2.6.1 – 2.6.5 .**
8) **Назовите минимально необходимую длину бит открытого текста, при которой возможно осуществить атаку на ЛСР.**

2.7 Нелинейная комбинация генераторов

Рассмотрев в предыдущем пункте основные комбинации и причины введения нелинейных сдвигов, настало время поговорить более подробно о нелинейной комбинации генераторов. Как мы уже выяснили достаточно использовать простейшие методы, которые ломают линейные свойства ЛСР и тем самым повышают стойкость наших криптосистем. В качестве примера мы можем взять классический генератор ключевого потока на основе НСР – генератор Геффа, схема которого представлена на рисунке 2.7.1. В этом генераторе используется три РСЛОС объединенных нелинейным образом. Пусть длина этих регистров R_1, R_2,R_3попарно простые числа, тогда нелинейную функцию для данного генератора можно записать так: $f(x_1, x_2, x_3) = x_1x_2 \oplus (1 + x_2)x_3 = x_1x_2 \oplus x_2x_3 \oplus x_3$. Тогда мы получаем следующую линейную сложность: R = R_1xR_2 + R_2xR_3 + R_3. Несмотря на достаточно высокую линейную сложность, данный генератор криптографически слаб, потому что информация о состоянии его генераторов содержится в выходном сигнале.

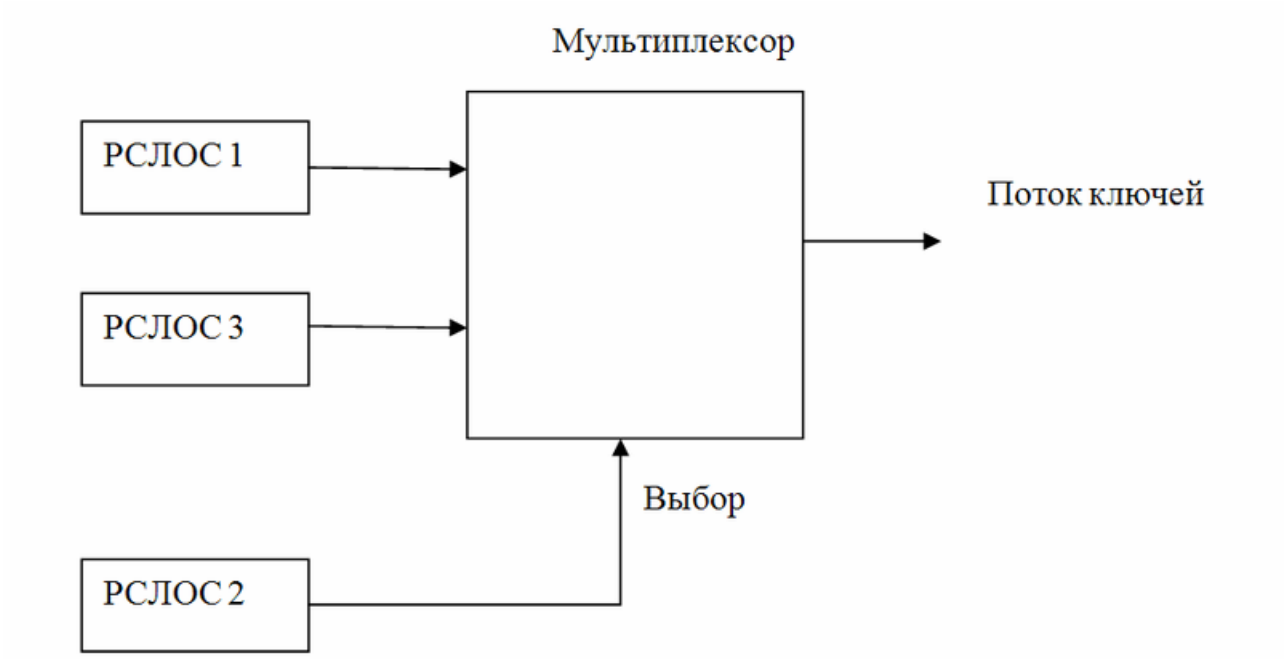

Рисунок 2.7.1 – Генератор Геффе.

Наряду с генератором Геффе, достаточно простым и часто используемым генератором является генератор переменного шага, основная идея которого заключается в управлении РСЛОС 1 (Master) передвижением битов двух других РСЛОС 2 и 3(Slave). Подобная конструкция довольно часто называется «часами». Алгоритм работы таких «часов» достаточно тривиален и представлен на рисунке 2.7.2.

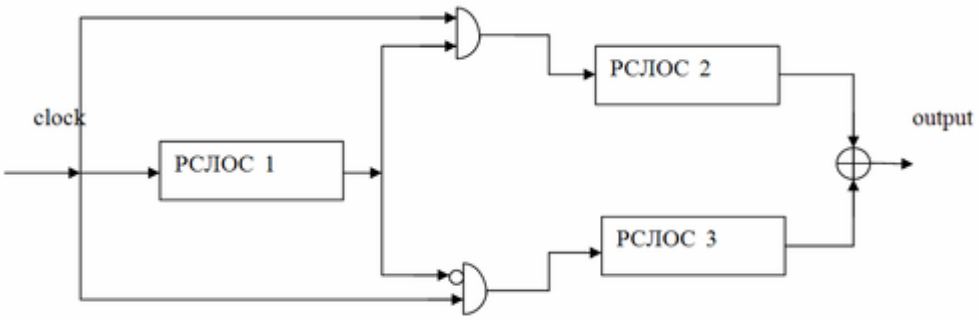

Рисунок 2.7.2 – Генератор переменного шага.

Схожим по структуре с генератором переменного шага, но не менее надежным является сжимающий генератор, его схема представлена на рисунке 2.7.3. Алгоритм несколько отличается и основан на том, что контролирующий регистр РСЛОС 1 используется для управления выходом РСЛОС 2, таким образом, что:

- регистры РСЛОС 1 и 2 синхронизированы общим сигналом;
- при равенстве единицы выходного бита РСЛОС 1, выход генератора формируется битом регистра РСЛОС 2;

– при равенстве нулю выходного бита РСЛОС 1, выходной бит регистра РСЛОС 2 попросту отбрасывается.

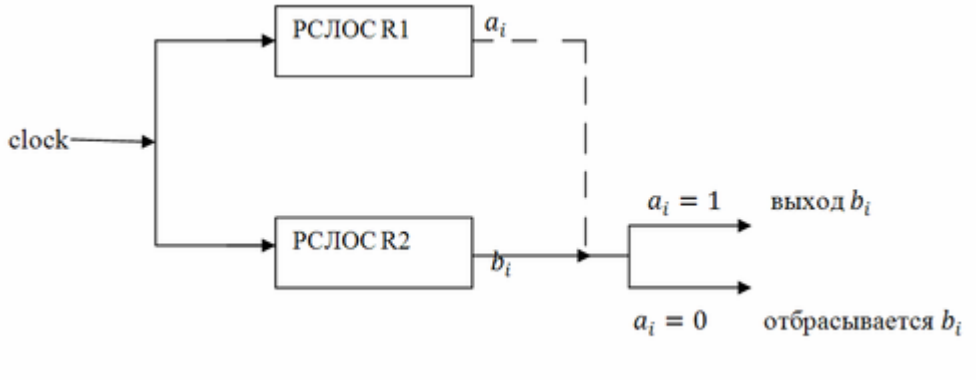

Рисунок 2.7.3 – Сжимающий генератор.

Сжимающий генератор прост, масштабирует и имеет хорошие защитные свойства, основной его недостаток заключается в скорости генерации ключа. Она не будет постоянной без применения дополнительных мер, что зачастую приводит к небольшим затруднениям. Для увеличения безопасности сжимающего генератора необходимо:

– чтобы длины обоих регистров были взаимно простыми числами;

– чтобы использовалось скрытое соединение между регистрами.

Контрольные вопросы:

1) **Опишите работу сжимающего генератора.**
2) **Опишите работу генератора переменного шага.**
3) **В чем отличие работы генератора переменного шага от сжимающего генератора?**
4) **К какому классу НРС относится сжимающий генератор?**
5) **К какому классу НРС относится генератор переменного шага?**
6) **Какие меры необходимо предпринять для увеличения безопасности сжимающего генератора?**
7) **Что произойдет со сжимающим генератором, если выходной бит РСЛОС 1 будет равен 1?**
8) **Что произойдет со сжимающим генератором, если выходной бит РСЛОС 1 будет равен 0?**

2.8 Алгоритм SEAL

Говоря о HCP, пожалуй, будет довольно уместно упомянуть о, пожалуй, самом последнем весьма и весьма удачном программном применении данного метода – алгоритме S.E.A.L. (Software – Optimized Encryption Algorithm).

Данный алгоритм является симметричным поточным алгоритмом шифрования данных, специально оптимизированным для программной реализации. Он был разработан в 1993 году в IBM Филом Рогэвеем и Доном Копперсмитом, после чего был оптимизирован под процессоры с 32-битной архитектурой. Скорость шифрования была равна 4 машинным тактам на 1 байт текста. Для шифрования и расшифровывания применялся 160-битный ключ, а во избежание нежелательной потери скорости из-за медленных операций обработки ключа, алгоритм предварительно выполняет несколько преобразований с ним, получая 3 таблицы определенного размера. По сути, для шифрования и расшифровывания текста алгоритм использовал не сам ключ, а полученные из него таблицы. Данный алгоритм считается весьма надежным и довольно быстрым. Схема его реализации представлена на рисунке 2.8.

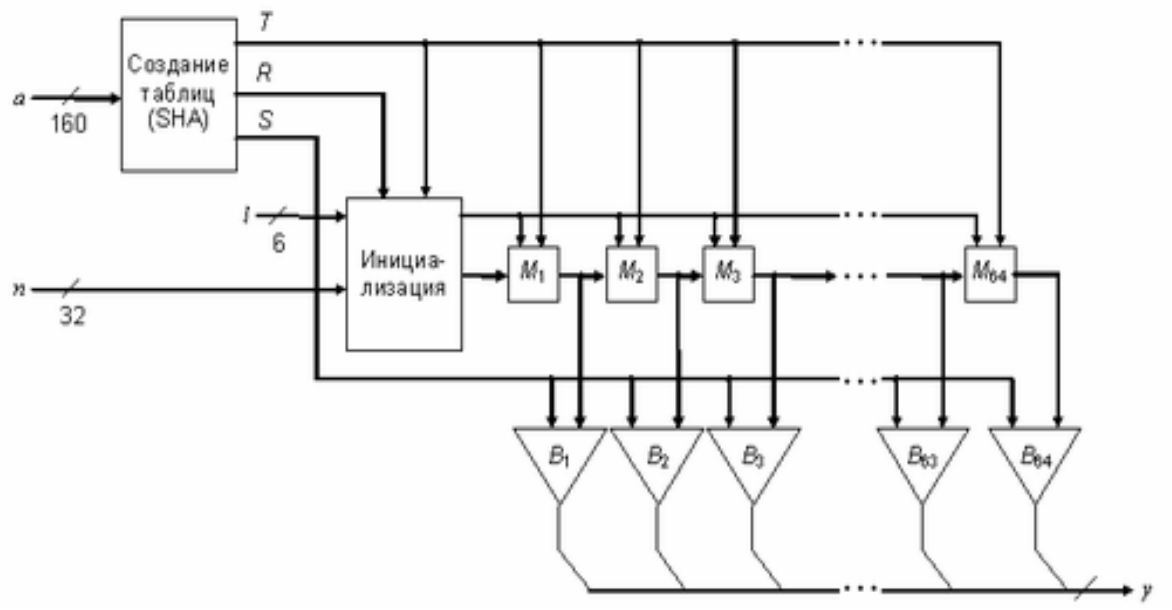

Рисунок 2.8 – Схема алгоритма SEAL.

Алгоритм обладает рядом основополагающих идей и свойств:

- использование таблиц получаемых из ключа в 160-бит;
- чередование простых арифметических и логических операций;
- использования состояния системы, которое явно не проявляется в потоке данных;

– использование различных друг от друга операций в зависимости от этапа итерации и ее номера.

Достаточно любопытным остается тот факт, что, несмотря на довольно приличный возраст, и морально устаревшую технику, на которой применялся данный алгоритм, он является одним из самых быстрых шифров мира.

Контрольные вопросы:

1) **Объясните рисунок 2.8.**
2) **Чем отличается алгоритм SEAL от других поточных шифров?**
3) **Назовите количество машинных тактов необходимое для шифрования и расшифровывания одного бита текста при использовании алгоритма SEAL.**
4) **Назовите год создания алгоритма SEAL.**
5) **Назовите количество таблиц получаемых при разбиении ключа в алгоритме SEAL.**
6) **К чему приведет использование 256 битного ключа в алгоритме SEAL?**

2.9 Линейное и предварительное шифрование

Под линейным шифрованием понимают способ шифрования, при котором процессы шифрования и расшифровывания производятся непосредственно в процессе передачи (приема) информации по каналу связи. Примером подобного шифрования является шифрование речевых данных в реальном времени. В данном случае шифрование происходит в одно и то же время, просто потому, что в противном случае связь прерваться и общение уже будет с задержками, что недопустимо. При этом обеспечение защиты информации указанным методом не должно нарушать работу сети в реальном масштабе времени, что возможно при выполнении шифрования со скоростью до 1 Гбит/с и выше. Дело в том, что существует два радикально отличающихся метода шифрования основанные на модели OSI. В теории мы можем использовать шифрование на любом из семи уровней данной модели, однако на практике зачастую используются либо самые верхние уровни, либо самые нижние. Таким образом, если данные шифруются на нижних уровнях модели, то такое шифрование называется канальным, а если на верхних, то сквозным.

По сути оба эти методы являются равноценными по определению, хоть и с небольшими оговорками по некоторым алгоритмам.

При канальном шифровании все данные без исключения будут шифроваться по каждому из каналов связи, включая открытый текст

сообщения, а также ту информацию, на основе которой происходит маршрутизация, включая информацию о коммутационном протоколе. При таком стечении обстоятельств у криптоаналитика нет даже простейшей информации, он попросту остается слеп, не зная даже источника этих данных, не говоря уже, о том кому они предназначены и какова их структура. А при зашумлении пустого канала случайным потоком бит-последовательностей, криптоаналитик даже не в состоянии ответить на вопрос «Где начинается и где заканчивается текст передаваемого сообщения?», что в свою очередь практически полностью отбрасывает возможность эффективной криптоатаки.

Пожалуй, самым большим недостатком такого подхода является тот факт, что данные приходится шифровать при передаче по каждому из физических каналов компьютерной сети, а отправка незашифрованного сообщения по каналу может поставить под угрозу безопасность всей сети, поэтому в крупных сетях такое шифрование вливается в значительные финансовые затраты. Кроме того нам потребуется защитить все каналы передачи данных, что приведет к еще большему увеличению затрат.

Альтернативой канальному шифрованию является сквозное (End-to-end шифрование) шифрование, происходящее, как мы уже говорили, на верхних уровнях модели OSI. При таком методе шифрованию подлежат лишь содержательные части сообщения, передаваемые по сети. После шифрования к ней добавляется определенная служебная информация, необходимая для маршрутизации сообщения, а результат переправляется на более низкие уровни модели с целью отправки адресату. Таким образом, нам больше не требуется постоянно расшифровывать и зашифровывать весь наш трафик на всем его пути, ведь сообщение остается зашифрованным до самого пункта назначения.

Основная проблема, с которой мы можем столкнуться это тот факт, что служебная информация, передаваемая вместе с сообщением, идет в незашифрованном виде. Что существенно увеличивает осведомленность криптоаналитика о нашем сообщении. А в зависимости от типов сетей и интерфейсах мы будем вынуждены использовать различные ключи шифрования, что также увеличивает количество затруднений.

В настоящий момент довольно активно используется комбинированное шифрование, объединяющее сквозное и канальное шифрование. Такой метод намного затратнее, но и надежнее.

Контрольные вопросы:

1) **Назовите уровни модели OSI.**
2) **Назовите отличие канального шифрования от сквозного.**
3) **Назовите сложности использования канального и сквозного шифрования.**

4) **Назовите преимущества использования канального и сквозного шифрования.**
5) **Дайте определение канальному и сквозному шифрованию.**
6) **Дайте определение линейному шифрованию.**

2.10 Методы получения случайных и псевдослучайных чисел

Как мы уже говорили в предыдущих пунктах, получение случайных чисел это один из самых важных процессов в криптографии и защите информации. Существует определенная сложность в получении настоящих случайных чисел. Источники настоящих случайных чисел найти трудно. Физические шумы, такие как детекторы событий ионизирующей радиации, дробовой шум в резисторе или космическое излучение могут быть такими источниками. Однако применяются такие устройства в приложениях сетевой безопасности редко. Сложности также вызывают грубые атаки на подобные устройства.

Альтернативным решением является создание некоторого набора из большого количества случайных чисел и опубликование его в некотором словаре. Тем не менее, и такие наборы обеспечивают очень ограниченный источник чисел по сравнению с тем количеством, которое требуется приложениям сетевой безопасности. Хотя данные наборы действительно обеспечивают статистическую случайность, они не достаточно случайны, так как противник может получить копию словаря. Криптографические приложения используют для генерации случайных чисел особенные алгоритмы. Эти алгоритмы заранее определены и, следовательно, генерируют последовательность чисел, которая теоретически не может быть статистически случайной. В то же время, если выбрать хороший алгоритм, полученная численная последовательность будет проходить большинство тестов на случайность. Такие числа называют псевдослучайными числами.

Генератор псевдослучайных чисел (ГПСЧ, англ. Pseudo random number generator, PRNG) – алгоритм, генерирующий последовательность чисел, элементы которой почти независимы друг от друга и подчиняются заданному распределению. Зачастую используется равномерное распределение.

Современная информатика широко использует псевдослучайные числа в самых разных приложениях – от метода Монте-Карло и имитационного моделирования до криптографии. При этом от качества используемых ГПСЧ напрямую зависит качество получаемых результатов. Таким образом, мы просто не можем оставлять без внимание столь щепетильный момент.

Существует огромное количество генераторов псевдослучайных чисел, мы познакомимся лишь с основными из них.

Детерминированные генераторы простых случайных чисел достаточно распространены, хоть никакой детерминированный алгоритм и не может генерировать полностью случайные числа, он может только аппроксимировать некоторые их свойства. Любой ГПСЧ с ограниченными ресурсами рано или поздно зацикливается – начинает повторять одну и ту же последовательность чисел. Длина циклов ГПСЧ зависит от самого генератора и составляет около 2n/2, где n – размер внутреннего состояния в битах, хотя линейные конгруэнтные и LFSR–генераторы обладают максимальными циклами порядка 2n. Если порождаемая ГПСЧ последовательность сходится к слишком коротким циклам, то такой ГПСЧ становится предсказуемым и непригодным для практических приложений. Большинство простых арифметических генераторов хотя и обладают большой скоростью, но страдают от многих серьёзных недостатков:

- слишком короткий период\периоды;
- последовательные значения не являются независимыми;
- некоторые биты повторяются чаще, чем другие;
- неравномерное распределение;
- обратимость.

В частности, алгоритм RANDU, десятилетиями использовавшийся на мейнфреймах, оказался ужасно ненадежным, что вызвало сомнения в достоверности результатов многих исследований, использовавших этот алгоритм. Наиболее распространены линейный конгруэнтный метод, метод Фибоначчи с запаздываниями, регистр сдвига с линейной обратной связью, Generalized feed backshift register. Из современных ГПСЧ широкое распространение также получил «вихрь Мерсенна», предложенный в 1997 году Мацумото и Нисимурой. Его достоинствами являются колоссальный период, равномерное распределение в 623 измерениях (линейный конгруэнтный метод даёт более или менее равномерное распределение максимум в 5 измерениях), быстрая генерация случайных чисел (в 2-3 раза быстрее, чем стандартные ГПСЧ, использующие линейный конгруэнтный метод). Однако, существуют алгоритмы, распознающие последовательность, порождаемую вихрем Мерсенна, как неслучайную, что в очередной раз доказывает невозможность конкуренции псевдослучайных чисел с истинно случайными.

Наравне с существующей необходимостью генерировать легко воспроизводимые последовательности случайных чисел, также существует необходимость генерировать совершенно непредсказуемые или попросту абсолютно случайные числа. Такие генераторы называются генераторами

случайных чисел (ГСЧ – англ. Random number generator, RNG). Так как такие генераторы чаще всего применяются для генерации уникальных симметричных и асимметричных ключей для шифрования, они чаще всего строятся из комбинации криптостойкого ГПСЧ и внешнего источника энтропии (и именно такую комбинацию теперь и принято понимать под ГСЧ). Почти все крупные производители микрочипов поставляют аппаратные ГСЧ с различными источниками энтропии, используя различные методы для их очистки от неизбежной предсказуемости. Однако на данный момент скорость сбора случайных чисел всеми существующими микрочипами (несколько тысяч бит в секунду) не соответствует быстродействию современных процессоров. В персональных компьютерах авторы программных ГСЧ используют гораздо более быстрые источники энтропии, такие, как шум звуковой карты или счётчик тактов процессора. Сбор энтропии являлся наиболее уязвимым местом ГСЧ. Эта проблема до сих пор полностью не разрешена во многих устройствах (например, смарт-картах), которые таким образом остаются уязвимыми. Многие ГСЧ используют традиционные испытанные, хотя и медленные, методы сбора энтропии вроде измерения реакции пользователя (движение мыши и т. п.), как, например, в PGP и Yarrow, или взаимодействия между потоками, как, например, в Javasecurerandom.

Аппаратный генератор случайных чисел – устройство, которое генерирует последовательности случайных чисел на основе измеряемых параметров протекающего физического процесса. Работа таких устройств часто основана на процессах уровня элементарных частиц, таких как тепловой шум, фотоэлектрический эффект, другие квантовые явления. Эти процессы, в теории, абсолютно непредсказуемы. Аппаратные генераторы случайных чисел, основанные на квантовых процессах, обычно состоят из специального усилителя и преобразователя. Усилитель усиливает очень слабые сигналы, получаемые в результате проходящих физических явлений, до приемлемых размеров, которые преобразуются преобразователем к цифровому виду. Аппаратные генераторы случайных чисел относительно медленны и могут производить смещенные последовательности (когда определенная последовательность чисел повторяется чаще). Использование подобных генераторов зависит от потребностей конкретной предметной области и от устройства самого генератора.

