



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 ИУ «Информатика и системы управления»

КАФЕДРА ИУ-7 «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3
по дисциплине «Защита информации»
«Алгоритм шифрования AES»

Студент группы ИУ7-72Б

(Подпись, дата)

Х. М. Фам
(И.О. Фамилия)

Руководитель

(Подпись, дата)

И. С. Чиж
(И.О. Фамилия)

2024 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Аналитическая часть	5
1.1 Алгоритм AES	5
1.1.1 Получение ключей раунда	5
1.1.2 Раунд шифрования	7
1.2 Режимы работы алгоритма OFB	8
2 Конструкторская часть	10
2.1 Разработка алгоритмов	10
3 Технологическая часть	11
3.1 Средства реализации	11
3.2 Реализация алгоритма	11
3.3 Тестирование	13
Заключение	15
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	16

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 университетом стандартов 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 CFB — режима параллельного сцепления блоков шифра.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- 1) изучить шифровальный алгоритм AES и его режим работы CFB;
- 2) реализовать шифровальный алгоритм AES в виде программы, обеспечив возможности шифрования и расшифровки файла в режиме работы CFB;
- 3) протестировать разработанную программу, показать, что удаётся дешифровать 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 в режиме шифрования CFB.

1.1 Алгоритм AES

Шифровальный алгоритм AES (англ. *Advanced Encryption Standard* — AES) — симметричный блочный шифровальный алгоритм, разработанный в 2001 году Национальным 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 функцию 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 основного ключа длиной 128 бит, 13 ключей раунда для основного ключа длиной 192 бита и 15 ключей раунда для основного ключа длиной 256 бит.

Функция g:

