Лабораторная работа №2. Криптографические системы с секретным ключом (4 часа)

2.1. ЦЕЛЬ РАБОТЫ – изучение криптографических алгоритмов с секретным ключом.

2.2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Криптографические системы с секретным ключом или симметричные криптосистемы используются человеком очень давно. В качестве примера приведем криптографическую систему Цезаря, названную в честь римского императора Гая Юлия Цезаря (100 г. до н.э. – 44 г. до н.э.), использовавшего ее для своей секретной переписки.

Отличительной особенностью симметричных криптосистем является то, что и для шифрования и дешифрования данных используется один и тот же секретный ключ.

2.2.1. Алгоритм S-DES

Упрощенный DES (S-DES или Simplified DES) был разработан профессором Эдвардом Шейфером (Edward Schaefer) из университета Санта-Клары (Santa Clara University). Данный алгоритм используется для изучения структуры алгоритма DES при выполнении операций шифрования и дешифрования с использованием блочных шифров и ключей с небольшим количеством битов.

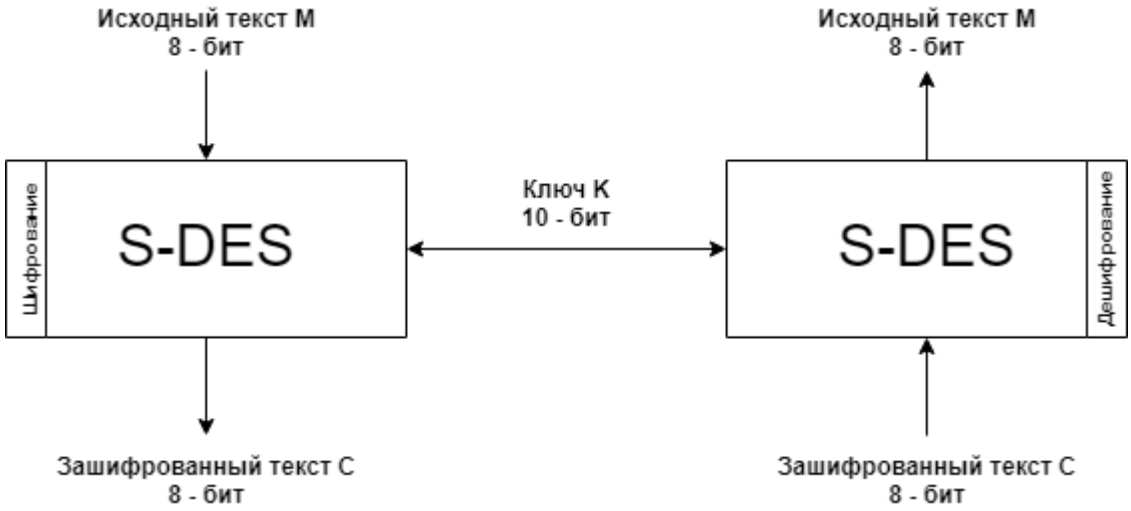

Рисунок 1. Общая схема работы алгоритма S-DES

При шифровании данных на вход данного алгоритма поступает 8-битовый блок исходного текста M (например, 1011 0110) и 10-битовый ключ K (например, 10010 10011). В результате работы на выходе вырабатывается 8-битовый блок зашифрованного текста C. При дешифровании на вход

подается 8-битовый блок зашифрованного текста C и ранее используемый для шифрования 10-битовый ключ K. В результате работы на выходе вырабатывается 8-битовый блок исходного текста C.

2.2.1.1. Генерация ключей

Размер входной последовательности *K* – 10 бит. В результате работы формируются два ключа *K1* и *K2*, значения которых используются в операциях шифрования и дешифрования.

