



Министерство науки и высшего образования Российской  
Федерации  
Федеральное государственное бюджетное образовательное  
учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ ИУ «Информатика и системы управления»

---

КАФЕДРА ИУ-7 «Программное обеспечение ЭВМ и информационные технологии»

---

## ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

*по дисциплине «Защита информации»*

*«Алгоритм шифрования DES»*

Студент группы ИУ7-72Б

Фам М. Х.

Руководитель

Чижев И. С.

2024 г.

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>4</b>
<b>1 Аналитическая часть</b>	<b>5</b>
1.1 Алгоритм DES . . . . .	5
1.2 DES-CBC . . . . .	7
<b>2 Конструкторская часть</b>	<b>8</b>
2.1 Разработка алгоритмов . . . . .	8
<b>3 Технологическая часть</b>	<b>11</b>
3.1 Средства реализации . . . . .	11
3.2 Реализация алгоритма . . . . .	11
3.3 Тестирование . . . . .	13
<b>Заключение</b>	<b>16</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>17</b>

# ВВЕДЕНИЕ

Шифрование информации — занятие, которым человек занимался ещё до начала первого тысячелетия, занятие, позволяющее защитить информацию от посторонних лиц.

Шифровальный алгоритм DES — алгоритм, разработанный в 1977 году компанией IBM и являющийся официальным стандартом шифрования.

**Целью данной работы** является реализация в виде программы на языке программирования С или С++ шифровального алгоритма DES в режиме работы CBC.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- 1) изучить шифровальный алгоритм DES и его режим работы CBC;
- 2) реализовать шифровальный алгоритм DES в виде программы, обеспечив возможности шифрования и расшифровки файла в режиме работы CBC;
- 3) описать и обосновать полученные результаты в отчёте о выполненной лабораторной работе.

# 1 Аналитическая часть

## 1.1 Алгоритм DES

Шифровальная алгоритм DES (англ. *Data Encryption Standard* — DES) — симметричный шифровальный алгоритм, разработанный в 1977 году компанией IBM. Он использует блочное шифрование, длина блока фиксирована и равна 64 битам. Однако каждые 8 бит в ключе игнорируются, что приводит к правильной длине ключа 56 бит в DES. Однако в любом случае один блок на 64 бита является вечной организацией DES. Он состоит из 3 следующих шагов, рисунок 1.1:

- начальная перестановка (англ. *Initial Permutation* — IP), во время которой биты переставляются в порядке, определённом в специальной таблице;
- 16 раундов шифрования;
- завершающей перестановки (англ. *Final Permutation* — FP), совершающей преобразования, обратные сделанным на первом шаге.

Раунд шифрования состоит из 5 следующих этапов

- 1) расширение (англ. *expansion* — E);
- 2) получение ключа раунда (англ. *Round Key* — RK);
- 3) скремблирование (англ. *substitution* — S);
- 4) перестановка (англ. *permutation* — P)
- 5) смешивание ключа (англ. *key mixing* — KM).



Рисунок 1.1 – Обобщенная схема шифрования в алгоритме DES

Расширение, во время которого каждая из половин блока шифрования по 32 бит дополняется путём перестановки и дублирования бит до длины в 48 бит.

Получение ключа раунда необходимо для применения в раунде шифрования 48-битного ключа раунда, полученного из основного ключа DES. Основной ключ имеет длину 64 бита, однако значащих бит из 64 всего 56, остальные добавлены для избыточности и контроля передачи ключа. Из этих 56 бит получают 48 путём разбиения на равные части и применению битовой операции циклического сдвига и нахождению нового значения посредством специальной таблицы.

Скремблирование предназначено для получения из 48-битного потока 32-битного путём разбиения на 6 частей по 8 бит и обработки каждой части в S-блоках (англ. *Substitution boxes*), которые заменяют блоки с длиной 6 бит на блоки 4 бит посредством использования специальной таблицы.

Перестановка представляет из себя перемешивания полученной после-

довательности из 32 бит при помощи таблицы перемешивания.

