Блочный алгоритм шифрования DES (Data Encryption Standard)

1. Процедура шифрования

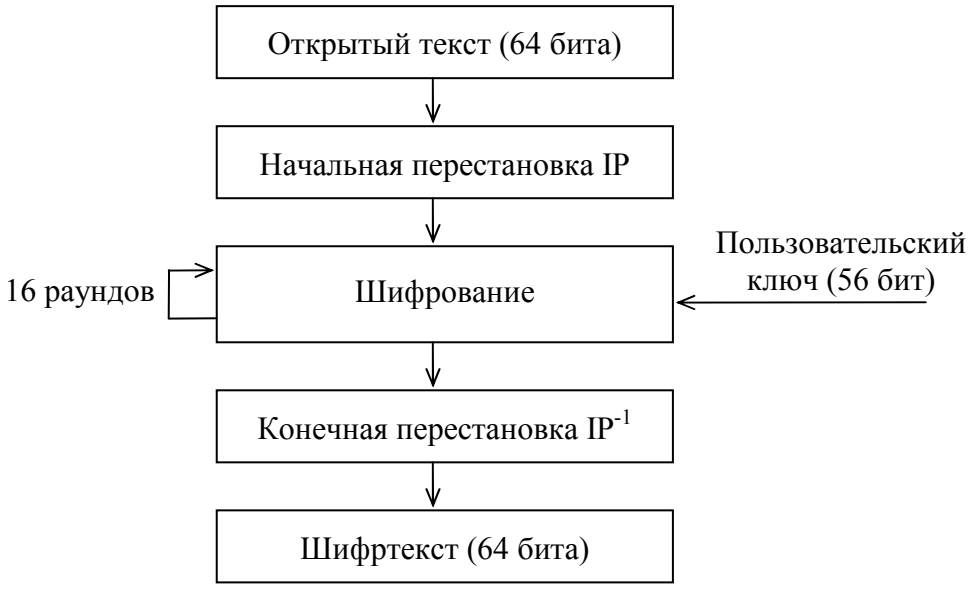

Рис.1. Общая схема процедуры шифрования в алгоритме DES

Исходный 64-битный блок открытого текста (разбит на 8 байт) представлен в таблице 1.1

Таблица 1.1

1	0	1	0	1	0	1	0	1	1	0	0	1	1	0	0
1	1	1	1	0	0	0	0	1	1	1	0	0	0	1	0
1	0	1	0	1	0	1	0	1	1	0	0	1	1	0	0
1	1	1	1	0	0	0	0	1	1	1	0	0	0	1	0

Начальная перестановка IP определяется таблицей 1.2 и необходима для предварительной рандомизации открытого текста. Следует отметить, что все перестановки и коды в этой и последующих стандартных таблицах подобраны разработчиками алгоритма DES таким образом, чтобы максимально затруднить процесс взлома шифра путем подбора ключа

Таблица 1.2

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

Каждый байт в результате перестановки формируется из 8 элементов, взятых с определенной позиции в каждом из восьми байт открытого текста.

Первый байт состоит из 2 элемента в каждом из входных байтов, считая с последнего;

Второй байт – из 4 элемента в каждом из байтов открытого текста, третий – из шестого элемента, четвертый – из 8 элемента, пятый – из 1, шестой – из 3, седьмой – из 5, восьмой – из 7 элемента.

Результат начальной перестановки IP представлен в таблице 1.3.

Таблица 1.3

1	1	1	0	1	1	1	0	0	1	0	0	0	1	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1
0	0	1	1	0	0	1	1	1	0	0	1	1	0	0	1

В соответствии с общей схемой алгоритма DES, представленной на рис.1, после начальной перестановки осуществляются 16 раундов шифрования. Подробная схема процесса шифрования представлена на рис.2

