Отчёт по лабораторной работе №2 по курсу «Криптография»

Выполнил Попов Матвей, группа М8О-308Б-20

Задание

Разложить число на нетривиальные сомножители. Ниже представлены 16 вариантов. Вариант выбрать следующим образом: свое ФИО подать на вход в хеш-функцию, являющуюся стандартом, выход хеш-функции представить в шестнадцатеричном виде и рассматривать младший разряд как номер варианта. В отчете привести подробности процесса вычисления номера варианта.

Ход работы

Первым делом я занялся определением своего варианта. Для этого я воспользовался сайтом convertstring.com и подал на вход хеш-функции своё ФИО (Попов Матвей Романович). Последней цифрой полученного числа была 4, значит мне предстоит разложить следующее число:

`309086911254871141538991434992575166692891121664241483573664946812 1`

Сначала я попробовал разложить заданное число наивным алгоритмом, перебрав в цикле все меньшие числа, начиная с 2:

```
import logging
import datetime
import time
import math

n = 309086911254871141538991434992575166692891121664241483573664946812 1

logging.basicConfig(filename='log.txt', level=logging.INFO, format='%(asctime)s %(message)s')

for p in range(2, n):
    if n % p == 0:
        q = n / p
        logging.info('ANSWER: p = {}   q = {}'.format(p, q))
        break

    if datetime.datetime.now().minute == 0 and datetime.datetime.now().second == 0:
        progress = p // math.sqrt(n)
        logging.info('Last calculated p = {}   progress = {}'.format(p, progress))
        time.sleep(1)
```

Я предвидел, что вычисления займут долгое время, поэтому предусмотрел запись последнего вычисленного числа p в файл с логами раз в час. Примерно за полчаса программа успела перебрать примерно $2 \cdot 10^9$ чисел (возможно проверка времени на каждой итерации цикла сильно замедляет вычисления). После недолгих подсчётов выяснилось, что результат будет получен в худшем случае примерно через 10^{50} лет, что намного больше, чем время, отведённое на выполнение лабораторной работы. Пришлось искать другой способ.

Необходимую задачу решает сайт cryptool.org, с его помощью и была получена факторизация заданного числа:

```
p = 166792310943277237652975071130135 3
q = 185312446063527160984695930894705 7
```

Вывод

Проделав лабораторную работу, я на собственном опыте убедился, что крайне сложно найти нетривиальные сомножители большого числа, а значит такие разложения могут успешно использоваться в операциях шифрования и дешифрования в системах с открытым ключом.