Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Исследование ассиметричных шифров RSA и Эль-Гамаля**

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 асимметричных шифров RSA и Эль-Гамаля.

1. **Задание**

1. С помощью простого консольного приложения составить табличную или графическую форму зависимости времени вычисления параметра *у*, функционально заданного выражением вида: *у* ≡ *ax* mod *n*, от параметров: *а* (десятичные числа от 5 до 35; можно взять 1 или 2 числа), *х* (числа, желательно простые, из диапазона от 103 до 10100; для примера взять 5–10 чисел, равномерно распределенных в указанном диапазоне), *n* (для примера взять числа, в двоичном виде состоящие из 1024 и 2048 битов).

2. Разработать авторское оконное приложение в соответствии с целью лабораторной работы. При этом можно воспользоваться доступными библиотеками либо программными кодами. В основе вычислений – кодировочные таблицы Base64 и ASCII. Приложение должно реализовывать следующие операции:

• зашифрование и расшифрование текстовых документов на основе алгоритмов RSA и Эль-Гамаля;

• определение времени выполнения операций.

Исходный текст для зашифрования – собственные фамилия, имя, отчество. Для численного представления блоков текста можно в том числе пользоваться указанными выше кодировочными таблицами. Ключевую информацию для обоих алгоритмов можно сгенерировать самостоятельно либо воспользоваться, например, одной из утилит криптографической библиотеки OpenSSL, с помощью которой, в частности, можно сгенерировать ключевую информацию для алгоритма RSA.

3. Используя примерно одинаковый порядок ключевой информации, оценить производительность обоих алгоритмов и относительное изменение объемов криптотекстов (по отношению к объемам открытых текстов).

1. **Ход работы**

**Теоретическая часть**

**Математические основы асимметричных шифров**

Асимметричная криптография основана на сложности решения некоторых математических задач. По существу, таких задач две:

– разложение больших чисел на простые сомножители (задача факторизации);

– вычисление дискретного логарифма в конечном поле, а также вычислительные операции над точками эллиптической кривой

Все системы асимметричного зашифрования основаны:

– на проблеме факторизации (RSA)

– на проблеме дискретного логарифма (Эль-Гамаля)

**RSA**

Безопасность RSA основана на *трудности разложения на множители больших чисел*. Открытый и закрытый ключи являются функциями двух больших простых чисел. Предполагается, что восстановление открытого текста по шифртексту и открытому ключу эквивалентно разложению на множители двух больших чисел.

Используются два больших случайных числа *p* и *q* (для максимальной криптостойкости нужно выбирать их равной длины)для генерации тайного и открытого ключа (по сути – двух взаимосвязанных частей одного ключа, т.е. ключа, принадлежащего одному физическому лицу). Рассчитываем *n* = *pq.* Ключ состоит из *n*, *e*, *d*.

Выбираем случайным образом *e,* такое что *e* и (*p* – 1)(*q* – 1) являются взаимно простыми числами.

Затем находим *d*:





В результате получаем ключ из трех чисел, которые образуют две взаимосвязи  
публичный ключ (*e*, *n*); тайный ключ (*d*, *n*).

Зашифрование RSA: если шифруется сообщение *М*, состоящее из r блоков: , , …, , …, , то шифртекст С будет состоять из такого же числа (*r*) блоков, представляемых числами:



Расшифрование RSA:



**Эль-Гамаля**

Как подчеркивалось выше, безопасность алгоритма Эль-Гамаля, как и безопасность алгоритма Диффи – Хеллмана, основана на *трудности вычисления дискретных логарифмов*.

Алгоритм Эль-Гамаля фактически использует схему Диффи – Хеллмана, чтобы сформировать общий секретный ключ для абонентов, передающих друг другу сообщение, и затем сообщение шифруется путем умножения его на этот ключ.

*Генерация ключевой информации:*

Выбирается простое число *р*. Выбирается число (*g*, *g* < *p*), являющееся первообразным корнем числа *р* – очень важный элемент с точки зрения безопасности алгоритма.

Далее выбирается число *х* (*х* < *p*) и вычисляется последний компонент ключевой информации:



Отправка сообщения, зашифрованного ключами *p*, *g*, *y*. Расшифровка сообщения ключами *p*, *g*, *x*. Как видим, тайным является только *x*.

При больших *p* нахождение *x* затруднено.

Алгоритм Эль-Гамаля предназначен для шифрования сообщений и создания цифровой подписи. Этот алгоритм основан на сложности вычисления дискретного логарифма в конечном поле.

Хотя в теории возможно использование алгоритма Эль-Гамаля для расшифрования сообщений, это не является его основной функцией и не является практически осуществимым в общем случае. Для расшифрования сообщений, зашифрованных алгоритмом Эль-Гамаля, необходимо знать закрытый ключ, который был использован для шифрования сообщения.

Поэтому алгоритм Эль-Гамаля обычно используется только для шифрования сообщений и создания цифровой подписи, а не для расшифровки сообщений, зашифрованных им.

**Практическая часть**

Для первого задания в качестве параметров были выбраны следующие значения:

*a* – [5, 15];

*x* – [103, 1000, 5000, 10000, 100000];

*n* – [21024, 22048].

