Федеральное государственное автономное образовательное учреждение высшего образования

Университет ИТМО

Дисциплина: Информационная безопасность (Криптография)

# Лабораторная работа 2.1

Вариант 12

**Работу выполнил студент группы P34111:**
Кривоносов Егор Дмитриевич

**Преподаватель:**
Маркина Татьяна Анатольевна

2022 г.

г. Санкт-Петербург

# Оглавление

# Цель работы

Изучить атаку на алгоритм шифрования RSA посредством метода Ферма.

# Задание

| Вариант | Модуль, N | Экспонента, e | Блок зашифрованного текста, C |
|---|---|---|---|
| 12 | 74701165267919 | 3145553 | 32035658541536<br>35242897170964<br>6268303368709<br>6877322610982<br>16329207109754<br>35007623593376<br>26715311593240<br>36220800128563<br>25019660581036<br>61639733671958<br>21186453949445<br>72477207535811 |

# Ход работы

1. Вычисляем n = [sqrt(N)] + 1.
   Видим сообщение “[error]”, которое значит, что N - не квадрат целого числа.
2. Вычисляем t1 = n + 1 и далее d1 = t1^2 – N.
3. Проверяем, является ли d1 квадратом целого числа аналогично первому шагу.
   Снова видим сообщение “[error]”
4. Вычисляем t2 = t1 + 1 и d2 аналогично шагу 2.
5. Повторяем вычисления пока не дойдем до квадрата целого числа т.е. пока не перестанем видеть сообщение “[error]”
6. Дойдя до d4 не получаем сообщения об ошибке.
7. Вычисляем квадратный корень из d4.
8. Вычисляем p = t4 + sqrt(d4).
9. Вычисляем q = t4 – sqrt(d4).
10. Вычисляем Phi(N) = (p – 1)(q – 1).
11. Вычисляем d, как обратный к e: d = e^(-1) mod Phi(N).
12. Построчно выполняем дешифрацию текста. На каждую строку блока C вычисляем M= C^d mod N5.
13. Переводим каждое число в текстовый вид text(M).

$n = \left[\sqrt{N}\right] + 1 = 8642984$
$t_1 = n + 1 = 8642984 + 1 = 8642985$
$w_1 = t_1^{\,2} - N = 24442306$

$t_2 = n + 2 = 8642984 + 2 = 8642986$
$w_2 = t_2^{\,2} - N = 41728277$

$t_3 = n + 3 = 8642984 + 3 = 8642987$
$w_3 = t_3^{\,2} - N = 59014250$

$t_4 = n + 4 = 8642984 + 4 = 8642988$
$w_4 = t_4^{\,2} - N = 76300225$

$p = t + \sqrt{w_4} = 8651723$
$q = t - \sqrt{w_4} = 8634253$
$\varphi(N) = (p - 1)(q - 1) = 74701147981944$
$d = e^{-1} \, mod \, \varphi(N) = 23647864249265$

$p$ и $q$ - множители модуля
$\varphi(N)$ - значение функции Эллера для данного модуля
$d$ - обратное значение экспоненты по модулю $\varphi(N)$

# Листинг разработанной программы

```
import math

N = 74701165267919
e = 3145553
C = '''
32035658541536
35242897170964
6268303368709
6877322610982
16329207109754
35007623593376
26715311593240
36220800128563
25019660581036
61639733671958
21186453949445
72477207535811
'''


def solver(N, e, C):
    print("==========================")
    print("========== DATA ==========")
    print("==========================")
    print(f"N = {N}")
    print(f"e = {e}")
    print(f"C = {C}")

    print("==========================")
    print("== SOLVE - method Ferma ==")
    print("==========================")
    n = int(math.sqrt(N) // 1 + 1)
    print(f"n = [sqrt(N)] + 1 = {n}")
    i = 0
    while True:
        i += 1
        t = n + i
        print(f"t{i} = n + {i} = {t}")
        w = t ** 2 - N
        print(f"w{i} = t{i} ^ 2 - N = {t ** 2} - {N} = {w}")
        sqrt_w = math.sqrt(w)
        if sqrt_w % 1 == 0:
            sqrt_w = int(sqrt_w)
            print(f"sqrt(w) = {sqrt_w}", "\n")
            break
        else:
            print(f"sqrt(w) = {sqrt_w} - error", "\n")

    p = t + sqrt_w
    q = t - sqrt_w
```

```
    phi = round((p - 1) * (q - 1))
    d = pow(e, -1, phi)

    print(f"p = t + sqrt(w) = {t} + {sqrt_w} = {p}")
    print(f"q = t - sqrt(w) = {t} - {sqrt_w} = {q}")
    print(f"Phi(N) = (p - 1) * (q - 1) = ({p - 1}) * ({q - 1}) = {phi}")
    print(f"d = e^(-1) mod Phi(N) = {e}^(-1) mod {phi} = {d}", "\n")

    message = ""
    for i, c in enumerate(C.split()):
        m = pow(int(c), d, N)
        part = m.to_bytes(4, byteorder='big').decode('cp1251')
        print(f'm{i} = C[{i}]^d mod N = {c}^{d} mod {N} = {m} => text({m}) = {part}')
        message += part
    print(f"message = {message}")


solver(N, e, C)
```

