Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Криптографические методы защиты информации

Студент: Сенченя В.И.

ФИТ 3 курс 5 группа

Преподаватель: Савельева М. Г.

Минск 2023

**Лабораторная работа №10**

**Тема «****Исследование асимметричных шифров**

**RSA и Эль-Гамаля»**

Цель: изучение и приобретение практических навыков разработки и использования приложений для реализации асимметричных шифров RSA и Эль-Гамаля.

Задачи:

1. Закрепить теоретические знания по алгебраическому описанию, алгоритмам реализации операций зашифрования/расшифрования и оценке криптостойкости асимметричных шифров RSA и Эль-Гамаля.
2. . Разработать приложение для реализации асимметричного зашифрования/расшифрования на основе алгоритмов RSA и Эль-Гамаля.
3. Выполнить анализ криптостойкости асимметричных шифров RSA и Эль-Гамаля
4. Оценить скорость зашифрования/расшифрования реализованных шифров.
5. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.

**Ассиметричный шифр**

В основу асимметричной криптографии положена идея использовать ключи парами: один – для зашифрования (открытый, или публичный, ключ), другой – для расшифрования (тайный ключ). Отметим, что указанная пара ключей принадлежит получателю зашифрованного сообщения. Все алгоритмы шифрования с открытым ключом основаны на использовании односторонних функций, к числу которых, как известно, относится вычисление дискретного логарифма.

**RSA**

**RSA** – это асимметричный криптографический алгоритм, который используется для шифрования данных и подписи сообщений. Он основан на сложности факторизации больших целых чисел.

Алгоритм работает следующим образом:

1. Генерация ключей: сначала выбираются два больших простых числа *p* и *q*. Затем вычисляются их произведение *n* = *p*×*q*, которое называется модулем. Далее выбирается целое число e, которое является открытой экспонентой и взаимно просто с функцией Эйлера от *n*. Затем вычисляется число *d*, которое является секретной экспонентой и обратным по модулю функции Эйлера от n к открытой экспоненте *e*. Полученные пары чисел (*n*, *e*) и (*n*, *d*) являются открытым и секретным ключами соответственно.
2. Шифрование: для шифрования сообщения *M* используется открытый ключ (*n, e*). Сообщение *M* сначала преобразуется в целое число *m*, которое должно быть меньше, чем модуль *n*. Затем вычисляется шифрованное сообщение *C* по формуле: *C* = *me* mod *n*.
3. Дешифрование: для дешифрования сообщения *C* используется секретный ключ (*n, d*). Зашифрованное сообщение *C* сначала преобразуется в целое число *c*, которое также должно быть меньше, чем модуль *n*. Затем вычисляется дешифрованное сообщение *M* по формуле: *M* = *cd* mod *n*.

Пример кода для реализации RSA предоставлен на рисунке 1.1, а результат работы на рисунке 1.2.

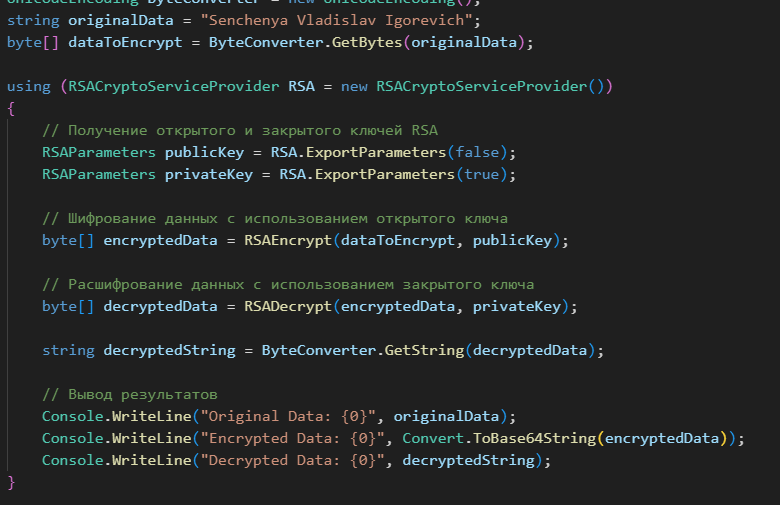


Рисунок 1.1 – Код RSA

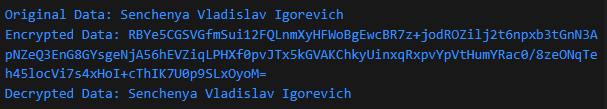


Рисунок 1.2 – Результат шифрования RSA

Скорость выполнения алгоритма предоставлена на рисунке 1.5

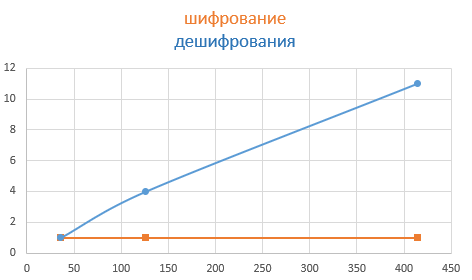


Рисунок 1.5 – Скорость выполнения алгоритма RSA

Важно заметить, что оценка скорости выполнения алгоритма RSA может быть сложным при наличии компьютера со слабыми вычислительными мощностями, так ка для шифрования большого объёма данных необходимо создавать ключи больших размеров.

**Эль-Гамаль**

**Эль-Гамаль** – это асимметричный алгоритм шифрования, основанный на сложности вычисления дискретного логарифма в конечных полях.

