МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Рубцовский индустриальный институт (филиал)
ФГБОУ ВПО «Алтайский государственный технический университет им. И.И. Ползунова»

Н.А. ЛАРИНА

ЗАЩИТА ИНФОРМАЦИИ. КРИПТОЛОГИЯ

Методическое пособие
для бакалавров направления подготовки
09.03.01 – «Информатика и вычислительная техника»
дневной формы обучения

Рубцовск 2014

УДК 004.056

Ларина Н.А. Защита информации. Криптология: Методическое пособие для бакалавров направления подготовки 09.03.01 «Информатика и вычислительная техника» / Рубцовский индустриальный институт. – Рубцовск, 2014. –56 с.

Пособие содержит теоретический, практический и справочный материалы по предмету «Защита информации». В пособие включены описания основных стандартов шифрования данных, алгоритмы и примеры шифрования данных, задания к лабораторным и курсовым работам по данному предмету.

Рассмотрено и одобрено на заседании научно-методического совета Рубцовского индустриального института
Протокол № 9 от 25.12.14

Рецензент: к.ф.-м.н., зав. кафедрой ВМиФХ Г.А. Обухова

Содержание

ВВЕДЕНИЕ

Целью изучения дисциплины является теоретическая и практическая подготовка студентов в области защиты информации, в такой степени, чтобы они могли выбирать необходимые аппаратные, алгоритмические, программные и технологические ресурсы для решения задач защиты компьютерной информации криптографическими методами.

Задача изучения дисциплины заключается в изучении и закреплении на практике решения задач защиты информации, с применением классических и современных стандартов, методов и технологий.

В методическом пособии рассматриваются основные понятия и проблемы уязвимости информации в современных системах обработки данных, дается обзор методов, технических приемов и аппаратуры защиты информации. Основное внимание уделяется криптографическим методам защиты информации, методам защиты от компьютерных вирусов, организационно-правовому обеспечению безопасности информации. Пособие может быть полезно при курсовом и дипломном проектировании, при выполнении лабораторных работ, а также кругу читателей, интересующихся современными проблемами криптографической защиты информации.

Дисциплина «Защита информации» входит в профессиональный цикл базовой части для бакалавров направления 09.03.01 «Информатика и вычислительная техника». Предмет изучается в 7 - 8 семестрах и служит для расширения специальных знаний и умений о защите информации, хранящейся в компьютере и передаваемой по каналам связи. Данная дисциплина тесно связана с дисциплинами «Прикладное программное обеспечение», «Технологии программирования», «Базы данных», изученными ранее.

Таблица 1

Индекс	Наименование	Формы контроля		Часов					ЗЕТ		Курс 4						
				По ЗЕТ	Всего	в том числе			Эксперт-ное	Фактическое	Семестр 7						
		Экзамены	Зачёты			Экз	СРС	Ауд			17 недель						
											Лек	Лаб	Пр	КСР	СРС	Экз	ЗЕТ
Б3.Б8	Защита информации	-	+	85	85	-	34	51	2,25	2,25	34	17	-	-	34	-	2,25

Таблица 2

Индекс	Наименование	Формы контроля		Часов					ЗЕТ		Курс 4						
				По ЗЕТ	Всего	в том числе			Эксперт-ное	Фактическое	Семестр 8						
		Экзамены	Зачёты			Экз	СРС	Ауд			17 недель						
											Лек	Лаб	Пр	КСР	СРС	Экз	ЗЕТ
Б3.Б8	Защита информации	+	-	95	95	+	44	51	2,75	2,75	34	17	-	-	44	+	2,75

Таблица 3

Требования к результатам освоения дисциплины

Содержание компетенции (или её части)	В результате изучения дисциплины обучающиеся должны:		
	знать	уметь	владеть
-готов к кооперации с коллегами, работе в коллективе -стремится к саморазвитию, повышению своей квалификации и мастерства - осознаёт сущность и значение информации в развитии современного общества; владеет основными методами, способами и средствами получения, хранения, переработки информации.	Понятие «защита информации», в том числе «коллективная», её характеристики. Угрозы безопасности. О моделях решения задач защиты компьютерной информации. Алгоритмы защиты. Методы защиты баз данных. Проблемы обеспечения целостности информационных массивов в АС.	Определять необходимые, из имеющихся, средства для защиты информации. Использовать методы и способы защиты в локальных и глобальных сетях ЭВМ, БД, Интернете и электронной почте. Решать элементарные задачи, связанные с кодированием и защитой информации. Настраивать и запускать основные виды ПО, направленного на защиту безопасности компьютерной информации.	Основами защиты компьютерной информации и сведений, составляющих государственную тайну. Методами защиты компьютерной информации. Правила-ми хранения информации на ЭВМ. Навыками настройки ЭВМ и периферийных устройств, направленными на защиту компьютерной информации

ТРАДИЦИОННЫЕ СИММЕТРИЧНЫЕ КРИПТОСИСТЕМЫ

Основные понятия и определения

Большинство средств защиты информации базируются на использовании криптографических шифров и процедур шифрования – расшифрования. В соответствии со стандартом ГОСТ 28147-89 *под шифром понимают совокупность обратимых преобразований множества открытых данных на множество зашифрованных данных, задаваемых ключом и алгоритмом криптографического преобразования.*

Ключ – это конкретное секретное состояние некоторых параметров алгоритма криптографического преобразования данных, обеспечивающее выбор только одного варианта из всех возможных для данного алгоритма.

Основной характеристикой шифра является криптостойкость, которая определяет его стойкость к раскрытию методами криптоанализа. Обычно эта характеристика определяется интервалом времени, необходимого для раскрытия шифра.

К шифрам, используемым для криптографической защиты информации, предъявляется ряд требований:

- достаточная криптостойкость (надёжность закрытия данных);
- простота процедур шифрования и расшифрования;
- незначительная избыточность информации за счёт шифрования;
- нечувствительность к небольшим ошибкам шифрования и др.

В той или иной мере этим требованиям отвечают:

- шифры перестановок;
- шифры замены;
- шифры гаммирования;
- шифры, основанные на аналитических преобразованиях шифруемых данных.

Шифрование перестановкой заключается в том, что символы шифруемого текста переставляются по определённому правилу в пределах некоторого блока этого текста. При достаточной длине блока, в пределах которого осуществляется перестановка, и сложном неповторяющемся порядке перестановки можно достигнуть приемлемой для простых практических приложений стойкости шифра.

Шифрование заменой (подстановкой) заключается в том, что символы шифруемого текста заменяются символами того же или другого алфавита в соответствии с заранее обусловленной схемой замены.

Шифрование гаммированием заключается в том, что символы шифруемого текста складываются с символами некоторой случайной последовательности, именуемой *гаммой шифра*. Стойкость шифрования определяется в основном длиной (периодом) неповторяющейся части гаммы шифра. Поскольку с помощью ЭВМ можно генерировать практически бесконечную гамму шифра, то данный способ является одним из основных для шифрования информации в автоматизированных системах.

Шифрование аналитическим преобразованием заключается в том, что шифруемый текст преобразуется по некоторому аналитическому правилу (формуле).

Например, можно использовать правило умножения вектора на матрицу, причём умножаемая матрица является ключом шифрования (поэтому её размер и содержание должны храниться в секрете), а символами умножаемого вектора последовательно служат символы шифруемого текста. Другим примером может служить использование так называемых однонаправленных функций для построения криптосистем с открытым ключом.

Процессы шифрования и дешифрования осуществляются в рамках некоторой криптосистемы. Характерной чертой симметричной криптосистемы является применение одного и того же секретного ключа как при шифровании, так и при дешифровании сообщения.

Как открытый текст, так и шифртекст образуются из букв, входящих в *конечное множество символов*, называемых *алфавитом*. Примером алфавита являются конечные множества всех заглавных букв, конечное множество всех заглавных и строчных букв и цифр и т.п. В общем виде некоторый алфавит $\sum$ можно представить так:

$$\sum=\{a_0, a_1, a_2, \ldots a_{m-1}\}.$$

Объединяя по определённому правилу буквы из алфавита $\sum$, можно создать новые алфавиты:

- алфавит $\sum^2$, содержащий m^2 биграмм a_0a_0, a_1a_1, a_2a_2, … $a_{m-1}a_{m-1}$;
- алфавит $\sum^3$, содержащий m^3 триграмм $a_0a_0a_0$, $a_1a_1a_1$, $a_2a_2a_2$, … $a_{m-1}a_{m-1}a_{m-1}$.

В общем случае, объединяя по n букв, получаем алфавит $\sum^n$, содержащий m^n n-грамм.

Например, алфавит английских букв $\sum$={ABCDEFGH…WXYZ} объёмом m=26 букв позволяет сгенерировать посредством операции конкатенации алфавит из 26^2=676 биграмм: AA, AB, …, XZ, ZZ, алфавит из 26^3=17576 триграмм: AAA, AAB, …, ZZX, ZZZ и т.д.

При выполнении криптографических преобразований полезно заменить буквы алфавита целыми числами 0, 1, 2, 3, … , что позволяет упростить выполнение необходимых алгебраических преобразований. Например, можно установить взаимно однозначное соответствие между русским алфавитом

$$\sum_{рус}=\{АБВГДЕ…ЮЯ\}$$

и множеством целых чисел $\bar{Z}_{32}$={0, 1, 2, 3, … 31};
между английским алфавитом

$$\sum_{англ}=\{\ ABCDEF…YZ\}$$

и множеством целых чисел $\bar{Z}_{26}$={0, 1, 2, 3, … 25}.

Далее в наших примерах будем использовать алфавит $\bar{Z}_m$={0, 1, 2, 3, … m-1}, содержащий m «букв» (в виде чисел).

Замена букв традиционного алфавита числами позволяет более чётко сформулировать основные концепции и приёмы криптографических преобразований. Одновременно, для некоторых иллюстраций, будет использоваться алфавит естественного языка.

Соответствие межу русским алфавитом и множеством целых чисел $\bar{Z}32=\{0, 1, 2, 3, \dots 31\}$ иллюстрирует следующая таблица.

Таблица 4

Буква	Число	Буква	Число	Буква	Число	Буква	Число
А	0	И	8	Р	16	Ш	24
Б	1	Й	9	С	17	Щ	25
В	2	К	10	Т	18	Ь	26
Г	3	Л	11	У	19	Ы	27
Д	4	М	12	Ф	20	Ъ	28
Е	5	Н	13	Х	21	Э	29
Ж	6	О	14	Ц	22	Ю	30
З	7	П	15	Ч	23	Я	31

Соответствие межу английским алфавитом и множеством целых $\bar{Z}26=\{0, 1, 2, 3, \dots 25\}$ иллюстрирует другая таблица.

Таблица 5

Буква	Число	Буква	Число	Буква	Число
A	0	J	9	S	18
B	1	K	10	T	19
C	2	L	11	U	20
D	3	M	12	V	21
E	4	N	13	W	22
F	5	O	14	X	23
G	6	P	15	Y	24
H	7	Q	16	Z	25
I	8	R	17		

Текст с n буквами из алфавита $\bar{Z}_m$ можно рассматривать как n-грамму $\bar{X}=(x_0, x_1, x_2, \dots x_{n-1})$, где $x_i \in \bar{Z}_m$, $0 \le i < n$, для некоторого целого n=1, 2, 3, … .

Через $\bar{Z}_{m,n}$ будем обозначать множество n-грамм, образованных из букв множества $\bar{Z}_m$.

Криптографическое преобразование E представляет собой совокупность преобразований $E=\{E^{(n)}: 1<=n<\infty\}$,

$E^{(n)}: \bar{Z}_{m,n} \rightarrow \bar{Z}_{m,n}$.

Преобразование $E^{(n)}$ определяет, как каждая n-грамма открытого текста $\bar{X} \in \bar{Z}_{m,n}$ заменяется n-граммой шифротекста $\bar{Y}$, т.е.

$\bar{Y}= E^{(n)} (\bar{X})$, причём $\bar{X}$, $\bar{Y} \in \bar{Z}_{m,n}$ при этом обязательным является требование взаимной однозначности преобразования $E^{(n)}$ на множестве $\bar{Z}_{m,n}$.

Криптографическая система может трактоваться как семейство криптографических преобразований $\bar{E} = \{E_K : K \in \bar{K}\}$, помеченных параметром K, называемым ключом.

Множество значений ключа образует ключевое пространство $\overline{K}$.

Рассмотрим традиционные (классические) методы шифрования, отличающиеся симметричной функцией шифрования. К ним относятся:

- шифры перестановки,
- шифры простой и сложной замены,

а также некоторые их модификации и комбинации.

ШИФРЫ ПЕРЕСТАНОВКИ

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

Шифр перестановки «скитала»

Ещё в Vв. до нашей эры спартанцы шифровали свои донесения с помощью *скитала*, как криптографического устройства для шифрования методом перестановки.

Метод шифрования состоял в том, что брали цилиндрический стержень (скитала), наматывали на него виток к витку полоску пергамента и писали на ней вдоль стержня несколько строк текста сообщения. Сняв полоску со стержня, получали текст из хаотично расположенных букв. Это могло выглядеть следующим образом:

	Н У Н А А	А П И В	С Л Е Т	Т Е З Р	

Рис. 1. Вид «скитала»

Такой же результат можно получить, если выбирать в шифр буквы текста по кольцу через определённое число позиций, пока текст не будет исчерпан.

Для дешифрования такого текста нужно не только знать правило шифрования, но и обладать стержнем определённого диаметра, если применяли скитала. Рассмотренный шифр многократно совершенствовался.

Шифрующие таблицы

С начала XIVв. возрождается криптография, которая применяется не только в политике, дипломатии и военном деле, но и для защиты интеллектуальной собственности от преследования инквизиции или от плагиата. В то время разрабатываются шифры перестановки с применением шифрующих таблиц, которые задают правила перестановки в сообщении.

В качестве ключа в шифрующих таблицах используются:

- размеры таблицы;
- слово или фраза, задающие перестановку;

- особенности структуры таблицы.