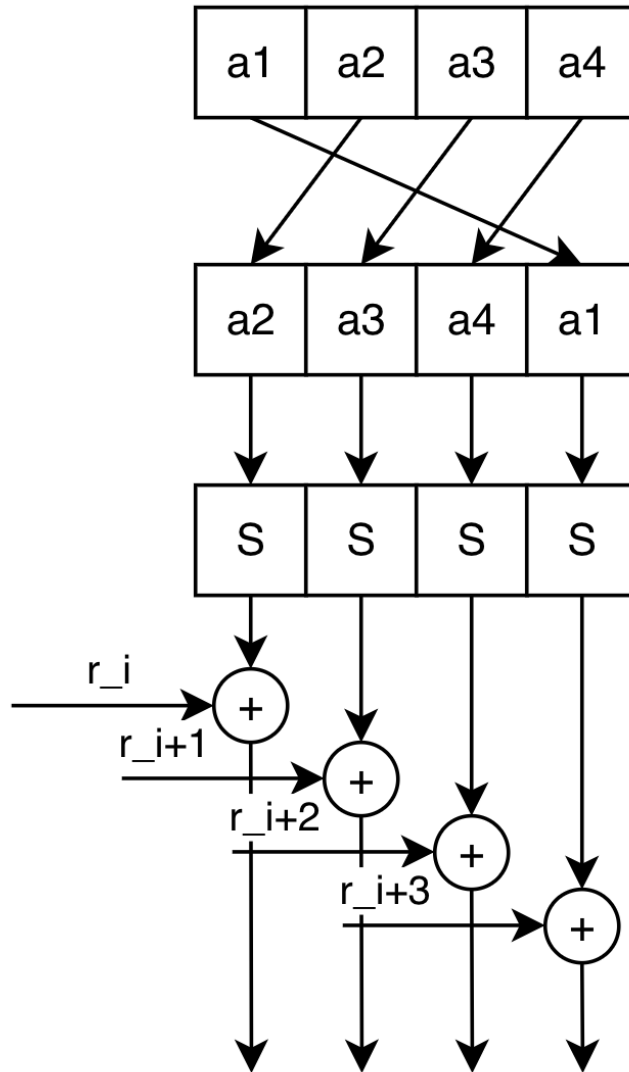


Рисунок 1.1 – Схема функции g

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

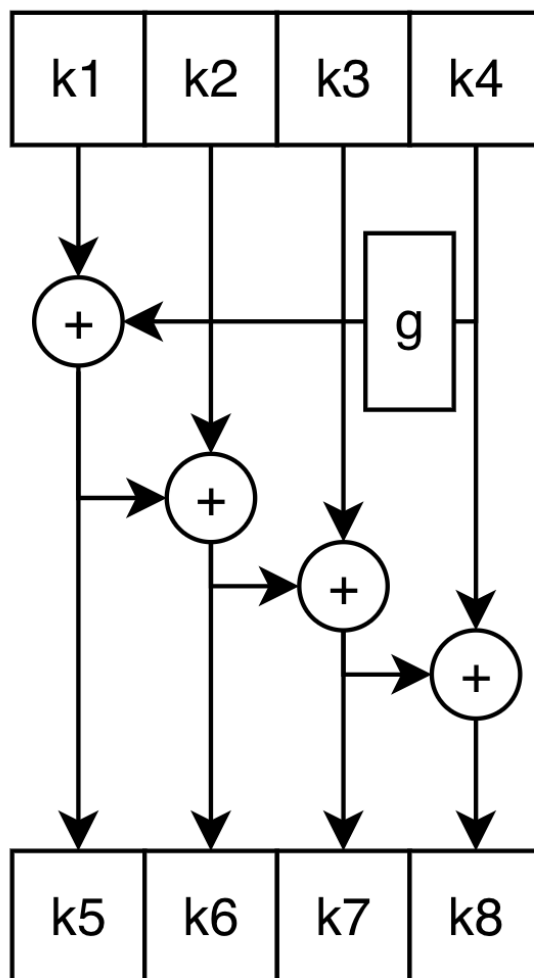


Рисунок 1.2 – Схема функции g

1.1.2 Раунд шифрования

Раунд шифрования состоит из 5 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, обрабатывая каждый байт состояния, производя нелинейную замену байт с использованием таблицы замен.

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

Процедура перестановки столбцов 4 байта каждого столбца смешиваются с использованием 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 OFB

Алгоритм CFB схематично представлен на рисунке 1.3. Суть алгоритма заключается в том, что изначально берется блок из 128 битов 0 , который называется синхропосылкой. Вектор инициализации (или результаты прошлого XOR) шифруется алгоритмом AES или DES (в нашем случае AES). Затем результат суммируется по модулю 2 с блоком 0 и при этом результат используется для шифрования следующего блока. Таким образом каждый блок суммируется с результатом шифрования предыдущего блока.

Особенностью данного режима является распространение ошибки на весь последующий текст. Применяется как правило для шифрования потков информации видео и аудио.

