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, искусственный интеллект и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ

по лабораторной работе № 2

по курсу «Защита информации»

на тему: «Программная реализация алгоритма DES с применением режима шифрования PCBC»

Студент ИУ7-73Б _____ Марченко В.
(Группа) (Подпись, дата) (И. О. Фамилия)

Преподаватель _____ Чиж И. С.
(Подпись, дата) (И. О. Фамилия)

2023 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

DES (англ. Data Encryption Standard) — алгоритм для симметричного шифрования, разработанный фирмой IBM и утвержденный правительством США в 1977 году как официальный стандарт (FIPS 46-3). Размер блока для DES равен 64 битам. В основе алгоритма лежит сеть Фейстеля с 16 циклами (раундами) и ключом, имеющим длину 56 бит. Алгоритм использует комбинацию нелинейных (S-блоки) и линейных (перестановки E, IP, IP^{-1}) преобразований [1].

Для DES рекомендовано несколько режимов шифрования [1]:

1) ECB (англ. electronic code book) — режим «электронной кодовой книги»;
2) CBC (англ. cipher block chaining) — режим сцепления блоков;
3) PCBC (англ. propagating cipher block chaining) — режим распространяющегося сцепления блоков шифра [2];
4) CFB (англ. cipher feed back) — режим обратной связи по шифротексту;
5) OFB (англ. output feed back) — режим обратной связи по выходу.

Прямым развитием DES в настоящее время является алгоритм Triple DES (3DES). В 3DES шифрование/расшифровка выполняются путем троекратного выполнения алгоритма DES [1].

Целью данной лабораторной работы является программная реализация алгоритма шифрования DES с применением режима шифрования PCBC.

Задачи лабораторной работы:

1) изучить принцип работы алгоритма DES;
2) изучить принцип работы режима PCBC;
2) разработать программное обеспечение для шифрования и расшифровки файлов с применением PCBC;
4) протестировать разработанное программное обеспечение.

1 Алгоритм шифрования DES

DES работает с битами (двоичными числами). Алгоритм шифрует блоки по 64 бита. Для шифрования DES использует ключи, длина которых также составляет 64 бита. Однако в алгоритме DES игнорируется каждый восьмой бит ключа, поэтому эффективный размер ключа составляет 56 бит. Но в любом случае 64 бита — это число, вокруг которого организован DES [3].

Так как алгоритм шифрует блоки по 64 бита, длина открытого текста должна быть кратна 8 байтам. Часто тексты не обладают таким свойством, поэтому в качестве решения данной проблемы при шифровании можно дополнить открытый текст необходимым количеством нулевых байтов [3].

DES — это блочный шифр, то есть он работает с блоками открытого текста заданного размера (64 бита) и возвращает блоки зашифрованного текста того же размера. Таким образом, DES приводит к перестановке среди 2^{64} возможных комбинаций 64-х бит. Каждый блок из 64-х бит делится на два блока по 32 бита каждый, левый полублок L и правый полублок R [3].

DES работает с 64-битными блоками, используя ключи длиной 56 бит. Ключи фактически хранятся в виде последовательности 64-х бит, но каждый восьмой бит ключа не используется (т. е. биты с порядковыми номерами 8, 16, 24, 32, 40, 48, 56 и 64) [3].

Алгоритм DES состоит из двух основных шагов: создание 16-и ключей по 48 бит и непросредственно шифрование блока открытого текста [3].

Создание 16-и ключей. 64-битный ключ уменьшается до 56-битного с помощью таблицы перестановки PC-1 размером 8×7. Т. е., как было сказано выше, каждый 8-й бит ключа отбрасывается [3].

Далее 56-битный ключ разбивается на две части по 28 бит. Для получения следующего ключа используется предыдущий. В зависимости от раунда шифрования ключ циклически сдвигается на 1 или 2 позиции влево (то есть значения старших битов не теряются, а записываются в младшие). Во всех раундах, кроме 1, 2, 9 и 16, сдвиг происходит на две позиции. После сдвига в каждом раунде ключи соединяются и с помощью таблицы PC-2 размером 8×6 уменьшаются до 48-и бит [3].