Существует огромное количество генераторов случайных чисел, каждый и которых имеет свои плюсы и минусы, однако с основной задачей большинство из них справляется. Зачастую большинство алгоритмов использует псевдослучайные числа определенной степени случайности, которой будет вполне достаточно для обеспечение приемлемого уровня криптостойкости. Однако для истинных «гурманов» существует специальный раздел – квантовая криптография (о которой поговорим ниже), способной черпать свои силы из физической составляющей информационных процессов,

что позволяет ей удовлетворять потребности в случайных числах даже такие шифры как шифр Вернама.

Контрольные вопросы:

1) **Назовите основные недостатки простых генераторов случайных чисел.**
2) **Назовите основные виды генераторов псевдослучайных чисел.**
3) **Что такое генератор случайных чисел.**
4) **Назовите отличие псевдослучайных чисел от истинно случайных.**
5) **Охарактеризуйте аппаратный генератор случайных чисел.**
6) **Чем отличается детерминированные ГПСЧ от ГПСЧ с источником энтропии?**

2.11 Анализ генераторов псевдослучайных чисел

Создание генераторов псевдослучайных чисел достаточно распространенное явление, однако как нам определить степень пригодности и случайности полученных результатов? В математической статистике существует целый ряд тестов, которые называются критериями согласия для проверки функции распределения случайной величины на предмет ее соответствия теоретически ожидаемому закону распределения. Таких критериев существует достаточно много, вот основные из них:

- критерий Пирсона;
- критерий Колмогорова-Смирнова;
- критерий серий.

Для любой изучаемой нами случайной величины можно сделать два предположения о том, что наша величина имеет равномерное распределение(нулевая гипотеза) и противоположное предположение(альтернативная гипотеза), что наша величина не имеет равномерное распределения. Дело в том, что все приведенные выше критерии являются статистическими, и в состоянии лишь установить отличие теоретического от экспериментального распределений, поэтому нулевая гипотеза выдвигается только для проверки на наличие оснований для ее отброса. Таким образом мы видим, что невозможно доказать абсолютную случайность последовательности, но можно определить степень вероятности опровергнуть противоположное предположение. Поэтому, для решения является ли различие достоверным необходимо установить границы для

близости, то есть различие частот в выборке и теоретически ожидаемых частот. Данная величина называется уровнем значимости, и обычно принимает значения 5%, 1%, 0.1%. Результат называется значимым на уровне 5%, если правильная нулевая гипотеза будет отклонена не более, чем в 5% случаев.

Мы не будем вдаваться в детали математических изысканий, но отметим, что чем больше тестов мы проведем, тем больше мы будем уверены в достаточности случайного распределения, а значит и в степени случайности нашей сгенерированной последовательности.

Контрольные вопросы:

1) **Что такое нулевая гипотеза?**
2) **Перечислите основные математические критерии для определения случайности последовательности.**
3) **В каких приделах колеблется уровень значимости для лучших ГПСЧ?**
4) **Назовите достаточный уровень значимости для метода «одноразовых блокнотов».**
5) **Что будет означать выражение «Уровень значимости для данного распределения равен нулю»?**

2.12 Гаммирование. Шифр RC 4

Одним из основополагающих понятий в криптографии является гаммирование.

Гаммирование – метод шифрования, основанный на «наложении» гамма–последовательности на открытый текст. Обычно это суммирование в каком–либо конечном поле, зачастую используются логические функции. При расшифровывании такая операция производится повторно, что даст в результате исходный открытый текст. Данный вид шифра был известен достаточно давно, и широко применяется и по сей день. В типичном шифре гаммирования вместо случайной гаммы наложения используется псевдослучайная последовательность, напрямую зависящая от ключа. Выбор схемы шифратора, как правило, ориентирован на элементарную базу, при помощи которой в последствии алгоритм и будет реализован. Примером современного поточного шифра гаммирования является шифр RC4. Он оптимизирован под программное исполнение и прекрасно зарекомендовал себя в программной защите информации.

По сути, шифр RC 4 представляет из себя целое семейство алгоритмов, задаваемых параметром k, которой является положительным целым числом с

рекомендованным значением k=8. Внутреннее состояние генератора RC 4 в момент времени t состоит из таблицы $S_t = (S_t(l))_{l=0}^{2^k-1}$, содержащей k-битных слов и из двух k-битных слов указателей i_j и j_t . От сюда получаем размер внутренней памяти равный M = k+ 2k бит. Предположим, что выходное k-битное слово генератора на момент t будет z_t, тогда начальное значение слов указателей будет равно 0, а следовательно состояние и функция выхода RC 4 для каждого t ≥ 1 задается следующими соотношениями:

- $i_t = i_{t-1} + 1$;
- $j_t = j_{t-1} + S_{t-1}(i_t)$;
- $S_t(i_t) = S_{t-1}(i_t)$, $S_t(j_t) = S_{t-1}(i_t)$;
- $z_t = S_t(S_t(i_t) + S_t(j_t))$.

Все сложения должны выполняться по модулю. Таким образом, мы получаем, что все слова кроме подвергаемых изменению по формулам, остаются теми же самыми, а выходная последовательность k-битных слов обозначается как $Z = (z_t)_{t=1}^{n}$. Начальная таблицаS_0 задается в терминах ключевой последовательности $K = (K_t)_{t=0}^{2^k-1}$, а используя туже самую функцию следующего состояния, начиная с таблицы единичной подстановки $(l)_{t=0}^{2^k-1}$,мы получим таблицу, представляющую S_0. Ключевая последовательность K составляется из секретного ключа, который возможно будет повторяться, и случайного ключа, предаваемого в открытом виде с целью ресинхронизации. Таким образом, шифрованный текст получается в результате сложения по модулю 2 двоичных векторов S_t длинной k=8 и вектора z_t, длинной k.

Данный алгоритм был разработал в 1987 году Роном Ривевестом и держался в коммерческой тайне до 1994 года. Любопытно, что в США официальной рекомендованной длинной ключа является значение в 128 бит, в то время как для RC4 введен специальный статус, который означает, что разрешено экспортировать шифры с 40 и 56 битными ключами, для всех заграничных филиалов американских компаний.

Контрольные вопросы:

1) **Опишите алгоритм получения шифротекста в RC4.**
2) **Дайте определение понятию гаммирования.**
3) **Что произойдет если увеличить k?**
4) **Какова рекомендованная длина ключа США?**
5) **В каком году был разработан RC4?**
6) **Укажите формулу формирования ключевой последовательности.**
7) **Кто являлся разработчиком RC 4?**

8) **Предложите программную реализацию RC4 на известном языке программирования.**
9) **Что подразумевает под собой специальный статус RC4?**

2.13 Роторные машины

Современные компьютерные шифровальные машины не всегда были на страже криптографии, когда-то давно все начиналось с простого обруча и палки, вокруг которого наматывалась шифрованная записка. С течением времени на смену примитивным шифрующим приспособлениям пришли роторные машины. Пожалуй, наиболее известной из современного кинематографа является Энигма. Портативная шифровальная машина, которая широко применялась во времена второй мировой войны.

На самом деле Энигма это целое семейство шифровальных машин, применявшихся еще с 20 годов прошлого века. Примечательно, то, что изначально с точки зрения математики криптостойкость Энигмы была на очень низком уровне, а на практике в сочетании с ошибками операторов и процедурными изъянами и прочими уловками (например, передачи заведомо известных сообщений и захватом шифрованных машин вместе с книгами шифров) позволяла без особого труда разгадывать шифры и читать сообщения. Но факт остается фактом, Энигма – самая знаменитая шифровальная машина.

Как и другие роторные машины, Энигма состояла из комбинаторики механических и электрических систем. Механическая часть включала в себя клавиатуру и набор вращающихся дисков (роторов), которые были расположены вдоль вала и плотно прилегали к нему. Вся система приводилась в действие механизмом, который вращал один и более роторов при нажатии на клавиши оператором. В каждой серии механизм работы мог отличаться, но принцип работы оставался неизменным: при каждом нажатии на клавиши самый правый ротор сдвигался на одну позицию, а при определенных условиях сдвигались и остальные роторы. Таким образом, движение роторов приводило к различным криптографическим преобразованиям при каждом следующем нажатии на клавиши. Механические части двигались и перемыкали контакты, образуя постоянно меняющийся электрический контур. Практически весь процесс шифрования был реализован электрически. При нажатии на клавиши контуры замыкались и ток подавался на одну из набора лампочек, которая и отображала искомую букву кода. Примером такого подхода является следующая картина:

– оператору необходимо зашифровать слово Void;

– при нажатии на V сдвигается ротор и замыкается контур таким образом, что загорается лампочка над буквой E, следовательно первой буквой в шифротексте будет E;

– продолжаем набирать наше сообщение, замыкая разные контуры, таким образом, мы получим искомый шифротекст;

– получившийся шифротекст принял вид Elor.

При кажущейся простоте в шифровании подобным образом это далеко не обычная замена, дело в том, что в зависимости от того какие клавиши были нажаты ранее будет меняться и сама замена. Роторы – это сердце Энигмы. Каждый ротор представлял собой диск примерно 10 см в диаметре, сделанный из эбонита или бакелита, с пружинными штыревыми контактами на одной стороне ротора, расположенными по окружности. На другой стороне находилось соответствующее количество плоских электрических контактов. Штыревые и плоские контакты соответствовали буквам в алфавите, обычно это были 26 букв от A до Z. При соприкосновении контакты соседних роторов замыкали электрическую цепь. Внутри ротора каждый штыревой контакт был соединён с одним из плоских. Порядок соединения мог быть различным. Как мы уже говорили, сам по себе ротор производил очень простой шифр элементарной замены, но при использовании нескольких роторов надежность шифрования повышалась в разы.

Преобразование Энигмы для каждой буквы может быть определено математически как результат перестановок. Рассмотрим трёхроторную армейскую модель. Предположим, что P обозначает коммутационную панель, U обозначает отражатель, а L, M, R обозначают действия левых, средних и правых роторов соответственно. Тогда шифрование E может быть выражено как: $E = PRMLUL^{-1}M^{-1}R^{-1}p^{-1}$. После каждого нажатия клавиш роторы движутся, меняя трансформацию, а при движении ротора на i позиций происходила трансформация p^iRp^{-i}, где p циклическая перестановка от A к B, от B к C и так далее, по такому же принципу и все остальные роторы связаны алфавитом. Функция шифрования может быть представлена следующим образом:

$$E = P(p^iRp^{-i})(p^iMp^{-i})(p^kRp^{-k})U(p^kL^{-1}p^{-k})(p^jM^{-1}p^{-j})(p^iR^{-1}p^{-i})P^{-1}.$$

Для работы Энигмы было характерно первоначальное состояние ключа, которое включало в себя следующие параметры:

– расположение роторов и их выбор;

– первоначальная позиция роторов, выбранная оператором;

– настройки алфавитных колец;

– настройки штепселей на коммутационной панели.

Пожалуй, коммутационной панели стоит уделить особое внимание, ведь именно она внесла огромный вклад в усложнение шифровальных машин. Намного большее, чем введение дополнительных роторов. Дело в том, что с Энигмой без коммутационно панели можно справиться при помощи листка бумаги и самого аппарата, однако после добавления коммутационно панели взломщикам пришлось изобретать целые машины. Суть коммутационно панели была достаточно проста: кабель, помешенный на панель, попарно соединял буквы. Эффект состоял в перестановке этих букв до и после похождения сигнала через роторы. А именно при нажатии на E, сигнал отправлялся на Q, и только после этого уже уходит во второй ротор. Примечательно, что таких коммутационных проводов могло использоваться до 13 штук, а значит шифр охватывал все 26 клавиш и увеличивал сложность системы в несколько раз. Каждая буква на коммутационно панели имела два гнезда. При вставке штепселя разъединялось верхнее гнездо от клавиатуры и нижнее гнездо от ротора. Штепсель на другом конце вставлялся в гнездо другой буквы.

На протяжении всей эксплуатации Энигмы к ней было придумано масса дополнительных вспомогательных устройств, которые облегчали работу с ней или усложняли алгоритм шифрования. Но, пожалуй, самым выдающимся приспособлением того времени стала дополнительная клавиатура, которая ставилась поверх основной клавиатуры и соединялась с Энигмой, она записывала на бумагу получившийся шифротекст. По сути это давало возможность обходиться всего одним оператором, а персонал, который будет пересылать данное сообщение, будет иметь дело только с конечным шифротекстом, и никак не будет контактировать с защищаемой информацией.

Контрольные вопросы:

1) **Что такое роторные машины?**
2) **Опишите механизм шифрования Энигмы.**
3) **Укажите основной минус в системе Энигмы.**
4) **Для чего вводилась коммутационная панель?**
5) **Опишите работу коммутационно панели.**
6) **Дайте математическое описании е работы Энигмы.**
7) **Каковы характерные особенности ключа Энигмы?**

2.14 Атаки на поточные шифры

Как не бывает добра без зла, дня и ночи, так не бывает и криптографии без криптоанализа. Криптоанализ – наука о методах получения исходного значения зашифрованной информации, не имея доступа к секретной

информации (ключу), необходимой для этого. Результаты криптоанализа конкретного шифра называют криптографической атакой на этот шифр.

Любой шифр создается с целью обеспечить конфиденциальность защищаемой информации, а исходя из этой концепции всегда найдутся люди желающие получить доступ к данной информации. Все методы криптоанализа поточных шифров обычно разделяются на силовые, статистические и аналитические.

К силовому классу относятся атаки, осуществляемые полным перебором всех возможных вариантов. Сложность такого перебора зависит от количества всех возможных вариантов. Этот класс атак применим абсолютно ко всем видам систем поточного шифрования. Поэтому при разработке систем шифрования обычно стараются, чтобы данный вид атаки являлся наиболее эффективным по сравнению с другими методами взлома, проектируя систему таким образом, что полный перебор имеет настолько огромное пространство решений, что результат может не последовать в течение нескольких лет, а иногда и столетий.

В криптографии на вычислительной сложности полного перебора основывается оценка криптостойкости шифров. В частности, шифр считается криптостойким, если не существует метода «взлома» существенно более быстрого, чем полный перебор всех ключей. Криптографические атаки, основанные на методе полного перебора, являются самыми универсальными, но и самыми долгими. В связи с этим существуют и разрабатываются целые методы оптимизации полного перебора, к ним относятся такие методы как:

- метод ветвей и границ;
- метод параллельных вычислений;
- метод радужных таблицы.

Метод ветвей и границ это общий алгоритмический метод для нахождения оптимальных решений различных задач. По сути, здесь происходит отсев подмножества допустимых решений, заведомо не содержащих оптимальных решений. Данный метод был впервые предложен в 1960 году Лендом и Дойгом, для решения задач целочисленного программирования.

Параллельные вычисления, это, по сути, подходы применяемые для увеличения скорости подбора ключа. Зачастую используется два подхода. Первый подход это конвейер, а второй это разбиение множества решений на N подмножеств, каждое из которых намного меньше оригинала, таким образом распределив каждое из подмножеств на отдельную машину, полный перебор сократиться в n раз, в зависимости от того на сколько подмножеств разбили первоначальное множество. Основная проблема, естественно, связана с

определением и разбиением основного множества решений. Перебор осуществляется до тех пор, пока одна из машин не найдет искомый ключ.

Радужная таблица в основе своей создается путем построения цепочек возможных паролей. Каждая из таких цепочек начинается со случайного возможного пароля, а затем подвергается действию хеш-функции и функции редукции (о них поговорим позже). Данные действия преобразуют результат хеш-функции в некоторый возможный пароль. Промежуточные пароли в цепочке отбрасываются, и в таблицу записывается только первый и последний элемент цепочек. Создание таких таблиц требует больше времени, чем создание таблиц поиска, но это значительно сокращает объем необходимой памяти для поиска слов. Для восстановления пароля данное значение хеш-функции подвергается функции редукции и ищется в таблице. Если не было найдено совпадения, то снова применяется хеш-функция и функция редукции. Данная операция продолжается, пока не будет найдено совпадение. После нахождения совпадения цепочка, содержащая его, восстанавливается для нахождения отброшенного значения, которое и будет искомым паролем. В результате получается таблица, которая может с высокой вероятностью восстановить пароль за небольшое время.

Таким образом, существует таблица продолжительности полного перебора паролей, в ней представлено оценочное время полного перебора паролей, в зависимости от их длинны.

Кол-во знаков	Кол-во вариантов	Стойкость	Время перебора
1	36	5 бит	Менее секунды
2	1296	10 бит	Менее секунды
3	46656	15 бит	Менее секунды
4	1679616	21 бит	17 секунд
5	60466176	26 бит	10 минут
6	2176782336	31 бит	6 часов
7	78364164096	36 бит	9 дней
8	2,8211099x1012	41 бит	11 месяцев
9	1,0155995x1014	46 бит	32 года
10	3,6561584x1015	52 бита	1162 года
11	1,3162170x1017	58 бит	41823 года
12	4,7383813x1018	62 бита	1505615 лет

Таблица 2.14 – Таблица времени полного перебора

В таблице 2.14 представлено примерное время перебора всех вариантов ключей пароля, исходя из того, что количество различных символов 36, а скорость перебора 100 000 вариантов в секунду. Примечательно, что число 36 соответствует лишь латинским буквам одно регистра в сочетании с цифрами, поэтому неудивительно, что при добавлении букв разных регистров и

специальных символов количество времени необходимого для полного перебора увеличивается в несколько раз. Так же из таблицы 2.14 хорошо видно, что пароли длинной менее 8 символов крайне уязвимы к полному перебору.

Наряду с методом «грубой силы» или полного перебора, стоят методы статистических атак. Они разделены на два подкласса:

- метод криптоанализа статистических свойств шифрующей гаммы;
- метод криптоанализа сложности последовательности.

Первый случай направлен на изучение выходной последовательности криптоалгоритмов. В этом случае криптоаналитик пытается установить значение следующего бита последовательности с вероятностью выше вероятности случайного выбора с помощью различных статистических тестов.

Во втором случае криптоаналитик попросту пытается найти способ сгенерировать последовательность аналогичную гамме, но более простым способом.

Завершают хит-парад наиболее часто используемых классов атак – аналитические атаки. Этот вид атак рассматривается в предположении, что криптоаналитику известны описания генератора, открытый и закрытые тексты. То есть задача криптоаналитика сводится к определению использованного ключа, посредством заполнения начальных регистров. Существует несколько видов аналитических атак применяемых к синхронным поточным шифрам:

- корреляционные;
- компромиссные;
- инверсионные;
- по ключевой нагрузке и реинициализации;
- XSL-атака;
- атака «Предполагай и определяй».

Наиболее распространенными атаками для взлома поточных шифров являются корреляционные. Криптоаналитики знают, что работа по вскрытию криптосистемы может быть заметно сокращена, если нелинейная функция пропускает на выход информацию о внутренних компонентах генератора. Поэтому для восстановления начального заполнения регистров корреляционные атаки исследуют корреляцию выходной последовательности шифросистемы с выходной последовательностью регистров. Корреляционные атаки разделяются на следующие подклассы:

- базовые;

– атаки на низко-весовых проверках четности;

– атаки на использовании сверхточных кодов;

– атаки на технике турбо кодов;

– атаки на восстановлении линейных полиномов;

– быстрые.

Все корреляционные атаки основаны на базовых, поэтому рассмотрим мы именно их. Схема базовой корреляционной атаки представлена на рисунке 2.14.

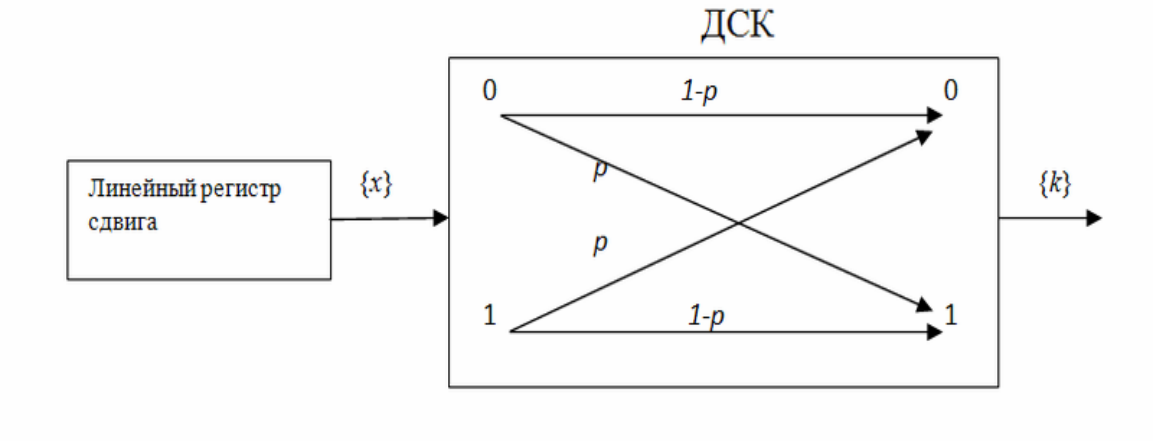

Рисунок 2.14 – Схема базовой корреляционной атаки.

Для выявления j-го линейного регистра сдвига с выходной последовательностью x^i на гамму шифра g, моделируется часть генератора, как двоичный симметричный канал, изображенный на рисунке 2.14. Алгоритм атаки состоит из 2 этапов:

– вычисления корреляционной вероятности;

– перебор начальных заполнений регистра.

Атаки, основанные на низко-весовых проверках четности, основываются на идее использования уравнения проверки четности для полинома обратной связи линейного регистра. Примером атакой атаки может являться корреляционная атака Майера и Штаффельбаха.

В случае значительной длинны перехваченной гаммы и заведомо малого размера ключа, наиболее эффективно использовать компромиссную атаку, или атаку «время-память». Суть данной атаки состоит в восстановлении исходного регистра сдвига, используя известную схему устройства и фрагмент шифрующей последовательности. Такая атака состоит из двух этапов:

– построение большого словаря, в котором записаны всевозможные пары «состояние–выход»;

– предположение о начальном заполнении регистра сдвига, генерация выхода, просмотр перехваченной выходной последовательности и поиск соответствия со сгенерированным выходом.

В том случае, если произошло совпадение, то данное предположительное заполнение с большой вероятностью является начальным. Примером таких атак является атака Бирюкова-Шамира.

Атака «Предполагай и определяй» основана на предположении, что криптоаналитику известна гамма, полином обратной связи, количество сдвигов регистра между выходами схемы и фильтрующей функции. Данная атака состоит из 3 этапов:

– предположение о заполнении некоторых ячеек регистра;

– определение полного заполнения регистра на основании предположения о знании криптоаналитика;

– генерация выходной последовательности;

  - если она совпадает с гаммой, то предположение на первом этапе было верно;
  - если не совпадает, то возвращаемся к этапу 1.