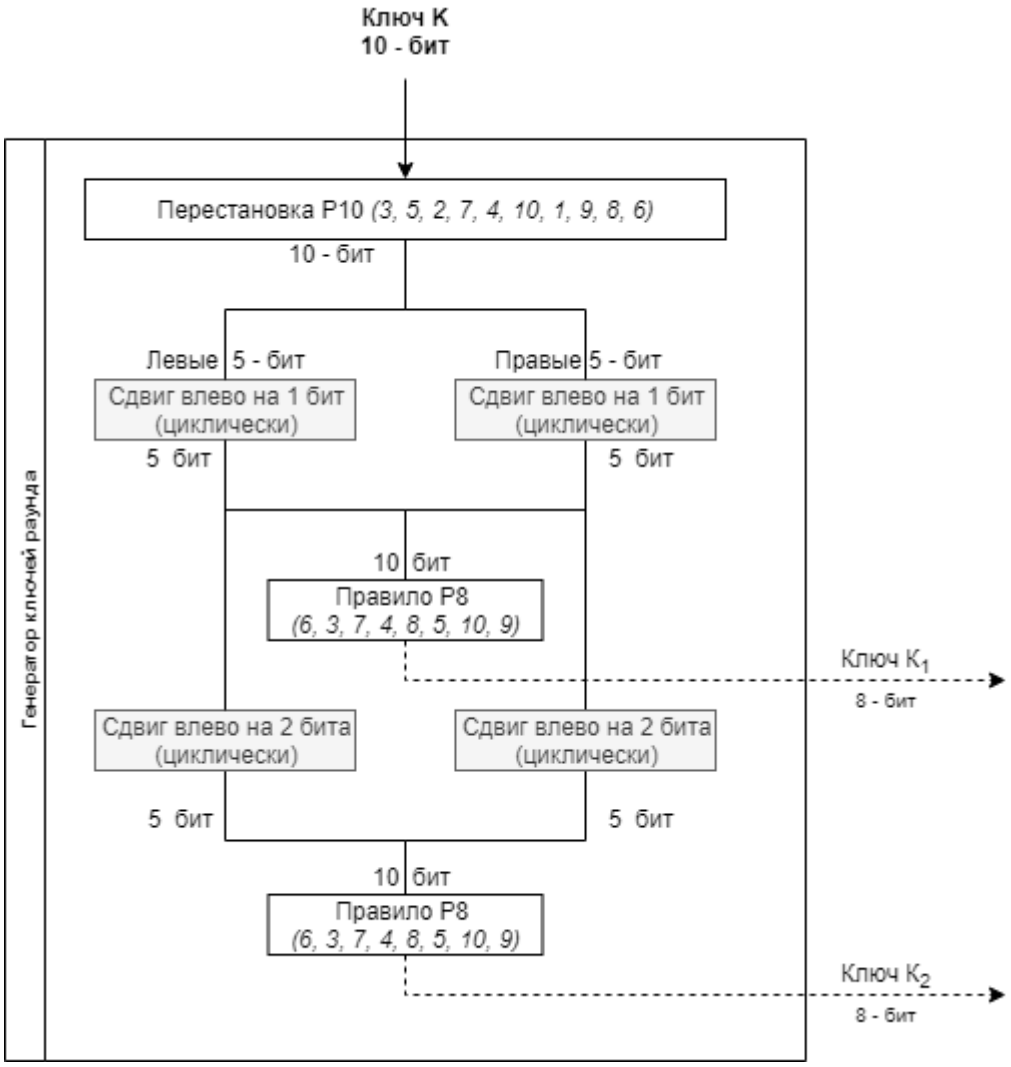

Рисунок 2. Схема генерации ключей

Приведем пример работы:

Ключ *K*: *1001010011*

1. Переставляем биты ключа *K* согласно правилу перестановки *P10 (см. рисунок 3). P10 = (3, 5, 2, 7, 4, 10, 1, 9, 8, 6): 00001 11101*.
2. Сдвигаем циклически на один бит влево левую *L* (5 бит) и правую *R* (5 бит) части последовательности отдельно друг от друга: *00010 11011*.

