12. СИММЕТРИЧНЫЕ КРИПТОСИСТЕМЫ

12.1 Понятие о симметричной криптосистеме

Большинство средств защиты информации базируется на использовании криптографических шифров и процедур шифрования - расшифрования осуществляющихся в рамках некоторой криптосистемы.
Характерной особенностью симметричной криптосистемы является применение одного и того же секретного ключа как при шифровании, так и при расшифровании сообщений.

Как открытый текст, так и шифртекст образуются из букв, входящих в конечное множество символов, называемых алфавитом. Примерами алфавитов являются конечное множество всех заглавных букв, конечное множество всех заглавных и строчных букв и цифр и т. п. В общем виде некоторый алфавит **Z** можно представить так:

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

Объединяя по определенному правилу буквы из алфавита **Z** можно создать новые алфавиты:

- алфавит 2 , содержащий m биграмм $a_0 a_0$,$a_0 a_1$,...., $a_{m-1} a_{m-1}$
- алфавит $\mathbf{X}^3$, содержащий m триграмм $a_0 a_0 a_0$, $a_0 a_0 a_{01}$,...., $a_{m-1} a_{m-1} a_{m-1}$.

В общем случае, объединяя по n букв, получаем алфавит T^n

содержащий *m n*-грамм. Например, английский алфавит

^ = {ABCDEFGH ... WXYZ}

объемом *m*=26 букв позволяет сгенерировать посредством операции конкатенации алфавит из 26 =676 биграмм:

AA, AB, … ,XZ, ZZ,

алфавит из 26 =17576 триграмм:

AAA, AAB, … ,XZZ, ZZZ и т.д.

При выполнении криптографических преобразований полезно заменить буквы алфавита целыми числами 0, 1, 2, 3,... . Это позволяет упростить выполнение необходимых алгебраических манипуляций.

Например, можно установить взаимно однозначное соответствие между русским алфавитом ^ = {АБВГД ... ЮЯ} и множеством целых Z_{32} ={0,1,2,3,...,31}; между английским алфавитом $Y_{,англ}$={*ABCDEF..YZ}* и множеством целых Z_{26} ={0,1,2,3,...,25} (см. табл. 12.1 и 12.2).

В дальнейшем будет обычно использоваться алфавит

Z_{26} ={0,1,2,3,..., 25}

содержащий *m* «букв» (в виде чисел).

Замена букв традиционного алфавита числами позволяет более четко сформулировать основные концепции и приемы криптографических преобразований. В то же время в большинстве иллюстраций будет использоваться алфавит естественного языка.

Таблица 12.1 - Соответствие между русским алфавитом и множеством целых $Z_{32} = \{0,1,2,3,...,31\}$

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	б	О	14	Ц	22	Ю	30
З	7	П	15	Ч	23	Я	31

Таблица 12.2 - Соответствие между английским алфавитом и множеством целых $Z_{26} = \{0,1,2,3,...,25\}$

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

Текст с n буквами из алфавита Z_m можно рассматривать как n-грамму

$$x = (x_0, x_1, x_2, ..., x_{n1}),$$

где x_i е Z_m $0 < i < n$, для некоторого целого $n = 1, 2, 3, \ldots$.Через Z_{mn} будем обозначать множество n-грамм, образованных из букв множества Z .

Криптографическое преобразование E представляет собой совокупность преобразований

$$E = \{E^{(n)}: 1 < n < \mathbf{oo}\}\ E$$

$$Z \quad \text{—>} \ Z^{m,n}$$
$$m,n \qquad m$$

Преобразование $E^{(n)}$ определяет, как каждая n-грамма открытого текста $\bar{x}$ **е** $Z_{m\ n}$ заменяется n-граммой шифртекста $\bar{y}$, т. е.

$$\bar{y} = E^{(n)}(\bar{x})$$

причем

$$\bar{x}, \bar{y}\ \mathbf{G}\ Z$$

при этом обязательным является требование взаимной однозначности преобразования $E^{(n)}$ на множестве $Z_{m\ n}$

Криптографическая система может трактоваться как семейство криптографических преобразований

$$E = \{E_k : K\ \hat{}\ K\},$$

помеченных параметром K, называемым ключом.

