УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Лабораторная работа №2.1

Дисциплина «Информационная безопасность»

Вариант 13

Выполнил: студент группы P34131

Кузнецов Максим Александрович

Преподаватель:

Маркина Татьяна Анатольевна

Санкт-Петербург, 2023 г.

Цель работы

Изучить атаку на алгоритм шифрования RSA посредством метода Ферма.

Задание

Вариант	Модуль, N	Экспонента, e	Блок зашифрованного текста, C
13	72903890242273	3261683	37429454018574 65632293727338 71955235122455 71474662312159 18537435780920 58372142077460 68330829196451 60882917270796 24142764117328 31238010810556 66143215653810 30769266886306

Ход работы

1. Вычисляем $n = \lceil \sqrt{N} \rceil + 1$.
2. Вычисляем $t = n + 1$ и далее $d = t^2 - N$.
3. Проверяем, является ли *d* квадратом целого числа. Повторяем шаги 2-3, пока не получится квадрат целого числа.
4. Вычисляем $p = t + \sqrt{d}$.
5. Вычисляем $q = t - \sqrt{d}$.
6. Вычисляем $Phi(N) = (p - 1)(q - 1)$.
7. Вычисляем *d*, как обратный к *e*: $d = e^{-1}\ mod\ Phi(N)$.
8. Построчно выполняем дешифрацию текста. На каждую строку блока C вычисляем $M = C^d\ mod\ N$.
9. Переводим *M* в текстовый вид.

Для решения задачи была разработана программа на Python.

Листинг разработанной программы

```
import math

N = 72903890242273
e = 3261683
C = '''
37429454018574
65632293727338
71955235122455
71474662312159
18537435780920
58372142077460
68330829196451
60882917270796
24142764117328
31238010810556
66143215653810
30769266886306
'''


n = int(math.sqrt(N) // 1 + 1)
print(f"1. Шаг первый: n = [sqrt(N)] + 1 = {n}")

step = 0
while True:
        step += 1
        t = n + step
        d = t ** 2 - N
        sqrt = math.sqrt(d)
        print(f"2. Шаг второй, попытка ({step}): t = {t}, d = {d}, sqrt =
{sqrt}")
        if sqrt % 1 == 0:
            sqrt = int(sqrt)
            print("3. Шаг третий: получен квадрат целого числа.")
            break
        else:
            print("3. Шаг третий: повторение шага 2.")
            pass

p = t + sqrt
q = t - sqrt
```

```
phi = round((p - 1) * (q - 1))
d = pow(e, -1, phi)
print(f"4. Шаг четвертый: p = {p}")
print(f"5. Шаг пятый: q = {q}")
print(f"6. Шаг шестой: phi = {phi}")
print(f"7. Шаг седьмой: d = {d}")

output = ""

for i, c in enumerate(C.split()):
        m = pow(int(c), d, N)
        msg = m.to_bytes(4, byteorder='big').decode('cp1251')

        print(f'8. Шаг восьмой: {m} = {msg}')
        output += msg

print(f"9. Шаг девятый: итоговый текст --> {output}")
```

Результат работы программы:

```
1. Шаг первый: n = [sqrt(N)] + 1 = 8538378
2. Шаг второй, попытка (1): t = 8538379, d = 25705368, sqrt = 5070.046153636079
3. Шаг третий: повторение шага 2.
2. Шаг второй, попытка (2): t = 8538380, d = 42782127, sqrt = 6540.804766999241
3. Шаг третий: повторение шага 2.
2. Шаг второй, попытка (3): t = 8538381, d = 59858888, sqrt = 7736.852590039441
3. Шаг третий: повторение шага 2.
2. Шаг второй, попытка (4): t = 8538382, d = 76935651, sqrt = 8771.296996453832
3. Шаг третий: повторение шага 2.
2. Шаг второй, попытка (5): t = 8538383, d = 94012416, sqrt = 9696.0
3. Шаг третий: получен квадрат целого числа.
4. Шаг четвертый: p = 8548079
5. Шаг пятый: q = 8528687
6. Шаг шестой: phi = 72903873165508
7. Шаг седьмой: d = 16406932632835
8. Шаг восьмой: 4059818986 = сылк
8. Шаг восьмой: 3894472160 = и на
8. Шаг восьмой: 552792288 =  тра
8. Шаг восьмой: 3992055790 = нспо
8. Шаг восьмой: 4042452462 = ртно
8. Шаг восьмой: 3961582576 = м ур
8. Шаг восьмой: 4007849445 = овне
8. Шаг восьмой: 539828463 =  - п
8. Шаг восьмой: 4042194914 = рояв
8. Шаг восьмой: 3959416306 = ляет
8. Шаг восьмой: 4060029165 = ся н
8. Шаг восьмой: 3760217951 = а __
9. Шаг девятый: итоговый текст --> сылки на транспортном уровне - проявляется на __
```

Итоговый текст: сылки на транспортном уровне - проявляется на __

Вывод

В ходе выполнения данной лабораторной работы я:

- ознакомился с методом Ферма для атаки на алгоритм шифрования RSA
- Реализовал данный метод на языке Python.