# ЛАБОРАТОРНАЯ РАБОТА №1

“Атака на алгоритм шифрования RSA посредством метода Ферма”

по дисциплине

‘Информационная безопасность’

Вариант 15

***Выполнил:***
Соболев Иван Александрович
**Группа:** P34312

***Преподаватель:***
Маркина Татьяна Анатольевна

Санкт-Петербург, 2024

# Цель работы

Изучить атаку на алгоритм шифрования RSA посредством метода Ферма.

# Программные и аппаратные средства

Для выполнения лабораторной работы был использован компьютер со следующими характеристиками:

- Процессор: Apple M2
- Видеокарта: Apple M2
- Объем оперативной памяти: 8GB
- Использована операционная система: macOS 14.4.1
- Версия Python: 3.12

# Задание

| Вариант | Модуль, N | Экспонента, e | Блок зашифрованного текста, C |
|---|---|---|---|
| 15 | 67510894259489 | 3543923 | 1834956116931<br>7762509478845<br>22384877417897<br>36443182878894<br>61287041306052<br>17680469174617<br>14632055288035<br>23212409940234<br>45782556562975<br>7533626343287<br>14537172455552<br>60777304839141 |

# Листинг разработанной программы

main.py

```
from RSA import decrypt_block, calculate_phi, find_d_component
from ferma import ferma_attack
from io_utils import int_to_bytes, print_green, print_red, read_config


def decrypt_ciphertexts(N, e, ciphertexts, p, q):
    """Дешифрует список шифротекстов и возвращает расшифрованные байты."""
    bytes = []
    phi = calculate_phi(p, q)
```

```
    print(f'Результат вычисления функции Эйлера: {phi}')
    d = find_d_component(e, phi)
    print(f'Результат вычисления параметра d: {d}\n')
    for c in ciphertexts:
        decrypted_block = decrypt_block(d, c, N)
        bytes.append(int_to_bytes(decrypted_block))
    decrypted_bytes = b''.join(bytes)
    return decrypted_bytes


def main():
    N, e, ciphertexts = read_config('config.json')
    p, q = ferma_attack(N)
    print_green(f"Результат факторизации Ферма:")
    print(f"p = {p}, q = {q}\n")

    decrypted_bytes = decrypt_ciphertexts(N, e, ciphertexts, p, q)
    try:
        plaintext = decrypted_bytes.decode('cp1251')
        print_green("Расшифрованный текст:")
        print(plaintext)
    except UnicodeDecodeError:
        print_red("Ошибка декодирования сообщения:")
        print(decrypted_bytes)


if __name__ == "__main__":
    main()
```

Io_utils.py

```
import json

def int_to_bytes(m):
    """Метод конфертации чисел в байты"""
    hex_str = hex(m)[2:]
    if len(hex_str) % 2:
        hex_str = '0' + hex_str
    return bytes.fromhex(hex_str)


def read_config(file_path):
    """Метод чтения параметров из конфигурационного файла."""
    with open(file_path, 'r') as f:
        config = json.load(f)
    return config['N'], config['e'], config['ciphertexts']


def print_green(message: str) -> None:
    """Метод для вывода ключа"""
    print(f"\033[92m{message}\033[0m")
```

```
def print_red(message: str) -> None:
    """Метод для вывода расшифрованного текста"""
    print(f"\033[91m{message}\033[0m")
```

ferma.py

```
import math

def ferma_attack(n):
    """Метод для проведения факторизации Ферма"""
    a = math.isqrt(n)
    b2 = a * a - n

    while b2 < 0 or not is_square(b2):
        a += 1
        b2 = a * a - n

    b = math.isqrt(b2)

    p = a - b
    q = a + b

    return p, q


def is_square(x):
    """Метод проверки числа на квадрат"""
    s = int(math.isqrt(x))
    return s * s == x
```

RSA.py

```
def decrypt_block(d, c, N):
    """Метод дешифрации блока"""
    return pow(c, d, N)


def find_d_component(e, phi):
    """Метод вычисления параметра закрытого ключа"""
    return pow(e, -1, phi)


def calculate_phi(p, q):
    """Метод вычисления функции Эйлера"""
    return (p - 1) * (q - 1)
```

config.json

```
{
  "N": 67510894259489,
  "e": 3543923,
  "ciphertexts": [
    1834956116931,
```

```
  7762509478845,
  22384877417897,
  36443182878894,
  61287041306052,
  17680469174617,
  14632055288035,
  23212409940234,
  45782556562975,
  7533626343287,
  14537172455552,
  60777304839141
 ]
}
```

## Промежуточные вычисления

$\sqrt{N} = 8216501$

Пусть a = $\frac{p+q}{2}$, b = $\frac{p-q}{2}$

[$(a_i, a_i^2 - N)$] = (8216502, 10856515), (8216503, 27289520), (8216504, 43722527), (8216505, 60155536)

60155536 - полный квадрат => a = 8216505, b = 7756

p = a + b = 8216505 + 7756= 8224261
q = a - b = 8216505 - 7756= 8208749

φ(n) = (p - 1)(q - 1) = 67510877826480

$d = e^{-1} mod$ φ$(n)$ = 13087298491547

## Результаты работы программы

```
Результат факторизации Ферма:
p = 8224261, q = 8208749

Результат вычисления функции Эйлера: 67510877826480
Результат вычисления параметра d: 13087298491547

Расшифрованный текст:
подобной ситуации свидетельствует о потере паке_
```