УНИВЕРСИТЕТ ИТМО

ФГАОУ ВО «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

Лабораторная работа №3

Вариант 7

Лабушев Тимофей

Группа P3402

Санкт-Петербург

2021

Лабораторная работа №3. Атака на алгоритм шифрования RSA посредством метода Ферма

Цель работы

Изучить атаку на алгоритм шифрования RSA посредством метода факторизации Ферма.

Задание

Вариант 7:

- Модуль n: 84032429242009
- Экспонента e: 2581907
- Блок зашифрованного текста C: 54879925681459 72167008182929 17828219756166 17814399744948 37136636080011 77223434260215 4272415279426 73759271926435 74021335775875 16903113250201 77520052156956 41247980943013

Алгоритм шифрования RSA

1. Выбираются два больших простых числа p и q, вычисляется $n = pq$.
2. Вычисляется $\phi(n) = (p-1)(q-1)$.
3. Находится любое число e, которое является взаимно простым $\phi(n)$: $\{e \in \mathbb{Z} \mid \gcd(e, \phi(n)) = 1, 1 < e < \phi(n)\}$.
4. Вычисляется такое число d, что de сравнимо с единицей по модулю $\phi(n)$: $\{d \in \mathbb{Z} \mid de \equiv 1 \pmod{\phi(n)}\}$.

Пара (n, e) является публичным ключом, который используется для шифрования. Сообщение разделяется на блоки $t, t < n$, для каждого из которых вычисляется $c = t^e \mod n$.

Пара (n, d) является секретным ключом, который используется для дешифрования: $t = c^d \mod n$.

Атака на алгоритм RSA

Известны (n, e) — публичный ключ, c — зашифрованные данные. Необходимо найти d путем факторизации n, т.е. нахождения p и q. При правильном выборе чисел данная задача является вычислительно сложной, однако если p и q близки друг к другу, то они могут быть найдены за небольшое число шагов методом факторизации Ферма.

Атака состоит в решении уравнения $t^2 - w^2 = n$, т.е. поиске такого t, что $t^2 - n$ является квадратом целого числа. Поиск начинается с $t = \lceil \sqrt{n} \rceil$ — наименьшего t, при котором $t^2 - n \geqslant 0$. На каждой итерации к t прибавляется 1 и вычисляется $w^2 = t^2 - n$. Если полученное значение является квадратом целого w, то $p = t + w$ и $q = t - w$. Знание p и q позволяет вычислить d, следуя шагам 2-4 алгоритма шифрования.

```
Given public key (n, e): (84032429242009, 2581907)
Breaking private key (n, d):

Iter 1: t = 9166921, w^2 = t^2 - n = 11378232, w = 3373.1635003361457
Iter 2: t = 9166922, w^2 = t^2 - n = 29712075, w = 5450.878369584117
Iter 3: t = 9166923, w^2 = t^2 - n = 48045920, w = 6931.516428603484
Iter 4: t = 9166924, w^2 = t^2 - n = 66379767, w = 8147.377921761086
Iter 5: t = 9166925, w^2 = t^2 - n = 84713616, w = 9204.0

p = t + w = 9176129
q = t - w = 9157721
phi(n) = (p - 1)(q - 1) = 84032410908160
d = e^-1 mod phi(n) = 2475823295643
Computed private key (n, d): (84032429242009, 2475823295643)

Decoding C:
c = 54879925681459, c' = c^d (mod n) = 4024496352 = ef e0 f0 e0 = |пара|
c = 72167008182929, c' = c^d (mod n) = 3958105579 = eb eb e5 eb = |ллел|
c = 17828219756166, c' = c^d (mod n) = 4243454956 = fc ed fb ec = |ьным|
c = 17814399744948, c' = c^d (mod n) = 3894471918 = e8 20 ec ee = |и мо|
c = 37136636080011, c' = c^d (mod n) = 4059226348 = f1 f2 e0 ec = |стам|
c = 77223434260215, c' = c^d (mod n) = 3895206112 = e8 2c 20 e0 = |и, а|
c = 04272415279426, c' = c^d (mod n) = 0551743973 = 20 e2 f1 e5 = | все|
c = 73759271926435, c' = c^d (mod n) = 3974164728 = ec e0 f0 f8 = |марш|
c = 74021335775875, c' = c^d (mod n) = 4042519277 = f0 f3 f2 ed = |рутн|
c = 16903113250201, c' = c^d (mod n) = 4226097391 = fb e5 20 ef = |ые п|
c = 77520052156956, c' = c^d (mod n) = 3773490674 = e0 ea e5 f2 = |акет|
c = 41247980943013, c' = c^d (mod n) = 4213189983 = fb 20 2d 5f = |ы -_|

Decoded message:
параллельными мостами, а всемаршрутные пакеты -_
```

```julia
with_terminal() do
    print("Given public key (n, e): ($(task_n), $(task_e))\n")
    print("Breaking private key (n, d):\n\n")

    t::UInt64 = round(sqrt(task_n))
    w::Float64 = 0.0

    # Find such t that w = sqrt(t^2 - n) is an integer
    iter = 1
    while true
        t += 1
        w2 = t^2 - task_n
        w = sqrt(w2)
        print("Iter $(iter): t = $(t), w^2 = t^2 - n = $(w2), w = $(w)\n")

        if isinteger(w)
            break
        end

        iter += 1
    end

    p::UInt64 = t + w
    print("\np = t + w = $(p)\n")
    q::UInt64 = t - w
    print("q = t - w = $(q)\n")
    phi_n::UInt64 = (p - 1)*(q - 1)
    print("phi(n) = (p - 1)(q - 1) = $(phi_n)\n")
    d = invmod(task_e, phi_n)
    print("d = e^-1 mod phi(n) = $(d)\n")

    print("Computed private key (n, d): ($(task_n), $(d))\n\n")
    print("Decoding C:\n")

    msg = []

    for c in task_c
        # Decode the block by computing c^d (mod n)
        c_::UInt64 = powermod(c, d, task_n)
        # Each block contains four 8-bit characters encoded as windows-1251
        chs = [(c_ >> 24) % UInt8, (c_ >> 16) % UInt8, (c_ >> 8) % UInt8, c_ % UInt8]
        textchs = decode(chs, "WINDOWS-1251")

        @printf("c = %014d, c' = c^d (mod n) = %010d = %s = |%s|\n",
            c, c_, join(map(n -> string(n, base=16), chs), " "), textchs)
        append!(msg, textchs)
    end

    print("\nDecoded message:\n$(join(msg))")
end
```

Вывод

В ходе выполнения лабораторной работы был изучен принцип работы алгоритма шифрования RSA, а также рассмотрены на практике последствия неудачного выбора параметров криптосистемы: зашифрованное сообщение может быть восстановлено без знания секретного ключа за небольшое число шагов.