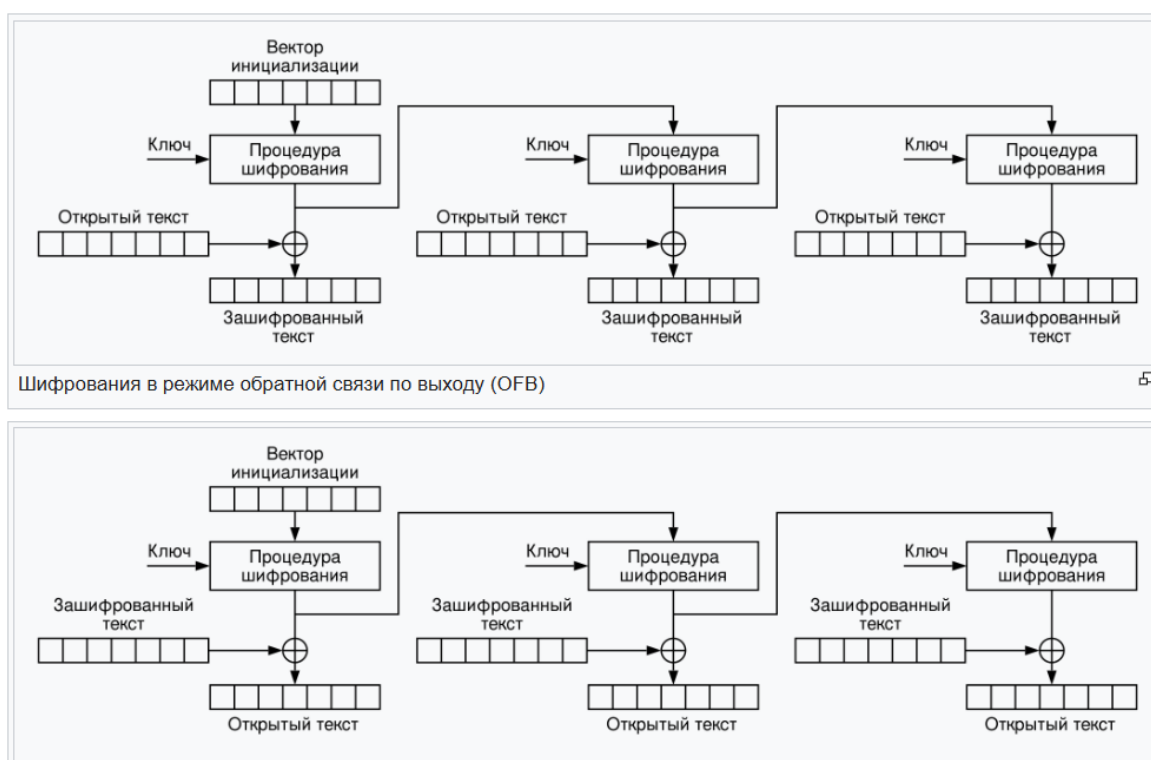


Рисунок 1.3 – Обобщенная схема алгоритма режима шифрования CFB

2 Конструкторская часть

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

2.1 Разработка алгоритмов

На рисунках 2.1 представлены схемы алгоритма AES, раунда AES.

