Лабораторная работа № 3

Стандарты симметричного шифрования DES и ГОСТ 28147-89

Цель работы

Изучить алгоритмы симметричного шифрования информации DES и ГОСТ 28147-89. Познакомиться с критериями оценки свойств лавинного эффекта.

Алгоритм шифрования DES

Алгоритм шифрования данных ***DES*** (*Data Encryption Standard*) относится к группе методов симметричного блочного шифрования. Он действовал в США в качестве стандарта в течение 25 лет (с 1977 по 2002 гг.) [1, 2, 4, 6].

Общая схема процесса шифрования DES показана на рисунке 1. Как и в случае любой схемы шифрования, здесь на вход функции шифрования подаётся два типа данных – открытый текст, который требуется зашифровать, и ключ. В данном случае длина открытого текста предполагается равной 64 битам, а длина ключа – 56 битам.

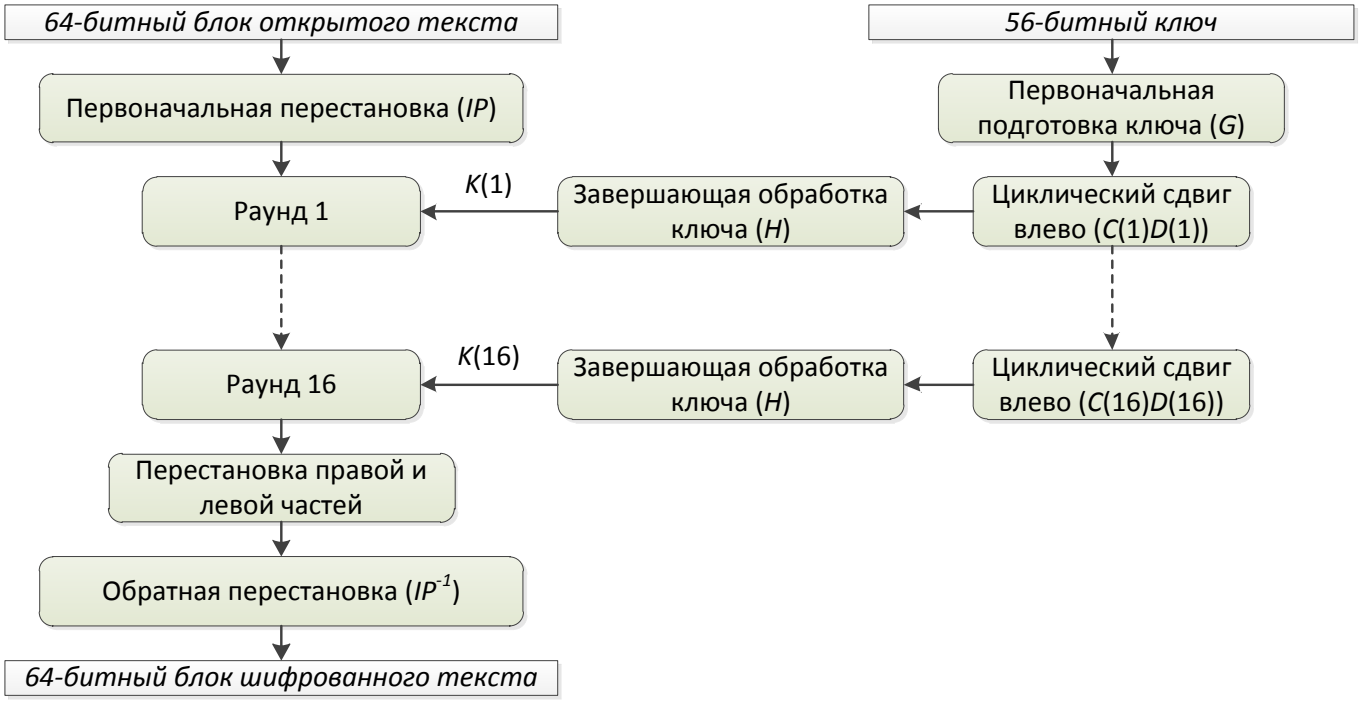

Рисунок 1 – Общая схема алгоритма шифрования DES

Из левой части рисунка 1 видно, что процесс преобразования открытого текста состоит из трёх этапов. Сначала 64-битный блок открытого текста поступает для обработки на вход начальной перестановки (IP), в результате чего получаются переставленные входные данные. Затем следует этап, состоящий из 16 раундов применения одной и той же функции, в которой используются операции перестановки и подстановки. На выходе последнего (16-го) раунда получается 64-битная последовательность, являющаяся некоторой функцией открытого текста и ключа. Левая и правая половины полученной последовательности данных меняются местами, образуя предрезультат. Наконец, этот предрезультат проходит через перестановку IP^{-1}, обратную начальной, в результате чего получается 64-битный блок шифрованного текста. Согласно рекомендациям Шеннона, в каждом раунде выполняется один шаг перемешивания (с использованием соответствующего раундового ключа и S-блоков), после которого следует шаг рассеивания, не зависящий от ключа.