Далее, по формуле *у* ≡ *ax* mod *n* были вычислены значения *y*, а также было оценено время подсчёта этих значений, которое мы можем увидеть на графике 3.1.

Рисунок 3.1 – Графическая форма зависимости времени вычисления параметра *у*

Для второго задания были разработаны классы ElGamal и Rsa для реализации соответствующих алгоритмов шифрования. Их коды представлены на рисунках 3.2 – 3.3.

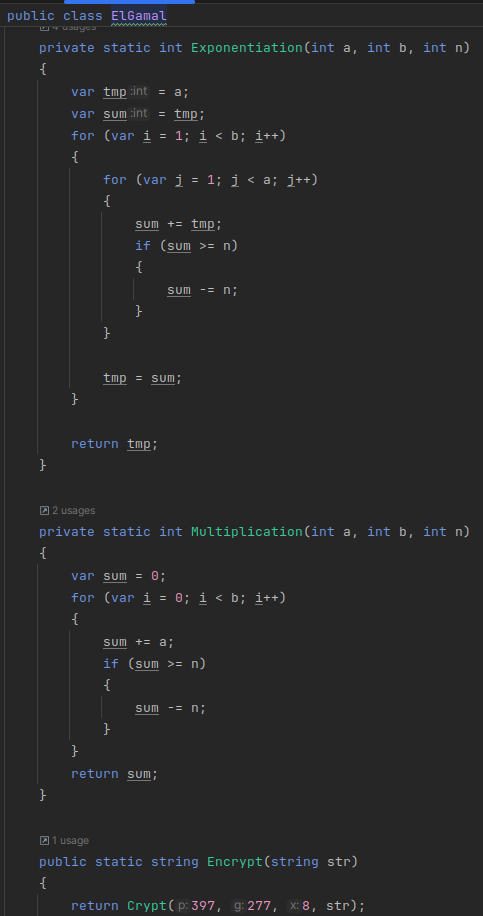


Рисунок 3.2 – Класс ElGamal

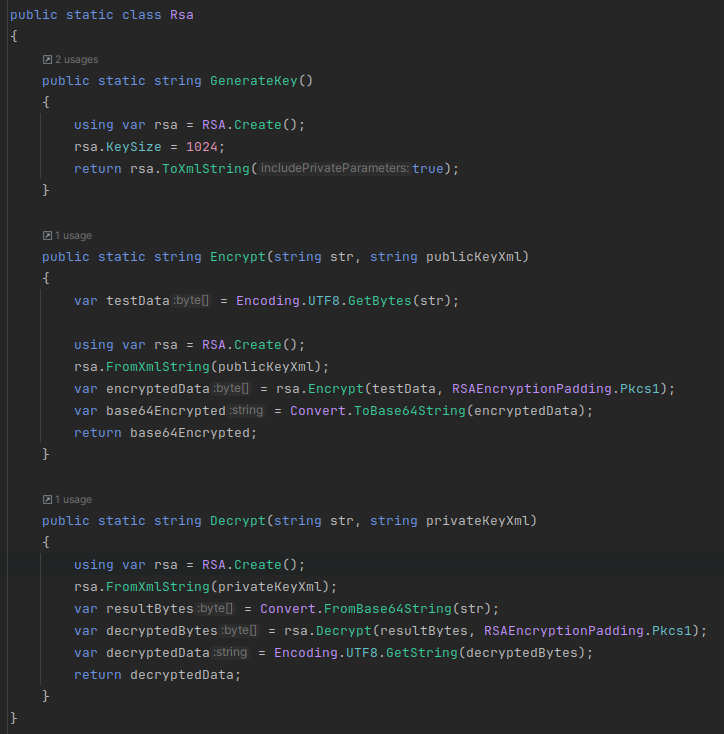


Рисунок 3.3 – Класс Rsa

Шифруемые строки в обоих случаях – «Gvozdovskiy Kirill Vladimirovich». Тут же была оценена производительность обоих алгоритмов и относительное изменение объемов криптотекстов (по отношению к объемам открытых текстов). В результате при использовании двух шифров получим следующий вывод, представленный на рисунке 3.4.