Множество значений ключа образует ключевое пространство K Далее рассматриваются 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. Следует отметить, что комбинации шифров перестановок и шифров замены образуют все многообразие применяемых на практике симметричных шифров.

12.2 Шифры перестановки

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

12.2.1 Шифрующие таблицы

С начала эпохи Возрождения (конец XIV столетия) начала возрождаться и криптография. Наряду с традиционными применениями криптографии в политике, дипломатии и военном деле появляются и другие задачи - защита интеллектуальной собственности от преследований инквизиции или заимствований злоумышленников. В разработанных шифрах перестановки того времени применяются шифрующие таблицы, которые в сущности задают правила перестановки букв в сообщении.

В качестве ключа в шифрующих таблицах используются:

1. размер таблицы;
2. слово или фраза, задающие перестановку;
3. особенности структуры таблицы.

Одним из самых примитивных табличных шифров перестановки является простая перестановка, для которой ключом служит размер таблицы. Этот метод шифрования сходен с шифром скитала. Например, сообщение

ТЕРМИНАТОР ПРИБЫВАЕТ СЕДЬМОГО В ПОЛНОЧЬ

записывается в таблицу поочередно по столбцам. Результат заполнения таблицы из 5 строк и 7 столбцов показан на рисунке 12.1.

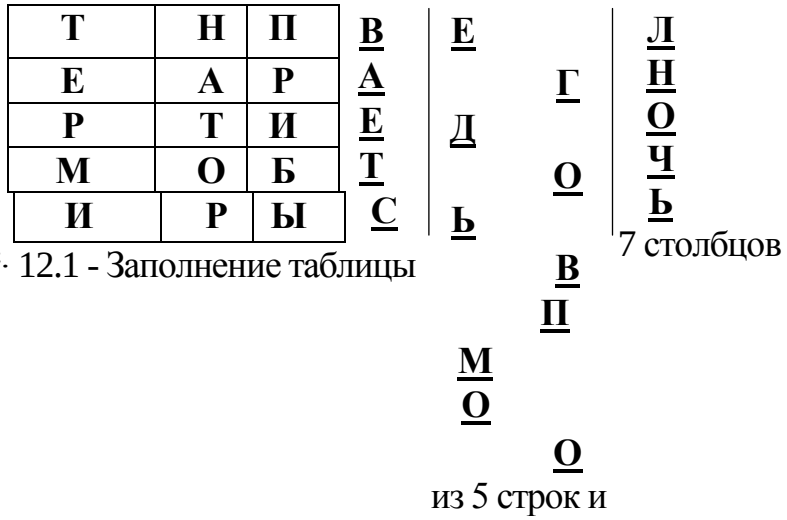


Рис. 12.1 - Заполнение таблицы

После заполнения таблицы текстом сообщения по столбцам для формирования шифртекста считывают содержимое таблицы по строкам. Если шифртекст записывать группами по пять букв, получается такое шифрованное сообщение:

ТНПВЕ ГЛЕАР АДОНР ТИЕЬВ ОМОБТ МПЧИР ЫСООЬ

Естественно, отправитель и получатель сообщения должны заранее условиться об общем ключе в виде размера таблицы. Следует заметить, что объединение букв шифртекста в 5-буквенные группы не входит в ключ шифра и осуществляется для удобства записи несмыслового текста. При расшифровании действия выполняют в обратном порядке.

Несколько большей стойкостью к раскрытию обладает метод шифрования, называемый одиночной перестановкой по ключу. Этот метод отличается от предыдущего тем, что столбцы таблицы переставляются по ключевому слову, фразе или набору чисел длиной в строку таблицы.

Применим в качестве ключа, например, слово

ПЕЛИКАН,

а текст сообщения возьмем из предыдущего примера. На рисунке 12.2 показаны две таблицы, заполненные текстом сообщения и ключевым словом, при этом левая таблица соответствует заполнению до перестановки, а правая таблица-заполнению после перестановки.

В верхней строке левой таблицы записан ключ, а номера под буквами ключа определены в соответствии с естественным порядком соответствующих букв ключа в алфавите. Если бы в ключе встретились одинаковые буквы, они бы были понумерованы слева направо. В правой таблице столбцы переставлены в соответствии с упорядоченными номерами букв ключа.