Не сложно догадаться, что сложность такой атаки зависит от устройства генератора и от количества сделанных предположений.

Контрольные вопросы:

1) **Дайте определение понятию криптоанализ.**
2) **Дайте определение понятию криптографическая атака.**
3) **На какие основные классы делятся криптографические атаки?**
4) **Назовите методы атак полным перебором.**
5) **Назовите подклассы статистических атак.**
6) **Назовите виды аналитических атак.**
7) **На какие подклассы делятся корреляционные атаки.**
8) **Опишите механизм базовой корреляционной атаки.**
9) **Опишите компромиссную атаку.**
10) **Опишите работу атаки «время-память».**
11) **Опишите работу атаки «Предполагай и определяй».**
12) **Опишите работу метода параллельных вычислений.**
13) **Опишите ход построения радужных таблиц.**

14) **К чему приведет увеличение количества вариантов символов в подбираемом ключе?**

15) **Какое количество символов в ключе является рекомендуемым?**

16) **Для чего необходимо применять специальные символы и различный регистр при составлении ключей?**

Глава 3

БЛОЧНЫЕ ШИФРЫ

3.1 Классификация блочных шифров

В предыдущей главе мы рассмотрели поточные шифры и вплотную подошли к изучению блочных шифров. Блочный шифр – симметричный шифр, при котором весь открытый текст делится на n блоков фиксированной длины и обрабатывается поблочно. Блочный шифр всегда преобразуется определенный блок открытого текста в один и тот же шифротекст, независимо от того, что за данные были зашифрованы до этого. Блочные шифры бывают двух основных видов:

- шифры перестановки (P-блоки);
- шифры замены (S-блоки).

Шифры перестановки переставляют элементы открытых данных в некотором новом порядке. Существует несколько видов такой перестановки:

- горизонтальные;
- вертикальные;
- двойные;
- решетки;
- лабиринты.

Шифры замены заменяют элементы открытых данных на другие элементы по определенной закономерности. В основном различают шифры просто и сложной замены.

Каким бы не был блочный шифр, он всегда будет придерживаться двух основных характерных принципов:

- рассеивание;
- перемешивание.

Под рассеиванием следует понимать изменение любого знака открытого текста или ключа повлияет на большое число знаков получившегося

шифротекста, таким образом, будут скрыты статистические свойства открытого текста.

Под перемешиванием следует понимать использование преобразований, затрудняющих получение статистических зависимостей между шифрованным и открытым текстом. Пожалуй, основным достоинством блочных шифров является тот факт, что процедура шифрования и расшифровывания достаточно схожи, и, как правило, отличаются лишь последовательностью действий. Данное свойство существенно облегчает процедуру создания устройств шифрования, в виду того, мы можем использовать одни и те же блоки в цепях шифрования и дешифрования.

Блочный шифр всегда состоит из двух взаимосвязанных алгоритмов, алгоритмов шифрования и расшифровывания. Входными данными служит блок размером n бит и ключ, размером k бит. На выходе мы получим n-битный зашифрованный блок. Обозначим алгоритм шифрования за E, а алгоритм расшифровывания за E^{-1}, тогда для любого фиксированного ключа функция расшифровывания будет являться обратной к функции шифрования $E_K^{-1}(E_K(M)) = M$. Очевидно, что шифр будет работать тогда и только тогда, когда каждому ключу K, E_K будет являться однозначным отображением. Проще говоря, множеству битов ключа будет взаимно-однозначно соответствовать множество битов зашифрованного текст побитно.

Размер блока n это фиксированный параметр блочного шифра, обычно равный 64 или 128 битам, хотя некоторые блочные шифры используют другие значения. Длина в 64 бита была приемлема до середины 90-х годов, затем длина плавно увеличилась до 128 бит, что примерно соответствует машинному слову и позволяет наиболее рационально реализовывать алгоритмы на большинстве вычислительных платформ. Различные алгоритмы шифрования позволяют шифровать открытый текст произвольной длины. Каждый из алгоритмов имеет определенный характеристики:

- вероятность ошибки;
- простота доступа;
- уязвимость.

Существуют характерные размеры ключей, которые увеличиваются в соответствии с растущими вычислительными способностями современных суперкомпьютеров. Типичные размеры ключа:

- 40 бит;
- 56 бит;
- 64 бита;
- 80 бит;

– 128 бит;

– 192 бита;

– 256 бит.

По состоянию на 2010 год 128-битный ключ способен был выстоять перед атакой полного перебора. При использовании блочного перед исполнителем зачастую встает проблема дополнения. Дело в том, что размер шифруемого блока не всегда кратен размеру блока. Допустим, что мы имеем дело с алгоритмом, разбивающим данные по 16 байт, а в последнем используемом блоке мы получаем остаток из 5 байт. Отбросить данные мы не можем, поэтому данные 5 бит нам придется тащить за собой. Можно передать оставшиеся 5 бит в незашифрованном виде вместе с основной частью, но тогда мы получим возможность, хоть и малую, для атаки на наш шифр. В основном с оставшимися байтами поступают следующим образом:

– определения остатка из е–байт, не способных образовать целый блок;

– дополнением блок случайными цифрами предпоследнего байта;

– в оставшемся последнем байте записывается число, равно количеству случайных символов.

Таким образом, при расшифровывании, после зашифровывания всех блоков необходимо будет просто отбросить столько байт с конца, сколько было указано в последнем байте последнего блока. Правда, если исходное сообщение имело длину, кратную размеру блока, то возникает некоторая сложность, т. к. требуется добавить 0 байт, и последний байт должен содержать число байт дополнения. Для разрешения этой проблемы при зашифровывании добавляется новый блок, последний байт которого содержит размер блока. Таким образом, дополнительный блок будет целиком отброшен при расшифровывании.

Контрольные вопросы:

1) **Что такое блочный шифр?**
2) **Для чего используются блоки?**
3) **Назовите характерный размер блоков.**
4) **Назовите виды блочных шифров.**
5) **Назовите типичные размеры ключа.**
6) **Назовите основные характеристики алгоритмов блочного шифрования.**
7) **Какой последовательности действий подвергаются байты открытого текста, которые не способны образовать целый блок?**
8) **Опишите основной принцип работы блочных шифров.**

9) **Что такое принцип рассеивания и принцип перемешивания в блочных шифрах?**

3.2 Режимы использования блочных шифров

В зависимости от области применения, степени конфиденциальности данных, объема открытого текста и применяемых ключей блочные шифры могут быть использованы в различных режимах.

Режим шифрования – метод применения блочного шифра, позволяющий преобразовать последовательность блоков открытых данных в последовательность блоков зашифрованных данных. При этом для шифрования одного блока могут использоваться данные другого блока. Обычно режимы шифрования используются для модификации процесса шифрования так, чтобы результат шифрования каждого блока был уникальным вне зависимости от шифруемых данных и не позволял сделать какие–либо выводы об их структуре. Это обусловлено, прежде всего, тем, что блочные шифры шифруют данные блоками фиксированного размера, и поэтому существует потенциальная возможность утечки информации о повторяющихся частях данных шифруемых на одном и том же ключе.

Каждый из режимов имеет свои характерные особенности, свои положительные и отрицательные стороны, так например разные режимы могут быть более или менее устойчивыми к определенным видам атак, а некоторые лучше или хуже умеют восстанавливаться при сбоях. Список возможных режимов использования блочных шифров можно вести до бесконечности, но среди них можно выделить наиболее популярные:

- режим простой замены;
- с зацеплением блоков шифротекста;
- с обратной связью по шифротексту;
- с обратной связью по выходу;
- по счетчику;
- с зацеплением блоков открытого текста;
- с обратной связью по открытому тексту;
- с усиленным сцеплением блоков шифротекста;
- с обратной связью по выходу и нелинейной функцией;
- по счетчику с нелинейной функцией.

Некоторым из режимов мы уделим особое внимание, ввиду их повсеместной распространенности.

Контрольные вопросы:

1) **Что такое режим шифрования?**
2) **Назовите основные режимы шифрования.**

3.3 Режим простой замены

Режим простой замены или режим электронной кодовой книги (ECB), один из основных режимов использования блочных шифров. При использовании данного режима все блоки открытого текста зашифровываются независимо друг от друга, однако статистические свойства открытых данных частично сохраняются, так как каждому одинаковому блоку данных однозначно соответствует зашифрованный блок данных. Таким образом, при большом объеме повторяющихся блоков мы можем получить частичную утечку данных. Это наиболее характерно при шифровании изображений, таким образом, ведь цвета передаются обычно одним массивом данных, таким образом, мы можем получить отдаленно похожую на оригинал картинку.

Рисунок 3.3.1 – Алгоритм шифрования ECB**.**

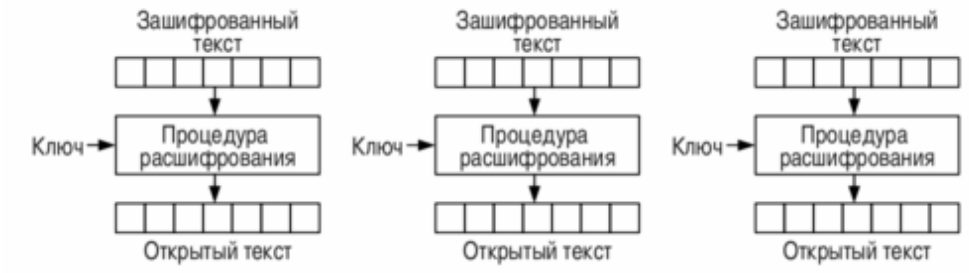

Рисунок 3.3.2 – Алгоритм расшифровывания ECB.

Алгоритм шифрования и расшифровывания в режиме электронной кодовой книги представлен на рисунках 3.3.1 и 3.3.2. Непосредственно этот режим применяется для шифрования небольших объемов данных, чтобы

избежать повторяющихся фрагментов. Основным достоинством этого шифра является простота реализации и отсутствие необходимости в последовательном шифровании каждого из элементов. Проще говоря, мы можем зашифровать сначала середину, затем конец, а затем начало открытого текста, таким образом, мы получаем возможность параллельного шифрования. К недостаткам данного режима относятся следующие моменты:

- при использовании одного ключа идентичные блоки открытого текста шифруются в идентичные блоки зашифрованного текста, что делает режим неустойчивым к статистическому анализу;
- возможность дублирования перехваченного блока, что приведет к искажению исходных данных.

Примером отечественного применения данного режима является ГОСТ 28147-89, о котором мы поговорим чуть ниже.

Контрольные вопросы:

1) **Опишите алгоритм шифрования ECB.**
2) **Опишите алгоритм расшифровывания ECB.**
3) **Назовите достоинства режима ECB.**
4) **Назовите недостатки режима ECB.**
5) **Опишите процесс параллельного шифрования.**

3.4 Режим шифрования с зацеплением

Режим шифрования с зацеплением или режим CBC – один из режимов шифрования для симметричного блочного шифра с использованием механизма обратной связи. Каждый блок открытого текста побитово складывается по модулю 2 с предыдущим результатом шифрования. Естественно с первым блоком мы ничего не делаем. Алгоритм шифрования и расшифровывания изображен на рисунках 3.4.1 и 3.4.2. Математически режим шифрования с зацеплением имеет вид: $C_0 = IV$, $C_i = E_k(P_i XOR C_{i-1})$, где i – номера блоков,IV – синхроссылка, C_i,P_i – блоки зашифрованного и открытого текста соответственно, а E_k – функция блочного шифрования. Расшифровка производится обратным образом: $P_i = C_{i-1} XOR E_k(C_i)$.

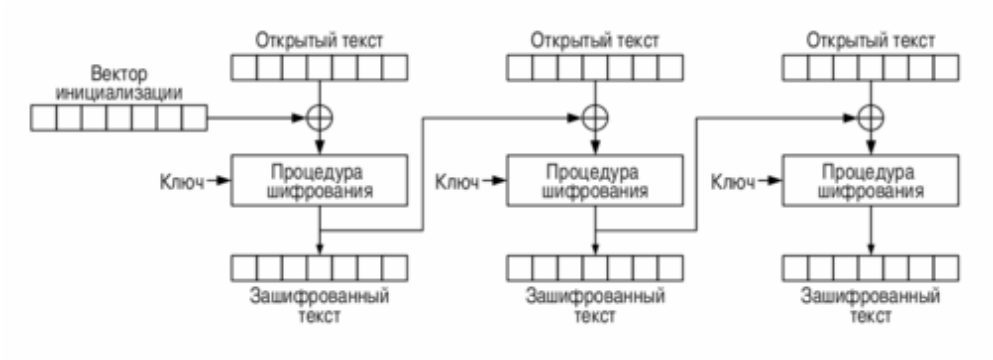

Рисунок 3.4.1 Алгоритм шифрования CBC.

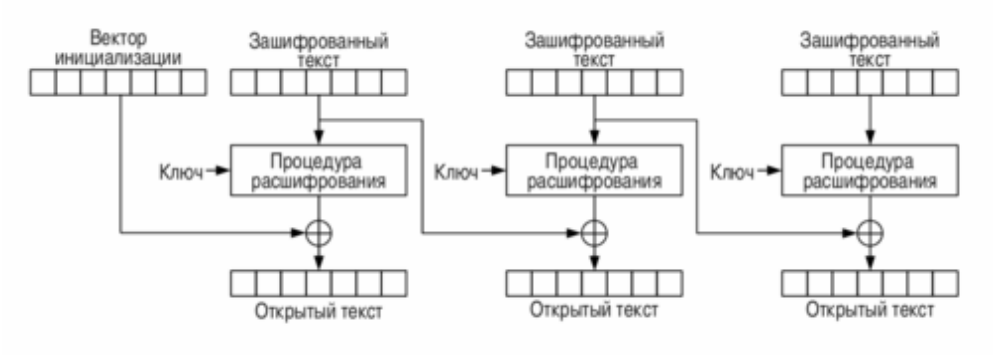

Рисунок 3.4.2 – Алгоритм расшифровывания CBC.

К явным достоинствам данного режима шифрования можно отнести самовосстановление шифра. Дело в том, что при потере одного бита шифротекста в процессе передачи мы получим ошибку во всем блоке, данная ошибка распространиться и на следующий, но она исчезнет через один блок, таким образом, оставшаяся часть придет без изменений.

Недостатками данного режима являются:

- неустойчив к ошибкам, связанным с потерей или вставкой битов, если не используется дополнительный механизм выравнивания блоков;
- злоумышленник имеет возможность добавить блоки к концу зашифрованного сообщения, дополняя тем самым открытый текст, но, не имея ключа шифрования, получится бессвязный бред;
- для очень крупных сообщений сохраняется возможность применение атак, основанных на структурных особенностях открытого текста.

Контрольные вопросы:

1) **Что такое режим CBC?**
2) **Опишите процесс шифрование режима CBC.**
3) **Опишите процесс расшифровывания режима CBC.**
4) **Назовите основные достоинства и недостатки режима CBC.**

5) **Объясните математически процесс шифрования и расшифровывания режима CBC.**
6) **К чему приведет потеря бита шифротекста?**
7) **Объясните процесс самовосстановления шифра CBC.**

3.5 Режим обратной связи по шифротексту

Режим обратной связи по шифротексту или режим гаммирования с обратной связью (CFB) – один из вариантов использования симметричного блочного шифра, при котором для шифрования следующего блока открытого текста он складывается по модулю 2 с перешифрованным результатом шифрования предыдущего блока. Алгоритм шифрования и расшифровывания CFB представлен на рисунке 3.5.1 и 3.5.2.

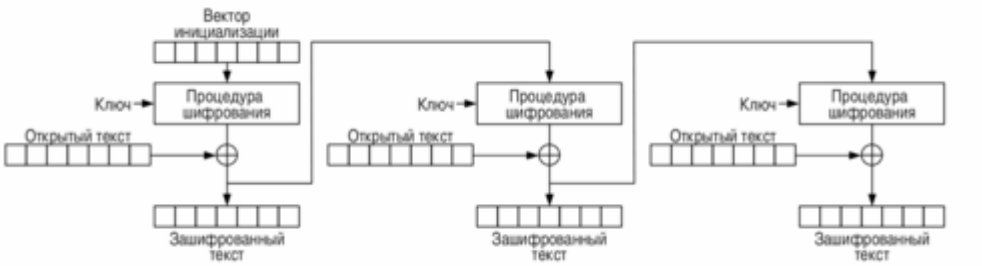

Рисунок 3.5.1 – Алгоритм шифрования CFB.

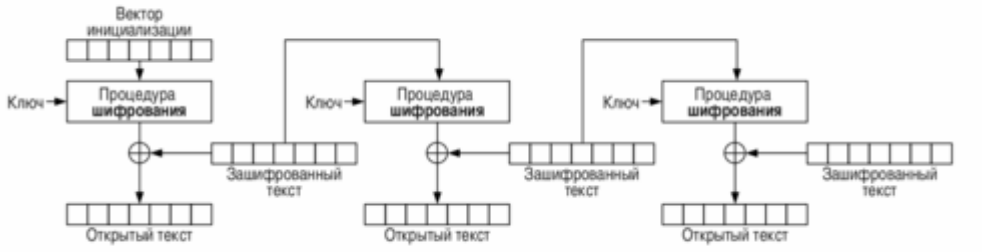

Рисунок 3.5.2 – Алгоритм расшифровывания CFB.

Математическое описание данного режима шифрования выглядит следующим образом: $C_0 = IV$, $C_i = E_k(C_{i-1}) XOR P_i$, а расшифровывания $P_i = E_k(C_{i-1}) XOR C_i$, где i – номера блоков,IV – синхроссылка, C_i,P_i – блоки зашифрованного и открытого текста соответственно, а E_k – функция блочного шифрования. Как и в предыдущих случаях, математическое описание шифрования и расшифровывания схожи, по сути как мы и говорили это просто обратная функция. Здесь синхроссылку как и в режиме CBC, допускается делать открытой, однако она должна быть уникальной. К достоинствам данного режима относится тот факт, что при наличии помех в канале передачи данных потеря части блока приведет лишь к невозможности расшифровать данный

блок, а последующие блоки останутся без изменения, таким образом, при получении сообщения, часть текста не имеющая логической связи с остальными частями текста просто отсеивается, и оператор запрашивает повторную передачу поврежденного блока.

Зачастую данный режим шифрования применяется для передачи по незащищенным каналам оцифрованной речи и потокового видео.

Контрольные вопросы:

1) **Что такое CFB?**
2) **Объясните принцип шифрования и расшифровывания режима CFB.**
3) **Что произойдет если использовать повторяющиеся синхроссылки?**
4) **Назовите основное достоинство CFB.**

3.6 Режим шифрования с обратной связью по выходу

Режим шифрования с обратной связью по выходу или OFB основан на превращении блочного шифра в синхронный шифропоток, что приводит к генерации ключевых блоков, которые являются результатом сложения с блоками открытого текста для получения зашифрованного. Так же, как с другими шифрами потока, зеркальное отражение в зашифрованном тексте производит зеркально отраженный бит в открытом тексте в том же самом местоположении. Это свойство позволяет многим кодам с исправлением ошибок функционировать как обычно, даже когда исправление ошибок применено перед кодированием. Алгоритм шифрования и расшифровывания представлен на рисунках 3.5.1 и 3.5.2.

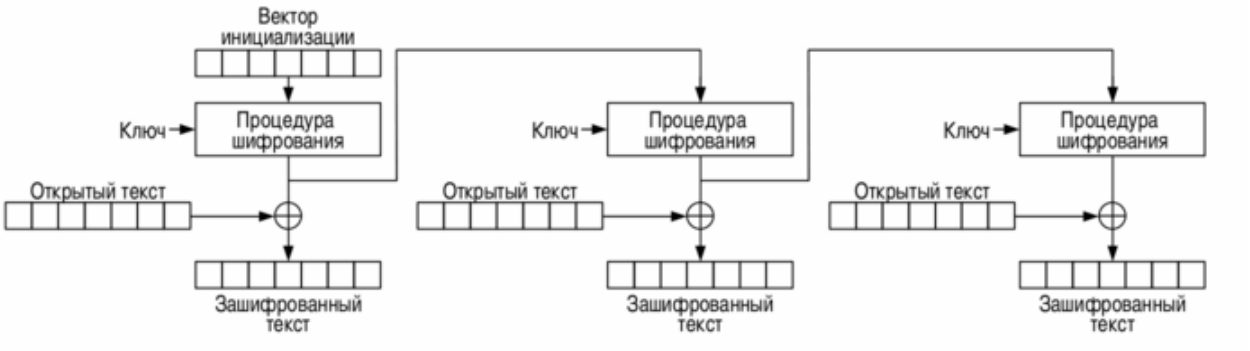

Рисунок 3.5.1– Алгоритм шифрования OFB.

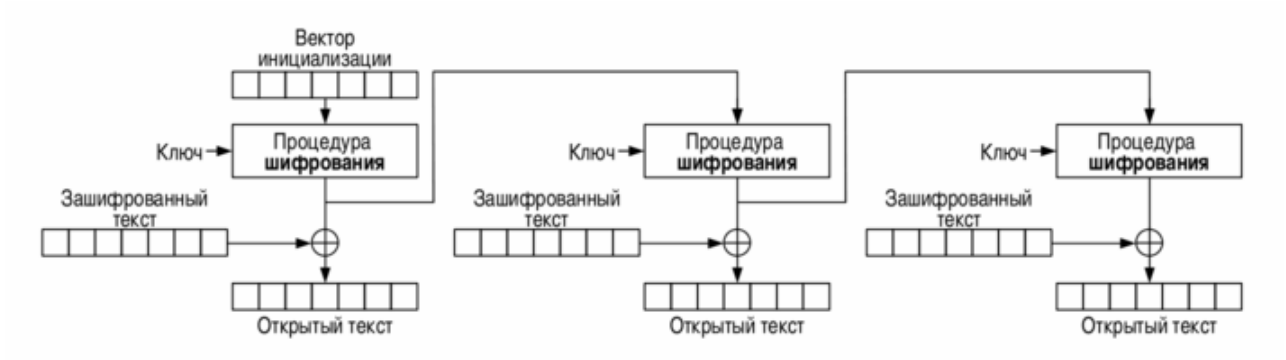

Рисунок 3.5.2– Алгоритм расшифровывания OFB**.**

Каждая операция блочного шифра обратной связи вывода зависит от всех предыдущих и поэтому не может быть выполнена параллельно. Однако, из-за того, что открытый текст или зашифрованный текст используются только для конечного сложения, операции блочного шифра могут быть выполнены заранее, позволяя выполнить заключительное шифрование параллельно с открытым текстом.

