Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

ИССЛЕДОВАНИЕ АСИММЕТРИЧНЫХ 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 сверхвозрастающей последовательности (тайного ключа);
* вычисление нормальной последовательности (открытого ключа);
* зашифрование сообщения, состоящего из ФИО;
* расшифрование сообщения;
* оценка времени выполнения операций зашифрования и расшифрования.

1. **Методика выполнения поставленных задач**

Для реализации генерации сверхвозрастающей последовательности была разработана функция, представленная на рисунке 2.1.

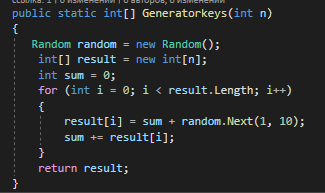


Рис. 2.1 – Реализация генерации тайного ключа

Данная функция вычисляет последовательно случайные числа, с условием, что каждый последующий элемент будет больше суммы всех предыдущих.

Для вычисления нормальной последовательности была разработана функция, представленная на рисунке 2.2.

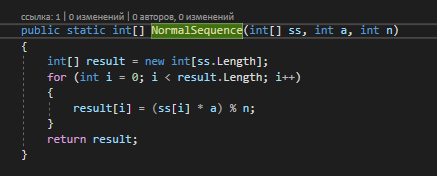


Рис. 2.2 – Реализация вычисления открытого ключа

Данная функция вычисляет элементы последовательности *e*={} по формуле: *=* *×a* (mod *n*), где *di* – члены тайного ключа, n больше суммы всех чисел последовательности, причем НОД (*a*,*n*) = 1.

Для зашифрования сообщения *M* каждый его символ *mi* был представлен в двоичной форме. Функция для шифрования, представленная на рисунке 2.3

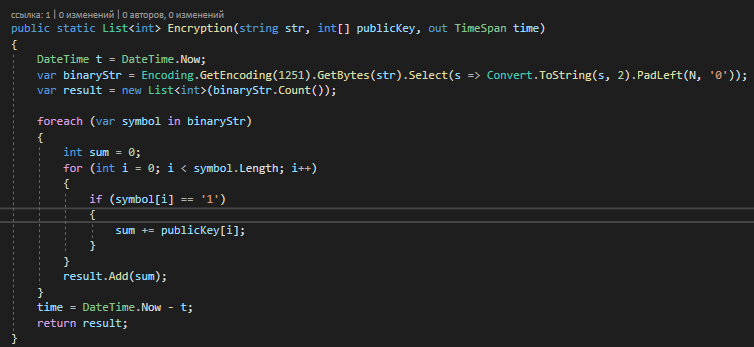


Рис. 2.3 – Реализация шифрования сообщения

Для расшифрования сообщения был использован сгенерированный тайный ключ.

Реализация данной функции представлена на рисунке 2.4.

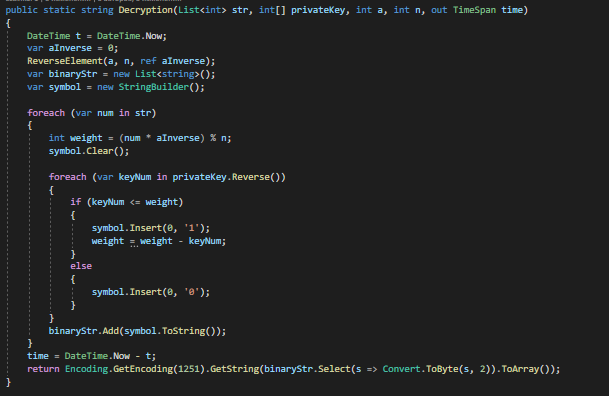


Рис. 2.4 – Реализация функции расшифрования

Результат выполнения данного консольного приложения представлен на рисунке 2.5.

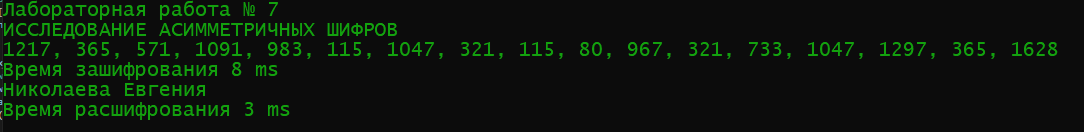


Рис. 2.5 – Результат работы приложения

Также, нами была оценена скорость выполнения зашифрования/расшифрования сообщения с помощью встроенной возможности C# – Stopwatch. Вычисленное время составило 8 мс и 3мс соответственно, что является неплохим результатом.

Сравним результаты работы при увеличении символов и использовании других кодировок. Посмотрим на рисуноки 2.6 и 2.7.

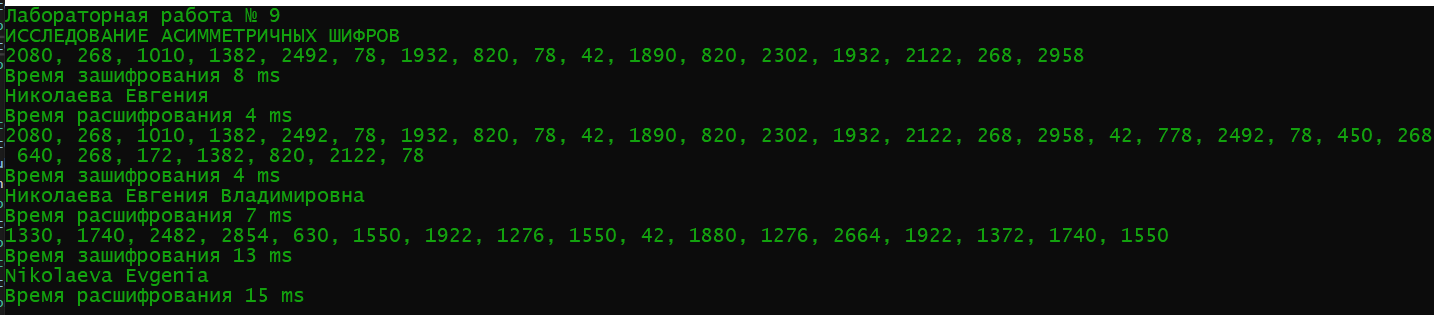


Рис. 2.6 – Результат работы приложения с разными данными

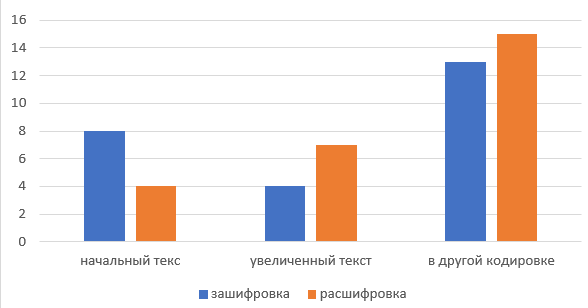


Рис. 2.7 – График зависимости времени по разным критериям

Как можно увидеть из графика, то значительный скачок наблюдается при перемени кодировки, при увеличении текста той же кодировки время не увеличивается, такое возможно, но весьма редко. В моем случае, у меня много повторяющихся буков, потому кодировка прошла быстрее, так как в кеше храниться уже ранее зашифрованные символы.

**Вывод**

В ходе лабораторной работы были приобретены практические навыки разработки и использования приложений для реализации асимметричных шифров. Было разработано приложение для реализации методов генерации ключевой информации и ее использования. Также была оценена скорость зашифрования/расшифрования.