Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Исследование потоковых шифров

Студент: Валдайцев А. Д.

ФИТ 3 курс 5 группа

Преподаватель: Савельева М. Г.

Минск 2023

# Потоковые шифры

Потоковый шифр это – симметричный шифр, преобразующий каждый символ *mi* открытого текста в символ шифрованного *ci*, зависящий от ключа и расположения символа в тексте.

Основной задачей потоковых шифров является выработка некоторой последовательности (гаммы) для зашифрования, то есть выходная гамма является ключевым потоком (ключом) для сообщения.

Потоковый шифр максимально должен имитировать одноразовый блокнот. В соответствии с этим ключ должен по своим свойствам максимально походить на случайную числовую последовательность. Ключевые последовательности (случайные последовательности (СП), либо псевдослучайные последовательности (ПСП)) вырабатываются специальными блоками систем потокового шифрования – генераторами.

# Генерация ПСП на основе алгоритма BBS

Начальное значение *x0* генератора вычисляется на основе соотношения



где *n* является произведением простых чисел *p* и *q*, однако в нашем случае эти простые числа должны быть сравнимы с числом 3 по модулю 4; число *x* должно быть взаимно простым с *n*; число *n* называют числом Блюма.

Выходом генератора на *t*-м шаге является младший бит числа *xt*:



Для реализации алгоритма генерации псевдослучайных последовательностей с помощью алгоритма BBS был создан одноименный класс, представленный на рисунке 1.1.

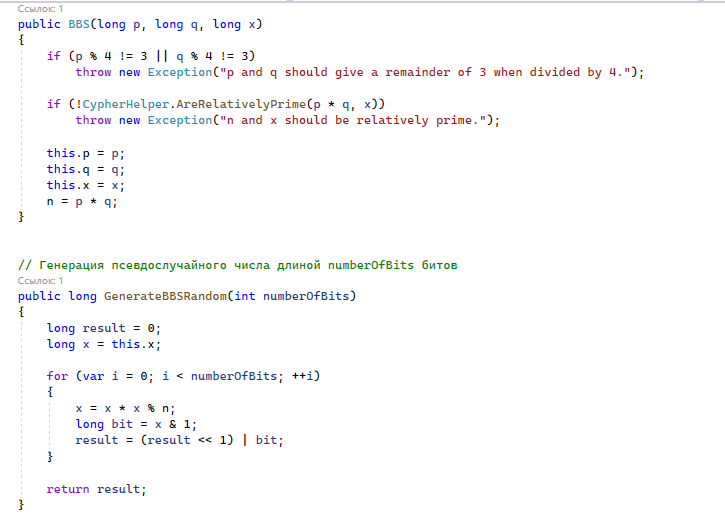


Рисунок 1.1 – Код генерации ПСП с помощью BBS

В соответствии с вариантом, на вход алгоритма подаются следующие числа: *p* = 56155, *q* = 87151, *x* = 512. Результат работы алгоритма, создающий 16-битное псевдослучайное число, представлен на рисунке 1.2.

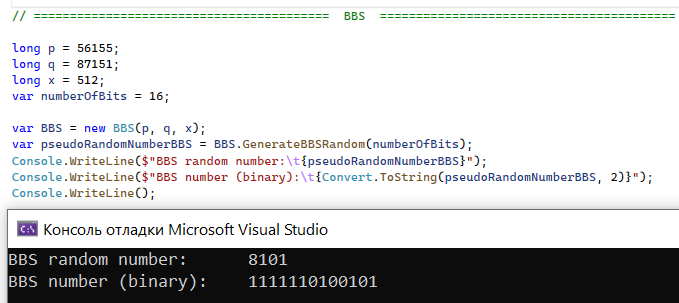


Рисунок 1.2 – Вывод алгоритма генерации ПСП

# Шифрование алгоритмом RC4

Для работы алгоритма необходимо выполнить следующие действия:

* инициализировать *S*-блок таблицы начальных замен;
* сгенерировать на основании *S*-блока псевдослучайные числа для ключей *K*;
* выполнить операцию сложения по модулю 2 каждого 8-битного ключа *K* с открытым текстом для получения шифротекста.

Для инициализации *S*-блока начальных замен реализована следующая функция, представленная на рисунке 1.3.

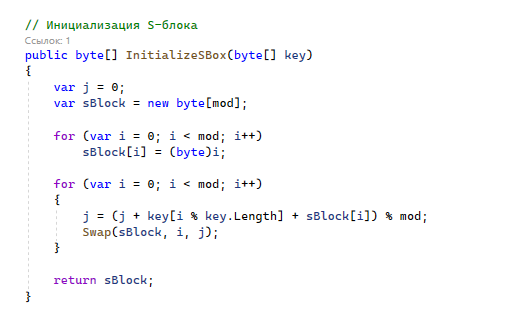


Рисунок 1.3 – Код функции инициализации S-блока

Далее по формуле необходимо создать псевдослучайную последовательность ключей *K*, что представлено на рисунке 1.4.