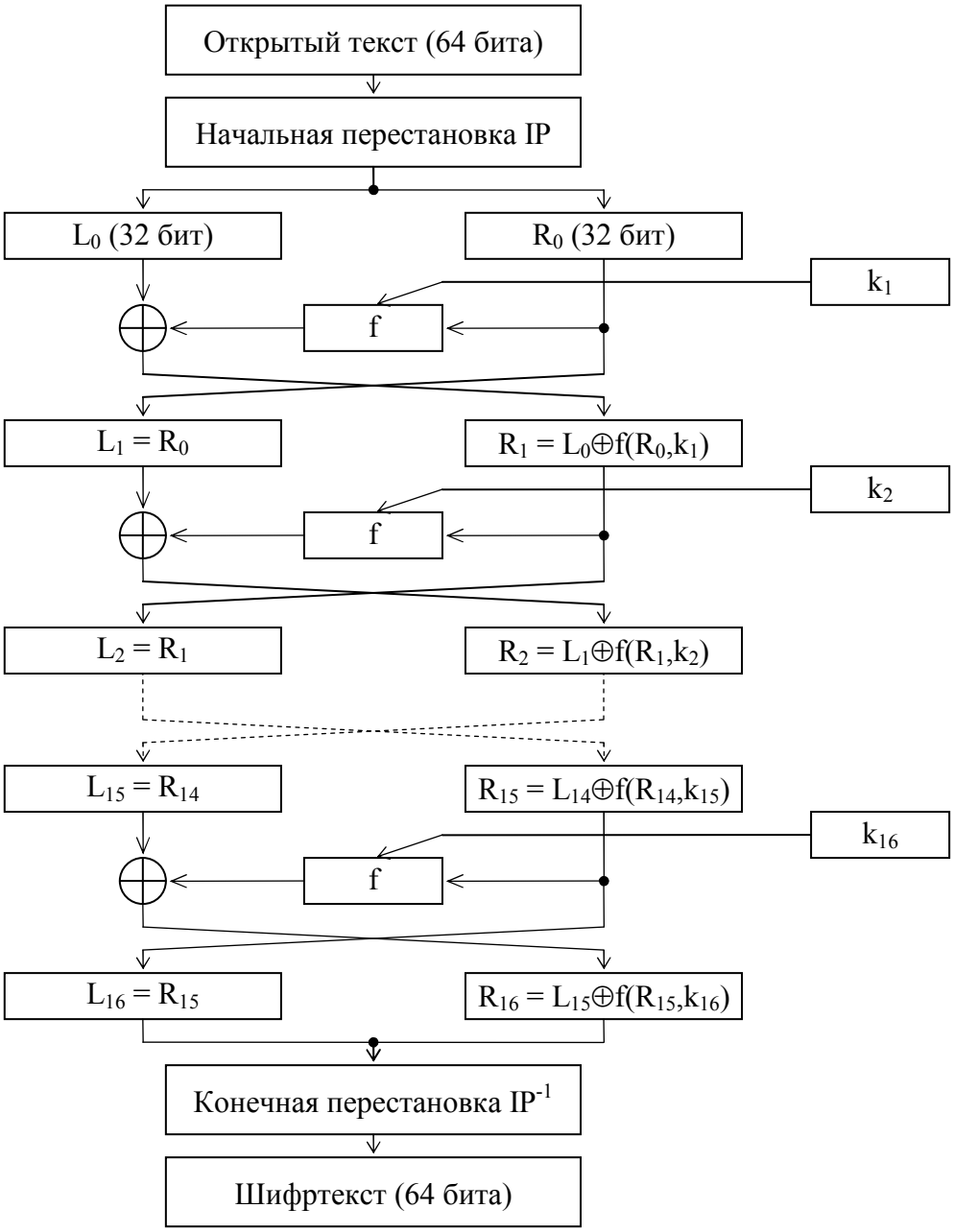

Рис.2. Схема шифрования одного 64-битного блока открытого текста, основанная на конструкции Фейстеля

64-битный блок, прошедший начальную перестановку и представленный в таблице 3 разбивается на два 32-битных подблока: L_0 (левый подблок) и R_0 (правый подблок).

L_0:

1	1	1	0	1	1	1	0	0	1	0	0	0	1	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0

R_0:

1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1
0	0	1	1	0	0	1	1	1	0	0	1	1	0	0	1

Как видно из рис.2 правый подблок R_i подвергается нелинейному преобразованию функцией шифрования f, подробная схема которой представлена на рис.3 и состоит из функции расширения E, сложению по модулю два с раундовым ключом, функцией подстановки S и перестановки P.

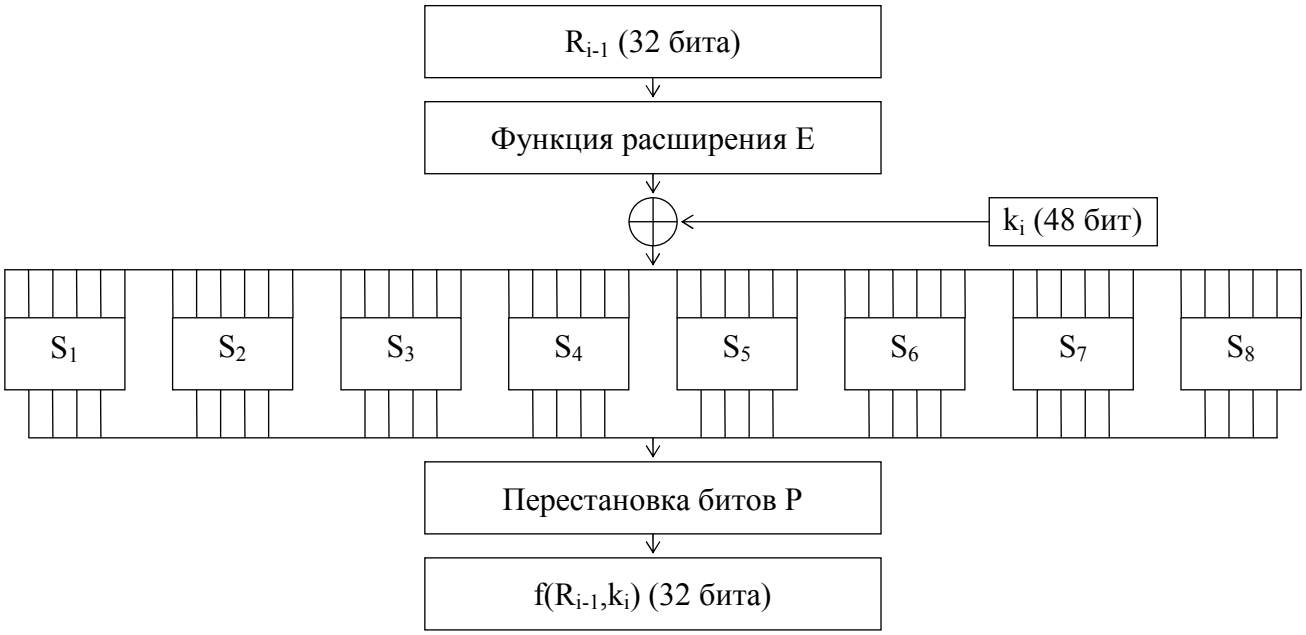

Рис.3. Схема функции шифрования $f(R_{i-1}, k_i)$.