Ключ

П	Е	Л	И	К	А	Н
7	2	5	3	4	1	6
Т	Н	П	В	Е	Г	Л
Е	А	Р	А	Д	О	Н
Р	Т	И	Е	Ь	В	О
М	0	Б	Т	М	П	Ч
И	Р	Ы	С	О	О	Ь

А	Е	И	К	Л	Н	П
1	2	3	4	5	б,	7
Г	Н	В	Е	П	Л	Т
О	А	А	Д	Р	Н	Е
В	Т	Е	Ь	И	О	Р
П	О	Т	М	Б	Ч	М
О	Р	С	О	Ы	Ь	И

Рис. 12.2 – Шифрование с ключом

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

ГНВЕП ЛТООА ДРНЕВ ТЕЬИО РПОТМ БЧМОР СОЫЬИ

Для обеспечения дополнительной скрытности 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. Сначала в таблицу записывается текст сообщения, а потом поочередно переставляются столбцы, а затем строки. При расшифровании порядок перестановок должен быть обратным.

Пример выполнения шифрования методом двойной перестановки показан на рисунке 12.3.

	4	1	3	2
3	П	Р	И	Л
1	Е	Т	А	Ю
4	В.	О	С	Ь
2	М	О	Г	О

Исходная таблица

		Л	И	П			Ю			4
		Ю								М
4				В			Л	И		П
				М						В

Перестановка столбцов Перестановка строк

Рис. 12.3 - Пример выполнения шифрования методом двойной перестановки

Если считывать шифртекст из правой таблицы построчно блоками по четыре буквы, то получится следующее:

ТЮАЕ ООГМ РЛИП ОЬСВ

Ключом к шифру двойной перестановки служит последовательность номеров столбцов и номеров строк исходной таблицы (в нашем примере последовательности 4132 и 3142 соответственно).

Число вариантов двойной перестановки быстро возрастает при увеличении размера таблицы:

- для таблицы 3x3 - 36 вариантов;
- для таблицы 4x4 - 576 вариантов;
- для таблицы 5x5 - 14400 вариантов.

Однако двойная перестановка не отличается высокой стойкостью и сравнительно просто "взламывается" при любом размере таблицы шифрования.

12.2.2 Система шифрования Цезаря

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

При шифровании исходного текста каждая буква заменялась на другую букву того же алфавита по следующему правилу. Заменяющая буква определялась путем смещения по алфавиту от исходной буквы на *K* букв. При достижении конца алфавита выполнялся циклический переход к его началу. Цезарь использовал шифр замены при смещении *K*=3. Такой шифр замены можно задать таблицей подстановок, содержащей соответствующие пары букв открытого текста и шифртекста. Совокупность возможных подстановок для *K*= 3 показана на рисунке 12.4.

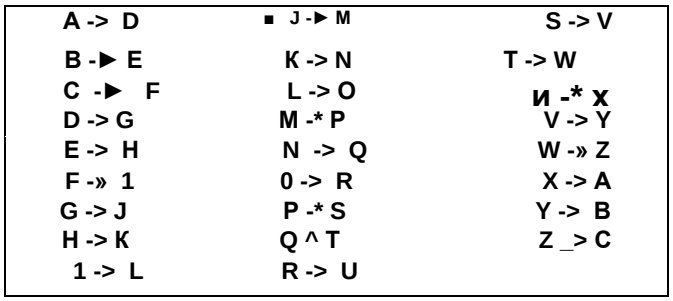


Рис. 12.4 - Совокупность возможных подстановок для *K*= 3

Например, послание Цезаря:

VENI VIDI VICI

(в переводе на русский означает "Пришел, Увидел, Победил"), направленное его другу Аминтию после победы над понтийским царем Фарнаком, сыном Митридата, выглядело бы в зашифрованном виде так:

YHQL YLGL YLFL

Достоинством системы шифрования Цезаря является простота шифрования и расшифрования.

К недостаткам системы 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 *K* изменяются только начальные позиции такой последовательности;
- число возможных ключей *K* мало;
- шифр Цезаря легко вскрывается на основе анализа частот появления букв в шифртексте.

Криптоаналитическая атака против системы одноалфавитной замены начинается с подсчета 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 на букву с наивысшей частотой появления в английском языке и т.д. Вероятность успешного вскрытия системы шифрования повышается с увеличением длины шифртекста.