# Скриншоты работы программы Python

```
============================
========== DATA ==========
============================
N = 74701165267919
e = 3145553
C =
32035658541536
35242897170964
6268303368709
6877322610982
16329207109754
35007623593376
26715311593240
36220800128563
25019660581036
61639733671958
21186453949445
72477207535811
```

```
==========================
== SOLVE - method Ferma ==
==========================
n = [sqrt(N)] + 1 = 8642984
t1 = n + 1 = 8642985
w1 = t1 ^ 2 - N = 74701189710225 - 74701165267919 = 24442306
sqrt(w) = 4943.916059157963 - error

t2 = n + 2 = 8642986
w2 = t2 ^ 2 - N = 74701206996196 - 74701165267919 = 41728277
sqrt(w) = 6459.742796737344 - error

t3 = n + 3 = 8642987
w3 = t3 ^ 2 - N = 74701224282169 - 74701165267919 = 59014250
sqrt(w) = 7682.073287856606 - error

t4 = n + 4 = 8642988
w4 = t4 ^ 2 - N = 74701241568144 - 74701165267919 = 76300225
sqrt(w) = 8735

p = t + sqrt(w) = 8642988 + 8735 = 8651723
q = t - sqrt(w) = 8642988 - 8735 = 8634253
Phi(N) = (p - 1) * (q - 1) = (8651722) * (8634252) = 74701147981944
d = e^(-1) mod Phi(N) = 3145553^(-1) mod 74701147981944 = 23647864249265
```

```
m0 = C[0]^d mod N = 32035658541536^23647864249265 mod 74701165267919 = 3991269360 => text(3991269360) = непр
m1 = C[1]^d mod N = 35242897170964^23647864249265 mod 74701165267919 = 3772967147 => text(3772967147) = авил
m2 = C[2]^d mod N = 6268303368709^23647864249265 mod 74701165267919 = 4243451625 => text(4243451625) = ьной
m3 = C[3]^d mod N = 6877322610982^23647864249265 mod 74701165267919 = 552592880 => text(552592880) =  пер
m4 = C[4]^d mod N = 16329207109754^23647864249265 mod 74701165267919 = 3857841131 => text(3857841131) = есыл
m5 = C[5]^d mod N = 35007623593376^23647864249265 mod 74701165267919 = 3941081327 => text(3941081327) = ки п
m6 = C[6]^d mod N = 26715311593240^23647864249265 mod 74701165267919 = 3773490674 => text(3773490674) = акет
m7 = C[7]^d mod N = 36220800128563^23647864249265 mod 74701165267919 = 4007796781 => text(4007796781) = ов -
m8 = C[8]^d mod N = 25019660581036^23647864249265 mod 74701165267919 = 552595170 => text(552595170) =  пов
m9 = C[9]^d mod N = 61639733671958^23647864249265 mod 74701165267919 = 4075745517 => text(4075745517) = торн
m10 = C[10]^d mod N = 21186453949445^23647864249265 mod 74701165267919 = 4226097391 => text(4226097391) = ые п
m11 = C[11]^d mod N = 72477207535811^23647864249265 mod 74701165267919 = 3857769773 => text(3857769773) = ере-
message = неправильной пересылки пакетов - повторные пере-
```

**Полученный результат:** “неправильной пересылки пакетов - повторные пере-”

# Вывод

В ходе выполнения данной лабораторной работы я ознакомился с методом Ферма для атаки на алгоритм шифрования RSA и реализовал его работу на языке Python.

# Полезные ссылки

[Пример работы метода Ферма]

$N = p * q$, где p и q - взаимно простые числа
p>q:
$N = (\frac{p+q}{2})^2 + (\frac{p-q}{2})^2 \Rightarrow t = \frac{p+q}{2};\ w = \frac{p-q}{2}$