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 «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчёт по лабораторной работе №2 по курсу «Защита информации»

Тема Шифровальный алгоритм DES

Студент Калашков П. А.

Группа ИУ7-76Б

Оценка (баллы)

Преподаватели Чиж И. С.

Москва — 2023 г.

Введение

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

Шифровальная 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 C или C++ шифровального алгоритма DES в режиме работы CFB — режима обратной связи по шифротексту.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1) изучить шифроовальный алгоритм DES и его режим работы CFB;

2) реализовать шифровальный алгоритм DES в виде программы, обеспечив возможности шифрования и расшифровки файла в режиме работы CFB;

3) протестировать разработанную программу, показать, что удаётся дешфировать все файлы;

4) описать и обосновать полученные результаты в отчёте о выполненной лабораторной работе.

1 Аналитическая часть

В этом разделе будут рассмотрен шифровальный алгоритм DES, а также его работа в режиме обратной связи по шифротексту (CFB).

1.1 Алгоритм DES

Шифровальная алгоритм DES (англ. *Data Encryption Standart* — DES) — симметричный шифровальный алгоритм, разработанный в 1977 году компанией IBM. Он использует блочное шифрование, длина блока фиксирована и равна 64 битам. Он состоит из 3 следующих шагов: начальная перестановка (англ. *Initial Permutation* — IP), во время которой биты переставляются в порядке, определённом в специальной таблице, 16 раундов шифрования, а также завершающей перестановки (англ. *Final Permutation* – FP), соверщающей преобразования, обратные сделанным на первом шаге. Рассмотрим подробнее раунд шифрования.

1.1.1 Раунд шифрования

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).

Расширение, во время которого каждая из половин блока шифрования по 32 бит дополняется путём перестановки и дублировоания 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 последовательности из 32 бит при помощи таблицы перемешивания.

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

1.2 Режимы работы алгоритма DES

Режим шифрования — метод применения блочного шифра, позволяющий преобразовать последовательность блоков открытых данных в последовательность блоков зашифрованных данных.

Для DES рекомендованы следующие режими работы:

1) режим электронной кодовой книги (англ. *Electronic Code Bloc* — ECB);

2) режим сцепления блоков (англ. *Cipher Block Chaining* – CBC);

3) режим параллельнооого сцепления блоков (англ. *Parallel Cipher Block Chaining* – PCBC);

4) режим обратной связи по шифротексту (англ. *Cipher Feed Back* — CFB);

5) режим обратной связи по выходу (англ. *Output Feed Back* — OFB).

В данной работе будет рассмотрен режим обратной связи по шифротексту (CFB).

1.2.1 Режим обратной связи по шифротексту

В данном режиме используется вектор исполнения (англ. *Initialization vector* — IV)— случайная последовательность символов, которую добавляют к ключу шифрования для повышения его безопасности. Он затрудняет определение закономерностей в рядах данных и делает их более устойчивыми ко взлому.

В режиме CFB вектор исполнения IV зашифровывается при помощи алгоритма DES с использованием основного ключа. После этого, если происходит зашифровка, полученное значение подвергается операции XOR с блоком, который нужно зашифровать. Полученное значение является зашифрованным блоком данных и значением IV для следующего шифрования — это обеспечивает взаимосвязь блоков данных.

Если происходит расшифровка, полученное значение подвергается операции XOR с блоком, который нужно расшифровать. Полученное значение является расшифрованным блоком данных, а блок зашифрованного текста явлляется следующим значением IV.

Вывод

В данном разделе был рассмотрен шифровальный алгоритм DES, его составляющие и режимы работы, а также режим обратной связи по шифротексту (CFB).

2 Конструкторская часть

В этом разделе будут представлены описания используемых типов данных, а также схема алгоритма разрабатываемой программы.