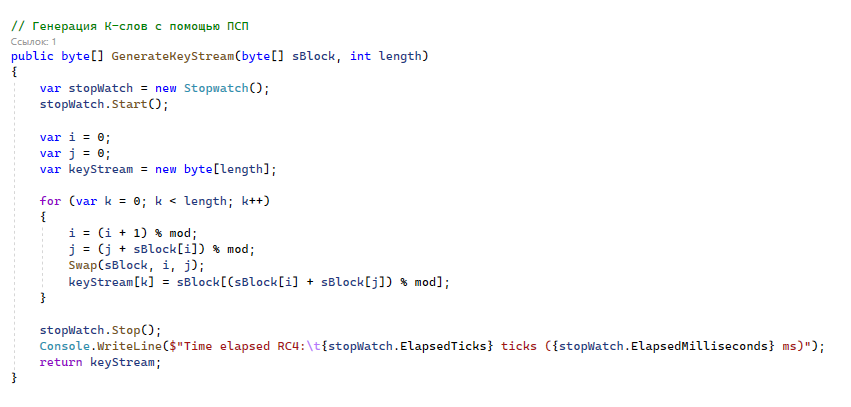


Рисунок 1.4 – Код функции генерации ПСП

Методы для зашифрования и расшифрования, а также вспомогательный метод для замены элементов местами в *S*-блоке представлены на рисунке 1.5.

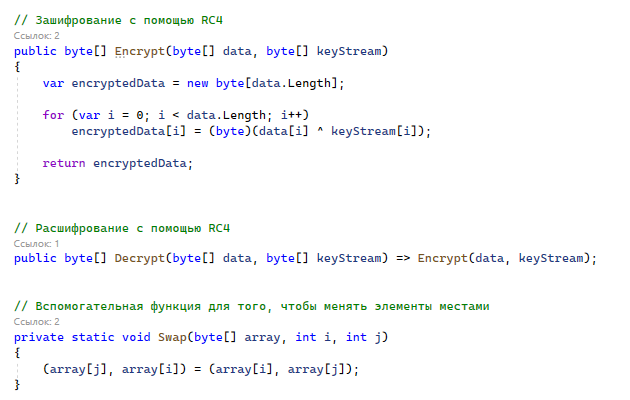


Рисунок 1.5 – Код функций зашифрования и расшифрования

Вывод функции зашифрования и расшифрования и затраченного времени представлен на рисунке 1.6.

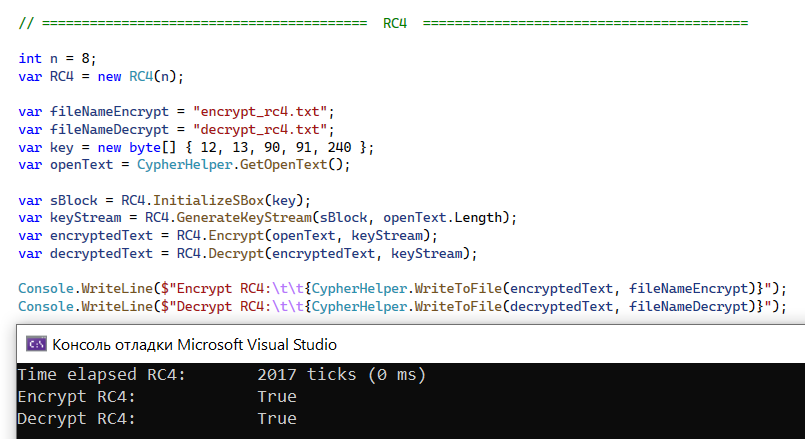


Рисунок 1.6 – Вывод функций зашифрования и расшифрования

# Время выполнения генерации ПСП

Для оценки времени выполнения операций генерации ПСП были созданы псевдослучайные последовательности разной длины. График представлен на рисунке 2.1.



Рисунок 2.1 – Время выполнения генерации ПСП

При анализе данного графика можно заметить, что генерация ПСП происходит гораздо быстрее, чем операции зашифрования и расшифрования в остальных алгоритмах, из чего можно сделать вывод, что алгоритм RC4 может быть довольно производительным для создания псевдослучайных последовательностей.

# Вывод

В данный лабораторной работе были изучены и приобретены практические навыки разработки и использования потоковых шифров.