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

Студент Ковель Александр Денисович

Группа ИУ7-76Б

Предмет Защита информации

Студент		**Ковель А. Д.**
	подпись, дата	*фамилия, и.о.*
Преподаватель		**Чиж И. С.**
	подпись, дата	*фамилия, и.о.*

2023 г.

ВВЕДЕНИЕ

Цель лабораторной работы — разработать программу шифровальной машины «DES» [1].

Задачи лабораторной работы:

1) провести анализ работы шифровальной машина «DES»;
2) описать алгоритм шифрования;
3) релизовать описанный алгоритм.

1 Аналитическая часть

1.1 Алгоритм шифрования DES

DES (Data Encryption Standard) [1] — 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 шифрования** (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. **Завершающая 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.

Алгоритм шифрования DES может использоваться в следующих режимах.

1. **ECB** (Electronic Code Book) — режим «электронной кодовой книги» (простая замена);
2. **CBC** (Cipher Block Chaining) — режим сцепления блоков;
3. **CFB** (Cipher Feed Back) — режим обратной связи по шифротексту;
4. **OFB** (Output Feed Back) — режим обратной связи по выходу.

2 Конструкторская часть

2.1 Разработка алгоритма

На рисунке 1 представлена схема алгоритма шифрования DES.

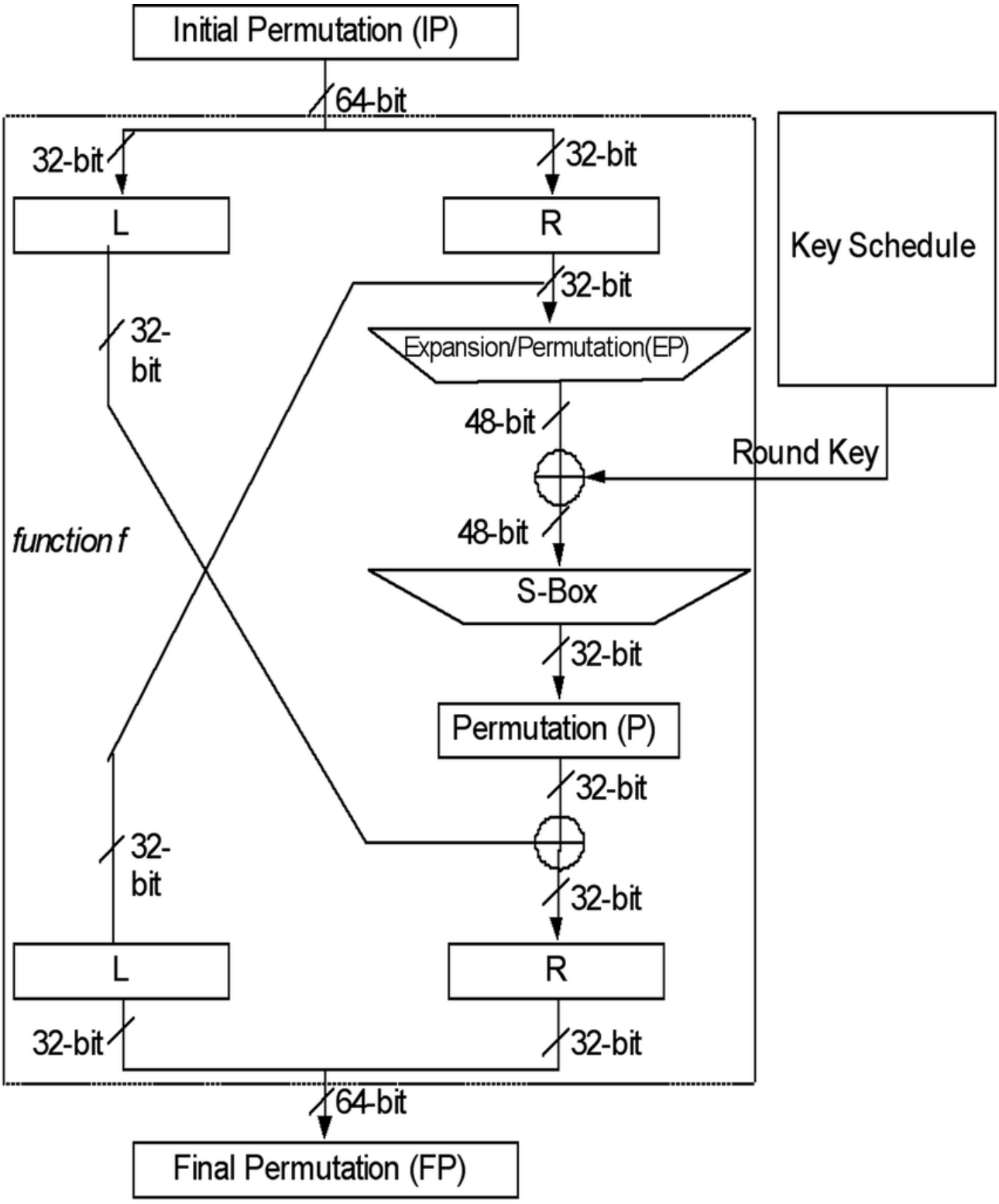

Рисунок 1 – Схемы алгоритма DES

3 Технологическая часть

3.1 Средства реализации

Для реализации ПО был выбран язык C++ [2]. В данном языке есть все требующиеся инструменты для данной лабораторной работы. В качестве среды разработки была выбрана среда VS code [3].

3.2 Реализация алгоритма

Реализация РСВС.

```
void messageBlocksToFile(std::string filenameOut,
    std::vector<std::vector<int>> messageBlocks, size_t truncateBytesCount) {
    std::vector<std::string> messageBinary;

    for (int i = 0; i < int(messageBlocks.size()) - 1; i++) {
        for (int j = 0; j < 8; ++j) {
            std::string curStr;
            for (int s = j * 8; s < messageBlocks[i].size() && s < (j + 1)
