An toàn và An ninh thông tin

Nguyễn Linh Giang.
Bộ môn Truyền thông và Mạng máy tính.

Phương pháp RSA

Thuật toán mã hoá công khai RSA

- Cơ sở lý thuyết
- Sơ đồ mã hóa và giải mã
- Tạo khóa
- Vấn đề tính toán trong RSA
- Thám mã RSA

Lý thuyết số

- Số học modun
- Định lý Euler và định lý Fermat
- Kiểm tra số nguyên tố
- Thuật toán Euclid
- Định lý số dư Trung Hoa
- Sinh giả ngẫu nhiên các số nguyên lớn

Số học modun

- ***Định lý về số dư.*** Cho một số nguyên dương n và một số nguyên a. Khi đó tồn tại duy nhất các số q và r với , sao cho a = qn + r.

  r gọi là số dư của phép chia a cho n.
- ***Định nghĩa số dư.*** Cho một số nguyên dương n và số nguyên a. Ký hiệu a mod n là số dư khi chia a cho n.

  a = x n + (a mod n)
- ***Định nghĩa 2.*** Hai số a và b được gọi là đồng dư theo mođun n nếu a mod n = b mod n. Và viết là a $\equiv$ b (mod n)
- Ví dụ: 11 = 1 x 7 + 4 => 11 mod 7 = 4

  -11 = (-2) x 7 + 3 => -11 mod 7 = 3

  73 $\equiv$ 4 (mod 23)

Số học modun (tiếp)

- Phép toán số học mođun.
  1. [(a mod n) + (b mod n)] mod n = (a + b) mod n
  2. [(a mod n) – (b mod n)] mod n = (a – b) mod n
  3. [(a mod n) x (b mod n)] mod n = (a x b) mod n
- Chứng minh
- Ví dụ

  11 mod 8 = 2; 15 mod 8 = 2

  [(11 mod 8) + (15 mod 8)] mod 8 = 10 mod 8 = 2

Số học modun (tiếp)

- ***Tính chất của số học mođun***

  Tính giao hoán:

  (w + y) mod n = (y + w) mod n

  (w x x) mod n = (x x w) mod n

  Tính kết hợp:

  [(w + x) + y] mod n = [w + (x + y)] mod n

  [(w x x) x y] mod n = [w x (x x y)] mod n

  Tính phân phối:

  [w x (x + y)] mod n = [(w x x) + (w x y)] mod n

  Phần tử trung hoà

  (0 + w) mod n = w mod n

  (1 x w) mod n = w mod n

  Phần tử đối xứng của phép cộng:

  Với mỗi w $\in$ Z_n

  tồn tại z sao cho w + z = 0 mod n

Định lý Euler và định lý Fermat

- Định lý Fermat
- Hàm Euler
- Định lý Euler

Định lý Fermat

- ***Phát biểu***
  Nếu p là số nguyên tố và a là số nguyên dương không chia hết cho p thì

$$a^{p-1} \equiv 1 (\mathrm{mod}\ p)$$

- ***Chứng minh***
- ***Ví dụ***
  a = 7, p = 19
- ***Định lý trên có thể phát biểu dưới dạng tương đương như sau:***
  Nếu p là số nguyên tố và a là một số nguyên dương bất kỳ, thì

$$a^{p} \equiv a \ (\mathrm{mod}\ p)$$

Hàm Euler

- Hàm Euler được ký hiệu là $\phi(n)$ là số các số nguyên dương nhỏ hơn n và nguyên tố cùng nhau với n.
- ***Ví dụ***

$\phi(21)$ = 12 (12 số nguyên đó là [1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20])

Định lý Euler

- Phát biểu
- Ví dụ

  a = 3; n = 10;
- Chứng minh
  - Trường hợp n là số nguyên tố:
  - Trường hợp n là số nguyên bất kỳ:
- Phát biểu dạng khác

Kiểm tra số nguyên tố

- Định lý

Nếu p là số lẻ thì $x^2 \equiv 1 \pmod{p} \Rightarrow x \equiv 1$ hoặc $x \equiv -1$

- Chứng minh
  - Xét trường hợp (x + 1), (x – 1) đồng thời chia hết cho p.
  - Xét trường hợp (x - 1) chia hết cho p.
  - Tương tự xét trường hợp (x + 1) chia hết cho p ta suy ra x -1 (mod p)
- Kết quả suy ra

Nếu tồn tại $x \mid x^2 \equiv 1 \pmod{n}, x \neq \pm 1$ thì n không phải là số nguyên tố.

1.3 Kiểm tra số nguyên tố (tiếp)

- Thuật toán Miller, Rabin: kiểm tra một số có phải là một số nguyên tố không dựa vào kết quả của định lý trên.