Вначале блок R_0 подвергается операции расширения E: из 4 байт (32 бита) путем дублирования определенных битов получается 6 байт (48 бит). Порядок дублирования указан в таблице 1.4.

Таблица 1.4

32	1	2	3	4	5
4	5	6	7	8	9
8	9	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	1

Таблица намеренно имеет вид 8 (строк) × 6 (столбцов), поскольку для дальнейших преобразований нам понадобятся именно восемь 6-битных блоков.

После расширения блок R_0 имеет вид, приведенный в таблице 1.5

Таблица 1.5

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	0	1	1
1	1	1	0	1	0
1	0	0	1	1	0
1	0	0	1	1	1
1	1	0	0	1	1
1	1	0	0	1	1

Далее осуществляется сложение по модулю два (логическая операция XOR) расширенного подблока R_0 (48 бит) с раундовым ключом k_1 (также 48 бит), приведенным ниже в таблице 1.6. Генерация раундовых ключей рассмотрена ниже на стр. 8-9

Таблица 1.6

0	0	1	1	1	1
1	0	1	1	0	0
1	1	1	1	0	1
0	0	0	1	1	1
0	0	1	0	1	1
0	0	0	0	1	0
0	1	1	1	0	1
1	0	0	1	0	1

Результат сложения R_0 и k_i (48 бит) приведен в таблице 1.7 и состоит из восьми 6-битовых блоков B_i

Таблица 1.7

1	1	0	0	0	0	B_1
0	1	0	0	1	1	B_2
0	0	0	1	1	0	B_3
1	1	1	1	0	1	B_4
1	0	1	1	0	1	B_5
1	0	0	1	0	1	B_6
1	0	1	1	1	0	B_7
0	1	0	1	1	0	B_8

Первый и последний биты каждого из блоков B_i (выделены в таблице светлым) являются двоичной записью числа a, $0 \le a \le 3$. Средние 4 разряда (выделены темным) представляют число b, $0 \le b \le 15$.
Для нашего случая числа a и b в двоичном и десятичном представлении сведены в таблицу 1.8

Таблица 1.8

	a_2	a_{10}	b_2	b_{10}
B_1	10	2	1000	8
B_2	01	1	1001	9
B_3	00	0	0011	3
B_4	11	3	1110	14
B_5	11	3	0110	6
B_6	11	3	0010	2
B_7	10	2	0111	7
B_8	00	0	1011	11

Далее каждый из блоков B_i трансформируется в 4-битовый блок B'_i при помощи соответствующего S-блока (блока подстановки), которые сведены в таблицу 1.9

Таблица 1.9

		Номер столбца, определяется числом b_{10}																
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Номер строки, определяется числом a_{10}	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	S_1
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	
	2	4	1	4	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	
	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	S_2
	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	
	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	S_3
	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	S_4
	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	S_5
	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	S_6
	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	1	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	S_7
	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	S_8
	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	

Число c, соответствующее пересечению пары чисел a и b соответствующего блока B_i преобразуется в двоичный вид и является в 4-битовым блоком B'_i.
Для нашего случая все числа c в десятичном и двоичном представлении сведены в таблицу 1.10

Таблица 1.10

c_{10}	c_2	
15	1111	B'_1
0	0000	B'_2
14	1110	B'_3
2	0010	B'_4
2	0010	B'_5
3	0011	B'_6
14	1110	B'_7
14	1110	B'_8

Полученный в результате преобразования 32-битный блок $B'_1 \div B'_8$ приведен в таблице 1.11

Таблица 1.11

1	1	1	1	0	0	0	0	1	1	1	0	0	0	1	0
0	0	1	0	0	0	1	1	1	1	1	0	1	1	1	0

Окончательное значение функции шифрования $f(R_0, k_1)$ получается перестановкой P, которая определяется нижеследующей таблицей 1.12

Таблица 1.12

16	7	20	21	29	12	28	17	1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

Блок данных после перестановки будет иметь вид, представленный в таблице 1.13

Таблица 1.13

0	0	0	0	1	0	0	0	1	1	1	1	0	0	1	1
1	0	1	0	0	1	1	1	1	0	1	0	0	1	1	1

Результирующее значение функции f (32 бита), складывается по модулю два с подблоком L_0. Результат сложения приведен в таблице 1.14

Таблица 1.14

1	1	1	0	0	1	1	0	1	0	1	1	0	1	1	1
1	0	0	0	0	1	0	1	1	0	1	0	0	1	1	1

Согласно схеме алгоритма, приведенной на рис.2, 32-битный блок, приведенный в табл.1.14, на втором раунде шифрования становится правым подблоком R_1, который подвергается всем преобразованиям, аналогичным вышеприведенным.

