****

**Московский авиационный институт**

**(Национальный исследовательский университет)**

**Институт № 3**

**Кафедра 311**

**Информационная безопасность**

**Лабораторная работа № 5**

**«Шифр DES»**

**Выполнил студент  
Максимов Евгений Викторович**

**Группа М3З-302-БК**

**Дата 07.07.2022 г.**

**Принял преподаватель  
Кос Оксана Игоревна**

Оглавление

[Цель лабораторной работы 3](#_Toc89796281)

[Глава 1. Принцип Шифра DES. 4](#_Toc89796282)

[Глава 2. Схема шифрования DES 5](#_Toc89796283)

[Глава 3. Функция Фейстеля 7](#_Toc89796284)

[Глава 4. Генерирование ключей ki 9](#_Toc89796285)

[Глава 3. Шифрование текста 10](#_Toc89796286)

[Глава 4. Расшифровка текста. 11](#_Toc89796287)

[Глава 3. Листинг программы 12](#_Toc89796288)

[Итоги лабораторной работы 13](#_Toc89796289)

[Список литературы 14](#_Toc89796290)

# Цель лабораторной работы

Изучить:

* 1. Принцип шифрования DES;
  2. Зашировать текст DES;
  3. Расшифрованить текст.

# Глава 1. Принцип Шифра DES.

DES (англ. Data Encryption Standard) — алгоритм для симметричного шифрования, разработанный фирмой IBM и утверждённый правительством США в 1977 году как официальный стандарт (FIPS 46-3). Размер блока для DES равен 64 битам. В основе алгоритма лежит сеть Фейстеля с 16 циклами и ключом, имеющим длину 56 бит. Алгоритм использует комбинацию нелинейных и линейных преобразований.

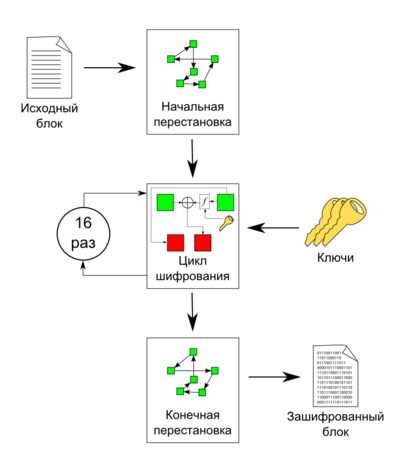


Рисунок 1.1 – Схема шифрования DES

# Глава 2. Схема шифрования DES

1. Начальная перестановка:

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

Таблица 1. Начальная перестановка IP

По таблице первые 3 бита результирующего блока IP(T) после начальной перестановки IP являются битами 58, 50, 42 входного блока T, а его 3 последние бита являются битами 23, 15, 7 входного блока.

1. Циклы шифрования:

Полученный блок участвует в 16 циклах преобразования Фейстеля.

Разбить IP(T) на две части L0, R0, гдe L0, R0, — соответственно 32 старших битов и 32 младших битов блока T0IP(T) = L0R0

Пусть Ti-1 = Li-1Ri-1 результат (i-1) итерации, тогда результат i-ой итерации Ti = LiRi определяется:

Li = Ri-1

Ri = Li-1 + f(Ri-1, ki)

Левая половина Li равна правой половине предыдущего вектора Li-1Ri-1. А правая половина Ri — это битовое сложение Li-1 и f(Ri-1, ki) по модулю 2.

В 16-циклах преобразования Фейстеля функция f играет роль шифрования.

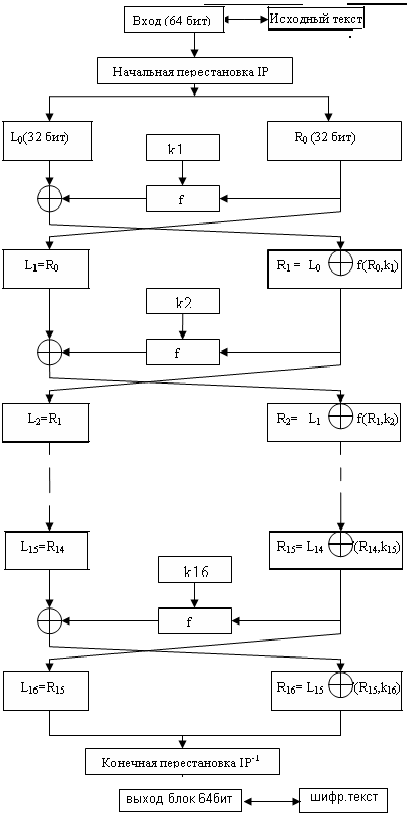


Рисунок 2.1 – Подробная схема шифрования DES

# Глава 3. Функция Фейстеля

Аргументами функции f являются 32-битовый вектор Ri-1 и 48-битовый ключ ki, который является результатом преобразования 56-битового исходного ключа шифра k. Для вычисления функции f последовательно используются:

1. функция расширения E;
2. сложение по модулю 2 с ключом ki;
3. преобразование S, состоящее из 8 преобразований S-блоков S1S2S3…S8;
4. перестановка P.

