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 Реализация DES

Студент Волков Г.В.

Группа ИУ7-71Б

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.

Существует множество методов шифрования. Одним из них является блочное шифрование. Блочные шифры оперируют группами бит фиксированной длины — блоками, характерный размер которых меняется в пределах 64–256 бит. Одним из самых используемых был шифр DES. Это алгоритм для симметричного шифрования, разработанный фирмой IBM. Размер блока для DES равен 64 битам. В основе алгоритма лежит сеть Фейстеля с 16 раундами и ключом, имеющим длину 56 бит. Алгоритм использует комбинацию нелинейных (S-блоки) и линейных (перестановки E, IP, IP-1) преобразований.

Цель — реализация программы шифрования симметричным алгоритмом DES с применением режима шифрования PCBC.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- изучить алгоритм работы алгоритма DES;
- изучить режим шифрования PCBC;
- реализовать в виде программы алогритм шифрования DES с применением режима шифрования PCBC;
- обеспечить шифрование и расшифровку произвольного файла с использованием разработанной программы;
- предусмотреть работу программы с пустым, однобайтовым файлом и с файлами архива (rar, zip или др.).

1 Аналитическая часть

1.1 Алгоритм DES

Data Encryption Standard (DES) — это стандарт шифрования данных, изобретенный в США в 80-х годах XX века. DES перестал быть пригодным в условиях сверхбыстрой техники и больших объемов данных из-за ограничений в 56 бит на ключ и 64 бит на данные. Однако он все еще используется.

Хотя DES признан устаревшим и не удовлетворяющим современным требованиям, он может быть использован, например, в виде 3DES, когда шифр применяется три раза подряд. Такой подход снимает ограничение в размере ключа, но блок шифруемых данных остается прежним. В свое время DES был достаточно быстрым и криптоустойчивым шифром. Сейчас это не так, а 3DES и вовсе работает втрое медленнее. Несмотря на это DES по-прежнему используется в ряде систем, но его применение в новых проектах запрещено.

Перед началом шифрования исходный текст разбивается на блоки по 64 бита. Если размер текста не кратен 64, то последний блок дополняется. Далее каждый блок проходит начальную перестановку по специальной таблице. Далее каждый блок проходит 16 раундом по схеме Фейстеля. После всех циклов осуществляется конечная перестановка и получается шифротекст.

Каждый раунд блок разбивается пополам. Правый блок и ключ раунда проходят через основную функция шифрования и делают XOR с левым блоком. Перед началом следующего раунда половинки меняются местами.

В самой функции правый блок проходит через расширяющую функцию до 48 бит и делается XOR с ключом раунда. Результат разбивается на 8 подблоков по 6 бит. Каждый подблок преобразовывается с помощью своей таблицы подстановки в новый 4 битовый блок. Для подстановки из подблока выделяются 2 крайних и 4 серединных бита. Крайние биты склеиваются и получается номер строки в соответствующем S-блоке, а серединные номером столбца. Далее осуществляется перестановка. Схема работы функции представлена на рисунке 1.1.

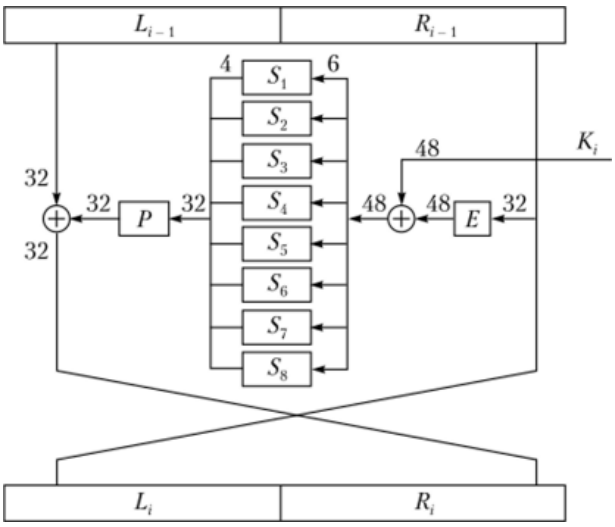

Рисунок 1.1 – Схема работы функции f

Для работы DES необходимо расширить 1 64 битный ключ до 16 48 битных ключей. Сначала ключ проходит сжимающую перестановку до 56 бит, выкидывается каждый 8 бит. Ключ шифра DES имеет длину 64 бита, но каждый восьмой предназначается лишь для контроля чётности. Далее он разбивается пополам. В каждом раунде половинки сначала циклически сдвигаются, затем объединятся и проходят сжимающую перестановку. Схема работы алгоритма генерации ключей представлена на рисунке 1.2.

