Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Исследование потоковых шифров**

Студент: Гвоздовский К.В.

ФИТ 3 курс 6 группа

Преподаватель:

Нистюк Ольга Александровна

1. **Цель работы**

Изучение и приобретение практических навыков разработки и использования приложений для реализации блочных шифров.

1. **Задание**

Разработать авторские многооконные приложения в соответствии с целью лабораторной работы. При этом можно воспользоваться готовыми библиотеками либо программными кодами, реализующими заданные алгоритмы.

Приложение 1 должно реализовывать генерацию ПСП в соответствии с вариантом из табл. 6.6.

Приложение 2 должно реализовывать алгоритм RC4 в соответствии с вариантом из табл. 6.7, а также дополнительно выполнять оценку скорости выполнения операций генерации ПСП.

1. **Ход работы**

**Теоретические сведения**

***Классификация и общие свойства потоковых шифров***

Потоковый шифр (иногда говорят «поточный») – симметричный шифр, преобразующий каждый символ открытого текста в символ шифрованного , зависящий от ключа и расположения символа в тексте. Термин «потоковый шифр» обычно используется в том случае, когда шифруемые символы открытого текста представляются одной буквой, битом или реже – байтом.

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

Все потоковые шифры делятся на 2 класса: *синхронные* и *асинхронные* (или самосинхронизирующиеся).

Синхронные потоковые шифры (СПШ) характеризуются тем, что поток ключей генерируется независимо от открытого текста и шифртекста. Главное свойство СПШ – нераспространение ошибок. Синхронные потоковые шифры уязвимы к атакам на основе изменения отдельных битов шифртекста.

В самосинхронизирующихся потоковых шифрах символы ключевой гаммы зависят от исходного секретного ключа шифра и от конечного числа последних знаков зашифрованного текста. Недостаток этих потоковых шифров – распространение ошибок, так как искажение одного бита в процессе передачи шифртекста приведет к искажению нескольких битов гаммы и, соответственно, расшифрованного сообщения.

***Генераторы ключевой информации***

Ключевые последовательности (случайные последовательности (СП), либо псевдослучайные последовательности (ПСП)) вырабатываются специальными блоками систем потокового шифрования – генераторами.

Действующий в Беларуси стандарт СТБ 34.101.47–2017 определяет базовые понятия в рассматриваемой предметной области:

– случайные числа (последовательности) – последовательность элементов, каждый из которых не может быть предсказан (вычислен) только на основе знания предшествующих ему элементов данной последовательности;

– псевдослучайные числа – последовательность элементов, полученная в результате выполнения некоторого алгоритма и используемая в конкретном случае вместо последовательности случайных чисел.

***Линейный конгруэнтный генератор***

Часто используемый алгоритм генерирования ПСП реализуется на основе так называемого линейного конгруэнтного генератора, описываемого следующим рекуррентным соотношением:  


где и – соответственно *t*-й (предыдущий) и (*t* + 1)-й (текущий, вычисляемый) члены числовой последовательности; *а*, *с* и *n* – константы. Период такого генератора (период ПСП) не превышает *n*.

***Генератор псевдослучайных чисел на основе алгоритма RSA***

Генератор же ПСП на основе RSA устроен следующим образом. Последовательность генерируется с использованием соотношения



Начальными параметрами служат *n*, большие простые числа *p* и *q* (причем *n* = *pq*), целое число е, взаимно простое с произведением (*р* – 1)(*q* – 1), а также некоторое случайное начальное значение .

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

Безопасность генератора опирается на сложность взлома алгоритма RSA, т. е. на разложение числа *n* на простые сомножители.

**Практическая часть**

В соответствии с вариантом, необходимо было реализовать генерацию ПСП с помощью алгоритма BBS. Для этого был разработан класс BBSGenerator, представленный на рисунке 3.1.



Рисунок 3.1 – Класс BBSGenerator

Параметрами для данного алгоритма являются числа *p*, *q*, *n* и *x*, 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, т. е. при делении *p* и *q* на 4 должен получаться одинаковый остаток: 3; число *x* должно быть взаимно простым с *n*; число *n* называют числом Блюма. Выходом генератора на *t*-м шаге является младший бит числа *xt*: *xt* = (*xt* - 1)2 mod *n*. А начальное значение x0 генератора вычисляется на основе соотношения *x0* = *x2* mod *n*.

Для данного варианта были выбраны следующие числа: *p* = 11, *q* = 23, получаем *n* = 253 . В качестве взаимно простого числа с n был выбран *x* = 2. В результате получим вывод, представленный на рисунке 3.2.

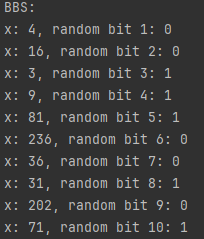


Рисунок 3.2 – Результат генерации ПСП с помощью алгоритма BBS

Для следующего приложения необходимо было реализовать шифрования и расшифрование с помощью потокового шифра RC4, который строится на основе генератора псевдослучайных битов. На вход генератора записывается ключ, а на выходе читаются псевдослучайные биты. Длина ключа может составлять от 40 до 2048 бит.

Чтобы выполнить эту задачу, был разработан класс RC4, представленный на рисунке 3.3.

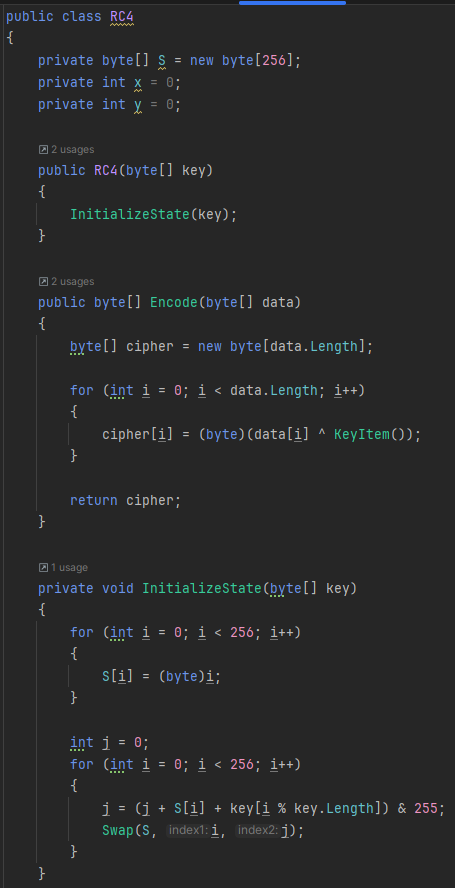


Рисунок 3.3 – Класс RC4

В соответствии с вариантом, размер блока *n* = 8, ключ {43,45,100,21,1}. Шифруемая строка – «Gvozdovskiy Kirill Vladimirovich». В результате получаем вывод, представленный на рисунке 3.4

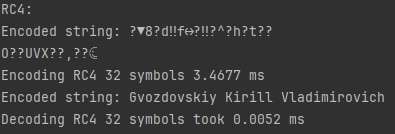


Рисунок 3.4 – Результат шифрования и расшифрования с помощью RC4

Кроме того, необходимо было оценить скорость выполнения операций шифрования и расшифрования с помощью потокового шифра RC4. В результате мы получили графики, представленные на рисунках 3.5 – 3.6.

Рисунок 3.5 – Скорость выполнения операции шифрования

Рисунок 3.6 – Скорость выполнения операции расшифрования

**Вывод**

В ходе лабораторной работы были изучены и приобретены практические навыки разработки и использования приложений для реализации потоковых шифров.

Также было разработано авторское приложение в соответствии с целью лабораторной работы.