12.2.3 Аффинная система подстановок Цезаря

В данном преобразовании буква, соответствующая числу *t,* заменяется на букву, соответствующую числовому значению *(at +b)* по модулю *m.*

Следует заметить, что преобразование $E_{a,b}$ *(t)* является взаимно однозначным отображением на множестве Z_m только в том случае, если наибольший общий делитель чисел a и т, 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 мы получаем следующее соответствие между числовыми кодами букв:

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:

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

Исходное сообщение **HOPE** преобразуется в шифртекст **AVYR**

Достоинством аффинной системы является удобное управление ключами - ключи шифрования и расшифрования представляются в компактной форме в виде пары чисел *(a, b)*. Недостатки аффинной системы аналогичны недостаткам системы шифрования Цезаря.

12.2.4 Система Цезаря с ключевым словом

Система шифрования Цезаря с ключевым словом является одноалфавитной системой подстановки. Особенностью этой системы является использование ключевого слова для смещения и изменения порядка символов в алфавите подстановки.

Выберем некоторое число k, $0<k<25$, и слово или короткую фразу в качестве *ключевого слова*. Желательно, чтобы все буквы ключевого слова были различными. Пусть выбраны 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
```

Теперь мы имеем подстановку для каждой буквы произвольного сообщения.

Исходное сообщение **SEND MORE MONEY**,
шифруется как **HZBY TCGZ TCBZS.**

Следует отметить, что требование о различии всех букв ключевого слова не обязательно. Можно просто записать ключевое слово (или фразу) без повторения одинаковых букв. 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
```

Несомненным 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.

12.3 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_o исходного сообщения заменяется символом y_o из алфавита B_o, 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.

Общая схема многоалфавитной подстановки для случая $r = 4$ показана на рисунке 12.5.

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

Рис. 12.5 - Схема *r*-алфавитной подстановки для случая $r = 4$

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

Многоалфавитные шифры замены предложил и ввел 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 криптологии. Кроме шифра многоалфавитной замены, Альберти также подробно описал устройства из вращающихся колес для его реализации. Криптологи всего мира почитают Л. Альберти основоположником криптологии.

12.4 Одноразовая система шифрования

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

Одноразовая система шифрует исходный открытый текст

$$X = (x_0, x_1, \ldots, x_{n1})$$ в

шифртекст

посредством подстановки Цезаря

$$Y_i = (X_i + K_i) \bmod m, \quad 0 < i < n,$$

где *Ki - i-й* элемент случайной ключевой последовательности.

Ключевое пространство *K* одноразовой системы представляет собой набор дискретных случайных величин из *Z* и содержит *m* значений. Процедура расшифрования описывается соотношением

$$X_i = (Y_i - K_i) \bmod m,$$

где *Ki - i-й* элемент той же самой случайной ключевой последовательности.

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

Этот шифр абсолютно надежен, 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 не-раскрываемыми системами, поскольку их шифртекст не содержит достаточной информации для восстановления открытого текста.

Казалось бы, что благодаря данному достоинству одноразовые системы следует применять во всех случаях, требующих абсолютной информационной безопасности. Однако возможности Применения одноразовой 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 не будет слишком обременительным для передачи действительно важных одноразовых сообщений, например, по горячей линии Вашингтон-Москва. Однако такое требование практически неосуществимо для

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

12.5 Шифрование методом гаммирования

Под ***гаммированием*** понимают процесс наложения по определенному, закону гаммы шифра на открытые данные.

Гамма шифр - *это псевдослучайная последовательность, выработанная по заданному алгоритму для зашифрования открытых данных и расшифрования зашифрованных данных.*

Процесс зашифрования заключается в генерации гаммы шифра и наложении полученной гаммы на исходный открытый текст обратимым образом, например с использованием операции сложения по модулю 2.

Следует отметить, что перед зашифрованием открытые данные разбивают на блоки $T_0^{(}$ одинаковой длины, обычно по 64 бита. Гамма шифр вырабатывается в виде последовательности блоков $Г_ш^{(i)}$ аналогичной длины.

Уравнение зашифрования можно записать в виде

$$T_ш^{(i)} = Г_ш^{(i)} \oplus T_0^{(i)}, i = 1 \ldots M,$$

где $T_ш^{(i)}$ - *i*-й блок шифртекста; $Г_ш$ - *i-й* блок гаммы шифра; $T_0^{(}$ - *i-й* блок открытого текста; *M* - количество блоков открытого текста.