Самый примитивный табличный шифр перестановки - *простая перестановка,* ключ которой – размер таблицы. Этот метод подобен шифру скитала. Например, сообщение

«СЛОВО ЗАДАЮЩЕЕ ПЕРЕСТАНОВКУ ЭТО ПЕЛИКАН»

записывается в таблицу 5x7 (т.к. в сообщении 35 букв) по столбцам:

С	З	Щ	Р	Н	Э	Л
Л	А	Е	Е	О	Т	И
О	Д	Е	С	В	О	К
В	А	П	Т	К	П	А
О	Ю	Е	А	У	Е	Н

Рис. 2. Матрица перестановок

Для формирования шифртекста считать содержимое таблицы по строкам. Если для удобства чтения шифрованный текст записывать группами по несколько букв (к примеру, по пять), то получится такое шифрованное сообщение:

СЗЩРН ЭЛЛАЕ ЕОТИО ДЕСВО КВАПТ КПАОЮ ЕАУЕН

Отправитель и получатель сообщения должны заранее условиться об общем ключе в виде *размера таблицы и направления шифрования*-дешифрования (строки и столбцы или столбцы и строки). При дешифровании алгоритм шифрования работает в обратном порядке.

Несколько большей стойкостью к раскрытию обладает метод шифрования, называемый *одиночной перестановкой по ключу*. Рассматриваемый метод отличается от предыдущего тем, что столбцы таблицы переставляются по ключевому слову, фразе или набору чисел длиной в одну строку таблицы. Как было сказано в предыдущем сообщении, в качестве ключа применим слово ПЕЛИКАН размером в 7 букв, что равно количеству столбцов в предыдущем примере. Текст сообщения возьмём из предыдущего примера. Добавим к таблице 5x7 ещё две строки выше и запишем исходное сообщение так же по столбцам буквы на прежние места, (можно и по строкам). В верхней строке поместим ключ.

П	**Е**	**Л**	**И**	**К**	**А**	**Н**
7	***2***	***5***	***3***	***4***	***1***	***6***
С	З	Щ	Р	Н	Э	Л
Л	А	Е	Е	О	Т	И
О	Д	Е	С	В	О	К
В	А	П	Т	К	П	А
О	Ю	Е	А	У	Е	Н

А	Е	И	К	Л	Н	П
1	***2***	***3***	***4***	***5***	***6***	***7***
Э	З	Р	Н	Щ	Л	С
Т	А	Е	О	Е	И	Л
О	Д	С	В	Е	К	О
П	А	Т	К	П	А	В
Е	Ю	А	У	Е	Н	О

Рис. 3. Шифрующие таблицы

Определим номера для букв ключа в естественном порядке их следования в алфавите. Если в ключе встречаются одинаковые буквы (масса, рассказ и т.п.) они нумеруются по порядку слева направо (а-1, а-2, м-3, с-4, с-5 для слова «масса» или а-1, а-2, -з-3, к-4, р-5, с-6, с-7 для ключевого слова «рассказ»). В правой таблице столбцы переставлены согласно номерам букв ключа.

При считывании содержимого правой таблицы по строкам и записи шифртекста группами по пять букв получим шифрованное сообщение:

ЭЗРНЩ ЛСТАЕ ОЕИЛО ДСВЕК ОПАТК ПАВЕЮ АУЕНО

Для обеспечения дополнительной стойкости шифра можно повторно зашифровать сообщение, которое уже прошло шифрование. Такой *метод шифрования* называется *двойной перестановкой*. В случае двойной перестановки столбцов и строк таблицы перестановки определяются отдельно для столбцов и отдельно для строк. С начала в таблицу записывается текст сообщения (можно по строкам или столбцам), а затем поочерёдно переставляются столбцы, а затем строки или строки, а затем столбцы. При дешифровании порядок перестановок должен быть обратным. Рассмотрим пример шифрования двойной перестановкой. Есть текст: «НАСТУПАЕМ ВТОРОГО».

Запишем его в исходную таблицу а) по строкам. Ключом к шифру двойной перестановки служит последовательность номеров столбцов и номеров строк исходной таблицы (4132 и 3142).

	4	1	3	2
3	Н	А	С	Т
1	У	П	А	Е
4	М	В	Т	О
2	Р	О	Г	О

	1	2	3	4
3	А	Т	С	Н
1	П	Е	А	У
4	В	О	Т	М
2	О	О	Г	Р

	1	2	3	4
1	П	Е	А	У
2	О	О	Г	Р
3	А	Т	С	Н
4	В	О	Т	М

Рис. 4.

а) исходная таблица; б) перестановка столбцов; в) перестановка строк.

Если считать шифртекст из таблицы в) построчно блоками по четыре буквы, то получится следующее: ПЕАУ ООГР АТСН ВОТМ

Число вариантов двойной перестановки резко возрастает при увеличении размера таблицы:

- для таблицы 3х3 36 вариантов;
- для таблицы 4х4 576 вариантов;
- для таблицы 5х5 14 400 вариантов.

Двойная перестановка также не отличается высокой стойкостью и сравнительно просто «взламывается» при любом размере таблицы шифрования.

Применение для шифрования магических квадратов

В средние века для шифрования перестановкой применялись и магические квадраты.

Магическим квадратом называют квадратные таблицы с вписанными в их клетки последовательными натуральными числами, начиная от 1, которые дают в сумме по каждому столбцу, каждой строке и каждой диагонали одно и то же число.

Шифруемый текст записывали в магические квадраты в соответствии с нумерацией клеток. Рассмотрим пример магического квадрата и его заполнение сообщением «НАСТУПАЕМ ВТОРОГО».

О 16	С 3	А 2	Р 13
У 5	В 10	Т 11	Е 8
М 9	П 6	А 7	О 12
Т 4	Г 15	О 14	Н 1

Рис. 5. Вид магического квадрата шифрования

Если затем выписать содержимое такой таблицы по строкам, то получится шифртекст, сформированный благодаря перестановке букв исходного сообщения:

ОСАР УВТЕ МПАО ТГОН

Число магических квадратов быстро возрастает при увеличении размера квадрата. Существует только один магический квадрат размером 3x3 (если не учитывать его повороты). Количество магических квадратов 4x4 составляет уже 880, а количество магических квадратов 5x5 – около 250 000.

Магические квадраты средних и больших размеров могли служить хорошей базой для нужд шифрования того времени, т.к. ручной перебор всех вариантов такого шифра практически был невозможен.

ШИФРЫ ПРОСТОЙ ЗАМЕНЫ

При шифровании заменой (подстановкой) символы шифруемого текста заменяются символами того же или другого алфавита по заранее установленным правилам замены. В шифре простой замены каждый символ исходного текста заменяется символом того же алфавита одинаково на всём протяжении текста. Часто шифры простой замены называют шифрами одноалфавитной подстановки.

Полибианский квадрат

Считается, что одним из первых шифров простой замены был так называемый полибианский квадрат. За два века до нашей эры греческий писатель и историк Полибий изобрёл для целей шифрования квадратную таблицу размером 5x5, заполненную 24 буквами греческого алфавита и пробелом в случайном порядке:

Таблица 6

λ	ε	υ	ω	γ
ρ	ζ	δ	σ	ο
μ	η	β	ξ	τ
ψ	π	θ	α	x
χ	ν		φ	ι

При шифровании текста находили в квадрате соответствующую букву и записывали в шифртекст букву, расположенную ниже её в том же столбце. Буквы нижней строки шифровали буквами верхней строки того же столбца. Например, для слова **ταυροσ** получим шифртекст **χφδμτξ**.

Концепция полибианского квадрата оказалась плодотворной и нашла применение в криптосистемах последующего времени.

Квадрат, подобный полибианскому, можно подготовить и из букв русского, исключив некоторые буквы, или английского алфавитов.

Система шифрования Цезаря

Шифр Цезаря является частным случаем шифра простой замены (одноалфавитной подстановки). Название шифр получил по имени императора Гай Юлия Цезаря, который использовал этот шифр при переписке с Цицероном (около 50г. до н.э.).

При шифровании каждая буква исходного текста заменялась на другую букву того же алфавита по правилу. Заменяющая буква определялась путём смещения по алфавиту от исходной буквы на К букв. При достижении конца алфавита выполнялся циклический переход к его началу. Цезарь использовал шифр замены со смещением К=3. Такой шифр замены можно задать таблицей подстановок, содержащей соответствующие пары букв открытого текста и шифртекста. Например, послание Цезаря VENI VIDI VICI, что в переводе на русский означает «Пришёл, Увидел, Победил», направленное его другу Аминтию после победы над понтийским царём Фарнаком в зашифрованном виде с использованием таблицы подстановок:

Таблица 7

Шифр замена

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

выглядело бы как: YHQL YLGL YLFL

В отличие от шифра Цезаря, *система шифрования* Цезаря образует по существу семейство одноалфавитных подстановок для выбираемых значений ключа К, причём $0 \le K < m$, где m - количество букв алфавита.

Достоинством системы шифрования Цезаря является простота шифрования и дешифрования. К недостаткам системы шифрования Цезаря следует отнести следующие:

- подстановки, выполняемые в соответствии с системой Цезаря, не маскируют частот появления различных букв исходного открытого текста;
- сохраняется алфавитный порядок в последовательности заменяющих букв: при изменении значения К изменяются только начальные позиции такой последовательности;
- число возможных ключей К мало;
- шифр Цезаря легко вскрывается на основе анализа частот появления букв в шифртексте.

Криптоаналитическая атака против системы одноалфавитной замены начинается с подсчёта частот появления символов: определяется число появлений каждой буквы в шифртексте. Затем полученное распределение частот букв в шифртексте сравнивается с распределением частот букв в алфавите исходных сообщений, например в английском. Буква, с наивысшей частотой появления в шифртексте, заменяется буквой с наивысшей частотой появления в английском языке и т.д. Вероятность успешного вскрытия системы шифрования повышается с увеличением длины шифртекста.

Концепция, заложенная в систему шифрования Цезаря, оказалась весьма плодотворной, о чём свидетельствуют её многочисленные модификации:

- аффинная система подстановок Цезаря;
- система Цезаря с ключевым словом;
- шифрующие таблицы Трисемуса;
- биграммный шифр Плейфейра;
- криптосистема Хилла;
- система омофонов.

Аффинная система подстановок Цезаря

В системе шифрования Цезаря использовались только аддитивные свойства множества целых $\bar{Z}_m$. Однако символы множества $\bar{Z}_m$ можно умножить по модулю m. Применяя одновременно операции сложения и умножения по модулю m над элементами множества $\bar{Z}_m$, можно получить систему подстановок, которую называют аффинной системой подстановок Цезаря.

Определим преобразование в такой системе:

$$E_{a,b} : \bar{Z}_m \to \bar{Z}_m,$$

$$E_{a,b} : t \to E_{a,b}(t),$$

$$E_{a,b}(t) = at + b (mod\ m),$$

где a,b - целые числа, $0 \le a,b < m$, НОД (a, m) =1.

В данном преобразовании буква, соответствующая числу t, заменяется буквой, соответствующей числовому значению $(at + b)$ по модулю m.

Заметим, что преобразование $E_{a,b}(t)$ является взаимно однозначным отображением на множестве $\bar{Z}_{\text{m}}$ только в том случае, если наибольший общий делитель чисел a и m, обозначаемый как НОД(a, m), равен единице, т.е. a и m должны быть взаимно простыми числами.

Например: пусть m=26, a=3, b=5. Тогда, очевидно, НОД(3, 26)=1, и получаем следующее соответствие между числовыми кодами букв:

Таблица 8

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
3t+5	5	8	11	14	17	20	23	0	3	6	9	12	15	18	21	24	1	4	7	10	13	16	19	22	25	2

Преобразуя числа в буквы английского языка, получаем следующее соответствие для букв открытого текста и шифртекста:

Таблица 9

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

Так исходное сообщение HOPE преобразуется в шифртекст AVYR

Достоинством аффинной системы является удобное управление ключами – ключи шифрования и дешифрования представляются в компактной форме в виде пары чисел (a и b). Недостатки аффинной системы аналогичны недостаткам системы шифрования Цезаря. Аффинная система использовалась на практике несколько веков назад, а сегодня её применение ограничивается большей частью иллюстрациями основных криптологических положений.

Система Цезаря с ключевым словом

Данная система является одноалфавитной системой подстановок. Особенностью этой системы является использование ключевого слова для смещения и изменения порядка символов в алфавите подстановки.

Выберем некоторое число k, $0 \leq k < 25$, и слово или короткую фразу в качестве ключевого слова. Желательно, чтобы все буквы ключевого слова были различными. Пусть выбрано слово DIPLOMAT в качестве ключевого слова и число k =5.

Ключевое слово записывается под буквами алфавита, начиная с буквы, числовой код которой совпадает с выбранным числом k:

0 1 2 3 4 **5** 10 15 20 25
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

D I P L O M A T

Оставшиеся буквы алфавита подстановки записываются после ключевого слова в алфавитном порядке:

5

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

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

Во второй строке получили буквы, которыми заменяются буквы исходного текста при шифровании. Пусть исходное сообщение: VENI V I DI VI C I

шифруется так: NZBL NLYL NLXL

Требование о различии всех букв ключевого слова необязательно. Можно просто записать ключевое слово (или фразу), исключив одинаковые буквы. Например, ключевая фраза: КАК ДЫМ ОТЕЧЕСТВА НАМ СЛАДОК И ПРИЯТЕН и число k=3 порождают следующую таблицу подстановок:

0 3

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

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

Большим достоинством системы Цезаря с ключевым словом является то, что количество возможных ключевых слов практически неисчерпаемо. Недостатком этой системы является возможность взлома шифртекста на основе анализа частот появления букв.

Шифрующие таблицы Трисемуса

В 1508г. аббат из Германии Иоганн Трисемус написал печатную работу по криптологии под названием «Полиграфия». В этой книге он впервые систематически описал применение шифрующих таблиц, заполненных алфавитом в случайном порядке.

Для получения такого шифра замены обычно использовались таблица для записи букв алфавита и ключевое слово (или фраза).

В таблицу сначала вписывалось по строкам ключевое слово, причём повторяющиеся буквы исключались. Затем эта таблица дополнялась не вошедшими в неё буквами алфавита по порядку. Т.к. ключевое слово или фразу легко хранить в памяти, то такой подход упрощал процесс шифрования и дешифрования. Например, для русского алфавита можно взять таблицу размера 4x8. В качестве ключа возьмём слово БАНДЕРОЛЬ и получим шифрующую таблицу:

Таблица 10

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

Подобно полибианскому квадрату, при шифровании находят в этой таблице очередную букву открытого текста и записывают в шифртекст букву,

расположенную ниже её в том же столбце. Буквы нижней строки таблицы шифруем буквами верхней строки из того же столбца.

Например, при шифровании с помощью этой таблицы сообщения:

ПРИЛЕТАЕМПЯТОГО

получим шифртекст ШИФКЗЫВЗЧШЛЫЙСЙ

Такие табличные шифры называются *монограммными*, так как шифрование выполняется по одной букве. Трисемус первым заметил, что шифрующие таблицы позволяют шифровать сразу по две буквы. Такие шифры называются *биграммными*.

