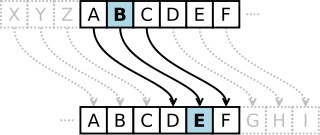
**5 Лекция. Основы криптографии. Криптография с симметричными ключами.**

**Шифры подстановки**

**Шифр Цезаря**

[http://ru.wikipedia.org/wiki/Шифр\_Цезаря](http://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80_%D0%A6%D0%B5%D0%B7%D0%B0%D1%80%D1%8F)



Используется подстановка (замена) букв из измененного алфавита. В классическом варианте алфавит получают с помощью сдвига букв на три позиции.

С точки зрения современной парадигмы, можно выделить:

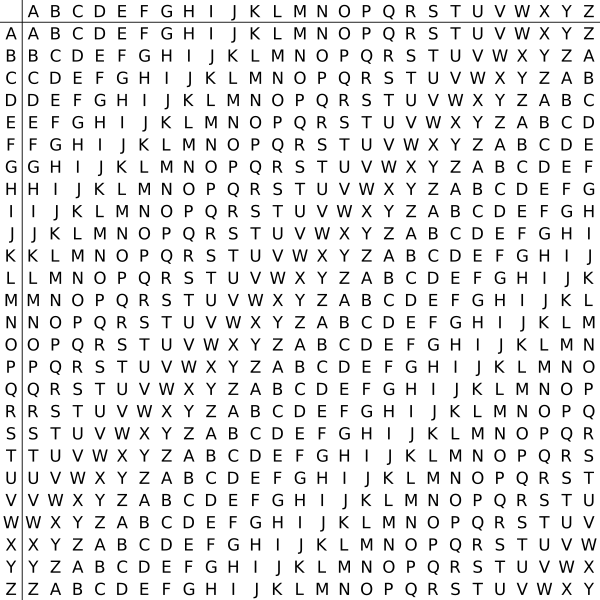
1. алгоритм (замена букв из полученного с помощью сдвига алфавита)
2. ключ - k (величина сдвига)

**Пример: зашифрованный текст с различными ключами.**

k=0 зашифрованный текст с различными ключами  
k=1 ибщйхспгбооьк уёлту т сбимйшоьнй лмяшбнй  
k=2 йвъкцтрдвппэл фжмуф у твйнкщпэок мнащвок  
k=3 кгылчусегррюм хзнфх ф угколърюпл нобъгпл  
k=4 лдьмшфтёдссян циохц х фдлпмысярм опвыдрм  
k=5 меэнщхужеттао чйпцч ц хемрньтасн пргьесн  
k=6 нёюоъцфзёуубп шкрчш ч цёнсоэубто рсдэёто

**Шифр Виженера**

[http://ru.wikipedia.org/wiki/Шифр\_Виженера](http://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80_%D0%92%D0%B8%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D0%B0)



Используется несколько алфавитов.

Исходный текст: HELLOWORLD

Создается ключевое слово: WELCOM

Ключевое слово повторяется циклически до заполнения всего исходного текста.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| W | E | L | C | O | M | W | E | L | C |
| H | E | L | L | O | W | O | R | L | D |

Берем букву исходного теста, выбираем алфавит соответствующий букве ключевого слова, и делаем замену из этого алфавита.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| W | E | L | C | O | M | W | E | L | C |
| H | E | L | L | O | W | O | R | L | D |
| D | I | W | N | C | I | K | V | W | F |

Получили зашифрованный текст: DIWNCIKVWF

**Шифр Вернама (одноразовые блокноты)**

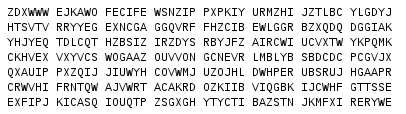
[http://ru.wikipedia.org/wiki/Одноразовый\_блокнот](http://ru.wikipedia.org/wiki/%D0%9E%D0%B4%D0%BD%D0%BE%D1%80%D0%B0%D0%B7%D0%BE%D0%B2%D1%8B%D0%B9_%D0%B1%D0%BB%D0%BE%D0%BA%D0%BD%D0%BE%D1%82)

Каждая страница блокнота является ключом и используется один раз.

