УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Лабораторная работа №2.2

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 повторного шифрования.

Задание

Вариант	Модуль, N	Экспонента, e	Блок зашифрованного текста, C
13	915012974539	1001953	763770087861 432343847598 764682728575 206635140312 627210520886 794063631890 309297959146 68118108284 116045398315 912085643674 257483784869 167814127445 55188158350

Ход работы

$y_1 = y,\ y_i = y_{i-1}^e (mod\ N),\ i > 1$. Проделываем данное действие до тех пор, пока $y_i \neq y$. Как только получим $y_i = y$, то это значит, что y_{i-1} наш текст. И так для каждого зашифрованного фрагмента C.

Строим последовательность: $y_1 = y,\ y_i = y_{i-1}^e (\mathrm{mod}\ N),\ i > 1$. Итак, $y_m = y^{e^m} (\mathrm{mod}\, N)$, а так как $\mathrm{НОД}(e, \varphi(N)) = 1$, то существует такое натуральное число m, что $e^m \equiv 1 (\mathrm{mod}\, \varphi(N))$. Но тогда $y^{e^m - 1} \equiv 1 (\mathrm{mod}\, N)$, отсюда следует, что $y^{e^m} \equiv y (\mathrm{mod}\, N)$, значит, y_{m-1} – решение сравнения $y = x^e (\mathrm{mod}\, N)$.

Для решения задачи была разработана программа на Python.

Листинг разработанной программы

```
import math
import random

N = 915012974539
e = 1001953
C = '''
763770087861
432343847598
764682728575
206635140312
627210520886
794063631890
309297959146
68118108284
116045398315
912085643674
257483784869
167814127445
55188158350
'''

output = ""

for enc in list(map(int, C.split())):
        y_next = pow(enc, e, N)
        enc_msg = 0

        while y_next != enc:
            enc_msg = y_next
            y_next = pow(y_next, e, N)

        msg = enc_msg.to_bytes(4, byteorder='big').decode('cp1251')
        print(f"Оригинал: {enc}, полученный текст: {enc_msg} -- {msg}")
        output += msg

print(f"Итоговый текст -->{output}")
```

Ссылка на программу: ссылка

Результат работы программы:

```
Оригинал: 763770087861, полученный текст: 4092719856 -- устр
Оригинал: 432343847598, полученный текст: 3773687277 -- анен
Оригинал: 764682728575, полученный текст: 3909034223 -- ия п
Оригинал: 206635140312, полученный текст: 4042187243 -- робл
Оригинал: 627210520886, полученный текст: 3857513262 -- емы.
Оригинал: 794063631890, полученный текст: 549777121 --  Доб
Оригинал: 309297959146, полученный текст: 3772967909 -- авле
Оригинал: 68118108284, полученный текст: 3991463200 -- ние
Оригинал: 116045398315, полученный текст: 3974687472 -- микр
Оригинал: 912085643674, полученный текст: 4008702190 -- опро
Оригинал: 257483784869, полученный текст: 4142264817 -- цесс
Оригинал: 167814127445, полученный текст: 4008763436 -- ора,
Оригинал: 55188158350, полученный текст: 550422483 --  ОЗУ
Итоговый текст -->устранения проблемы. Добавление микропроцессора, ОЗУ
```

Итоговый текст: устранения проблемы. Добавление микропроцессора, ОЗУ

Вывод

В ходе выполнения данной лабораторной работы я:

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