Биграммный шифр Плейфейра

Шифр Плейфейра, изобретённый в 1854г, это наиболее известный биграммный шифр замены. Он применялся Великобританией во время Первой мировой войны. Основой шифра Плейфейра является шифрующая таблица со случайно расположенными буквами алфавита исходного текста.

Для удобства запоминания шифрующей таблицы отправителем и получателем сообщений можно использовать ключевое слово (или фразу) при заполнении начальных строк таблицы. В целом структура шифрующей таблицы системы Плейфейра полностью аналогична структуре таблицы Трисемуса. Воспользуемся ею.

Таблица 11

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

Процедура шифрования включает следующие шаги:

1. Открытый текст исходного сообщения разбивается на пары букв (биграммы). Текст должен иметь чётное количество букв, и в нём не должно быть биграмм, содержащих две одинаковые буквы. Если эти требования не выполнены, то текст модифицируется даже из-за незначительных орфографических ошибок.
2. Последовательность биграмм открытого текста преобразуется с помощью шифрующей таблицы в последовательность биграмм шифртекста по следующим правилам:
   а) Если обе буквы биграммы открытого текста не попадают на одну строку или столбец (например, буквы А и Й), тогда находят буквы в углах прямоугольника, определяемого данной парой букв. (В нашем примере это буквы АЙОВ). Пара букв АЙ отображается в пару ОВ. Последовательность букв в биграмме шифртекста должна быть зеркально расположенной по отношению к последовательности букв в биграмме открытого текста.

b) Если буквы биграммы открытого текста принадлежат одному столбцу таблицы, то буквами шифртекста считаются буквы, которые расположены под ними. (Например, биграмма НС даёт биграмму шифртекста ГЩ). Если при этом буква открытого текста находится в нижней строке, то для шифртекста берётся соответствующая буква из верхней строки того же столбца. (Например, биграмма ВШ даёт биграмму шифртекста ПА).

c) Если обе буквы открытого текста принадлежат одной строке таблицы, то буквами шифртекста считаются буквы, которые лежат справа от них. (Например, биграмма НО даёт биграмму шифртекста ДЛ). Если при этом буква открытого текста находится в правом крайнем столбце, то для шифра берут соответствующую букву из левого столбца в той же строке. (Например, биграмма ФЦ даёт биграмму шифртекста ХМ).

Зашифруем текст: ВСЕ ТАЙНОЕ СТАНЕТ ЯВНЫМ. Разобьём текст на биграммы: ВС ЕТ АЙ НО ЕС ТА НЕ ТЯ ВН ЫМ.

Зашифруем полученные биграммы с помощью таблицы шифрования и вышеописанных правил. Получим последовательность биграмм шифртекста: ГП ДУ ОВ ДЛ НУ ПД ДР ЦЫ ГА ЧТ. При дешифровании применяется обратный порядок действий.

Шифрование биграммами резко повышает стойкость шифров к вскрытию.

Система омофонов

Система омофонов обеспечивает простейшую защиту от криптоаналитических атак, основанных на подсчёте частот появления букв в шифртексте. Система омофонов является одноалфавитной, хотя при этом буквы исходного сообщения имеют несколько замен. Число замен берётся пропорциональным вероятности появления букв в открытом тексте.

Существуют таблицы распределения вероятностей букв в русских и английских текстах.

Таблица 12а

Распределения вероятностей букв в русских и английских текстах.

Буква	Вероят-ность	Буква	Вероят-ность	Буква	Вероят-ность	Буква	Вероят-ность
Пробел	0,175	Р	0,040	Я	0,018	Х	0,009
О	0,090	В	0,038	Ы	0,016	Ж	0,007
Е	0,072	Л	0,035	З	0,016	Ю	0,006
А	0,062	К	0,028	Ъ	0,014	Ш	0,006
И	0,062	М	0,028	Б	0,014	Ц	0,004
Н	0,053	Д	0,025	Г	0,013	Щ	0,003
Т	0,053	П	0,023	Ч	0,012	Э	0,003
С	0,045	У	0,021	Й	0,010	Ф	0,002

Таблица 12б

Буква	Вероят-ность	Буква	Вероят-ность	Буква	Вероят-ность
E	0,123	L	0,040	B	0,016
T	0,096	D	0,036	G	0,016
A	0,081	C	0,032	V	0,009
O	0,079	U	0,031	K	0,005
N	0,072	P	0,023	Q	0,002
I	0,071	F	0,023	X	0,002
S	0,066	M	0,022	J	0,001
R	0,060	W	0,020	Z	0,001
H	0,051	Y	0,019		

Буквы в таблицах указаны в порядке убывания вероятности их появления в тексте. Например, русская буква Е встречается в 36 раз чаще, чем буква Ф, а английская буква Е встречается в 123 раза чаще, чем буква Z.

Шифруя букву исходного сообщения, выбирают случайным образом одну из замен. Замены (часто называемые омофонами) могут быть представлены трёхзначными числами от 000 до 999. Например, в английском алфавите букве Е присваиваются 123 случайных номера, буквам B и G – по 16 номеров, а буквам J и Z – по одному номеру. Если омофоны (замены) присваиваются случайным образом различным появлениям одной и той же буквы, тогда каждый омофон появляется в шифртексте равновероятно.

При таком подходе к формированию шифртекста простой подсчёт частот уже ничего не даёт криптоаналитику. Однако в принципе полезна также информация о распределении пар и троек букв в различных естественных языках. Если эту информацию использовать при криптоанализе, он будет проведён более успешно.

ШИФРЫ СЛОЖНОЙ ЗАМЕНЫ

Шифры сложной замены называют многоалфавитными, так как для шифрования каждого символа исходного сообщения применяют свой шифр простой замены. Многоалфавитная подстановка последовательно и циклически меняет используемые алфавиты.

При r - алфавитной подстановке символ x_0 исходного сообщения заменяется символом y_0 из алфавита B_0, символ x_1 - символом y_1 из алфавита B_1, и так далее, символ x_{r-1} заменяется символом y_{r-1} из алфавита B_{r-1}, символ x_r заменяется символом y_r из алфавита B_0, и т. д.

Схема r - алфавитной подстановки для случая r=4 выглядит следующим образом:

Таблица 13

Схема r - алфавитной подстановки

Входной символ	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9
Алфавит подстановки	B_0	B_1	B_2	B_3	B_0	B_1	B_2	B_3	B_0	B_1

Эффект использования многоалфавитной подстановки заключается в том, что обеспечивается маскировка естественной статистики исходного языка, так как конкретный символ из исходного алфавита А может быть преобразован в несколько различных символов шифровальных алфавитов B_j. Степень обеспеченности защиты теоретически пропорциональна длине периода r в последовательности используемых алфавитов B_j.

Многоалфавитные шифры замены предложил и ввёл в практику криптографии Леон Батист Альберти, который также был известным архитектором и теоретиком искусства. Его книга «Трактат о шифре», написанная в 1566 г., представляла собой первый в Европе научный труд по криптографии. Кроме шифра многоалфавитной замены Альберти также подробно описал устройства из вращающихся колёс для его реализации. Криптологи всего мира почитают Л. Альберти основоположником криптологии.

Шифр Гронсфельда

Этот шифр сложной замены представляет собой модификацию шифра Цезаря числовым ключом. Для этого под буквами исходного сообщения записывают цифры числового ключа. Если ключ короче сообщения, то его запись циклически повторяют. Шифртекст получают примерно, как в шифре Цезаря, но отсчитывают по алфавиту не третью букву (как это делается в шифре Цезаря), а выбирают ту букву, которая смещена по алфавиту на соответствующую цифру ключа. Например, применяя в качестве ключа группу из четырёх начальных цифр числа *e* (основания натуральных логарифмов), а именно 2718, получим из исходного сообщения ВОСТОЧНЫЙ ЭКСПРЕСС следующий шифртекст:

Сообщение В О С Т О Ч Н Ы Й Э К С П Р Е С С
Ключ 2 7 1 8 2 7 1 8 2 7 1 8 2 7 1 8 2
Шифртекст:Д Х Т Ь Р Ю О Г Л Д Л Щ С Ч Ж Щ У
Рис. 6.Пример шифрования методом Гронсфельда

Чтобы зашифровать первую букву сообщения В, используя первую цифру ключа 2, нужно отсчитать вторую по порядку букву от В в алфавите: 1-Г, 2-Д, получаем первую букву шифр текста Д.

Следует отметить, что шифр Гронсфельда вскрывается относительно легко, если учесть, что в числовом ключе каждая цифра имеет только десять

значений, а значит, имеется лишь десять вариантов прочтения каждой буквы шифртекста. С другой стороны, шифр Гронсфельда допускает дальнейшие модификации, улучшающие его стойкость, в частности двойное шифрование разными числовыми ключами. Шифр Гронсфельда представляет собой по существу частный случай системы шифрования Вижинера.

Система шифрования Вижинера

Эта система впервые была опубликована в 1586г. и является одной из старейших и наиболее известных многоалфавитных систем. Система названа по имени французского дипломата XVI века Блеза Вижинера, который развивал и совершенствовал криптографические системы.

Система Вижинера подобна такой системе шифрования Цезаря, у которой ключ подстановки меняется от буквы к букве. Этот шифр многоалфавитной замены можно описать таблицей шифрования, называемой таблицей (квадратом) Вижинера. Для букв русского алфавита таблица Вижинера выглядит следующим образом (Таблица 3).

Таблица используется для шифрования и дешифрования, имея два входа:

- верхнюю строку подчёркнутых символов используют для считывания очередной буквы исходного открытого текста;
- крайний левый столбец ключа.

Таблица 14

ключ	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я
0	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я
1	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а
2	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б
3	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в
4	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г
5	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д
6	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е
7	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж
8	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з
9	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и
10	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й
11	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к
12	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л
13	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м
14	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н
15	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о
16	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
17	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р
18	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с
19	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т

20	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у
21	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф
22	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х
23	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц
24	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч
25	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш
26	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ
27	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь
28	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы
29	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ
30	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э
31	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю

Последовательность ключей обычно получают из числовых значений букв ключевого слова. При шифровании исходного сообщения его выписывают в строку, а под ним записывают ключевое слово (или фразу). Если ключ короче сообщения, то его циклически повторяют, подобно ключу шифра Гронсфельда. В процессе шифрования находят в верхней строке таблицы очередную букву исходного текста и в левом столбце очередное значение ключа. Буква шифртекста находится на пересечении столбца, определяемого шифруемой буквой, и строки, определяемой числовым значением ключа.

Рассмотрим пример. Необходимо зашифровать сообщение: ПРИЛЕТАЮ СЕДЬМОГО. Выберем ключевым слово АМБРОЗИЯ.

Согласно алгоритму, в 1-ю строку запишем сообщение, во 2-ю – ключевое слово, а в 3-ю будем выписывать буквы шифра применив, таблицу Вижинера.

Сообщение	П	Р	И	Л	Е	Т	А	Ю		С	Е	Д	Ь	М	О	Г	О
Ключ	А	М	Б	Р	О	З	И	Я		А	М	Б	Р	О	З	И	Я
Шифртекст	**П**	**Ъ**	**Й**	**Ы**	**У**	**Щ**	**И**	**Э**		**С**	**С**	**Е**	**К**	**Ь**	**Х**	**Л**	**Н**

Рис. 7.Пример шифрования методом Вижинера

Шифр «двойной квадрат» Уитстона

В 1854г. англичанин Чарльз Уитстон разработал новый метод шифрования биграммами, который называют «двойным квадратом», подобно полибианскому квадрату. Шифр Уитстона открыл новый этап в истории развития криптографии. В отличие от полибианского шифра «двойной квадрат» использует сразу две таблицы, размещённые по одной горизонтали, а шифрование идёт биграммами, как в шифре Плейфейра. Эти не столь важные модификации привели к появлению на свет качественно новой криптографической системы ручного шифрования. Шифр «двойной квадрат» оказался очень надёжным и удобным. Он применялся Германией даже в годы Второй мировой войны.

Рассмотрим процедуру шифрования на примере. Пусть имеются две таблицы со случайно расположенными в них русскими алфавитами.

Таблица 15

«Двойной квадрат» Уитстона

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

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

Перед шифрованием исходное сообщение разбивают на биграммы. Каждая биграмма шифруется отдельно. Первую букву биграммы находят в левой таблице, а вторую букву – в правой таблице. Затем мысленно строят прямоугольник так, чтобы буквы биграммы лежали в его противоположных вершинах. Другие две вершины этого прямоугольника дают буквы биграммы шифртекста.

Предположим, что шифруется биграмма исходного текста ИЛ. Буква И находится в столбце 1 и строке 2 левой таблицы. Буква Л находится в столбце 5 и строке 4 правой таблицы. Это означает, что прямоугольник образован строками 2 и 4, а также столбцами 1 левой таблицы и 5 правой таблицы. Следовательно, в биграмму шифртекста входят буква О, расположенная в столбце 5 и строке 2 правой таблицы, и буква В, расположенная в столбце 1 и строке 4 левой таблицы, т.е. получаем биграмму шифртекста ОВ.

Если обе буквы биграммы лежат в одной строке, то и буквы шифртекста будут из той же строки. Первую букву биграммы берут из левой таблицы в столбце, соответствующем второй букве биграммы сообщения. Вторая же буква биграммы шифртекста берётся из правой таблицы в столбце, соответствующем первой букве биграммы сообщения. Поэтому биграмма сообщения ТО шифруется как ЖБ. Зашифруем текст «ПРИЛЕТАЮ ШЕСТОГО».

Сообщение ПР ИЛ Е Т АЮ - Ш ЕС ТО ГО

Шифртекст ПЕ ОВ ЩН ФМ ЕШ РФ БЖ ДЦ

Рис. 8. Пример шифра Уитстона

Шифрование методом «двойного квадрата» даёт весьма устойчивый к вскрытию и простой в применении шифр. Взламывание шифртекста «двойной квадрат» требует больших усилий, при этом длина сообщения должна быть не менее тридцати строк.

Одноразовая система шифрования

Почти все применяемые на практике шифры характеризуются как условно надёжные, т.к. могут быть в принципе раскрыты при наличии неограниченных вычислительных возможностей. Абсолютно надёжные шифры нельзя разрушить даже при использовании неограниченных вычислительных возможностей. Существует единственный такой шифр, применяемый на практике, - одноразовая система шифрования. Характерной особенностью такой системы является одноразовое использование ключевой последовательности.

