Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №1 по курсу «Криптография»

Студент:	А. О. Дубинин
Преподаватель:	А. В. Борисов
Группа:	М8О-306Б
Дата:	
Оценка:	
Подпись:	

Москва, 2020

Вариант №5

Условие

Разложить каждое из чисел n1 и n2 на нетривиальные сомножители.

n1=274114822339589629024026495441557479713813228028980117869052278950681241194819
n2=1598756544210860812002683252504666631284038535154979340910964824673923578639226
39791813442919273700585418817797705917785824385599080398127566569091297553409104136
17018434655781017338634797816807916559595783204421083716340483743135242021931986948
94536452471646868825144743014452957912743920239954473534374422647748020165306769379
39619004459951311039306246130283924435675474106532077501151477472315586373159518289
2822790709843296375075272651902641460504103291775361

1 Метод решения

Изучив способы решения задачи, я посмотрел несколько алгоритмов и попробовал решить эту задачу основными алгоритмами разложения: Полларда p-1, Полларда p, Бента, Полларда Монте-Карло, Ферма. Их реализацию я взял с emaxx, но их эффективности не хватило, чтобы разложить даже первое число за короткое время. Прочитав на wikipedia, что в настоящее время самыми эффективными алгоритмами факторизации являются вариации решета числового поля, я нашел реализацию написанную Джейсоном Пападопулосом - msieve, которая уже в свою очередь разложила первое число за 3 минуты.

Второе же число я разложил с подсказки одногруппников, что первый множитель находится, как НОД с числом другого варианта, а второе число разложение путем деления моего числа на НОД. Чтобы найти это число, я написал небольшой скрипт на python, который парсит числа вариантов и проверяет НОД моего числа и числа другого варианта больше 1 или нет. По результату скрипта я узнал, что такое число одно, у которого НОД с моим больше 1. Это число n2 из варианта 6.

2 Исходный код

Мой скрипт, реализующий разложение второго числа.

```
from math import gcd

my_num = 15987565442108608120026832525046666312840
38535154979340910964824673923578639226397918134429
19273700585418817797705917785824385599080398127566
56909129755340910413617018434655781017338634797816
80791655959578320442108371634048374313524202193198
69489453645247164686882514474301445295791274392023
99544735343744226477480201653067693793961900445995
13110393062461302839244356754741065320775011514774
72315586373159518289282279070984329637507527265190
2641460504103291775361

var = 0
num = 0

with open('nums.txt') as f:
    for line in f:

        if line[0] == 'n':
            num = int(line[3: -2])
            gcd_num = gcd(my_num, num)
            if num != my_num and gcd_num != 1:
                tmp = my_num // gcd_num
                print("  -  {}".format(var))
                print("n{} = {}\n".format(line[1], num))
                print("a = p{}: {}".format(len(str(gcd_num)), gcd_num))
                print("b = p{}: {}".format(len(str(tmp)), tmp))
                check = tmp * gcd_num
                print("\na * b = {}".format(check))
        elif line != '\n':
            var = line[0]
```

3 Консоль

```
art@mars:~/study/Cryptography/lab_1/msieve$ ./msieve -m -q 2962902402649544155
747971381322802898011786905227895068124119 4819
```

```
274114822339589629024026495441557479713813228028980117869052278950681241194819
p39: 328253845119913323621537864865768604393
p39: 835069646296005008810210549818386161483

art@mars:~/study/Cryptography/lab_1$ python main.py
Сопряженное число -вариант 6
n2 = 1611765569148804856242867384258680719850010286298191204635154152942043219
7290447526886147483136114545465725205417369977940016871273001825655775233013 74
5768986374654630793295442477747872835121549831617371165626457442345657277097 46
3641140055832315479670230254145694131224473280404169708453094322175307224333 41
5061668790581352676527375610862399155982339310065668240742080964683365204046 93
8632685331174477299911625792360364160144090922283544048098857799988000765501 37

a = p309: 1633976960658210746809026559968255701597067952360459065215594609625 7
8519078856105725648968556569072711406616529723182939501812794722662366814883 63
1619640072792920581850719503493330646427755230896373119814690571985811278115 57
7251609542362580175148578313739080898244696381665260084479643389434792645421 90
8712913

b = p154: 9784449736468975763201829278312387831168780624139266884140660021121 9
0235927562589775919671584561568156563279255861754301952215856620403312006237 02
40875697

Проверка
a * b = 1598756544210860812002683252504666631284038535154979340910964824673923
5786392263979181344291927370058541881779770591778582438559908039812756656909 12
9755340910413617018434655781017338634797816807916559595783204421083716340483 74
3135242021931986948945364524716468688251447430144529579127439202399544735343 74
4226477480201653067693793961900445995131103930624613028392443567547410653207 75
0115147747231558637315951828928227907098432963750752726519026414605041032917 75
361
```

4 Ответ

Разложение первого числа:

- 32825384511991332362153786486576860 4393

- 83506964629600500881021054981838616 1483

Разложение второго числа:

- 16339769606582107468090265599682557015970679523604590652155946096257851907885610572564896855656907271140661652972318293950181279472266236681488363161964007279292058185071950349333064642775523089637311981469057198581127811557725160954236258017514857831373908089824469638166526008447964338943479264542190871 2913

- 97844497364689757632018292783123878311687806241392668841406600211219023592756258977591967158456156815656327925586175430195221585662040331200623702408756 97

5 Выводы

Решение этой лабораторной работы оказалось очень познавательным, ведь я познакомился с очень интересной темой - факторизацией чисел. Узнал что, по основной теореме алгебры, такое разложение существует для любого натурального числа, причем единственное. Так же я узнал, что задача факторизация является вычислительно сложной и именно поэтому она используется в разных алгоритмах криптографии, в том числе в RSA.

Я посмотрел множество алгоритмов по разложению, которые показались совсем не очевидными на первый взгляд. Делая вывод по этим алгоритмам, я понял, что наука криптография - это далеко не простая вещь, а алгоритмы факторизации - сложны и очень инетересны.