Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Исследование асимметричных шифров

RSA и Эль-Гамаля

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 следующие задачи:

* генерация ключевой информации;
* зашифрование текстовых документов на основе алгоритма RSA;
* расшифрование на основе алгоритма RSA;
* зашифрование на основе алгоритма Эль-Гамаля;
* расшифрование на основе алгоритма Эль-Гамаля;
* оценка времени выполнения операций зашифрования и расшифрования.

1. **Методика выполнения поставленных задач**

**2.1. Алгоритм Эль-Гамаля**

Для реализации генерации ключевой информации, выполнены следующие действия.

Во-первых, простое число p, х определяет пользователь. Параметр *х* определяется в диапазоне [1; *p*-1]. Во-вторых, уже приложение выбирает число *g*, такое что *g*<*p* и является первообразным по модулю числа *p*. Т.е., степени числа *g* дают все возможные по модулю *p* остатки, которые взаимно-просты с *р.*

После производим вычисление открытого ключа y по формуле

*y* = mod *p*. Теперь у нас есть вся необходимая ключевая информация (*p*, *g*, *x*, *y*) для осуществления операций зашифрования и расшифрования.

Зашифрования каждого отдельного блока исходного сообщения (в нашем случае 1 блок равен 1 символу) предусматривает использование некоторого случайного числа *k* (1<*k*<*p*-1).

На рисунке 2.1 изображена функция зашифрования.

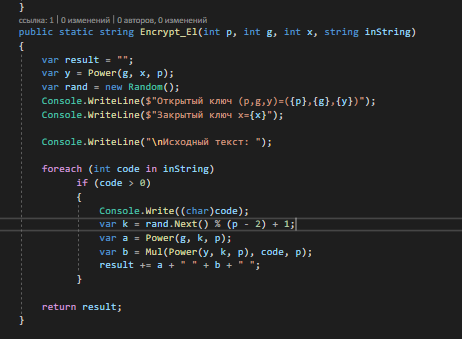


Рис. 2.1 – Реализация зашифрования

Полученный шифротекст следует расшифровать. Для этого вычислим каждый блок исходного сообщения по формуле: *m* = (*b*×(*a*)*p*-*x*-1)mod*p*. Реализация расшифрования представлена на рисунке 2.2.

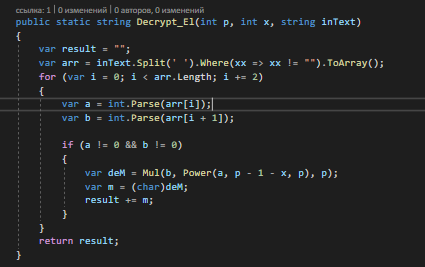


Рис. 2.2 – Реализация расшифрования

**2.2. Алгоритм RSA**

Далее рассмотрим программную реализацию зашифрования и расшифрования на основе алгоритма RSA. Данный алгоритм гораздо проще для понимания и реализации чем предыдущий.

Действие алгоритма заключается в следующих шагах.

Во-первых, пользователю необходимо ввести большие простые числа *p*, *q*. Желательно, чтобы они были равной длины, тогда алгоритм станет еще более криптостойким за счет того, что найти сомножители будет труднее. Если пользователь введет не простое число в поле ввода *p*, *q*, система предупредит его об этом.

Во-вторых, необходимо выбрать число *е*, взаимно простое с функцией Эйлера *φ*(*n*)=(*p*-1)(*q*-1). Пара (*e*,*n*) будет являться открытым ключом алгоритма.

В-третьих, необходимо вычислить число *d* по формуле Евлида. Пара (*d*,*n*) будет являться закрытым ключом алгоритма.

Рассмотрим алгоритм зашифрования сообщения. Каждый блок шифротекста вычисляется отдельно по формуле *c* = (*m*)*e*mod*n*, где *m* – блок исходного сообщения. Часть программного кода, непосредственно реализующая данные вычисления, представлена на рисунке 2.3.

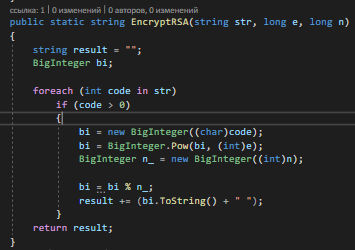


Рис. 2.3 – Реализация зашифрования

Расшифрование проводится подобным образом по формуле *m* = (*c*)*d* mod *n*, с использованием закрытого ключа *d*. Функция расшифрования изображена на рисунке 2.4.

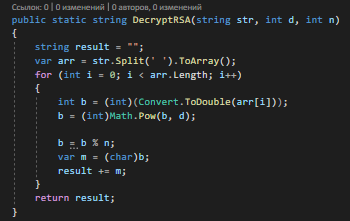


Рис. 2.4 – Реализация расшифрования

Результат работы всего приложения представлен на рисунке 2.5.