2.1 Сведения о модулях программы

Программа состоит из четырёх модулей:

1) *main.c* — файл, содержащий точку входа;

2) *menu.c* — файл, содержащий код меню программы;

3) *des.c* — файл, содержайший реализацию алгоритма шифрования DES;

4) *cfb.c* — файл, содержащий реализацию режима работы CFB.

2.2 Разработка алгоритмов

На рисунках 2.1–2.5 представлены схемы алгоритмов DES, раунда DES, функции Фейстеля, а также режимы работы CFB при зашифровке и расшифровке.

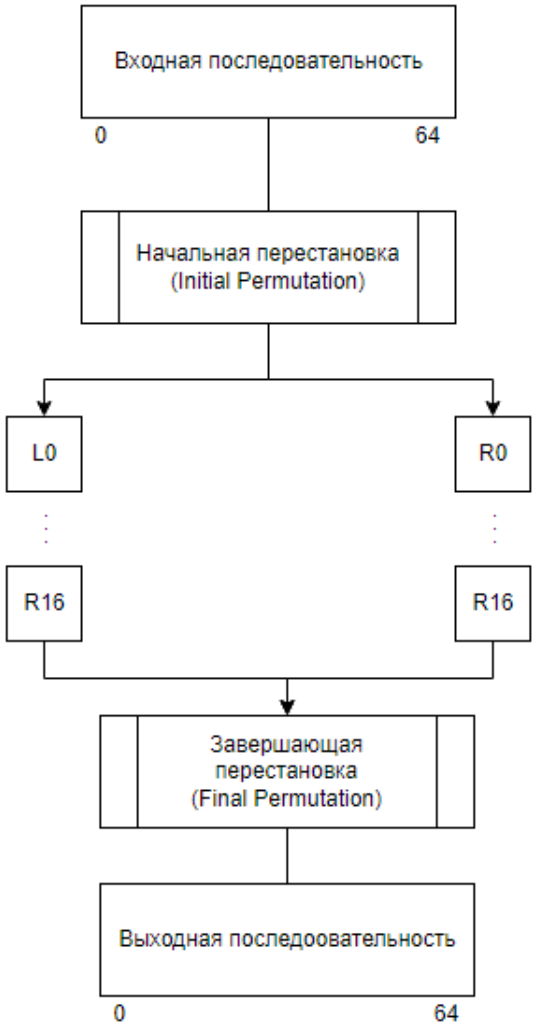

Рисунок 2.1 – Схема шифровального алгоритма DES

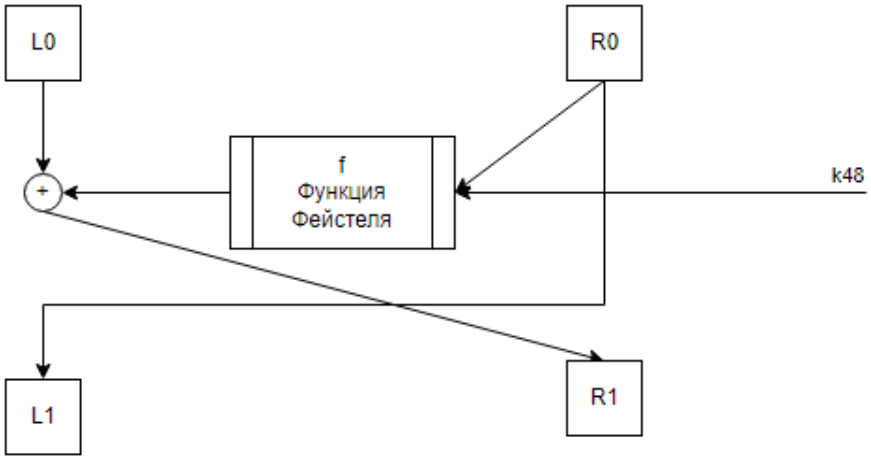

Рисунок 2.2 – Схема алгоритма раунда DES

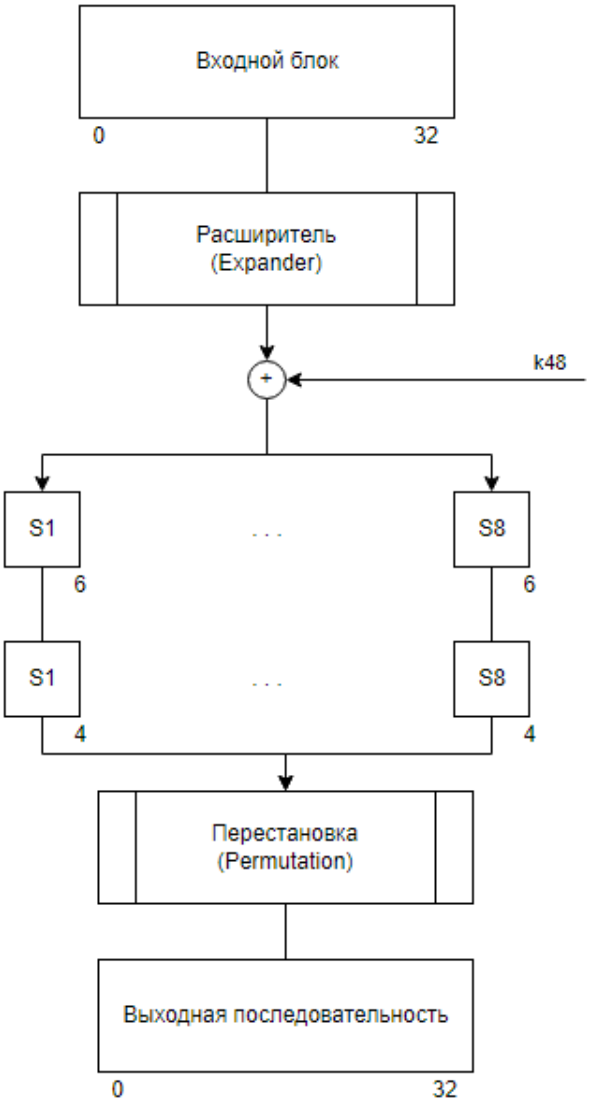

Рисунок 2.3 – Схема алгоритма функции Фейстеля

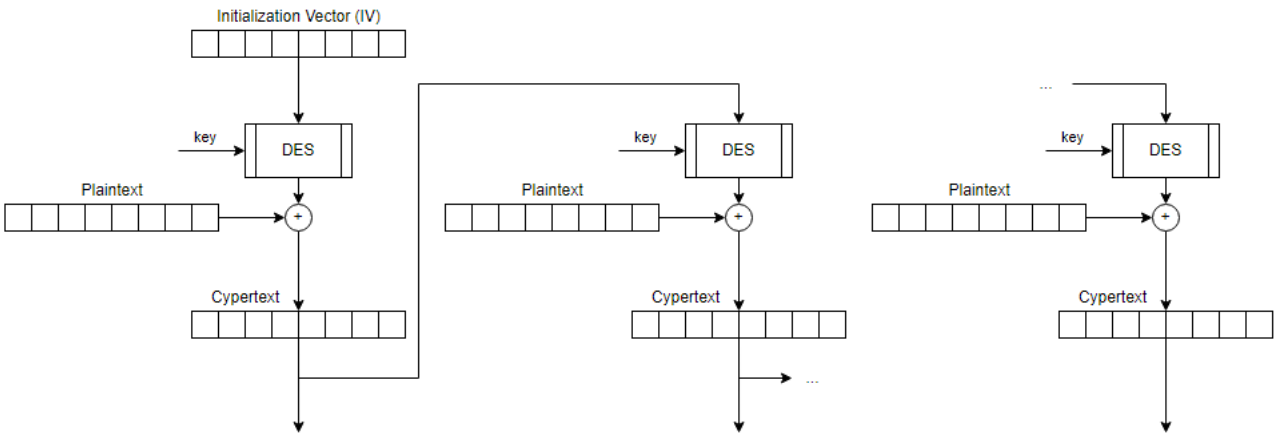

