Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа 3

«Атака на алгоритм шифрования RSA посредством метода Ферма»

Вариант № 4

Группа: P34102

Выполнил: Лапин А.А.

Проверил:
Рыбаков С.Д.

Санкт-Петербург
2024г.

Оглавление

Введение

Цель работы: изучить атаку на алгоритм шифрования RSA посредством метода Ферма.

Текст задания

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

Ход работы

Теория:

Метод факторизации Ферма

Если $N > 0$ и N нечетное, то существует взаимно однозначное соответствие между разложением на множители $n = (x - y) \cdot (x + y)$ и представлением в виде разности квадратов $n = x^2 - y^2$ с $x > y > 0$.

$$\left.\begin{array}{l} p = x - y \\ q = x + y \end{array}\right| \Rightarrow \begin{array}{l} x = p + q = 2x \\ y = p - q = -2y \end{array} \left| \Rightarrow \begin{array}{l} x = \dfrac{p+q}{2} \\ y = \dfrac{q-p}{2} \end{array} \right| \Rightarrow N = \left(\frac{p+q}{2}\right)^2 - \left(\frac{q-p}{2}\right)^2$$

Если p и q близки друг к другу, то $\left(\frac{q-p}{2}\right)^2 \to 0$, и $N \approx \left(\frac{p+q}{2}\right)^2$.
Пусть $t = \frac{p+q}{2}$, а $s = \frac{q-p}{2}$, тогда $N = t^2 - s^2$.
Тогда $t^2 - N = s^2$.
$t \approx \sqrt{N}$.
Найдем t методом перебора, начиная с $\lceil \sqrt{N} \rceil$.
В результате вычисления $t^2 - N$ мы должны получить квадрат некоторого целого числа s.
$p = t + s$ и $q = t - s$.
$\phi(N) = (p - 1) \cdot (q - 1)$.
$d = e^{-1} \mod \phi(N)$.

Вычисление руками:

$t = \lceil \sqrt{N} \rceil = 9450846$
$t^2 - N = 9450846^2 - 89318473363897 = 16751819$
$s = \sqrt{16751819} = 4092.89$

$t = 9450846 + 1 = 9450847$
$t^2 - N = 9450847^2 - 89318473363897 = 35653512$
$s = \sqrt{35653512} = 5971.05$

$t = 9450847 + 1 = 9450848$
$t^2 - N = 9450848^2 - 89318473363897 = 54555207$
$s = \sqrt{54555207} = 7386.14$

$t = 9450848 + 1 = 9450849$
$t^2 - N = 9450849^2 - 89318473363897 = 73456904$
$s = \sqrt{73456904} = 8570.7$

$t = 9450849 + 1 = 9450850$

$t^2 - N = 9450850^2 - 89318473363897 = 92358603$
$s = \sqrt{92358603} = 9610.3$

$t = 9450850 + 1 = 9450851$
$t^2 - N = 9450851^2 - 89318473363897 = 111260304$
$s = \sqrt{111260304} = 10548.0$

$p = t + s = 9450851 + 10548 = 9461399$
$q = t - s = 9450851 - 10548 = 9440303$
$\phi(N) = (p-1) \cdot (q-1) = 9461398 \cdot 9440302 = 89318454462196$
$d = e^{-1} \mod \phi(N) = 2227661^{-1} \mod 89318454462196 = 15910526683025$

Решение на Python:

Listing 1: main.py

```python
import math
from omegaconf import DictConfig
import hydra

# RSA cryptanalysis using Fermat factorization

def fermat_factor(N):
    """
    Perform Fermat's factorization on N.
    Returns a tuple of factors (p, q).
    """
    t = math.isqrt(N)
    if t * t < N:
        t += 1
    s2 = t * t - N
    while not is_perfect_square(s2):
        t += 1
        s2 = t * t - N
    s = math.isqrt(s2)
    p = t + s
    q = t - s
    return p, q

def is_perfect_square(n):
    """
    Check if n is a perfect square.
    """
    return math.isqrt(n) ** 2 == n


def decrypt_block(c, d, N):
    """
    Decrypt a single ciphertext block.
    """
    return pow(c, d, N)

def int_to_bytes(m):
    """
```

```
    Convert an integer to bytes.
    """
    hex_str = hex(m)[2:]
    if len(hex_str) % 2:
        hex_str = '0' + hex_str
    return bytes.fromhex(hex_str)


@hydra.main(version_base=None, config_path=".", config_name="config")
def main(cfg: DictConfig):
    N = cfg.N
    e = cfg.e
    ciphertexts = cfg.c

    print(f"N = {N}")
    print(f"e = {e}")
    print(f"Ciphertexts = {ciphertexts}")

    # Factor N to find p and q
    print("Factoring N using Fermat's method...")
    p, q = fermat_factor(N)
    print(f"Factors found: p = {p}, q = {q}")

    # Compute phi(N)
    phi = (p - 1) * (q - 1)
    print(f"phi(N) = {phi}")

    # Compute the private exponent d
    d = pow(e, -1, phi)
    print(f"Private exponent d = {d}")

    # Decrypt each ciphertext block
    print("Decrypting ciphertext blocks...")
    decrypted_bytes = b''.join([int_to_bytes(decrypt_block(c, d, N)) for c in
        ciphertexts])

    try:
        plaintext = decrypted_bytes.decode('cp1251')
        print(f"Plaintext: {plaintext}")
    except UnicodeDecodeError:
        print("Decrypted bytes could not be decoded to UTF-8. Raw bytes:")
        print(decrypted_bytes)

if __name__ == "__main__":
    main()
```

Listing 2: config.yaml

```
# RSA Configuration Parameters

# The modulus N, which is the product of two primes p and q.
N: 89318473363897

# The public exponent e.
e: 2227661

# The list of ciphertext blocks to be decrypted.
c:
```

```
  - 3403106899606
  - 26746900101177
  - 67769260919924
  - 77873792354218
  - 15782947730235
  - 15100267747684
  - 28877721728826
  - 62898555111378
  - 4989704651236
  - 55293402838380
  - 4108112294245
  - 8492269964172
```

Результаты работы программы

Listing 3: Вывод в консоль

```
> python main.py
N = 89318473363897
e = 2227661
Ciphertexts = [3403106899606, 26746900101177, 67769260919924, 77873792354218,
    15782947730235, 15100267747684, 28877721728826, 62898555111378, 4989704651236,
    55293402838380, 4108112294245, 8492269964172]
Factoring N using Fermat's method...
t = 9450851, s = 10548
Factors found: p = 9461399, q = 9440303
phi(N) = 89318454462196
Private exponent d = 15910526683025
Decrypting ciphertext blocks...
Plaintext: одномаршрутный (single route) и всемаршрутный (a
```

Заключение

В ходе выполнения лабораторной работы была реализована атака на алгоритм шифрования RSA посредством метода Ферма.