Одноразовая система шифрует открытый текст $\bar{X}=(X_0, X_1, \ldots, X_{n-1})$ в шифртекст $\bar{Y}=(Y_0, Y_0, \ldots, Y_{n-1},)$ посредством подстановки Цезаря $Y_i=(X_i+K_i) \bmod m$, $0 \leq i < n$, где K_i –i –й элемент случайной ключевой последовательности.

Ключевое пространство $\bar{K}$ одноразовой системы представляет собой набор дискретных случайных величин из $\bar{Z}_m$ и содержит m^n значений.

Процедура расшифровывания описывается соотношением: $X_i = (Y_i - K_i) \bmod m$, где K_i –i-й элемент той же самой случайной ключевой последовательности.

Одноразовая система изобретена в 1917г. американцами Дж. Моборном и Г. Вернамом. Для реализации этой системы подстановки иногда используют одноразовый блокнот. Этот блокнот состоит из отрывных страниц, на каждой из которых напечатана таблица со случайными числами (ключами) K_i. Блокнот выполняется в двух экземплярах: один используется отправителем, а другой получателем. Для каждого символа X_i сообщения используется свой ключ K_i из таблицы только один раз. После использования таблица должна быть удалена из блокнота и уничтожена. Шифрование нового сообщения начинается с новой страницы.

Этот шифр абсолютно надёжен, если набор ключей K_i действительно случаен и непредсказуем. Если криптоаналитик попытается использовать для заданного шифртекста все возможные наборы ключей и восстановить все возможные варианты исходного текста, то они все окажутся равновероятными. Не существует способа выбрать исходный текст, который был действительно послан. Теоретически доказано, что одноразовые системы являются нераскрываемыми системами, т.к. их шифртекст не содержит достаточной информации для восстановления открытого текста.

Кажется, что благодаря данному достоинству одноразовые системы следует применять во всех случаях для абсолютной информационной безопасности. Но возможности применения этой системы ограничены чисто практическими аспектами. Существенным моментом является требование одноразового использования случайной ключевой последовательности. Ключевая последовательность с длиной, не меньшей длины сообщения, должна передаваться получателю сообщения заранее или отдельно по некоторому секретному каналу. Это требование необременительно для передачи важных одноразовых сообщений. Но такое требование практически неосуществимо для

современных систем обработки информации, где требуется шифровать многие миллионы символов.

В некоторых вариантах одноразового блокнота прибегают к более простому управлению ключевой последовательностью, но это приводит к снижению надёжности шифра. Например, ключ определяется указанием места в книге, известной отправителю и получателю сообщения. Ключевая последовательность начинается с указанного места этой книги и используется таким же образом, как система Вижинера. Иногда такой шифр называют *шифром с бегущим ключом*. Управление ключевой последовательностью в таком варианте шифра намного проще, но эти ключи не будут случайными. Поэтому у криптоаналитиков появляется возможность использовать информацию о частотах букв исходного естественного языка.

Шифрование гомофонической заменой

При гомофонической замене одному символу открытого текста ставится в соответствие несколько символов шифр текста. Этот метод применяется для искажения статистических свойств шифртекста: информацию о частотах букв исходного естественного языка.

Рассмотрим пример. Пусть открытый текст содержит слово «ЗАМЕНА». Подстановка алфавита гомофонической замены может выглядеть следующим образом:

Таблица 16

Пример шифрования гомофонической заменой

Алфавит открытого текста	А	Б		Е	Ж	З		М	Н	
Алфавит шифртекста	17	23		97	47	76		32	55	
	31	44	…	51	67	19	…	28	84	…
	48	63		15	33	59		61	34	

Числа замены являются случайными. Заменим каждую букву открытого текста соответствующим (расположенным в том же столбце) случайным числом. Если буква встречается ещё раз или два, то её заменяют числом из второй или третьей строки соответствующего столбца.

Согласно этому правилу получим шифртекст: 76 17 32 97 55 31

Обратное преобразование шифртекста в открытый текст выполняется по обратному алгоритму: выбираем очередное число и находим его в гомофоническом алфавите, а в верхней строки найденного столбца считываем букву. Отправителю и получателю должны быть известны два алфавита.

Таким образом, при гомофонической замене каждая буква открытого текста заменяется по очереди числами соответствующего столбца, и проследить правильную частотность становится затруднительно.

Шифрование методом Вернама

Система шифрования Вернама, в сущности, является частным случаем системы шифрования Вижинера при значении модуля m=2. Конкретная версия

этого шифра, предложенная в 1926г. Гилбертом Вернамом, сотрудником фирмы AT&T США, использует двоичное представление символов исходного текста.

Каждый символ исходного открытого текста из английского алфавита {A,B,C,D,…,Z}, расширенного шестью вспомогательными символами (пробел, возврат каретки и т.п.), сначала кодировался в 5-тибитовый блок (b_0, b_1,…, b_4) телеграфного кода Бодо.

Случайная последовательность ключей k_0, k_1, k_2,… заранее записывалась на бумажной ленте.

Схема передачи сообщений с использованием шифрования методом Вернама выглядела следующим образом:

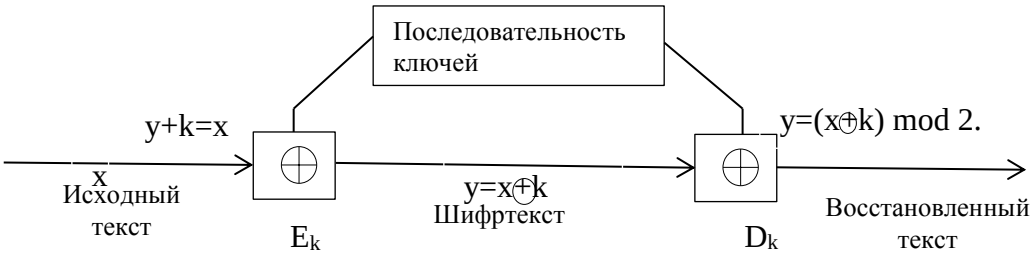

Рис.9. Схема передачи шифрованного сообщения методом Вернама

Шифрование исходного текста, предварительно преобразованного в последовательность двоичных символов *x*, осуществлялось путём сложения по модулю 2 символов *x* с последовательностью ключей *k*. Символы шифртекста:

Таблица 17

Соответствие символов и двоичных кодов

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	;
00000	00001	00010	00011	01000	00101	00110	00111	01000	01001	01010	01011	01100	01101	01110	01111	0000	10001	10010	10011	10100	10101	10110	10111	11000	11001	11010	11011	11100	11101	11110	11111

Например, так преобразуются в последовательность двоичных символов буквы латинского алфавита и ещё шесть добавочных символов. Возможно такое преобразование букв русского алфавита. Тогда буква *C* открытого текста с ключом *k*=*G* преобразуется в букву *E* шифра, так как, выполнив сложение по модулю 2, получим:

Первое слагаемое (исходный символ)	*C*	0	0	0	1	0
Второе слагаемое (символ ключа)	*G*	0	0	1	1	1
Результат (символ шифра)	***E***	**0**	**0**	**1**	**0**	**1**

Расшифрование состоит в сложении по модулю 2 символов *y* шифртекста с той же последовательностью ключей *k*: y+k=x+k+k=x.

При этом последовательности ключей, использованные при шифровании и расшифровании, компенсируют друг друга (при сложении по модулю 2), и в результате восстанавливаются символы *x* исходного текста:

Первое слагаемое (символ шифра)	*E*	0	0	1	0	1
Второе слагаемое (символ ключа)	*G*	0	0	1	1	1
Результат (исходный символ)	***C***	**0**	**0**	**0**	**1**	**0**

При разработке своей системы Вернам проверял её с помощью закольцованных лент, установленных на передатчике и приёмнике для того, чтобы использовалась одна и та же последовательность ключей.

Метод Вернама не зависит от длины последовательности ключей, и, кроме того, он позволяет использовать случайную последовательность ключей. Но при реализации этого метода возникают серьёзные проблемы, связанные с необходимостью доставки получателю такой же последовательности ключей, как у отправителя, либо с необходимостью безопасного хранения идентичных последовательностей ключей у отправителя и получателя. Эти недостатки системы шифрования Вернама преодолены при шифровании методом гаммирования.

Шифрование методом гаммирования

Принцип шифрования заключается в генерации гаммы шифра с помощью генератора псевдослучайных чисел (ПСЧ) и наложении полученной гаммы на открытый текст обратимым образом (например, при использовании логической операции «Исключающее ИЛИ» или сложения по модулю 2).

Перед шифрованием открытые данные делят на блоки одинаковой длины, обычно по 64 бита. Гамма шифра вырабатывается в виде последовательности блоков аналогичной длины.

Расшифрование данных сводится к повторной генерации гаммы шифра при известном ключе и наложению этой гаммы на зашифрованные данные.

Получаемый этим методом шифртекст достаточно труден для раскрытия, т.к. ключ является переменным. По сути дела гамма шифра должна изменяться случайным образом для каждого шифруемого блока. Если период гаммы превышает длину всего шифруемого текста и злоумышленнику неизвестна никакая часть исходного текста, то такой шифр можно раскрыть только прямым перебором всех вариантов ключа. В этом случае криптостойкость шифра определяется длиной ключа.

Чтобы получить линейные последовательности элементов гаммы, длина которых превышает размер шифруемых сообщений, можно использовать, например, линейный генератор ПСЧ, который вырабатывает последовательности псевдослучайных чисел $T(i)$, $T(i+1)=[AT(i)+C] \bmod M$, где A и C – константы, T(i) – исходная величина, выбранная в качестве порождающего числа [T(0)=T(i)].

Такой датчик ПСЧ генерирует псевдослучайные числа с определённым периодом повторения, зависящим от выбранных значений A и C. Значение M обычно устанавливается равным 2^b, где b – длина последовательности (слова ЭВМ) в битах (2^{24}).

Различают методы конечной гаммы и бесконечной гаммы. В качестве конечной гаммы может использоваться фраза, в качестве бесконечной – последовательность, вырабатываемая датчиком псевдослучайных чисел.

Например, открытый текст: «ПРИКАЗ». Согласно соответствию между русским алфавитом и множеством целых $\bar{Z}_{32}=\{0, 1, 2, 3, \ldots 31\}$, текст примет вид: 15 16 08 10 00 07. Гамма пусть будет конечной: «ГАММА»: 03 00 12 12 00. Выполним операцию сложения по модулю 33. Получим последовательность: $y_1=(15+3) \bmod 33=18$ $y_2=(16+0) \bmod 33=16$ $y_3=(8+12) \bmod 33=20$ $y_4=(10+12) \bmod 33=22$ $y_5=(0+0) \bmod 33=0$ $y_6=(7+3) \bmod 33=10$

Шифртекст: РФЦАК (18 16 20 22 00 10).

Стандарт шифрования данных DES

Стандарты по защите данных ЭВМ от несанкционированного доступа требовались в таких областях, как шифрование, установление подлинности личности и данных (аутентификация), контроль доступа, надёжное хранение и передача данных. В результате сотрудничества трёх организаций США – Национального бюро стандартов (NBC), Управления национальной безопасности (NSA), и фирмы IBM - подобный стандарт, получивший название DES (Data Encryption Standart), был разработан и опубликован в 1977г. и остаётся пока распространённым блочным алгоритмом, используемым в системах защиты коммерческой информации.

Алгоритм DES базируется на научной работе Шеннона 1949г., связавшей криптографию с теорией информации. Шеннон выделил два общих принципа, используемых в практических шифрах: рассеивание и перемещение. Рассеиванием он, назвал распространение влияния одного знака открытого текста на множество знаков шифртекста, что позволяет скрыть статистические свойства открытого текста. Под перемещением Шеннон понимал использование взаимосвязи статистических свойств открытого и шифрованного текстов. Но шифр должен не только затруднять раскрытие, а ещё обеспечивать лёгкость шифрования и дешифрования при известном секретном ключе. Поэтому была принята идея использовать произведение простых шифров, каждый из которых вносит небольшой вклад в значительное суммарное рассеивание и перемещение.

В составных шифрах в качестве элементарных составляющих чаще всего используются простые замены (подстановки) и перестановки. Все перестановки и коды в таблицах подобраны разработчиками таким образом, чтобы максимально затруднить процесс дешифрования текста путём подбора ключа.

Алгоритм DES строится в соответствии с методологией сетей Фейстеля и состоит из чередующейся последовательности перестановок и замен. Алгоритм DES осуществляет шифрование 64-битовых блоков данных с помощью 64-битового ключа, в котором значащими являются 56 бит (остальные 8 – проверочные биты для контроля на чётность).

Процесс шифрования состоит из: начальной перестановки битов 64-битового блока, 16 циклов (раундов) шифрования; конечной перестановки битов.

Этот процесс можно представить в виде схемы:

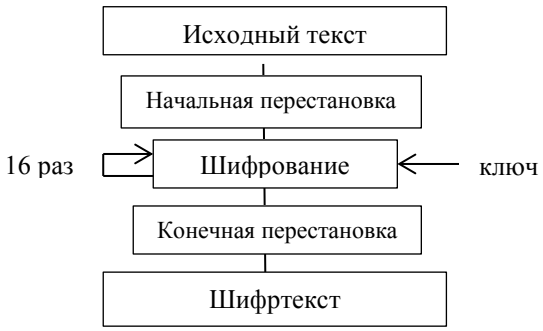

Рис. 10. Схема шифрования DES

Процесс дешифрования результата работы алгоритма DES является инверсным к процессу шифрования. Все операции должны быть выполнены в обратном шифрованию порядке. Это означает, дешифруемые данные сначала переставляются в соответствии с матрицей обратной перестановки, а затем над последовательностью битов выполняются те же действия, что и в процессе шифрования, но в обратном порядке, и затем выполняется перестановка, обратная начальной.

Основные достоинства алгоритма DES:

- используется только один ключ длиной 56 бит;
- зашифровав сообщение с помощью одного пакета программ, для дешифрования можно использовать любой другой пакет программ, соответствующий стандарту DES;
- относительная простота алгоритма обеспечивает высокую скорость обработки;
- криптостойкость алгоритма вполне достаточна для обеспечения информационной безопасности большинства коммерческих приложений.

ГОСТ 28147-89 –отечественный стандарт на шифрование данных

В нашей стране установлен единый алгоритм криптографического преобразования данных для систем обработки информации в сетях, отдельных вычислительных комплексах и ЭВМ, который определяется ГОСТ 28147-89.

