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 №3 по курсу «Защита информации»

Тема Шифровальный алгоритм AES

Студент Авдейкина В. П.

Группа ИУ7-76Б

Оценка (баллы)

Преподаватели Чиж И. С.

Москва — 2024 г.

Введение

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

Шифровальная алгоритм AES — алгоритм, разработанный в 2001 году Национальным универитетом стандатов и технологий США и пришедший на смену алгоритму DES.

Целью данной работы является реализация в виде программы на языке программирования C или C++ шифровального алгоритма AES в режиме работы PCBC — режима параллельного сцепления блоков шифра.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1) изучить шифроовальный алгоритм AES и его режим работы PCBC;

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

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 AES, а также его работа в режиме PCBC.

1.1 Алгоритм AES

Шифровальная алгоритм AES (англ. *Advanced Encryption Standart* — AES) — симметричный блочнный шифровальный алгоритм, разработанный в 2001 году Национальный институтом стандартов и технологий США. Он использует блочное шифрование, длина блока фиксирована и равна 128 битам, длина ключа 128, 192 либо же 256 бит. Он состоит раундов шифрования,количество которых зависит от длины ключа: 10 раундов для ключа размером 128 бит, 12 раундов для ключа размером 192 бита и 14 раундов для ключа размером 256 бит.

Прежде чем перейти к раундам шифрования, происходит генерация ключей раунда (раундовых ключей) из исходного ключа, Рассмотрим, как это происходит.

1.1.1 Получение ключей раунда

Определим фунцию g, изменяющую четырёхбайтовое слово так, как указано на рисунке 1.1.

Ключей раундов k_i необходимо на 1 больше, чем количество раундов, т.е. 11 ключей раундов для основногоключа длиной 128 бит, 13 ключей раунда для основного ключа длиной 192 бита и 15 ключей раунда для основного ключа длинной 256 бит.

Алгоритм получения ключа раунда из исходного ключа преставлен в виде схемы алгоритма на рисунке 1.2.

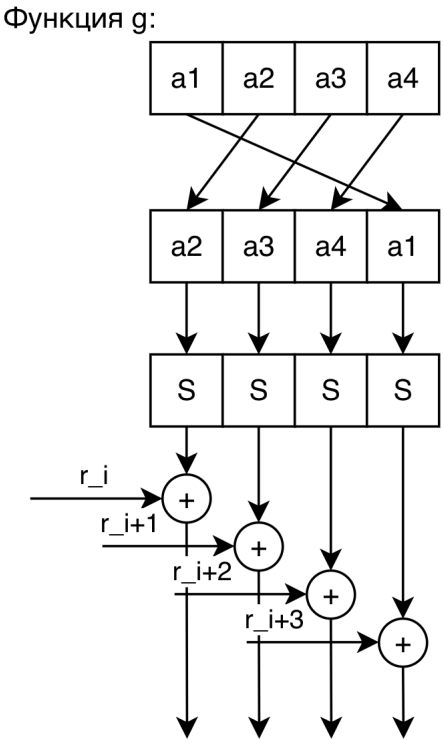

Рисунок 1.1 – Схема функции g

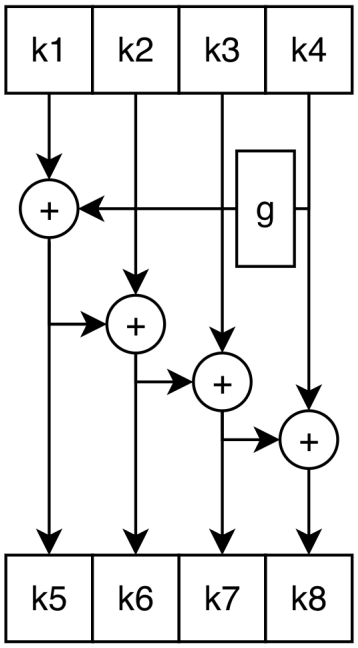

Рисунок 1.2 – Схема функции g

1.1.2 Раунд шифрования

Раунд шифрования состоит из 4 следующих этапов

1) замена (англ. *confussion*);

2) процедура перестановки строк (англ. *row-row mix procedure* — RR);

3) процедура перестановки столбцов (англ. *row-columns mix* — RC);

4) смешивание ключа (англ. *key mixing* — KM).

Замена обеспечивает нелинейность алгоритма шифрования, обрабатываая каждый байт состояния, производя нелинейную замену байт с использованием таблицы замен.

Процедура перестановки строк представляет из себя циклический сдвиг строки ссостояний на количество байт, зависящее от номера строки.

Процедура перестановки столбцов 4 байта каждого столбца смешиваются с использовоанием обратимой линейной трансформации. На последнем раунду эта процедура не выполняется.