Шифрование блока открытого текста. Перед началом процесса шифрования выполняется начальная перестановка с помощью таблицы IP размером 8×8. Затем блок делится на левую половину L_0 из 32 бит и

правую половину R_0 такой же длины. Затем выполняется 16 итераций с использованием функции f, которая работает с двумя блоками — блоком данных из 32-х бит и ключом K_n из 48-и бит — для создания блока из 32-х бит. Для вычисления L_0 и R_0 используются следующие формулы [3]:

$$L_n = R_{n-1}, \tag{1.1}$$

$$R_n = L_{n-1} \oplus f(R_{n-1}, K_n). \tag{1.2}$$

Чтобы вычислить f, сначала каждый блок R_{n-1} расширяется с 32-х бит до 48-и. Это делается с помощью таблицы E размером 8×6, которая повторяет некоторые биты в R_{n-1}. Затем нужно выполнить операцию XOR для 48-битного блока и 48-битного ключа. Теперь есть 8 групп по 6 бит. Нкжно использовать их как индексы в таблицах, называемых «S-блоками». Каждая группа из шести бит даст индекс в отдельном S-блоке. По этому индексу будет находиться 4-битное число. Это 4-битное число заменит исходные 6 бит. Конечным результатом является то, что восемь групп по 6 бит преобразуются в 8 групп по 4 бита (4-битные выходные данные из S-блоков), всего 32 бита [3].

Далее следует перестановка P, которая определена таблицей размером 8×4. P дает 32-битный выход из 32-битного входа путем перестановки бит входного блока [3].

С помощью вычисленных L_n и R_n можно перейти к следующему раунду шифрования и вычислению L_{n+1} и R_{n+1} по приведенным выше формулам [3].

В 16-м раунде нужно объединить оба блока текста $R_{16}L_16$. Затем следует финальная переставновка с помощью таблицы IP^{-1} размером 8×8 [3].

Таким образом, 64-битный текст открытого текста был зашифрован. Данный алгоритм повторяется для всех блоков открытого текста.

Расшифровка — это просто операция, обратная шифрованию, выполняющая те же шаги, что и при шифровании, но с обратным порядком применения ключей [3].

На рисунке 1.1 показана схема работы алгоритма DES.