В правой части рисунка 1 показано, каким образом используется 56-битный ключ. Сначала к ключу тоже применяется функция перестановки. Затем с помощью циклического сдвига влево и некоторой перестановки из полученного результата для каждого из 16 раундов генерируется под-

ключ $K(i)$. Функция перестановки одна и та же для всех раундов, но генерируемые подключи оказываются разными.

Необходимо отметить, что **все** таблицы, приведённые в данной лабораторной работе, **стандартные**, а, следовательно, должны включаться в реализацию алгоритма в неизменном виде. Все перестановки и коды в таблицах подобраны разработчиками таким образом, чтобы максимально затруднить процесс дешифрования путём подбора ключа. Структура алгоритма DES показана на рисунке 2.

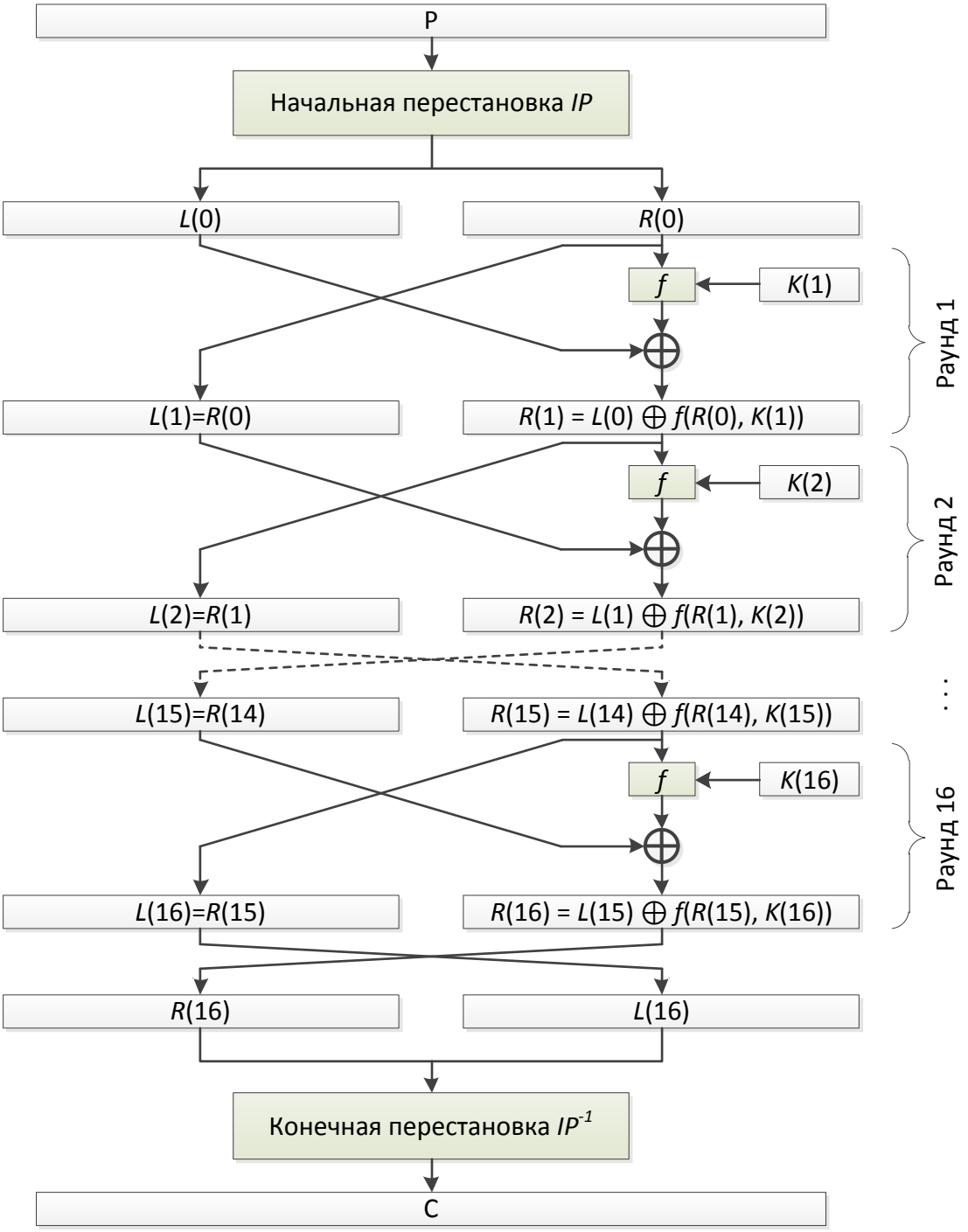

Рисунок 2 – Структура алгоритма шифрования DES

Пусть из файла считан очередной 8-байтный блок P, который преобразуется с помощью матрицы начальной перестановки IP (см. таблицу 1) следующим образом: бит 58 блока P становится битом 1, бит 50 – битом 2 и т.д., что даст в результате: $P(0) = IP(P)$. Полученная последовательность бит $P(0)$ разделяется на две последовательности по 32 бита каждая: $L(0)$ – левые или старшие биты, $R(0)$ – правые или младшие биты.

Таблица 1 – Матрица начальной перестановки IP

58	50	42	34	26	18	10	02
60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06
64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01
59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05
63	55	47	39	31	23	15	07

Затем выполняется шифрование, состоящее из 16 итераций. Результат i-й итерации описывается следующими формулами:

$$L(i) = R(i-1),$$
$$R(i) = L(i-1) \oplus f\Big(R(i-1),\ K(i)\Big).$$