Для сохранения целостной картины шифрования выполним последний, 16-ый раунд шифрования. Все преобразования будут аналогичны приведенным в таблицах 1.4 – 1.14
Итак, после 15 раунда шифрования имеются два 32-битных блока:
L_{15}:

1	1	1	0	0	1	1	0	1	1	0	0	1	0	0	0
0	1	0	1	0	0	1	0	1	1	1	1	1	0	0	1

R_{15}:

1	0	1	1	1	1	0	0	1	1	0	0	1	1	1	0
0	1	1	0	1	1	0	1	1	0	1	0	0	1	0	1

В соответствии со схемой функции шифрования, изображенной на рис.3, блок R_{15} подвергается вначале операции расширения E согласно таблице 1.4. Результирующий расширенный блок приведен в таблице 1.15

Таблица 1.15

1	1	0	1	1	1
1	1	1	0	0	1
0	1	1	0	0	1
0	1	1	1	0	0
0	0	1	1	0	1
0	1	1	0	1	1
1	1	0	1	0	0
0	0	1	0	1	1

Далее выполняется сложение по модулю два расширенного подблока R_{15} с раундовым ключом k_{16}, приведенным в таблице 2.9. Результат сложения приведен в таблице 1.16

Таблица 1.16

0	1	0	0	1	0	B_1
0	1	1	1	1	1	B_2
1	1	0	1	1	1	B_3
1	0	1	0	1	1	B_4
0	0	1	0	1	0	B_5
0	0	1	0	0	0	B_6
0	1	1	0	1	0	B_7
0	0	0	1	0	0	B_8

Числа a и b каждого из блоков B_i в двоичном и десятичном представлении сведены в таблицу 1.17

Таблица 1.17

	a_2	a_{10}	b_2	b_{10}
B_1	00	0	1001	9
B_2	01	1	1111	15
B_3	11	3	1011	11
B_4	11	3	0101	5
B_5	00	0	0101	5
B_6	00	0	0100	4
B_7	00	0	1101	13
B_8	00	0	0010	2

4-битовые блоки B'_i, полученные из таблицы 1.9 путем подстановки a_{10} и b_{10} в соответствующие S-блоки (таблица 1.9) сведены в таблицу 1.18

Таблица 1.18

c_{10}	c_2	
10	1010	B'_1
5	0101	B'_2
3	0011	B'_3
1	0001	B'_4
10	1010	B'_5
9	1001	B'_6
10	1010	B'_7
8	1000	B'_8

Полученный в результате S-преобразования 32-битный блок $B'_1 \div B'_8$ приведен в таблице 1.19

Таблица 1.19

1	0	1	0	0	1	0	1	0	0	1	1	0	0	0	1
1	0	1	0	1	0	0	1	1	0	1	0	1	0	0	0

Окончательное значение функции шифрования $f(R_{15}, k_{16})$ получается перестановкой P, которая определяется таблицей 1.12. Результат перестановки будет иметь вид, представленный в таблице 1.20

Таблица 1.20

1	0	0	1	1	1	0	1	1	0	0	0	0	0	0	0
0	1	1	0	0	1	1	0	1	0	0	1	0	1	0	1

Результирующее значение функции $f(R_{15}, k_{16})$, складывается по модулю два с подблоком L_{15}. Результат сложения приведен в таблице 1.21

Таблица 1.21

0	1	1	1	1	0	1	1	0	1	0	0	1	0	0	0
0	0	1	1	0	1	0	0	0	1	1	0	1	1	0	0

Согласно схеме шифрования на рис.2 блок R_{15} становится блоком L_{16}, а результат шифрования в 16 раунде, приведенный в таблице 1.21 становится блоком R_{16}.
Результирующий 64-битный блок, полученный после выполнения всех 16 раундов шифрования приведен в таблице 1.22

Таблица 1.22

1	0	1	1	1	1	0	0	1	1	0	0	1	1	1	0
0	1	1	0	1	1	0	1	1	0	1	0	0	1	0	1
0	1	1	1	1	0	1	1	0	1	0	0	1	0	0	0
0	0	1	1	0	1	0	0	0	1	1	0	1	1	0	0

Для получения шифртекста над результатом шифрования, приведенным в таблице 1.22 выполняется конечная перестановка IP^{-1}, обратная начальной перестановке IP. Перестановка осуществляется в соответствии с таблицей 1.23

Таблица 1.23

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

Результирующий блок шифртекста приведен в таблице 1.24. Также как и входной блок открытого текста, он состоит из 64 бит

Таблица 1.24

1	0	0	0	0	1	0	1	1	0	0	1	0	0	0	0
0	1	0	1	1	1	1	1	1	1	1	1	0	1	1	0
1	1	0	0	1	0	0	0	1	1	0	0	1	1	1	1
1	0	1	1	0	1	1	0	0	1	0	1	0	0	0	1

2. Генерация раундовых ключей k_i

Генерация ключей при шифровании

Секретный 64-битный пользовательский ключ приведен в таблице 2.1. Его особенность в том, что только 56 бит из 64 являются значащими. Младший разряд каждого из байтов 64-битного ключа (выделены в таблице цветом) являются проверочными битами, служат для контроля ошибок при передаче ключа и не участвуют в процессе шифрования. Таким образом, все множество секретных ключей приблизительно оценивается как 2^{56}.