Ключ должен обладать тремя свойствами:

1. быть истинно случайным
2. совпадать по размеру с заданным исходным текстом
3. применяться только один раз

Пример ключа:



**Алгоритм:**

Зашифруем исходный текст "HELLO" ключом "XMCKL"

Каждой букве присваивается порядковый номер в алфавите "A" => 0, "B" => 1 и т.д.

         H         E         L         L         O  исходный текст

     7 (H)   4 (E)  11 (L)  11 (L)  14 (O) исходный текст

+ 23 (X)  12 (M)   2 (C)  10 (K)  11 (L) ключ

= 30        16        13        21        25     исходный текст + ключ

=  4 (E)   16 (Q)  13 (N)  21 (V)  25 (Z) исходный текст + ключ (mod 26)

       E       Q       N       V       Z  ? зашифрованный текст

Если число больше 25, то начинают снова с начала алфавита, записывается остаток после вычитания 26 (30-26=4).

Расшифровка:

         E         Q          N         V          Z  зашифрованный текст

     4 (E)  16 (Q)  13 (N)  21 (V)  25 (Z) зашифрованный текст

 -  23 (X)  12 (M)   2 (C)  10 (K)  11 (L) ключ

 = -19         4        11        11       14     зашифрованный текст — ключ

=   7 (H)     4 (E)   11 (L)  11 (L)   14 (O) зашифрованный текст — ключ (mod 26)

         H          E          L          L          O  ? исходный текст

Если число отрицательное то прибавляется 26 (-19+26=7)

**Шифры перестановки**

**исходный текст:** простой шифр перестановки

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| п | т | и | е | т | в |
| р | о | ф | р | а | к |
| о | й | р | е | н | и |
| с | ш | п | с | о |  |

**шифрованный текст:** птиетв  рофрак ойрени сшпсо

Алгоритм: запись по столбцам, считывается по строкам.

Секретный ключ - размер таблицы.

**Поточные шифры**

[http://ru.wikipedia.org/wiki/Поточный\_шифр](http://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D0%BE%D1%87%D0%BD%D1%8B%D0%B9_%D1%88%D0%B8%D1%84%D1%80)

Генератор гаммы выдает ключевой поток бит (гамму), который, например, складывается (XOR-ИСКЛЮЧАЮЩЕЕ ИЛИ) с потоком открытого текста.

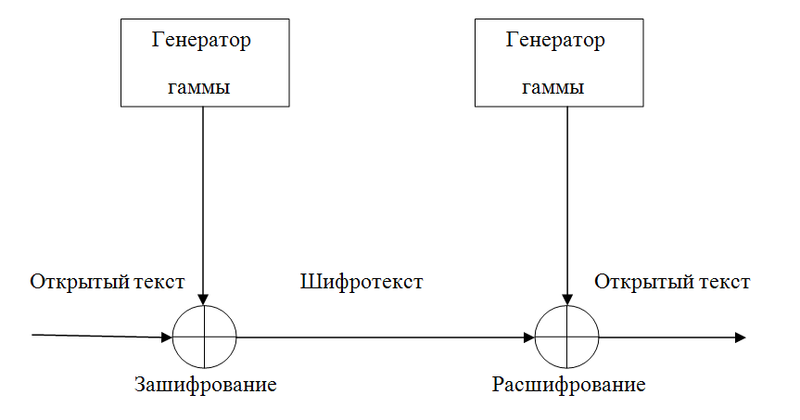


Рис. Принцип работы поточного шифра

**Шифр Вернама (одноразовые блокноты) тоже является поточным шифром.**

**Алгоритм XOR**

Работает с двоичным кодом.