Смешивание ключа представляет из себя операцию XOR полученного 32-битного значения с ключом раунда.

## 1.2 DES-CBC

На рисунке 1.2 приведена схема алгоритма DES в режиме CBC.

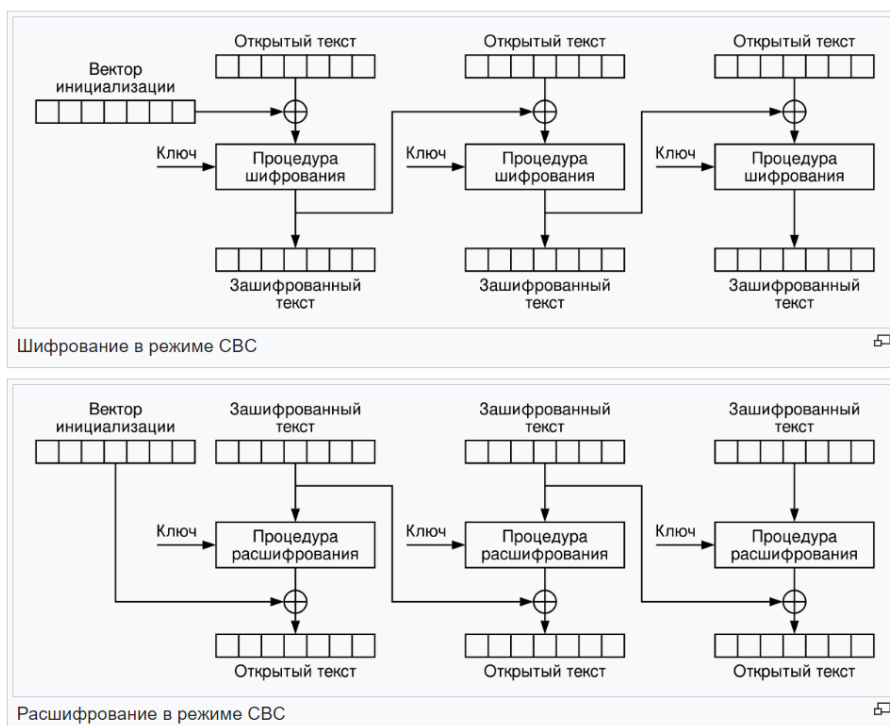


Рисунок 1.2 – Режим сцепления блоков шифротекста CBC

## 2 Конструкторская часть

В этом разделе представлена схема алгоритма шифровальной машины «Энигма».

### 2.1 Разработка алгоритмов

На рисунках ??–2.3 представлены схемы алгоритмов DES, раунда DES, функции Фейстеля.