                * 8; s++) {
                curStr += std::to_string(messageBlocks[i][s]);
            }
            messageBinary.push_back(curStr);
        }
    }

    if (!messageBlocks.empty()){
        for (int j = 0; j < 8 - truncateBytesCount; ++j) {
            std::string curStr;
            std::vector<int> block = messageBlocks.back();
            for (int s = j * 8; s < block.size() && s < (j + 1) * 8; s++) {
                curStr += std::to_string(block[s]);
            }
            messageBinary.push_back(curStr);
        }
    }

    std::vector<int> messageOrd;
    for (const auto& x : messageBinary) {
        messageOrd.push_back(std::stoi(x, 0, 2));
    }
```

```
        std::vector<unsigned char> messageBytes;
        for (const auto& x : messageOrd) {
            messageBytes.push_back(static_cast<unsigned char>(x));
        }

        std::ofstream f_out(filenameOut, std::ios::binary);
        for (const auto& x : messageBytes) {
            f_out.write(reinterpret_cast<const char*>(&x), 1);
        }
        f_out.close();
    }
```

3.3 Тестовые данные

В таблице 1 приведены тесты для алгоритма шифрования DES. Применена методология черного ящика. Тесты пройдены *успешно*.

Таблица 1 – Функциональные тесты

Входная строка	Выходная строка
$ABOBA$	$BCRGJ$
$BCRGJ$	$ABOBA$
$<<>>$	$<<>>$
A	T
T	A

ЗАКЛЮЧЕНИЕ

В данной лабораторной работе:

1) проведен анализ работы шифровальной машина «DES»;
2) описан алгоритм шифрования;
3) реализован описанный алгоритм;

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Список литературы

1. DES. hhttps://www.researchgate.net/publication/220850878_A_cryptography_core_tolerant_to_DFA_fault_attacks. дата обращения: 17.09.2023.

2. Язык программирования C++. https://learn.microsoft.com/en-us/cpp/cpp/cpp-language-reference?view=msvc-170. дата обращения: 17.09.2023.

3. Vscode. https://code.visualstudio.com/. дата обращения: 17.09.2023.