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»

Отчет по лабораторной работе №2

по курсу "Защита информации"

Тема Алгоритм работы DES

Студент Золотухин А. В.

Группа ИУ7-74Б

Преподаватели Чиж И.С.

Москва, 2023 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Цель данной лабораторной работы — реализовать программу шифрования симметричным алгоритмом DES.

Информацию для разных целей пытались засекречивать с помощью шифрования на протяжении всей истории человечества. Шифр — это множество обратимых преобразований открытого текста, проводимых с целью его защиты от несанкционированного использования. Одним из таких шифров является DES.

В рамках выполнения лабораторной работы необходимо решить следующие задачи:

— описать алгоритм DES;

— реализовать алгоритм с режимом шифрования CBC.

1 Аналитический раздел

1.1 Алгоритм шифрования DES

DES (Data Encryption Standard) - это симметричный шифровальный алгоритм, разработанный в 1970-х годах, который использует блочное шифрование с фиксированной длиной блока в 64 бита. Вот основные шаги и логика работы DES:

1. Начальная перестановка (Initial Permutation): Исходный текст (64 бита) проходит через начальную перестановку, где биты переставляются в определенном порядке согласно предопределенной таблице перестановок.

2. Раунды (Rounds): DES состоит из 16 раундов шифрования, каждый из которых включает несколько шагов:

— Расширение (Expansion): 32-битный входной блок расширяется до 48 бит путем перестановки и дублирования некоторых битов.

— Ключ раунда (Round Key): к 48-битному расширенному блоку применяется 48-битный ключ раунда, полученный из основного ключа DES.

— Скремблирование (Substitution): 48-битный блок проходит через S-блоки (Substitution-boxes), которые заменяют блоки по 6 бит на блоки по 4 бита с использованием заранее определенных таблиц замен.

— Перестановка (Permutation): после замены, полученный блок по 32 бита проходит через таблицу перестановки, которая перемешивает биты в блоке.

— Обработка ключа (Key Mixing): к полученному блоку применяется операция XOR с ключом раунда для обеспечения взаимодействия ключа и данных.

3. Последняя перестановка (Final Permutation): После 16 раундов, 64-битный блок проходит через последнюю перестановку, обратную начальной перестановке, чтобы получить зашифрованный текст.

Основным элементом DES является ключ, который состоит из 56 бит, и который используется для генерации ключей раунда. Ключ разбивается на две половины, и каждая половина сдвигается влево на определенное количество бит в зависимости от номера раунда. Затем, из полученных половинок формируется ключ раунда.

Таким образом, DES использует комбинацию перестановок, замен и операций XOR для шифрования данных. Эти шаги повторяются 16 раз, в каждом раунде используется уникальный ключ. Результат - зашифрованный блок данных, который без знания правильного ключа практически невозможно расшифровать.

Для DES рекомендовано несколько режимов:

— ECB (англ. electronic code book) — режим «электронной кодовой книги» (простая замена);

— CBC (англ. cipher block chaining) — режим сцепления блоков;

— CFB (англ. cipher feed back) — режим обратной связи по шифротексту;

— OFB (англ. output feed back) — режим обратной связи по выходу.

Вывод

В данном разделе был рассмотрен алгоритм симметричного шифрования DES.

2 Конструкторский раздел

В данном разделе будет представлена схема алгоритма шифрования DES.

2.1 Алгоритм шифрования DES

На рисунке 2.1 изображена обобщенная схема шифрования DES.

Рисунок 2.1 — Обобщенная схема шифрования DES

На рисунке 2.2 изображена структура алгоритма DES.

