Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Исследование асимметричных шифров

RSA и Эль-Гамаля

Студент: Димитриади А.В.

ФИТ 3 курс 5 группа

Преподаватель:

Савельева Маргарита Геннадьевна

Минск 2023

## 1. Описание приложения

Приложение позволяет выполнить следующие задачи:

* 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.

## 2. Методика выполнения поставленных задач

## 2.1. Алгоритм Эль-Гамаля

Для реализации генерации ключевой информации, выполнены следующие действия.

Разработанное ПС генерирует простое число *p* случайным образом в диапазоне [3; 1000]. Реализация процесса генерации числа *p* продемонстрирована на рисунке 2.1.

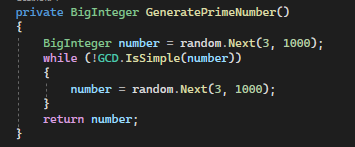


Рисунок 2.1 – Генерация числа p

Нам следует выбрать число *g*, такое что *g*<*p* и является первообразным по модулю числа *p*. Т.е., степени числа *g* (*g*i, 1≤ i ≤ *p*-1) дают все возможные по модулю *p* остатки, которые взаимно-просты с *р*, реализация данной функций представлена на рисунке 2.1

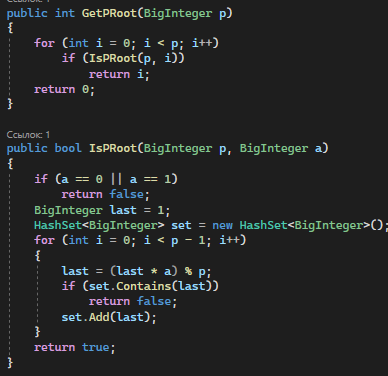


Рисунок 2.2 – Генерация числа *p*

Также необходимо сгенерировать закрытый ключ х случайным образом в диапазоне [2; *p*-1], т.е. *x*<*p*, который будет использоваться для дальнейших вычислений.

Далее производим вычисление открытого ключа *y* по формуле  
*y* = *g*X 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).

Блок шифротекста (ci) состоит из двух чисел и :

= *gk* mod *p*

= *y*k *×* mod *p*

Вычисление всех указанных выше действий зашифрования реализованы в функции, представленной на рисунке 2.3.

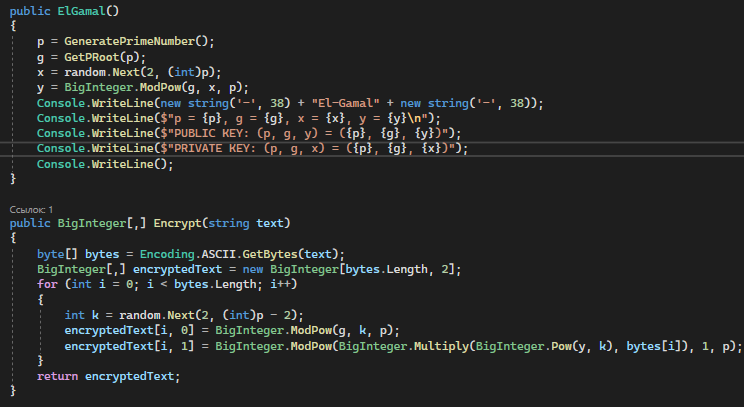


Рисунок 2.3 – Реализация зашифрования

Полученный шифротекст следует расшифровать. Для этого вычислим каждый блок исходного сообщения по формуле: = ( ×*p-x*-1) mod *p*. Реализация расшифрования представлена на рисунке 2.4.