3. Выбираем восемь значений, используя правило *P8=(6, 3, 7, 4, 8, 5, 10, 9)*, и формируем ключ *K1*: *1011 0011*.
4. Берем результат, полученный на шаге 2, и сдвигаем циклически на два бита влево левую *L* (5 бит) и правую *R* (5 бит) части последовательности отдельно друг от друга: *01000 01111*.
5. Выбираем восемь значений, используя правило *P8=(6, 3, 7, 4, 8, 5, 10, 9)*, и формируем ключ *K2*: *00101011*.

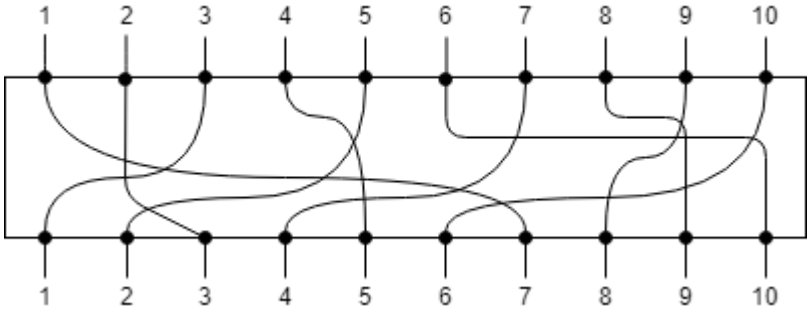

Рисунок 3. Перестановка 10 бит последовательности по правилу *P10 = (3, 5, 2, 7, 4, 10, 1, 9, 8, 6)*

2.2.1.2. Шифрование

При выполнении данной операции исходный текст *M* размером 8 бит преобразуется в зашифрованный текст *C* размером 8 бит. Перед выполнением операции шифрования ключи *K1* и *K2* должны быть сгенерированы (см. раздел 2.2.1.1).

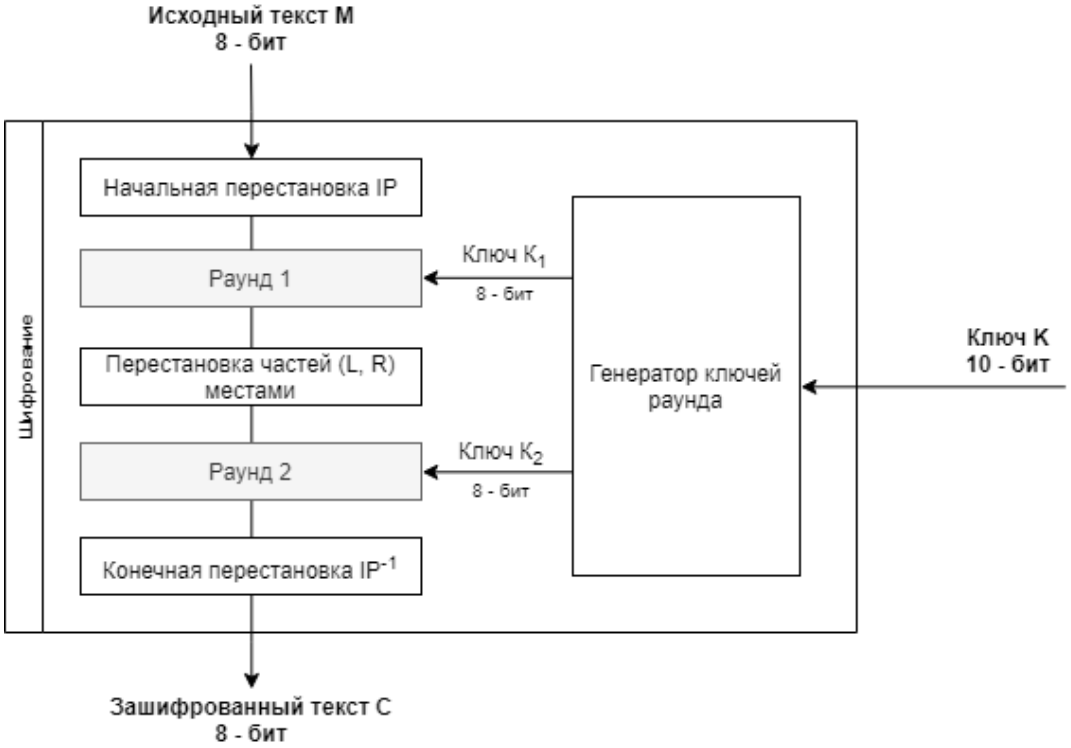

Рисунок 4. Алгоритм S-DES. Шифрование

Исходный текст *M: 1011 0110*

Ключ *K1: 1011 0011*
Ключ *K2: 0010 1011*

Приведем обобщенный алгоритм шифрования исходного текста *M*:

