МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информатики и вычислительной техники

Кафедра ИБ

# Лабораторная работа № 9

по дисциплине «Программирование на языке С++»

Выполнил студент БИ-31

\_\_\_\_\_\_\_\_\_\_\_Шестаков А. Д.\_\_\_\_\_\_\_\_\_\_

подпись дата

# Проверил: ст. пр. кафедры ИБ \_\_\_\_\_\_\_\_\_Сорокин О. Л.

подпись дата

Йошкар-Ола

2021 г.

**Оглавление**

**Теоретическая часть...…………………………………… 3**

**Листинг……………………………………………………. 6**

**Демонстрация работы………………………………….. 9**

**Вывод и ссылки……...………………………………….. 11**

**Теоретическая часть**

В данной работе реализовано шифрование Unicode строк блочным методом шифрования AES (Advanced encryption standard) в режиме CBC (Chiper Block Chaining).

**Advanced encryption standard**

**AES** – симметричный алгоритм блочного шифрования, прнятый в качестве стандарта шифрования правительством США. Этот алгоритм хорошо проанализирован и сейчас широко используется.

*Вспомогательные определения:*

**Block –** последовательность бит, из которых состоит input, output, state и round key. Также под Block можно понимать последовательность байтов.

**Cipher Key –** секретный криптографический ключ, который используется Key Expresion процедурой, чтобы произвести набор ключей для раундов (Round Keys);

**Cipher Text –** выходные данные алгоритма шифрования.

**Key Expansion –** процедура генерации Round Keys из Cipher Key.

**Round Key –** получаются из Cipher Key использованием процедуры Key Expresion.

**State –** промежуточный результат шифрования, который может быть представлен как прямоугольный массив байтов, имеющий 4 строки и *Nb* колонок.

**S-box –** нелинейная таблица замен, использующаяся в нескольких трансформациях замены байтов в процедуре Key Expresion для взаимно-однозначной замены значения байта.

**Rcon –** массив, который состоит из битов 32-разрядного слова и является постоянным для данного раунда.

*Вспомогательные процедуры:*

**AddRoundKey() -** трансформация при шифровании и обратном шифровании, при которой Round Key XOR’ится c State. Длина RoundKey равна размеру State.

**InvMixColumns() -** трансформация при расшифровании, которая является обратной по отношению к MixColumns().

**InvShiftRows() -** трансформация при расшифровании, которая является обратной по отношению к ShiftRows().

**InvSubBytes() -** трансформация при расшифровании, которая является обратной по отношению к SubBytes().

**MixColumns() -** трансформация при шифровании, которая берёт все столбцы State и смешивает их данные, чтобы получить новые столбцы.

**RotWord() -** функция, использующаяся в процедуре Key Expansion, которая берёт 4-байтовое слово и производит над ним циклическую перестановку.

**ShiftRows() -** трансформации при шифровании, которые обрабатывают State, циклически смещая последние три строки State на разные величины.

**SubBytes() -** трансформации при шифровании, которые обрабатывают State, используя нелинейную таблицу замещения байтов (S-box), применяя её независимо к каждому байту State.

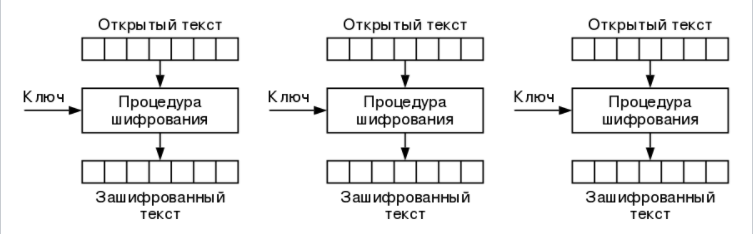
**SubWord() -** функция, используемая в процедуре Key Expansion, которая берёт на входе четырёхбайтовое слово и, применяя S-box к каждому из четырёх байтов, выдаёт выходное слово.