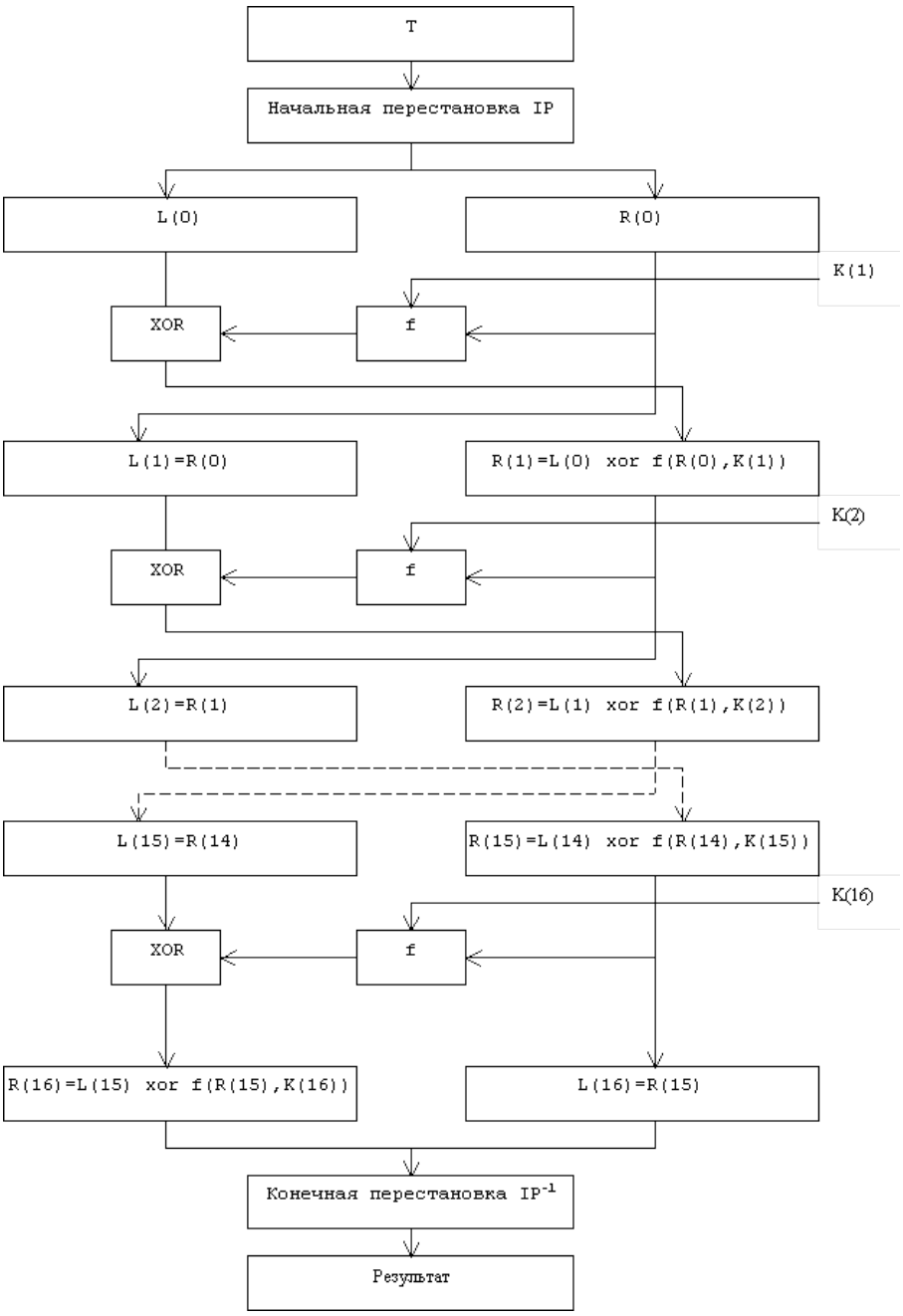

Рисунок 2.2 — Структура алгоритма DES

На рисунке 2.3 схематически изображено вычисление функции $f(R(i-1), K(i))$.