Таблица 2.1

0	1	1	1	0	1	1	0	1	0	1	1	0	0	0	0
1	1	0	1	1	0	1	0	1	1	1	0	0	0	1	1
1	1	1	0	1	1	1	1	1	0	0	0	1	1	0	0
1	0	0	1	0	0	0	1	0	1	0	1	0	1	1	1

Для формирования сеансовых ключей выполняется начальная перестановка секретного ключа. В перестановке не участвуют контрольные биты, находящие на позициях 8, 16, 24, 32, 40, 48, 56, 64. Перестановка выполняется согласно нижеследующей таблице 2.2

Таблица 2.2

57	49	41	33	25	17	9	1	58	50	42	34	26	18	C_0
10	2	59	51	43	35	27	19	11	3	60	52	44	36	
63	55	47	39	31	23	15	7	62	54	46	38	30	22	D_0
14	6	61	53	45	37	29	21	13	5	28	20	12	4	

В результате перестановки появляются два блока C_0 и D_0 (начальные сеансовые ключи) по 28 бит каждый. Для нашего случая начальные сеансовые ключи C_0 и D_0 приведены в таблице 2.3

Таблица 2.3

0	1	1	1	1	1	1	0	1	0	0	1	1	1	0	1	0	0	0	1	1	0	1	1	1	1	0	0	C_0
1	0	0	1	1	1	0	1	1	0	1	1	0	0	0	1	0	0	1	1	0	1	0	0	0	1	1	1	D_0

Сеансовые ключи C_i, D_i (i = 1,2,…,16) получаются из C_{i-1}, D_{i-1} одним или двумя левыми циклическими сдвигами согласно таблице 2.4. Общее количество сдвигов равно 28, так что в результате выполнения всех раундов шифрования приводит в начальному состоянию сеансового ключа.

Таблица 2.4

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

Ключи C_1 и D_1 для нашего случая приведены в таблице 2.5

Таблица 2.5

1	1	1	1	1	1	0	1	0	0	1	1	1	0	1	0	0	0	1	1	0	1	1	1	1	0	0	0	C_1
0	0	1	1	1	0	1	1	0	1	1	0	0	0	1	0	0	1	1	0	1	0	0	0	1	1	1	1	D_1

Раундовый ключ k_i (i = 1,…,16) состоит из 48 бит, выбранных из битов вектора C_iD_i (56 бит) согласно нижеследующей таблице 2.6

Таблица 2.6

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Как видно из таблицы, при перестановке не используются биты, стоящие на позициях 9, 18, 22, 25, 35, 38, 43, 54.

Раундовый ключ k_1, полученный по приведенной выше таблице из битов вектора C_1D_1 представлен в таблице 2.7. Именно этот ключ приведен в таблице 1.6 и был использован для сложения по модулю два с подблоком R_0.

Таблица 2.7

0	0	1	1	1	1
1	0	1	1	0	0
1	1	1	1	0	1
0	0	0	1	1	1
0	0	1	0	1	1
0	0	0	0	1	0
0	1	1	1	0	1
1	0	0	1	0	1

Раундовый ключ k_{16}, используемый в последнем раунде шифрования получается аналогично, путем последовательного выполнения всех левых циклических сдвигов, приведенных в таблице 2.4. Отметим, что ключи C_{16} и D_{16} аналогичны ключам C_0 и D_0 и приведены в таблице 2.8

Таблица 2.8

0	1	1	1	1	1	1	0	1	0	0	1	1	1	0	1	0	0	0	1	1	0	1	1	1	1	0	0	C_{16}
1	0	0	1	1	1	0	1	1	0	1	1	0	0	0	1	0	0	1	1	0	1	0	0	0	1	1	1	D_{16}

Раундовый ключ k_{16} получается путем выбора 48 бит из вектора $C_{16}D_{16}$ (56 бит) согласно таблице 2.6. Ключ k_{16} приведен в табл.2.9

Таблица 2.9

1	0	0	1	0	1
1	0	0	1	1	0
1	0	1	1	1	0
1	1	0	1	1	1
0	0	0	1	1	1
0	1	0	0	1	1
1	0	1	1	1	0
0	0	1	1	1	1

Генерация ключей при дешифровании

Для корректного восстановления информации в процессе дешифрования, раундовые ключи при дешифровании должны совпадать с аналогичными раундовыми ключами при шифровании.

Таким образом, санкционированный пользователь-получатель информации, обладающий секретным 64-битным ключом, приведенным в таблице 2.1, выполняет перестановку ключа согласно таблице 2.2 для получения начальных сеансовых ключей C_0 и D_0.

Сеансовые ключи C_i, D_i ($i = 16,15,\ldots,1$) получаются из C_{i+1}, D_{i+1} правыми циклическими сдвигами согласно таблице 2.10

Таблица 2.10

Итерация i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Величина сдвига	0	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Раундовый ключ k_{16}, используемый на первом раунде дешифрования, получается путем выбора 48 бит из вектора $C_{16}D_{16}$ (56 бит) согласно таблице 2.6. Аналогичные рассуждения справедливы и для всех остальных раундовых ключей k_i.