Функция f называется ***функцией шифрования***. Её аргументы – это 32-битная последовательность $R(i-1)$, полученная на $(i-1)$-й итерации, и 48-битный ключ $K(i)$, который является результатом преобразования 64-битного ключа K. Подробно функция шифрования и алгоритм получения ключей $K(i)$ описаны ниже. На 16-й итерации получают последовательности $R(16)$ и $L(16)$, которые дополнительно меняются местами и конкатенируют в результирующую 64-битную последовательность $R(16)L(16)$.

Затем позиции бит этой последовательности переставляют в соответствии с матрицей IP^{-1} (см. таблицу 2).

Таблица 2 – Матрица обратной перестановки IP^{-1}

40	08	48	16	56	24	64	32
39	07	47	15	55	23	63	31
38	06	46	14	54	22	62	30
37	05	45	13	53	21	61	29
36	04	44	12	52	20	60	28
35	03	43	11	51	19	59	27
34	02	42	10	50	18	58	26
33	01	41	09	49	17	57	25

Матрицы IP^{-1} и IP соотносятся следующим образом: значение 1-го элемента матрицы IP^{-1} равно 40, а значение 40-го элемента матрицы IP равно 1, значение 2-го элемента матрицы IP^{-1} равно 8, а значение 8-го элемента матрицы IP равно 2, и т.д.

Процесс расшифрования данных является инверсным по отношению к процессу шифрования. Это означает, что расшифровываемые данные сначала переставляются в соответствии с матрицей IP, а затем над последовательностью бит $R(16)L(16)$ выполняются те же действия, что и в процессе шифрования, но в обратном порядке. Итеративный процесс расшифрования может быть описан следующими формулами:

$$R(i-1) = L(i),\ \ i = \overline{1,16},$$
$$L(i-1) = R(i) \oplus f\Big(L(i),\ K(i)\Big),\ \ i = \overline{1,16}.$$

На 16-й итерации получают последовательности $L(0)$ и $R(0)$, которые меняют местами и конкатенируют в 64-битовую последовательность $L(0)R(0)$. Затем позиции битов этой последовательности переставляют в соответствии с матрицей IP^{-1}. Результат такой перестановки – исходная 64-битовая последовательность.

