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:	

Москва, 2022

Цель работы: приобрести знания об алгоритмах разложения чисел на простые сомножители, попробовать написать алгоритм, производящий разложение быстрее наивного (за O($\sqrt{n}$)).

Задание:
Разложить каждое из чисел n1 и n2 на нетривиальные сомножители.

n1 = 1087623532924484874412476636855136588931676469306271789461288899676431721541 27

n2 = 646002235431258257279334310060416308721637294121486556909
6184912826464912951934844843290548057971769185825571008898779
8272858291985798882301364315097296445215058206642106717967854
0872788178708847285854423732384364945675319578636015333873001
44173347748607377883493771712701177835395171471037298674059747
61065566886813619687414228558901198237125901551456589579117850
29816244348821565371105787425197172644908925192896908251634559
6779536154854135917899503811275677859

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($n^{\frac{1}{4}}$).
Сам алгоритм:

```
def Po_Polard(N, xseed = None, yseed = None):
    if IsPrime(N): return N

    F = lambda x: (x**2 - 1) % N
    x = randint(2, N) if xseed == None else xseed
    y = randint(2, N) if yseed == None else yseed
    g = gcd(abs(x - y), N)

    while g == 1:
        x = F(x)
        y = F(F(y))
        g = gcd(abs(x - y), N)

    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 с реализацией. Разложил лишь одно из двух чисел на простые множители.