Математически данный режим может быть описан следующим образом: $C_i = P_i \; XOR \; O_i$, $P_i = C_i \; XOR \; O_i$, $O_i = E_k(O_{i-1})$, $O_0 = IV$, где i – номера блоков,IV – синхроссылка,O_i– ключевой поток,C_i, P_i – блоки зашифрованного и открытого текста соответственно, а E_k – функция блочного шифрования. В качестве достоинства данного режима выступает оперативность обработки, что достигается симметричностью исполнения и простоты математических функций. Область применения: потоки видео, аудио или данных, для которых необходимо быстрое шифрование и доставка.

Контрольные вопросы:

1) **Что такое режим OFB?**
2) **Опишите процесс шифрования и расшифровывания режима OFB.**
3) **Опишите математически режим OFB.**
4) **Как формируется ключевой поток в режиме OFB?**
5) **Назовите основное достоинство режима OFB.**
6) **Какова область применения режима OFBи почему?**

3.7 DES

DES (Data Encryption Standard) – симметричный алгоритм шифрования, разработанный фирмой IBM и утвержденный правительством США в 1977 году как официальный стандарт (FIPS 46–3). В своем составе DES имеет блоки по 64 бита и 16 цикловую структуру сети Фейстеля, для шифрования использует

ключ с длиной 56 бит. Алгоритм использует комбинацию нелинейных (S-блоки) и линейных (перестановки E, IP, IP–1) преобразований. Для DES рекомендовано несколько уже рассмотренных нами режимов:

- режим ECB;
- режим CBC;
- режим CFB;
- режим OFB.

В 1972 году, после проведения исследования потребностей правительства США в компьютерной безопасности, американское национальное бюро стандартов – теперь переименовано национальный институт стандартов и технологий – определило необходимость в общеправительственном стандарте шифрования некритичной информации.

15 мая 1973 года, после консультации с агентством национальной безопасности, НБС объявило конкурс на шифр, который удовлетворит строгим критериям проекта, но ни один конкурсант не обеспечивал выполнение всех требований. Второй конкурс был начат 27 августа 1974. На сей раз, шифр Lucifer, представленный IBM и развитый в течение периода 1973-1974 сочли приемлемым, он был основан на более раннем алгоритме Хорста Фейстеля.17 марта 1975 года предложенный алгоритм DES был издан в Федеральном Регистре. В следующем году было проведено 2 открытых симпозиума по обсуждению этого стандарта, где подверглись жёсткой критике изменения, внесённые АНБ в алгоритм: уменьшение первоначальной длины ключа и S-блоков, критерии проектирования которых не раскрывались. АНБ подозревалось в сознательном ослаблении алгоритма с целью, чтобы АНБ могло легко просматривать зашифрованные сообщения. После всех дебатов сенатом США была проведена проверка действий АНБ, результатом которой стало заявление, опубликованное в 1978, в котором говорилось о том, что в процессе разработки DES АНБ убедило IBM, что уменьшенной длины ключа более чем достаточно для всех коммерческих приложений, использующих DES, косвенно помогало в разработке S-перестановок, а также, что окончательный алгоритм DES был лучшим, по их мнению, алгоритмом шифрования и был лишён статистической или математической слабости. Также было обнаружено, что АНБ никогда не вмешивалось в разработку этого алгоритма.

Часть подозрений в скрытой слабости S-перестановок была снята в 1990, когда были опубликованы результаты независимых исследований Эли Бихама и Ади Шамира по дифференциальному криптоанализу – основному методу взлома блочных алгоритмов шифрования с симметричным ключом. S-блоки алгоритма DES оказались намного более устойчивыми к атакам, чем, если бы их выбрали случайно. Это означает, что такая техника анализа была известна АНБ ещё в 70-х годах XX века.

Но еще в 1998 году используя суперкомпьютер стоимостью 250 тыс. долл., сотрудники RSA Laboratory «взломали» утвержденный правительством США алгоритм шифрования данных менее чем за три дня. Предыдущий рекорд по скорости взлома был установлен с помощью огромной сети, состоящей из десятков тысяч компьютеров, и составил 39 дней. На специально организованной по этому случаю пресс-конференции ученые с беспокойством говорили о том, что злоумышленники вряд ли упустят случай воспользоваться подобной уязвимостью. Эксперимент проходил в рамках исследования DES Challenge II, проводимого RSA Laboratory под руководством общественной организации Electronic Frontier Foundation, которая занимается проблемами информационной безопасности и личной тайны в сети интернет. Суперкомпьютер, построенный в RSA Laboratory для расшифровки данных, закодированных методом DES по 56-разрядному ключу, получил название EFF DES Cracker. Как утверждали правительственные чиновники и некоторые специалисты, для взлома кода DES требуется суперкомпьютер стоимостью в несколько миллионов долларов. «Правительству пора признать ненадежность DES и поддержать создание более мощного стандарта шифрования», – сказал президент EFF Барри Штайнхардт. Экспортные ограничения, накладываемые правительством США, касаются технологий шифрования по ключам длиной более 40 бит. Однако, как показали результаты эксперимента RSA Laboratory, существует возможность взлома и более мощного кода. Проблема усугубляется тем, что стоимость постройки подобного суперкомпьютера неуклонно снижается.

Как уже говорилось DES – блочный шифр, входными данными для которого служит блок размером nбит и ключ, размером k бит. Далее DESсовершает преобразование путем многократного применения к блокам исходного текста операций простого и сложного преобразования между частями блока и одной из локальной части блока. Хотелось бы отметить, что все данные вне зависимости от своего формата должны предварительно быть представлены в бинарном виде, и только после этого разбиты на блоки необходимого размера. Все это может быть реализовано как программными, так и аппаратными средствами. Как мы уже говорили, в DES используется преобразование сетью Фейстеля и прежде чем продолжить изучение DES необходимо разобраться с этим понятием.

Сеть Фейстеля это один из методов построения блочных шифров. Сама сеть представляет собой определённую многократно повторяющуюся структуру, называющуюся ячейкой Фейстеля. При переходе от одной ячейки к другой меняется ключ, причём выбор ключа зависит от конкретного алгоритма. Операции шифрования и расшифровывания на каждом этапе очень просты, и при определённой доработке совпадают, требуя только обратного порядка используемых ключей. Шифрование при помощи данной конструкции легко реализуется как на программном уровне, так и на аппаратном, что обеспечивает

широкие возможности применения. Прямое и обратное преобразование сетью Фейстеля в DES представлено на рисунках 3.7.1 и 3.7.2.

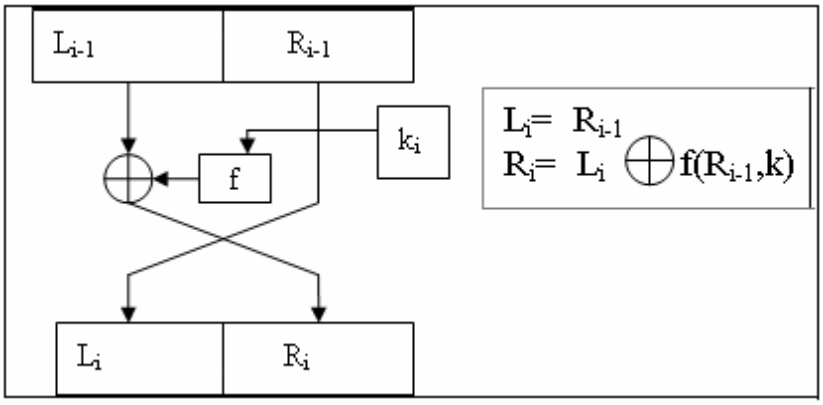

Рисунок 3.7.1 – Прямое преобразование сетью Фейстеля.

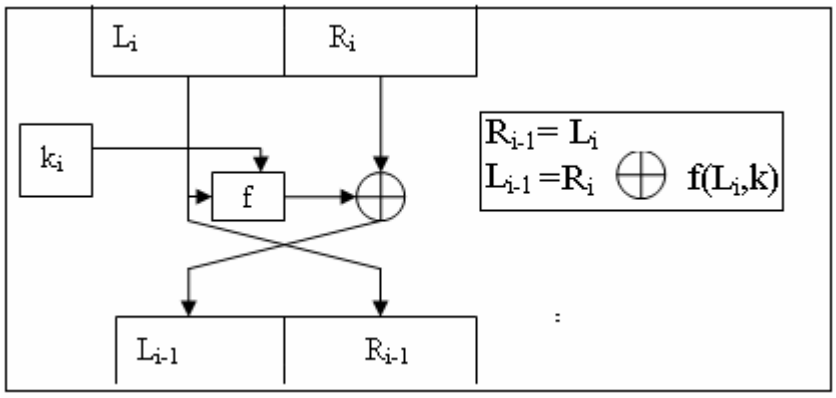

Рисунок 3.7.2 – Обратное преобразование сетью Фейстеля.

В DES прямое преобразование сетью Фейстеля применяется при шифровании, а при расшифровывании соответственно применяется обратное. Схема шифрования DES изображена на рисунке 3.6.3. Алгоритмически процесс шифрование представляется в следующем виде:

- подается исходный текст в виде блока в 64 бита;
- происходит процесс начальной перестановки;
- подается ключ, в результате чего происходит шифрование;
- начинается 16-ти кратное шифрование функцией Фейстеля;
- происходит конечная перестановка;
- получения зашифрованного блока.

В начальной перестановке исходный блок T в 64 бита преобразуется с помощью таблицы 3.7.1. По таблице первые 3 бита реализующего блока IP после начальной перестановки являются битами 58 50 42 входного блока T, а последние соответственно 23 15 7.

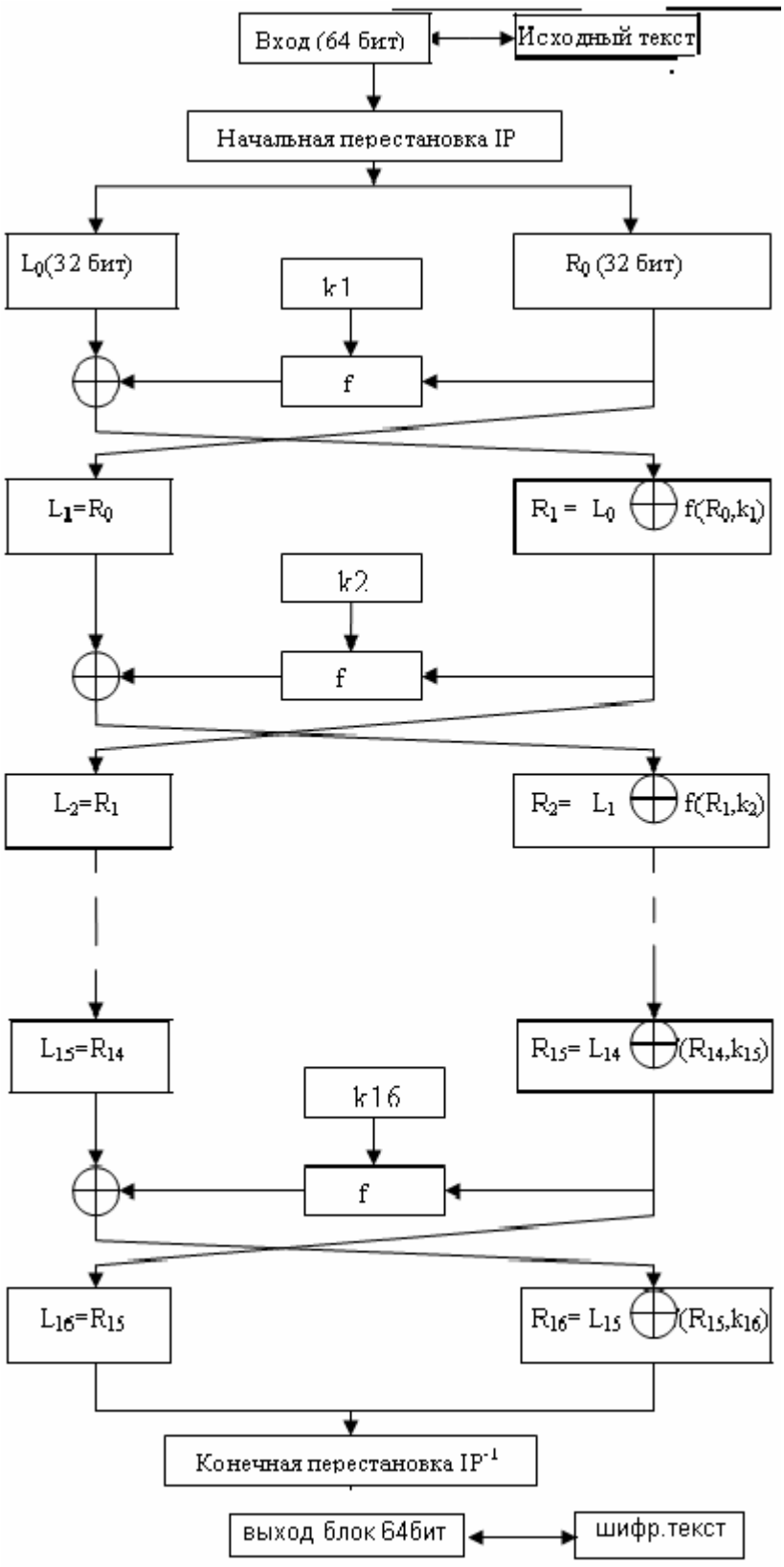

Рисунок 3.7.3 – Алгоритм шифрования DES.

Полученный после начальной перестановки 64 битный блок IP (T) участвует в 16-ти циклах преобразования сетью Фейстеля, в котором происходят следующие повторяющиеся действия:

– разбиение блока IP (T) на две равные части L_0R_0, по 32 бита, при этом $T_0\ IP(T) = L_0R_0$;

– при этом если результат преобразования i–1есть $T_{i-1} = L_{i-1}R_{i-1}$, то, следовательно, результат i преобразований определяется как $L_i = R_{i-1}, R_i = L_{i-1}\ XOR\ F(R_{i-1}, k_i)$;

– F – функция Фейстеля, она будет являться основной функцией шифрования.

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

Таблица 3.7.1 – Таблица начальной перестановки.

Аргументами функции Фейстеля будет являться наш полученный 32-битный блок R_{i-1}и 48 битный ключ k_i, который получается в результате преобразования исходного 56 битного ключа путем 8 перестановок S-блоками.

Конечная перестановка IP^{-1} действует на наш блок T после шифрования и используется для восстановления позиций. Она является обратной к перестановке IP .

В результате таких преобразований получается шифрованный блок, который и отправляется в пункт назначения вместе с остальными шифрованными блоками. Алгоритм расшифровывания полученных блоков представлен на рисунке 3.7.4.

DES был национальным стандартом США в 1977-1980 гг., но в настоящее время DES используется (с ключом длины 56 бит) только для устаревших систем, чаще всего используют его более криптоустойчивый вид 3DES. 3DES является простой эффективной заменой DES, и сейчас он рассмотрен как стандарт. Алгоритм DES широко применяется для защиты финансовой информации.

Для увеличения криптостойкости и придания нелинейности необходим выбор S-блоков, удовлетворяющих следующим условиям:

– каждая строка каждого блока должна быть перестановкой множества от 0 до 15;

– S-блоки не должны являться линейной или аффинной функцией своих аргументов;

– изменение одного бита на входе S-блока должно приводить к изменению, по крайней мере, двух битов на выходе;

– для каждого S-блока и его аргумента x значения S(x) и S(x XOR 16) должны различаться, по крайней мере, двумя битами.

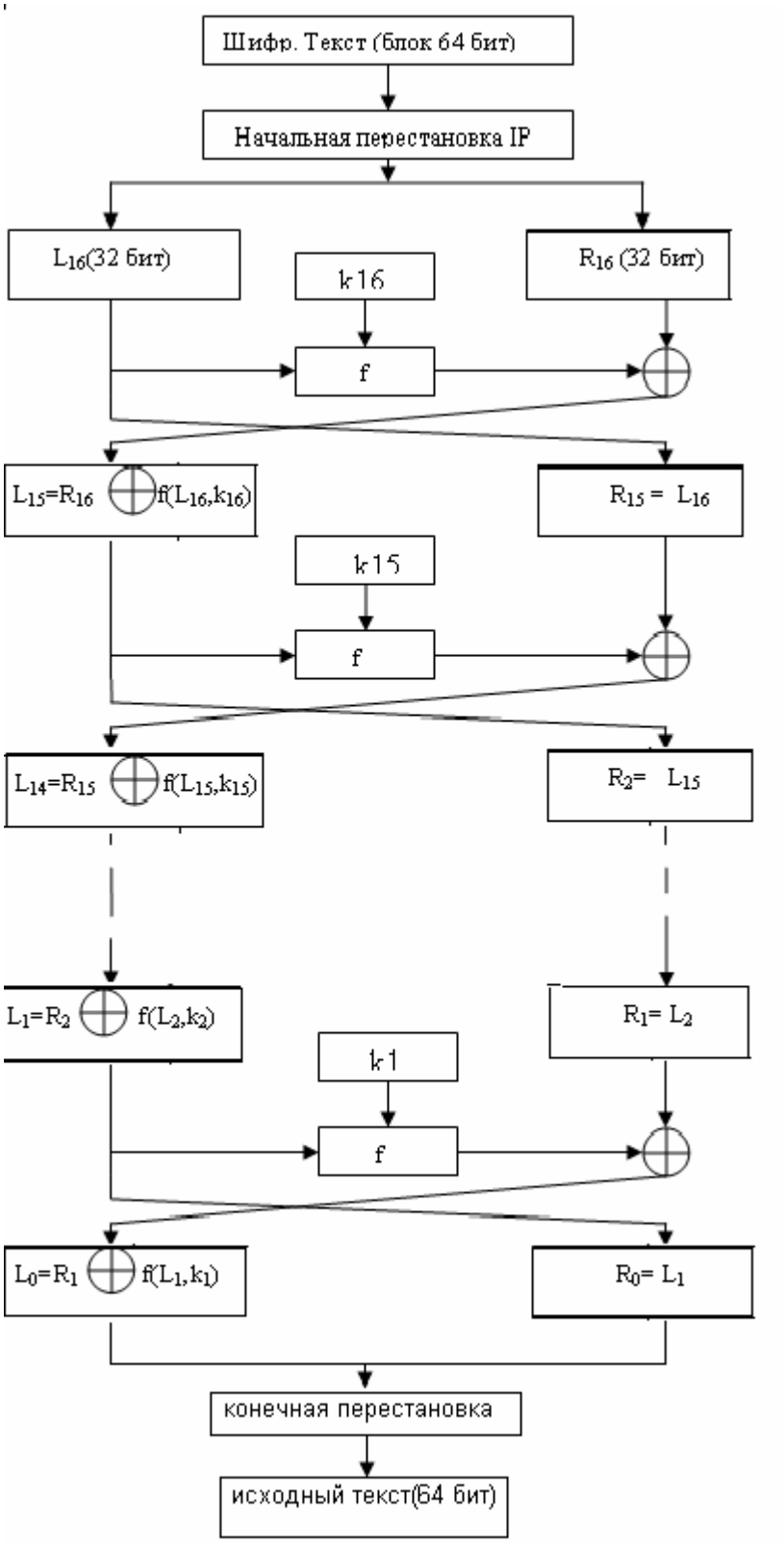

Рисунок 3.7.4 – Алгоритм расшифровывания DES.

Надо отметить, что в алгоритме DES существуют слабые и частично слабые ключи. Ввиду малого разнообразия ключей, всего, атака путем полного

перебора ключей с учетом современных вычислительных способностей не займет более 12 часов. Для DES характерные следующие типы атак:

- полный перебор;
- дифференциальный криптоанализ;
- линейны криптоанализ.

Для применения линейного и дифференциального криптоанализа требуется достаточно большой объём памяти, для сохранения выбранных открытых текстов до начала атаки.

Для увеличения криптостойкости DES существует несколько вариантов, но в основном это все прежний алгоритм, но с увеличенным размером ключа, до 112 бит в 2DESи 168 бит в 3DES. Существует 3 типа алгоритма 3DES:

- шифрование с 3 разными ключами;
- операции шифровка-расшифровка-шифровка с 3 разными ключами;
- операции шифровка-расшифровка-шифровка с 2 разными ключами, где 1 и 3 операция использую одинаковый ключ.

Второй метод является наиболее популярным, и стал применяться повсеместно. Следует отметить, что в 3DES ключи можно выбирать как независимо друг от друга, так и с установлением определенных зависимостей, таким образом, первый ключ будет влиять на формирование последующих двух. Ну а тот факт, что все три ключа могут быть одинаковыми, не вызывает сомнений. Причем последний вариант никак не повлияет на криптостойкость алгоритма, теоретически.

Контрольные вопросы:

1) **Что такое DES?**
2) **Когда и кем был изобретен DES?**
3) **Опишите принцип работы DES.**
4) **Объясните рисунок 3.7.3.**
5) **Объясните рисунок 3.7.4.**
6) **Какие ключи являются слабыми?**
7) **Объясните принцип работы таблиц 3.7.1.**
8) **Каким образом происходит генерация ключей DES?**
9) **Что такое функция Фейстеля?**
10) **Что такое сеть Фейстеля?**
11) **Объясните процесс начальной и конечной перестановки.**

12) Объясните процесс шифрования функцией Фейстеля.

13) Что такое основная функция шифрования?

14) Сколько блоков преобразования проходит ключ?

15) Каков размер исходного и преобразованного ключа?

16) Каким образом выражается R_i через L_i и наоборот?

3.8 ГОСТ 28147–89

ГОСТ 28147-89 – Советский и Российский стандарт симметричного шифрования, введённый в 1990 году, также является стандартом СНГ. Полное название – «ГОСТ 28147–89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования».

Блочный шифр, при использовании метода шифрования с гаммированием, может выполнять функции поточного шифра. Алгоритм, положенный впоследствии в основу стандарта, родился, предположительно, в недрах Восьмого Главного управления КГБ СССР, а ныне 8 управления ФСБ, скорее всего, в одном из подведомственных ему закрытых НИИ, вероятно, ещё в 1970-х годах в рамках проектов создания программных и аппаратных реализаций шифра для различных компьютерных платформ. С момента опубликования ГОСТа на нём стоял ограничительный гриф «Для служебного пользования», и формально шифр был объявлен «полностью открытым» только в мае 1994 года. История создания шифра и критерии разработчиков по состоянию на 2010 год не опубликованы.