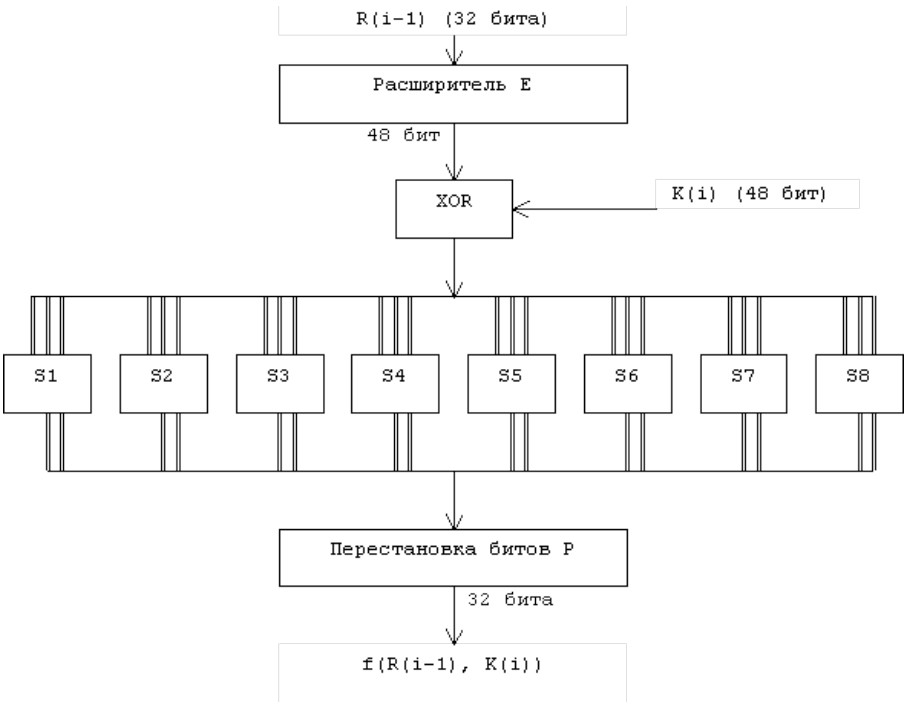

Рисунок 2.3 — Вычисление функции $f(R(i-1), K(i))$

Вывод

В данном разделе были приведены схемы алгоритма шифрования DES.

3 Технологический раздел

3.1 Средства реализации

В качестве языка программирования для реализации данной лабораторной работы использовался язык программирования C++ [1], так как он позволяет работать с файлами и массивами. В качестве среды разработки использовалась Visual Studio Code [2].

3.2 Реализация алгоритмов

В листингах 3.1–3.3 представлена реализация алгоритма шифрования DES.

Листинг 3.1 — Функция шифрования файла

```
1  int FileEncryption::cipher(string input, string output, bool mode)
2  {
3      ifstream ifile;
4      ofstream ofile;
5      ui64 buffer;
6
7      if(input.length() < 1)
8      ifile = ifstream(stdin);
9      else
10     ifile.open(input, ios::binary | ios::in | ios::ate);
11
12     if(output.length() < 1)
13     ofile = ofstream(stdout);
14     else
15     ofile.open(output, ios::binary | ios::out);
16
17     ui64 size = ifile.tellg();
18     ifile.seekg(0, ios::beg);
19
20     ui64 block = size / 8;
21     if(mode) block--;
22
23     for(ui64 i = 0; i < block; i++)
24     {
25         ifile.read((char*) &buffer, 8);
26         if (mode)
27         buffer = des.decrypt(buffer);
28         else
29         buffer = des.encrypt(buffer);
30
31         ofile.write((char*) &buffer, 8);
32     }
33
```

```
34      if(mode == false)
35      {
36          ui8 padding = 8 - (size % 8);
37
38          if (padding == 0)
39          padding  = 8;
40
41          buffer = (ui64) 0;
42          if(padding != 8)
43          ifile.read((char*) &buffer, 8 - padding);
44
45          ui8 shift = padding * 8;
46          buffer <<= shift;
47          buffer  |= (ui64) 0x0000000000000001 << (shift - 1);
48
49          buffer = des.encrypt(buffer);
50          ofile.write((char*) &buffer, 8);
51      }
52      else
53      {
54          ifile.read((char*) &buffer, 8);
55          buffer = des.decrypt(buffer);
56
57          ui8 padding = 0;
58
59          while(!(buffer & 0x00000000000000ff))
60          {
61              buffer >>= 8;
62              padding++;
63          }
64
65          buffer >>= 8;
66          padding++;
67
68          if(padding != 8)
69          ofile.write((char*) &buffer, 8 - padding);
70      }
71
72      ifile.close();
73      ofile.close();
74      return 0;
75  }
```