1. Переставляем биты входного текста *M* согласно правилу начальной перестановки *IP=(2, 6, 3, 1, 4, 8, 5, 7)*: *0111 1001*.
2. Выполняем *Раунд* шифрования, применяя ключ *K1*. На вход подаем последовательность, полученную в п. 1. Получаем «перемешанную» последовательность размером 8 бит: *0111 1001*.
3. Выполняем перестановку местами левой *L* и правой *R* частей последовательности (левая часть становится на место правой, а правая – на место левой): *1001 0111*.
4. Выполняем *Раунд* шифрования, применяя ключ *K2*. На вход подаем последовательность, полученную в п. 3. Получаем «перемешанную» последовательность размером 8 бит: *0100 0111*.
5. Переставляем биты последовательности согласно правилу конечной перестановки $IP^{-1}=(4, 1, 3, 5, 7, 2, 8, 6)$. Получаем зашифрованный текст *C: 00001111* размером 8 бит, который является результатом шифрования исходного текста *M*.

2.2.1.3. Дешифрование

При выполнении данной операции входной зашифрованный текст *C* размером 8 бит преобразуется в исходный текст *M* размером 8 бит. Перед выполнением операции дешифрования ключи *K1* и *K2* должны быть сгенерированы.

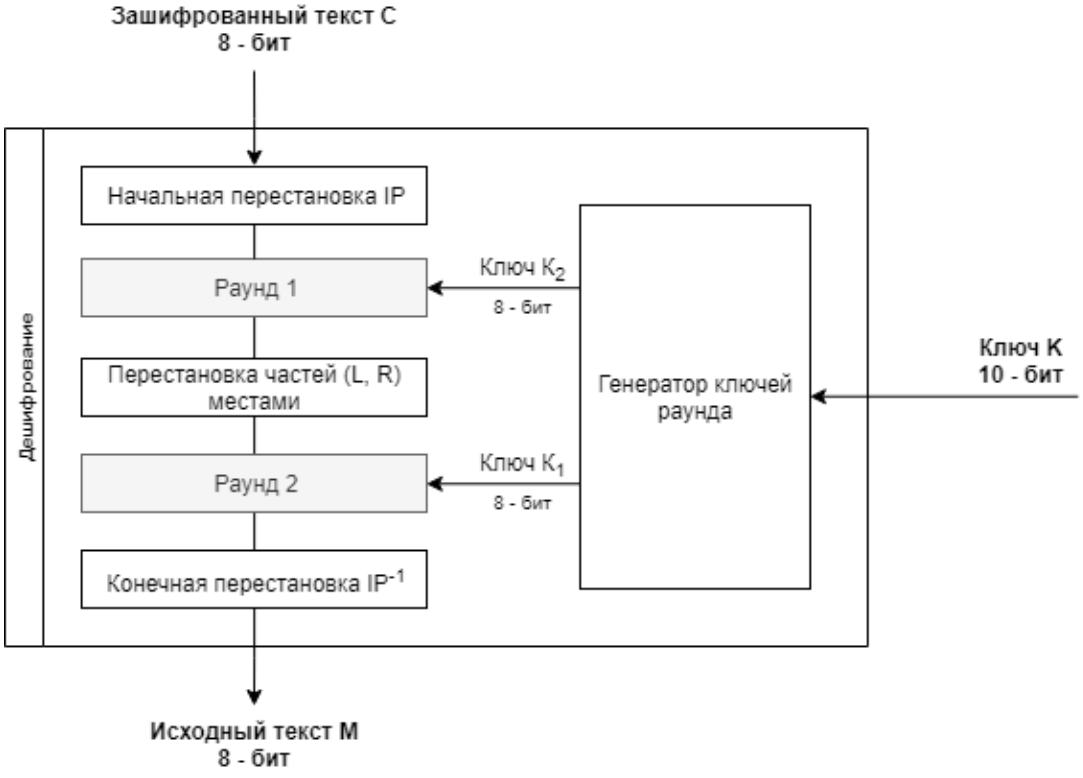

Рисунок 5. Алгоритм S-DES. Дешифрование

Зашифрованный текст *C: 0000 1111*
Ключ *K1: 1011 0011*
Ключ *K2: 0010 1011*

Приведем обобщенный алгоритм дешифрования:

