МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

**НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе №3

по курсу «Защита информации»

**Выполнил:** студент 4-го курса гр. 17208

Гафиятуллин А.Р

Новосибирск, 2020

Оглавление

[1. ЦЕЛИ РАБОТЫ 3](#_Toc58002960)

[2. ХОД РАБОТЫ 3](#_Toc58002961)

[2.1 RC4. 3](#_Toc58002962)

[2.2 AES. 5](#_Toc58002963)

[2.3 Хеш-функция на основе AES. 14](#_Toc58002964)

[3. ВЫВОДЫ 15](#_Toc58002965)

[4. СПИСОК ЛИТЕРАТУРЫ 15](#_Toc58002966)

[5. ПРИЛОЖЕНИЕ 15](#_Toc58002967)

# ЦЕЛИ РАБОТЫ

Запрограммировать алгоритмы RC4, AES и хеш-функцию на основе AES.

# ХОД РАБОТЫ

## RC4.

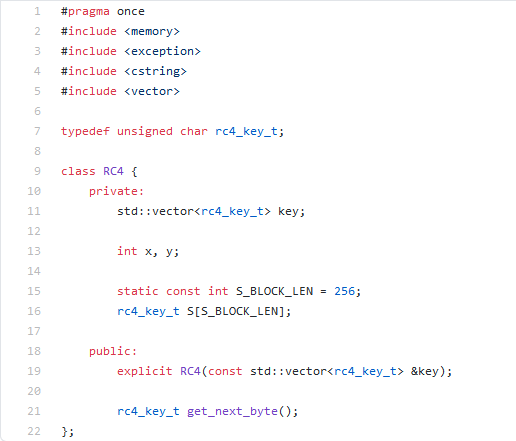


Рисунок 1. – Класс RC4.

В классе объявлены элементы алгоритма: **ключ**, **S-блок** и функция **get\_next\_byte** для генерации псевдослучайных битов.

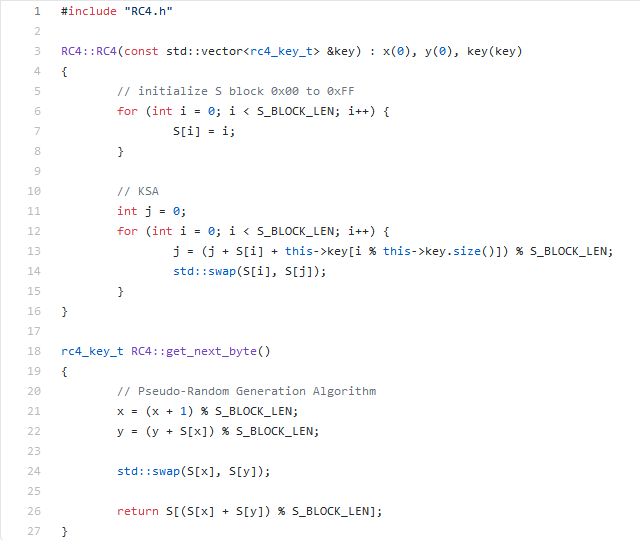


Рисунок 2. – Реализация методов класса RC4.

В конструкторе класса происходит инициализация S-блока с помощью алгоритма ключевого расписания (**Key-Scheduling Algorithm**).

Метод get\_next\_byteгенерирует псевдослучайную последовательность с помощью перестановки в S-блоке.

Генерируемые этим методом биты складываются по модулю 2 с символами из входного потока, как при шифровании, так и при дешифровке.

## AES.

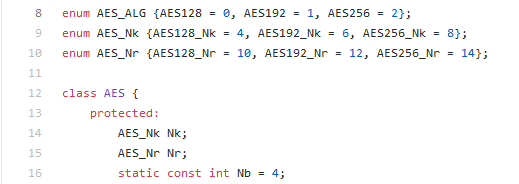


Рисунок 3. – Параметры алгоритма AES.

* **Nb** - число столбцов (32-битных слов), составляющих **state –** матрицу 4x4, представляющую промежуточный результат шифрования. Для AES **Nb** = 4;
* **Nk -** число 32-битных слов, составляющих ключ. Для AES **Nk** = 4, 6, или 8;
* **Nr -** число раундов, которое является функцией **Nk** и **Nb**. Для AES **Nr** = 10, 12, 14.

Эти параметры выбираются при создании объекта класса **AES**.