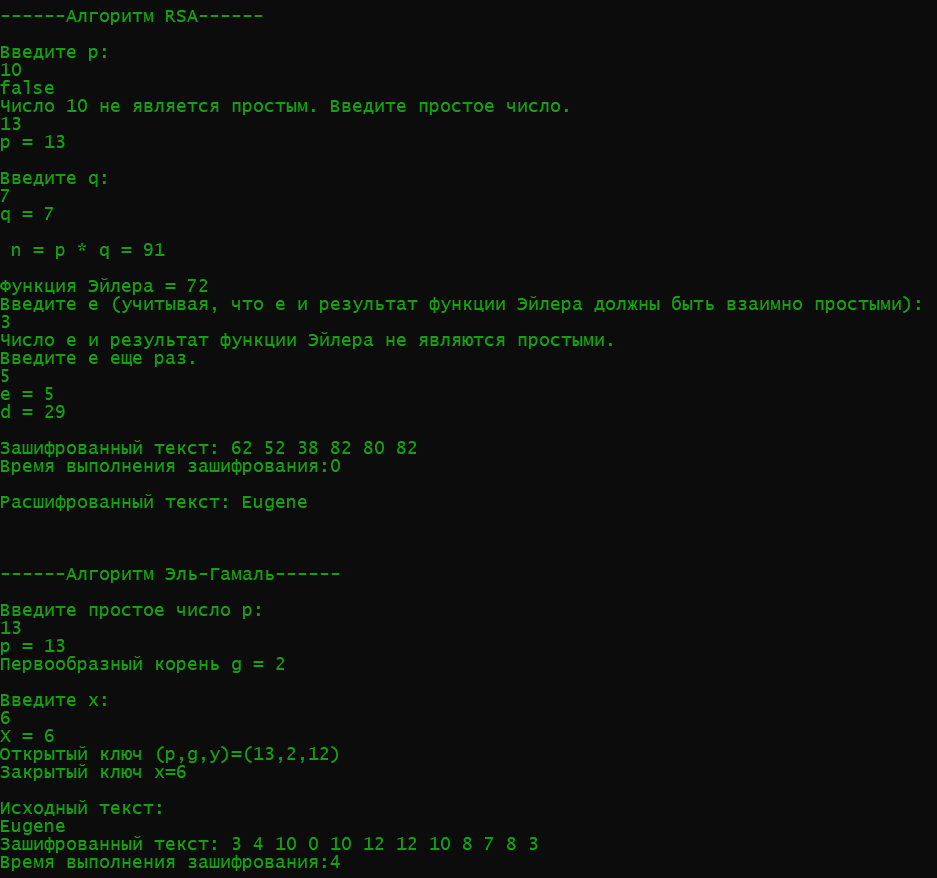


Рис 2.5 – Результат работы приложения

Проделаем то же самое несколько раз, дабы построить график. Результат представлен на рисунках 2.6 и 2.7.