3. Процедура дешифрования

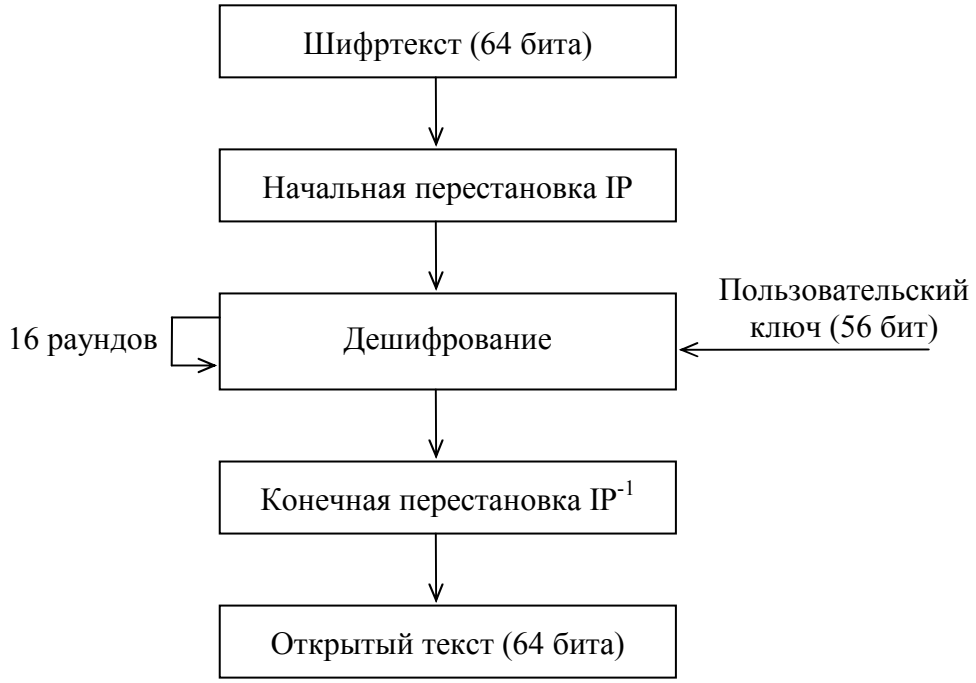

Рис.4. Общая схема процедуры дешифрования в алгоритме DES

Шифртекст (64 бита), полученный после последнего раунда шифрования и конечной перестановки IP^{-1} представлен в таблице 1.24. Для сохранения упорядоченности изложения продублируем его здесь в таблице 3.1

Таблица 3.1

1	0	0	0	0	1	0	1	1	0	0	1	0	0	0	0
0	1	0	1	1	1	1	1	1	1	1	1	0	1	1	0
1	1	0	0	1	0	0	0	1	1	0	0	1	1	1	1
1	0	1	1	0	1	1	0	0	1	0	1	0	0	0	1

Согласно схеме алгоритма дешифрования вначале необходимо выполнить начальную перестановку IP всего блока шифртекста. Начальная перестановка выполняется согласно таблице 1.2. Результаты перестановки приведены в таблице 3.2

Таблица 3.2

1	0	1	1	1	1	0	0	1	1	0	0	1	1	1	0
0	1	1	0	1	1	0	1	1	0	1	0	0	1	0	1
0	1	1	1	1	0	1	1	0	1	0	0	1	0	0	0
0	0	1	1	0	1	0	0	0	1	1	0	1	1	0	0

Отметим, что результат перестановки аналогичен блоку данных $L_{16}R_{16}$, полученному после последнего раунда шифрования и представленному в таблице 1.22.

После выполнения начальной перестановки IP блок данных, представленный в таблице 3.2 подвергается 16 раундам дешифрования. Подробная схема процедуры дешифрования приведена ниже на рис.5