Данный стандарт предназначен для аппаратной и программной реализации, удовлетворяет криптографическим требованиям и не налагает ограничений на степень секретности защищаемой информации. Алгоритм шифрования данных, определяемый ГОСТ 28147-89, представляет собой 64 - битовый блочный алгоритм с 256 – битовым ключом.

Данные, подлежащие шифрованию, разделяют на 64 – разрядные блоки, которые, в свою очередь, разбиваются на два субблока N_1 и N_2 по 32 бит. Субблок N_1 обрабатывается определённым образом, после чего его значение складывается со значением субблока N_2 (сложение выполняется по модулю 2,

т.е. применяется логическая операция XOR – «исключающее или»), а затем субблоки меняются местами. Данное преобразование выполняется циклически определённое число раз («раундов») – 16 или 32, в зависимости от режима работы алгоритма.

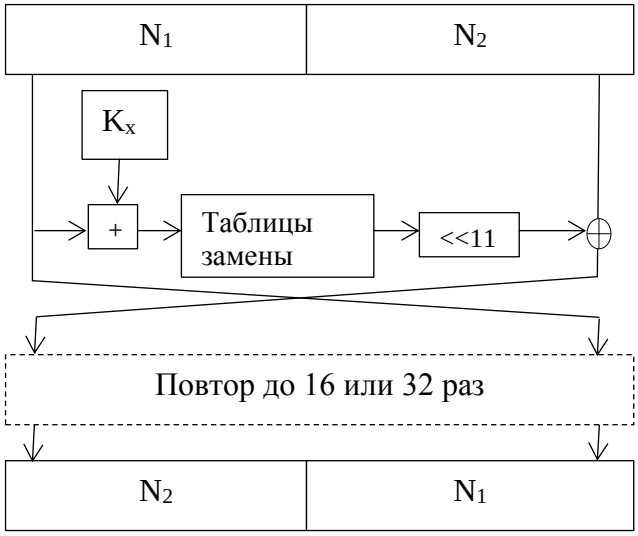

Рис. 11. Схема алгоритма ГОСТ 28147- 89

В каждом раунде выполняются две операции.

Первая операция – наложение ключа. Содержимое субблока N_1 складывается по модулю 2^{32} с 32 – битовой частью ключа K_x. Полный ключ шифрования представляется в виде конкатенации 32 – битовых подключей: K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7. В процессе шифрования используется один из этих подключей – в зависимости от номера раунда и режима работы алгоритма.

Вторая операция – табличная замена. После наложения ключа субблок N_1 разбивается на 8 частей по 4 бит, значение каждой из которых заменяется в соответствии с таблицей замены для каждой части субблока. Затем выполняется побитовый циклический сдвиг субблока влево на 11 бит.

Табличные замены. Блок подстановки S-box (Substitution box) состоит из 8 – узлов замены (S – блоков замены) S_1, S_2, …, S_8 с памятью 64 бит каждый. Поступающий на блок подстановки S 32 битовый вектор разбивают на 8 последовательно идущих 4 – битовых вектора, каждый из которых преобразуется в 4 – битовый вектор соответствующим узлом замены. Каждый узел замены можно представить в виде таблицы – перестановки 16 4 – битовых двоичных чисел в диапазоне 0000…1111. Входной вектор указывает адрес строки в таблице, а число в этой строке является выходным вектором. Затем 4 – битовые выходные векторы последовательно объединяют в 32 – битовый вектор. Узлы замены (таблицы - перестановки) представляют собой ключевые элементы, которые являются общими для сети ЭВМ и редко изменяются. Эти узлы замены сохраняются в секрете.

Алгоритм, определяемый ГОСТ 28147- 89, предусматривает четыре режима работы: *простой замены, гаммирования, гаммирования с обратной*

связью и *генерации имитовставок* (криптографических контрольных сумм, вычисляемых с использованием ключа шифрования и предназначенных для проверки целостности сообщения). При обмене информацией имитовставки служат своего рода дополнительным средством контроля. Особенно полезна имитовставка для проверки правильности расшифрования ключевой информации при использовании многоключевых схем.

Алгоритм ГОСТ 28147- 89 является очень стойким алгоритмом – в настоящее время для его раскрытия не предложено более эффективных методов, чем метод «грубой силы». Его высокая стойкость достигается в первую очередь за счёт большой длины ключа – 256 бит. При использовании секретной синхропосылки эффективная длина ключа увеличивается до 320 бит, а засекречивание таблицы замен прибавляет дополнительные биты. Кроме того, криптостойкость зависит от количества раундов преобразований, которых по ГОСТ 28147- 89 должно быть 32 (полный эффект рассеивания входных данных достигается уже после 8 раундов).

Стандарт шифрования данных AES

В 1997 г. Американский институт стандартизации NIST (National Institute of Standards & Technology) объявил конкурс на новый стандарт симметричного криптоалгоритма, названного AES (Advanced Encryption Standard). К его разработке были подключены самые крупные центры криптологии всего мира. Победитель этого соревнования фактически становился мировым криптостандартом на ближайшие 10 – 20 лет.

К криптоалгоритмам - кандидатам на новый стандарт AES – были предъявлены следующие требования:

- алгоритм должен быть симметричным;
- алгоритм должен быть блочным шифром;
- алгоритм должен иметь длину блока 128 бит и поддерживать три длины ключа: 128, 192, 256 бит.

Дополнительно разработчикам криптоалгоритмов рекомендовалось:

- использовать операции, легко реализуемые как аппаратно (в микрочипах), так и программно (на персональных компьютерах и серверах);
- ориентироваться на 32 – разрядные процессоры;
- не усложнять без необходимости структуру шифра, для того чтобы все заинтересованные стороны были в состоянии самостоятельно провести независимый криптоанализ алгоритма и убедиться, что в нём не заложено каких – либо недокументированных возможностей.

В октябре 2000 г. был объявлен алгоритм – победитель Rijndael, разработанный двумя криптографами из Бельгии, Винсентом Риджменом и Джоан Даймен. Этот алгоритм стал новым стандартом шифрования данных AES.

Алгоритм AES не похож на большинство известных алгоритмов симметричного шифрования, структура которых носит название «сеть

Фейстеля» и аналогична российскому ГОСТ 28147-89. В отличие от отечественного стандарта шифрования, алгоритм AES представляет каждый блок обратимых данных в виде двухмерного байтового массива размером 4x4, 4x6 или 4x8 в зависимости от установленной длины блока (допускается использование нескольких фиксированных размеров шифруемого блока информации). Далее на соответствующих этапах производятся преобразования либо над независимыми столбцами, либо над независимыми строками, либо вообще над отдельными байтами.

Алгоритм AES состоит из определённого количества раундов (от 10 до 14, что зависит от размера блока и длины ключа) и выполняет четыре преобразования:

BS (ByteSub) – табличная замена каждого байта массива с использованием таблицы замены (подстановок);

SR (ShiftRow) – сдвиг строк массива по правилу:

- элементы первой строки остаются без изменения,
- элементы второй строки циклически побайтно сдвигаются влево на 1 байт,
- элементы третьей строки циклически побайтно сдвигаются влево на 2 байта и т.д. до последней строки массива.

Например, для массива 4x4 строки 2, 3, 4 сдвигаются соответственно на 1, 2 и 3 байта;

MC (MixColumn) – операция над независимыми столбцами массива, когда каждый столбец по определённому правилу умножается на фиксированную матрицу *c(x);*

AK(AddRoundKey) – добавление ключа. Каждый бит массива складывается по модулю 2 с соответствующим битом ключа раунда, который в свою очередь определённым образом вычисляется из ключа шифрования.

Последовательность операций при шифровании выглядит следующим образом:

AK, {BS, SR, MC, AK} (повторяется R – 1 раз), BS, SR, AK.

Количество раундов шифрования R в алгоритме AES переменное (10, 12 или 14 раундов) и зависит от размеров блока и ключа шифрования (для ключа также предусмотрено несколько фиксированных размеров).

Восстановление (расшифровывание) шифртекста выполняется с помощью обратного алгоритма.

Все преобразования в шифре AES имеют строгое математическое обоснование. В структуре алгоритма заложена возможность параллельного исполнения некоторых операций, что может поднять скорость шифрования на многопроцессорных рабочих станциях в 4 раза. Алгоритм AES стал новым стандартом шифрования данных благодаря ряду преимуществ перед другими алгоритмами. Прежде всего, он обеспечивает высокую скорость шифрования на всех платформах: как при программной, так и при аппаратной реализации. Кроме того, требования к ресурсам для его работы минимальны, что важно при его использовании в устройствах, обладающих ограниченными вычислительными возможностями.

Недостатком алгоритма AES можно считать лишь его нетрадиционную схему. Свойства алгоритмов, основанных на «сети Фейстеля», хорошо исследованы, а AES может содержать скрытые уязвимости, которые могут обнаружиться только по прошествии какого-то времени с момента его широкого распространения.

Для шифрования данных применяются и другие симметричные блочные криптоалгоритмы.

АСИММЕТРИЧНЫЕ КРИПТОСИСТЕМЫ

Эффективными системами криптографической защиты данных являются асимметричные криптосистемы, называемые также криптосистемами с открытым ключом. В таких системах для шифрования данных используется один ключ, а для дешифрования – другой ключ (отсюда и название - асимметричные).

Криптографические системы с открытым ключом
КРИПТОСИСТЕМА ШИФРОВАНИЯ ДАННЫХ RSA

Наиболее перспективными системами криптографической защиты данных являются системы с открытым ключом, использующие асимметричные алгоритмы шифрования. В таких системах для шифрования данных используется один ключ, а для дешифрования - другой. Первый ключ не является секретным и может быть опубликован для использования всеми участниками системы, которые шифруют данные. Дешифрование данных с помощью известного ключа невозможно. Для дешифрования данных получатель шифрованного текста использует второй ключ, который является секретным.

Наиболее развитым методом криптографической защиты информации с известным ключом является RSA. Алгоритм RSA в 1978г. предложили три автора: Р. Райвест (**R**ivest), А. Шамир (**S**hamir) и А. Адлеман (**A**dleman), который и был назван по первым буквам фамилий его авторов. Этот алгоритм стал первым полноценным алгоритмом с открытым ключом, который может работать как в режиме шифрования данных, так и в режиме электронной цифровой подписи. Для рассмотрения метода RSA необходимо вспомнить некоторые термины.

Под *простым* числом понимают такое число, которое делится только на 1 и на само себя. *Взаимно простыми* числами называют такие числа, которые не имеют ни одного общего делителя, кроме 1. Под результатом операции $i \bmod j$ понимают остаток от целочисленного деления i на j.

Для использования алгоритма RSA необходимо сначала сгенерировать открытый и секретный ключи, выполнив следующие шаги:

1. Выбрать два *очень больших* простых числа p и q равной длины и хранить в секрете.
2. Определить n – модуль как результат умножения p на q ($n=pq$).

3. Выбрать большое случайное число *d*, которое должно быть взаимно простым с результатом умножения (p-1)(q-1) (вычисляется функция Эйлера).
4. Определить такое число *e*, для которого является истинным следующее соотношение: $ed \bmod ((p-1)(q-1))=1$.
5. Назвать открытым ключом числа *e* и *n*, а секретным ключом – числа *d* и *n*.

Чтобы зашифровать данные по известному ключу {*e, n*}, необходимо разделить открытый текст на блоки, каждый из которых может быть представлен в виде числа M(i)=0, 1, …,n-1, зашифровать текст, рассматриваемый как последовательность *чисел M(i),* используя формулу: $\boldsymbol{C(i)=M(i)^e \bmod (n)}$**.** В качестве алгоритма быстрого вычисления значения *C(i)* используют ряд последовательных возведений в квадрат целого *M(i)* и умножений на *M(i)* с приведением по модулю *n*. Обращение функции $C(i)=M(i)^e \bmod (n)$, т.е. определение значения *M(i)* по известным значениям *C(i), e, n,* практически неосуществимо при $n \approx 2^{512}$.

Для дешифрования данных используют секретный ключ {*d, n*}. Выполняются следующие вычисления: $\boldsymbol{M(i) = C(i)^d \bmod (n)}$**.** В результате будет получено множество *чисел M(i),* которое представляет собой исходный текст.

Чтобы алгоритм RSA имел практическую ценность, необходимо иметь возможность без существенных затрат генерировать большие простые числа, уметь оперативно вычислять значения ключей *e* и *d.*

Криптоаналитику известны лишь значения *e, n*. Если бы он смог разложить число *n* на множители *p* и *q*, то он узнал бы тройку чисел *p, q, e,* вычислил значение функции Эйлера *(p-1)(q-1)* и определил значение секретного ключа *d.*

Но разложение очень большого *n* на множители вычислительно неосуществимо (при условии, что длины выбранных *p* и *q* составляют не менее 100 десятичных знаков).

Например, пусть необходимо зашифровать сообщение «ЕДА». Для простоты вычислений будем использовать очень маленькие числа, не в пример используемых на практике.

1. Выберем $p=3$ и $q=11$.
2. Определим $\boldsymbol{n=3*11=33}$.
3. Найдём $(p-1)(q-1)=2*10=20$. Следовательно, в качестве *d* выберем любое число, которое является взаимно простым с 20, например $\boldsymbol{d=3}$.
4. Выберем число *e*. В качестве такого числа может быть взято любое число, для которого удовлетворяется соотношение $e*3 \bmod (20)=1$, например $\boldsymbol{e=7}$.
5. Представим открытый текст как последовательность целых чисел в диапазоне 0…32. Пусть букве Е соответствует число 6, букве Д -

число 5, а букве А – число 1. Тогда сообщение будет представлено в виде последовательности чисел 6 5 1. Зашифруем это сообщение, используя ключ {*7, 33*}

$$C_1=6^7 \bmod (33)=279936 \bmod (33)=30,$$

$$C_2=5^7 \bmod (33)=78125 \bmod (33)=14,$$

$$C_3=1^7 \bmod (33)=1 \bmod (33)=1.$$

Расшифруем сообщение {*30 14 1*}, полученное в результате шифрования по известному ключу, на основе секретного ключа {*3, 33*}:

$$M_1=30^3 \bmod (33)=27000 \bmod (33)=6,$$

$$M_2=14^3 \bmod (33)=2744 \bmod (33)=5,$$

$$M_3=1^3 \bmod (33)=1 \bmod (33)=1.$$

Таким образом, в результате дешифрования сообщения получено исходное сообщение «ЕДА».

Криптоалгоритм RSA всесторонне исследован и признан стойким при достаточной длине ключей. При увеличении мощности процессоров стойкость к атаке не теряется, т.к. в этом случае появится возможность применять более длинные ключи, что повышает стойкость RSA.