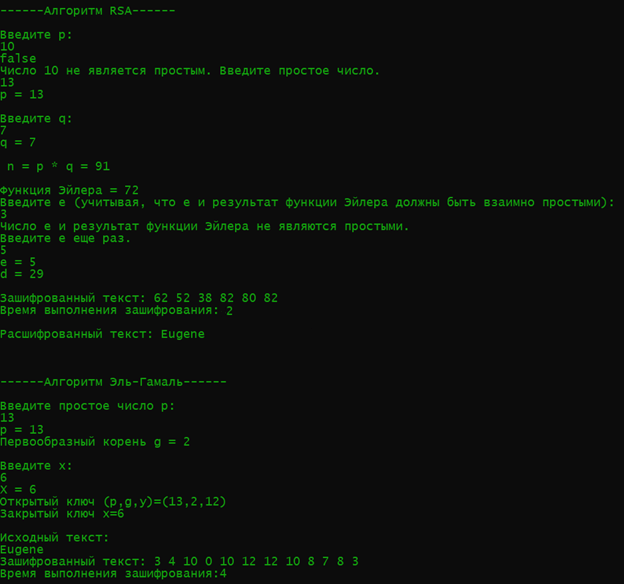


Рис 2.6 – Результат работы приложения

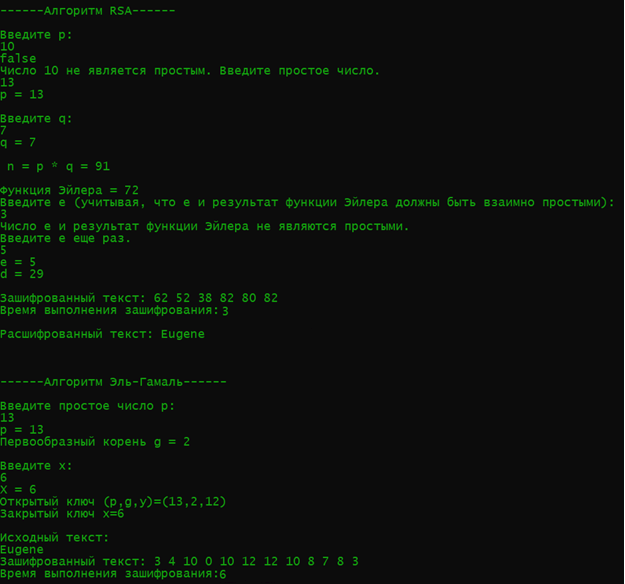


Рис 2.7 – Результат работы приложения

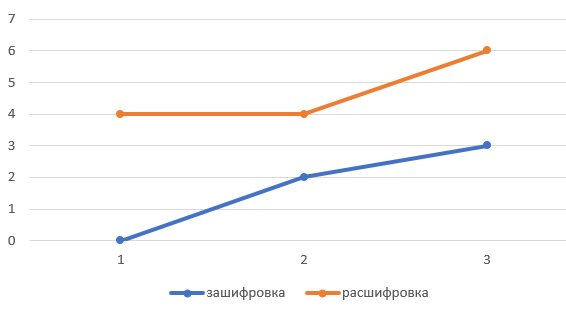


Рис 2.8 – График среднего времени работы

1. **Используя примерно одинаковый порядок ключевой информации, оценить производительность обоих алгоритмов и относительное изменение объемов криптотекстов**

Добавим к исходному тексту ещё немного текста, например, отчество. На рисунке 2.9. посчитаем количество символов при зашифровке текста «Eugene» при RSA: получается 6 чисел на 6 буков. Если же зашифруем «Eugene Nikolaeva Vladimirovna» получится на 27 буков 27 чисел. Потому получается на 1 букву 1 число.

При Эль-Гамале на 6 буков получается 12 чисел. Так же на 27 буков получается 60 чисел. Из этого можно сделать вывод, что при Эль-Гамале получается больше чисел, чем буков изначально примерно в 2 раза!

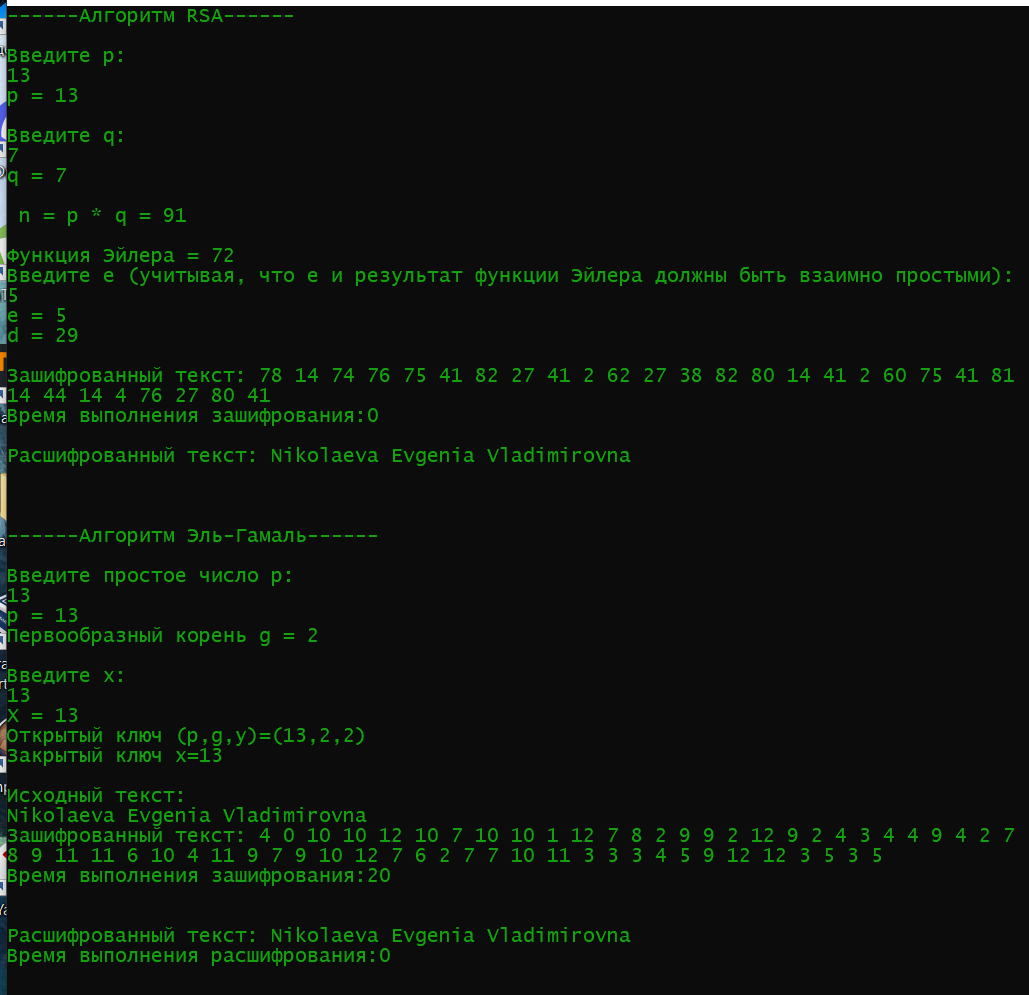


Рис 2.9– Результат работы приложения при увеличении текста

**Вывод**

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