Отчет по лабораторной работе №1 по курсу «Криптография»

Выполнил Моисеенков Илья Павлович, М8О-308Б-19.

Задание

Разложить каждое из чисел n1 и n2 на нетривиальные сомножители.

n1 =
37445690250873943521827325867122445734134840648853318819552882781962751323269

n2 =
63885323020856692286157713889834529480079417435681639705609468311573823440582391262298060528597378884877839378467209630230381530653482796708541490177874705748120068368851151068919108210525892260526854238146115200136060764411497599931538752584310398366878491477295583540652120665235922812546099439623062206418674291756938073911777134247594957005403707072783141530790649775750885101043734346727311116889093744079926530181765063441744614120970218748390138503050010 81

Ход работы

Для разложения первого числа я воспользовался готовой онлайн реализацией метода квадратичного решета. Этот метод считается вторым по быстроте (после общего метода решета числового поля). И до сих пор является самым быстрым для целых чисел до 100 десятичных цифр и устроен значительно проще чем общий метод решета числового поля. Разложение первого числа (78 разрядов) заняло примерно 10 минут.

Результат:
n1 = p1 * q1
p1 = 573733447260765268493955136095177485171
q1 = 652667025596202591708982935779112779639

Со вторым числом все оказалось гораздо интереснее. Оно состоит из 463 разрядов. Ради интереса я попробовал факторизовать это число тем же методом, но спустя сутки я не получил никакого результата. Я посмотрел на много реализаций различных алгоритмов, но ни одна из них не смогла разложить мое число даже спустя сутки.
Воспользовавшись подсказкой, что числа из вариантов строятся по похожей схеме, я решил прибегнуть к хитрости и поискать НОД своего числа и чисел из остальных вариантов. К счастью, мне удалось найти число, НОД с которым был отличен от единицы. Полученный НОД - первый множитель разложения. Второй множитель - мое число, деленное на этот НОД.

Результат:
n2 = p2 * q2
p2 =
20591237055704179681625834644880484498444984995772117549539485534190132602467078164211345327341051193704702361911450307821215708165717411436692397336069833

q2 =
31025490526883713875056003355383544708207861763406547230026302799803765947913761018263212606818945651730725821591247007256739142049786350520262700436477249659755262098977944592445788398277725363350244021408277779928960892571133203714735427492080530632886596390163063104426685307406376044277873086459725583825 7

Код

Приведу код, который помог мне в нахождении НОД.

```
NUMS = [] # все числа из других вариантов (в отчет не вставлял)

MY_NUM = n2  # число n2 моего варианта


import math

for num in NUMS:
    gcd = math.gcd(num, MY_NUM)
    if gcd != 1 and gcd != MY_NUM:
        print(MY_NUM)
        print('=')
        print(gcd)
        print('*')
        print(MY_NUM // gcd)
```

Выводы

Изначально работа показалась мне максимально простой. Я рассчитывал на то, что мне удастся найти какой-нибудь маленький делитель и разложить число на два множителя. Но проблема оказалась в том, что эти числа довольно огромные. И получены они перемножением двух больших простых чисел. Поэтому тривиальные алгоритмы не сработали.

Мне пришлось почитать про другие алгоритмы факторизации, которые работают для больших чисел. С помощью готовой реализации одного из таких алгоритмов - метода квадратичного решета - я смог получить разложение первого числа за 10 минут.

Второе число, как мне показалось, вообще почти нереально разложить на множители за приемлемое время. Задание мне удалось выполнить только прибегнув к хитрости.

Но благодаря этому заданию я понял, почему в криптографии так часто фигурируют простые числа - перемножив всего лишь два больших простых числа, мы получаем довольно большое и почти не факторизуемое число, которое можно использовать в различных алгоритмах шифрования.