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](https://www.convertstring.com/Hash/SHA512) и подал на вход хеш-функции своё ФИО (Попов Матвей Романович). Последней цифрой полученного числа была 4, значит мне предстоит разложить следующее число:

3090869112548711415389914349925751666928911216642414835736649468121

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 = 3090869112548711415389914349925751666928911216642414835736649468121

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 в файл с логами раз в час. Примерно за полчаса программа успела перебрать примерно чисел (возможно проверка времени на каждой итерации цикла сильно замедляет вычисления). После недолгих подсчётов выяснилось, что результат будет получен в худшем случае примерно через лет, что намного больше, чем время, отведённое на выполнение лабораторной работы. Пришлось искать другой способ. Необходимую задачу решает сайт [cryptool.org](https://www.cryptool.org/en/cto/msieve), с его помощью и была получена факторизация заданного числа:

p = 1667923109432772376529750711301353

q = 1853124460635271609846959308947057

# 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.