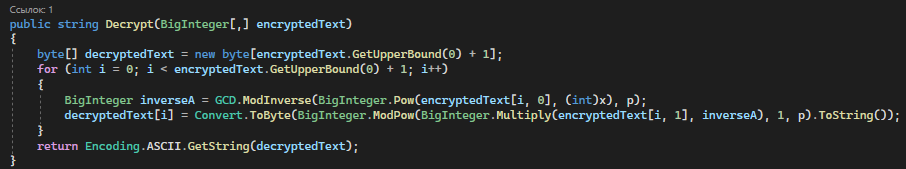


Рисунок 2.4 – Реализация расшифрования

## 2.2. Алгоритм RSA

RSA (Rivest-Shamir-Adleman) - это асимметричный криптографический алгоритм, широко используемый для шифрования, создания цифровых подписей и обмена ключами. Алгоритм RSA был разработан в 1977 году Рональдом Ривестом, Ади Шамиром и Леонардом Адлеманом.

Основная идея алгоритма RSA основана на сложности факторизации больших целых чисел. Алгоритм использует два ключа: открытый ключ (public key) и закрытый ключ (private key). Открытый ключ используется для шифрования данных, а закрытый ключ – для расшифровки данных или создания цифровой подписи.

Основные шаги алгоритма RSA:

1. Генерация ключей:

* Выбираются два различных простых числа *p* и *q*.
* Вычисляется их произведение *n* = *p* × *q*, которое является модулем шифрования.
* Вычисляется значение функции Эйлера от числа n: φ(n) = (*p* -1) × (*q* -1).
* Выбирается целое число *e* (1 < *e* < φ(*n*)), взаимно простое с φ(n). Это становится открытым ключом.
* Вычисляется значение секретного числа *d*, обратного e по модулю φ(*n*), так что (*d* × *e*) mod φ(*n*) = 1. Это становится закрытым ключом.

1. Шифрование данных:

* Каждому символу сообщения присваивается числовое значение, которое меньше n.
* Вычисляется шифротекст с использованием открытого ключа: ciphertext = (plaintext^*e*) mod *n*.

1. Расшифровка данных:

* Вычисляется исходный текст с использованием закрытого ключа: plaintext = (ciphertext^ *d*) mod *n*.

1. Цифровая подпись:

* Чтобы создать цифровую подпись сообщения, применяется процесс хэширования (например, с использованием алгоритма SHA-256).

Алгоритм RSA является одним из основных алгоритмов криптографии и широко применяется для защиты информации во многих системах. Однако он требует значительных вычислительных ресурсов для операций с большими числами, поэтому часто используется для шифрования ключей или малых объемов данных, в то время как само сообщение шифруется симметричными алгоритмами, такими как AES.

На рисунке 2.5 алгоритм функций, используемых для генерации простых чисел и взаимно простых чисел, которые необходимы для создания ключей в алгоритме RSA. Простые числа используются для генерации модуля, а взаимно простые числа используются для выбора открытого ключа.

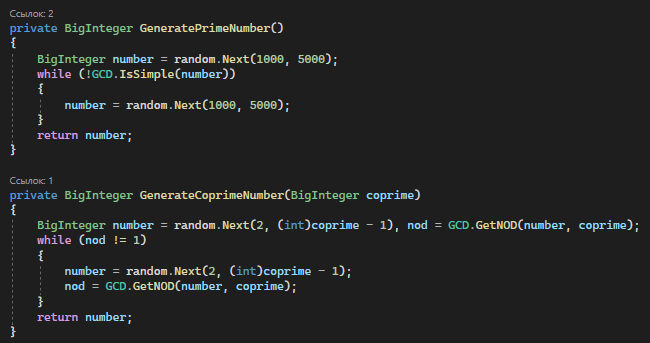


Рисунок 2.5 – Реализация функций для генерации простых чисел

Функция шифрования представлена на рисунке 2.6.