Смешивание ключа представляет из себя операцию XOR с ключом раунда, полученным заранее.

1.2 Режимы работы алгоритма AES

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

Для AES рекомендованы следующие режимы работы:

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

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

3) режим параллельного 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 параллельного сцепления блоков

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 PCBC вектор исполнения IV подвергается операции XOR с фрагментом открытого текста, результат операции шифруется при помощи алгоритма AES. Полученное значение является фрагментом шифротекста. После этого оно подвергается операции XOR с фрагментом открытого текста, рузельтат операции становится новым значением вектора IV.

Если происходит расшифровка, фрагмент шифротекста расшифровывается при помощи алгоритма AES,после чего подвергается операции XOR с вектором IV. Полученное значение является фрагментом открытого текста. Оно подвергается операции XOR с фрагментом шифротекста, результат операции становится новым значением вектора IV.

Вывод

В данном разделе был рассмотрен шифровальный алгоритм AES, его составляющие и режимы работы, а также режим параллельного сцепления блоков (PCBC).

2 Конструкторская часть

В этом разделе будут представлены описания модулей программы, а также схема алгоритма шифроваания AES.

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) *aes.c* — файл, содержайший реализацию алгоритма шифрования AES;

4) *pcbc.c* — файл, содержащий реализацию режима работы PCBC.

2.2 Разработка алгоритмов

На рисунках 2.1–2.3 представлены схемы алгоритма AES, раунда AES, а также режима работы PCBC при зашифровке и расшифровке.