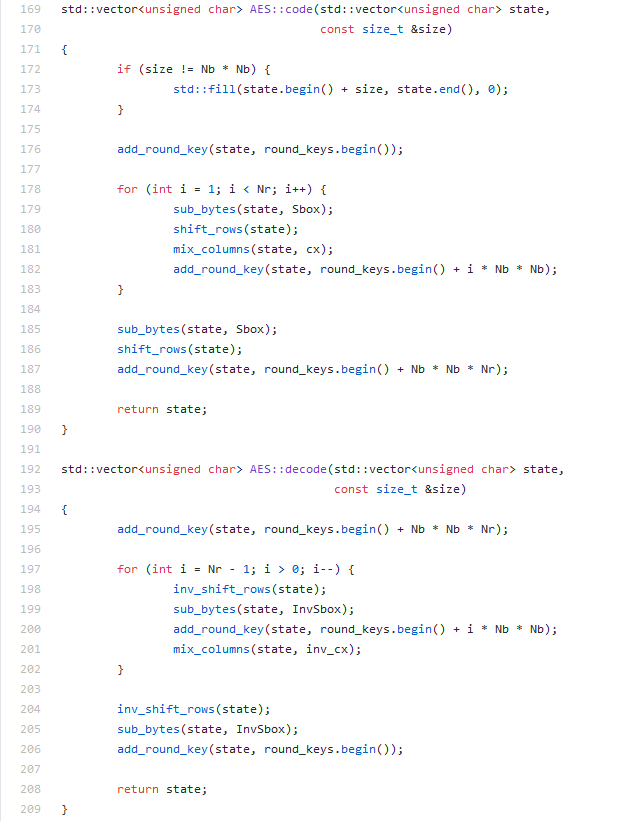


Рисунок 4. – Методы кодирования и декодирования 16-байтных массивов.

Метод **code** класса AESпроизводит различные трансформации в определенном порядке, а метод **decode** в обратном порядке. Метод code дополняет массив байтов нулями, если в нем не содержится 16 байтов. Метод decode этого не делает, так как размер зашифрованного файла в байтах кратен 16. Далее будут описаны преобразования, используемые алгоритмом AES.

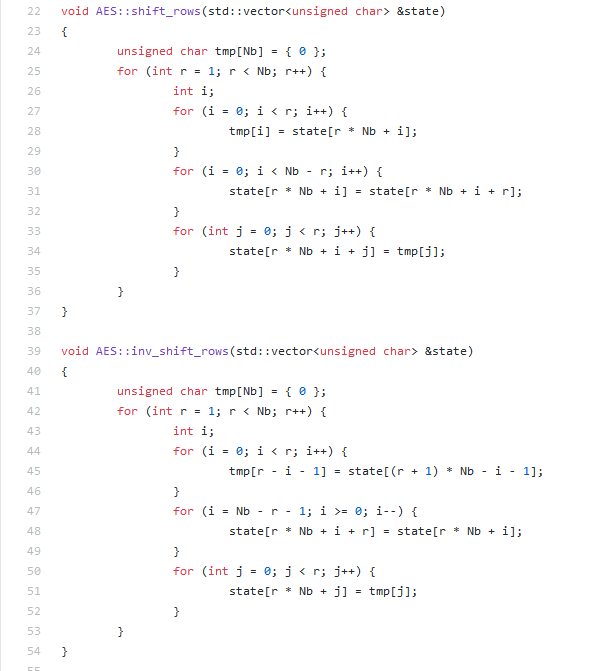


Рисунок 5. – Методы циклического сдвига строк матрицы state.

Массив state является матрицей 4x4, метод **shift\_rows** циклически сдвигает строки влево, а **inv\_shift\_rows** вправо, как показано на Рисунке 5.