Рисунок 2.4 – Схема алгоритма режимы работы CFB при зашифровке

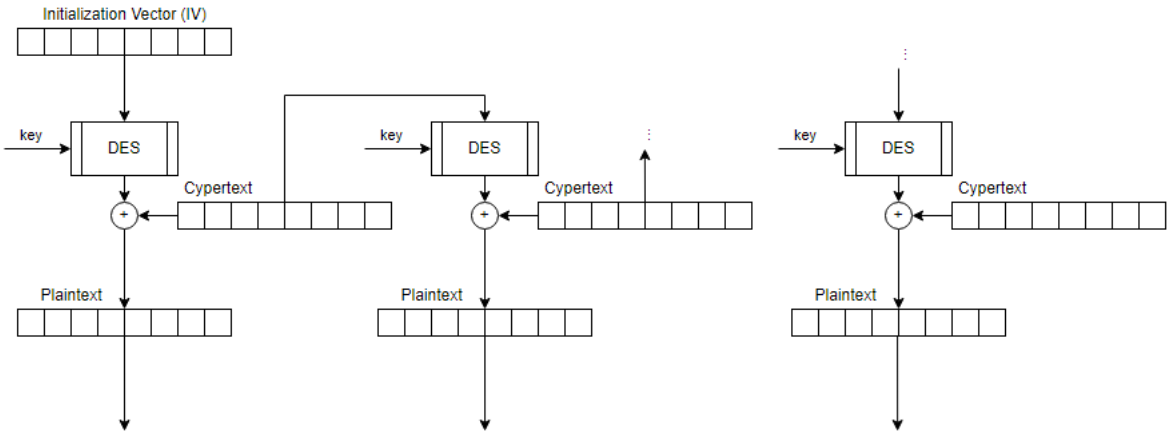

Рисунок 2.5 – Схема алгоритма режимы работы CFB при зашифровке

Вывод

В данном разделе были представлены сведения о модулях программы, а также схемы алгоритмов, которые нужно реализовать: алгоритма DES и его раунда с функцией Фейстеля, а также режима работы CFB с зашифровкой и расшифровкой.

3 Технологическая часть

В данном разделе будут рассмотрены средства реализации, а также представлены листинги реализации шифровального алгоритма DES и режима работы CFB, а также произведено тестирование.

3.1 Средства реализации

В данной работе для реализации был выбран язык программирования C. Данный язык удоволетворяет поставленным критериям по средствам реализации.

3.2 Реализация алгоритма

В листингах 3.1–3.3 представлена реализация шифровального алгоритма DES, на листинге 3.4 — реализация режима работы CFB.

Листинг 3.1 – Реализация шифровального алгоритма DES часть 1

```
1  uint64_t des(uint64_t input, uint64_t key, char mode) {
2      int i, j;
3      char row, column;
4      uint32_t C                 = 0;
5      uint32_t D                 = 0;
6      uint32_t L                 = 0;
7      uint32_t R                 = 0;
8      uint32_t s_output          = 0;
9      uint32_t f_function_res    = 0;
10     uint32_t temp              = 0;
11     uint64_t sub_key[16]       = {0};
12     uint64_t s_input           = 0;
13     uint64_t permuted_choice_1 = 0;
14     uint64_t permuted_choice_2 = 0;
15     uint64_t init_perm_res     = 0;
16     uint64_t inv_init_perm_res = 0;
17     uint64_t pre_output        = 0;
```