Процесс расшифрования сводится к повторной генерации Гаммы шифра и наложению этой гаммы на зашифрованные данные. Уравнение расшифрования имеет вид

Получаемый этим методом шифр-текст достаточно труден для раскрытия, поскольку теперь 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 крипто-стойкость шифра определяется длиной ключа.

12.6 Стандарт шифрования данных DES

Стандарт шифрования данных DES (Data Encryption Standard) опубликован в 1977 г. Национальным бюро стандартов США. Стандарт DES предназначен для защиты от несанкционированного доступа к важной, но несекретной информации в государственных и коммерческих организациях США. Алгоритм, положенный в основу стандарта, распространялся

достаточно быстро, и уже в 1980 г. был одобрен Национальным институтом стандартов и технологий США (НИСТ). С этого момента DES превращается в стандарт не только по названию (Data Encryption Standard), но и фактически. Появляются программное обеспечение и специализированные микро-ЭВМ, предназначенные для шифрования и расшифрования информации в сетях передачи данных.

К настоящему времени DES является наиболее распространенным алгоритмом, используемым в системах защиты коммерческой информации. Более того, реализация алгоритма DES в таких системах становится признаком хорошего тона.

Основные достоинства алгоритма DES:

- *используется только один ключ длиной 56 бит;*
- *зашифровав сообщение с помощью одного пакета программ, для расшифровки можно использовать любой другой пакет программ. соответствующий стандарту DES;*
- *относительная простота алгоритма обеспечивает высокую скорость обработки;*
- *достаточно высокая стойкость алгоритма.*

Алгоритм DES использует комбинацию подстановок и перестановок. DES осуществляет шифрование 64-битовых блоков данных с помощью 64-битового ключа, в котором значащими являются 56 бит (остальные 8 бит - проверочные биты для контроля на четность). Дешифрование в DES является операцией, обратной Шифрованию, и выполняется путем повторения операций шифрования в обратной последовательности. Обобщенная схема процесса шифрования в алгоритме DES показана на рисунке 12.6 в.

Процесс шифрования заключается в начальной перестановке битов 64-битового блока, шестнадцати циклах шифрования и, наконец, в конечной перестановке битов (рисунок 15.6, в). Следует сразу отметить, что все приводимые таблицы (на рисунке 15.6, а,б) являются стандартными и должны включаться в реализацию алгоритма DES в неизменном виде. Все перестановки и коды в таблицах подобраны разработчиками таким образом, чтобы максимально затруднить процесс расшифровки путем подбора ключа.

При описании алгоритма DES (рисунок 15.6, в) применены следующие обозначения:

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

Матрица начальной перестановки IP

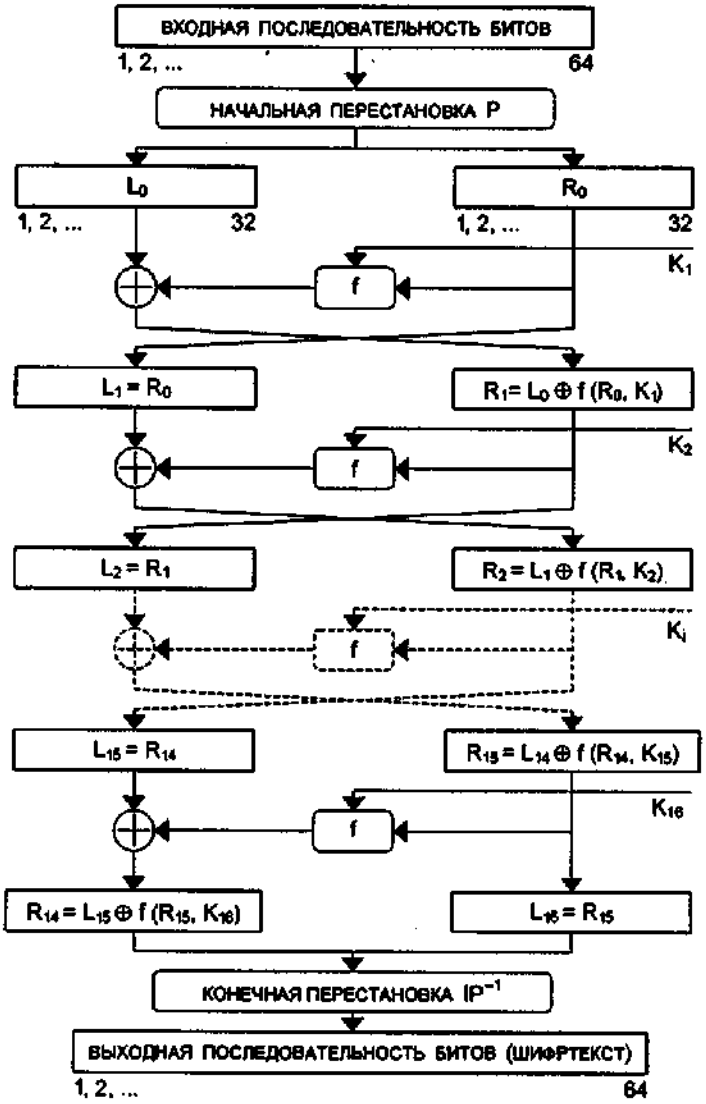
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