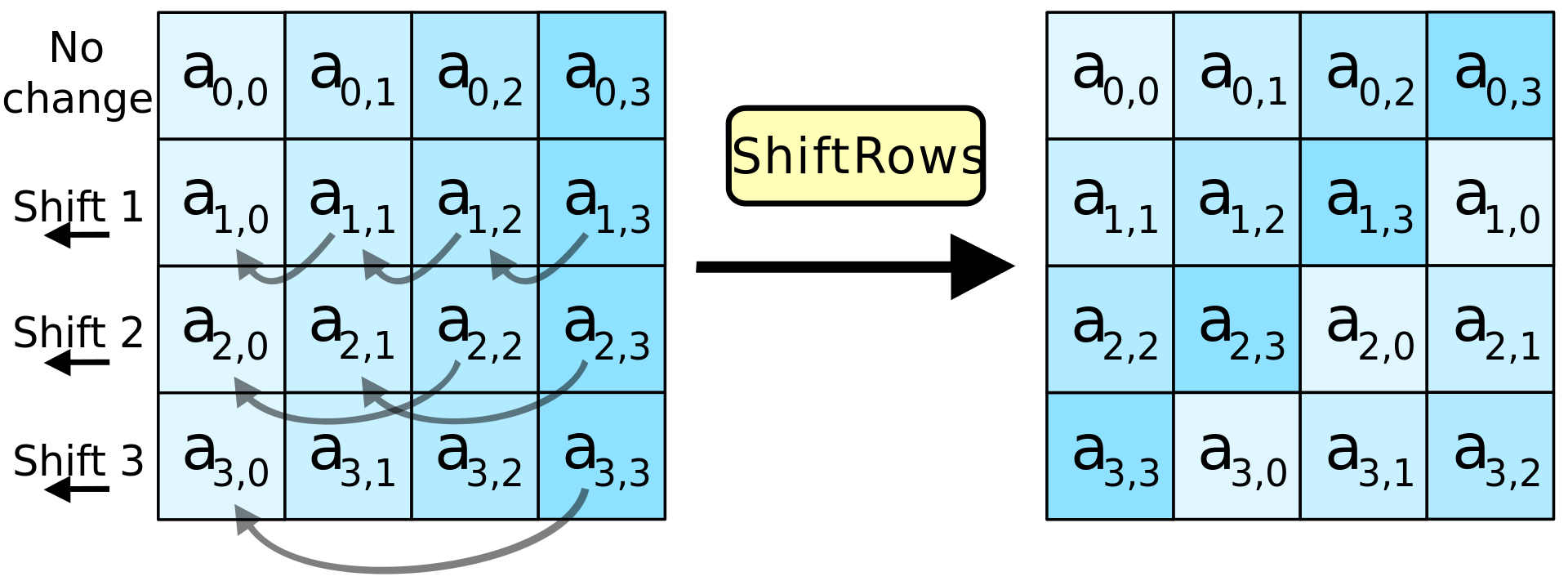


Рисунок 6. – Визуализация сдвига строк матрицы.

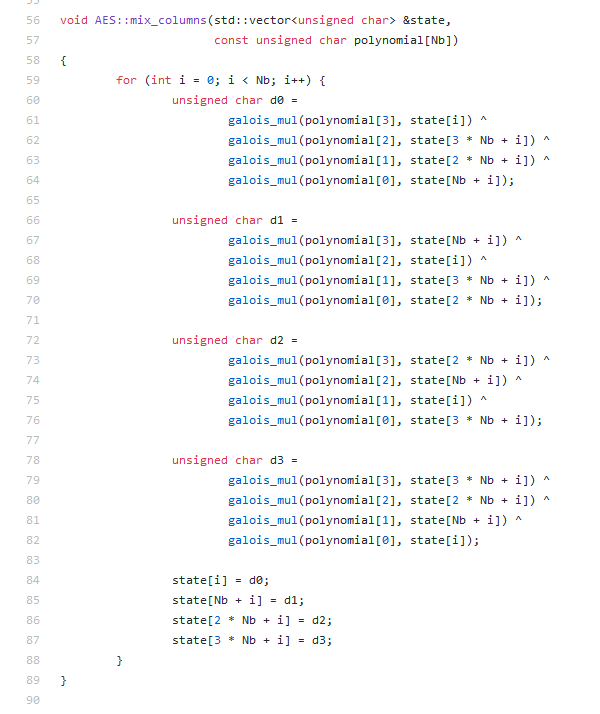
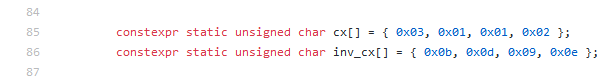


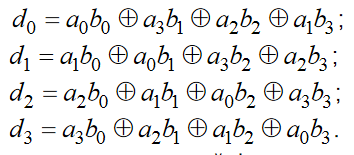
Рисунок 7. – Метод смешивания байтов колонок матрицы state.

В функции **mix\_columns** производится умножение по модулю x4 + 1 колонок матрицы state на многочлены, определенные в классе AES:



**cx** и **inv\_cx** это взаимно-обратные многочлены.

Умножение происходит по правилу:



Коэффициенты многочленов умножаются функцией **galois\_mul** в конечном поле GF(256).