Листинг 3.2 – Реализация шифровального алгоритма DES часть 2

```
1     for (i = 0; i < 64; i++) {
2         init_perm_res <<= 1;
3         init_perm_res |= (input >> (64-IP[i])) & LB64_MASK;
4     }
5     L = (uint32_t) (init_perm_res >> 32) & L64_MASK;
6     R = (uint32_t) init_perm_res & L64_MASK;
7     for (i = 0; i < 56; i++) {
8         permuted_choice_1 <<= 1;
9         permuted_choice_1 |= (key >> (64-PC1[i])) & LB64_MASK;
10    }
11    C = (uint32_t) ((permuted_choice_1 >> 28) & 0x000000000fffffff);
12    D = (uint32_t) (permuted_choice_1 & 0x000000000fffffff);
13    for (i = 0; i< 16; i++) {
14        for (j = 0; j < iteration_shift[i]; j++) {
15            C = (0x0fffffff & (C << 1)) | (0x00000001 & (C >> 27));
16            D = (0x0fffffff & (D << 1)) | (0x00000001 & (D >> 27));
17        }
18        permuted_choice_2 = 0;
19        permuted_choice_2 = (((uint64_t) C) << 28) | (uint64_t) D ;
20        sub_key[i] = 0;
21        for (j = 0; j < 48; j++) {
22            sub_key[i] <<= 1;
23            sub_key[i] |= (permuted_choice_2 >> (56-PC2[j])) &
                  LB64_MASK;
24        }
25    }
26    for (i = 0; i < 16; i++) {
27        s_input = 0;
28        for (j = 0; j< 48; j++) {
29            s_input <<= 1;
30            s_input |= (uint64_t) ((R >> (32-E[j])) & LB32_MASK);
31        }
32        if (mode == 'd') {
33            s_input = s_input ^ sub_key[15-i];
34        } else {
35            s_input = s_input ^ sub_key[i];
36        }
```

Листинг 3.3 – Реализация шифровального алгоритма DES часть 3

```
        for (j = 0; j < 8; j++) {
            row = (char) (((s_input & (0x0000840000000000 >> 6*j))
               >> 42)-6*j);
            row = (row >> 4) | (row & 0x01);
            column = (char) (((s_input & (0x0000780000000000 >>
               6*j)) >> 43)-6*j);
            s_output <<= 4;
            s_output |= (uint32_t) (S[j][16*row + column] & 0x0f);
        }
        f_function_res = 0;
        for (j = 0; j < 32; j++) {
            f_function_res <<= 1;
            f_function_res |= (s_output >> (32 - P[j])) & LB32_MASK;
        }
        temp = R;
        R = L ^ f_function_res;
        L = temp;
    }
    pre_output = (((uint64_t) R) << 32) | (uint64_t) L;
    for (i = 0; i < 64; i++) {

        inv_init_perm_res <<= 1;
        inv_init_perm_res |= (pre_output >> (64-PI[i])) & LB64_MASK;

    }
    return inv_init_perm_res;
}
```

Листинг 3.4 – Реализация режима работы CFB

```
uint64_t cfb(uint64_t input, char mode) {
    uint64_t result = des(IV, key, 'e');
    if (mode == 'e') {
        IV = result ^ input;
        return IV;
    } else {
        IV = input;
        return result ^ input;
    }
}
```

3.3 Тестирование

Тестирование разработанной программы производилось следующим образом: выбирались случайные значения ключа и вектора IV, а также получалась случайная последовательность блоков для шифрования длиной n. Она зашифровывалась и расшифровывалась, проверялось совпадение полученного результата с начальными данными. Данная процедура повторялась n раз для значений n от 1 до 100.

Вывод

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

Заключение

В результате лабораторной работы был реализован в виде программы шифровальный алгоритма DES в режиме работы CFB

Был и выполнены следующие задачи:

1) изучен шифроовальный алгоритм DES и его режим работы CFB;

2) реализован шифровальный алгоритм DES в виде программы, обеспечена возможность шифрования и расшифровки файла в режиме работы CFB;

3) протестирована разработанная программа;

4) описаны и обоснованы полученные результаты в отчёте о выполненной лабораторной работе.