Функция шифрования *f* для алгоритма DES

Теперь рассмотрим функцию шифрования $f\Big(R(i-1),\ K(i)\Big)$. Схематически она показана на рисунке 3:

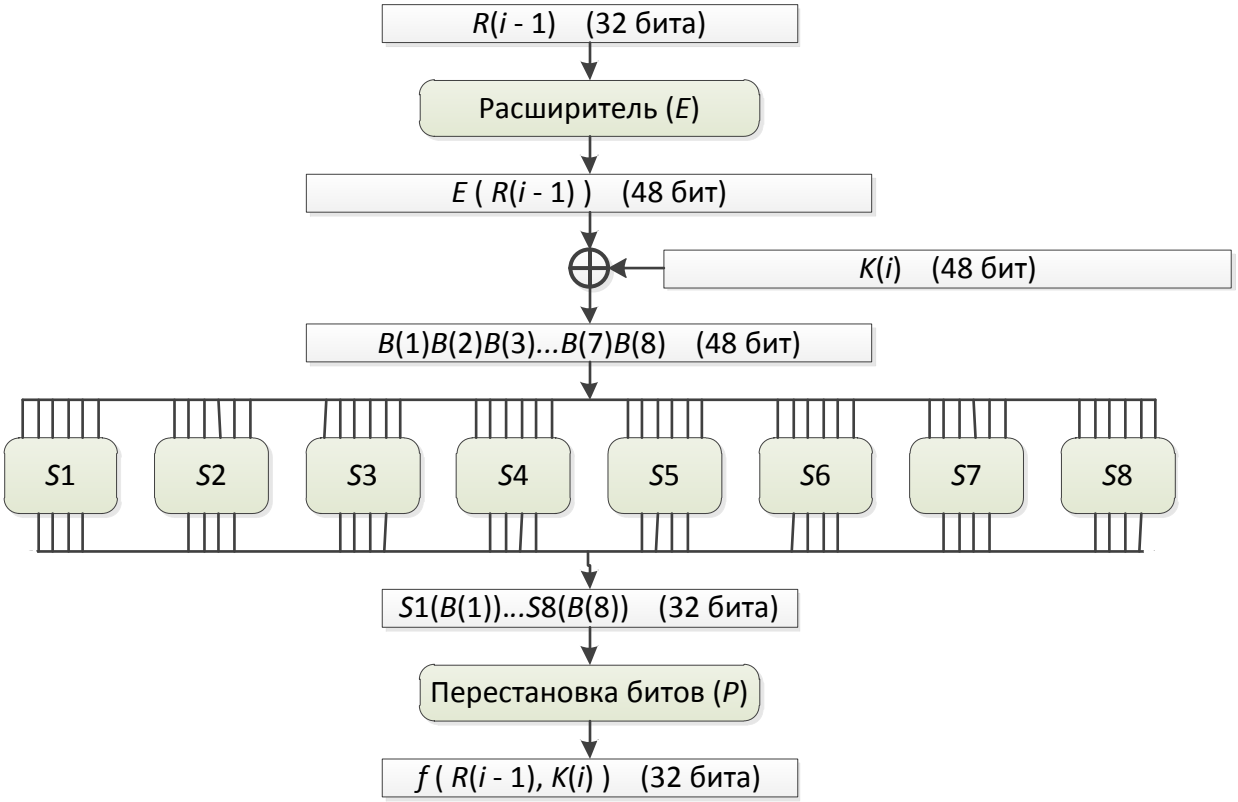

Рисунок 3 – Вычисление функции $f\left(R(i-1),\ K(i)\right)$

Для вычисления значения функции f используются следующие функции-матрицы:

- E – расширение 32-битной последовательности до 48-битной;
- $S1,\ S2,\ \dots,\ S8$ – преобразование 6-битного блока в 4-битный;
- P – перестановка бит в 32-битной последовательности.

Функция расширения E определяется таблицей 3. В соответствии с этой таблицей первые 3 бита $E\left(R(i-1)\right)$ – это биты 32, 1 и 2, а последние – 31, 32 и 1.

Таблица 3 – Функция расширения E

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	01

Результат функции $E\left(R(i-1)\right)$ есть 48-битная последовательность, которая складывается по модулю 2 (операция $\oplus$) с 48-битным ключом $K(i)$. Получается 48-битная последовательность, которая разбивается на восемь 6-битных блоков $B(1)B(2)B(3)B(4)B(5)B(6)B(7)B(8)$, т.е. $E\left(R(i-1)\right) \oplus K(i) = B(1)B(2)\dots B(8)$. Функции $S1,\ S2,\ \dots,\ S8$ определяются таблицей 4.

Таблица 4 – Функции преобразования $S1,\ S2,\ \dots,\ S8$

		Номер столбца															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Номер строки	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
																	S1
Номер строки	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
																	S2

		Номер столбца															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
	0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
	0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

К таблице 4 требуются дополнительные пояснения. Пусть на вход функции-матрицы Sj поступает 6-битный блок $B(j) = b_1b_2b_3b_4b_5b_6$, тогда двухбитное число b_1b_6 указывает номер строки матрицы, а $b_2b_3b_4b_5$ – номер столбца. Результатом $Sj\big(B(j)\big)$ будет 4-битный элемент, расположенный на пересечении указанных строки и столбца.