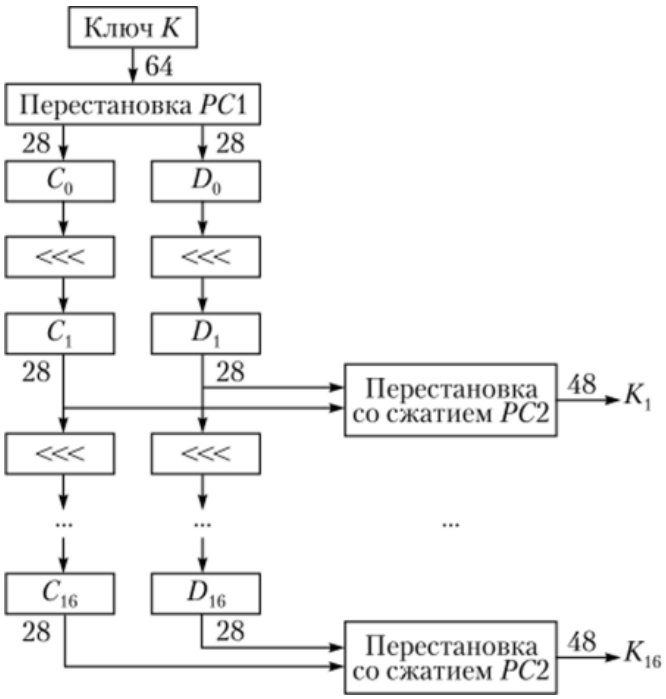

Рисунок 1.2 – Схема алгоритма генерации ключей

Общая схема шифрования алгоритма DES представлена на рисунке 1.3.

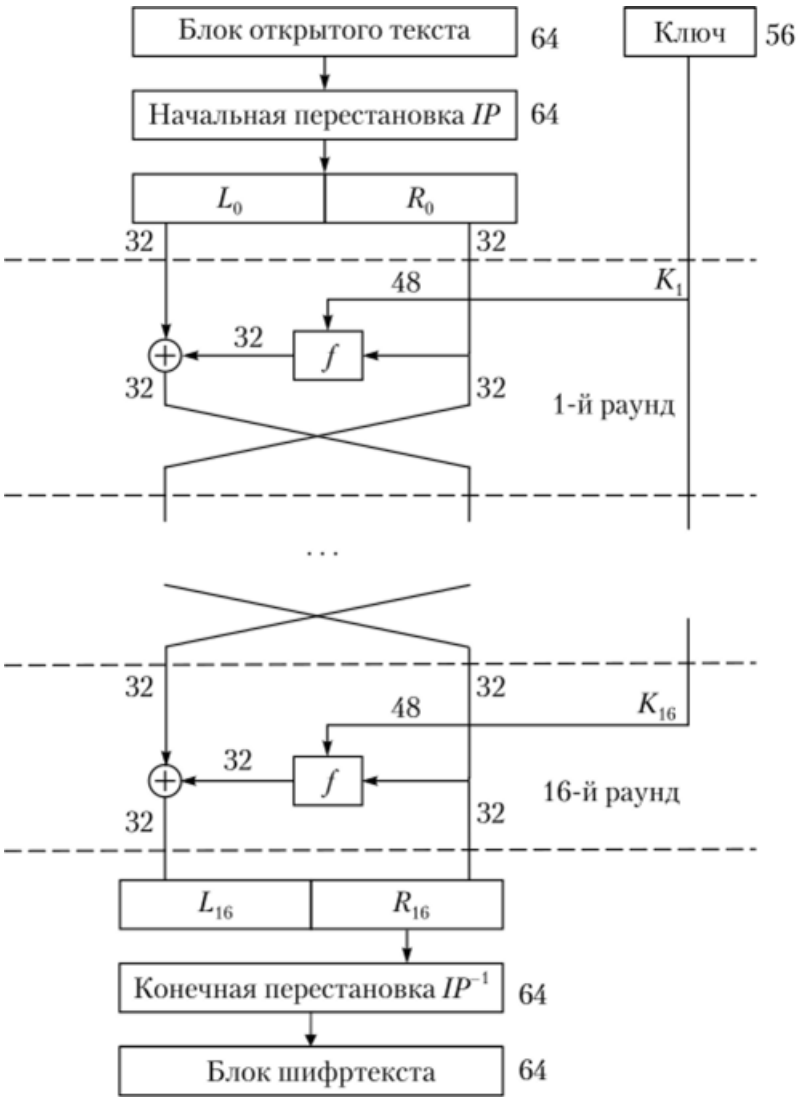

Рисунок 1.3 – Схема шифрования

Основной недостаток DES – короткий ключ. Для решения этой проблемы был создан алгоритм 3DES с ключом длинной 192 бита. По сути является трёхкратным применением DES.

