**Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»**

**Факультет программной инженерии и компьютерной техники**

**Информационная безопасность**

# Лабораторная работа №1

# Атака на алгоритм шифрования RSA посредством метода Ферма

## Вариант 14

Студент: Мокров Семён Андреевич

P34121

Преподаватель: Маркина Татьяна Анатольевна

Санкт-Петербург 2023

# Содержание

# Цель работы

Изучить атаку на алгоритм шифрования RSA посредством метода Ферма.

# Задание

– ознакомьтесь с теорией, изложенной в [3]. («Взлом алгоритма RSA при неудачном выборе параметров криптосистемы»);
– получите вариант задания у преподавателя;
12
– используя разложение модуля на простые числа методом Ферма и полученные исходные данные, определите следующие показатели:
– множители модуля (p и q);
– значение функции Эйлера для данного модуля $\varphi(N)$ ;
– обратное значение экспоненты по модулю $\varphi(N)$ ;
– дешифруйте зашифрованный текст, исходный текст должен быть фразой на русском языке;
– результаты и промежуточные вычисления оформите в виде отчета.

## Вариант задания

| | | | |
|---|---|---|---|
| 14 | 701091213690 29 | 3401467 | 65044661056628<br>62698810905915<br>6384243931214<br>64581496145197<br>34821902367398<br>47317941132118<br>31834994240307<br>32916261351098<br>27399527764660<br>20797651714466<br>56226270748693 |
| | | | 51223181240405 |

# Листинг разработанной программы

```
import math

from termcolor import colored

source: str = '''
65044661056628
62698810905915
6384243931214
64581496145197
34821902367398
47317941132118
31834994240307
32916261351098
27399527764660
20797651714466
56226270748693
51223181240405'''

N: int = 70109121369029
e: int = 3401467

print(colored("Стартовые данные:", "green"))
print("Зашифрованная строка: " + source)
print("N: " + str(N))
print("e: " + str(e))

list_source = source.split()

A = int(math.floor(math.sqrt(N)) + 1)
i = 1

while True:
    current_A = A + i
    current_B = math.sqrt(current_A ** 2 - N)

    if int(current_B) == float(current_B):
```

```
        break

    i += 1

print(colored("\nПолученный аргумент A: ", "green") + str(current_A))
print(colored("Полученный аргумент B: ", "green") + str(current_B))


p = current_A + current_B
q = current_A - current_B

eilor = (p - 1) * (q - 1)
d = pow(e, -1, int(eilor))

result: str = ""

print(colored("\nИтоговые результаты:", "green"))
print("p: " + str(p))
print("q: " + str(q))
print("Функция Эйлера: " + str(eilor))
print("d: " + str(d))

print(colored("\nРасшифровка сообщения:", "green"))

for sym in list_source:
    dec_sym_int = pow(int(sym), d, N)
    dec_sym_bytes = int.to_bytes(dec_sym_int, length=4, byteorder='big')
    dec_sym = dec_sym_bytes.decode('windows-1251')
    print(sym + " -> " + str(dec_sym_int) + " -> " + dec_sym)
    result += dec_sym

print(colored("\nПолученное сообщение: ", "green") + result)
```

# Исходный код

Исходный код расположен в репозитории:
**https://github.com/semwett0301/information-security**

# Результаты работы программы

## Скриншоты вывода программы

```
Стартовые данные:
Зашифрованная строка:
65044661056628
62698810905915
6384243931214
64581496145197
34821902367398
47317941132118
31834994240307
32916261351098
27399527764660
20797651714466
56226270748693
51223181240405
N: 70109121369029
e: 3401467

Полученный аргумент A: 8373123
Полученный аргумент B: 8210.0

Итоговые результаты:
p: 8381333.0
q: 8364913.0
Функция Эйлера: 70109104622784.0
d: 29002056932275
```

```
Расшифровка сообщения:
65044661056628 -> 4075872493 -> тран
62698810905915 -> 4059033328 -> спор
6384243931214 -> 4075679468 -> тном
64581496145197 -> 552857838 ->  уро
34821902367398 -> 3807241504 -> вне
47317941132118 -> 3975079141 -> моде
31834994240307 -> 3957858383 -> ли О
32916261351098 -> 1397304864 -> SI.
27399527764660 -> 3270436845 -> Возн
20797651714466 -> 3907710446 -> икно
56226270748693 -> 3806719464 -> вени
51223181240405 -> 3844104031 -> е __

Полученное сообщение: транспортном уровне модели OSI. Возникновение __
```

## Полученное сообщение

транспортном уровне модели OSI. Возникновение __

# Выводы

В данной лабораторной работе я:

- Ознакомился с методом шифрования данных RSA
- Ознакомился с методом факторизации Ферма
- Ознакомился с принципом взлома RSA при помощи метода Ферма
- Реализовал процесс взлома на Python