Например, $B(1) = 011011$. Тогда $S1\big(\mathrm{B}(1)\big)$ расположен на пересечении строки 1 и столбца 13. В столбце 13 строки 1 задано значение 5. Значит, $S1(011011) = 0101$.

Применив операцию выбора к каждому из 6-битных блоков $B(1)$, $B(2)$, … , $B(8)$, получим 32-битную последовательность $S1\big(B(1)\big)S2\big(B(2)\big)S3\big(B(3)\big)\dots S8\big(B(8)\big)$.

Наконец, для получения результата функции шифрования надо переставить биты этой последовательности. Для этого применяется функция перестановки P (см. таблицу 5). Во входной последовательности биты переставляются так, чтобы бит 16 стал битом 1, а бит 7 – битом 2, и т.д.

Таблица 5 – Функция перестановки P

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

Таким образом, $f\left(R(i-1),\ K(i)\right) = P\big(\ S1\big(B(1)\big),\ \dots\ ,\ S8\big(B(8)\big)\ \big)$.

Формирование подключей для алгоритма DES

Чтобы завершить описание алгоритма шифрования данных, осталось привести алгоритм получения 48-битных ключей $K(i)$, $i = \overline{1,16}$. На каждой итерации используется новое значение ключа $K(i)$, которое вычисляется из начального ключа K. K представляет собой 64-битный блок с восемью битами контроля чётности, расположенными в позициях 8, 16, 24, 32, 40, 48, 56, 64.

Для удаления контрольных битов и перестановки остальных используется функция G первоначальной подготовки ключа (см. таблицу 6).

Таблица 6 – Матрица G первоначальной подготовки ключа

57	49	41	33	25	17	09
01	58	50	42	34	26	18
10	02	59	51	43	35	27
19	11	03	60	52	44	36

```
63  55  47  39  31  23  15
07  62  54  46  38  30  22
14  06  61  53  45  37  29
21  13  05  28  20  12  04
```

Результат преобразования $G(K)$ разбивается на два 28-битных блока $C(0)$ и $D(0)$, причём $C(0)$ будет состоять из бит 57, 49, … , 44, 36 преобразованного ключа $G(K)$, а $D(0)$ будет состоять из бит 63, 55, … , 12, 4 преобразованного ключа $G(K)$. После определения $C(0)$ и $D(0)$ рекурсивно определяются $C(i)$ и $D(i)$, $i = \overline{1,16}$. Для этого применяют циклический сдвиг влево на один или два бита в зависимости от номера итерации, как показано в таблице 7.

Таблица 7 – Таблица сдвигов для вычисления ключа

Номер итерации	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Сдвиг (бит)	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Полученное значение вновь "перемешивается" в соответствии с матрицей H (см. таблицу 8).

Таблица 8 – Матрица H завершающей обработки ключа

```
14  17  11  24  01  05  03  28
15  06  21  10  23  19  12  04
26  08  16  07  27  20  13  02
41  52  31  37  47  55  30  40
51  45  33  48  44  49  39  56
34  53  46  42  50  36  29  32
```

Ключ $K(i)$ будет состоять из бит 14, 17, … , 29, 32 последовательности $C(i)D(i)$. Таким образом, $K(i) = H\big(C(i)D(i)\big)$.

Восстановление исходного текста осуществляется по этому же алгоритму, но вначале используется ключ $K(15)$, затем – $K(14)$, и т.д.

Алгоритм шифрования ГОСТ 28147-89

ГОСТ 28147-89 – это стандарт симметричного шифрования, принятый в 1989 году в Советском Союзе и установивший алгоритм шифрования данных, составляющих государственную тайну [5]. В начале 90-х годов алгоритм стал полностью открытым. В Российской Федерации установлен единый стандарт криптографического преобразования текста для информационных систем. Он рекомендован к использованию для защиты любых данных, представленных в виде двоичного кода, хотя не исключаются и другие методы шифрования. Данный стандарт формировался с учётом мирового опыта, и, в частности, были приняты во внимание недостатки и нереализованные возможности алгоритма DES, поэтому использование алгоритма ГОСТ предпочтительнее.