**Режимы шифрования**

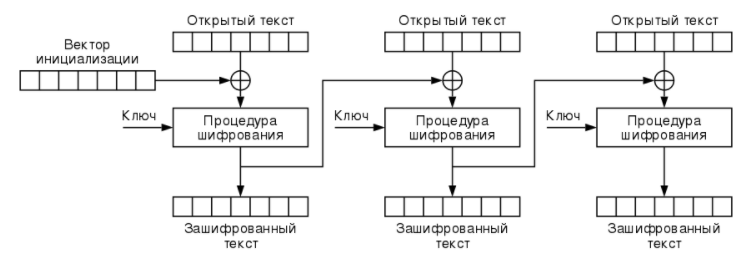
**Режим шифрования** – метод применения блочного шифра, позволяющий преобразовать последовательность блоков открытых данных в последовательность блоков зашифрованных данных. При этом для шифрования одного блока могут использоваться данные другого блока. Режимы шифрования используются для изменения процесса шифрования так, чтобы результат шифрования каждого блока был уникальным вне зависимости от шифруемых данных и не позволял сделать какие-либо выводы об их структуре.

В 1981 году был принят стандарт FIPS 81. В стандарте были описаны первые режимы работы блочных шифров: **ECB, CBC, OFB** и **CFB**.

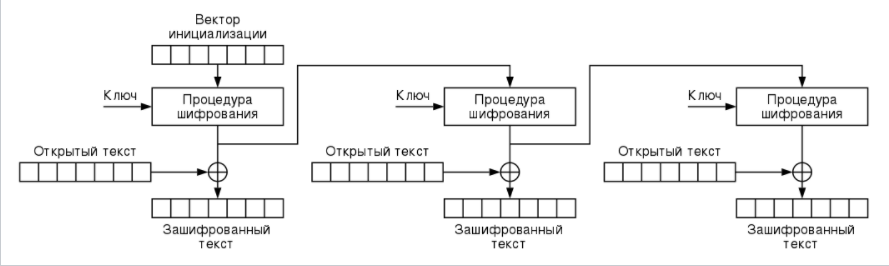
### **1) Electronic Codebook (ECB) –** режим простой замены.



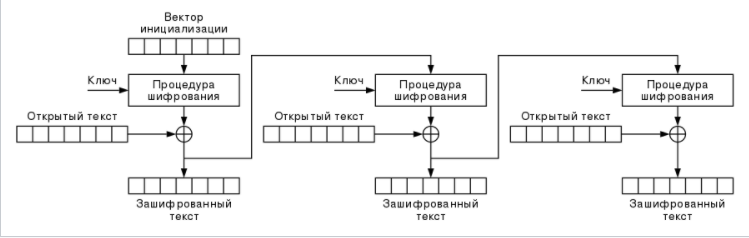
### **2) Cipher Block Chaining (CBC) –** режим сцепления блоков шифротекста.



### **3) Output Feedback (OFB) –** режим обратной связи по выходу.



### **4) Cipher Feedback (CFB) –** режим обратной связи по шифротексту.



**Листинг программы**

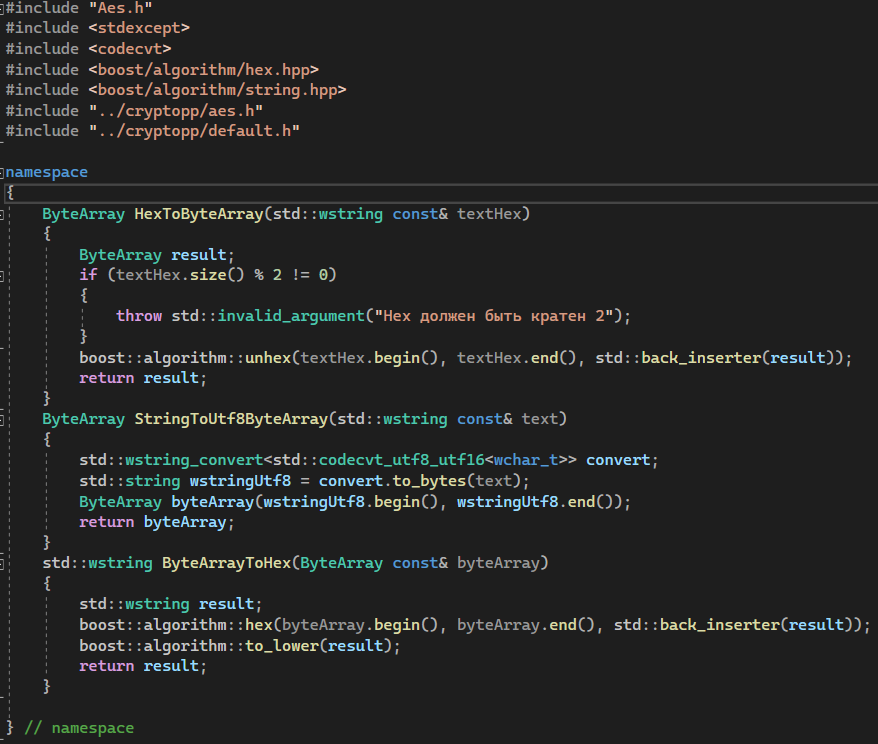
В данной программе реализован один класс **Aes**– даёт возможность шифровать/расшифровывать Unicode текст.