Если проследить историческую параллель, то становится практически очевидно, что данный алгоритм шифрования был разработан в ответ на американский DES. И в некоторых позициях полностью его дублирует.

ГОСТ 28147-89 – блочный шифр с 256-битным ключом и 32 циклами преобразования, оперирующий 64-битными блоками. Основа алгоритма шифра – сеть Фейстеля, также как и в DES. Базовым режимом шифрования по ГОСТ 28147-89 является режим простой замены, но может функционировать и в других режимах:

- режим простой замены;
- режим гаммирования;
- гаммирование с обратной связью;
- режим имитовставки.

Для зашифровывания в этом режиме открытый текст сначала разбивается на две половины (младшие биты – A, старшие биты – B). На i-ом цикле используется подключ K_i :$A_{i+1} = B_i \; XOR \; \mathcal{F}\,(A_i.K_i)$, $B_{i+1} = A_i$.

Для генерации подключей исходный 256–битный ключ разбивается на восемь 32-битных блоков: $K_1 \ldots\ldots K_8$. Ключи $K_9 \ldots\ldots K_{24}$ являются циклическим повторением ключей $K_1 \ldots\ldots K_8$. Ключи $K_{25} \ldots\ldots K_{32}$ являются ключами $K_1 \ldots\ldots K_8$, но идущими в обратном порядке. После выполнения всех 32 циклов шифрования все блоки A и B склеиваются, а результатом и будет являться конец работы алгоритма. Функция $\mathcal{F}\,(A_i.K_i)$ вычисляется путем сложения по модулю. Далее результат разбивается на восемь 4-х битных последовательности на вход своего узла таблицы замен – S-блок. Пример S-блока изображен в таблице 3.7.1. Каждый S-блок представляет собой перестановку чисел от 0 до 15. Первая 4-битная последовательность попадает на вход первого S-блока, вторая – на вход второго и так продолжается до последнего S-блока. Выходы всех восьми S-блоков объединяются в 32-битное слово, затем всё слово циклически сдвигается влево на 11 битов.

Для расшифровывания применяется тот же алгоритм, только с инвертированной последовательностью ключей. Схема алгоритма изображена на рисунке 3.8.1.

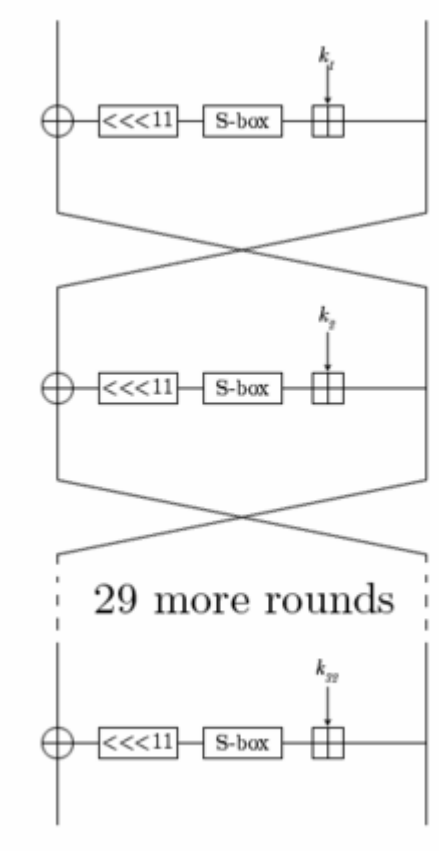

Рисунок 3.8.1 – Алгоритм работы ГОСТ 28147-89.

№ S-блока	Значения															
1	4	10	9	2	13	8	0	14	6	11	1	12	7	15	5	3
2	14	11	4	12	6	13	15	10	2	3	8	1	0	7	5	9
3	5	8	1	13	10	3	4	2	14	15	12	7	6	0	9	11
4	7	13	10	1	0	8	9	15	14	4	6	12	11	2	5	3
5	6	12	7	1	5	15	13	8	4	10	9	14	0	3	11	2
6	4	11	10	0	7	2	1	13	6	8	5	9	12	15	3	14
7	13	4	11	1	3	15	9	5	0	10	14	6	8	2	7	12
8	1	15	13	0	5	7	10	4	9	2	3	14	6	11	8	12

Таблица 3.8.1 – Пример S-блока для ГОСТ 28147-89.

Хотелось бы отметить следующие достоинства ГОСТ 28147-89:

- бесперспективность силовой атаки;
- эффективность реализации;
- быстродействие;
- наличие защиты от навязывания.

Несмотря на очевидные достоинства, ГОСТ все же подвергался критике, прежде всего из-за неполноты, в частности из-за генерации ключа и S-блоков. Существуют доказательства, что у ГОСТа есть слабые ключи и S-блоки, а в стандарте не описывается подбор ключей и отсеивание слабых. Алгоритм генерации S-блоков также не специализирован, что дает ряд недостатков:

- отсутствие возможности определения стойкости алгоритма, не зная S-блока;
- различная реализация алгоритма предполагает использование разных S-блоков, а значит, некоторые реализации могут быть несовместимы между собой;
- потенциальная возможность использования S-блоков, в которых узлы не являются перестановками, что так же влечет за собой снижение стойкости;
- возможность использования преднамеренно слабых S-блоков.

Контрольные вопросы:

1) **Что такое ГОСТ 28147-89?**
2) **На чем основывается ГОСТ 28147-89?**
3) **Объясните принцип работы ГОСТ 28147-89.**
4) **Чем отличаются S-блоки ГОСТ 28147-89от S-блоков DES?**

5) **В чем заключается схожесть ГОСТ 28147-89 и DES?**
6) **В чем различие ГОСТ 28147-89 и DES?**
7) **В каких режимах может быть использован ГОСТ 28147-89?**
8) **Что такое S-блок ГОСТ 28147-89?**
9) **Объясните рисунок 3.8.1.**
10) **Объясните принцип работы S-блока в таблице 3.8.1.**
11) **Назовите достоинства ГОСТ 28147-89.**
12) **Назовите недостатки ГОСТ 28147-89.**
13) **Когда и кем был создан ГОСТ 28147-89?**
14) **Каким образом происходит генерация ключей ГОСТ 28147-89?**

3.9 Атаки на блочные шифры

Блочные, как и поточные шифры подвержены атакам, как мы уже говорили, все шифры создаются с целью защитить какую-либо информацию, обеспечить ее конфиденциальность.

Все основные атаки применимы как к блочным, так и к поточным алгоритмам, но существуют специфические атаки, применяемые только для блочных шифров:

- атака с использованием только зашифрованного текста;
- атака с известным открытым текстом;
- атака с избранным открытым текстом;
- атака с избранным шифрованным текстом;
- атака, с использованием парадокса задачи о днях рождениях
- двухсторонняя атака;
- атака со связанным ключом;
- атака с избранным ключом;
- усеченный дифференциальный криптоанализ.

Последние три вида атак характерны именно для блочных шифров, и именно о них мы в первую очередь и поговорим.

Атака со связанным ключом уже в довольно преклонном возрасте, ей уже более 100 лет, и основана она на простом предположении о том, что

криптоаналитик имеет доступ к нескольким функциям шифрования. Все эти функции работают с неизвестными ключами, однако ключи связаны определенным соотношением, которое известно криптоаналитику. На практике большинство систем работают с разными ключами, связанным известным соотношением. Так, например, каждый последующий ключ может быть связан с предыдущим путем увеличения последнего на N значений. Зачастую эти связи более сложны и успешность атаки напрямую будет зависеть от того насколько сложна данная связь. На основе атаки со связанным ключом построена атака с избранным ключом, в данном случае криптоаналитик задает часть ключа, а на оставшуюся часть ключа выполняет атаку связанным ключом.

Усеченный дифференциальный криптоанализ – атака на блочные шифры, использующая обобщения дифференциального криптоанализа. В то время как обычный дифференциальный анализ использует разность между двумя полными текстами, в усеченном криптоанализе рассматривается разность между частями текста. Поэтому с помощью этой атаки можно предсказать значения лишь некоторых бит, а не целого блока. Данный вид атак был сформулирован относительно недавно, всего в 1994 году, но уже не раз применялся для атак на блочные шифры, правда с переменным успехом.

Рассмотрим оставшиеся криптоатаки из нашего списка. Атаки с использованием только шифрованного текста довольно просты для понимания, в данном случае криптоаналитик пытается расшифровать сообщение только при наличии шифрованного текста. При атаке с известным открытым текстом известен также и шифрованный текст, в данном случае цель атаки направлена на получение ключа. При атаке с избранным открытым текстом криптоаналитик может самостоятельно подбирать открытый текст. Имеется возможность отсылать любое количество простых текстов и получать в ответ соответствующие шифрованные тексты. Существуют автономная и оперативная виды атак. В первом случае выбор открытых текстов подготавливается заранее, до получения шифрованных текстов. Во втором случае каждый последующий открытый текст выбирается исходя из уже полученных шифрованных текстов. Атака с избранным шифрованным текстом основывается на возможности криптоаналитика подбирать как открытый, так и шифрованный текст. Для каждого подобранного открытого текста криптоаналитик получает шифрованный текст, для каждого подобранного шифрованного текста – соответствующий открытый текст. Атака, в основе которой лежит парадокс задачи о днях рождениях, пожалуй является самой интересной из всех перечисленных и для того чтобы понять в чем суть данной атаки сначала необходимо разобраться с самим парадоксом. Суть задачи гласит, что для любой группы людей из 23 человек и более, вероятность совпадения дней рождения (числа и месяца) хотя бы двух членов группы превышает 50%. Парадоксально, но лишь с очки зрения интуитивного восприятия, в математическом смысле логических противоречий тут нет. Таким

образом, наша атака будет основываться на том принципе, что одинаковые значения появляются быстрее, чем можно было ожидать.

Двусторонняя атака заключается в том, что аналитик строит таблицу ключей выбранных самостоятельно. Различие между атакой, в основе которой лежит парадокс о днях рождениях, и двусторонней атакой в том, что в первом случае криптоаналитик ждет, когда одно и то же значение появится дважды во множестве элементов, в двусторонней атаке он ждет, когда два множества пересекутся.

Хотелось бы уделить особое внимание XSL – атакам, мы уже говорили о них во второй главе данного пособия, но более детально разберем именно сейчас. Дело в том, что до сих пор не доказана возможность осуществления этой атаки на поточные шифры, ведь наличие такой возможности ставит под сомнение целостность практически любого алгоритма. Существуют предположения о том, что если XSL – атаки реальны, и способны расшифровывать поточные криптоалгоритмы, то такие атаки взломают практически любой шифр.

XSL – атака – метод, основанный на алгебраических свойствах шифра, предполагает решение особой системы уравнений. Результат работы S–блоков системы с многораундовым шифрованием записывается в виде уравнения:

$$\sum_{i,j} a_{ij} * x_i * x_j + \sum_{i,j} b_{ij} * y_i * y_j + \sum_{i,j} \gamma_{ij} * x_i * y_j + \sum_{i} \delta_i * x_i + \sum_{i} \epsilon_i * x_i + \eta = 0$$

Где x_i, y_i – соответственно биты на входе и выходе S–блоков i–того раунда шифрования. Далее для различных значений входных текстов и соответствующих им шифротекстов составляются таблицы истинности, на основе которых определяется значение ключа системы. Таким образом, решив данную систему уравнений, мы можем установить взаимно-однозначное значение ключа системы. Примечательно то, что сроки реализации данной атаки в теории существенно ниже времени полного перебора, что делает проблему изучения данных атак наиболее перспективной в ближайшее время.

Контрольные вопросы:

1) **Какие атаки на блочные шифры вам известны?**
2) **Какие специфические атаки на блочные шифры вам известны?**
3) **Объясните принципы работы известных вам атак на блочные шифры.**
4) **Что такое XSL атака?**
5) **Чем она отличается от дифференциального криптоанализа?**
6) **Чему равен результат работы S-блоков при XSL-атаке?**
7) **Объясните парадокс задачи о днях рождениях.**

3.10 Сравнительный анализ блочных и поточных симметричных алгоритмов шифрования

С момента изобретения симметричных способов шифрования и до появления ассиметричных алгоритмов прошел довольно значительный промежуток времени, шифры эволюционировали, приходили на смену друг другу. Как правило, последующий шифр учитывает недостатки своего предшественника и если не исключал их, то значительно усложнял процесс криптоанализа. Таким постепенным усложнением симметричные криптосистемы не только дошли до нашего времени, но и сохранили целесообразность своего применения, успешно конкурируя с ассиметричными криптосистемами. В настоящее время симметричные шифры разделяются на две концептуально различных реализации – поточный шифр и блочный шифр, и словно два враждующих племени каждый из шифров имеет своих сторонников, так исследованием и разработкой поточных шифров в основном занимаются европейские криптографические центры, в то время как блочных – американские. Таким образом, перед нами открывается весь актуальный вопрос, за каким алгоритмом будущее, какому из алгоритмов отдать предпочтение, при организации системы защиты информации? На эти вопросы мы и постараемся ответить в данном разделе.

Исходя из 2 и 3 главы данного учебного пособия, мы проанализировали основные отличия поточных шифров от блочных:

1. высокая скорость шифрования поточных шифров;
2. отсутствие в поточных шифрах эффекта размножения ошибок;
3. структура поточного ключа имеет уязвимые места, дающие возможность криптоаналитику дополнительную информацию о ключе;
4. высокая эффективность взлома поточных шифров при помощи линейного и дифференциального анализа;
5. разнообразные методы взлома поточных шифров;
6. наличие четких критериев надежности для поточных шифров;
7. более динамичное исследование поточных шифров;
8. большая линейная сложность;
9. каждый бит потока ключей должен быть сложным преобразованием большинства битов ключа.

Таким образом, проведя детальный анализ симметричных алгоритмов шифрования, мы можем с уверенностью заключить, что каждый из алгоритмов

имеет свои отличительные особенности, достоинства и недостатки, однако на наш взгляд наиболее перспективным направлением развития является именно поточные симметричные шифры. Ведь именно к ним относится знаменитый шифр Вернама – единственный алгоритм с доказанной абсолютной криптографической стойкостью.

Контрольные вопросы:

1) **В чем отличие блочных шифров от поточных шифров?**
2) **Какие алгоритмы работают быстрее?**
3) **У каких алгоритмов более высокая линейная сложность?**
4) **Укажите критерии надежности для поточных шифров.**
5) **Изучение каких шифров приоритетно в европейских институтах криптографии?**
6) **Изучение каких шифров приоритетно в американских институтах криптографии?**

ГЛАВА 4

КРИПТОГРАФИЯ ОТКРЫТОГО КЛЮЧА

4.1 Криптография открытого ключа

В традиционных криптографических системах отправитель и получать сообщения использует один и тот же ключ, называемый секретным. Отправитель зашифровывает сообщение секретным ключом, а получатель этим же ключом сообщение расшифровывает. Данный способ шифрования называется методом секретного ключа, или симметричной криптографией. Основная задача при этом состоит в том, что получателю и отправителю необходимо согласовать секретный ключ таким образом, чтобы он не стал известен посторонним. А если получатель и отправитель находятся далеко друг от друга, то возникает необходимость защищать ключ от перехвата при передаче, так как, перехватив ключ при передаче, злоумышленник получает возможность читать, изменять и подделывать сообщения, зашифрованные или подписанные этим ключом. Создание, передача и хранение ключей называется управлением ключами. Все криптосистемы должны иметь средства управления ключами.

Поскольку ключи должны оставаться засекреченными, то при использовании криптографии секретного ключа возникает необходимость защищенного управления ключами, особенно в открытых системах с большим количеством пользователей. Для решения этой задачи в 1976 году Уитфилд Диффи и Мартин Хеллман ввели новую концепцию – криптографию открытого ключа (ассиметричная система).

Криптосистемы открытого ключа имеют две основные области применения: шифрование и цифровые подписи. В таких криптосистемах каждый пользователь получает пару ключей: открытый ключ, которым информация зашифровывается, и частный ключ, которым информация расшифровывается. Причем открытый ключ публикуется открыто, а частный ключ сохраняется владельцем в секрете. Таким образом, устраняется необходимость передачи секретной информации. Пересылаются только открытые ключи, а частные ключи хранятся у владельца и не пересылаются, поэтому криптосистемы открытого ключа позволяют использовать незащищенные каналы. Единственное требование – каждый открытый ключ должен быть строго связан с владельцем (например, в официальном

справочнике). Каждый пользователь криптосистемы может посылать зашифрованное сообщение, используя для этого открытые ключи своих адресатов, но расшифровать такое сообщение может только обладатель частного ключа.

Кроме того, криптосистема открытого ключа используется не только для шифрования, но также для создания цифровых подписей и других задач. В криптосистемах открытого ключа частный ключ всегда математически связан с открытым ключом, поэтому существует возможность атаки системы методом расчета частного ключа на основе открытого ключа. Как правило, защита от такой атаки состоит в высокой сложности (вплоть до невозможности) получения частного ключа на основе открытого ключа. Например, некоторые ассиметричные криптосистемы разработаны таким образом, что вычисление частного ключа на основе открытого требует разложения на множители настолько большого числа, что при современных вычислительных ресурсах это невозможно выполнить. На этом основана криптосистема открытого ключа RSA.

Чтобы послать Бобу секретное сообщение, Алиса находит в справочнике открытый ключ Боба и зашифровывает им отсылаемое сообщение. Получив, сообщение Боб расшифровывает его своим частным ключом Никто другой, даже перехватив сообщение, не может его расшифровать. Каждый может послать Бобу зашифрованное сообщение, но только Боб может прочесть его, потому что только Боб знает собственный частный ключ. Чтобы подписать сообщение цифровой подписью, Алиса выполняет определенную операцию, используя свой частный ключ и само сообщение. Полученная в результате цифровая подпись прилагается к сообщению. Чтобы проверить цифровую подпись на сообщении, присланном якобы Алисой, Боб также выполняет определенные вычисления, используя цифровую подпись и открытый ключ Алисы. Совпадение результатов вычислений говорит о подлинности подписи, расхождение же результатов означает, что либо подпись подделана, либо сообщение было изменено.

Контрольные вопросы:

1) **Что такое открытый ключ?**
2) **Чем симметричное шифрование отличается от ассиметричного?**
3) **На чем основана система RSA?**
4) **Что такое частный ключ?**
5) **Что такое закрытый ключ?**

4.2 Электронная цифровая подпись

Одним из основных понятий в современной защите информации является электронная цифровая подпись или ЭЦП. Электронная цифровая подпись это совокупность прикрепляемой электронной информации к другой, подписываемой информации, применяемой для определения лица подписавшего информацию.

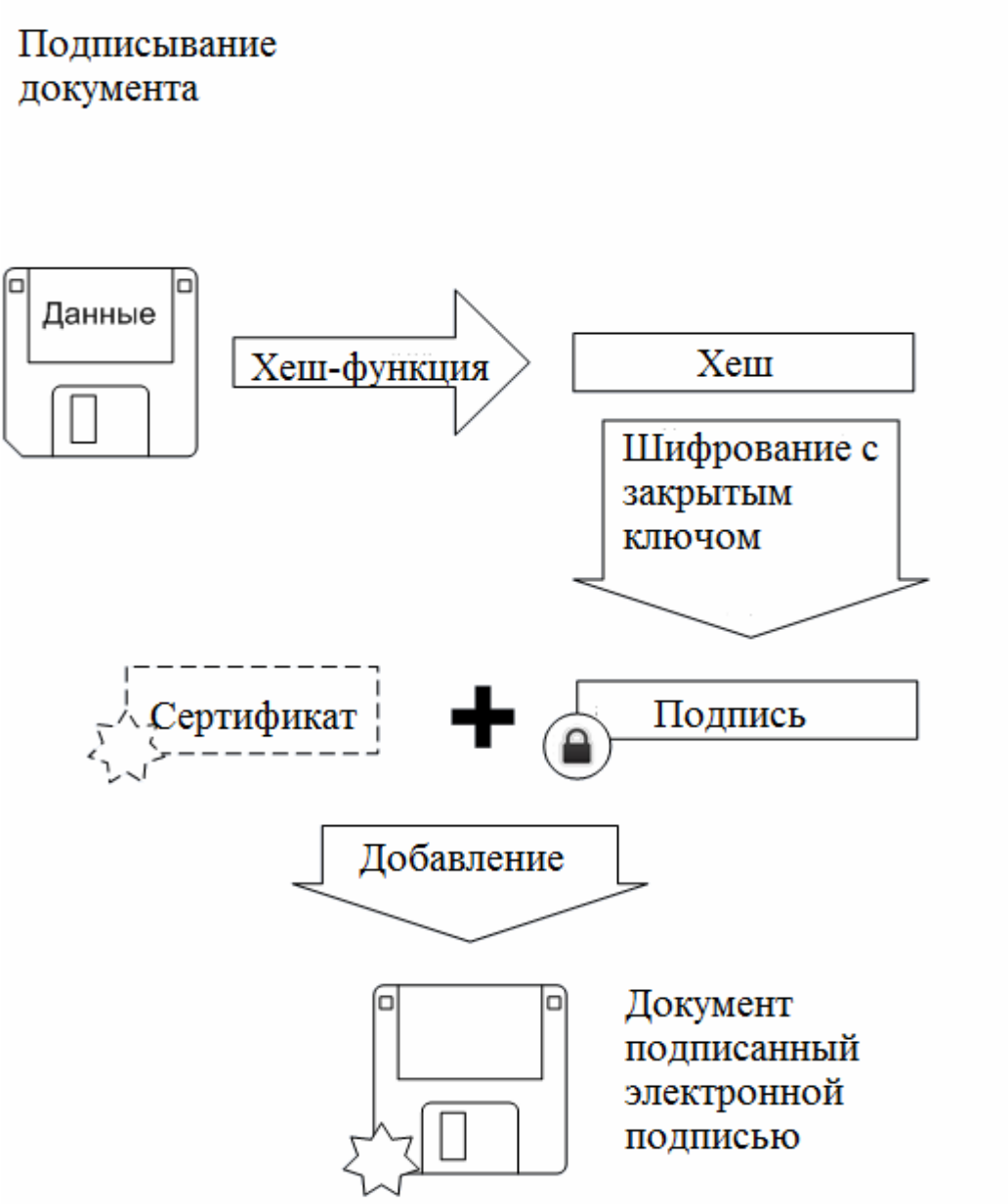

Рисунок 4.2.1 – Схема подписи документа.