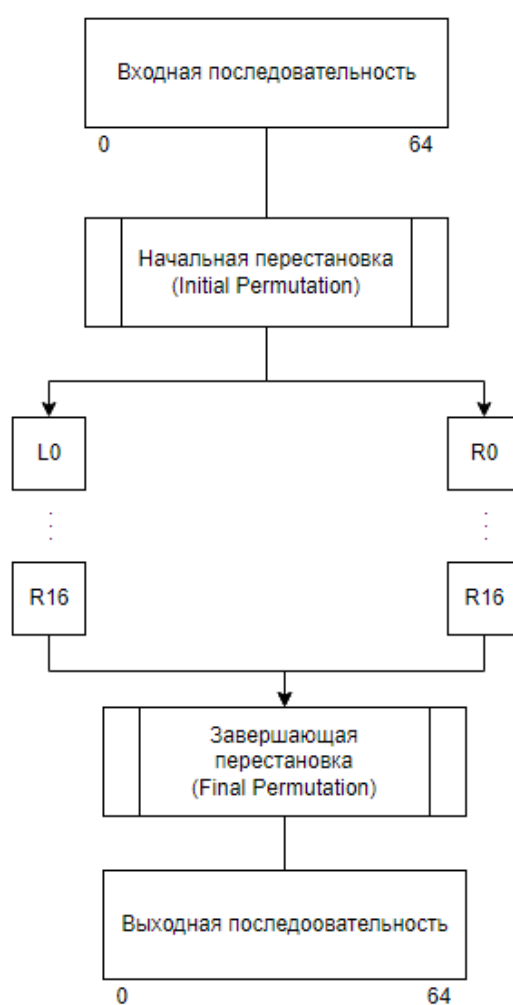


Рисунок 2.1 – Схема алгоритма DES

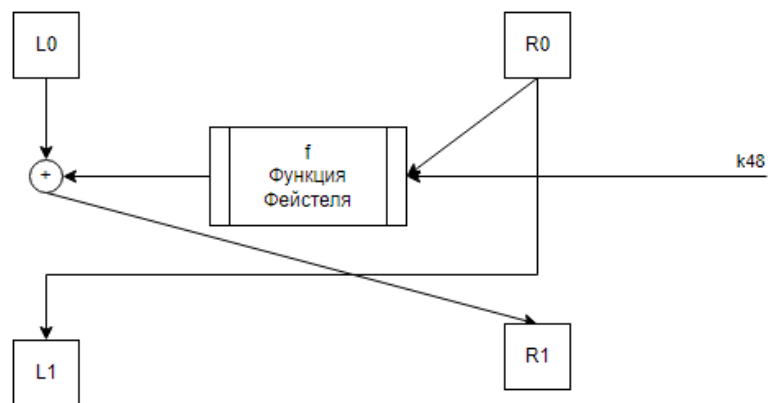


Рисунок 2.2 – Схема алгоритма раунда DES



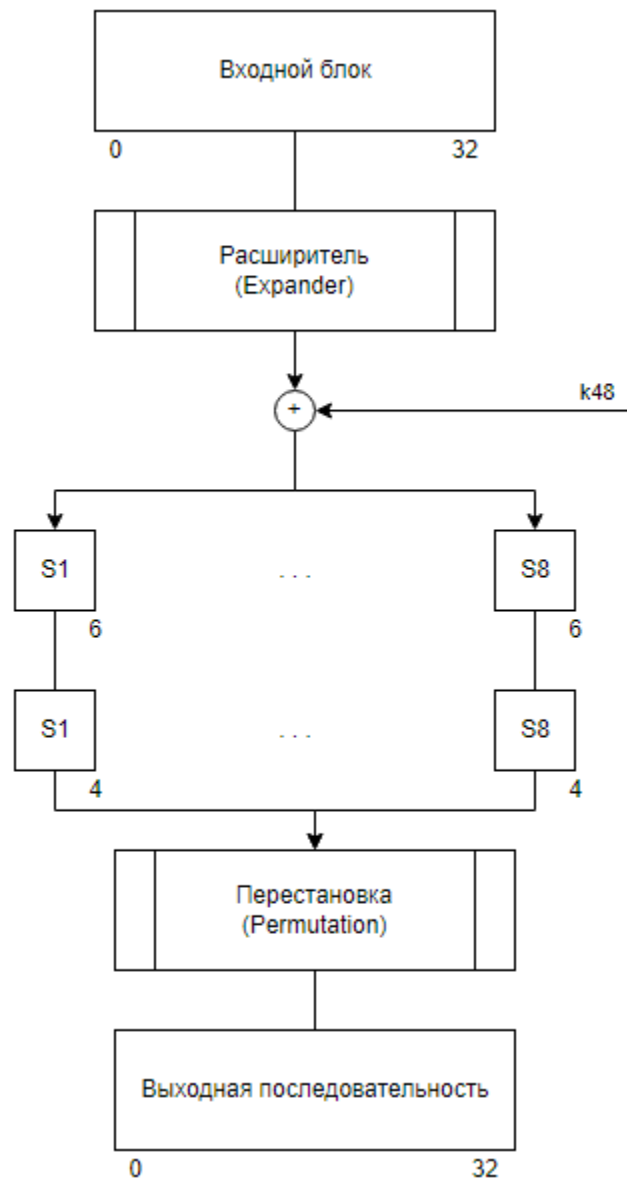


Рисунок 2.3 – Схема алгоритма функции Фейстеля

## 3 Технологическая часть

### 3.1 Средства реализации

Для программной реализации шифровальной машины был выбран язык C++ [1]. В данном языке есть все требующиеся инструменты для данной лабораторной работы. В качестве среды разработки была выбрана среда Visual Studio 2022 [2].