Быстродействие RSA существенно ниже быстродействия DES (примерно в 100 раз), а программная и аппаратная реализация криптоалгоритма RSA гораздо сложнее, чем DES. Поэтому криптосистема RSA, как правило, используется при передаче небольшого объём сообщений.

Кроме метода RSA есть ещё несколько криптосистем с открытым ключом, в той или иной мере распространённых в теоретическом и практическом плане.

Система шифрования Эль-Гамаля

Система Эль-Гамаля основана на трудности вычисления дискретных логарифмов в конечных полях. Эта система, предложенная в 1985г., может быть использована как для шифрования, так и цифровых подписей. Для того чтобы генерировать пару ключей (открытый ключ – секретный ключ), сначала выбирают некоторое большое простое число P и большое целое число G, причём G <P. Числа P и G могут быть распространены среди группы пользователей.

Затем выбирают случайное целое число X, причём X<P. Число X является секретным ключом и должно храниться в секрете. Далее вычисляют $Y=G^X \bmod P$. Число Y является открытым ключом.

Для того чтобы шифровать сообщение M, выбирают случайное число K, 1<K<P-1, такое, что числа K и P-1 являются взаимно простыми. Затем вычисляют $a=G^K \bmod P$, $b=Y^K M \bmod P$. Пара чисел (a, b) является шифртекстом. Заметим, что длина шифртекста вдвое больше длины исходного

открытого текста M. Для дешифрования шифртекста (a, b) вычисляют $M=b/a^x \mod P$.

Мак-Элис предложил криптосистему, основанную на кодах, исправляющих ошибки. Вычисления в этой системе реализуются в несколько раз быстрее, чем в системе RSA.

Схема шифрования Полига – Хеллмана

Схема Полига – Хеллмана сходна со схемой шифрования RSA. Она представляет собой несимметричный алгоритм, т.к. используются различные ключи для шифрования и дешифрования. В то же время эту схему нельзя отнести к классу криптосистем с открытым ключом, так как ключи легко выводятся один из другого. Оба ключа нужно держать в секрете.

В отличие от алгоритма RSA в этой схеме число n не определяется через два больших простых числа: число n должно оставаться частью секретного ключа. Если кто-либо узнает значения e и n, он сможет вычислить значение d.

Не зная значений e или d, криптоаналитик будет вынужден вычислить значение $e=\log_p C(i) \pmod n$. Известно, что это является трудной задачей. Схема шифрования Полига – Хеллмана запатентована в США и Канаде.

ПОСТРОЕНИЕ И ИСПОЛЬЗОВАНИЕ ХЕШ-ФУНКЦИЙ

Под термином хеш-функция понимается функция, отображающая электронные сообщения произвольной длины (иногда длина сообщения ограничена, но достаточно большим числом), в значения фиксированной длины. Последнее часто называют хеш – кодами. Таким образом, у всякой хеш – функции *h* имеется большое количество коллизий, т.е. пар значений *x* и *y* таких, что $h(x) = h(y)$. Основное требование, предъявляемое криптографическими приложениями к хеш – функциям, состоит в отсутствии эффективных алгоритмов поиска коллизий. Хеш – функция, обладающая такими свойствами, называется хеш – функцией, свободной от коллизий. Кроме того хеш – функция должна быть односторонней, т.е. функцией, по значению которой вычислительно трудно найти её аргумент, в то же время, функцией, для аргумента которой вычислительно трудно найти другой аргумент, который давал бы то же самое значение функции.

Схема электронной цифровой подписи – основная сфера применения хеш – функций. Так как используемые на практике схемы электронной подписи не приспособлены для подписания сообщений произвольной длины, а процедура, состоящая в разбиении сообщения на блоки и генерации подписи для каждого блока по отдельности, крайне неэффективна, единственным разумным решением представляется применение схемы подписи к хеш – коду сообщения. Таким образом, хеш – функции вместе со схемами электронной цифровой подписи предназначены для решения задач обеспечения целостности и достоверности передаваемых и хранимых на носителях информации электронных сообщений. В прикладных информационных системах требуется применение так называемых криптографически стойких хеш – функций. Под

термином «криптографически стойкая хеш - функция» понимается функция *h*, которая является односторонней и свободной от коллизий.

Введём обозначения. Хеш – функция *h* обозначается как *h(α)* и *h(α, β)* для одного и двух аргументов соответственно. Хеш – код функции *h* обозначается как *H*. При этом $H_0 = 1$ обозначает начальное значение (вектор инициализации) хеш – функции. Под обозначением $\oplus$ будет пониматься операция сложения по модулю 2 или логическая операция XOR («Исключающая ИЛИ»). Результат шифрования блока *B* блочным шифром на ключе *k* обозначается $E_k(B)$.

Приведём пример. Предположим, необходимо подписать при помощи заданного алгоритма электронной цифровой подписи достаточно длинное сообщение *M*. В качестве шифрующего преобразования в хеш – функции будет использоваться процедура шифра DES с ключом *k*. Тогда, чтобы получить хеш – код *H* сообщения *M* при помощи хеш – функции *h*, необходимо выполнить следующую итеративную операцию:

$$H_i = E_{Hi-1}(M_i) \oplus M_i, \text{ где } i=\overline{1,n};\ H_0 = 1;\ M = M_1, M_2, \ldots, M_n,$$

где сообщение *M* разбито на *n* 64-битных блока. Хеш – кодом данной хеш – функции является значение $H=(M, I)=H_n$.

Таким образом, на вход схемы электронной цифровой подписи вместо длинного сообщения *M* (как правило, несколько сотен или тысяч байтов) подаётся хеш – код H_n, длина которого ограничена длиной блока шифра DES, т.е. 64 битами. При этом в силу криптографической стойкости используемой хеш – функции практическая стойкость самой схемы подписи будет оставаться той же, что и при подписи сообщения *M*, в то время как эффективность всего процесса подписи электронного сообщения будет резко увеличена.

Стойкость (безопасность) хеш – функций

Один из основных методов криптоанализа хеш – функций заключается в проведении криптоаналитиком (нарушителем) атаки, основанной на «парадоксе дня рождения», основанной на элементарной теории вероятностей.

Атака, основанная на «парадоксе дня рождения», заключается в следующем. Пусть $\alpha\sqrt{n}$ предметов выбираются с возвращением из некоторого множества с помощью *n*. Тогда вероятность того, что два из них окажутся одинаковыми, составляет $1-\exp(-\alpha^2/2)$.

Практически это означает, что в случайно подобранной группе из 24 человек вероятность наличия двух лиц с одним и тем же днём рождения составляет значение около ½. Этот хорошо изученный парадокс и лежит в основе криптоанализа хеш – функции.

К основным методам предотвращения положительного криптоанализа можно отнести: увеличение длины получаемых хеш – кодов (увеличение мощности хеш – кодов) и выполнение требования интегрированности шифрующего преобразования.

Отечественный стандарт хеширования ГОСТ Р 34.11-2012 «Функции хеширования»

Стандарт ГОСТ Р 34.11-94 являлся обязательным для применения в качестве алгоритма хеширования в государственных организациях РФ и ряде коммерческих организаций до 1 января 2013 года, когда вступил в силу ГОСТ Р 34.11-2012 и заменил его. Алгоритм хеширования можно описать в виде схемы:

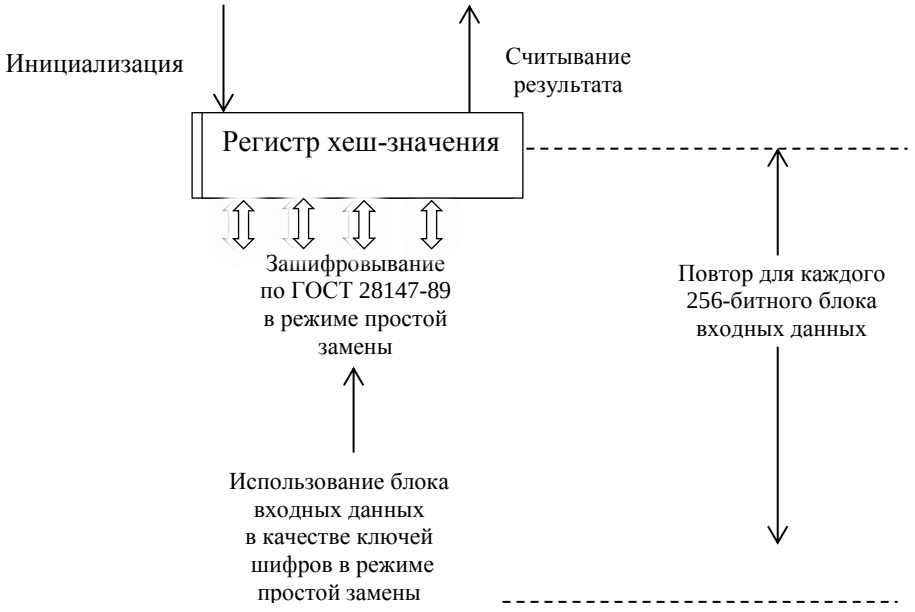

Рис. 12. Схема хеширования

Рассмотрим упрощённый процесс хеширования и его результат на примере. Хешируемое слово ДВА. Коэффициенты *p=7, q=3*. Вектор инициализации H_0 выберем равным 9 (выбираем случайным образом). Определим *n=p q=7*3=21*. Слово ДВА в исловом эквиваленте можно представить как 531 (по номерам букв в алфавите: ***А-1***, Б-2, ***В-3***, Г-4, ***Д-5*** …). Тогда хеш-код сообщения 531 получается следующим образом:

1-я итерация: $M_1+H_0=5+9=14$; $[M_1+H_0]^2 \bmod (21)=14^2 \bmod (21)=$**$7= H_1$;**

2-я итерация: $M_2+H_1=3+7=10$; $[M_2+H_1]^2 \bmod (21)=10^2 \bmod (21)=$**$16= H_2$;**

3-я итерация: $M_3+H_2=1+16=17$; $[M_3+H_2]^2 \bmod (21)=17^2 \bmod (21)=$**$16= H_3$;**

В итоге получаем хеш-код сообщения ДВА, равный 16.

Практические методы построения хеш-функций можно условно разделить на три группы: методы построения хеш – функций на основе какого - либо алгоритма шифрования (например, рассмотренных раньше), методы построения хеш – функций на основе какой – либо известной вычислительно трудной математической задачи и методы построения хеш – функций «с нуля». Хеш – функции можно строить на основе наиболее известных блочных шифров DES и FEAL. В качестве примера хеш – функций, построенных на основе

вычислительно трудной математической задачи, можно привести функцию из рекомендаций МККТТ X.509 (смотрите стандарты: ГОСТ Р 34.11-94 и ГОСТ Р 34.11-2012).

Пример хеширования с использованием хеш – функции X.509 может выглядеть следующим образом. Пусть необходимо получить хеш-код сообщения: «HASHING» с параметрами *p=17, q=19.*

Порядок вычисления хеш – кода:

1. получить значение модуля *n=p q=17*19=323;*
2. представить сообщение «HASHING» в виде кодов символов ASCII:

H	A	S	H	I	N	G
72	65	83	72	73	78	71

3. представить коды ASCII битовой строкой:

72	65	83	72	73	78	71
01001000	01000001	01010011	01001000	01001001	01001110	01000111

разбить байт пополам (разбиение октета на полуактеты), добавить в начало полубайта единицы и получить хешируемые блоки M_i:

M_1	M_2	M_3	M_4	M_5	M_6	M_7
11110100	11111000	11110100	11110001	11110101	11110011	11110100
M_8	M_9	M_{10}	M_{11}	M_{12}	$M3_1$	M_{14}
11111000	11110100	11111001	11110100	11111110	11110100	11110111

4. выполнить итеративные шаги:

первая итерация $M_1 = 11110100$

$\oplus$

$\underline{H_0 = I = 00000000}$

$H_0 \oplus M_1 = 11110100_2 = 244_{10}$

$H_1 = [(H_0 \oplus M_1)]^2 \pmod{323} = 244^2 \pmod{323} = 104_{10} = 01101000_2$

вторая итерация $M_2 = 11111000$

$\oplus$

$\underline{H_1 = 01101000}$

$H_1 \oplus M_2 = 10010000_2 = 144_{10}$

$H_2 = [(H_1 \oplus M_2)]^2 \pmod{323} = 144^2 \pmod{323} = 64_{10} = 01000000_2$

……

i итерация ….

Пример легко продолжить самостоятельно.

В итоге следует отметить, что при выборе практически стойких и высокоэффективных хеш – функций можно руководствоваться следующими эвристическими принципами, сформулированными Р. Ривестом:

- любой из известных алгоритмов построения коллизий не должен быть эффективнее метода, основанного на «парадоксе дня рождения»;
- алгоритм должен допускать эффективную программную реализацию на 32/64 разрядном процессоре;
- алгоритм не должен использовать сложных структур данных и подпрограмм;
- алгоритм должен быть оптимизирован с точки зрения его реализации на микропроцессорах типа Intel.

ЭЛЕКТРОННАЯ ЦИФРОВАЯ ПОДПИСЬ

Безбумажная информатика даёт ряд преимуществ при обмене документами (приказами, распоряжениями, письмами, постановлениями и т.д.) по сети связи или на машинных носителях. В этом случае временные затраты (распечатка, пересылка, ввод полученного документа с клавиатуры) существенно снижаются, убыстряется поиск документов, снижаются затраты на их хранение и т.д. Но при этом возникает проблема аутентификации автора документа и самого документа (т.е. установление подлинности подписи и отсутствия изменений в полученном документе). Эти проблемы в обычной (бумажной) информатике решаются за счёт того, что информация в документе жестко связана с физическим носителем (бумагой). На машинных носителях такой связи нет.

Проблема аутентификации является актуальной в вычислительных сетях, электронных системах управления и вообще там, где необходимо удостовериться в подлинности полученного по каналам связи или на машинных носителях сообщения (документа).

Задачи аутентификации можно разделить на следующие типы: аутентификация абонента, аутентификация принадлежности абонента к заданной группе, аутентификация хранящихся на машинных носителях документов.

Наиболее важной является аутентификация документов (или файлов). Если рассматривать случай обмена секретными документами (военная или дипломатическая связь), то с большой степенью уверенности можно предположить, что обмен осуществляют доверяющие и достойные доверия стороны. Но возможно, что обмен находится под наблюдением и управлением нарушителя, который способен выполнять сложные вычисления и затем либо создавать собственные документы, либо перехватывать и изменять документы законного источника. Иными словами, в этом случае, когда защищаться нужно только от противника – «свои» подвести не могут. В коммерческом мире верно почти обратное, т.е. передатчик и приёмник хотя и «свои», но могут обмануть друг друга даже в большей степени, чем посторонние.

