Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Исследование потоковых шифров

Студент: Николаева Е.В.

ФИТ 3 курс 5 группа

Преподаватель:

Савельева Маргарита Геннадьевна

Минск 2023

1. **Описание приложения**

Приложение написано на языке программирования C# и позволяет выполнить 2 задачи:

* генерация псевдослучайной последовательности на основе алгоритма Линейный конгруэнтный генератор;
* зашифрование и расшифрование строки с помощью алгоритма RC4 с параметрами: *n*=8; ключ={13, 19, 90, 92, 250}.

Также разработанное приложение позволяет оценить скорость выполнения операций генерации ПСП.

**2.1. Генерация ПСП на основе алгоритма Линейный конгруэнтный генератор**

Для реализации генерации псевдослучайной последовательности (ПСП) на основе алгоритма Линейного конгруэнтного генератора была разработана функция, представленная на рисунке 2.1.

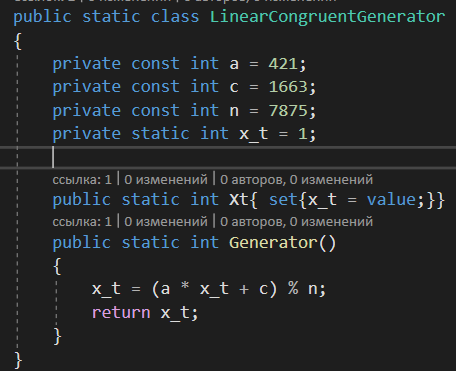


Рис. 2.1 – Реализация генерации числа ПСП

Данная функция вычисляет каждое число последовательности отдельно по формуле квадратичных вычетов: , где *xt* и *xt*+1 – 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.

Также, нами была оценена скорость выполнения генерации ПСП с помощью встроенной возможности C# - Stopwatch. Вычисленное время составило 5 мс (рисунок 2.2). Следовательно, данный алгоритм является сравнительно медленным.

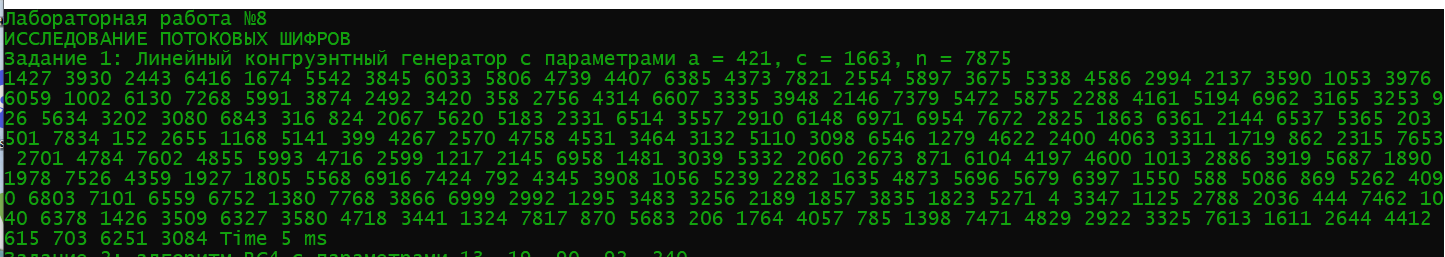


Рис. 2.1 – Результат работы первого задания

**2.2. Реализация алгоритма RC4**

Для решения поставленной задачи разработаем C#-класс RC4.

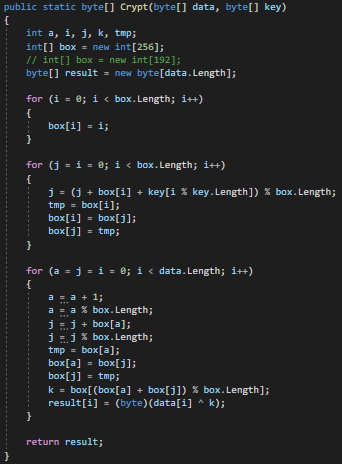


Рис. 2.3 – Реализация функции Crypt

Сначала *S*-блок пополняется линейно: 0,1…255. Затем заполняется секретным ключом другой массив [256]. Если необходимо, ключ повторяется многократно чтобы заполнить весь массив *K*0…*K*255. Далее массив *S* перемешивается путем перестановок, определяемых ключом.

Для зашифрования и расшифрования исходного сообщения используется функция Crypt, представленная на рисунке 2.3.

В параметры данной функции передается массив исходных байтов и их размер. Для каждого исходного байта запрашивается текущий байт ключа, после чего они объединяются при помощи XOR для получения 8-битного шифротекста.

Результат выполнения данного консольного приложения представлен на рисунке 2.4.

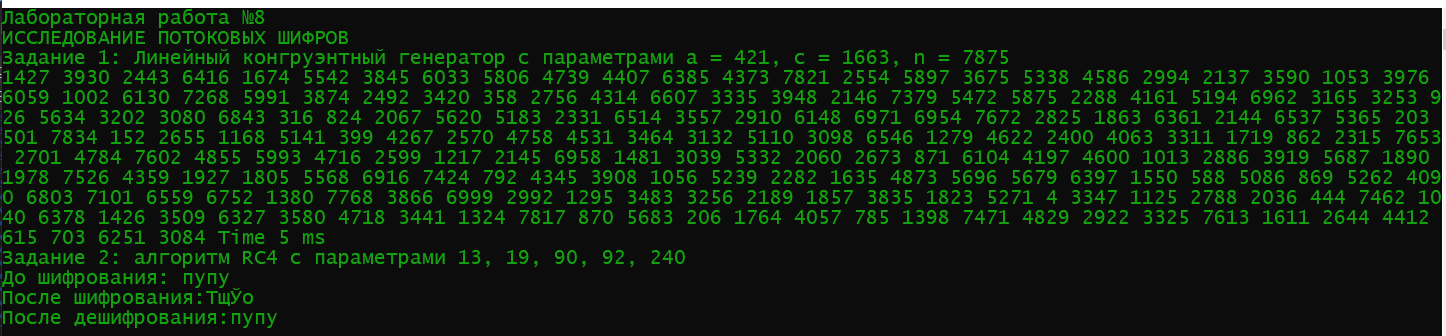


Рис. 2.4 – Результат работы приложения

**Вывод**

В ходе лабораторной работы были приобретены практические навыки разработки и использования приложений для реализации потоковых шифров. Было разработано приложение для реализации Линейного конгруэнтного генератора-алгоритма генерации псевдослучайной последовательности. Также, был реализован алгоритм RC-4 и выполнен анализ криптостойкости.