### 3.2 Реализация алгоритма

Листинг 3.1 – Реализация алгоритма DES

```
1      bitset<64> process_block(bitset<64> value, bitset<64> key,
2      bool decrypte = false)
3      {
4          // generate keys
5          auto keys = generate_keys(key, decrypte);
6
7          // initial permutation
8          auto round_val = IP_f(value);
9
10         // 16 rounds
11         for (auto rkey : keys) {
12             round_val = wround(round_val, rkey);
13         }
14
15         // final permutation
16         auto final_val = FP_f(round_val);
17
18         return final_val;
19     }
```

Листинг 3.2 – Реализация метода шифрования и дешифрования DES в режиме CBC

```
1 vector<char> cypher(vector<char> input, vector<char> key,
2     vector<char> vi)
3 {
4     vector<char> buffer = {};
5     vector<char> result = {};
6
7     int last_cnt = 0;
8
9     auto vi_initial = vchar_to_bitset64(vi);
10    auto key_b = vchar_to_bitset64(key);
11
12    for (auto sym : input) {
13        if (buffer.size() < 8) {
14            buffer.push_back(sym);
15        }
16
17        if (buffer.size() == 8) {
18            // buffer size is 8 -> can be cyphered
19            auto buf_b = vchar_to_bitset64(buffer) ^ vi_initial;
20
21            auto tmp_b_1 = _des.process_block(buf_b, key_b);
22
23            vi_initial = tmp_b_1;
24
25            auto tmp_res = bitset64_to_vchar(tmp_b_1);
26
27            result.insert(result.end(), tmp_res.begin(),
28                tmp_res.end());
29
30            buffer.clear();
31        }
32    }
```

```

31     }
32
33     if (buffer.size() > 0 && buffer.size() < 8) {
34         while (buffer.size() < 8) {
35             buffer.push_back((char)0);
36             last_cnt += 1;
37         }
38         auto buf_b = vchar_to_bitset64(buffer) ^ vi_initial;
39
40         auto tmp_b_1 = _des.process_block(buf_b, key_b);
41
42         vi_initial = tmp_b_1;
43
44         auto tmp_res = bitset64_to_vchar(tmp_b_1);
45
46         result.insert(result.end(), tmp_res.begin(),
47                        tmp_res.end());
48     }
49     result.push_back((char)last_cnt);
50
51
52     return result;
53 }
54 }

```

### 3.3 Тестирование

Таблица 3.1 – Функциональные тесты

Входная строка	Выходная строка
HelloEnigma123	>hfeinduehfo2m
hello	1ma8r
A	k
«»	«»

## Вывод

В данном разделе были рассмотрены средства реализации, а также представлены листинги реализации шифровального алгоритма DES и режима работы CBC.

# Заключение

В результате лабораторной работы был реализован в виде программы шифровальный алгоритм DES в режиме работы CBC.

Были выполнены следующие задачи:

- 1) изучен шифровальный алгоритм DES и его режим работы CBC;
- 2) реализован шифровальный алгоритм DES в виде программы, обеспечивающей возможность шифрования и расшифровки файла в режиме работы CBC;
- 3) описаны и обоснованы полученные результаты в отчёте о выполненной лабораторной работе.

## Список использованных источников

1. Язык программирования C++. <https://learn.microsoft.com/en-us/cpp/cpp/cpp-language-reference?view=msvc-170>. дата обращения: 15.10.2024.
2. Visual Studio 2022. <https://visualstudio.microsoft.com/vs/>. дата обращения: 15.10.2024.