По своему существу ЭЦП представляет собой реквизиты электронного документа, позволяющей установить искажение информации в электронном документе с момента формирования ЭЦП и проверить принадлежность подписи владельцу сертификата электронного ключа ЭЦП. Значение такого реквизита получается в результате криптографического преобразования информации с использованием закрытого ключа ЭЦП. ЭЦП предназначена для идентификации лица, подписавшего электронный документ, и является полноценной заменой (аналогом) собственноручной подписи в случаях, предусмотренных законом. Схема подписи и проверки ЭЦП изображена на рисунка 4.2.1 и 4.2.2.

Использование электронной подписи позволяет осуществить:

- Контроль целостности передаваемого документа: при любом случайном или преднамеренном изменении документа подпись станет недействительной, потому что вычислена она на основании исходного состояния документа и соответствует лишь ему;
- Защиту от изменений (подделки) документа: гарантия выявления подделки при контроле целостности делает подделывание нецелесообразным в большинстве случаев;
- Невозможность отказа от авторства. Так как создать корректную подпись можно, лишь зная закрытый ключ, а он должен быть известен только владельцу, то владелец не может отказаться от своей подписи под документом;
- Доказательное подтверждение авторства документа: Так как создать корректную подпись можно, лишь зная закрытый ключ, а он должен быть известен только владельцу, то владелец пары ключей может доказать своё авторство подписи под документом. В зависимости от деталей определения документа могут быть подписаны такие поля, как «автор», «внесённые изменения», «метка времени» и т. д.

Впервые механизм ЭЦП был предложен в 1976 году Мартином Хеллманом и Утфилдом Диффи, но лишь год спустя такой алгоритм был разработан и носил название RSA, подробнее о котором мы поговорим позднее. В соответствии с федеральным законом Российской Федерации №63 – ФЗ «Об электронной подписи» существует два основных вида ЭЦП: простая электронная подпись и усиленная электронная подпись, которая в свою очередь подразделяется еще на два вида: усиленная неквалифицированная электронная подпись и усиленная квалифицированная электронная подпись. Существует несколько схем построения цифровой подписи:

- На основе алгоритмов симметричного шифрования. Данная схема предусматривает наличие в системе третьего лица – арбитра, пользующегося доверием обеих сторон. Авторизацией документа

является сам факт зашифровывания его секретным ключом и передача его арбитру;

– На основе алгоритмов асимметричного шифрования. На данный момент такие схемы ЭЦП наиболее распространены и находят широкое применение.

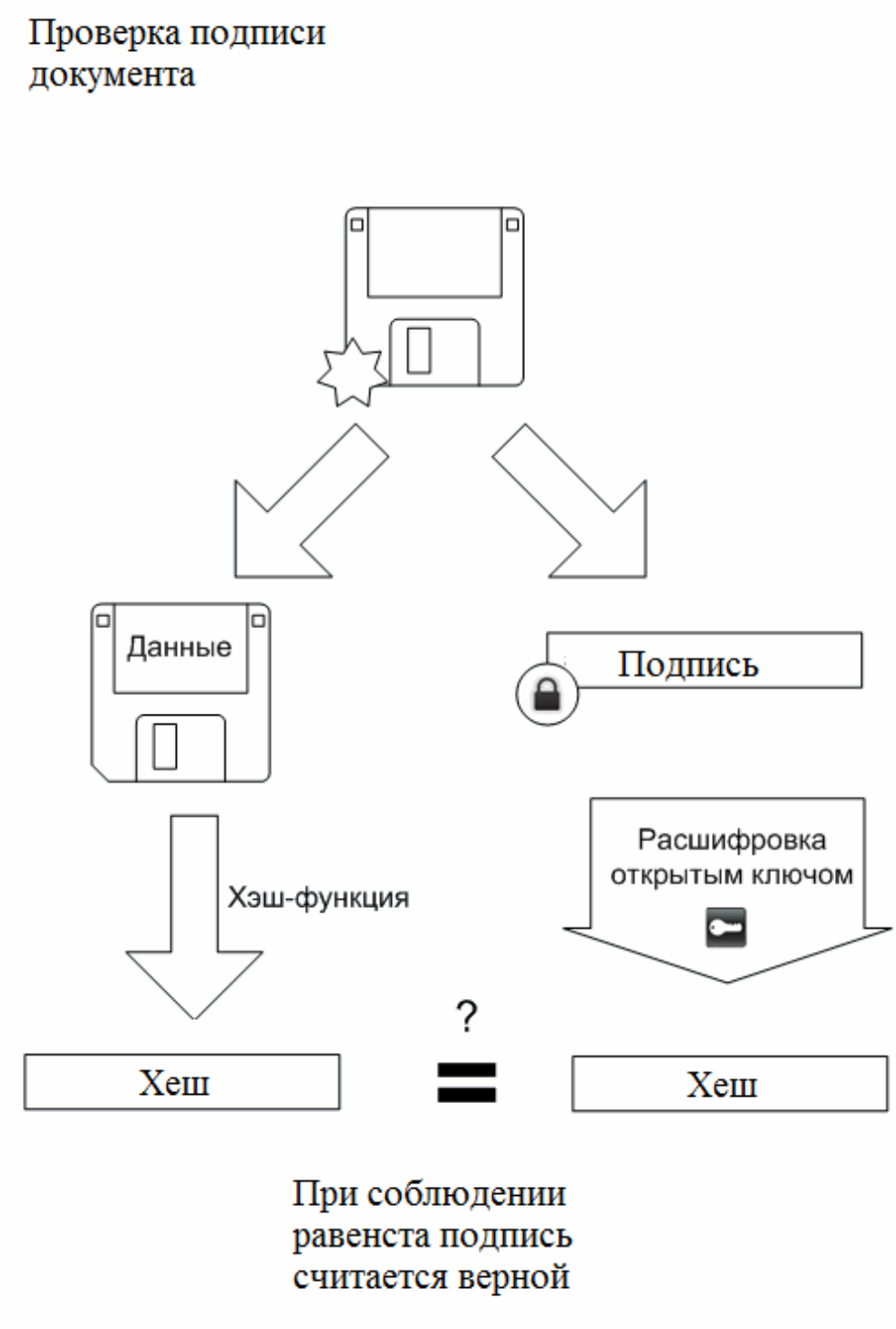

Рисунок 4.2.2 – Схема проверки подлинности ЭЦП.

Симметричные схемы ЭЦП менее распространены чем асимметричные, так как после появления концепции цифровой подписи не удалось реализовать

эффективные алгоритмы подписи, основанные на известных в то время симметричных шифрах. Первыми, кто обратил внимание на возможность симметричной схемы цифровой подписи, были основоположники самого понятия ЭЦП Диффи и Хеллман, которые опубликовали описание алгоритма подписи одного бита с помощью блочного шифра.

Асимметричные схемы цифровой подписи опираются на вычислительно сложные задачи, сложность которых еще не доказана, поэтому невозможно определить, будут ли эти схемы сломаны в ближайшее время. Как это произошло со схемой, основанной на задаче об укладке ранца. Также для увеличения криптостойкости нужно увеличивать длину ключей, что приводит к необходимости переписывать программы, реализующие асимметричные схемы, и в некоторых случаях перепроектировать аппаратуру. Симметричные схемы основаны на хорошо изученных блочных шифрах.

В связи с этим симметричные схемы имеют следующие преимущества:

– Стойкость симметричных схем ЭЦП вытекает из стойкости используемых блочных шифров, надежность которых также хорошо изучена;

– Если стойкость шифра окажется недостаточной, его легко можно будет заменить на более стойкий с минимальными изменениями в реализации.

Однако у симметричных ЭЦП есть и ряд недостатков:

– Нужно подписывать отдельно каждый бит передаваемой информации, что приводит к значительному увеличению подписи. Подпись может превосходить сообщение по размеру на два порядка;

– Сгенерированные для подписи ключи могут быть использованы только один раз, так как после подписания раскрывается половина секретного ключа.

Из-за рассмотренных недостатков симметричная схема ЭЦП Диффи-Хелмана не применяется, а используется её модификация, разработанная Березиным и Дорошкевичем, в которой подписывается сразу группа из нескольких бит. Это приводит к уменьшению размеров подписи, но к увеличению объема вычислений.

Асимметричные схемы ЭЦП относятся к криптосистемам с открытым ключом. В отличие от асимметричных алгоритмов шифрования, в которых зашифровывание производится с помощью открытого ключа, а расшифровывание – с помощью закрытого, в схемах цифровой подписи подписание производится с применением закрытого ключа, а проверка – с применением открытого.

Общепризнанная схема цифровой подписи охватывает три процесса:

– Генерация ключевой пары. При помощи алгоритма генерации ключа равновероятным образом из набора возможных закрытых ключей выбирается закрытый ключ, вычисляется соответствующий ему открытый ключ;

– Формирование подписи. Для заданного электронного документа с помощью закрытого ключа вычисляется подпись;

– Проверка (верификация) подписи. Для данных документа и подписи с помощью открытого ключа определяется действительность подписи.

Для того чтобы использование цифровой подписи имело смысл, необходимо выполнение двух условий:

– Верификация подписи должна производиться открытым ключом, соответствующим именно тому закрытому ключу, который использовался при подписании;

– Без обладания закрытым ключом должно быть, вычислительно сложно создать легитимную цифровую подпись.

Важной проблемой всей криптографии с открытым ключом, в том числе и систем ЭЦП, является управление открытыми ключами. Так как открытый ключ доступен любому пользователю, то необходим механизм проверки того, что этот ключ принадлежит именно своему владельцу. Необходимо обеспечить доступ любого пользователя к подлинному открытому ключу любого другого пользователя, защитить эти ключи от подмены злоумышленником, а также организовать отзыв ключа в случае его компрометации.

Задача защиты ключей от подмены решается с помощью сертификатов. Сертификат позволяет удостоверить заключённые в нём данные о владельце и его открытый ключ подписью какого–либо доверенного лица. Существуют системы сертификатов двух типов: централизованные и децентрализованные. В децентрализованных системах путём перекрёстного подписания сертификатов знакомых и доверенных людей каждым пользователем строится сеть доверия. В централизованных системах сертификатов используются центры сертификации, поддерживаемые доверенными организациями.

Центр сертификации формирует закрытый ключ и собственный сертификат, формирует сертификаты конечных пользователей и удостоверяет их аутентичность своей цифровой подписью. Также центр проводит отзыв истекших и компрометированных сертификатов и ведет базы выданных и отозванных сертификатов. Обратившись в сертификационный центр, можно получить собственный сертификат открытого ключа, сертификат другого пользователя и узнать, какие ключи отозваны.

Закрытый ключ является наиболее уязвимым компонентом всей криптосистемы цифровой подписи. Злоумышленник, укравший закрытый ключ

пользователя, может создать действительную цифровую подпись любого электронного документа от лица этого пользователя. Поэтому особое внимание нужно уделять способу хранения закрытого ключа. Пользователь может хранить закрытый ключ на своем персональном компьютере, защитив его с помощью пароля. Однако такой способ хранения имеет ряд недостатков, в частности, защищенность ключа полностью зависит от защищенности компьютера, и пользователь может подписывать документы только на этом компьютере.

В настоящее время существуют следующие устройства хранения закрытого ключа:

- дискеты;
- смарт-карты;
- USB-брелоки;
- таблетки Touch-Memory;

Кража или потеря одного из таких устройств хранения может быть легко замечена пользователем, после чего соответствующий сертификат может быть немедленно отозван.

Наиболее защищенный способ хранения закрытого ключа – хранение на смарт-карте. Для того, чтобы использовать смарт-карту, пользователю необходимо не только её иметь, но и ввести PIN-код, то есть, получается двухфакторная аутентификация. После этого подписываемый документ или его хеш передается в карту, её процессор осуществляет подписание хеша и передает подпись обратно. В процессе формирования подписи таким способом не происходит копирования закрытого ключа, поэтому все время существует только единственная копия ключа. Кроме того, произвести копирование информации со смарт-карты сложнее, чем с других устройств хранения. В соответствии с законом «Об электронной подписи», ответственность за хранение закрытого ключа владелец несет сам. В России юридически значимый сертификат электронной подписи выдаёт удостоверяющий центр. Правовые условия использования электронной цифровой подписи в электронных документах регламентирует Федеральный закон Российской Федерации от 6 апреля 2011 г. № 63-ФЗ «Об электронной подписи».

После становления ЭЦП при использовании в электронном документообороте между кредитными организациями и кредитными бюро в 2005 году активно стала развиваться инфраструктура электронного документооборота между налоговыми органами и налогоплательщиками. Начал работать приказ Министерства по налогам и сборам РФ от 2 апреля 2002 г. № БГ-3-32/169 «Порядок представления налоговой декларации в электронном виде по телекоммуникационным каналам связи». Он определяет общие

принципы информационного обмена при представлении налоговой декларации в электронном виде по телекоммуникационным каналам связи. В законе РФ от 10 января 2002 г. № 1-ФЗ «Об электронной цифровой подписи» описаны условия использования ЭЦП, особенности её использования в сферах государственного управления и в корпоративной информационной системе.

Благодаря ЭП теперь, в частности, многие российские компании осуществляют свою торгово–закупочную деятельность в Интернете, через системы электронной торговли, обмениваясь с контрагентами необходимыми документами в электронном виде, подписанными ЭЦП. Это значительно упрощает и ускоряет проведение конкурсных торговых процедур. С 1 июля 2012 года Федеральный закон от 10 января 2002 г. № 1-ФЗ утратит силу, на смену ему придет Федеральный закон от 6 апреля 2011 г. № 63-ФЗ «Об электронной подписи», поэтому приемлемо считать понятия ЭЦП и ЭП эквивалентными.

Контрольные вопросы:

1) **Что такое ЭЦП?**
2) **Какими нормативно–правовыми документами устанавливаются основные действия с ЭЦП в России?**
3) **Когда и кем впервые была предложена идея ЭЦП?**
4) **В чем принцип действия ЭЦП?**
5) **Назовите основные существующие подходы к созданию ЭЦП?**
6) **Объясните рисунок 4.2.1.**
7) **Объясните рисунок 4.2.2.**
8) **Перечислите основные ключевые носители данных.**
9) **Каковы условия использования ЭЦП?**
10) **Какие общие процессы охватываются при генерации ЭЦП?**
11) **Каким условиям должен соответствовать документ с ЭЦП, чтобы его подлинность не вызывала сомнений?**
12) **Чем обуславливается надежность ЭЦП?**

4.3 Криптосистема Elgamal

Схема шифрования Эль-Гамаля – криптосистема с открытым ключом, в основе которой лежит проблема трудности вычисления дискретных логарифмов в конечном поле. Криптосистема включает в себя алгоритм шифрования и алгоритм ЭЦП. Схема была предложена Тахером Эль-Гамалем в 1984 году.

Эль-Гамаль разработал один из вариантов алгоритма Диффи-Хеллмана. Он усовершенствовал систему Диффи-Хеллмана и получил два алгоритма, которые использовались для шифрования и для обеспечения аутентификации. В отличие от RSA алгоритм Эль-Гамаля не был запатентован и, поэтому, стал более дешевой альтернативой, так как не требовалась оплата взносов за лицензию. Считается, что алгоритм попадает под действие патента Диффи-Хеллмана.

При создании ключей по системе Эль-Гамаля, сперва генерируется случайное простое число p длинной в n бит, после чего выбирается произвольное число g, являющееся первообразным корнем по модулю p. Далее мы выбираем случайное целое число x такое, что 1<x<p, затем вычисляется $y = g^x mod\ p$. Таким образом, открытым ключом будет являться тройка (p, g, y), а закрытым ключом число x. Шифросистема Эль-Гамаля является фактически одним из способов выработки открытых ключей Диффи-Хеллмана. Схема шифрования Эль-Гамаля представлена на рисунке 4.3.1.

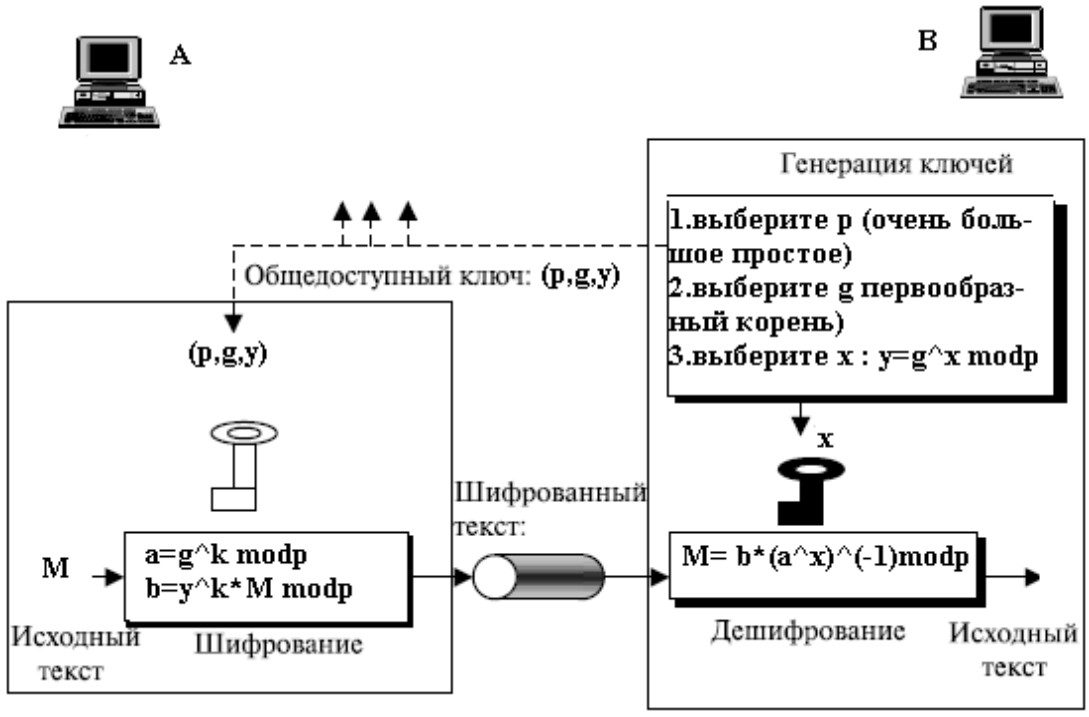

Рисунок 4.3.1 – Схема шифрования Эль-Гамаля.

При реализации алгоритма сообщение M шифруется следующим образом:

- Выбираем сессионный ключ – случайное число k, 1<k<p–1;
- Вычисляем $a = g^k mod\ p$ и $b = y^k\ M\ mod\ p$;
- Пара чисел (a, b) будет являться шифротекстом.

Следует заметить, что получившийся шифротекст в два раза больше исходного текста. Расшифровывание происходит в аналогичной

последовательности, зная закрытый ключ, мы преобразуем шифротекст (a, b) по формуле $M = b(a^x)^{-1} mod\ p$.

Система Эль-Гамаля, как мы уже говорили, может работать в режиме ЭЦП. Цифровая подпись служит для того чтобы можно было установить изменения данных и чтобы установить подлинность подписавшейся стороны. Получатель подписанного сообщения может использовать цифровую подпись для доказательства третьей стороне того, что подпись действительно сделана отправляющей стороной. При работе в режиме подписи предполагается наличие фиксированной хеш-функцииh, лежащей в интервале (1, p–1).

Для подписи сообщения M нам необходимо вычислить хеш сообщения M: m=h(M), затем выбираем случайное число k, так чтобы 1<k<p–1 взаимно простое с p–1 и вычисляется $r = g^k\ mod$. С помощью расширенного алгоритма Евклида вычисляем число s, удовлетворяющее сравнению m≡ xr+ks (modp –1), таким образом, подписью для сообщения M является пара (r,s). Схема ЭЦП представлена на рисунке 4.3.2.

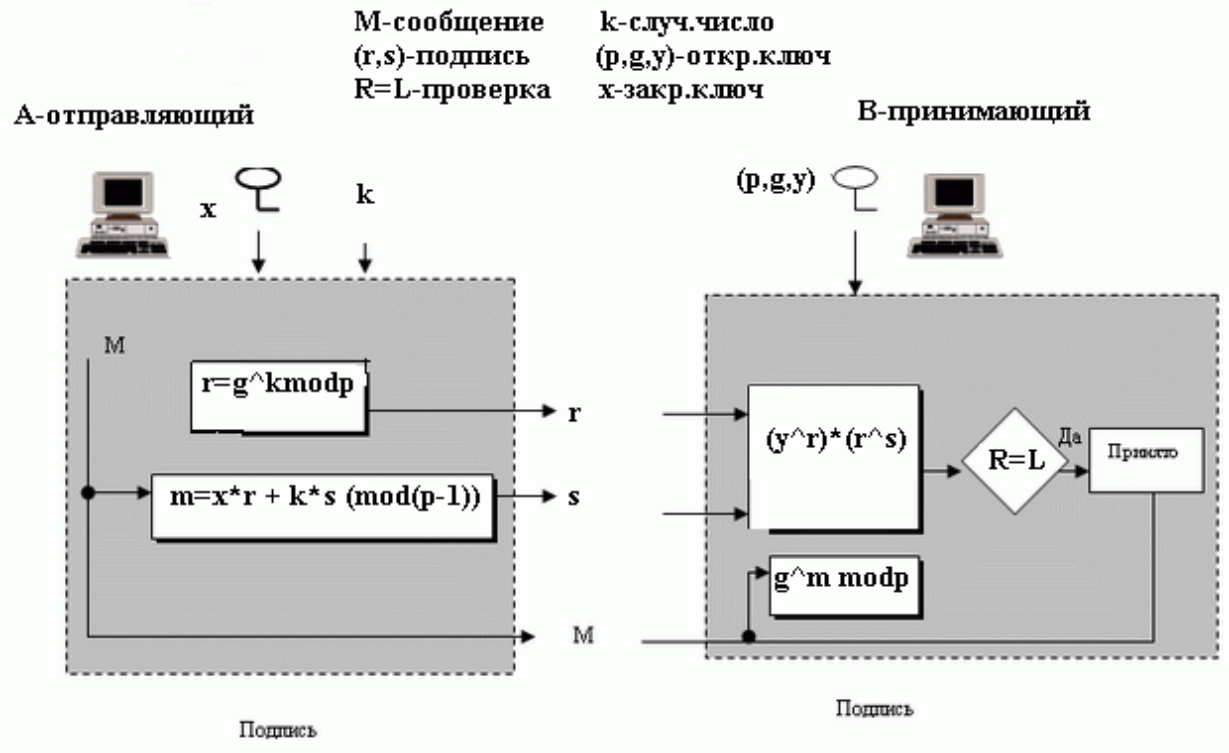

Рисунок 4.3.2 – Схема ЭЦП по системе Эль-Гамаля.

