Учреждение образования

«Белорусский государственный технологический университет»

**Лабораторная работа №10**

Исследование ассиметричных шифров RSA и Эль-Гамаля

Выполнил:

Студент 3 курса 5 группы ФИТ

Коршун Никита Игоревич

2024

В данной лабораторной работе было разработано приложение на языке Python.

**Алгоритм RSA:** Для генерации двух ключей: тайного и открытого (а по сути – двух взаимосвязанных частей одного ключа, т. е. ключа, принадлежащего одному физическому лицу (или группе лиц), либо одному юридическому лицу), используются два больших случайных простых числа *p* и *q*. Для максимальной большей криптостойкости нужно выбирать *p* и *q* равной длины. Рассчитывается произведение: *n* = *pq*. Это есть один из трех компонент ключа, состоящего из чисел *n*, *e*, *d*.

Затем случайным образом выбирается второй компонент ключа (открытый ключ или ключ зашифрования, *e*, такой что *e* и (*p* – 1)(*q* – 1) являются взаимно простыми числами; вспомним, что (*p* – 1)(*q* – 1) = φ(*n*) – функция Эйлера).

Наконец, расширенный алгоритм Евклида используется для вычисления третьего компонента ключа: ключа расшифрования *d* такого, что выполняется условие:



Таким образом, сформирован ключ, состоящий из трех чисел, которые в свою очередь образуют две вышеупомянутые взаимосвязанные части: открытый (публичный) ключ (*e*, *n*) и тайный ключ (*d*, *n*; на самом деле, как видим, тайным здесь является лишь первое из пары чисел).

Для зашифрования/расшифрования используется ключ получателя: отправитель шифрует сообщение открытым ключом, а получатель расшифровывает шифртекст своим тайным ключом.

**Алгоритм Эль-Гамаля:** 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:



Владельцу сформированной ключевой информации, состоящей из 4 чисел, может посылаться некоторый шифртекст, созданный с использованием открытого ключа получателя: *p*, *g*, *y*. Расшифрование шифртекста получатель производит своим тайным ключом: *p*, *g*, *х*. Как видим, на самом деле тайным является лишь одно число (как и в RSA): *х*.

В качестве *x* пологаются числа от 100 тысяч до 1 миллиона, в качестве *n* – некоторое 1024-битное число, в качестве *a* – некоторое дробное число.

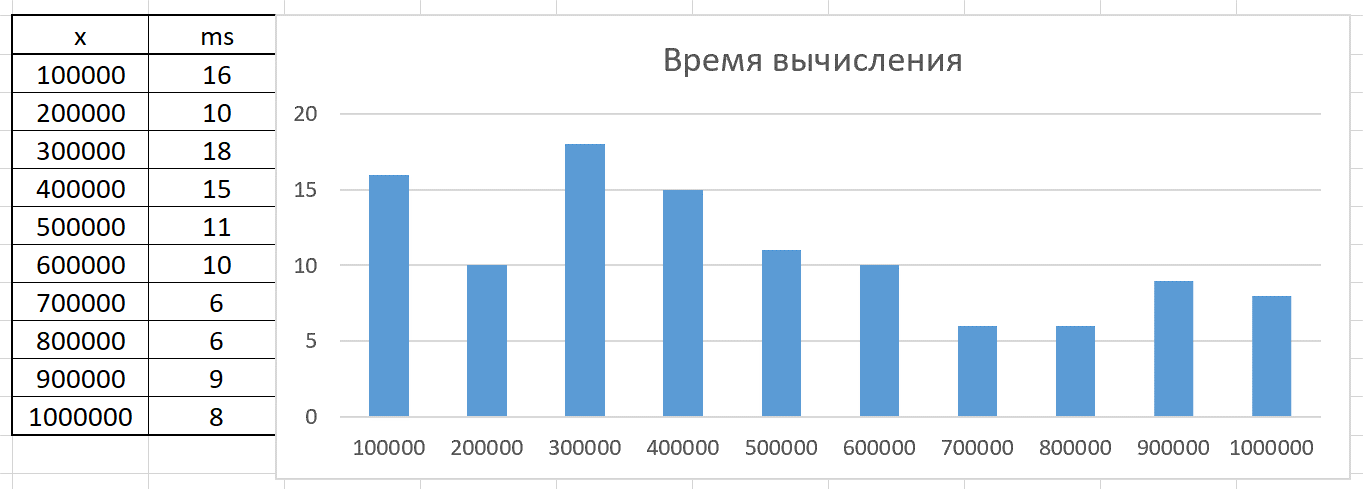


Рисунок 1 – График времени вычисления параметра *y*

Для сравнения алгоритмов RSA и Эль-Гамаля также было замерено время выполнения зашифрования и расшифрования.