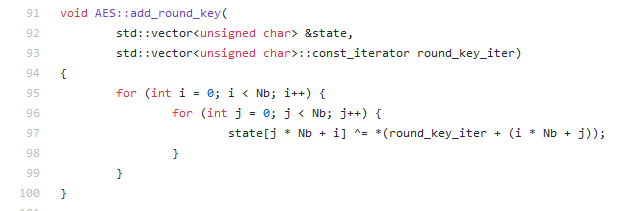


Рисунок 8. – Метод сложения матрицы state с переданным раундовым ключом.

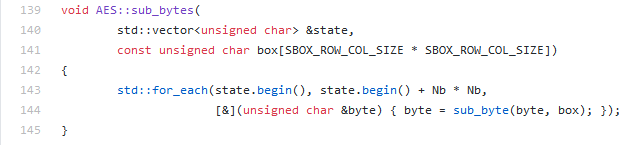


Рисунок 9. – Метод нелинейной замена байтов матрицы state с помощью Sbox/invSbox.

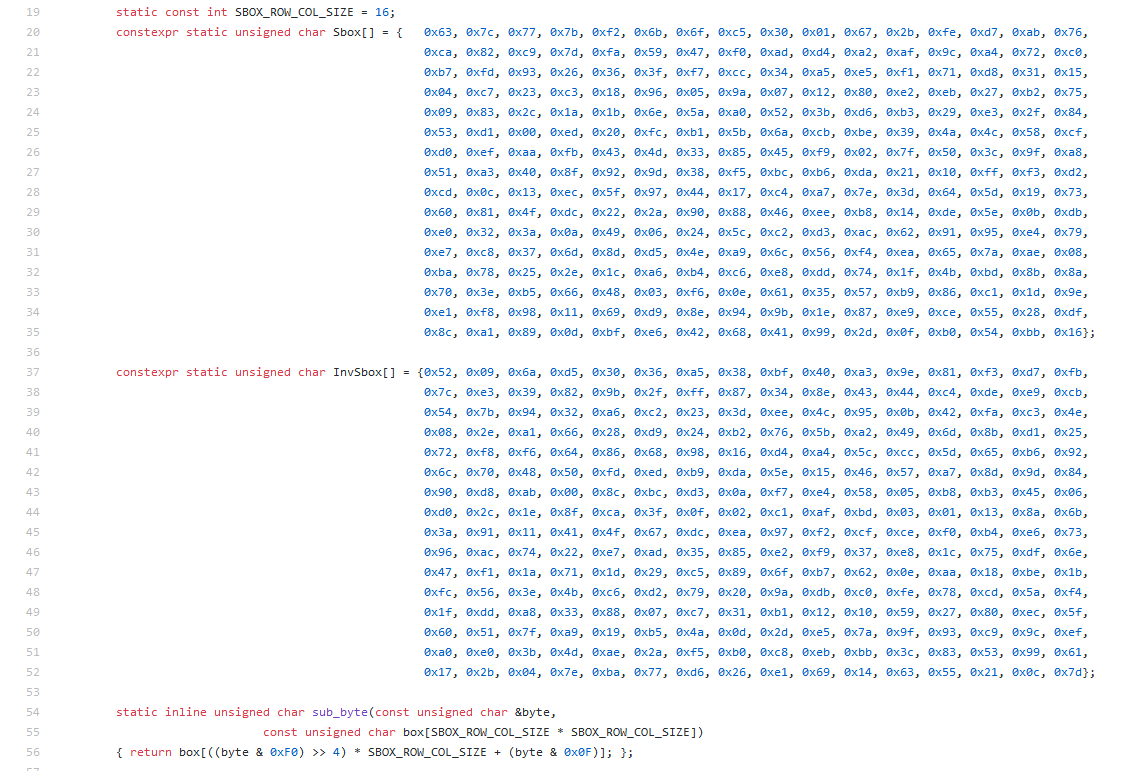


Рисунок 10. – Метод поиска по таблице Sbox/invSbox.

Таблицы **Sbox** и **invSbox** это взаимно-обратные таблицы замен байтов.

Метод **sub\_byte** осуществляет получения нового байта по таблицам. Первые 4 бита байта, переданного в этот метод, указывают на строку таблицы, а вторые на колонку. Метод **sub\_bytes** применяет это преобразование ко всем байтам в матрице state.

В эти методы передается Sbox при генерации ключевого расписания и шифрования и invSbox при дешифровке в методе decode.