ГОСТ предусматривает три режима шифрования (простая замена, гаммирование, гаммирование с обратной связью) и один режим выработки имитовставки. Первый из режимов шифрования предназначен для шифрования ключевой информации и не может использоваться для шифрования других данных, для этого предусмотрены два других режима. Режим выработки имитовставки (криптографической контрольной комбинации) предназначен для имитозащиты шифруемых данных, т.е. для их защиты от случайных или преднамеренных несанкционированных изменений.

Алгоритм построен по тому же принципу, что и DES: это классический блочный шифр с секретным ключом. Однако он отличается от алгоритма DES большей длиной ключа, большим количеством раундов и более простой схемой построения самих раундов. В таблице 9 приведены его основные параметры, для удобства – в сравнении с параметрами алгоритма DES:

Таблица 9 – Сравнение параметров алгоритмов DES и ГОСТ

Параметры	**ГОСТ**	**DES**
Размер блока шифрования	64 бита	64 бита
Длина ключа	256 бит	56 бит
Число раундов	32	16
Узлы замен (S-блоки)	не фиксированы	фиксированы
Длина ключа для одного раунда	32 бита	48 бит

Параметры	ГОСТ	DES
Схема выработки раундового ключа	простая	сложная
Начальная и конечная перестановки бит	нет	есть

В силу намного большей длины ключа алгоритм ГОСТ гораздо устойчивей алгоритма DES к вскрытию посредством полного перебора по множеству возможных значений ключа.

В алгоритме ГОСТ блок текста, подлежащий шифрованию, сначала разбивается на левую половину L и правую половину R. На этапе i используется подключ K_i. На этапе i алгоритма ГОСТ выполняется следующее:

$$L_i = R_{i-1},$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i).$$

Также, как и в алгоритме DES, после 32 раунда выполняется перестановка левой и правой частей полученного шифротекста: $L_{32}R_{32} \rightarrow R_{32}L_{32}$.

Один такт алгоритма ГОСТ показан на рисунке 4. Один такт алгоритма осуществляет одно преобразование Фейстеля [3].

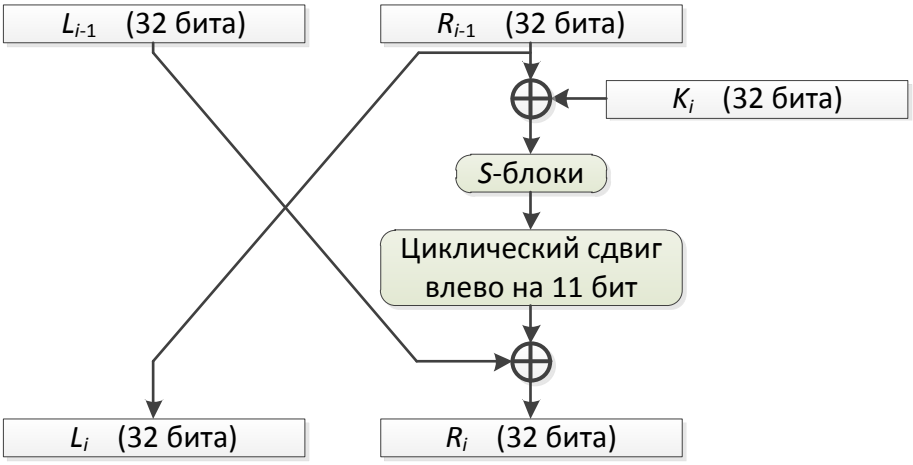

Рисунок 4 – Один такт алгоритма ГОСТ 28147-89

Описание функции f для алгоритма ГОСТ 28147-89

Сначала правый блок R_{i-1} складывается по модулю 2^{32} с подключом K_i. Полученное 32-битное сообщение делится на восемь 4-битных чисел. Каждое из этих 4-битных чисел преобразуется соответствующим S-блоком в другое 4-битное число. Поэтому любой S-блок определяется некоторой 16-битной перестановкой на множестве из 16 элементов 0, 1, … , 15. В ГОСТ использовались следующие S-блоки:

```
S1 = (  4 10  9  2 13  8  0 14  6 11  1 12  7 15  5  3),
S2 = ( 14 11  4 12  6 13 15 10  2  3  8  1  0  7  5  9),
S3 = (  5  8  1 13 10  3  4  2 14 15 12  7  6  0  9 11),
S4 = (  7 13 10  1  0  8  9 15 14  4  6 12 11  2  5  3),
S5 = (  6 12  7  1  5 15 13  8  4 10  9 14  0  3 11  2),
S6 = (  4 11 10  0  7  2  1 13  3  6  8  5  9 12 15 14),
S7 = ( 13 11  4  1  3 15  5  9  0 10 14  7  6  8  2 12),
S8 = (  1 15 13  0  5  7 10  4  9  2  3 14  6 11  8 12).
```

После преобразования S-блоками полученное 32-битное сообщение сдвигается циклически влево на 11 бит.