В первом случае («свои не обманывают») схему аутентификации построить несложно. Необходимо снабдить передающего и принимающего абонента надёжным шифром и комплектом уникальных ключей для каждого

пересылаемого документа, обеспечив тем самым защищённый канал связи. Отметим, что рассматриваемая задача предъявляет высокие требования к системе шифрования. Так, метод гаммирования в этом случае не подходит, так как нарушитель, анализируя открытый и шифрованный текст, получит гамму и сможет навязать любой нужный ему текст. Но существуют быстрые алгоритмы шифрования, удовлетворяющие предъявляемым требованиям.

Во втором случае (когда «любой из абонентов может обмануть») аналогичный подход, основанный на классической криптографии, неприменим, т.к. имеется принципиальная возможность злоумышленных действий одной из сторон, владеющей секретным ключом. Например, приёмная сторона может сгенерировать любой документ, зашифровать его на имеющемся ключе, общем для приёмника и передатчика, а потом заявить, что он получил его от законного передатчика. Здесь необходимо использовать схемы, основанные на двухключевой криптографии. В таких случаях у передающего абонента сети имеются свои секретные ключи подписи, а у принимающего абонента – несекретный открытый ключ подписи передающего абонента. Этот открытый ключ можно трактовать как набор проверочных соотношений, позволяющих судить об истинности подписи передающего абонента, но не позволяющих восстановить секретный ключ подписи. Передающий абонент несёт единоличную ответственность за свой секретный ключ. Никто, кроме него, не в состоянии сгенерировать корректную подпись. Секретный ключ передающего абонента можно рассматривать как личную печать, и владелец должен всячески ограничивать доступ к нему посторонних лиц.

Общепринятой является следующая модель аутентификации, в которой функционируют четыре участника: *A* – передатчик, *B* – приёмник, *C* – противник,
D – арбитр. В этом случае *A* посылает сообщения, *B* принимает, *C* пытается совершить злоумышленные действия, а *D* принимает решение в спорных случаях, т.е. определяет, утверждения чьей стороны с наибольшей вероятностью являются ложными. Естественно, в качестве *C* могут выступать *A* и *B*. Целью аутентификации документов является защита от возможных видов злоумышленных действий, среди которых выделим:

1. активный перехват – нарушитель, подключившийся к сети, перехватывает документы (файлы) и изменяет их;
2. маскарад – абонент *C* посылает документ от имени *A*;
3. ренегатство – абонент *A* заявляет, что не посылал сообщение абоненту *B*, хотя на самом деле посылал;
4. переделка – абонент *B* изменяет документ и утверждает, что данный документ (изменённый) получил от абонента;
5. подмена – абонент *B* формирует документ (новый) и заявляет, что получил его от абонента *A*;
6. повтор – абонент *C* повторяет ранее переданный документ, который абонент *A* послал абоненту *B*.

Эти виды злоумышленных действий наносят существенный вред функционированию банковских, коммерческих структур, государственным

предприятиям и организациям, частным лицам, применяющим в своей деятельности компьютерные информационные технологии. Кроме того, возможность злоумышленных действий подрывает доверие к компьютерной технологии. В связи с этим задача аутентификации представляется важной.

При выборе алгоритма и технологии аутентификации сообщений в сети необходимо предусмотреть надёжную защиту от всех вышерассмотренных видов злоумышленных действий. Наряду с такими характеристиками системы аутентификации, как быстродействие и требуемый для реализации объём памяти, степень защищённости (стойкость) от вышеперечисленных угроз является важнейшим параметром.

Математические схемы, используемые в алгоритмах, реализующих электронную цифровую подпись (ЭЦП), основаны на так называемых однонаправленных функциях.

Технология применения систем ЭЦП предусматривает сеть абонентов, посылающих друг другу электронные документы. Некоторые из этих абонентов могут только проверять подписанные другими сообщения, другие (назовём их абонентами с правом подписи) могут как проверять, так и самостоятельно подписывать сообщения. Кроме того, могут быть случаи, когда кто-либо может ставить свою ЭЦП только в качестве второй подписи после подписи определённого абонента – начальника (например, директор - бухгалтер).

Постановка и проверка подписи

Чтобы поставить ЭЦП под конкретным документом, необходимо проделать довольно большой объём вычислительной работы. Эти вычисления разбиваются на два этапа: генерация ключей и подписание документа.

Генерация ключей. На этом этапе для каждого абонента формируется пара ключей: секретный и открытый. Секретный ключ хранится абонентом в тайне. Он используется для формирования подписи. Открытый ключ связан с секретным с помощью особого математического соотношения. Открытый ключ известен всем другим пользователям сети и предназначен для проверки подписи. Его следует рассматривать как необходимый инструмент для проверки, позволяющий определить автора подписи и достоверность электронного документа, но не позволяющий вычислить секретный ключ.

Возможны два варианта проведения этого этапа. Естественным представляется вариант, когда генерацию ключей абонент может осуществлять самостоятельно. Не исключено, что в определённых ситуациях эту функцию целесообразно передать центру, который будет вырабатывать пару секретный – открытый ключи для каждого абонента и заниматься их распространением. Второй вариант имеет ряд преимуществ административного характера, однако обладает принципиальным недостатком - у абонента нет гарантии, что его личный секретный ключ является уникальным. Другими словами, можно сказать, что здесь все абоненты находятся «под колпаком» центра и центр может подделать любую подпись.

Подписание документа.

Прежде всего, документ «сжимают» до нескольких десятков или сотен байт с помощью хеш – функции, которая должна удовлетворять ряду условий:

- быть чувствительна к всевозможным изменениям в тексте, таким, как вставки, выбросы, перестановки и т.п.,
- обладать свойствами необратимости, т.е. задача подбора документа, который обладал бы требуемым значением хеш – функции, вычислительно неразрешима,
- вероятность того, что значения хеш – функций двух различных документов (в независимости от их длин) совпадут, должна быть ничтожна мала.

Далее к полученному значению хеш – функции применяют то или иное математическое преобразование (в зависимости от выбранного алгоритма ЭЦП: RSA, EGSA – Эль Гамаля, DSA, отечественный стандарт цифровой подписи) и получают собственно подпись документа. Эта подпись может иметь вполне читаемый, «буквенный» вид, но зачастую её представляют в виде последовательности «нечитаемых» символов. ЭЦП может храниться вместе с документом, например, стоять в его начале или конце, либо в отдельном файле.

Проверка подписи. При проверке подписи проверяющий должен располагать открытым ключом абонента, поставившего подпись. Этот ключ должен быть аутентифицирован, то есть проверяющий должен быть полностью уверен, что данный открытый ключ соответствует тому абоненту, который выдаёт себя за его «хозяина». В случае, когда абоненты самостоятельно обмениваются ключами, эта уверенность может подкрепляться связью по телефону, личным контактом или любым другим способом. В случае, когда абоненты действуют в сети с выделенным центром, открытые ключи абонентов подписываются (сертифицируются) центром, и непосредственный контакт абонентов между собой (при передаче или подтверждении подлинности ключей) заменяется на контакт каждого из них в отдельности с центром.

Процедура проверки ЭЦП состоит также из двух этапов: вычисления хеш – функции документа и собственно математических вычислений, предусмотренных в данном алгоритме подписи. Последнее заключается в проверке того или иного соотношения, связывающего хеш – функцию документа, подпись под этим документом и открытый ключ подписавшего абонента. Если рассматриваемое соотношение оказывается выполнимым, то подпись признаётся правильной, а сам документ – подлинным, в ином случае документ считается изменённым, а подпись под ним – недействительной.

Использование ЭЦП значительно расширяет и без того богатые возможности электронной почты, телефакса, телекса и др.

С новыми разновидностями ЭЦП можно ознакомиться на ближайших ежегодных выставках «Банк и офис», «Защита информации», «Безопасность» и др.

ЛАБОРАТОРНЫЕ РАБОТЫ (7 СЕМЕСТР)

ЛАБОРАТОРНАЯ РАБОТА № 1

ТЕМА: «Вирусы и антивирусы»
Необходимо:

1. Найти информацию (4-5стр.) на тему «Защита от вирусов» и оформить её в виде файла с именем Анти.TXT.
2. Написать программу, выполняющую шифрование строки текста (2-254 буквы), используя «сказочный» алгоритм фильма «Королевство кривых зеркал». К примеру: исходный текст: «первый опыт». Результат шифрования: «тыпо йывреп».

Ответьте на вопросы и выполните задания:

1. Как оптимизировать заданное шифрование?
2. Какова правовая основа защиты информации?
3. Перечислить виды и принципы защиты информации.
4. Что такое компьютерные вирусы и какова их классификация?
5. Что подразумевается под программами – шпионами? Перечислить их виды.

ЛАБОРАТОРНАЯ РАБОТА № 2

ТЕМА: «Парольная защита»
Необходимо:

1. Собрать 8-10 страниц текста на тему лабораторной работы №1 под именем IN.TXT.
2. В файле IN.RTF организовать автооглавление файла IN.TXT.
3. Скрыть файл IN.RTF.
4. Сжать копию файла IN.TXT с именем IN.DOC.
5. Установить парольную защиту для одного из Ваших файлов этой лабораторной работы.
6. Написать программу, выполняющую дешифрование строки текста (2-254 буквы), используя «сказочный» алгоритм фильма «Королевство кривых зеркал». К примеру: шифрованный текст: «тыпо йывреп». Результат дешифрования: «первый опыт».

Ответьте на вопросы и выполните задания:

1. Назовите основные каналы распространения вирусов и других вредоносных программ.
2. Каковы виды антивирусных программ и комплексов, их характеристики?
3. Что такое пароль и как установить парольную защиту?
4. Чем объясняется актуальность антивирусной защиты?

ЛАБОРАТОРНАЯ РАБОТА № 3

ТЕМА: «Преобразование информации методом перестановки». Шифр перестановки «скитала»

Шифрование перестановкой заключается в том, что символы шифруемого текста переставляются по определённому правилу в пределах некоторого блока этого текста. При достаточной длине блока и сложном неповторяющемся порядке перестановки можно достигнуть приемлемой стойкости шифра. Шифры перестановки являются самыми простыми и, вероятно, самыми древними шифрами.

Необходимо:

1. Взять файл IN.TXT.
2. Используя метод перестановки «скитала», зашифровать выбранный текст.
3. Записать зашифрованный текст в файл Z3.TXT.

Ответьте на вопросы и выполните задания:

1. Дать определение понятия «шифрование».
2. Дать определение понятия «алфавит» и привести примеры алфавитов.
3. Перечислить классические методы шифрования.
4. Как используется метод перестановки «скитала» для шифрования текста?

ЛАБОРАТОРНАЯ РАБОТА № 4

ТЕМА: «Преобразование информации методом перестановки»

Необходимо:

1. Выполнить дешифрование текста, содержащегося в файле Z3.TXT, зашифрованного в лабораторной работе № 3.
2. Сохранить полученный текст в файле R3.TXT.

Ответьте на вопросы и выполните задания:

1. Дать определение понятия «дешифрование».
2. Что необходимо знать для дешифрования текста, зашифрованного методом «скитала»?
3. Каков алгоритм дешифрования текста, к которому было применено шифрование методом перестановки «скитала»?

ЛАБОРАТОРНАЯ РАБОТА № 5

ТЕМА: «Шифрование методом простой замены (подстановки)»

При шифровании заменой (подстановкой) символы шифруемого текста заменяются символами того же или другого алфавита с заранее установленным правилом замены. В шифре простой замены каждый символ исходного текста заменяется символами того же алфавита одинаково на всём протяжении текста. Часто шифры простой замены называют шифрами одноалфавитной подстановки.

Необходимо:

1. Написать программу, шифрующую исходный текст методом простой замены. Каждая буква исходного текста заменяется буквой, стоящей *справа* от неё в русском «закольцованном» алфавите, т.е. буква «Я» заменяется на «А» (буквы заглавные).
2. Результаты шифрования и исходный текст размещайте в текстовом файле (результат выводить группами по пять букв).

Ответьте на вопросы и выполните задания:

1. Усовершенствовать программу, полученную в п.1, так, чтобы можно было регулировать направление замены, т.е. была бы возможность замены буквы текста на букву *слева* от неё в алфавите.
2. Какой алгоритм Вы применили для замены букв исходного текста?
3. Как организован регулятор направления замены?

ЛАБОРАТОРНАЯ РАБОТА № 6

ТЕМА: «Дешифрование текста, зашифрованного методом простой замены (подстановки)» (см. лаб.№5)

Необходимо:

1. Написать программу, восстанавливающую текст, зашифрованный в усовершенствованной программе предыдущей работы (выводить группами по пять букв). Направление замены регулируйте в программе.

Ответьте на вопросы и выполните задания:

1. Что относится к компьютерным преступлениям?
2. Какие атаки на уровне ОС возможны?
3. Какие возможны атаки на уровне ПО?
4. В чём состоит защита от программных закладок?
5. Что подразумевается под изолированным компьютером?

ЛАБОРАТОРНАЯ РАБОТА № 7

ТЕМА: «Многоалфавитное шифрование методом сложной замены (подстановки)»

Необходимо:

1. Написать программу, использующую систему шифрования Вижинера, для преобразования исходного текста.

Ответьте на вопросы и выполните задания:

1. В чем состоит смысл алгоритма шифрования с использованием квадрата Вижинера?
2. Чем является компьютерная система на языке ПБ?
3. Что такое политика безопасности?
4. Каким образом может быть выражена ПБ компьютерной системы?
5. Дать определение понятия «модель безопасности», перечислить их базовые принципы.

ЛАБОРАТОРНАЯ РАБОТА № 8

ТЕМА: «Дешифрование текста, зашифрованного методом сложной замены (подстановки)»

Необходимо:

1. Написать программу, использующую систему шифрования Вижинера, для преобразования зашифрованного текста в исходный.

Ответьте на вопросы и выполните задания:

1. Какой алгоритм используется для восстановления исходного текста зашифрованного по системе Вижинера?
2. Объяснить структуру монитора обращений.
3. Перечислить основные требования к реализации диспетчера доступа.
4. Дать определение дискреционного управления доступом.
5. Дать определение мандатного управления доступом.