1.2 Режим шифрования PCBC

Режим распространяющегося сцепления блоков шифра. Этот режим похож на CBC за исключением того, что предыдущий блок открытого текста и предыдущий блок шифротекста подвергается операции XOR с текущим блоком открытого текста перед шифрованием или после него. Был разработан для того, чтобы небольшие изменения в зашифрованном тексте распространялись бесконечно как при расшифровке, так и при шифровании. Схема

применения режима PCBC представлена на рисунках 1.4 и 1.5.

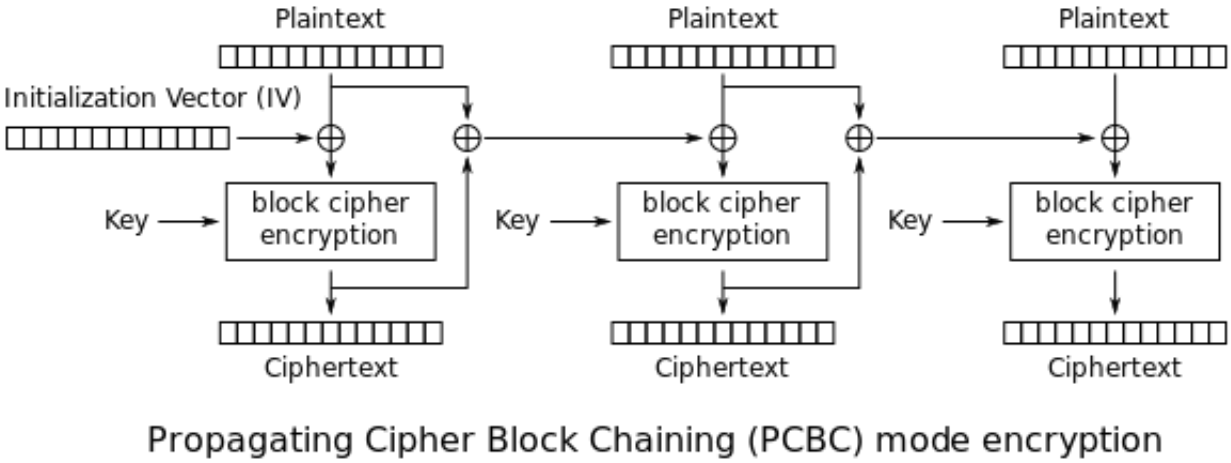

Рисунок 1.4 – Шифрование с PCBC

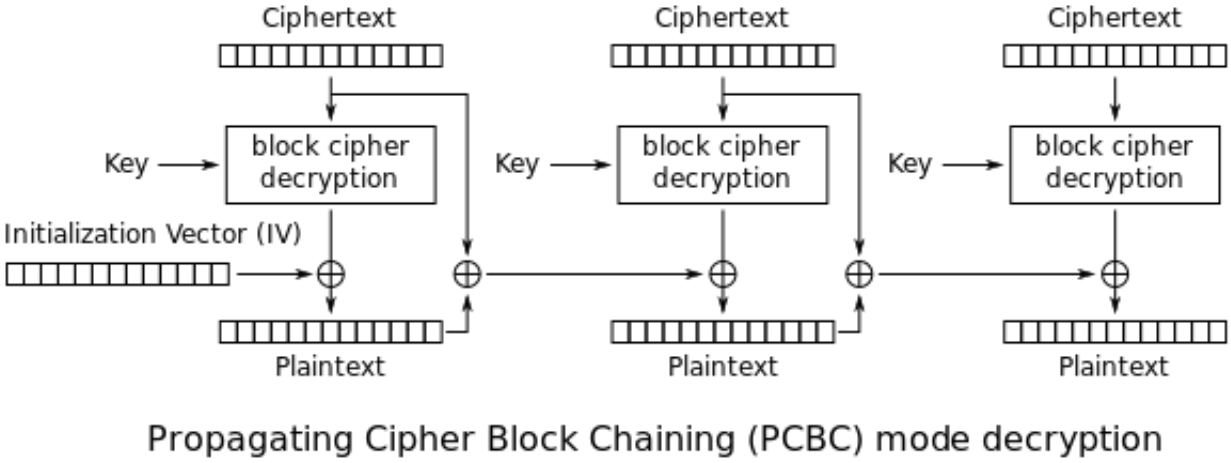

Рисунок 1.5 – Дешифрование с PCBC

Вывод

В данном разделе был рассмотрен алгоритм шифрования DES с использованием режима шифрования PCBC.