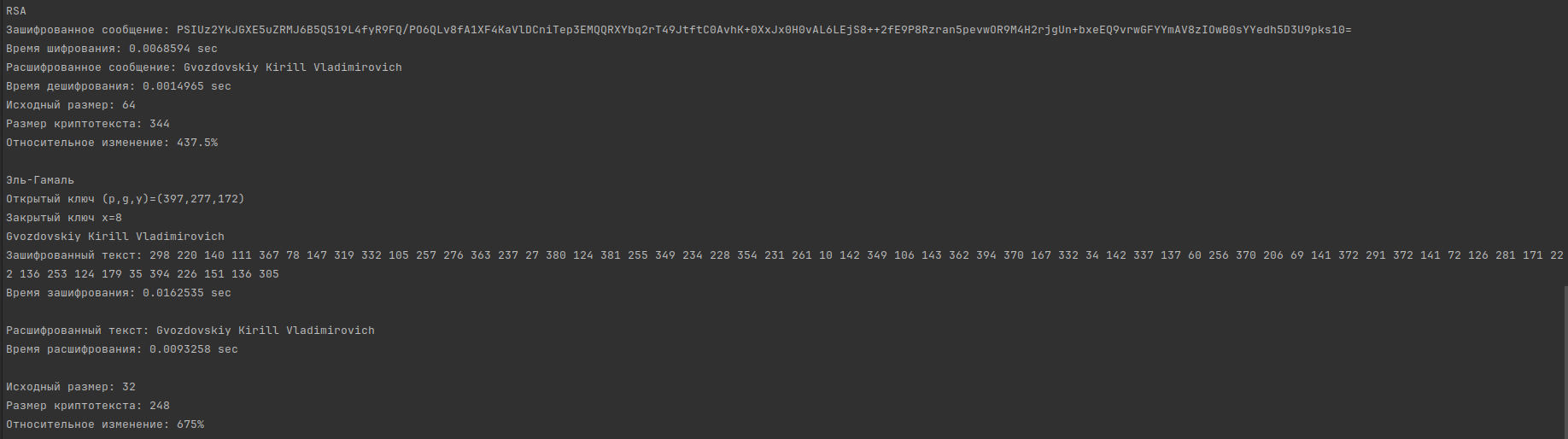


Рисунок 3.4 – Результаты шифрования и расшифрования для Base64

Данный вывод был получен для кодировки Base64. Теперь сравним его с выводом для кодировки ASCII, представленном на рисунке 3.5.

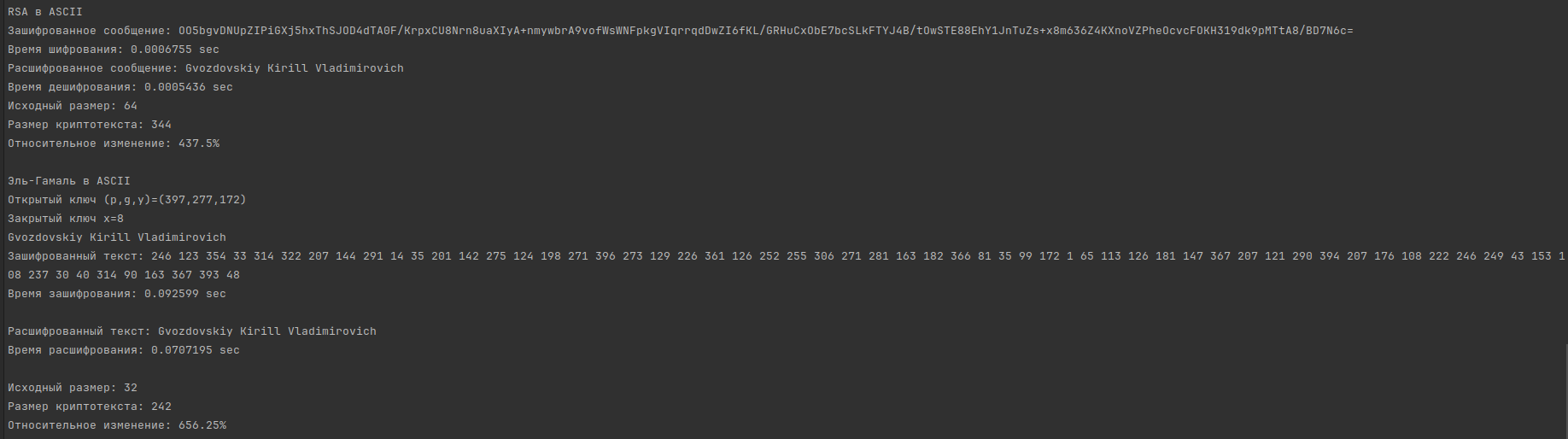


Рисунок 3.5 – Результаты шифрования и расшифрования для ASCII

Исходя из полученных результатов, можно увидеть, что кодировка ASCII справляется быстрее. Это обусловлено тем, что ASCII представляет символы непосредственно в виде однобайтовых значений, тогда как Base64 кодирует данные в последовательность символов, используя шесть битов из каждого байта и добавляя символы заполнения. Расшифровка ASCII-кодированного текста требует простого сопоставления каждого символа с его однобайтовым значением, что делает процесс очень быстрым и эффективным.

Кроме того, посмотрим на графики 3.6 – 3.9, демонстрирующие производительность обоих алгоритмов в разных кодировках:

Рисунок 3.6 – Оценка скорости шифрования и дешифрования для RSA с использованием Base64

Рисунок 3.7 – Оценка скорости шифрования и дешифрования для RSA с использованием Base64

Рисунок 3.8 – Оценка скорости шифрования и дешифрования для RSA с использованием Base64

Рисунок 3.9 – Оценка скорости шифрования и дешифрования для RSA с использованием Base64

**Вывод**

В ходе лабораторной работы были изучены и приобретены практические навыки разработки и использования приложений для реализации асимметричных шифров RSA и Эль-Гамаля.

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