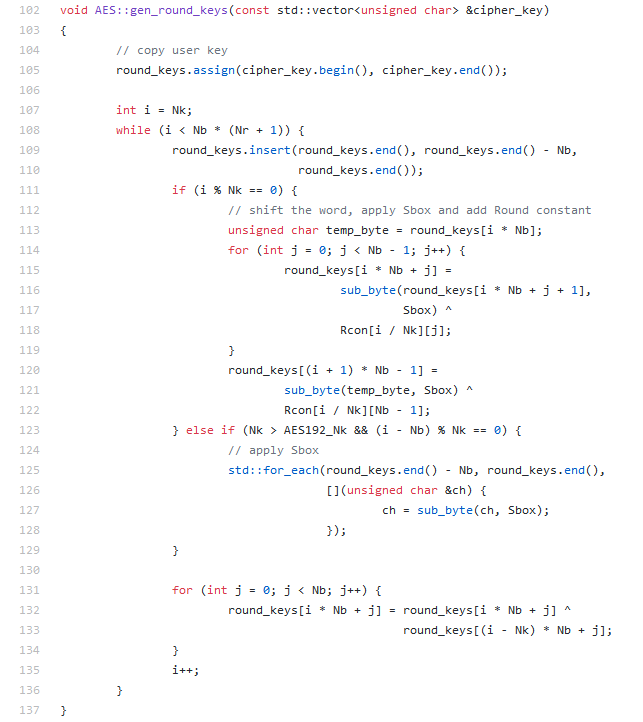


Рисунок 11. – Метод генерации ключевого расписания.

Первые Nk слов раундовых ключей совпадают со словами переданного в алгоритм ключа, далее происходит генерация остальных ключей по правилу:

1. Каждое Nk-ое слово получается из последовательного применения:
   1. циклического сдвига влево слова с предыдущей итерации;
   2. замены всех байтов методом sub\_byte;
   3. побитового сложения с (i / Nk)-ым словом из массива Rcon.
2. Если выбран AES256 и (i – 4) кратно Nk, то алгоритм модифицируется: дополнительно циклический сдвиг влево применяется к слову с предыдущего шага при выполнении этих условий;
3. Происходит побитовое сложение полученного модифицированного или оригинального слова с предыдущего шага со словом, полученным Nk шагов назад.

## Хеш-функция на основе AES.

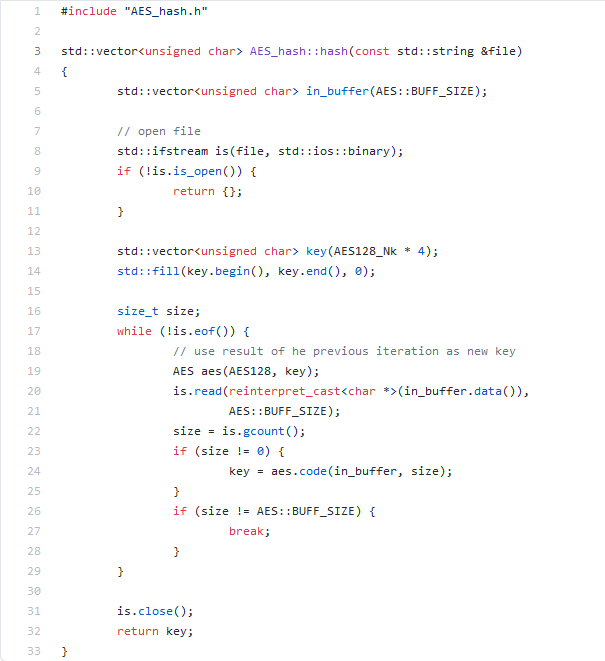


Рисунок 12. – Метод вычисления хэша на основе AES128.

Работа алгоритма заключается в передаче в качестве ключа для хеширования следующих 16 байт результат хеширования текущих 16 байт. Начальный ключ заполнен нулями.

# ВЫВОДЫ

Запрограммировали алгоритмы RC4, AES и хеш-функцию на основе AES.

# СПИСОК ЛИТЕРАТУРЫ

1. Простая реализация RC4 на C#. — URL: https://habr.com/ru/post/111510/ (дата обращения: 12.4.2020).
2. RC4.— URL: https://ru.wikipedia.org/wiki/RC4 (дата обращения: 12.4.2020).
3. Advanced Encryption Standard.— URL: https://ru.wikipedia.org/wiki/Advanced\_Encryption\_Standard (дата обращения: 12.4.2020).

# ПРИЛОЖЕНИЕ

Исходный код и примеры использования(в README.md): <https://github.com/gafiyatullin-a/NSU-FIT/tree/master/4th-year/informaition-security/laboratory-work-3>