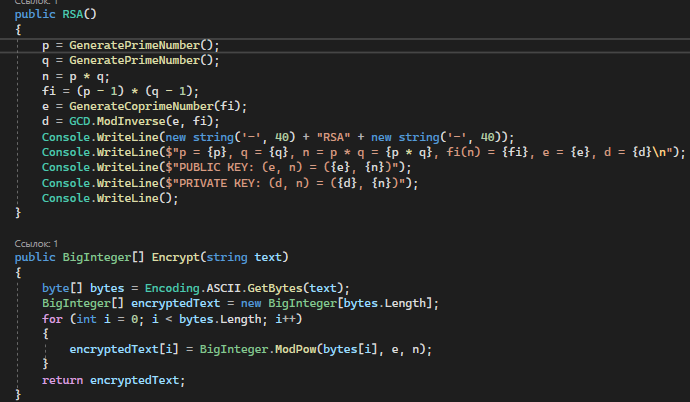


Рисунок 2.6 – Реализация функции шифрования

Функция дешифрования представлена на рисунке 2.7.

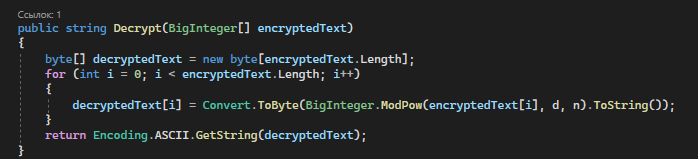


Рисунок 2.7 – Реализация функции дешифрования

Результат работы приложения представлен на рисунке 2.8.

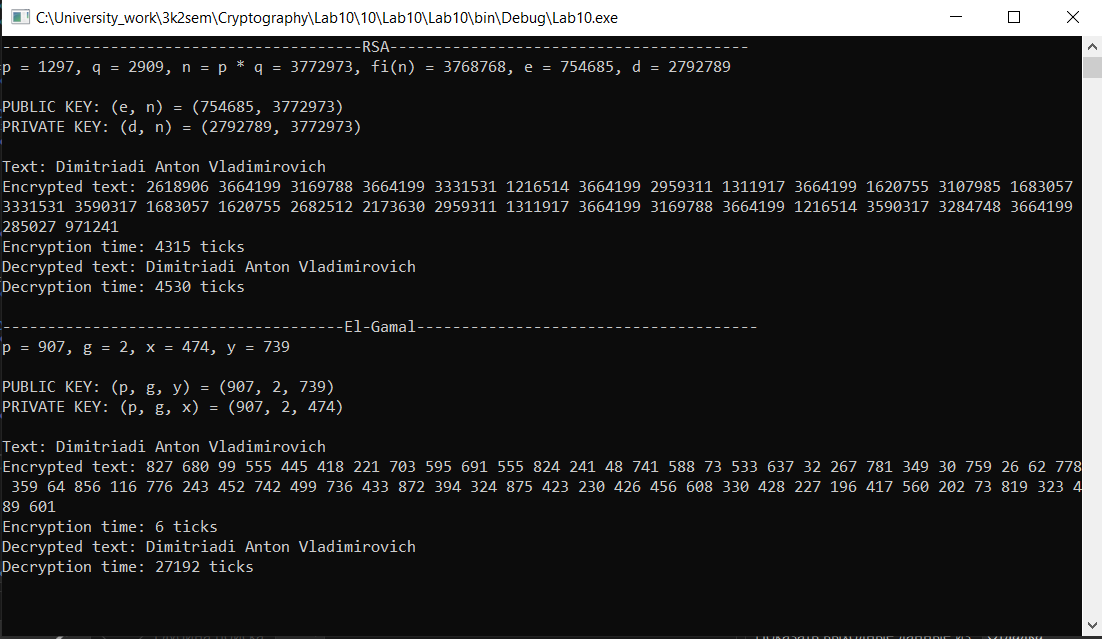


Рисунок 2.8 – Результат работы приложения

Ниже представлен оценочный график времени выполнения функций шифрования и расшифрования данных алгоритмов, рисунок 2.9.