Листинг 3.2 — Функция шифратора

```
1  ui64 DES::des(ui64 block, bool mode)
2  {
```

```
3       block = ip(block);
4
5       ui32 L = (ui32) (block >> 32) & L64_MASK;
6       ui32 R = (ui32) (block & L64_MASK);
7
8       for (ui8 i = 0; i < 16; i++)
9       {
10          ui32 F;
11          if (mode)
12          F = f(R, sub_key[15 - i]);
13          else
14          F = f(R, sub_key[i]);
15          ui32 temp = R;
16          R = L ^ F;
17          L = temp;
18      }
19
20      block = (((ui64) R) << 32) | (ui64) L;
21      return fp(block);
22  }
```

Листинг 3.3 — Функция Фейстеля

```
1   ui32 DES::f(ui32 R, ui64 k)
2   {
3       ui64 s_input = 0;
4       for (ui8 i = 0; i < 48; i++)
5       {
6           s_input <<= 1;
7           s_input |= (ui64) ((R >> (32-EXPANSION[i])) & LB32_MASK);
8       }
9       s_input = s_input ^ k;
10      ui32 s_output = 0;
11      for (ui8 i = 0; i < 8; i++)
12      {
13          char s = (s_input >> (42 - 6 * i)) & 0x3f;
14          char row = ((s >> 4) & 0b10) | s & 1;
15          char column = (s >> 1) & 0b1111;
16
17          s_output <<= 4;
18          s_output |= (ui32) (SBOX[i][16*row + column] & 0x0f);
19      }
20
21      ui32 f_result = 0;
22      for (ui8 i = 0; i < 32; i++)
23      {
24          f_result <<= 1;
```

```
25              f_result |= (s_output >> (32 - PBOX[i])) & LB32_MASK;
26          }
27
28          return f_result;
29      }
```

3.3 Тестирование реализации алгоритма

Было проведено тестирование на следующих входных данных:

1. Входящая последовательность байтов:

D09BD0B8D0BDD0B5D0B9D0BDD18BD0B9

Зашифрованный текст:

AB3B4DA99A15DCA7C13358E9D65EA07F

Расшифрованная последовательность байтов:

D09BD0B8D0BDD0B5D0B9D0BDD18BD0B9

2. Входящая последовательность байтов:

4141414141414141

Зашифрованная последовательность байтов:

4D41E973A3BF9604

Расшифрованная последовательность байтов:

4141414141414141

Все тесты пройдены успешно.

Вывод

В данном разделе были перечислены средства разработки, с помощью которых был реализованы алгоритм шифрования DES, приведена реализация алгоритма.

ЗАКЛЮЧЕНИЕ

В результате выполнения данной лабораторной работы была достигнута цель работы: реализована программа шифрования симметричным алгоритмом DES.

Были решены все задачи — описан и реализован алгоритм шифрования DES с режимом шифрования CBC.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация языка C++ [Электронный ресурс]. — Режим доступа: `https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2013/n3690.pdf` (дата обращения: 13.11.2022).

2. Visual Studio Code [Электронный ресурс]. — Режим доступа: `https://code.visualstudio.com/docs` (дата обращения: 20.09.2022).