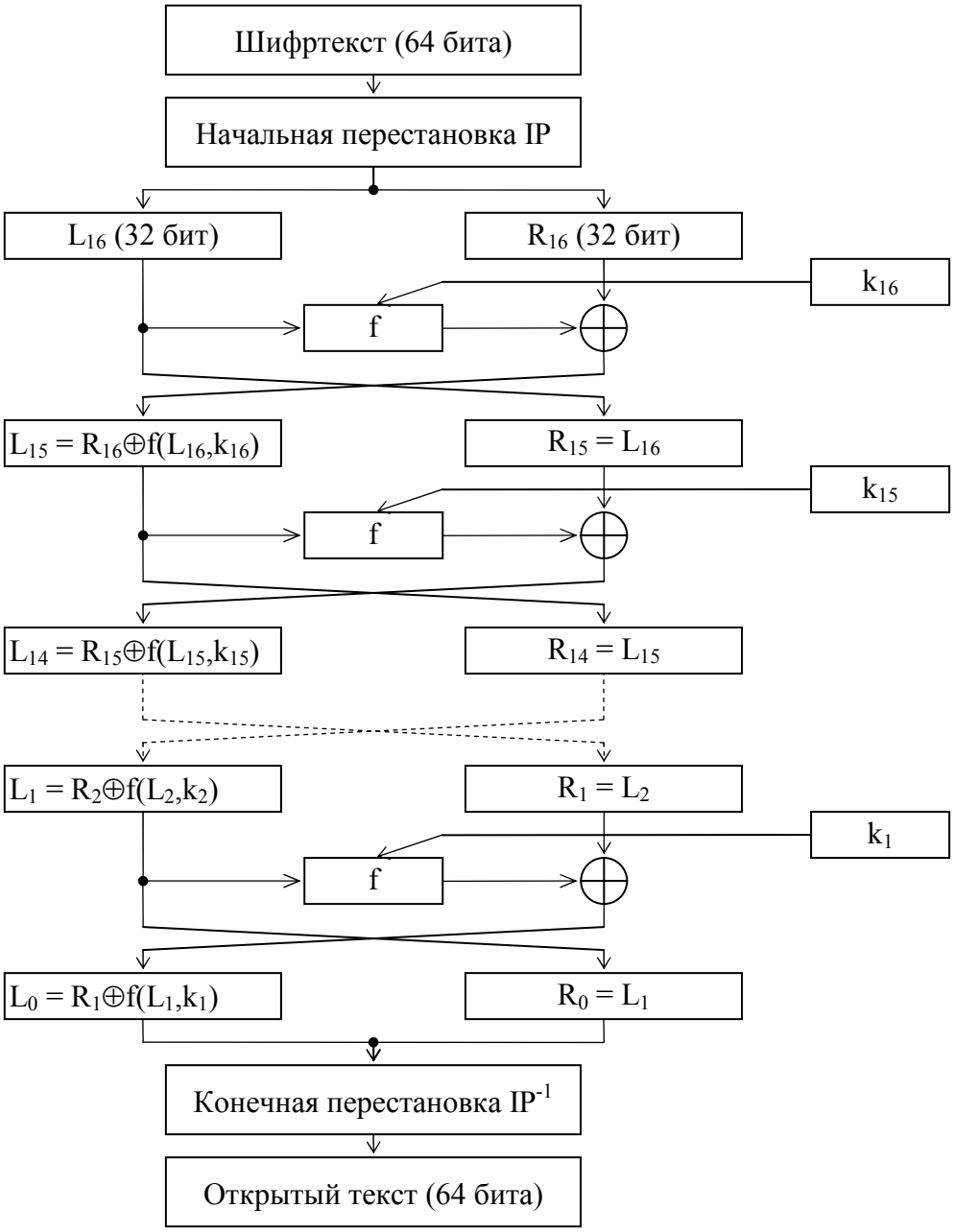

Рис.5. Схема дешифрования 64-битного блока шифртекста

В соответствии со схемой дешифрования 64-битный блок данных, представленный в таблице 3.2 разбивается на два 32-битных подблока L_{16} (левый подблок) и R_{16} (правый подблок).

L_{16}:

1	0	1	1	1	1	0	0	1	1	0	0	1	1	1	0
0	1	1	0	1	1	0	1	1	0	1	0	0	1	0	1

R_{16}:

0	1	1	1	1	0	1	1	0	1	0	0	1	0	0	0
0	0	1	1	0	1	0	0	0	1	1	0	1	1	0	0

Вначале над блоком L_{16} выполняется функция расширения в соответствии с табл.1.4. Результат расширения приведен в таблице 3.3

Таблица 3.3

1	1	0	1	1	1
1	1	1	0	0	1
0	1	1	0	0	1
0	1	1	1	0	0
0	0	1	1	0	1
0	1	1	0	1	1
1	1	0	1	0	0
0	0	1	0	1	1

Далее выполняется сложение по модулю два расширенного подблока L_{16} с раундовым ключом k_{16}, приведенным в таблице 2.9. Результат сложения приведен в таблице 3.4

Таблица 3.4

0	1	0	0	1	0	B_1
0	1	1	1	1	1	B_2
1	1	0	1	1	1	B_3
1	0	1	0	1	1	B_4
0	0	1	0	1	0	B_5
0	0	1	0	0	0	B_6
0	1	1	0	1	0	B_7
0	0	0	1	0	0	B_8

Числа a и b каждого из блоков B_i в двоичном и десятичном представлении сведены в таблицу 3.5

Таблица 3.5

	a_2	a_{10}	b_2	b_{10}
B_1	00	0	1001	9
B_2	01	1	1111	15
B_3	11	3	1011	11
B_4	11	3	0101	5
B_5	00	0	0101	5
B_6	00	0	0100	4
B_7	00	0	1101	13
B_8	00	0	0010	2

4-битовые блоки B'_i, полученные из таблицы 1.9 путем подстановки a_{10} и b_{10} в соответствующие S-блоки сведены в таблицу 3.6

Таблица 3.6

c_{10}	c_2	
10	1010	B'_1
5	0101	B'_2
3	0011	B'_3
1	0001	B'_4
10	1010	B'_5
9	1001	B'_6
10	1010	B'_7
8	1000	B'_8

Полученный в результате S-преобразования 32-битный блок $B'_1 \div B'_8$ приведен в таблице 3.7

Таблица 3.7

1	0	1	0	0	1	0	1	0	0	1	1	0	0	0	1
1	0	1	0	1	0	0	1	1	0	1	0	1	0	0	0

Окончательное значение функции дешифрования $f(L_{16}, k_{16})$ получается перестановкой P, которая определяется таблицей 1.12. Результат перестановки будет иметь вид, представленный в таблице 3.8