1. Переставляем биты входного текста *C* согласно правилу начальной перестановки *IP=(2, 6, 3, 1, 4, 8, 5, 7)*: *01000111*.
2. Выполняем *Раунд* дешифрования, применяя ключ *K2*. На вход подаем последовательность, полученную в п. 1. Получаем «перемешанную» последовательность размером 8 бит: *1001 0111*.
3. Выполняем перестановку местами левой *L* и правой *R* частей последовательности (левая часть становится на место правой, а правая – на место левой): *01111001*.
4. Выполняем *Раунд* дешифрования, применяя ключ *K1*. На вход подаем последовательность, полученную в п. 3. Получаем «перемешанную» последовательность размером 8 бит: *01111001*.
5. Переставляем биты последовательности согласно правилу конечной перестановки $IP^{-1}=(4, 1, 3, 5, 7, 2, 8, 6)$. Получаем исходный текст *M: 10110110* размером 8 бит, который является результатом дешифрования входного зашифрованного текста *C*.

2.2.1.4 .Последовательность операций *Раунд*

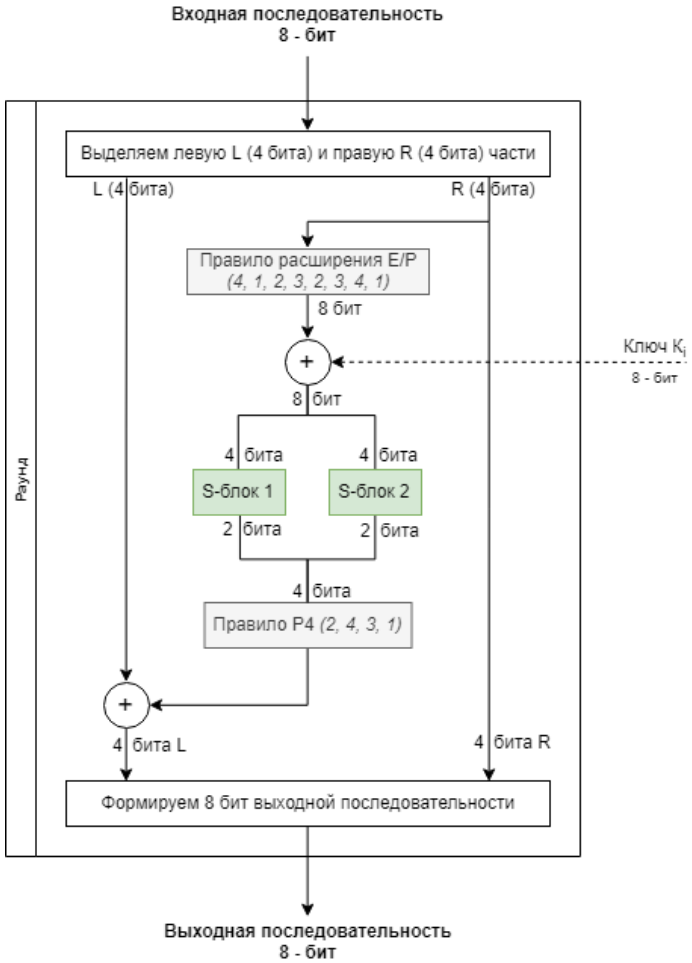

Рисунок 6. Схема выполнения последовательности операций Раунд

Раунд представляет собой последовательность операций, применяемых ко входной последовательности размером 8 бит. В результате выполнения операций, входящих в состав *Раунда*, выполняется «перемешивание» входной последовательности с использованием ключа K_i, передаваемого на вход *Раунда*, и формирование 8 бит выходной последовательности.