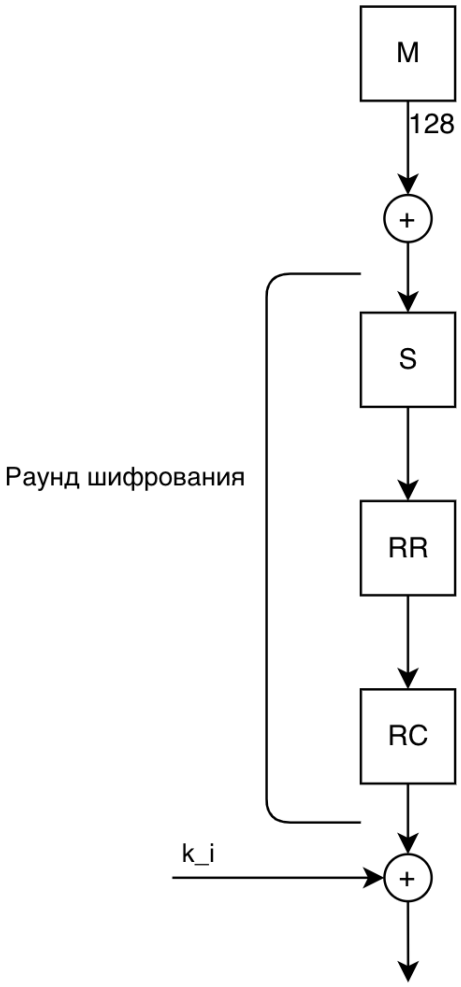

Рисунок 2.1 – Схема шифровального алгоритма AES

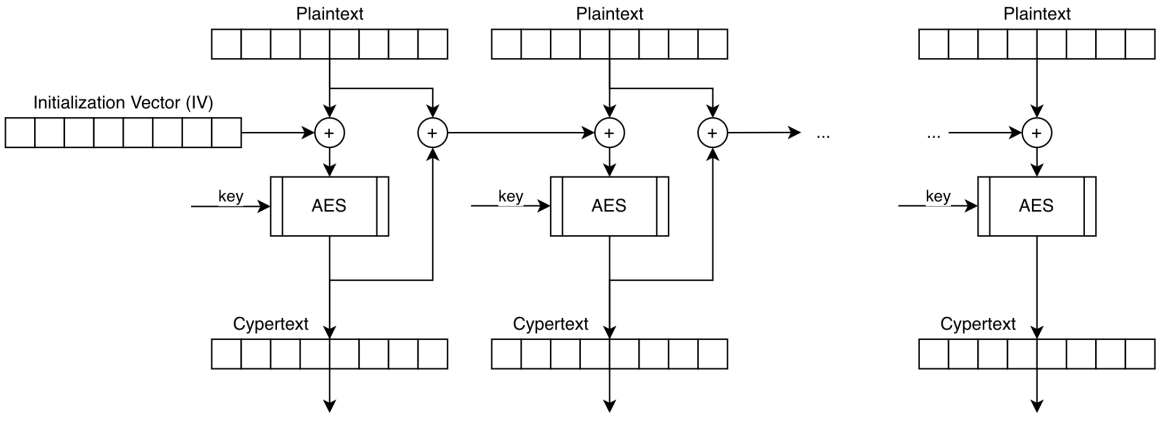

Рисунок 2.2 – Схема алгоритма режимы работы PCBC при зашифровке

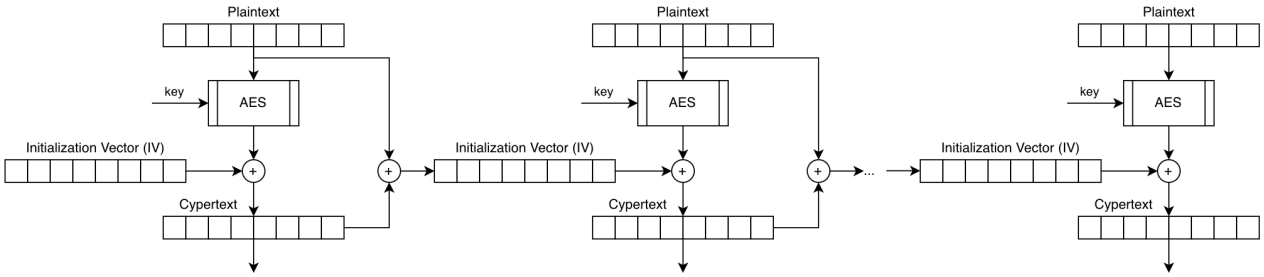

Рисунок 2.3 – Схема алгоритма режимы работы PCBC при расшифровке

Вывод

В данном разделе были представлены сведения о модулях программы, а также схемы алгоритмов, которые нужно реализовать: алгоритма AES, а также режима работы PCBC с зашифровкой и расшифровкой.

3 Технологическая часть

В данном разделе будут рассмотрены средства реализации, а также представлены листинги реализации шифровального алгоритма AES и режима работы PCBC, 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 3.1–3.2 представлена реализация шифровального алгоритма AES, на листинге 3.3 — реализация режима работы PCBC.

Листинг 3.1 – Реализация шифровального алгоритма AES

```
1  void EncryptAES128(const byte *msg, const byte *key, byte *c) {
2      int i;
3      byte keys[176];
4      expand_key128(key,keys);
5      memcpy(c, msg, 16);
6      xor_round_key(c,keys,0);
7      for(i=0; i<9; i++) {
8          sub_bytes(c,16);
9          shift_rows(c);
10         mix_cols(c);
11         xor_round_key(c, keys, i+1);
12     }
13     sub_bytes(c,16);
14     shift_rows(c);
15     xor_round_key(c, keys, 10);
16 }
```

Листинг 3.2 – Реализация шифровального алгоритма AES расшифровка

```
void DecryptAES128(const byte *c, const byte *key, byte *m) {
    int i;

    byte keys[176];
    expand_key128(key,keys);

    memcpy(m,c,16);
    xor_round_key(m,keys,10);
    shift_rows_inv(m);
    sub_bytes_inv(m, 16);

    for (i=0; i<9; i++) {
        xor_round_key(m,keys,9-i);
        mix_cols_inv(m);
        shift_rows_inv(m);
        sub_bytes_inv(m, 16);
    }
    xor_round_key(m, keys, 0);
}
```

Листинг 3.3 – Реализация режима работы PCBC

```
void pcbc(byte input128[], byte output128[], byte mode) {
    if (mode == 'e')
    {
        byte to_cypher[16] = {0};
        for (int i = 0; i < 16; i++)
            to_cypher[i] = IV[i] ^ input128[i];

        EncryptAES128(to_cypher, key, output128);
        for (int i = 0; i < 16; i++)
            IV[i] = input128[i] ^ output128[i];
    }
    else
    {
        byte almost_decyphered[16] = {0};
        DecryptAES128(input128, key, almost_decyphered);
        for (int i = 0; i < 16; i++)
        {
            output128[i] = IV[i] ^ almost_decyphered[i];
            IV[i] = input128[i] ^ output128[i];
        }
    }
}
```

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.

Таблица 3.1 – Функциональные тесты

Длина, байты	Шифруемое значение	Результат работы
8	12345678	Сообщение об ошибке
16	1234567812345678	cad29cf5b295a4bf 5905026c48d83c5
32	1234567812345678 1234567812345678	cad29cf5b295a4bf 590d5026c48d83c5 9ab00f0ae0135012 710c4ba8595b138c

Вывод

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

Заключение

В результате лабораторной работы был реализован в виде программы шифровальный алгоритма AES в режиме работы PCBC

Был и выполнены следующие задачи:

1) изучен шифроовальный алгоритм AES и его режим работы PCBC;

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

3) протестирована разработанная программа;

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