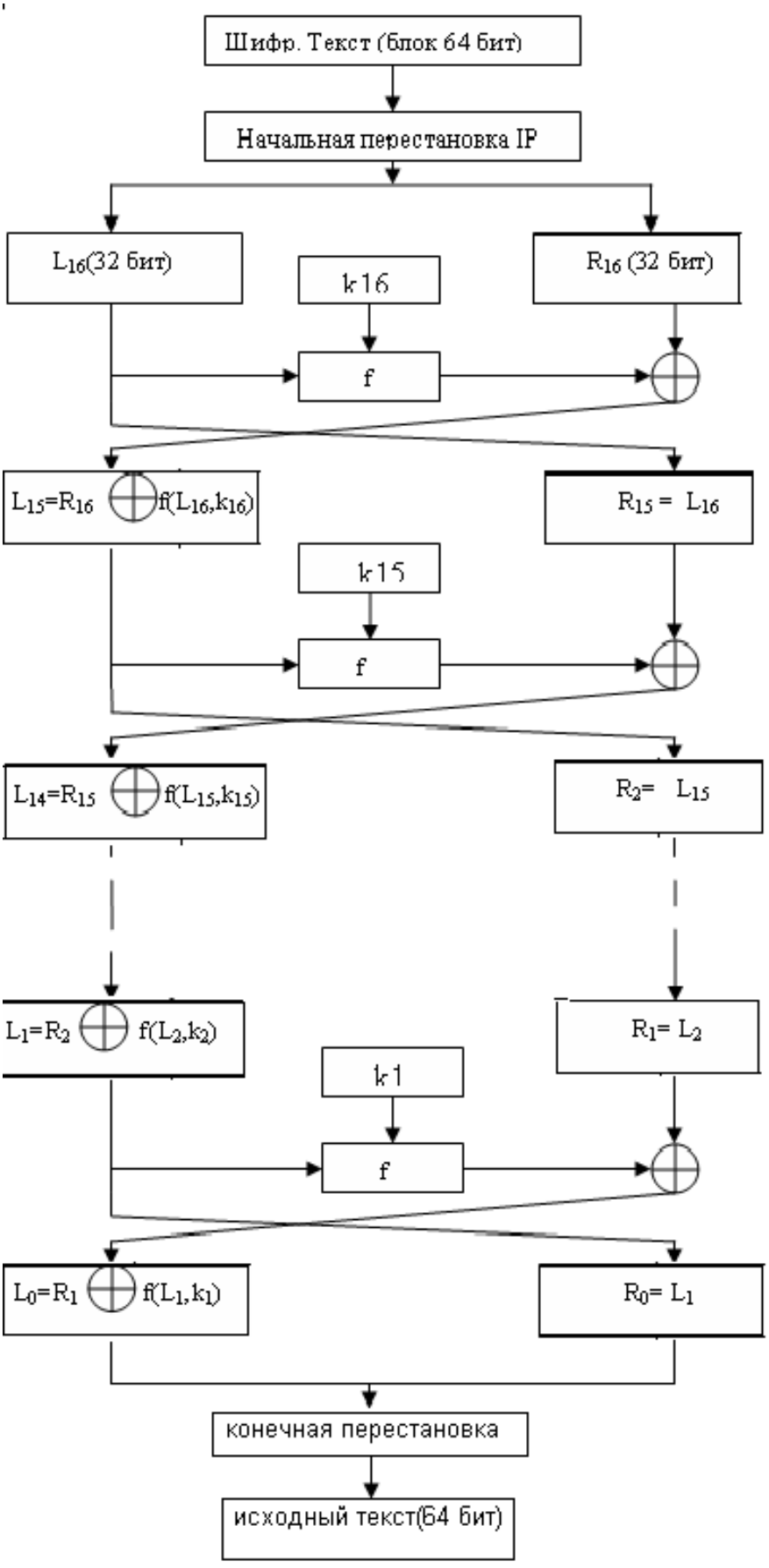

Рисунок 1.1 – Схема работы алгоритма DES [1]

2 Алгоритм режима шифрования PCBC

Недостатки режима CBC привели к созданию усовершенствованного режима распространяющегося сцепления блоков шифра. Естественно, 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 [2].

Режим шифрования PCBC применяется в протоколе Kerberos 4 версии и позволяет обнаруживать ошибки. Данный режим шифрования не является федеральным или международным стандартом. Режим PCBC — вариант режима CBC, обладающий специфическим свойством — ошибка шифротекста приводит к неправильному расшифрованию всех последующих блоков [2].

Конечно, этот режим не лишен недостатков. Так перестановка двух блоков шифротекста приводит к неправильной расшифровке двух соответствующих блоков открытого текста, но из-за XOR над открытым текстом и шифротекстом дальнейшие ошибки компенсируются. Поэтому, если при проверке целостности проверяются только несколько последних блоков расшифрованного текста, можно получить частично испорченное сообщение [2].

На рисунке 2.1 показана схема работы режима шифрования PCBC.

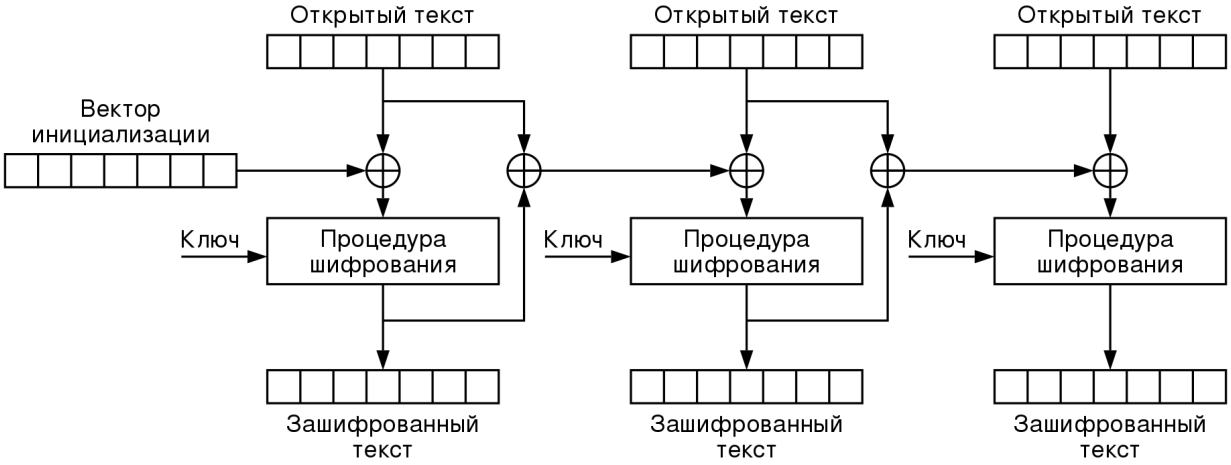

Рисунок 2.1 – Схема работы режима шифрования PCBC [2]

Вектор инициализации — 64-битная случайно сгенерированная последовательность.