Приведем обобщенную последовательность операций, выполняемых в рамках *Раунда*. На вход подается последовательность *01000111* и ключ K_i *00101011*:

1. Входную последовательность *01000111* разбиваем на 2 части: левую *L* (*0100*) и правую *R (0111)*.
2. Правую часть *R* расширяем до 8 бит, используя правило расширения *E/P=(4, 1, 2, 3, 2, 3, 4, 1)*: *10111110*.
3. Выполняем операцию XOR (исключающее ИЛИ) между результатом, полученным в п. 2, и поданным ключом K_i :

   1011 1110 (*результат из п. 2)*
   0010 1011 (K_i)
   1001 0101

4. Обрабатываем S-блоками (S-box) полученный результат. **Левые** 4 бита полученного результата (*1001*) обрабатываются **S-блоком 1**, а **правые** 4 бита полученного результата (*0101*) обрабатываются **S-блоком 2** (см. рисунок 7).

	0	1	2	3
0	1	0	3	2
1	3	2	1	0
2	0	2	1	3
3	3	1	3	2

S-блок 1

	0	1	2	3
0	0	1	2	3
1	2	0	1	3
2	3	0	1	0
3	2	1	0	3

S-блок 2

Рисунок 7. S-блоки (S-boxes)

Используя 4 бита левой *L* или правой *R* части, определяем ячейку, значение из которой необходимо взять в качестве результата. Крайние биты из 4-х (***1***001***1***) определяют номер строки, а средние биты из 4-х (1***00***1) определяют номер столбца ячейки в S-блоке, значение которой необходимо взять в качестве результата обработки этих 4-х бит.

Для левой *L* части (*1001*) входной последовательности:

1001***1*** *->11 (бинарный) ->3(десятичный) ->3 строка*
1***00***1 *->00 (бинарный) ->0 (десятичный) ->0 столбец*

В **S-блоке 1** определяем ячейку в строке 3 и столбце 0 и получаем значение *3*. Преобразуем данное значение в двоичное представление: *11*.

Для правой *R* части (*0101*) входной последовательности:

***0**10**1** -> 01 (бинарный) ->1 (десятичный) ->1 строка*
*0**10**1 ->10 (бинарный) ->2 (десятичный) ->2 столбец*

В **S-блоке 2** определяем ячейку в строке 1 и столбце 2 и получаем значение *1*. Преобразуем данное значение в двоичное представление: *01*.

В результате обработки входных 8 бит получается значение из 4-х бит, где левые 2 бита – результат, полученный из **S-блока 1**, а правые 2 бита – результат, полученный из **S-блока 2**:

11 01

5. Переставляем биты последовательности из п. 4 *(1101)* согласно правилу перестановки *P4=(2, 4, 3, 1)*: *1101*.
6. Выполняем операцию XOR (исключающее ИЛИ) между результатом, полученным в п. 5, и значением левой части *L* из п. 1:

1101	*(результат из п. 5)*
0100	*(L из п. 1)*
1001	

7. Формируем 8 бит результата работы *Раунда*. В значение левой части подставляем 4 бита из п. 6 (1001), а в значение правой части – 4 бита правой части *R* входной последовательности из п. 1 *(0111)*:

10010111

2.3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить теоретический материал по лабораторной работе.
2. Выполнить задание согласно своему варианту (см. табл. 2.1).
3. Реализованное программное средство должно зашифровывать и расшифровывать произвольный текстовый файл (*.txt) размером более 1 Kb.

Таблица 2.1

Вариант	Задание
№ 1	Реализовать шифратор и дешифратор алгоритма S-DES.