правило:

0+0=0  
0+1=1  
1+0=1  
1+1=0

**исходный текст: Wiki**

Wiki переведем в двоичный код  - 01010111 01101001 01101011 01101001

Секретный ключ: 11110011 11110011 11110011 11110011

**Шифрование:**

01010111 01101001 01101011 01101001 - открытый текст  
11110011 11110011  11110011  11110011 - секретный ключ  
10100100 10011010 10011000 10011010 - шифрованный текст

**Расшифрование:**

10100100 10011010 10011000 10011010 - шифрованный текст  
11110011  11110011  11110011  11110011 - секретный ключ  
01010111  01101001 01101011 01101001 - открытый текст

**Блочные шифры**

[http://ru.wikipedia.org/wiki/Блочный\_шифр](http://ru.wikipedia.org/wiki/%D0%91%D0%BB%D0%BE%D1%87%D0%BD%D1%8B%D0%B9_%D1%88%D0%B8%D1%84%D1%80)

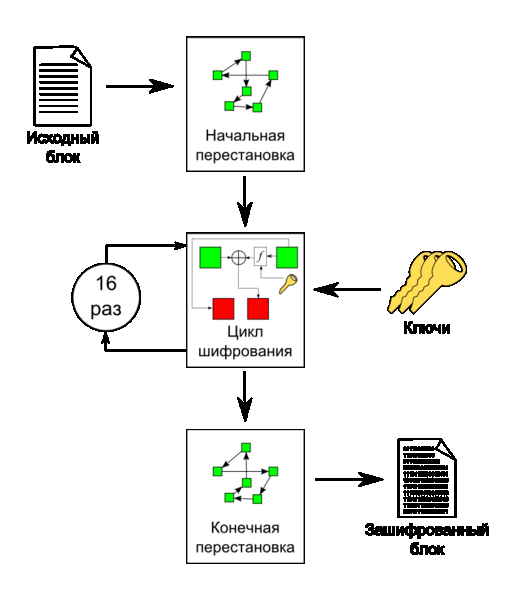
Данные шифруются блоками.

**DES (Data Encryption Standard)**

<http://ru.wikipedia.org/wiki/DES>

Данные шифруются блоками с размером 64 бит

Ключ: 56-бит.



**Начальная перестановка**  
Исходный текст T (блок 64 бит) преобразуется c помощью начальной перестановки IP которая определяется таблицей 1:

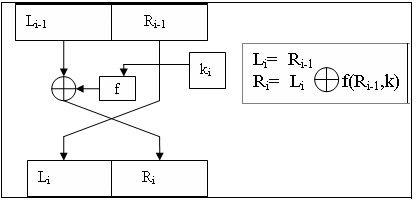
Таблица 1. Начальная перестановка IP

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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** |

В таблице новый порядок расположения бит в блоке. Номера это старый порядок (исходный блок).

**Циклы шифрования**  
После начальной перестановки 64-битовый блок данных IP(T) участвует в 16-циклах преобразования Фейстеля.  
— 16 циклов преобразования Фейстеля.

**Основная функция шифрования (функция Фейстеля)**



Блок данных IP(T) разбивается на две части по 32 бита L0 - левую, R0-правую.

Правая часть данных копируется в левую часть без изменений.  
Левая часть складывается по модулю два (xor - ИСКЛЮЧАЮЩЕЕ ИЛИ) с выходом из f-функции.

**f-функция:**

1) Функция Е расширяет 32-битовый вектор Ri ? 1 до 48-битового вектора E(Ri ? 1) путем дублирования некоторых битов из Ri ? 1; при этом порядок битов вектора E(Ri ? 1) указан в таблице 2.

Таблица 2. Функция расширения E

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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** |

По таблице 2 видно, что биты 1, 4, 5, 8, 9, 12, 13, 16, 17, 20, 21, 24, 25, 28, 29, 32 дублируются, и делается перестановка.

2) E и раундовый ключ ki(48 бит-генерация раундовых ключей показана ниже) складываются по модулю 2.