* **Aes.h**



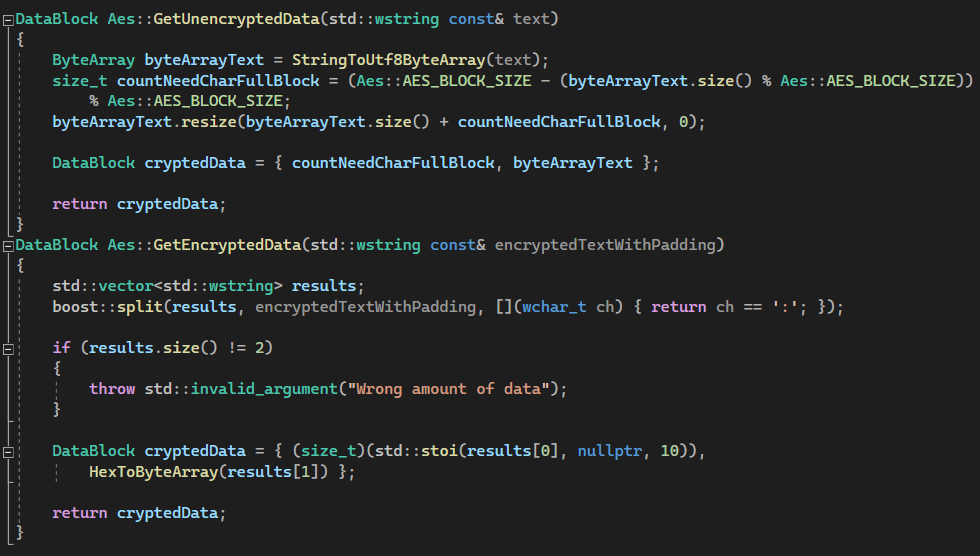
В данном файле описаны:

1. 2 функции которые делают шифрование/дешифрования: Encrypt и Decrypt соответственно.
2. 2 функции которые помогают для экземпляра класса установить ключ и вектор инициализации: SetKey и SetInitialVector.
3. 2 функции которые возвращают установленные в переменные класса значения ключа и вектора инициализации: GetKey и GetInitialVector.
4. Структура данных, где numPaddingBytes – число необходимых до полноты блока кратности 16 и ByteArray – это массив байт, где байтом считается беззнаковый char.