Формирование подключей ГОСТ 28147-89

Исходный 256-битный ключ делится на восемь 32-битных подключей K_i, $i = \overline{1,8}$. Они используются в 32 тактах в следующем порядке:

1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 8 7 6 5 4 3 2 1.

При дешифровании порядок использования подключей меняется на противоположный.

Критерии оценки свойств "лавинного эффекта"

Пусть $X^{(i)} = X \oplus E_i$, т.е. бинарный вектор, полученный инвертированием i-го бита в векторе $X \in \mathbb{U}$. Тогда бинарный вектор $Y^{(i)} \oplus Y = F(X^{(i)}, K) \oplus F(X, K)$ называется ***лавинным вектором по компоненту*** i (здесь F – функция шифрования). Пусть для рассматриваемых критериев X – это 1 блок открытого текста, размерность вектора X равна n, а Y – m и пусть $\mathbb{U}$ – это множество, состоящее из всех блоков открытого текста.

Введём следующее обозначение ***мощности множества*** A: $\#A$, т.е. $\#A$ – это количество элементов в множестве A.

Матрица зависимостей имеет следующий вид: $a_{ij} = \#\{X \mid Y_j^{(i)} \neq Y_j\}$. Эта матрица отражает зависимость j-го разряда выходного вектора от i-го разряда входного вектора.

Матрица расстояний имеет вид: $b_{ij} = \#\{Y^{(i)} \mid w(Y^{(i)} \oplus Y) = j\}$, где w – функция веса Хэмминга (число неравных нулю элементов вектора).

Существует 4 критерия, по которым предлагается проверять рассеивающие свойства блочных алгоритмов:

1. Среднее число бит выхода, изменяющихся при изменении одного бита входного вектора. Это число оценивается по формуле:

$$d_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{\sum_{j=1}^{m}(j \cdot b_{ij})}{\#\mathbb{U}}.$$

2. Степень полноты преобразования:

$$d_2 = 1 - \frac{\#\{(i, j) \mid a_{ij} = 0\}}{n \cdot m}.$$

3. Степень лавинного эффекта:

$$d_3 = 1 - \frac{\sum_{i=1}^{n}\left|\frac{1}{N}\sum_{j=1}^{m}2jb_{ij} - m\right|}{nm}, \text{ где } N = \#\mathbb{U}.$$

4. Степень соответствия строгому лавинному критерию:

$$d_4 = 1 - \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\left|\frac{2a_{ij}}{N} - 1\right|}{nm}, \text{ где } N = \#\mathbb{U}.$$

При исследовании диффузии, т.е. влияния бит исходного (открытого) текста на преобразованный (зашифрованный) текст, элементы матриц зависимостей и расстояний имеют вид:

$$a_{ij} = \#\left\{X \mid \left(f(X^{(i)}, K)\right)_j \neq \left(f(X, K)\right)_j\right\},$$
$$b_{ij} = \#\left\{X \mid w\left(f(X^{(i)}, K) \oplus f(X, K)\right) = j\right\}.$$

При исследовании конфузии, т.е. влияния бит ключа на преобразованный (зашифрованный) текст, элементы матриц зависимостей и расстояний имеют вид:

$$a_{ij} = \#\left\{X \mid \left(f(X, K^{(i)})\right)_j \neq \left(f(X, K)\right)_j\right\},$$
$$b_{ij} = \#\left\{X \mid w\left(f(X, K^{(i)}) \oplus f(X, K)\right) = j\right\}.$$

Задание

I. Реализовать приложение для шифрования, позволяющее выполнять следующие действия:
   1. Шифровать данные по заданному в варианте алгоритму:
      1) шифруемый текст должен храниться в одном файле, а ключ шифрования – в другом;
      2) зашифрованный текст должен сохраняться в файл;

3) в процессе шифрования предусмотреть возможность просмотра и изменения ключа, шифруемого и зашифрованного текстов в шестнадцатеричном и символьном виде;
4) программа должна показывать время шифрования.