3) полученный 48 битный блок данных делится на 8, получаются 8-мь 6 битных блоков (Bj).  
- каждый 6 битный блок Bj преобразуется (Sj -блоком) в 4 битный блок.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Таблица 3. Преобразования *Si*, i=1…8** | | | | | | | | | | | | | | | | | |
|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **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** | ***S*1** |
| **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** |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **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** | ***S*2** |
| **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** |  |
| **1** | **13** | **7** | **0** | **9** | **3** | **4** | **6** | **10** | **2** | **8** | **5** | **14** | **12** | **11** | **15** | **1** | ***S*3** |
| **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** | ***S*4** |
| **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** | ***S*5** |
| **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** | ***S*6** |
| **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** | ***S*7** |
| **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** | ***S*8** |
| **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** |  |

 Пример преобразование через S-блок:

1. берем блок **B**3 = 101111
2. первый и последний биты **B**3 =>11 являются двоичной записью десятичного числа **а**,  0<=**a**<=3, поэтому строки таблицы **S**3 нумеруются от 0 до 3
3. средние 4 бита =>0111 представляют десятичное число **b**, 0<=**b**<=15, столбцы таблицы **S**3 нумеруются от 0 до 15.
4. пара чисел (**а, b**) определяет десятичное число, находящееся в пересечении строки **а** и столбца **b**. Двоичное представление этого числа дает **B'**3 . В нашем случае **a** = 11 = 3, **b** = 0111 = 7, а десятичное число, определяемое парой (3,7), равно 7.
5. двоичное представление числа 7  B'3=0111.

Пример таблицы в двоичном виде, преобразование **B**5 = 011011 через  S5-блок  **B'5** = 1001

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S5** | | **Middle 4 bits of input** | | | | | | | | | | | | | | | |
| **0000** | **0001** | **0010** | **0011** | **0100** | **0101** | **0110** | **0111** | **1000** | **1001** | **1010** | **1011** | **1100** | **1101** | **1110** | **1111** |
| **Outer bits** | **00** | 0010 | 1100 | 0100 | 0001 | 0111 | 1010 | 1011 | 0110 | 1000 | 0101 | 0011 | 1111 | 1101 | 0000 | 1110 | 1001 |
| **01** | 1110 | 1011 | 0010 | 1100 | 0100 | 0111 | 1101 | 0001 | 0101 | 0000 | 1111 | 1010 | 0011 | 1001 | 1000 | 0110 |
| **10** | 0100 | 0010 | 0001 | 1011 | 1010 | 1101 | 0111 | 1000 | 1111 | 1001 | 1100 | 0101 | 0110 | 0011 | 0000 | 1110 |
| **11** | 1011 | 1000 | 1100 | 0111 | 0001 | 1110 | 0010 | 1101 | 0110 | 1111 | 0000 | 1001 | 1010 | 0100 | 0101 | 0011 |

Снова получаем 32 бита.  
- все 8 блоков по 4 бита пропускаются через **P-блок**  
Значение функции f(Ri ? 1,ki) (32 бит) получается перестановкой Р, применяемой к 32-битовому блоку B'1B'2...B'8. Перестановка Р задk=3 кгылчусегррюм хзнфх ф угколърюпл нобъгплtd style=font-size: small;p style=/strong/strongth style=background: #90ff00;/trthth1ана таблицей 4.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Таблица 4. Перестановка P** | | | | | | | |
| **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** |

 f(Ri ? 1,ki) = P(B'1B'2...B'8)  
Согласно таблице 4, первые четыре бита результирующего вектора после действия функции f — это биты 16, 7, 20, 21 вектора B'1B'2...B'8

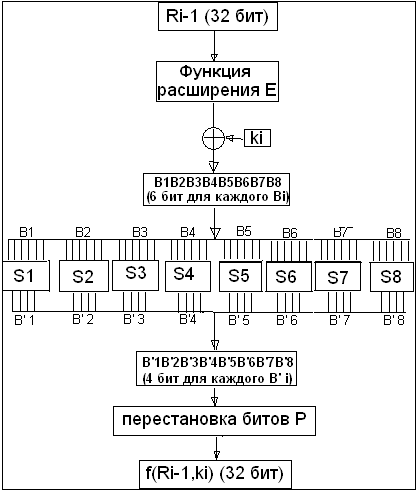


Рис. f-функция

**Генерирование ключей ki для каждого раунда.**  
  
Раундовые ключи ki получаются из начального ключа k добавлением бит из блока данных (64 бит = 8 байтов или 8 символов в ASCII) таким образом. Восемь битов, находящих в позициях 8, 16, 24, 32, 40, 48, 56, 64 добавляются в ключ k таким образом чтобы каждый байт содержал нечетное число единиц. Затем делают перестановку для расширенного ключа (кроме добавляемых битов 8, 16, 24, 32, 40, 48, 56, 64). Такая перестановка определена как в таблице 5.