Алгоритм Эль-Гамаля использует два ключа: открытый и закрытый. Открытый ключ используется для шифрования сообщения, а закрытый ключ используется для расшифровки сообщения.

Алгоритм выполняется по следующему принципу:

1. Генерация параметров: *p* – простое число, *g* – примитивный корень по модулю *p*.
2. Генерация ключей: Открытый ключ вычисляется как *y* = *gx* mod *p*.
3. Шифрование: Для шифрования сообщения *m* отправитель выбирает случайное целое число *k* и вычисляет два числа:

*a* = *gk* mod *p*; *bi* = *mi* × *yk* mod *p*

Затем отправитель отправляет получателю пару (*a*, *b*) в качестве зашифрованного сообщения.

1. Дешифрование: Для дешифрования сообщения получатель использует свой закрытый ключ *x* и полученную пару (*a*, *b*). Он вычисляет:

*ax* = (*gk*)*x* = *g*(*kx*) mod *p*

Используя свой закрытый ключ *x*, получатель может вычислить обратную операцию и найти *yk* mod *p*:

(*ax*) -1 = *g*-*k* mod *p*

*b* × (*ax*)-1 = *m* × *yk* × *g*-*k* mod *p* = *m*

Код реализации алгоритма предоставлен на рисунке 1.6, а результат выполнения на рисунке 1.7.



Рисунок 1.6 – Код алгоритма Эль-Гамаля

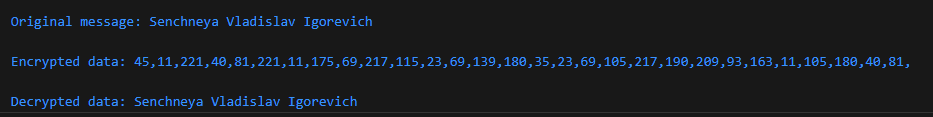


Рисунок 1.7 – Результат работы алгоритма Эль-Гамаля

Скорость шифрования и дешифрования предоставлена на рисунке 1.8.

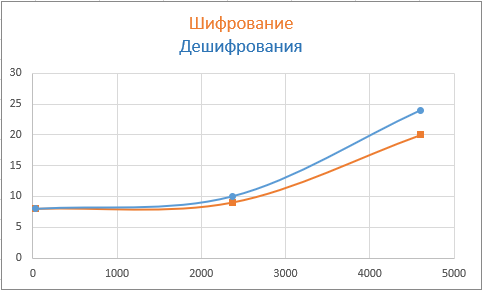


Рисунок 1.8 – Скорость выполнения алгоритма Эль-Гамаля.

**Вычисление параметра *y***

Параметр ***y*** вычисляется по следующей формуле:

*у* ≡ *ax* mod *n*

Код вычисления параметра ***y***предоставлен на рисунке 1.9, а скорость выполнения на рисунке 1.10.

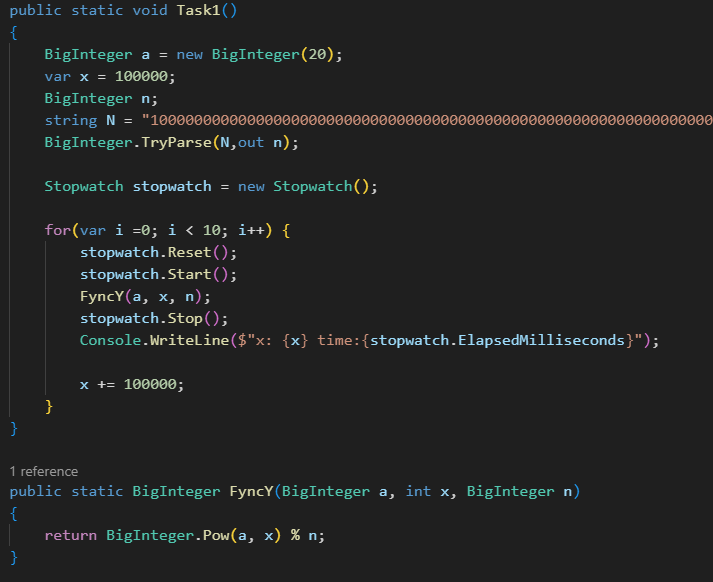


Рисунок 1.9 – Код вычисления параметра ***y***

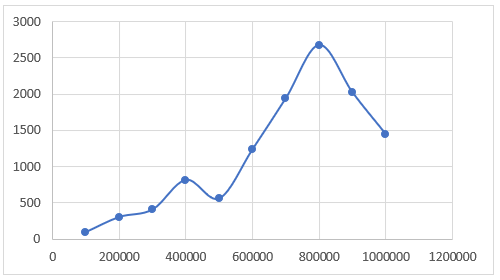


Рисунок 1.10 – Скорость вычисления параметра ***y***

**Сравнение алгоритмов**

При примерно одинаковых входных данных: 414 символов имеем следующее:

Скорость шифрования: 1мс и 4мс для RSA и Эль-Гамаля соответственно.

Скоро расшифрования: 10мс и 5мс для RSA и Эль-Гамаля соответственно.

Объём данных: 684 байт и 1047 байт для RSA и Эль-Гамаля соответственно.

**Вывод**

В ходе выполнения лабораторной работы я приобрёл и закрепил навыки практические навыки разработки приложения для реализации ассиметричных шифров RSA и Эль-Гамаля.