  Input của thuật toán là số nguyên n và một số nguyên a nào đó nhỏ hơn n. Nếu WITNESS có giá trị trả về là TRUE thì n không phải là số nguyên tố, nếu WITNESS có giá trị trả về là FALSE thì n có thể là số nguyên tố
- Ví dụ
- Đánh giá độ phức tạp

```
WITNESS(a, n)
1.    b_k b_{k-1} ... b_0 là biểu diễn nhị phân của (n – 1)
2.    d ← 1
3.    for i ← k downto 0 do {
4.          x ← d
5.          d ← (d x d) mod n
6.          if d = 1 and x ≠ 1 and x ≠ n – 1 then
7.                return TRUE
8.          if b_i = 1 then
9.                d ← (d x a) mod n
      }
10.   if d ≠ 1 then
11.         return TRUE
12.   return FALSE
```

Thuật toán Euclid

- Tìm ước số chung lớn nhất

- Định lý Với 2 số nguyên dương a và b bất kỳ chúng ta có

$$\gcd(a, b) = \gcd(b, a \bmod b)$$

- Chứng minh
- Ví dụ
- Đánh giá độ phức tạp

Thuật toán Euclid (tiếp)

- Tìm phần tử đối xứng

  Thuật toán Euclid mở rộng sẽ trả về phần tử đối xứng của d nếu gcd(d, f) = 1.

EXTENDED EUCLID(d, f)

1. $(X1, X2, X3) \leftarrow (1, 0, f); (Y1, Y2, Y3) \leftarrow (0, 1, d)$
2. if $Y3 = 0$ return $X3 = \gcd(d, f)$;
3. if $Y3 = 1$ return $Y3 = \gcd(d, f)$; $Y2 = d^{-1} \bmod f$
4. $Q = \left\lfloor \frac{X3}{Y3} \right\rfloor$
6. $(T1, T2, T3) \leftarrow (X1 - QY1, X2 - QY2, X3 - QY3)$
7. $(Y1, Y2, Y3) \leftarrow (T1, T2, T3)$
8. goto 2

Định lý số dư Trung Hoa

- Định lý

$$M = \prod_{i=1}^{k} m_i$$

Trong đó m_i là nguyên tố cùng nhau từng đôi một, $\gcd(m_i, m_j) = 1$ với $1 \le i, j \le k$ và $i \ne j$. Chúng ta có thể biểu diễn bất kỳ số nguyên dương nào trong Z_M bởi k số trong các Z_{mi}:

$$A \leftrightarrow (a_1, a_2, \ldots, a_k)$$

$A \in Z_M$, $a_i \in Z_{mi}$ và $a_i = A \bmod m_i$ với $1 \le i \le k$.

  - Hai kết quả của định lý số dư Trung Hoa
  - Ứng dụng của định lý số dư Trung Hoa
  - Ví dụ

Sinh giả ngẫu nhiên các số nguyên lớn

- Bộ sinh số giả ngẫu nhiên

  Kỹ thuật được sử dụng rộng rãi trong việc sinh giả ngẫu nhiên là phương pháp đồng dư tuyến tính lần đầu tiên được đề xuất bởi Lehmer.
- Sinh số giả ngẫu nhiên dựa trên kỹ thuật mật mã
- Bộ sinh số giả ngẫu nhiên Blum Blum Shub

Sơ đồ mã hóa và giải mã RSA

- Xuất xứ
  - RSA do Ron Rivest, Adi Shamir và Len Adlenman phát minh năm 1977;
  - Hệ thống mã khoá công khai phổ biến và đa năng:
    - Được sử dụng trong các ứng dụng mã hóa/giải mã;
    - Chứng thực;
    - Phân phối và trao đổi khoá.

Sơ đồ mã hóa và giải mã RSA

- Thuật toán RSA:
  - Phương pháp mã hóa khối;
- Văn bản rõ và văn bản mật là các số nguyên có giá trị từ 0 đến n-1, n – số nguyên lớn;
- Mỗi khối có giá trị nhỏ hơn n. Như vậy, kích thước của khối (số bít) nhỏ hơn hoặc bằng $\log_2(n)$.
  - Thực tế, kích thước của khối là 2^k bit với $2^k < n \leq 2^{k+1}$.

Mô tả giải thuật (tiếp)

- Sinh khoá

Chọn p, q	p và q là số nguyên tố
Tính $n = p \times q$	
Tính $\phi(n) = (p-1)(q-1)$	
Chọn số nguyên e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Tính d	$d = e^{-1} \bmod \phi(n)$
Khoá công khai	$KU = [e, n]$
Khoá mật	$KR = [d, n]$

Mô tả giải thuật (tiếp)