Зная открытый ключ (p,g,y), подпись (r,s) сообщения M проверяется по следующей схеме:

- Проверяется выполнимость условий 0<r<p и 0 <s<p–1 , если хотя бы одно из условий не выполняется, то подпись считается неверной;
- вычисляется хеш m= h (M);

– подпись считается верной, если выполняется сравнение$y^r r^s \equiv g^m (mod p)$.

Главным преимуществом схемы цифровой подписи Эль–Гамаль является возможность вырабатывать цифровые подписи для большого числа сообщений с использованием только одного секретного ключа. Чтобы злоумышленнику подделать подпись, ему нужно решить сложные математические задачи с нахождением логарифма в поле.

Контрольные вопросы:

1) **Что такое система Эль-Гамаля?**
2) **В каких режимах работает криптосистема Эль-Гамаля.**
3) **На чем основывается принцип работы алгоритма Эль-Гамаля?**
4) **В чем отличие RSA от криптосистемы Эль-Гамаля?**
5) **Объясните схему шифрования, изображенную на рисунке 4.3.1.**
6) **Объясните схему ЭЦП изображенную на рисунке 4.3.2.**
7) **Зашифруйте сообщение M=5, используя значения p = 11, g = 2 , x = 8.**

4.4 ГОСТ 34.10-2001

ГОСТ Р 34.10-2001. «Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи» – российский стандарт, описывающий алгоритмы формирования и проверки электронной цифровой подписи. Принят и введён в действие Постановлением Госстандарта России от 12 сентября2001 года.

Данный алгоритм разработан главным управлением безопасности связи Федерального агентства правительственной связи и информации при Президенте Российской Федерации при участии Всероссийского научно-исследовательского института стандартизации. Цель написания данного ГОСТа это установление и приведение в соответствие с современными технологиями процесс создания и использования электронной цифровой подписи в России. По данному ГОСТу ЭЦП позволяет:

– аутентифицировать лицо, написавшее сообщение;

– контролировать целостность сообщения;

– защищать сообщение от подделок;

– установить и доказать авторство лица, подписавшего сообщение.

Основой работы ГОСТа лежат эллиптические кривые, а его стойкость основывается на сложности вычисления дискретного логарифма в группе точек эллиптической кривой, а также на стойкости хеш-функций, которые перекачивали к нему от его предшественника – ГОСТа 34.11-94. После подписания сообщения M к нему дописывается цифровая подпись размером 512 бит и текстовое поле. В текстовом поле могут содержаться данные о дате и времени отправки и различные другие данные об отправителе. Данный ГОСТ не описывает механизм генерации параметров, необходимых для формирования подписи, а только определяет, каким образом на основании таких параметров получить цифровую подпись. Механизм генерации параметров определяется на месте в зависимости от разрабатываемой системы. Схема ЭЦП согласно ГОСТу представлена на рисунке 4.4.1.

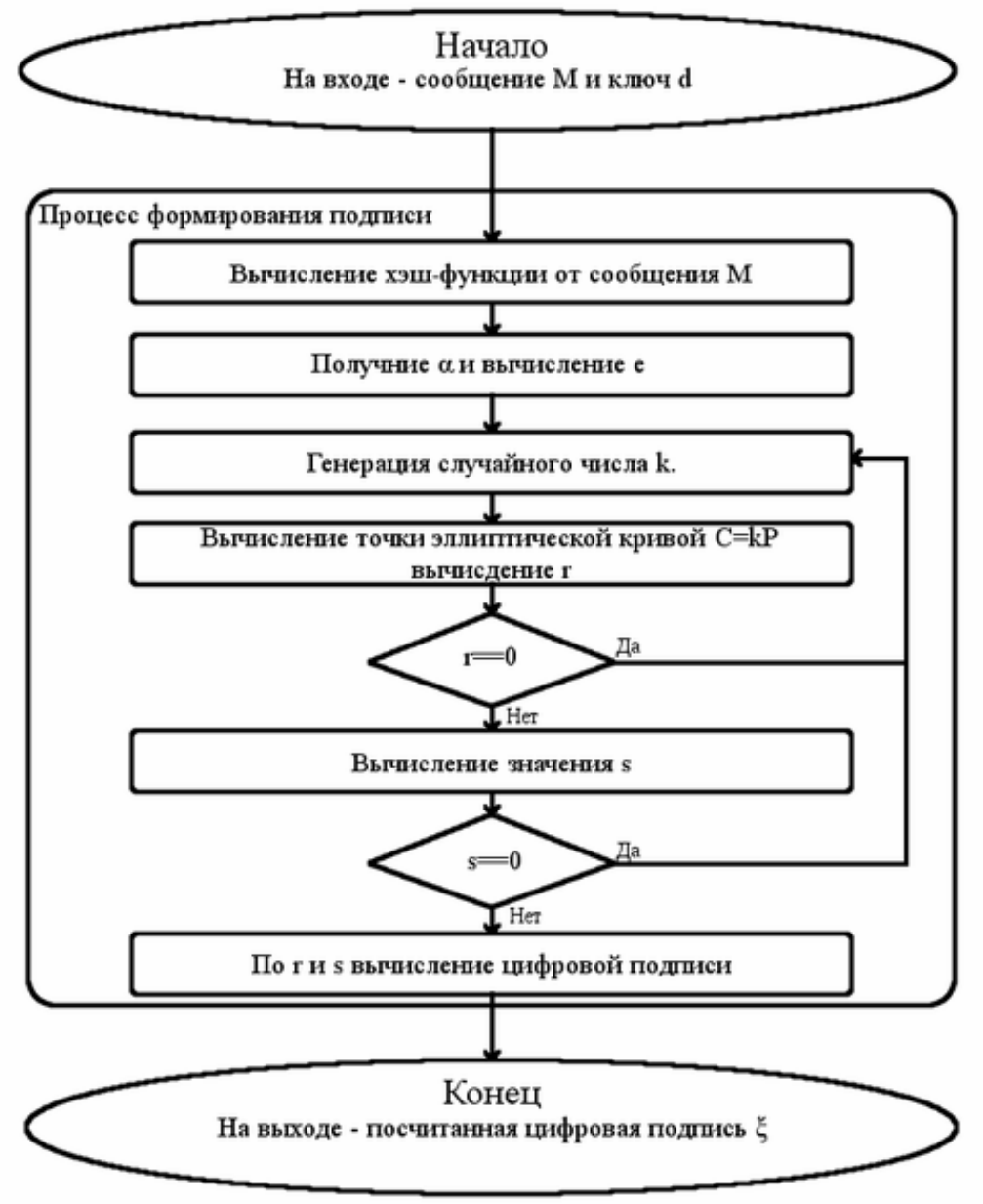

Рисунок 4.4.1 – Схема ЭЦП ГОСТ 34.10–2001.

В процессе формирования ЭЦП сперва вычисляется хеш h от сообщения M, после чего вычисляется $e = z \bmod q$. Далее мы генерируем случайное число k, так чтобы 0<k<q. Далее мы вычисляем точку эллиптической кривой отличную от 0, и вычисляем значение S, которое также должно быть отличным от нуля. После чего происходит непосредственная генерация подписи по двум векторам соответствующим r и s, где r – координата точки на эллиптической кривой. Схема проверки ЭЦП согласно ГОСТу представлена на рисунке 4.4.2.

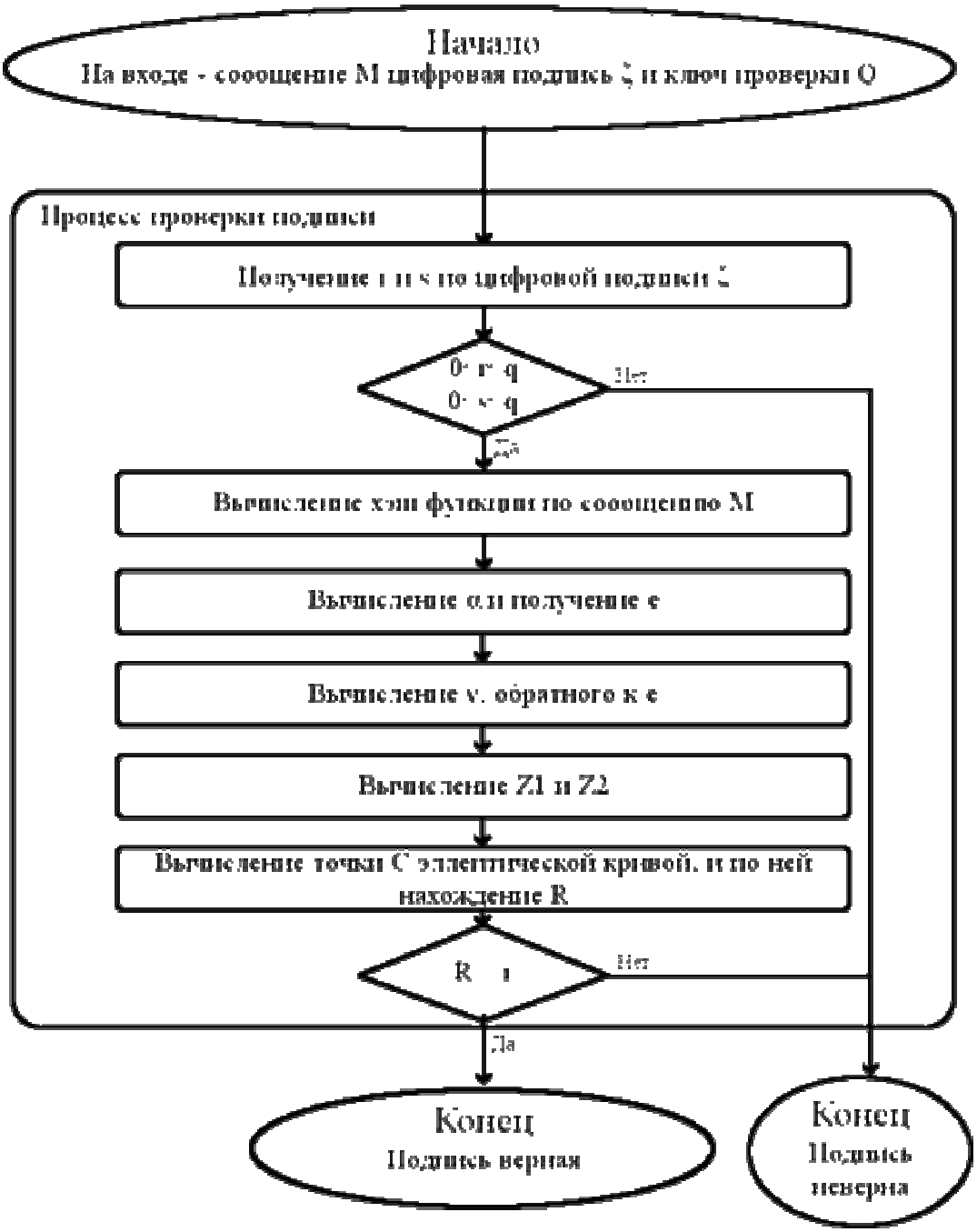

Рисунок 4.4.2 – Схема проверки ЭЦП ГОСТ 34.10–2001.

Проверка ЭЦП согласно ГОСТу происходит в обратной последовательности, так как это изображено на рисунке. Наряду с

установленными способами применения данного ГОСТа, существуют еще несколько возможных способов применения:

- использование пары ключей для установления ключа сессии;
- использование в сертификатах открытых ключей;
- использование в других протоколах;
- использование для защиты в SSL и HTTPS протоколах передач данных;
- при защите целостности Интернет адресов и имен.

Контрольные вопросы:

1) **Что такое ГОСТ 34.10-2001?**
2) **На чем основан принцип работы ГОСТа?**
3) **В чем его отличие от ЭЦП по системе Эль-Гамаля?**
4) **Для чего предназначен ГОСТ 34.10-2001?**
5) **Какие преобразования возможны над ГОСТ 34.10-2001?**
6) **В качестве чего возможно использовать ГОСТ 34.10-2001?**
7) **Объясните рисунок 4.4.1.**
8) **Объясните рисунок 4.4.2.**
9) **Какие операции позволяет выполнять ГОСТ 34.10-2001?**

4.5 DSS

DSS – американский стандарт, описывающий Digital Signature Algorithm, который может быть использован для генерации цифровой подписи. Цифровая подпись служит для установления изменений данных и для установления подлинности подписавшейся стороны. По сути, весь гост описывает алгоритм ЭЦП по методу DSA. Подпись создается секретно, но может быть публично проверена. Это означает, что только один субъект может создать подпись сообщения, но любой может проверить её корректность. Алгоритм основан на вычислительной сложности взятия логарифмов в конечных полях. Алгоритм был предложен Национальным институтом стандартов и технологий США в августе 1991 и является запатентованным.

Для подписания сообщений необходима пара ключей – открытый и закрытый. При этом закрытый ключ должен быть известен только тому, кто подписывает сообщения, а открытый – любому желающему проверить подлинность сообщения. Также общедоступными являются параметры самого

алгоритма. Для обеспечения такого доступа достаточно авторитетная организация (или несколько организаций) поддерживает базу соответствия между реальными реквизитами автора (это может быть как частное лицо, так и организация) и открытыми ключами, а также всеми необходимыми параметрами схемы цифровой подписи (используемая хеш-функция). Эта организация также выдает цифровые сертификаты.

Для построения системы цифровой подписи, желающий должен произвести следующие действия:

1) Выбор криптографической хеш-функции H(x);
2) Выбор большого простого числа q, размерность которого N в битах совпадает с размерностью в битах значений хеш-функции H(x);
3) Выбор простого числа p, такого, что (p–1) делится на q. Битовая длина p обозначается L;
4) Выбор числа g такого, что его мультипликативный порядок по модулю p равен q. Для его вычисления можно воспользоваться формулой $g = h^{\frac{p-1}{q}} \bmod p$, где h – некоторое произвольное число, $h \in (1; p-1)$ такое, что $g \neq 1$. В большинстве случаев значение h = 2 удовлетворяет этому требованию.

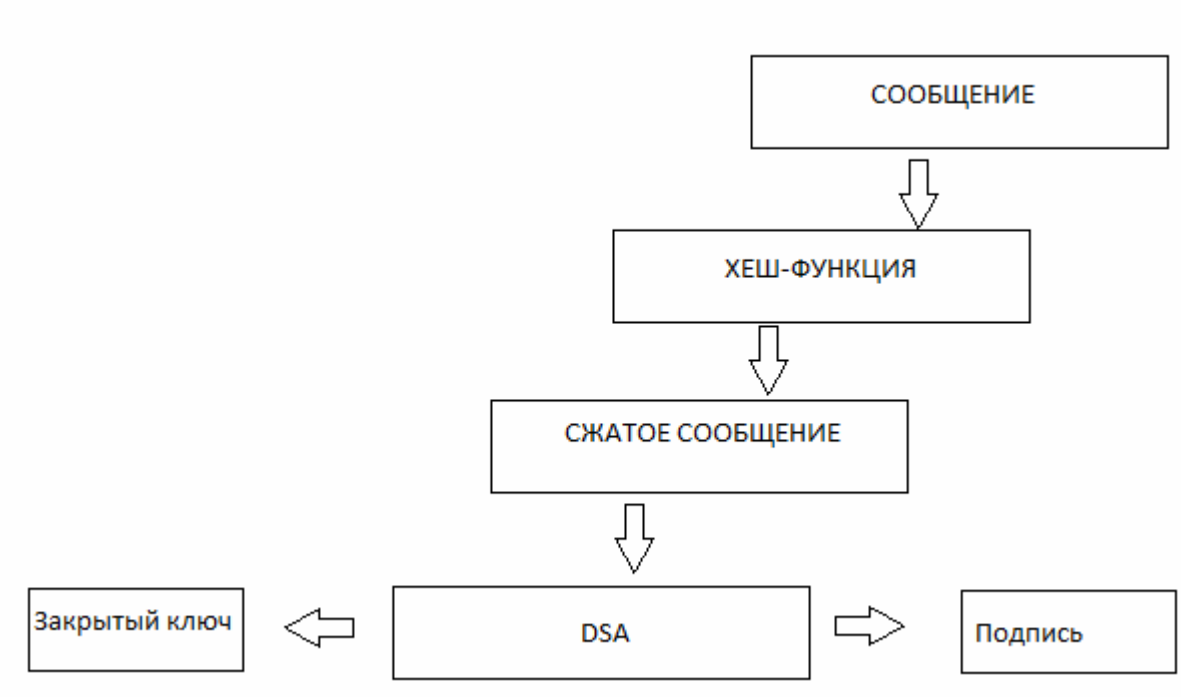

Рисунок 4.5.1 – Алгоритм ЭЦП DSS.

Как мы уже говорили, в DSS первоочередным параметром схемы цифровой подписи является используемая криптографическая хеш-функция, необходимая для преобразования текста сообщения в число, которое

собственно и будет подписано. Открытыми параметрами являются числа (p, q, g, y). Закрытый параметр только один – число x. При этом числа (p, q, g) могут быть общими для группы пользователей, а числа x и y являются соответственно закрытым и открытым ключами конкретного пользователя. При подписании сообщения используются секретные числа x и k, причем число k должно выбираться случайным образом (на практике, как мы уже знаем, псевдослучайным) при подписании каждого следующего сообщения.

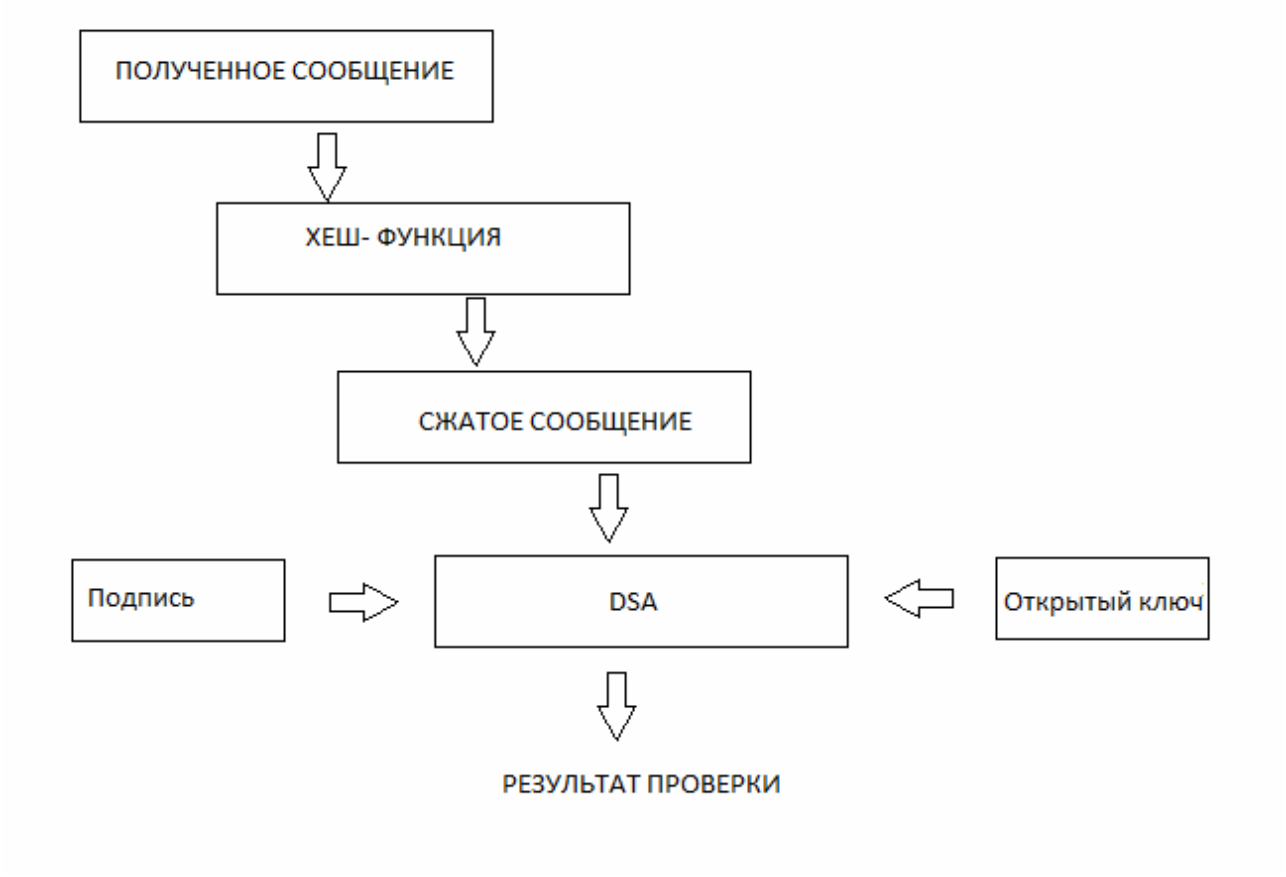

Рисунок 4.5.2 – Алгоритм проверки ЭЦП DSS.

Поскольку (p, q, g) могут быть использованы для нескольких пользователей, на практике часто делят пользователей по некоторым критериям на группы с одинаковыми (p, q, g). Поэтому эти параметры называют доменными параметрами. Схема подписания и проверки подписи представлена на рисунках 4.5.1 и 4.5.2.

Контрольные вопросы:

1) **Что такое DSS?**
2) **Что такое DSA?**
3) **В чем отличие DSA от ГОСТ 34.10-2001?**
4) **В чем заключается принцип действия DSA?**
5) **Объясните рисунок 4.5.1.**

6) Объясните рисунок 4.5.2.

4.6 Протокол обмена ключами Диффи-Хеллмана

Как мы неоднократно убеждались, проблема передачи ключей при использовании систем шифрования с секретными ключами стояла очень остро, однако в 1976 г. Уилтфилду Диффи и Мартину Хеллману удалось решить эту проблемы. Они изобрели протокол, позволяющий передавать секретный ключ по частично защищенному каналу связи, т.е. канал связи защищен от подмены и незащищен от прослушивания. Этот протокол дал начало криптографии открытого ключа.

Суть протокола Диффи-Хеллмана приведена на рисунке 4.6.1.

