RSA 從入門到放棄

OAlienO

BambooFox

2018/10/31

Table of Contents

1 RSA
- Introduction
- Implementation
- Factor Attack
- Wiener's Attack

RSA 簡介

- 1997 年由 Ron Rivest, Adi Shamir, Leonard Adleman 提出的非對稱式加密演算法
- 廣泛應用於
  - https 加密連線
  - ssh 公鑰認證
  - WannaCry

RSA 產生密鑰

```
def genkey():
    # choose p, q, e
    p, q, e = getPrime(1024), getPrime(1024), 65537
    # calculate d
    n, phi = p * q, (p - 1) * (q - 1)
    d = inverse(e, phi)
    # return publicKey, privateKey
    return (n, e), (n, d)
```

RSA 加解密

```python
def enc(m, public):
    n, e = public
    return pow(m, e, n)

def dec(c, private):
    n, d = private
    return pow(c, d, n)
```

費馬小定理 (Fermat's little theorem)

條件

a 是正整數, p 是質數, gcd(a, p) = 1

費馬小定理

$a^{p-1} \equiv 1 \pmod{p}$

歐拉函數 (Euler's totient function)

- 對正整數 n
- $\varphi(n)$ 是小於等於 n 的正整數中與 n 互質的數的數目
- $\varphi(n) = n \prod_{p|n} \left(1 - \frac{1}{p}\right)$

歐拉定理 (Euler's theorem)

條件

a, n 是正整數, gcd(a, n) = 1

條件

$a^{\varphi(n)} \equiv 1 \pmod{n}$

註解

n 是質數, $\varphi(n) = n - 1$
費馬小定理實際上是歐拉定理的一個特例

RSA 正確性

假設

gcd(m, n) = 1

驗證 $m^{ed} \equiv m \pmod{n}$

$$
\begin{aligned}
ed &\equiv 1 \pmod{\varphi(n)} \\
ed &= 1 + k\varphi(n) \text{ for some } k \\
m^{ed} &= m^{1+k\varphi(n)} = m(m^{\varphi(n)})^k \equiv m(1)^k = m \pmod{n}
\end{aligned}
$$

RSA 正確性

假設

gcd(m, n) $\neq$ 1

驗證 $m^{ed} \equiv m \pmod{n}$

$$m^{ed} = m^{1+k\varphi(n)} = m \pmod{p} \text{ holds for all } m$$

For $m \equiv 0 \pmod{p}$, it is trivial

For $m \not\equiv 0 \pmod{p}$, we have shown in last slide

Similar statement can be made for q

Implementation

crypto 會用到的 python packages

- pycrypto
- gmpy2
- sage
- primefac
- sympy

Implementation

```python
from Crypto.PublicKey import RSA
public  = RSA.importKey(open('public.pem').read())
private = RSA.importKey(open('private.pem').read())
```

Integer Factorization

- 只要能分解 $n = pq$
- 可以照著原本產生公私鑰的步驟產生私鑰，進而解密密文
- 目前最好的演算法是 General Number Field Sieve(GNFS)
- 一些特殊情況下的演算法
  - Pollard's $p - 1$ Algorithm
  - Williams's $p + 1$ Algorithm
  - Fermat's Factorization Method

Common Factor Attack

- 當兩個 n 有共同質因數
- $gcd(n_1, n_2)$ 能有效率的分解 n_1, n_2

Common Factor Attack

CTF Challenges

SECCON 2017 Online CTF - Ps and Qs

Pollard's p - 1 Algorithm

假設

正整數 a, 合數 n, 質數 p
gcd(a, p) = 1 且 p | n

Pollard's p - 1 Algorithm

$$a^{p-1} \equiv 1 \pmod{p}$$
$$a^{k(p-1)} \equiv 1 \pmod{p}$$
$$a^{k(p-1)} - 1 \equiv 0 \pmod{p} \text{ for some } k$$
$$p \mid gcd(a^{k(p-1)} - 1, n)$$

Pollard's p - 1 Algorithm

Pollard's p - 1 Algorithm (cont.)

測試 $gcd(2^{1}-1,n), gcd(2^{1\times 2}-1,n), gcd(2^{1\times 2\times 3}-1,n), \cdots$
只要 $p-1 \mid 1\times 2\times\cdots,\ \ gcd(2^{1\times 2\times\cdots}-1,n)>1$

Pollard's p - 1 Algorithm

使用條件

$p-1$ 最大的質因數很小

Pollard's p - 1 Algorithm

```python
def pollard(n):
    a = 2
    b = 2
    while True:
        a = pow(a, b, n)
        d = gcd(a - 1, n)
        if 1 < d < n: return d
        b += 1
```