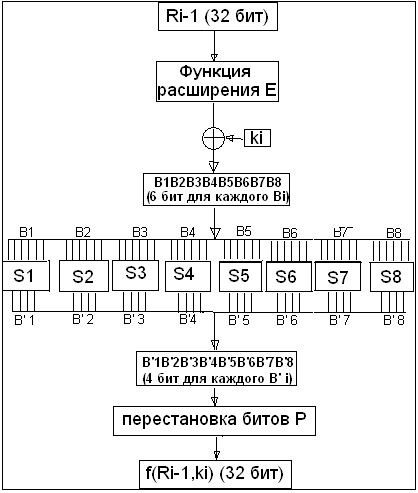


Рисунок 3.1 – Функция Фейстеля

1. Функция E расширяет 32-битовый вектор Ri-1 до 48-битового вектора   
   Е(Ri-1) путём дублирования некоторых битов из Ri-1

Порядок битов вектора Е(Ri-1) указан в таблице 2.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 2. Функция расширения Е

1. Полученный после перестановки блок Е(Ri-1) складывается по модулю 2 с ключами ki и затем представляется в виде восьми последовательных блоков B1, B2, B3, …B8.
2. Каждый Bj является 6-битовым блоком. Далее каждый блок Bj транформируется в 4-битовый блок B`j с помощью преобразований Sj.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a/b | 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 | S1 |
| 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 |

Таблица 3. Преобразования Sj

Предположим, B1 = 101111. Первый и последний разряды являются двоичной записью числа a (112 = 3), средние 4 разряда представляют число b (01112 = 7).

На пересечении числе 3 и 7 находится число 7, его двоичное представление 0111. => B`1 = 0111.

1. Значение функции f(Ri-1, ki) получается перестановкой P, применяемой к 32-битовому блоку B`1, B`2, B`3, …B`8.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

Таблица 4 – перестановка Р

# Глава 4. Генерирование ключей ki

Ключи ki получаются из начального ключа k (56 бит = 7 байтов или 7 символов в ASCII) следующим образом. Добавляются биты в позиции 8, 16, 24, 32, 40, 48, 56, 64 ключа k таким образом, чтобы каждый байт содержал нечетное число единиц. Это используется для обнаружения ошибок при обмене и хранении ключей. Затем делают перестановку для расширенного ключа (кроме добавляемых битов 8, 16, 24, 32, 40, 48, 56, 64). Такая перестановка определена в таблице 5.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 | 58 | 50 | 42 | 34 | 26 | 18 | C0 |
| 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 | D0 |
| 14 | 6 | 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 | 28 | 20 | 12 | 4 |

Таблица 5 – перестановка k

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. CiDi, i=1,2,3…получаются из   
Ci-1Di-1 одним или двумя левыми циклическими сдвигами согласно таблице 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 |

Таблица 6 – таблица сдвигов ключа

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.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

Таблица 7 – перестановка вектора CiDi

# Глава 3. Шифрование текста

**Задача:** зашифровать текст алгоритмом DES

**Исходный текст:** «Съешь еще этих мягких французских булок да выпей чаю»

**Зашифрованный текст:** «LrccIpzpmhop1WhrRaAYS9EOze8d9HP0o49hgejoUs4tj8dz7YHHUpP86K5IKuozTFlR3vP0FAb3wAxQ2RmO6oJquSO30Rs0ls+HOdPAZ0obR3ckShQrIvOC4PT11uwDjNZG5eFwWEI=»

\*Зашифрованные биты кодируются в Base64 кодировку для удобства восприятия

# Глава 4. Расшифровка текста.

**Задача:** расшифровать текст, зашифрованный алгоритмом DES

**Зашифрованный текст:** «LrccIpzpmhop1WhrRaAYS9EOze8d9HP0o49hgejoUs4tj8dz7YHHUpP86K5IKuozTFlR3vP0FAb3wAxQ2RmO6oJquSO30Rs0ls+HOdPAZ0obR3ckShQrIvOC4PT11uwDjNZG5eFwWEI=»

**Расшифрованный текст:** «Съешь еще этих мягких французских булок да выпей чаю»

# Глава 3. Листинг программы

public class DES {  
 private final static String SOURCE\_STRING = "Съешь еще этих мягких французских булок да выпей чаю";  
 private static Cipher DES;  
 private static SecretKey KEY;  
  
 public static void main(String[] args) throws Exception {  
 DES = Cipher.getInstance("DES");  
 KEY = KeyGenerator.getInstance("DES").generateKey();  
 DES.init(ENCRYPT\_MODE, KEY);  
  
 String encodedString = encode(SOURCE\_STRING);  
 DES.init(DECRYPT\_MODE, KEY);  
 String decodedString = decode(encodedString);  
  
 System.out.println("Исходная строка: " + SOURCE\_STRING);  
 System.out.println("Зашифрованная строка: " + encodedString);  
 System.out.println("Расшифрованная строка: " + decodedString);  
 }  
  
 private static String decode(String encodedString) throws IOException, IllegalBlockSizeException, BadPaddingException {  
 byte[] encodedStringBytes = new BASE64Decoder().decodeBuffer(encodedString);  
 byte[] decodedBytes = DES.doFinal(encodedStringBytes);  
  
 return new String(decodedBytes, UTF\_8);  
 }  
  
 private static String encode(String sourceString) throws IllegalBlockSizeException, BadPaddingException {  
 byte[] sourceStringBytes = sourceString.getBytes(UTF\_8);  
 byte[] encryptedBytes = DES.doFinal(sourceStringBytes);  
  
 return new BASE64Encoder().encode(encryptedBytes);  
 }  
}

# Итоги лабораторной работы

Мы научились шифровать и расшифровывать текст методом DES.

# Список литературы

1. [[Wikipedia.org](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80_%D0%A6%D0%B5%D0%B7%D0%B0%D1%80%D1%8F)] – DES;