На рисунке 2.2 показана схема работы режима расшифровки PCBC.

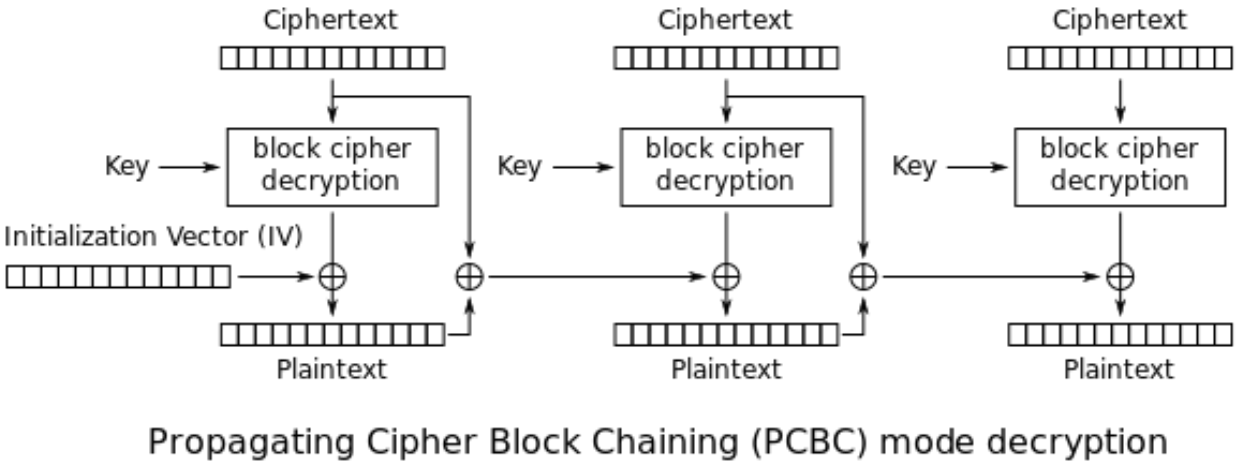

Рисунок 2.2 – Схема работы режима расшифровки PCBC [2]

3 Требования к входным данным

Программа принимает три аргумента командной строки. Первый аргумент — путь к файлу, который содержит открытый текст. Второй аргумент — путь к файлу, в который будет записан зашифрованный текст. Третий аргумент — путь к файлу, в который будет записан расшифрованный текст.

При наличии ошибок в аргументах командной строки или при передаче на вход программе пустого файла программа выдаст сообщение об ошибке и завершится.

В каталоге cfg есть текстовые файлы с конфигурациями таблиц.

Программное обеспечение для шифрования файлов с помощью алгоритма DES было написано на языке программирования C.

Программа может шифровать любые файлы: .txt, .png, .jpg, .rar и т. п. Максимальный размер файла — 100 КБ.

4 Тестирование программного обеспечения

В таблице 4.1 приведены тесты для проверки корректности работы реализованного программного обеспечения.

Таблица 4.1 – Тесты

Описание	**Открытый текст**	**Результат шифрования**
Пустой входной файл		Error: empty input file.
Кол-во аргументов командной строки не равно трем		Error: program requires 3 filenames.
Один байт	a	a163 75a3 210a 6bf6 (в hex виде)
Обычный открытый текст	Hello world!	523e 9c00 c0c5 3bd0 e6ca 82c5 6c5e be88 (в hex виде)

Помимо приведенных выше тестов были зашифрованы и расшифрованы архивы с текстовыми файлами и фотографиями. Все тесты пройдены успешно.

ЗАКЛЮЧЕНИЕ

В результате выполнения данной лабораторной работы был реализован алгоритм шифрования DES с применением режима PCBC.

Были выполнены следующие задачи:

1) изучен принцип работы алгоритма DES;

2) изучен принцип работы режима PCBC;

2) разработано программное обеспечение для шифрования и расшифровки файлов с применением PCBC;

4) протестировано разработанное программное обеспечение.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Википедия*. DES. — 2023. — (Дата обращения: 12.10.2023). https://ru.wikipedia.org/wiki/DES/.
2. *Википедия*. Режим шифрования. — 2023. — (Дата обращения: 12.10.2023). https://ru.wikipedia.org/wiki/PrPҳPцPчPё_CЇPчChCЃP«PўPѳP,,PчCS/.
3. *Grabbe J. O.* The DES Algorithm Illustrated. — (Дата обращения: 12.10.2023). https://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm/.