ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет программной инженерии и компьютерной техники

Дисциплина «Информационная безопасность»

Лабораторная работа №2.1

«Атака на алгоритм шифрования RSA посредством метода Ферма»

Вариант: 4

Учебно-методическое пособие: Криптографические системы с секретным и открытым ключом: учебное пособие. / А.А. Ожиганов; УНИВЕРСИТЕТ ИТМО. — Санкт-Петербург, 2015

Автор: Калинин Даниил Дмитриевич

Группа: P34141

Преподаватель: Маркина Татьяна Анатольевна

г. Санкт-Петербург

2024

Содержание

Цель работы

Изучить атаку на алгоритм шифрования RSA посредством метода Ферма.

Порядок выполнения работы

- Ознакомьтесь с теорией, изложенной в [3]. («Взлом алгоритма RSA при неудачном выборе параметров криптосистемы»);
- Получите вариант задания у преподавателя;
- 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

Вариант	**Модуль, *N***	**Экспонента, *e***	**Блок зашифрованного текста, *C***
4	89318473363897	2227661	3403106899606 26746900101177 67769260919924 77873792354218 15782947730235 15100267747684 28877721728826 62898555111378 4989704651236 55293402838380 4108112294245 8492269964172

Выполнение работы

Код

```
import math

def fermats_method(N, e):
    """Вычисляет закрытый ключ и параметры шифрования методом Ферма"""

    print("-- Метод Ферма --")
    n = math.trunc(math.sqrt(N)) + 1
    print(f'n = [sqrt(N)] + 1')

    i = 1
    while True:
        t = n + i
        w = pow(t, 2) - N

        print(f't_{i} = n + i = {n} + {i} = {t}')
        print(f'w_{i} = t_{i}^2 - N = {pow(t, 2)} - {N} = {w}')

        if (math.sqrt(w) % 1 != 0):
            # w - не квадрат целого числа
            print(f'w_{i} - не квадрат целого числа')
            i += 1
        else:
            # w - квардрат целого числа
            print(f'w_{i} - квадрат целого числа')
            break

    p = t + int(math.sqrt(w))
    q = t - int(math.sqrt(w))
    euler_function = (p - 1) * (q - 1)
    d = pow(e, -1, euler_function)

    print()
    print(f'p = t + sqrt(w) = {t} + {int(math.sqrt(w))} = {p}')
    print(f'q = t - sqrt(w) = {t} - {int(math.sqrt(w))} = {q}')
    print(f'euler_function = (p - 1)(q - 1) = {euler_function}')
    print(f'd = e^(-1) mod euler_function = {d}')

    return p, q, euler_function, d


def decode_part(N, d, part):
    """Декодирует часть сообщения в текст"""

    int_decoded_part = pow(part, d, N)
    return int_decoded_part.to_bytes(4, byteorder='big').decode('cp1251')


def decode(N, d, C):
    """Декодирует полученное сообщение в текст"""

    print("-- Дешифрование сообщения --")

    # Раздаляем закодированное сообщение на части и подготавливаем их
    raw_parts = C.split("\n")
```

```
    parts = []
    for i in range(len(raw_parts)):
        if raw_parts[i].strip() != "":
            parts.append(int(raw_parts[i].strip()))

    # Декодируем каждую часть
    original_message = ""
    for part in parts:
        decoded_part = decode_part(N, d, part)
        original_message += decoded_part
        print(f'Декодирована часть {part} -----> {decoded_part}')

    return original_message


if __name__ == '__main__':
    # Описание варианта
    N = 89318473363897
    e = 2227661
    C = """
        3403106899606
        26746900101177
        67769260919924
        77873792354218
        15782947730235
        15100267747684
        28877721728826
        62898555111378
        4989704651236
        55293402838380
        4108112294245
        8492269964172
        """

    print("-- Исходные данные --")
    print(f'N = {N}')
    print(f'e = {e}')
    print(f'C = \"{C}\"')
    print("\n")

    # Вычисляем закрытый ключ и требуемые параметры
    p, q, euler_function, d = fermats_method(N, e)

    print("\n")

    # Декодируем сообщение
    original_message = decode(N, d, C)
    print(f'\nОригинальное сообщение - \"{original_message}\"')
```

Результаты работы программы

```
C:\Python310\python.exe "D:\Учеба\4 курс\7 семестр\(ИБ) Информационная безопасность\information-security-labs\lab_2_1\lab_2.1.py"
-- Исходные данные --
N = 89318473363897
e = 2227661
C = "
        3403106899606
        26746900101177
        67769260919924
        77873792354218
        15782947730235
        15100267747684
        28877721728826
        62898555111378
        4989704651236
        55293402838380
        4108112294245
        8492269964172
        "


-- Метод Ферма --
n = [sqrt(N)] + 1
t_1 = n + i = 9450846 + 1 = 9450847
w_1 = t_1^2 - N = 89318509017409 - 89318473363897 = 35653512
w_1 - не квадрат целого числа
t_2 = n + i = 9450846 + 2 = 9450848
w_2 = t_2^2 - N = 89318527919104 - 89318473363897 = 54555207
w_2 - не квадрат целого числа
t_3 = n + i = 9450846 + 3 = 9450849
w_3 = t_3^2 - N = 89318546820801 - 89318473363897 = 73456904
w_3 - не квадрат целого числа
t_4 = n + i = 9450846 + 4 = 9450850
w_4 = t_4^2 - N = 89318565722500 - 89318473363897 = 92358603
w_4 - не квадрат целого числа
t_5 = n + i = 9450846 + 5 = 9450851
w_5 = t_5^2 - N = 89318584624201 - 89318473363897 = 111260304
w_5 - квадрат целого числа

p = t + sqrt(w) = 9450851 + 10548 = 9461399
q = t - sqrt(w) = 9450851 - 10548 = 9440303
euler_function = (p - 1)(q - 1) = 89318454462196
d = e^(-1) mod euler_function(N) = 15910526683025


-- Дешифрование сообщения --
Декодирована часть 3403106899606 -----> одно
Декодирована часть 26746900101177 -----> марш
Декодирована часть 67769260919924 -----> рутн
Декодирована часть 77873792354218 -----> ый (
Декодирована часть 15782947730235 -----> sing
Декодирована часть 15100267747684 -----> le r
Декодирована часть 28877721728826 -----> oute
Декодирована часть 62898555111378 -----> ) и
Декодирована часть 4989704651236 -----> всем
Декодирована часть 55293402838380 -----> аршр
Декодирована часть 4108112294245 -----> утны
Декодирована часть 8492269964172 -----> й (a

Оригинальное сообщение - "одномаршрутный (single route) и всемаршрутный (a"

Process finished with exit code 0
```

Вывод

В ходе лабораторной работы была совершена атака на алгоритм шифрования RSA посредством метода Ферма, в следствии чего было декодировано исходное сообщение. В процессе выполнения был изучен алгоритм шифрования RSA, а также вариант атаки на данный алгоритм шифрования с использованием метода Ферма.