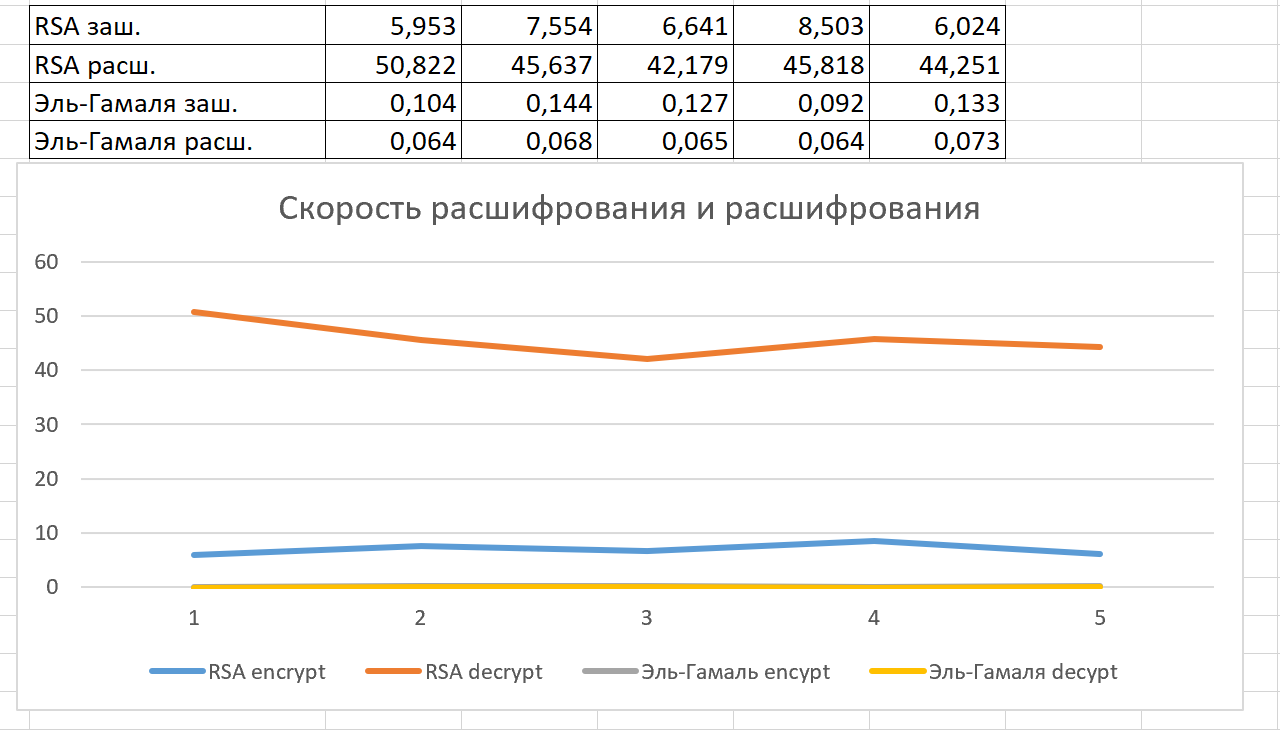


Рисунок 2 – График сравнения времени выполнения алгоритмов

Как видно из графика, при малых значениях входных текстов (примерно 100 символов) скорость выполнения алгоритма RSA сильно падает, что связано с тем, что значение ключа остается большим. Алгоритм же Эль-Гамаля показывает примерно линейную зависимость.

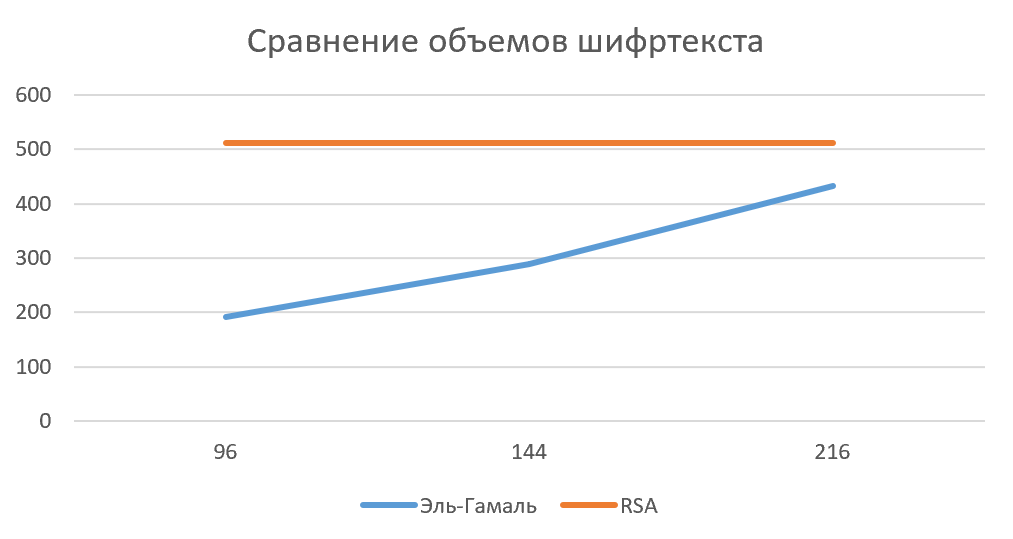


Рисунок 3 – График сравнения объемов шифртекста

Из сравнительного графика следует, что при использовании RSA шифротекст имеет длину 512. В алгоритме Эль-Гамаля длина шифротекста в два раза больше, чем длина открытого текста.

**Вывод:** Оба алгоритма основаны на сложности факторизации больших простых чисел и дискретного логарифмирования. RSA основан на задаче факторизации, а Эль-Гамаля – на задаче дискретного логарифмирования в конечных полях. По сегодняшним стандартам, при правильной выборе параметров, оба алгоритма считаются безопасными и устойчивыми к атакам.