ЛАБОРАТОРНЫЕ РАБОТЫ (8 СЕМЕСТР)

ЛАБОРАТОРНАЯ РАБОТА № 9

ТЕМА: «Шифрование с использованием метода гаммирования с бесконечной гаммой»

Необходимо:

1. Написать программу, использующую метод гаммирования с бесконечной гаммой, для преобразования исходного текста.
2. Одной из функций программы должна быть функция генерации ПСЧ.

Ответьте на вопросы и выполните задания:

1. Доказать на контрольном примере правильность работы алгоритма программы.
2. Объяснить смысл ролевой политики безопасности.

ЛАБОРАТОРНАЯ РАБОТА № 10

ТЕМА: «Дешифрование текста, зашифрованного методом гаммирования с бесконечной гаммой»

Необходимо:

1. Написать программу, использующую метод гаммирования с бесконечной гаммой, для преобразования зашифрованного текста в исходный.

Ответьте на вопросы и выполните задания:

1. Доказать на контрольном примере правильность работы алгоритма программы дешифрования.
2. Перечислить стандартные политики и модели безопасности.
3. Объяснить смысл дискреционной политики безопасности.

ЛАБОРАТОРНАЯ РАБОТА № 11

ТЕМА: «Алгоритм шифрования данных DES»

Необходимо:

1. Написать программу, использующую алгоритм шифрования данных DES, для преобразования исходного текста. Использовать «зеркальную» перестановку как начальную, одну замену с использованием шифра Гронсфельда и простую перестановку с использованием шифрующей таблицы.

Ответьте на вопросы и выполните задания:

1. На контрольном примере доказать правильность работы алгоритма программы.
2. Изменить программу так, чтобы блок замены выполнялся 16 раз.
3. Укажите достоинства и недостатки стандарта шифрования DES.

ЛАБОРАТОРНАЯ РАБОТА № 12

ТЕМА: «Дешифрование текста, зашифрованного с использованием алгоритма шифрования данных DES»

Необходимо:

1. Написать программу для преобразования текста, зашифрованного с использованием алгоритма шифрования данных DES, в исходный. Порядок работы обратный порядку лабораторной работы № 11: перестановка обратная конечной, обратные замены и перестановка обратная начальной.

Ответьте на вопросы и выполните задания:

1. На контрольном примере доказать правильность работы алгоритма программы.
2. Где находит применение алгоритм шифрования DES?
3. Стандарт шифрования AES.

ЛАБОРАТОРНАЯ РАБОТА № 13

ТЕМА: «Асимметричные криптоалгоритмы. Система с открытым ключом»

Необходимо:

1. Написать программу, использующую алгоритм шифрования данных RSA, для преобразования исходного текста.

Ответьте на вопросы и выполните задания:

1. Какова последовательность шифрования текста при использовании алгоритма RSA?
2. Из чего происходит название алгоритма RSA?
3. Как получить открытый ключ?
4. Перечислить другие системы шифрования с открытым ключом.
5. Объяснить особенности системы Эль-Гамаля.

ЛАБОРАТОРНАЯ РАБОТА № 14

ТЕМА: «Асимметричные криптоалгоритмы. Система с открытым ключом»

Необходимо:

1. Написать программу для преобразования текста, зашифрованного с использованием алгоритма шифрования данных RSA, в исходный.

Ответьте на вопросы и выполните задания:

1. Какова последовательность дешифрования данных при использовании алгоритма RSA?
2. Почему криптосистемы называют асимметричными?
3. Как получить закрытый ключ?
4. Чем отличается схема шифрования Полига –Хеллмана от RSA?
5. В чём отличие системы Мак-Элиса от RSA?

ЛАБОРАТОРНАЯ РАБОТА № 15

ТЕМА: «Хеширование сообщений»

Необходимо:

1. Написать программу для получения хеш - кода заданного по номеру i сообщения, используя упрощённый вариант хеш - функции X.509 с параметрами по номеру j. Вариант задания определяется цифрами номера студенческого билета (i – предпоследняя, j – последняя). Вектор инициализации H_0 следует выбирать самостоятельно.

i	сообщение	i	сообщение
0	ПРЕДЕЛ	5	ЧИСЛИТЕЛЬ
1	ИНТЕГРАЛ	6	АМПЕРСАНТ
2	МИНУС	7	КОРЕНЬ
3	МОДУЛЬ	8	ОСТАТОК
4	плюс	9	СТЕПЕНЬ

j	параметры	j	параметры
0	(3, 7)	5	(3, 17)
1	(13, 7)	6	(11, 5)
2	(3, 13)	7	(5, 17)
3	(5, 7)	8	(7, 11)
4	(13, 5)	9	(11, 3)

Ответьте на вопросы и выполните задания:

1. Что понимается под термином хеш-функция?
2. Сформулируйте основное назначение хеш-функции.
3. Объяснить алгоритм хеширования сообщения на своём примере.

ЛАБОРАТОРНАЯ РАБОТА № 16

ТЕМА: «Постановка электронной цифровой подписи»

Необходимо:

1. С помощью программы выполнить шифрование текста и постановку ЭЦП к нему. Алгоритм хеширования и шифрования использовать из предыдущих лабораторных работ по своему усмотрению.

Ответьте на вопросы и выполните задания:

1. Укажите основные этапы технологии постановки электронной цифровой подписи.
2. На контрольном примере доказать правильность работы программы, вычисляющей ЭЦП.

ЛАБОРАТОРНАЯ РАБОТА № 17

ТЕМА: «Проверка электронной цифровой подписи»

Необходимо:

1. С помощью программы выполнить проверку ЭЦП и расшифровать текст, зашифрованный в лабораторной работе № 16.

Ответьте на вопросы и выполните задания:

1. Из каких этапов состоит процедура проверки ЭЦП? Перечислить и объяснить.
2. В каком случае ЭЦП признаётся правильной?

КУРСОВАЯ РАБОТА

К концу 7 семестра студенты выполняют курсовые работы, которые состоят в написании программ, шифрующих или расшифровывающих текст с использованием метода перестановки или замены, изученных в течение семестра, но не вошедших в лабораторные работы. Программа тестируется и отлаживается на контрольных примерах, параллельно оформляется документация к программе по форме, соответствующей курсовой работе.

Результаты шифрования одного студента могут быть полезны другому, задачей которого является дешифрование текста, преобразованного этим методом. По этой причине студенты могут работать парами над отладкой программ. Темы курсовых работ:

№ варианта	Реализовать метод:
1	Шифрование двойной перестановкой
2	Дешифрование двойной перестановкой
3	Шифрование с использованием магического квадрата
4	Дешифрование с использованием магического квадрата
5	Шифрование заменой: «Полибианский квадрат»
6	Дешифрование заменой: «Полибианский квадрат»
7	Шифрование с использованием системы Цезаря (простая замена)
8	Дешифрование с использованием системы Цезаря (простая замена)
9	Шифрование «Аффинная система подстановок Цезаря»
10	Дешифрование «Аффинная система подстановок Цезаря»
11	Шифрование с использованием системы Цезаря с ключевым словом
12	Дешифрование с использованием системы Цезаря с ключевым словом
13	Шифрование, основанное на таблицах Трисемуса
14	Дешифрование, основанное на таблицах Трисемуса
16	Дешифрование с использованием биграммного шифра Плейфейра

17	Шифрование сложной заменой «Шифр Гронсфельда»
18	Дешифрование сложной заменой «Шифр Гронсфельда»
19	Шифрование с использованием двойного квадрата Уитстона
20	Дешифрование с использованием двойного квадрата Уитстона
21	Шифрование и дешифрование с использованием метода Вернама
22	Шифрование с использованием гомофонической замены
23	Дешифрование с использованием гомофонической замены
24	Шифрование двумя методами (см. варианты 1 и 7)
25	Дешифрование двумя методами (см. варианты 8 и 2)
26	Шифрование двумя методами (см. варианты 9 и 3)
27	Дешифрование двумя методами (см. варианты 4 и 10)
28	Шифрование двумя методами (см. варианты 3 и 5)
29	Дешифрование двумя методами (см. варианты 6 и 4)

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
по выполнению курсовых работ предмет «Защита информации»
для подготовки бакалавров 09.03.01 «Информатика и вычислительная техника»

Курсовая работа выполняется с целью закрепления теоретического лекционного материала и практических навыков по защите информации. На практике реализовать алгоритмы криптографических преобразований данных для обеспечения целостности, подлинности и конфиденциальности информации.

В курсовые работы входят темы: шифры перестановки, простой и сложной замены, изучаемые в 7 семестре.

Работа выполняется самостоятельно, с использованием рекомендованной литературы за счёт отведённых по программе часов и служит расширению и углублению компетенций будущего специалиста.

Перед работой следует перечитать конспект курса, выбрать и ознакомиться с литературой, как обязательной, так и дополнительной, имеющей отношение к теме работы. Возможно, будут найдены электронные теоретические материалы и аналоги программ, их также необходимо критически проанализировать и лучшее использовать в своей работе.

После осмысления темы следует описать структуры (входные и выходные), выбрать математическую модель будущей задачи, составить

алгоритм её решения. Программа должна быть отлажена, и её правильность доказана на контрольных примерах.

Параллельно студент должен описывать ход работы, отражая его в документации.

Защита курсовой работы по предмету «Защита информации» происходит в два этапа:

I. Сдача отчёта в его составе части:
   a. постановка задачи,
   b. математическое описание,
   c. алгоритм работы программы,
   d. контрольные примеры,
   e. результаты работы: листинг, распечатки входных и выходных данных.

II. Презентация работы программы с комментариями автора.

Оценка работы соотносится с требованиями рейтинговой технологии обучения.

В АлтГТУ принята **100-балльная** шкала оценок. Именно эти оценки учитываются при подсчете рейтингов, назначении стипендии и других случаях. Традиционная шкала будет использоваться только в зачетных книжках при выставлении зачёта. Соответствие оценок устанавливается следующим образом: 100 - 26 баллов – **«зачтено»**, 25 и менее баллов – **«незачтено».**

Критерии оценивания курсовых работ

95 баллов	Работа выполнена правильно. Доказана правильность её функцианирования. Документация полностью оформлена. В работе используются оптимальные методы кодирования. Работа соответствует стандартам. Студент чётко отвечает на все вопросы по выполнению и содержанию работы. За оригинальность выполнения добавляется 5 баллов.
80 баллов	Работа выполнена правильно. Доказана правильность её функцианирования. Документация полностью оформлена.В работе используются не оптимальные методы кодирования. Работа выполнена по правилам стандарта. Студент отвечает на вопросы по содержанию работы.
70 баллов	Работа выполнена в основном правильно. Документация полностью оформлена. В работе не используются оптимальные методы кодирования и защиты информации. Работа соответствует стандартам. Студент отвечает на вопросы по выполнению и содержанию работы.
50 баллов	Работа выполнена в основном правильно. Документация оформлена. В работе не используются оптимальные методы кодирования и защиты информации. Работа выполнена с некоторым отступлением от правил стандарта. Студент слабо отвечает на вопросы по выполнению и содержанию данной работы.
26 баллов	Документация оформлена. Предложенная работа выполнена, основные определения даны, используемые методы слабо соответствуют стандартам.

Примечание:

Если студент не планирует «защищать курсовую работу, но выполняет её правильно, он может сдать преподавателю её в письменном виде, аккуратно оформленную. В этом случае максимальный балл составляет **60 баллов.**

СПИСОК ЛИТЕРАТУРЫ

1. Галицкий А.В., Рябко С.Д., Шаньгин В.Ф. Защита информации в сети – анализ технологий и синтез решений М., ДМК Пресс, 2004.
2. ГОСТ 28147-89. Система обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. М., 1989.
3. ГОСТ Р 34.10-94. Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма. М., 1994.
4. ГОСТ Р 34.10-2001. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. М., 2001.
5. ГОСТ Р ИСО / МЭК 15408-1-2002. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 1. Введение и общая модель. М., 2002.
6. ГОСТ Р ИСО / МЭК 15408-2-2002. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 2. Функциональные требования безопасности. М., 2002.
7. ГОСТ ГОСТ Р ИСО / МЭК 15408-3-2002. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 3. Требования доверия к безопасности. М., 2002.
8. Гостехкомиссия России. Специальные требования и рекомендации по технической защите конфиденциальной информации (СТР-К): руководящий документ. М., 2001.
9. ИСО / МЭК 14888-1-98. Информационная технология. Методы защиты. Цифровые подписи с приложением. Часть 1. Общие положения.
10. ИСО / МЭК 14888-2-99. Информационная технология. Методы защиты. Цифровые подписи с приложением. Часть 2. Механизмы на основе подтверждения подлинности.
11. ИСО / МЭК 10118-1-94. Информационная технология. Методы защиты. Хеш – функции. Часть 1. Общие положения.
12. ИСО / МЭК 10118-2-94. Информационная технология. Методы защиты. Хеш – функции. Часть 2. Хеш – функции с использованием n – битного блочного алгоритма шифрации.
13. Лукацкий А.А. Безопасность беспроводных сетей // Технологии и средства связи. 2005. №1.
14. Монин С. Защита информации и беспроводные сети // КомпьютерПресс. 2005. №4.
15. Панасенко С.П., Батура В.П. Основы криптографии для экономистов: учеб. пособие / под ред. Г. Гагариной. М., Финансы и статистика. 2005.

16. Шаньгин В.Ф. Информационная безопасность компьютерных систем и сетей: учеб. пособие / В.Ф. Шаньгин. – М., ИД «ФОРУМ»: ИНФРА-М, 2013. -416 с.
17. Шрамко В.Н. Комбинированные системы идентификации и аутентификации // PCWeek/RE. 2004. №45.
18. ISO 17799 – Международный стандарт безопасности информационных систем. 2002.

Интернет - ресурсы

1. Базовый стандарт организации беспроводных локальных сетей IEEE 802.11 http://standards/ieee/org/reading/ieee/std/lanman/802/11-1999/pdf
2. Беляев А.В. Методы и средства защиты информации. http://www.citforum.ru/internet/nfsecure/its2000_01.shtml
3. Касперский Е. Компьютерные вирусы. http://www/kaspersky/ru/

Нина Александровна Ларина

ЗАЩИТА ИНФОРМАЦИИ. КРИПТОЛОГИЯ

Методическое пособие для бакалавров направления подготовки
09.03.01 -«Информатика и вычислительная техника»

Редактор Е.Ф. Изотова
Подписано к печати 25.12.14. Формат 60х84 1/16.
Усл. п. л. 3,5. Тираж 50 экз. Заказ 141331. Рег. № 136

Отпечатано в ИТО Рубцовского индустриального института.
658207, Рубцовск, ул. Тракторная 2/6.