Fermat's Factorization Method

CTF Challenges

SECCON 2017 Online CTF - Very Smooth

Fermat's Factorization Method

假設

奇合數 n 是兩質數 pq 乘積

Fermat's Factorization Method

$$a = \frac{p+q}{2}, b = \frac{p-q}{2}$$

$$n = (a+b)(a-b) = a^2 - b^2$$

猜 $a = \lceil \sqrt{n} \rceil$ 並測試 $b^2 = a^2 - n$ 是不是平方數

不行就把 $a+1$ 再猜一次

Fermat's Factorization Method

使用條件

$|p-q|$ 很小

Fermat's Factorization Method

```
def fermat(n):
    a = ceil(sqrt(n))
    b2 = a * a - n
    while not gmpy2.iroot(b2, 2)[1]:
        a = a + 1
        b2 = a * a - n
    b = gmpy2.iroot(b2, 2)[0]
    return [a + b, a - b]
```

Fermat's Factorization Method

CTF Challenges

Codegate CTF 2018 Preliminary - Miro

觀察一

觀察

$$\begin{aligned}\varphi(n) &= (p-1)(q-1) && (1)\\ &= n-p-q+1 && (2)\\ &= n-p-\frac{n}{p}+1 && (3)\end{aligned}$$

$$p^2 + p(\varphi(n) - n - 1) + n = 0 \qquad (4)$$

解釋

只要我們知道 $\varphi(n)$，式子 4 就只是個一元二次方程式，解出來的兩個根 p_1, p_2 就是 p, q，我們就成功分解 n

觀察二

觀察

$$ed \equiv 1 \pmod{\varphi(n)} \tag{5}$$

$$ed = k\varphi(n) + 1 \tag{6}$$

$$\varphi(n) = \frac{ed - 1}{k} \tag{7}$$

解釋

已知 e，只要知道 k, d 就可以求出 $\varphi(n)$

Lemma 1

Lemma 1

如果 $p \approx q \approx \sqrt{n}$

$$n - \varphi(n) < 3\sqrt{n} \tag{8}$$

Proof

$$\begin{aligned} n - \varphi(n) &= n - (p-1)(q-1) && (9) \\ &= n - pq + p + q - 1 && (10) \\ &= p + q - 1 && (11) \\ &< 3\sqrt{n} && (12) \end{aligned}$$

Lemma 2

Lemma 2

如果 $d < \frac{1}{3}n^{\frac{1}{4}}$

$$k < \frac{1}{3}n^{\frac{1}{4}} \tag{13}$$

Proof

$$k\varphi(n) = ed - 1 < ed < \varphi(n)d \tag{14}$$

$$k < d < \frac{1}{3}n^{\frac{1}{4}} \tag{15}$$

Lemma 3

Lemma 3

如果 $d < \frac{1}{3}n^{\frac{1}{4}}$

$$\frac{1}{2d} > \frac{1}{n^{\frac{1}{4}}} \tag{16}$$

Proof

$$d < \frac{1}{3}n^{\frac{1}{4}} \tag{17}$$

$$2d < 3d < n^{\frac{1}{4}} \tag{18}$$

$$\frac{1}{2d} > \frac{1}{n^{\frac{1}{4}}} \tag{19}$$

Legendre's theorem in Diophantine approximations

Legendre's theorem in Diophantine approximations

給定 $\alpha \in \mathbb{R}, \frac{a}{b} \in \mathbb{Q}$，並且滿足 $\left|\alpha - \frac{a}{b}\right| < \frac{1}{2b^2}$
那麼 $\frac{a}{b}$ 會是 α 的 convergent of the continued fraction expansion

Proof

Too Hard...

Wiener's Attack

如果 $d < \frac{1}{3}n^{\frac{1}{4}}$

$$\left|\frac{e}{n} - \frac{k}{d}\right| = \left|\frac{ed - nk}{nd}\right| \tag{20}$$

$$= \left|\frac{1 + k\varphi(n) - nk}{nd}\right| \tag{21}$$

$$= \frac{k(n - \varphi(n)) - 1}{nd} < \frac{3k\sqrt{n} - 1}{nd} < \frac{3k\sqrt{n}}{nd} \tag{22}$$

$$< \frac{1}{n^{\frac{1}{4}}d} < \frac{1}{2d^2} \tag{23}$$

Wiener's Attack

根據 Legendre's theorem in Diophantine approximations，$\frac{k}{d}$ 會是 $\frac{e}{n}$ 的 convergents of the continued fraction expansion，我們只要遍歷 $\frac{e}{n}$ 的 convergents of the continued fraction expansion 並檢查從觀察一和觀察二推回去的 $p_1 p_2 = n$，該 p_1, p_2 就是 p, q

Wiener's Attack

結論

條件: $d < \frac{1}{3}n^{\frac{1}{4}}$
結果: 分解 n