2. Исследовать лавинный эффект (исследования проводить на одном блоке текста):
   1) для бита, который будет изменяться, приложение должно позволять задавать его позицию (номер) в открытом тексте или в ключе;
   2) приложение должно уметь после каждого раунда шифрования подсчитывать число бит, изменившихся в зашифрованном тексте при изменении одного бита в открытом тексте либо в ключе;
   3) приложение может строить графики зависимости числа бит, изменившихся в зашифрованном тексте, от раунда шифрования, либо графики можно строить в стороннем ПО, но тогда приложение для шифрования должно сохранять в файл необходимую для построения графиков информацию.

II. Реализовать приложение для дешифрования, позволяющее выполнять следующие действия:
1. Дешифровать данные по заданному в варианте алгоритму:
   1) зашифрованный текст должен храниться в одном файле, ключ – в другом;
   2) расшифрованный текст должен сохраняться в файл;
   3) в процессе дешифрования предусмотреть возможность просмотра и изменения ключа, зашифрованного и расшифрованного текстов в шестнадцатеричном и символьном виде.

III. Реализовать приложение, вычисляющее значения 1–4 критериев для алгоритмов DES и ГОСТ. Можно взять стороннюю реализацию того алгоритма, который не указан в варианте.

IV. С помощью реализованных приложений выполнить следующие задания:
1. Протестировать правильность работы разработанных приложений.
2. Исследовать лавинный эффект при изменении одного бита в открытом тексте и в ключе: построить графики зависимостей числа бит, изменившихся в зашифрованном сообщении, от раунда шифрования (всего должно быть построено 2 графика).
3. Сравнить значения критериев 1–4 для алгоритмов DES и ГОСТ при изменении одного бита в блоке открытого текста и одного бита в ключе при использовании одного и того же сообщения. Сообщение должно состоять хотя бы из пяти блоков (чем больше, тем точнее будут оценки критериев 1–4).
4. Сделать выводы о проделанной работе.

Дополнительные критерии оценивания качества работы

1. Наглядность приложений:
   1 – приложения позволяют просматривать и изменять ключи, шифруемый и зашифрованный тексты во всех предусмотренных заданием представлениях;
   0 – приложения позволяют просматривать ключи, шифруемый и зашифрованный тексты только в каком-то одном представлении;
   л.р. не принимается – иначе.
2. Построение графиков:
   1 – программа сама строит графики лавинного эффекта;
   0 – программа только выгружает необходимые для построения графиков данные;
   л.р. не принимается – программа не строит графики и не выгружает данные.
3. Программное оценивание свойств лавинного эффекта (по критериям 1–4):
   1 – программа вычисляет значения критериев 1–4;
   0 – программа не вычисляет значения критериев 1–4.

Варианты

Бригады с нечётным номером реализуют алгоритм DES, а бригады с чётным – ГОСТ.

Вопросы для защиты

1. Общая схема алгоритма шифрования DES.
2. Почему длина ключа для алгоритма DES равна 56 бит?
3. Как соотносятся между собой матрицы IP^{-1} и IP в алгоритме DES?
4. Как происходит шифрование функции $f\big(R(i-1), K(i)\big)$ в алгоритме DES?
5. Алгоритм получения раундовых ключей $K(i)$ в алгоритме DES.
6. В чём заключается процесс расшифрования данных в DES?
7. Что представляет собой функция $f()$ в алгоритме ГОСТ?
8. Как используется заданный ключ при шифровании в алгоритме ГОСТ?
9. Общие черты и отличия алгоритмов ГОСТ и DES.
10. Критерии оценки свойств лавинного эффекта.

Список литературы

1. Мао, В. Современная криптография: теория и практика : Пер. с англ. / В. Мао. – М. : Издательский дом "Вильямс", 2005. – 768 с.
2. Столлингс, В. Криптография и защита сетей: принципы и практика : Пер. с англ. / В. Столлингс. – 2-е изд. – М. : Издательский дом "Вильямс", 2001. – 672 с.
3. Харин, Ю.С. Математические и компьютерные основы криптологии : учебное пособие / Ю.С. Харин, В.И. Берник, Г.В. Матвеев, С.В. Агиевич. – Мн. : Новое знание, 2003. – 382 с.
4. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. – М. : Триумф, 2002. – 816 с.
5. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования ГОСТ 28147-89. – М. : Издательство стандартов, 1989. – 26 с.
6. Data Encryption Standard (DES) : FIPS Publication 46–3. – Gaithersburg : National Institute of Standards and Technology, 1999. – 22 p.