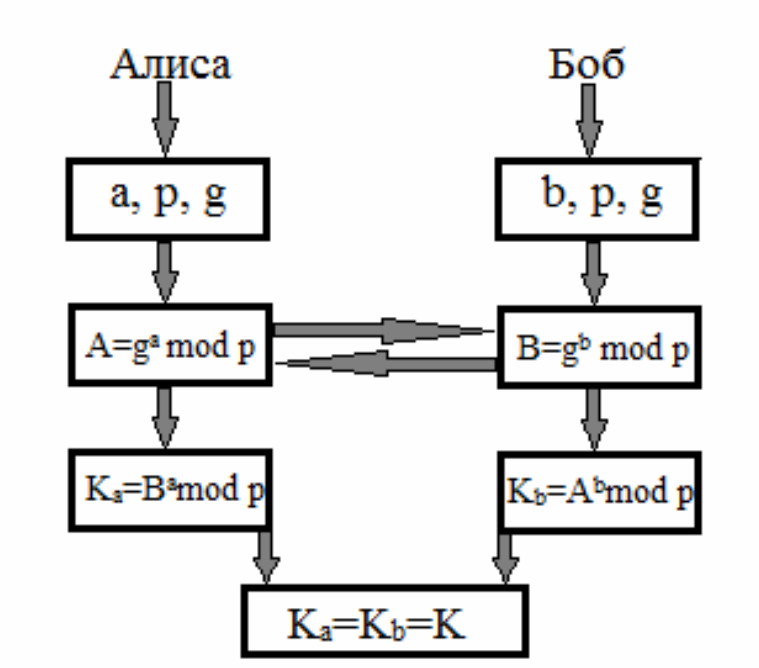

Рисунок 4.6.1 – Алгоритм протокола обмена ключами Диффи–Хеллмана.

У нас есть два пользователя, которым необходимо передать друг другу ключи, пользователь Алиса и пользователь Боб. Им обоим известны несекретные числа p и g.

Для создания секретного ключа оба пользователя генерируют большие случайные числа, Алиса генерирует *a*, Боб – *b* (a и b – это закрытый ключ Алисы и Боба соответственно).

– Далее Алиса вычисляет значение открытого ключа $A=g^a \bmod p$ и передает Бобу. Боб вычисляет открытый ключ $B=g^b \bmod p$ и передает Алисе.

– Затем Алиса вычисляет общий секретный ключ по своему закрытому ключу и открытому ключу Боба $K_a=B^a \text{mod} p$, тоже самое делает Боб – $K_b=A^b \text{mod} p$. Нетрудно убедится, что

$$K_a=B^a \bmod p=(g^b \bmod p)^a \bmod p=g^{ab} \bmod p=(g^a \bmod p)^b \bmod p=A^b \bmod p= K_b$$

– Поскольку $K_a= K_b$, то Алиса и Боб теперь имеют один и тот же секретный ключ *k*. Числа p, g, a и b должны быть достаточно велики.

Стойкость протокола Диффи-Хеллмана заключается в том, что для получения секретного ключа Алисе и Бобу не понадобится много времени, однако, чтобы злоумышленнику получить секретный ключ, ему нужно будет решить задачу дискретного логарифмирования, что является трудноразрешимой математической задачей.

Также данный протокол можно использовать в качестве алгоритма шифрования с открытым ключом. Здесь последовательность действий остается прежней, но с некоторыми отличиями. Так, после создания открытых ключей все пользователи размещают свои открытые ключи в общедоступном справочнике. Данный справочник должен быть заверен специально созданным доверительным центром, чтобы исключить возможные нападения путем подмены открытых ключей или навязывания ложных открытых ключей. После чего сообщение шифруется симметричным алгоритмом, с помощью ключа *k* и передается получателю. Пример такого алгоритма используют в других криптосистемах, например в RSA.

Данный протокол уязвим для атак типа «человек посередине», так как протокол обмена ключами Диффи-Хеллмана не обеспечивает аутентификации пользователей. Злоумышленник, внедрившись в канал связи между Алисой и Бобом, может подменить сообщения. Атака типа «человек посередине» происходит на рисунке 4.6.2.

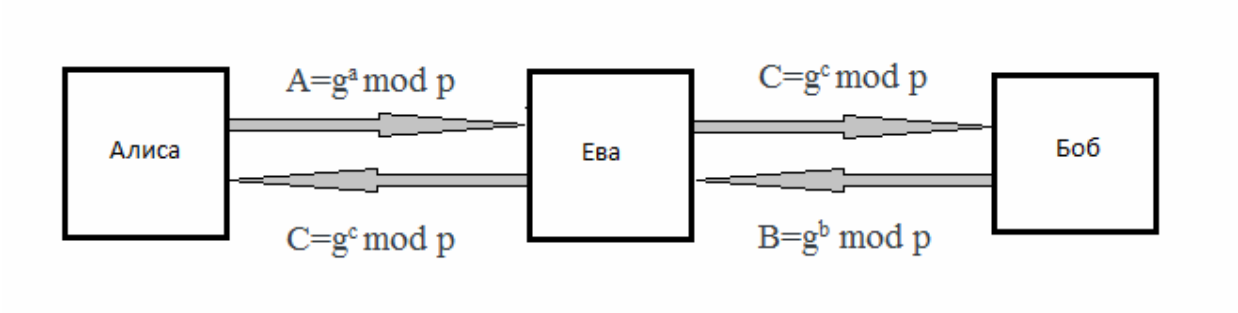

Рисунок 4.6.2 – Атака «человек посередине».

– Злоумышленник (Ева) перехватывает первое сообщение Алисы Бобу и блокирует его. Затем Ева маскируется под Алису и посылает сообщение Бобу.

– Алиса вычисляет и отправляет значение $A=g^a \bmod p$ Бобу (Еве). Боб (Ева) генерирует случайное число – *c*, вычисляет $C=g^c \bmod p$ и отправляет это значение Бобу.

– Боб посылает сообщение Алисе (Еве), это сообщение также блокируется и подменяется.

– Боб вычисляет и отправляет Алисе (Еве) значение $B=g^b \bmod p$. Алиса (Ева) посылает $C=g^c \bmod p$ Алисе.

– Ева и Алиса согласовывают между собой секретный ключ, и Ева и Боб согласовывают секретный ключ, хотя Алиса и Боб считают, что согласовывает ключ между друг другом.

– Алиса вычисляет значение $K_1=C^a \bmod p$. Боб вычисляет значение $K_2=C^b \bmod p$.

Теперь Ева может подменять сообщения, которыми обмениваются Алиса и Боб, а также имитировать их.

Контрольные вопросы:

1) **В чем заключается суть протокола обмена ключами Диффи-Хеллмана?**
2) **На чем основывается принцип атаки «человек посередине?».**
3) **Объясните рисунки 4.6.1 и 4.6.2.**

4.7 Схема идентификации Feige-Fiat-Shamir

Feige-Fiat-Shamir был первым практическим протоколом идентификации. Он минимизировал вычисления, увеличивая число итераций и аккредитаций на итерацию. Для ряда реализаций, например, для интеллектуальных карточек, это не слишком подходит. Обмены с внешним миром требуют времени, а хранение данных для каждой аккредитации может быстро исчерпать ограниченные возможности карточки.

В 1986 году Уриель Фейге, Амос Фиат и Ади Шамир модифицировали алгоритм цифровой подписи, изобретенный ранее Фиатом и Шамиром. Данный протокол стал первым практическим протоколом идентификации и считается лучшим доказательством подлинности с нулевым разглашением. Доказательство с нулевым разглашением – это протокол позволяющий убедить одного субъекта в том, что первый субъект обладает определенной информацией, не раскрывая её. Суть протокола заключается в следующем:

Сначала генерируем открытый и закрытый ключ:

– Генерируем *n*, которое является произведением двух больших простых чисел (длинна модуля *n* должна быть не меньше 512 битов).

– Алиса выбирает число *k* из различных чисел: υ_1, υ_2,… υ_k, где каждое υ_i квадратичный остаток mod n, то есть уравнение $x^2 = \upsilon_i \pmod n$ должно иметь решение, и существовать $\upsilon_i^{-1} \bmod n$. Здесь υ_1, υ_2,… υ_k и есть открытый ключ.

– Далее вычисляются наименьшие значения s_i, такие, что $s_i = \text{sqrt}(\upsilon_i^{-1}) \pmod n$. Здесь строка s_1, s_2,… s_k является закрытым ключом.

После определения ключей переходим не посредственно к самому протоколу, рисунок 4.7.1:

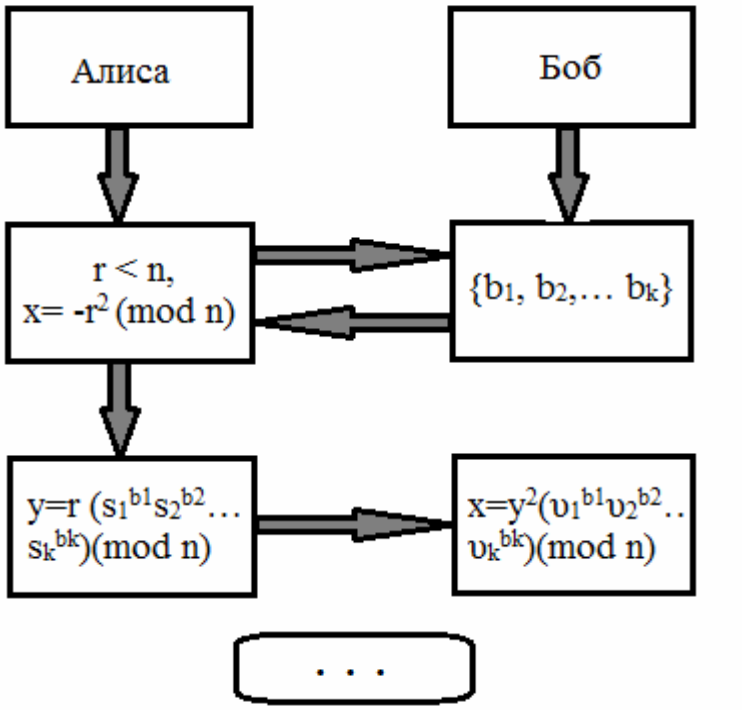

Рисунок 4.7.1 – Схема идентификации Feige-Fiat-Shamir.

– Алиса выбирает случайное число r, такое, что r<n, вычисляет $x = -r^2 \pmod n$ и посылает Бобу.

– Боб посылает Алиса строку из *k* случайных битов: b_1, b_2,… b_k.

– Алиса вычисляет $y = r\,(s_1^{b1} s_2^{b2} \ldots s_k^{bk}) \pmod n$. Алиса перемножает значения s_i, при которых $b_i=1$ (то есть если первый бит Боба $b_1=1$, то s_1 войдет в произведение, если $b_1=0$, то нет, и т.д.). Значение *y* Алиса посылает Бобу.

– Боб проверяет $x = y^2\,(\upsilon_1^{b1} \upsilon_2^{b2} \ldots \upsilon_k^{bk}) \pmod n$. Здесь также перемножаются только те υ_i, при которых $b_i=1$. На этом заканчивается первый этап протокола, называемый аккредитацией.

– Далее Алиса и Боб повторяют протокол до тех пор, пока Боб не убедится, что Алиса знает все s_1, s_2,… s_k.

Вероятность, что Алисе удастся обмануть Боба t раз, равно $1\backslash 2^{kt}$. Рекомендуется брать значение k=5, t=4, в случае необходимости можно увеличить эти значения. В протокол можно внести улучшения, а именно идентификационные данные (имя, телефон, адрес, любая личная информация).

Контрольные вопросы:

1) **Что такое схема идентификации Feige-Fiat-Shamir?**
2) **В чем суть данной схемы?**
3) **Объясните рисунок 4.7.1.**

4.8 RSA

Алгоритм RSA предложили 1978 г. Три автора: Роналдьд Райвест (Ronald Rivest), Ади Шамир (Adi Shamir) и Леонард Адльман (Leonard Adlman). Алгоритма получил свое название по первым буквам фамилий их авторов. Алгоритм RSA стал первым полноценным алгоритмом с открытым ключом, который может работать как в режиме шифрования данных, так и в режиме электронной цифровой подписи. Надежность алгоритма основывается на трудности факторизации больших чисел и трудности вычисления дискретных логарифмов. Первый этап любого асимметричного алгоритма – создание пары ключей – состоит для схемы RSA из следующих операций:

– Выбираются два достаточно больших простых числа p и q (простым называется число, делящееся на единицу и на само себя).

– Вычисляется их произведение, n = pq (n называется модулем).

– Выбирается произвольное число e ($1<e<\varphi(n)$), взаимно простое со значением функции $\varphi(n)$.

– затем вычисляется значение функции Эйлера от числа n:

$$\varphi(n) = (p - 1)(q - 1).;$$

– вычисляется число d, удовлетворяющее условию: $ed = 1 (mod\ \varphi(n))$ и $1 \leq d \leq n-1$. Пара чисел (n, e) – публикуется как открытый ключ. Число (n,d) хранится в строжайшем секрете – это и есть закрытый ключ, который позволит читать все послания, зашифрованные с помощью пары ключей (n, e). Число d – частный показатель; e – открытый показатель.

Второй этап – собственно шифрование с помощью открытого ключа.

– Исходный текст переводится в числовую форму. Далее отправитель разбивает свое сообщение на блоки, лежащие в диапазоне от 0 до n–1. Процесс шифрования одинаков для каждой части. Поэтому можно считать, что исходный текст представлен числом *x* таким, что 0<x<n–1.

– Пользователь A, отправляющий сообщение *x*, шифрует его следующим образом: E(x)=x^e(mod n). Блоки $E(x)$ и есть зашифрованное сообщение. Их можно без опасения передавать по открытому каналу, поскольку операция возведения в степень по модулю простого числа является трудноразрешимой математической задачей.

Третий этап – дешифрование послания с помощью секретного ключа. Пользователь B получает сообщение и расшифровывает его следующим образом.

Пользователь B находит число d такое, что 1≤ d≤ n–1 и $ed = 1(mod\ \varphi(n))$. Это сравнение разрешимо единственным образом, поскольку $(e, \varphi(n)) = 1.$ для решения сравнения $ed = 1(mod\ \varphi(n))$ пользователь B должен вычислить $\varphi(n)$, что для него не составит труда. Так как $\varphi(n) = \varphi(pq) = \varphi(p)\varphi(q) = (p-1)(q-1).$ Любой другой пользователь, который знает только n, вынужден находить p и q, т.е. разлагать число n на простые сомножители. А эта задача при больших p и q является трудноразрешимой математической задачей и имеет большую вычислительную сложность;

Далее, имея в распоряжении число y, пользователь B вычисляет $D(y) = y^d(mod n)$, которая и есть представление *x* исходного текста. Действительно, применяя теорему Эйлера, получаем:

$$y^d = x^{ed} = x^{\varphi(n)k+1} = \left(x^{\varphi(n)}\right)^k x = x(mod\ n).$$

Атаки на RSA можно разбить на три группы это:

– атаки на ключ;

– атаки на сообщение;

– атаки на саму криптосистему.

Атака на секретный ключ состоит в следующем. Необходимо найти главные сомножители (факторы) модуля n, то есть p и q. На основании значений сомножителей и общего показателя (e) злоумышленник может легко вычислить частный показатель d. Здесь основная сложность заключается в поиске сомножителей n, так как факторинг (разложение на множители) является трудноразрешимой математической задачей, не имеющей эффективных способов решения. Безопасность криптосистемы RSA напрямую зависит от факторинга, поэтому к выбору сомножителей (факторам) предъявляются некоторые требования:

– числа p и q должны иметь примерно одинаковую длину;

– числа p и q не должны быть близкими друг другу;

– числа p и q не должны быть слишком малы.

Размер модуля n (n = pq) зависит от требований к безопасности. Чем больше модуль, тем большую безопасности он обеспечивает, однако, большой размер модуля будет также и замедлять работу алгоритма RSA. Поэтому размер модуля определяется из требований безопасности предъявляемых к информации. Так на данный момент рекомендуется брать модуль размером в 768-бит. Ключ размером 1024 бит рекомендуется брать для обычных задач, не требующих высокого уровня защищенности, для особо важных задач рекомендуется брать ключ размером 2048 бит. Усовершенствование вычислительного оборудования не увеличит вероятность взлома криптосистемы, так как с увеличением мощности вычислительной техники будет увеличивать размер ключа и стойкость криптосистемы, также будет увеличиваться.

Атаки на отдельное сообщение заключается в том, что злоумышленник, имеющий зашифрованный текст, предполагает наличие в зашифрованном тексте каких-либо слов, предложений, фраз. Затем зашифровывает эти слова открытым ключом получателя и сравнивает получившийся зашифрованный текст с исходным зашифрованным. Однако есть простой и эффективный способ предотвращения атак такого типа, необходимо просто добавить в конец сообщения несколько случайных битов. Существуют и другие атаки подобного типа.

Контрольные вопросы:

1) **Что такое RSA?**
2) **Кем был придуман RSA?**
3) **Для чего используется RSA?**
4) **В чем принцип работы RSA?**
5) **От чего зависит безопасность RSA?**
6) **На какие группы можно разбить атаки на RSA?**
7) **Каким образом происходит создание секретного ключа?**
8) **Какие основные этапы заключает в себе RSA?**

Содержание

Литература

1. Алфёров А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В Основы криптографии.
2. Бабаш А.В, Шанкин Г.П. Критпография. Под редакцией В.П. Шерстюка, Э.А. Применко / А.В. Бабаш, Г.П. Шанкин. – М.: СОЛОН_ПРЕСС, 2007. – 512 с., ил. – (Серия книг «Аспекты защиты»).
3. Блейхут Р. Теория и практика кодов, контролирующих ошибки = Theory and Practice of Error Control Codes. – М.: Мир, 1986. – 576 с.
4. Венбо Мао Современная криптография. Теория и практика – М.: Вильямс, 2005. – 768 с. – 2 000 экз. – ISBN 5–8459–0847–7, ISBN 0–13–066943–1
5. Герасименко В.А., Малюк А.А. Основы защиты информации. – М.: МГИФИ, 1997.
6. Грушо А.А., Применко Э.А., Тимонина Е.Е. Анализ и синтез криптоалгоритмов. Курс лекций. Москва 2000.
7. Дадуков, Н.С. Советская шифровальная техника [Текст]: ленинградский период: 1935-1941 / Н. С. Дадуков [и др.] // Защита информации. Инсайд. – 2006. – N 1. – С. 91-96. – 2006.
8. Дональд Э. Кнут Глава 3. Случайные числа // Искусство программирования. – 3–е изд. – М.: Вильямс, 2000. – Т. 2. Получисленные алгоритмы. – 832 с. – ISBN 5–8459–0081–6
9. Жельников В. Криптография от папируса до компьютера. М.: ABF, 1996. 336 с.
10. Зима В.М., Молдовян А.А., Молдовян Н.А. Компьютерные сети и защита передаваемой информации. – СПб.: СПбГУ, 1998.
11. Иванов М.А., Чугунков И.В. Глава 4. Методика оценки качества генераторов ПСП // Теория, применение и оценка качества генераторов псевдослучайных последовательностей. – М.: КУДИЦ–ОБРАЗ, 2003. – 240 с. – ISBN 5–93378–056–1

12. Исагулиев К.П. Справочник по криптологии. – Минск: Новое издание, 2004.–237с. ISBN 985–475–079–5.

13. Корн Г., Корн Т. Справочник по математике (для научных работников и инженеров). Пер. с англ./ Под ред. И.Г. Арамановича. М.: Наука, 1973. 832 с.

14. Леонов А.П., Леонов К.П., Фролов Г.В. Безопасность автоматизированных банковских и офисных технологий. – Мн.: Нац. кн. палата Беларуси, 1996.

15. Математические и компьютерные основы криптологии: Учебное пособие / Ю.С. Харин, В.И. Берник, Г.В. Матвеев, С.В. Агиевич. – Минск: Новое издание, 2003. –382с. ISBN 985–475–016–7.

16. Мельников В.В. Защита информации в компьютерных системах. – М.: Финансы и статистика, 1997.

17. Петров А.А. Компьютерная безопасность. Криптографические методы защиты. М.: ЛАЙТ Лтд., 2002.

18. Романец Ю.В., Тимофеев П. А., Шаньгин В.Ф. Защита информации в компьютерных системах и сетях. – М.: Радио и связь, 1999.

19. Ростовцев А.Г., Маховенко Е.Б. Теоретическая критптография. АНО НПО «профессионал», СПб 2004

20. Рябко Б.Я., Фионов А.Н. Криптографические методы защиты информации. – Москва. – Изд-во Горяч. Линия-Телеком, 2005. – ISBN 5–93517–265–8.

21. Саломаа А. Криптография с открытым ключом. Пер. с англ. – М.: Мир, 1995. – 318 с., ил.

22. Смарт Н. Криптография. Серия «Мир программирования». Пер. с англ. С.А. Кулешова / Под ред. С.К. Ландо. М.: Техносфера, 2005. 528 с.

23. Теория электрической связи: учебное пособие / К.К. Васильев, В.А. Глушков, А.В. Дормидонтов, А.Г. Нестеренко; под общ. ред. К.К. Васильева. – Ульяновск: УлГТУ, 2008. – 452 с.

24. Фороузан Б.А.Схема цифровой подписи Эль-Гамаля // Управление ключами шифрования и безопасность сети / Пер. А. Н. Берлин – Курс лекций.

25. Х.К.А. Ван Тилборг. Основы криптологии. Профессиональное руководство и интерактивный учебник. – М.: Мир, 2006, стр. 471

26. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си – М.: Триумф, 2002. – С. 446-448. – 816 с. – 3000 экз. – ISBN 5–89392–055–4.

27. Menezes, P. van Oorschot, S. Vanstone Handbook of Applied Cryptography. – CRC Press, Inc. – 1997.

28. ГОСТ Р 34.10-2001. Информационные технологии. Криптографическая защита информации. Процессы формирования и проверки цифровой подписи.

В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена программа его развития на 2009-2018 годы. В 2011 году Университет получил наименование «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

КАФЕДРА МОНИТОРИНГА И ПРОГНОЗИРОВАНИЯ ИНФОРМАЦИОННЫХ УГРОЗ

Гатченко Нелли Александровна, Исаев Александр Сергеевич,
Яковлев Андрей Дмитриевич

Криптографическая защита информации

Учебное пособие

В авторской редакции

Компьютерный набор и верстка А.С. Исаев

Дизайн обложки М.С. Чичев

Редакционно-издательский отдел НИУ ИТМО

Зав. РИО Н.Ф. Гусарова

Лицензия ИД № 00408 от 05.11.99

Подписано к печати

Заказ №

Тираж 200 экз.

Отпечатано на ризографе