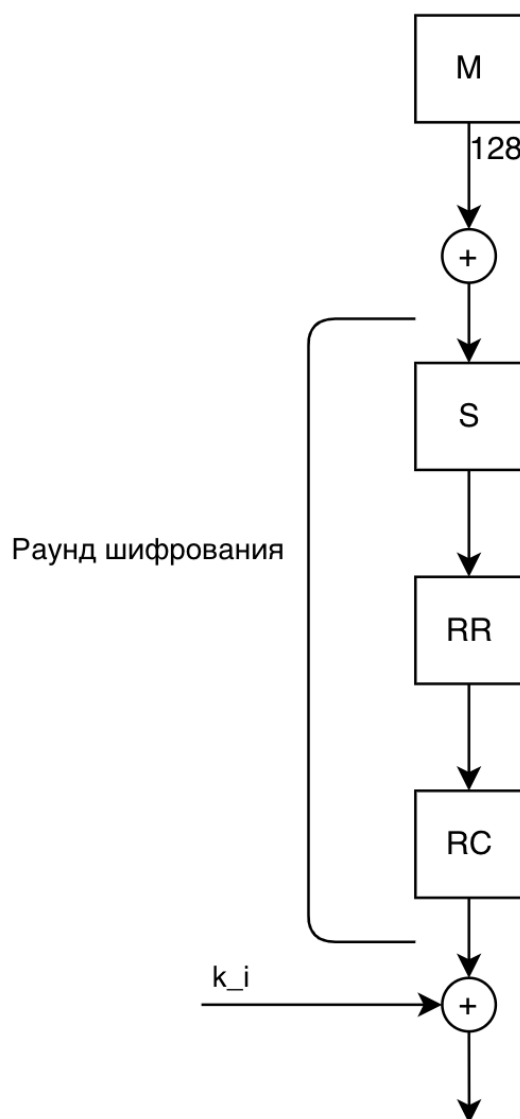


Рисунок 2.1 – Схема шифровального алгоритма AES

3 Технологическая часть

3.1 Средства реализации

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

3.2 Реализация алгоритма

Листинг 3.1 – Класс реализации режима CFB

```
1 class cfb {
2     public:
3     explicit cfb (const vector<uint8_t>& vi);
4
5     string crypt(string message, bool decrypt = false);
6
7     vector<block> divideBlocks(const vector<uint8_t> &message);
8     vector<uint8_t> mergeBlocks(const vector<block> &blocks);
9
10    vector<uint8_t> encrypt(vector<uint8_t>& block128, const
        vector<uint8_t>& key);
11    vector<uint8_t> decrypt(vector<uint8_t>& block128, const
        vector<uint8_t>& key);
12
13    void setVI(const vector<uint8_t>& vi);
14
15    void print_bloks(const vector<block> &blocks);
16    void print_message(const vector<uint8_t> message);
17 };
```

Листинг 3.2 – Класс шифрования и дешифрования AES

```
1 class aes: public IEncoder{
2     public:
3         block CryptBlock(const block& block128, const key& key128,
4             bool decrypter = false);
5         block EncryptBlock(const block& block128, const key& key128)
6             override;
7         block DecryptBlock(const block& block128, const key& key128)
8             override;
9
10        vector<key> GetKeys128(key key128, bool decrypter = false);
11        void AddRoundKey(mtx& block, const key& roundKey);
12        void SubBytes(mtx& block);
13        void InvSubBytes(mtx& block);
14        void ShiftRows(mtx& block);
15        void InvShiftRows(mtx& block);
16        void MixColumns(mtx& block);
17        void InvMixColumns(mtx& block);
18
19        mtx ArrayToMrx4x4(const block& block);
20        block Mrx4x4ToArray(const mtx &mtx);
21        void AddPadding(vector<uint8_t>& data);
22        void RemovePadding(vector<uint8_t>& data);
23
24    private:
25        // Уожение поля Галуа
26        uint8_t GMul(uint8_t x, uint8_t y);
27
28        AES_MODE mode = AES128;
29 }
```

Листинг 3.3 – Реализация алгоритма AES

```
1 block aes::EncryptBlock(const block &block128, const key &key128)
2 {
3     // Расширение ключа — KeyExpansion
4     auto keys = GetKeys128(key128, false);
5
6     // для удобства конвертируем в матрицу
7     auto state = ArrayToMrx4x4(block128);
8
9     // Начальный раунд — сложение с основным ключом;
10    AddRoundKey(state, keys[0]);
11
12    // 9 раундов шифрования
13    for (int i = 1; i < 10; i++)
14    {
15        SubBytes(state);
16        ShiftRows(state);
17        MixColumns(state);
18        AddRoundKey(state, keys[i]);
19    }
20
21    // Финальный раунд
22    SubBytes(state);
23    ShiftRows(state);
24    AddRoundKey(state, keys[10]);
25
26    return Mrx4x4ToArray(state);
27 }
```

3.3 Тестирование

Тестирование разработанной программы производилось следующим образом: выбирались случайные значения ключа и вектора IV, а также получа-

лась 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.

Вывод

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

Заключение

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

Были и выполнены следующие задачи:

- 1) изучен шифровальный алгоритм AES и его режим работы CFB;
- 2) реализован шифровальный алгоритм AES в виде программы, обеспечена возможность шифрования и расшифровки файла в режиме работы CFB ;
- 3) протестирована разработанная программа;
- 4) описаны и обоснованы полученные результаты в отчёте о выполненной лабораторной работе.

Список использованных источников

1. И.М. Шолин. Алгоритм переносной шифровальной машины энигма. — Кубанский государственный технологический университет.
2. Язык программирования C++. <https://learn.microsoft.com/en-us/cpp/cpp/cpp-language-reference?view=msvc-170>. дата обращения: 15.10.2023.
3. CLion. [jetbrains.com](https://www.jetbrains.com/idea/). дата обращения: 15.10.2023.