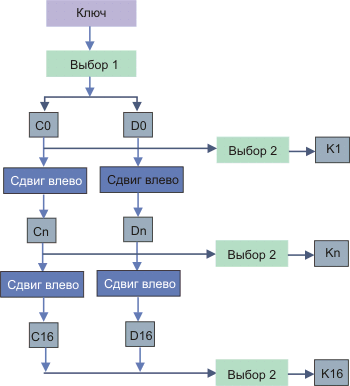
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Таблица 5.** | | | | | | | | | | | | | | |
| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 | 58 | 50 | 42 | 34 | 26 | 18 | *C*0 |
| 10 | 2 | 59 | 51 | 43 | 35 | 27 | 19 | 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 |  |

Эта перестановка определяется двумя блоками C0 и D0 по 28 бит каждый. Первые 3 бита C0 есть биты 57, 49, 41 расширенного ключа. А первые три бита D0 есть биты 63, 55, 47 расширенного ключа. Ci,Di i=1,2,3…получаются из Ci ? 1,Di ? 1 одним или двумя левыми циклическими сдвигами согласно таблице 6.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Таблица 6.** | | | | | | | | | | | | | | | | |
| **i** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| Число сдвига | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 |

Ключ ki, i=1,…16 состоит из 48 бит, выбранных из битов вектора CiDi (56 бит) согласно таблице 7. Первый и второй биты ki есть биты 14, 17 вектора CiDi

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Таблица 7.** | | | | | | | | | | | | | | | |
| 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 |



Получение раундовых ключей

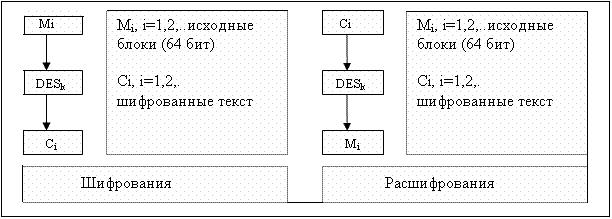
**Конечная перестановка**

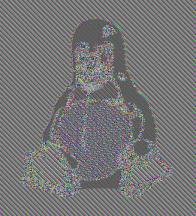
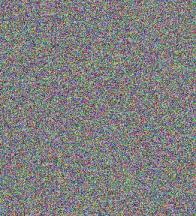
Конечная перестановка IP ? 1 действует на T16 и используется для восстановления позиции. Она является обратной к перестановке IP. Конечная перестановка определяется таблицей 8.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Таблица 8. Обратная перестановка *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 |

**Режимы использования DES**

**(ECB — Electronic Code Book)** - каждый блок шифруется отдельно.



оригинальная битовая карта            криптограмма в режиме ECB         другие режимы шифрования

**(СВС — Cipher Block Chaining) -**режим сцепления блоков. Каждый очередной блок, перед зашифровыванием складывается по модулю 2 со следующим блоком открытого текста.

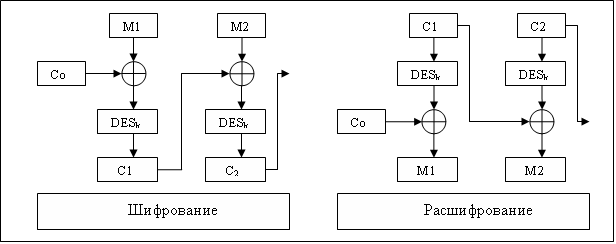


Рис. СВС**-**режим сцепления блоков

 Существуют и другие рижимы использования DES.