- Mã hoá

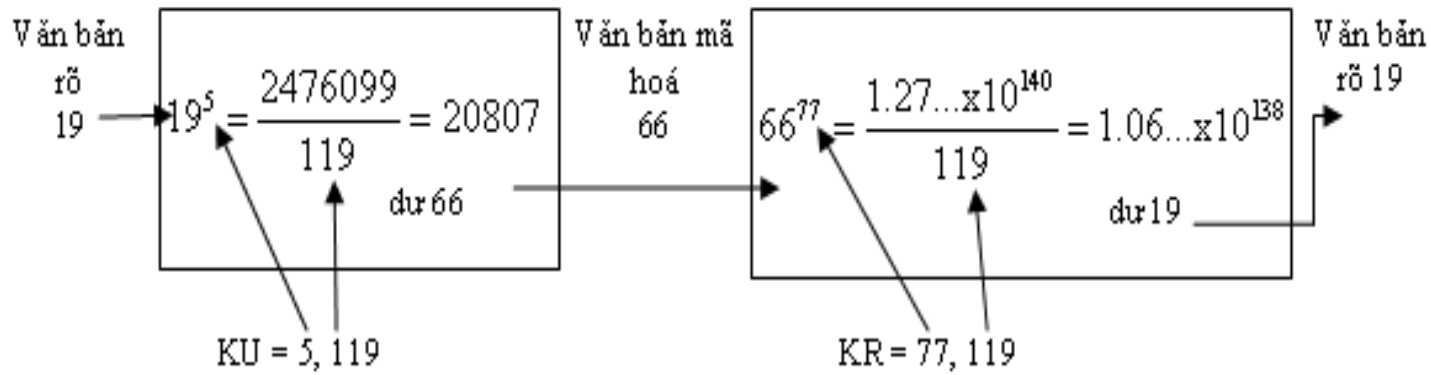
Bản rõ	M < n
Mật mã	$C = M^e \bmod n$

- Giải mã

Mật mã	C
Bản rõ	$M = C^d \bmod n$

Mô tả giải thuật (tiếp)

- Ví dụ



Thực hiện giải thuật

- Mã hoá và giải mã
  - Vấn đề trong thuật toán mã hoá và giải mã RSA là việc thực hiện phép toán luỹ thừa và phép toán đồng dư với số nguyên lớn.
  - Giải quyết dựa trên tính chất của phép toán modun:

    [(a mod n) x (b mod n)] mod n = (a x b) mod n

Thực hiện giải thuật (tiếp)

- Sinh khoá
  - Xác định số nguyên tố p, q (sử dụng thuật toán Miller – Rabin)
    1. Chọn một số nguyên lẻ n ngẫu nhiên (sử dụng bộ sinh số giả ngẫu nhiên).
    2. Chọn một số nguyên a < n ngẫu nhiên.
    3. Thực hiện thuật toán xác suất để kiểm tra số nguyên tố. Nếu n test thành công thì loại bỏ giá trị n và quay lại bước 1.
    4. Nếu n test thành công với số lượng test đủ, chấp nhận n; mặt khác, quay lại bước 2.
  - Chọn d
  - Tính e từ d và $\phi(n)$ (sử dụng thuật toán Euclid)

Tính bảo mật của giải thuật RSA

- Tấn công vét cạn
- Tấn công toán học
- Tấn công dựa vào thời gian

Tấn công vét cạn

- Nội dung của phương pháp tấn công này là đối phương thực hiện vét cạn toàn bộ không gian khoá.
- Biện pháp đối phó:

  Một cách chung để chống lại phương pháp tấn công này là sử dụng không gian có khoá kích thước lớn, tức là tăng số bít của d và e. Tuy vậy, điều này sẽ làm quá trình sinh khoá, mã hoá, giải mã thực hiện chậm đi.

Tấn công toán học

- Trường hợp đơn giản nhất là người thám mã biết được $\phi(n)$
- Phân tích n thành tích của 2 thừa số nguyên tố. Thuật toán p–1

  Có nhiều thuật toán phân tích n thành hai thừa số nguyên tố. Có 3 thuật toán hiệu quả trên các số rất lớn là thuật toán sàng bình phương (quadratic sieve), đường cong elip (elliptic curve) và number field sieve. Các thuật toán được biết đến nhiều trước đây là thuật toán p – 1 của Pollard, thuật toán p + 1 của William, thuật toán chia nhỏ liên tiếp (continued fraction algorithm) và tất nhiên là thuật toán thử chia (trial division).

Tấn công dựa vào thời gian

- Nội dung của phương pháp này dựa vào việc xem xét thời gian thực hiện thuật toán giải mã
- Biện pháp đối phó:
  - Thời gian tính mũ là hằng: Làm cho thời gian tính mũ là như nhau trước khi trả về kết quả. Biện pháp này đơn giản nhưng làm giảm hiệu năng.
  - Thực hiện trễ ngẫu nhiên: Thêm các trễ thời gian ngẫu nhiên vào thuật toán mũ hoá.
  - Blinding: Nhân văn bản mật với một số ngẫu nhiên trước khi thực hiện mũ hoá.