**Московский авиационный институт**
**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»
Кафедра: 806 «Вычислительная математика и программирование»
Дисциплина: «Криптография»

# Лабораторная работа № 3

## Тема: Факторизация больших полупростых чисел

Студент: Чекменев В.А.
Группа: М8О-307Б-20
Дата:

Москва, 2022

**Инструменты**
1. MSIEVE - https://sourceforge.net/projects/msieve/
2. SHA256 - https://emn178.github.io/online-tools/sha256.html

**Условие**
Разложить число на нетривиальные сомножители. 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**
1. ФИО – «**Чекменев Вячеслав Алексеевич**», используя инструмент (2) вычисляем хэш по алгоритму SHA256 для представленной строки.
Младший байт – “39” – это и есть вариант.
2. Рассматриваемое число N
**48533303631306565605539497719370479530746468784245625071289554074903757860268286619766927965793565110278775252365042652419046005688936219238021425862628365128873934162866880649102186317501232264661800402098993834278929250844354853806477625069689207660909962959363877659832295298520772907296659325610873524871 7**.
3. Факторизацию данного числа невозможно произвести «в домашних условиях» на персональном компьютере. Для факторизации даже 512 битного числа потребуется около одного дня работы алгоритма (информация с сайта GGNFS). Поэтому был использован трюк с поиском GCD между числами в других вариантах.
4. Для поиска НОД был написан скрипт на питоне (файл nums.txt содержит все числа вариантов):

```
from math import gcd
numbers = list()
f = open('nums.txt', 'r')
for l in f.readls():
    numbers.append(int(l))
k = 0
for n1 in numbers:
    for n2 in numbers:
        gcd_ = gcd(n1, n2)
        k += 1
        if gcd_ != 1 and n1 // gcd_ != 1:
            print(n1,' = ',gcd_,' * ',n1 // gcd_)
            break
```

5. В выводе программы увидим, что наш ответ:
P =
28064627448753481898903135859199518277708808203129435728315744150235355516844349811560523823848441800847516187696539243949832105942496666711

140752980920074451253492632978177914950175914039830144722836798214666442310385502977001724 29

Q = 172934074111367618180521976192071266425193880321305491080671096147117367476673

6. Проверку результата возможно произвести посредством проверки **P*Q=N**. Проверка успешна.