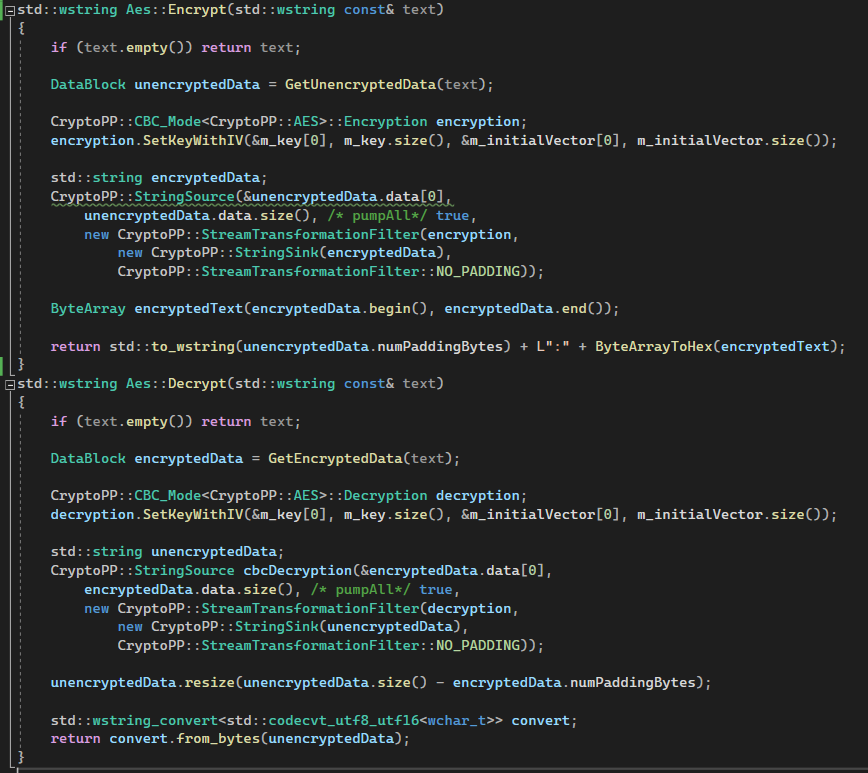
* **Aes.cpp**

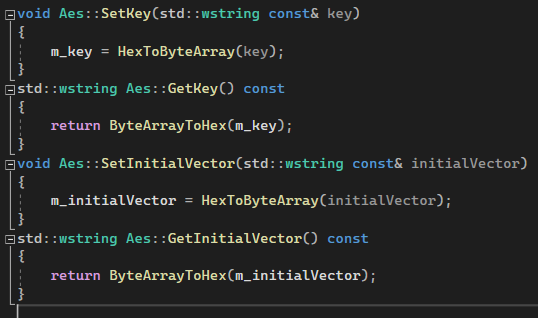
В безымянном пространстве описаны вспомогательные функции HexToByteArray, StringToUtf8ByteArray, ByteArrayToHex. Данные функции необходимы для конвертации из ByteArray в Hex и обратно. А также функция для конвертации из wstring в string.

Для облегчения переводов в Hex была использована boost библиотека. Hex – это запись числа в шестнадцатеричной системе счисления. Она используется для корректной записи зашифрованной информации куда-либо.



Метод GetUnencryptedData(std::wstring const& text) получает на вход текст для шифрования, считает его длину и дополняет его в случае необходимости до кратности 16, а сем текст переводит в массив байт. Метод GetEnencryptedData(std::wstring const& text) получает на вход зашифрованный текст в виде: {число}:{hex текст}. После чего разбивает этот формат и выдаёт в виде объявленной ранее структуры DataBlock.

