**Московский авиационный институт**

# (национальный исследовательский университет)

Институт №8 «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

Дисциплина «Криптография»

# Лабораторная работа №1

Тема: Факторизация числа

|  |  |
| --- | --- |
| Студент: | Глушатов И.С. |
| Группа: | М8О-307Б-19 |
| Преподаватель: | Борисов А. В. |
| Дата: |  |
| Оценка: |  |

**Цель работы:** приобрести знания об алгоритмах разложения чисел на простые сомножители, попробовать написать алгоритм, производящий разложение быстрее наивного (за O()).

**Задание:**

Разложить каждое из чисел n1 и n2 на нетривиальные сомножители.

n1 = 108762353292448487441247663685513658893167646930627178946 128889967643172154127

n2 = 646002235431258257279334310060416308721637294121486556909 6184912826464912951934844843290548057971769185825571008898779827285829198579888230136431509729644521505820664210671796785408727881787088472858544237323843649456753195786360153338730014417334774860737788349377171270117783539517147103729867405974761065566886813619687414228558901198237125901551456589579117850298162443488215653711057874251971726449089251928969082516345596779536154854135917899503811275677859

**Ход работы**

Для факторизации первого числа после множества неудачных попыток в конечном счете я воспользовался интернет-ресурсом, который на локальном компьютере производил методы факторизации с помощью эллиптических кривых (ECM) и самоинициализирующегося квадратичного решета (SIQS).

В итоге понадобилось 2 минуты на моем компьютере, чтобы факторизовать 126-битное число с помощью вышеперечисленных методов:

108762 353292 448487 441247 663685 513658 893167 646930 627178 946128 889967 643172 154127 =

260 951289 862485 772644 727258 162652 873363 ×

416 791782 672403 295662 841737 728685 758229

Второе же число имеет длину в 1538 бит. Тестом Миллера – Рабина было проверено, что оно составное, однако за 5 часов ни один из возможных алгоритмов не дал никаких результатов, так что в предположении, что множители очень близки к корню и понимаю, что 1024-битные RSA ключи до сих пор не поддаются эффективному (и дешевому) способу факторизации, можно сделать вывод, что на данном этапе технического прогресса разложить второе число не представляется возможным.

В процессе поиска эффективных алгоритмов я реализовал на языке Python ρ-алгоритм Полларда, сложность которого O().

Сам алгоритм:

1. **def** Po\_Polard(N, xseed **=** None, yseed **=** None):
2. **if** IsPrime(N): **return** N
4. F **=** **lambda** x: (x**\*\***2 **-** 1) **%** N
5. x **=** randint(2, N) **if** xseed **==** None **else** xseed
6. y **=** randint(2, N) **if** yseed **==** None **else** yseed
7. g **=** gcd(abs(x **-** y), N)
8. **while** g **==** 1:
9. x **=** F(x)
10. y **=** F(F(y))
11. g **=** gcd(abs(x **-** y), N)
13. **return** g **if** g !**=** N **else** Po\_Polard(N)

**Выводы**

В ходе работы я узнал много алгоритмов факторизации (ECM, QS, алгоритмы Диксона и Полларда). Самостоятельно реализовал ρ - алгоритм Полларда, который смог найти делители 5, 6 и 9-го чисел Ферма. Ссылка на [GitHub](https://github.com/Igor743646/Krypt/blob/master/Lab1%20Kryptography.ipynb) с реализацией. Разложил лишь одно из двух чисел на простые множители.