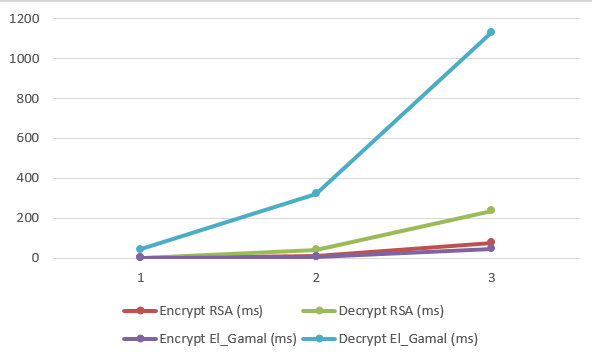


Рисунок 2.9 – Оценка времени выполнения алгроитмов

По графику можно сказать то что, очень много времени занимает расшифровка Эль-Гамаля, связано с тем что при шифровании Эль-Гамаль использует две опирации шифрования, что значительно увеличивает время его рассшифрования ввиду вычислительных сложностей.

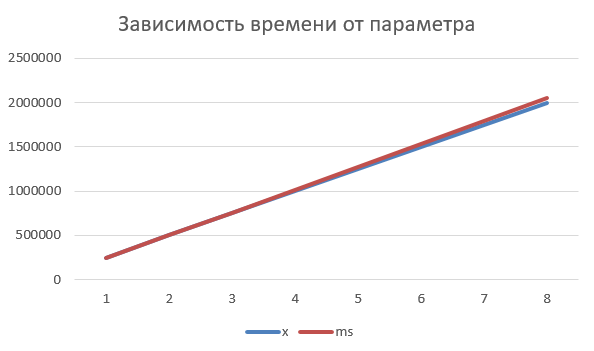


Рисунок 2.10 – Зависимость времени от параметров

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

Если посчитать количество символов на рисунке 2.8, то можно увидеть, в RSA на один букву приходится примерно 1 число, в Эль-Гамале же около двух. На рисунке 3.1. посчитаем количество символов при зашифровке текста «Dimitriadi» при RSA: получается 10 чисел на 10 букв. Потому получается на 1 букву 1 число.

При Эль-Гамале на 10 букв получается 20 чисел. Из этого можно сделать вывод, что при Эль-Гамале получается больше чисел, чем буков изначально примерно в 2 раза!

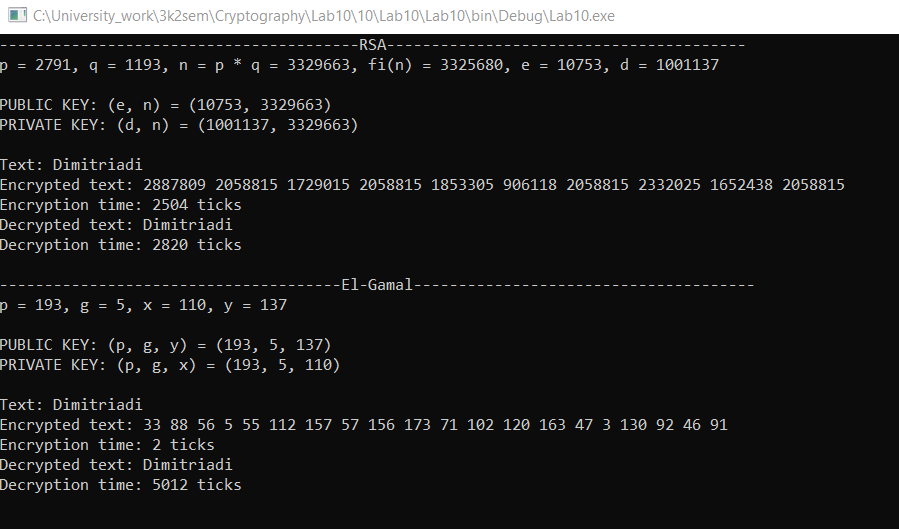
****

Рис 3.1– 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.