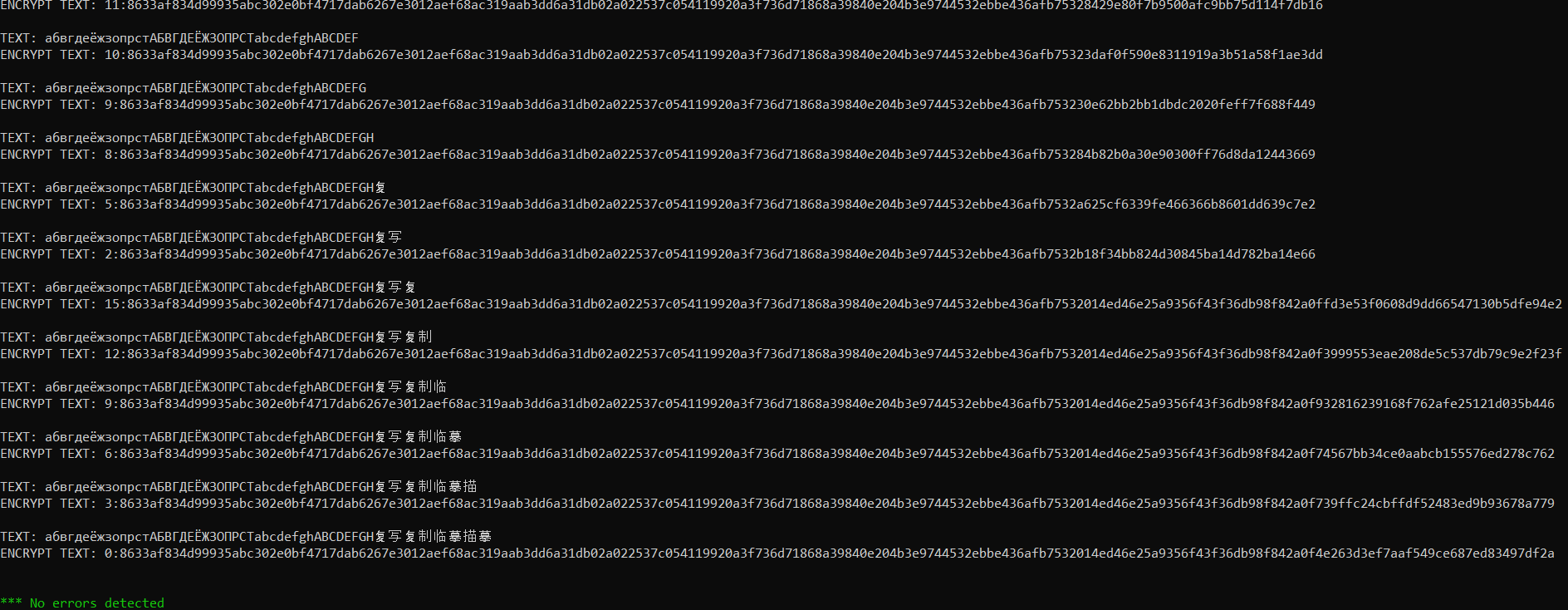




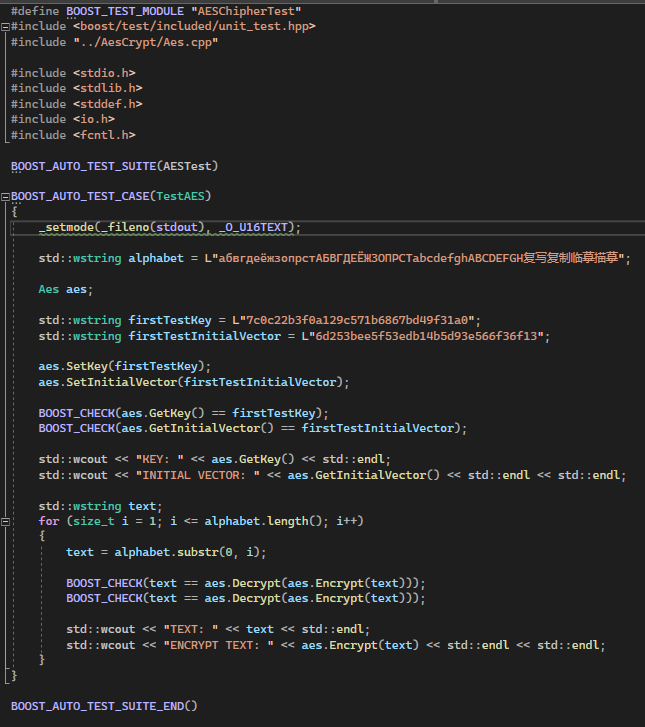
Методы геттеры и сеттеры помогают просто получить информацию о классовых переменных. В частности, они используются для проверки, верно ли установлен ключ и вектор инициализации в экземпляр класса.

Методы Encrypt и Decrypt производят непосредственные вычисления для шифрования/дешифрования. За реализацию преобразований, а также реализацию шифрования каким-либо способом была взята библиотека CryptoPP. Мод CBC так же был взят из данной библиотеке для демонстрации работы.

**Демонстрация работы:**



Для демонстрации работы были написан Unit тест, который помогает убедиться в том, что функция работает корректно. После запуска тестов, можно убедиться в корректности шифрования/дешифрования.

**AesCryptText.cpp**

За основу тестирования была взята boost библиотека. Тестирование состоит в том, чтобы на каждой итерации увеличивать символ шифрования на 1, пока не зашифрую всю тестовую строку. Специально были взяты символы из разных алфавитов, для проверки на возможность шифрования для разных языков.

**Вывод**

В процессе написания данной работы были освоены:

- То, как устроено шифрование/дешифрование в симметричном алгоритме AES.

- Работа с библиотекой CryptoPP.

- Изучены теоретические знания того, как происходят перевод из разных систем счисления.

- Умение отличать кодировки и работа с ними.