Таблица 3.8

1	0	0	1	1	1	0	1	1	0	0	0	0	0	0	0
0	1	1	0	0	1	1	0	1	0	0	1	0	1	0	1

Результирующее значение функции $f(L_{16}, k_{16})$, складывается по модулю два с подблоком R_{16}. Результат сложения приведен в таблице 3.9

Таблица 3.9

0	1	1	1	1	0	1	1	0	1	0	0	1	0	0	0
0	0	1	1	0	1	0	0	0	1	1	0	1	1	0	0

Согласно схеме дешифрования на рис.5 блок L_{16} становится блоком R_{15}, а результат дешифрования в 1 раунде, приведенный в таблице 3.9 становится блоком L_{15}.

Аналогично выполняется дешифрование на всех остальных раундах. Для составления полной картины "восстановления" исходных данных выполним также последний раунд дешифрования и конечную перестановку IP^{-1} для получения открытого текста.

Блоки L_1 и R_1, полученные после 15-го раунда дешифрования приведены ниже:

L_1:

1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1
0	0	1	1	0	0	1	1	1	0	0	1	1	0	0	1

R_1:

1	1	1	0	0	1	1	0	1	0	1	1	0	1	1	1
1	0	0	0	0	1	0	1	1	0	1	0	0	1	1	1

Вначале над блоком L_1 выполняется функция расширения в соответствии с табл.1.4. Результат расширения приведен в таблице 3.10

Таблица 3.10

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	0	1	1
1	1	1	0	1	0
1	0	0	1	1	0
1	0	0	1	1	1
1	1	0	0	1	1
1	1	0	0	1	1

Далее выполняется сложение по модулю два расширенного подблока L_1 с раундовым ключом k_1, приведенным в таблице 2.8. Результат сложения приведен в таблице 3.11

Таблица 3.11

1	1	0	0	0	0	B_1
0	1	0	0	1	1	B_2
0	0	0	1	1	0	B_3
1	1	1	1	0	1	B_4
1	0	1	1	0	1	B_5
1	0	0	1	0	1	B_6
1	0	1	1	1	0	B_7
0	1	0	1	1	0	B_8

Числа a и b каждого из блоков B_i в двоичном и десятичном представлении сведены в таблицу 3.12

Таблица 3.12

	a_2	a_{10}	b_2	b_{10}
B_1	10	2	1000	8
B_2	01	1	1001	9
B_3	00	0	0011	3
B_4	11	3	1110	14
B_5	11	3	0110	6
B_6	11	3	0010	2
B_7	10	2	0111	7
B_8	00	0	1011	11

4-битовые блоки B'_i, полученные из таблицы 1.9 путем подстановки a_{10} и b_{10} в соответствующие S-блоки сведены в таблицу 3.13

Таблица 3.13

c_{10}	c_2	
15	1111	B'_1
0	0000	B'_2
14	1110	B'_3
2	0010	B'_4
2	0010	B'_5
3	0011	B'_6
14	1110	B'_7
14	1110	B'_8

Полученный в результате S-преобразования 32-битный блок $B'_1 \div B'_8$ приведен в таблице 3.14

Таблица 3.14

1	1	1	1	0	0	0	0	1	1	1	0	0	0	1	0
0	0	1	0	0	0	1	1	1	1	1	0	1	1	1	0

Окончательное значение функции дешифрования $f(L_1, k_1)$ получается перестановкой P, которая определяется таблицей 1.12. Результат перестановки будет иметь вид, представленный в таблице 3.15

Таблица 3.15

0	0	0	0	1	0	0	0	1	1	1	1	0	0	1	1
1	0	1	0	0	1	1	1	1	0	1	0	0	1	1	1

Результирующее значение функции $f(L_1, k_1)$, складывается по модулю два с подблоком R_1. Результат сложения приведен в таблице 3.16

Таблица 3.16

1	1	1	0	0	1	1	0	1	0	1	1	0	1	1	1
1	0	0	0	0	1	0	1	1	0	1	0	0	1	1	1

Согласно схеме дешифрования на рис.5 блок L_1 становится блоком R_0, а результат дешифрования в 16 раунде, приведенный в таблице 3.9 становится блоком L_0.

L_0:

1	1	1	0	0	1	1	0	1	0	1	1	0	1	1	1
1	0	0	0	0	1	0	1	1	0	1	0	0	1	1	1

R_0:

1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1
0	0	1	1	0	0	1	1	1	0	0	1	1	0	0	1

Выполним окончательную перестановку IP^{-1} над полученным в результате всех раундов дешифрования 64-битным блоком данных L_0R_0. Перестановка выполняется в соответствии с таблицей 1.23 и ее результат приведен в таблице 3.17

Таблица 3.17

1	0	1	0	1	0	1	0	1	1	0	0	1	1	0	0
1	1	1	1	0	0	0	0	1	1	1	0	0	0	1	0
1	0	1	0	1	0	1	0	1	1	0	0	1	1	0	0
1	1	1	1	0	0	0	0	1	1	1	0	0	0	1	0

Результат дешифрования совпадает с блоком открытого текста, приведенного в таблице 1.1.