Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

Факультет программной инженерии и компьютерных технологий

Отчет по лабораторной работе № 1

“Атака на алгоритм шифрования RSA посредством метода Ферма”

по дисциплине Информационная безопасность

Вариант 10

|  |  |
| --- | --- |
| Студент группы № P34151 | Шипулин Павел Андреевич |
| Преподаватель | Маркина Татьяна Анатольевна |

Санкт-Петербург

2024

# Цель работы

Изучить атаку на алгоритм шифрования RSA посредством метода Ферма.

# Вариант задания

|  |  |  |  |
| --- | --- | --- | --- |
| **№**  **варианта** | **Модуль,** | **Экспонента,** | **Блок зашифрованного текста,** |
| 10 | 77027476849549 | 2936957 | 18937689886043  6667195679130  53238895771820  6189192838687  48623327840257  47264919314001  42510070950746  16878504505970  22744978157662  23644842894223  71614018816334  24651499733229 |

# Ход работы

1. Ознакомиться с теорией, изложенной в [3]. («Взлом алгоритма RSA при неудачном выборе параметров криптосистемы»).
2. Получить вариант у преподавателя
3. Используя разложение модуля на простые числа методом Ферма и полученные исходные данные, определить следующие показатели:
   1. Множители модуля ( и ).
   2. Значение функции Эйлера для данного модуля .
   3. Обратное значение экспоненты по модулю.
4. Дешифровать зашифрованный текст, исходный текст должен быть фразой на русском языке.
5. Результаты и промежуточные вычисления оформить в виде отчета.

# Листинг программ

Ссылка на репозиторий:

<https://github.com/PashcalE2/IS/tree/main/cryptography/second_part>

## Файл lab1.py

import math  
  
  
def lab1(N: int, e: int, C: str):  
 print("Рассчет параметров")  
 n = int(math.sqrt(N) + 1)  
 print(f"n = int(sqrt({N})) + 1 = {n}")  
  
 t = n  
 while True:  
 # Перебор t >= n  
 t += 1  
 sub = t \*\* 2 - N  
  
 sqrt\_sub = int(math.sqrt(sub))  
 if sqrt\_sub \*\* 2 == sub:  
 break  
  
 p = t + sqrt\_sub  
 print(f"p = {t} + {sqrt\_sub} = {p}")  
  
 q = t - sqrt\_sub  
 print(f"q = {t} - {sqrt\_sub} = {q}")  
  
 phi = round((p - 1) \* (q - 1))  
 print(f"ф(N) = {p - 1} \* {q - 1} = {phi}")  
  
 d = pow(e, -1, phi)  
 print(f"d = {e}^(-1) mod {phi} = {d}")  
  
 print("Дешифровка")  
 result = ""  
 for i, c in enumerate(C.split()):  
 num\_block = pow(int(c), d, N)  
 print(f"num\_block\_{i} = {c}^{d} mod {N} = {num\_block}")  
  
 text\_block = num\_block.to\_bytes(4, byteorder="big").decode("cp1251")  
 print(f"text\_block = {text\_block}")  
  
 result += text\_block  
  
 print(f"Результат = {result}")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 """  
 Вариант 10  
 """  
  
 \_N = 77027476849549  
 \_e = 2936957  
 \_C = """  
 18937689886043  
 6667195679130  
 53238895771820  
 6189192838687  
 48623327840257  
 47264919314001  
 42510070950746  
 16878504505970  
 22744978157662  
 23644842894223  
 71614018816334  
 24651499733229  
 """  
  
 lab1(\_N, \_e, \_C)

# Выполнение

## Результат выполнения программы

Рассчет параметров

n = int(sqrt(77027476849549)) + 1 = 8776530

p = 8776535 + 9474 = 8786009

q = 8776535 - 9474 = 8767061

ф(N) = 8786008 \* 8767060 = 77027459296480

d = 2936957^(-1) mod 77027459296480 = 8540915045653

Дешифровка

num\_block\_0 = 18937689886043^8540915045653 mod 77027476849549 = 4075692279

text\_block = то ч

num\_block\_1 = 6667195679130^8540915045653 mod 77027476849549 = 3908168686

text\_block = исло

num\_block\_2 = 53238895771820^8540915045653 mod 77027476849549 = 552592880

text\_block = пер

num\_block\_3 = 6189192838687^8540915045653 mod 77027476849549 = 3856982242

text\_block = едав

num\_block\_4 = 48623327840257^8540915045653 mod 77027476849549 = 3773164795

text\_block = аемы

num\_block\_5 = 47264919314001^8540915045653 mod 77027476849549 = 4112578792

text\_block = х ши

num\_block\_6 = 42510070950746^8540915045653 mod 77027476849549 = 4042189550

text\_block = роко

num\_block\_7 = 16878504505970^8540915045653 mod 77027476849549 = 3806722528

text\_block = веща

num\_block\_8 = 22744978157662^8540915045653 mod 77027476849549 = 4075154428

text\_block = тель

num\_block\_9 = 23644842894223^8540915045653 mod 77027476849549 = 3992712480

text\_block = ных

num\_block\_10 = 71614018816334^8540915045653 mod 77027476849549 = 4024494821

text\_block = паке

num\_block\_11 = 24651499733229^8540915045653 mod 77027476849549 = 4075741791

text\_block = тов\_

Результат = то число передаваемых широковещательных пакетов\_

# Вывод

Узнал и проверил, что при неудачном значении в методе шифрования RSA, можно легко найти секретный ключ с помощью простого перебора чисел и дальнейшей проверки на полный квадрат выражения .