а.

Матрица обратной перестановки IP^{-1}

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

б.



в Рисунок 12.6 – Таблицы перестановки и алгоритм DES

Пусть из файла исходного текста считан очередной 64-битовый (8-байтовый) блок ***T***. Этот блок ***T*** преобразуется с помощью *матрицы начальной перестановки* ***IP*** (рисунок 12.6, а).

Биты входного блока ***T*** (64 бита) переставляются в соответствии с матрицей ***IP:*** бит 58 входного блока ***T*** становится битом 1, бит 50-битом 2 и т.д. Эту перестановку можно описать выражением $T_0 = IP\{T\}$. Полученная последовательность битов T_0 разделяется на две последовательности: ***L0*** - левые или старшие биты, ***RQ*** - правые или младшие биты, каждая из которых содержит 32 бита.

Затем выполняется итеративный процесс шифрования, состоящий из 16 шагов (циклов). Пусть T_i - результат *i*-й итерации:

где$L_i = t_1\ t_2 \ldots t_{32}$ (первые 32 бита);$R_i = t_{33} t_{34} \ldots t_{64}$ (последние 32 бита).

Тогда результат *i*-й итерации описывается следующими формулами:

$$L_i = R_{i\text{ч}},\ i = 1,2,...,16;\quad R_i = L_i\ 1\ \text{®}\ f(R_{i\_1}, K_i),\ i = 1,2,...,16.$$

Функция ***f*** называется функцией шифрования. Ее аргументами являются последовательность ***Ri_!***, получаемая на предыдущем шаге итерации, и 48-битовый ключ ***Ki,*** который является результатом преобразования 64-битового ключа шифра ***K***. (Подробнее функция шифрования f и алгоритм получения ключа ***Ki*** описаны ниже.) На последнем шаге итерации получают последовательности ***Ru,*** и ***Lu,*** (без перестановки местами), которые конкатенируются в 64-битовую последовательность ***RuLu***

По окончании шифрования осуществляется восстановление позиций битов с помощью матрицы обратной перестановки ***IP*** (рисунок 12.6, б).

Процесс расшифрования данных является инверсным по отношению к процессу шифрования. Все действия должны быть выполнены в обратном порядке. Это означает, что расшифровываемые данные сначала переставляются в соответствии с матрицей ***IP*** , а затем над последовательностью битов ***RieLu*** выполняются те же действия, что и в процессе шифрования, но в обратном порядке:

$$R_{i\_1} = L_i,\ i = 1,2,...,16;\quad L_{i1} = R_i \text{®} f(L_i, K_i),\ i = 1,2,...,16.$$

В настоящее время блочный алгоритм DES считается относительно безопасным алгоритмом шифрования. Он подвергался тщательному криптоанализу в течение 20 лет, и самым практичным способом его взламывания является метод перебора всех возможных вариантов ключа. Ключ DES имеет длину 56 бит, поэтому существует 2 возможных вариантов такого ключа. Если предположить, что суперкомпьютер может испытать миллион вариантов ключа за секунду, то потребуется 2285 лет для нахождения правильного ключа. Если бы ключ имел длину 128 бит, то потребовалось бы 10 лет (для сравнения: возраст Вселенной около 10 лет).