2 Конструкторская часть

В этом разделе будут представлены сведенья о модулях программы.

2.1 Сведения о модулях программы

Программа состоит из двух модулей:

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

2) $DES.h$ — файл, содержащий описание функций шифрования.

3) $DES.c$ — файл, содержащий определение функций шифрования.

3 Технологическая часть

В данном разделе будут рассмотрены средства реализации, а также представлены листинги реализаций алгоритма шифрования DES.

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 и 3.2 представлена реализация алгоритма шифрования DES с использованием режима PCBC.

Листинг 3.1 – Шифрование DES

```
1  size_t encode(uint8_t *to, uint8_t *keys8b, uint8_t *from, size_t
       length) {
2      length = length % 8 == 0 ? length : length + (8 - (length % 8));
3
4      uint64_t keys48b[16] = {0};
5      uint32_t N1, N2;
6
7      key_expansion(join_8bits_to_64bits(keys8b),keys48b);
8
9      uint64_t lastCypher = 1, lastOpen = 1;
10     for (size_t i = 0; i < length; i += 8) {
11         uint64_t open = join_8bits_to_64bits(from + i);
12
13         uint64_t tmp = lastOpen;
14         lastOpen = open;
15
16         open ^= lastCypher;
17         open ^= tmp;
18
```

```
19          split_64bits_to_32bits(initial_permutation(open), &N1, &N2);
20          feistel_encode(&N1, &N2, keys48b);
21
22          lastCypher = final_permutation(join_32bits_to_64bits(N1,
              N2));
23          split_64bits_to_8bits(lastCypher, (to + i));
24      }
25
26      return length;
27  }
```

Листинг 3.2 – Дешифрование DES

```
1  size_t decode(uint8_t *to, uint8_t *keys8b, uint8_t *from, size_t
       length) {
2      length = length % 8 == 0 ? length : length + (8 - (length % 8));
3
4      uint64_t keys48b[16] = {0};
5      uint32_t N1, N2;
6
7      key_expansion(join_8bits_to_64bits(keys8b),keys48b);
8
9      uint64_t lastCypher = 1, lastOpen = 1;
10     for (size_t i = 0; i < length; i += 8) {
11         uint64_t cypher = join_8bits_to_64bits(from + i);
12
13         split_64bits_to_32bits(initial_permutation(cypher),&N1,
               &N2);
14         feistel_decode(&N1, &N2, keys48b);
15
16         uint64_t open = final_permutation(join_32bits_to_64bits(N1,
               N2));
17         open ^= lastOpen;
18         open ^= lastCypher;
19
20         lastOpen = open;
21         lastCypher = cypher;
22
23         split_64bits_to_8bits(lastOpen,(to + i));
24     }
25
26     return length;
```

```
27 }
```

3.3 Тестирование

Для тестирование написанной программы файл шифровался и дешифровался и сравнивалось их содержимое.

Таблица 3.1 – Функциональные тесты

Входной файл	Ожидаемый результат	Результат
1_in.txt (9 bytes) «encode me»	1_out.txt (9 bytes) «encode me»	1_out.txt (9 bytes) «encode me»
2_in.txt (1 bytes) «a»	2_out.txt (1 bytes) «a»	2_out.txt (1 bytes) «a»
3_in.txt (0 bytes) «»	3_out.txt (0 bytes) «»	3_out.txt (0 bytes) «»
4_in.tar.gz (21 Kb) test.png	4_out.tar.gz (21 Kb) test.png	4_out.tar.gz (21 Kb) test.png

Вывод

Были представлен листинг реализации алгоритма работы энигмы. Также в данном разделе была приведена информация о выбранных средствах для разработки алгоритмов.

Заключение

В результате лабораторной работы были изучены принципы работы алгоритма DES и режима PCBC, была написана его программная реализация.

Были решены следующие задачи:

— изучен алгоритм работы алгоритма DES;

— изучен режим шифрования PCBC;

— реализован в виде программы алогритм шифрования DES с применением режима шифрования PCBC;

— обеспечено шифрование и расшифровка произвольного файла с использованием разработанной программы;

— предусмотрена работа программы с пустым, однобайтовым файлом и с файлами архива (rar, zip или др.).