CHƯƠNG 3
CÁC HỆ MÃ HÓA CÔNG KHAI
(MÃ HÓA BẤT ĐỐI XỨNG)

Giới thiệu

- Nguồn gốc:
  - Hệ mật mã khóa đối xứng (cổ điển và hiện đại) không đáp ứng được 2 mục tiêu an toàn
    - Xác thực
    - Chống phủ nhận
- Quản lý khóa đối xứng là một vấn đề nan giải
  - Cần tìm một phương pháp mã hóa khác có thể giải quyết được các vấn đề của mã hóa đối xứng
  - Whitfield Diffie và Martin Hellman đã tìm ra một phương pháp mã hóa khóa công khai

Các khái niệm và sơ đồ

- Sơ đồ mã hóa bất đối xứng:
  - Có thể gọi là mã hóa khóa công khai (Public Key Cryptography – PKC) sử dụng một cặp khóa cho quá trình mã hóa và giải mã
  - Cặp khóa này phải đảm bảo tính toàn vẹn và xác thực cho chủ thể của khóa

Hệ mật mã khóa công khai

- Các giải thuật mật mã khóa công khai sử dụng một khóa để mật mã hóa và một khóa khác có liên quan để giải mật mã; có đặc điểm:
  - Không thể tính lại khóa giải mật mã nếu biết trước giải thuật mật mã hóa và khóa dùng mã hóa
  - Một trong hai khóa đều có thể dùng mã hóa và khóa còn lại dùng để giải mật mã

Các thành phần giải thuật khóa công khai

- Giải thuật khóa công khai gồm 6 thành phần:
  - **Bản rõ (Plaintext)**: thông điệp có thể đọc, đầu vào của giải thuật
  - **Giải thuật mật mã hóa**
  - **Khóa công khai và khóa bí mật**: một cặp khóa được chọn sao cho 01 khóa dùng để mật mã hóa và 01 khóa dùng để giải mật mã
  - **Bản mã (Cipher Text)**: thông điệp đầu ra ở dạng không đọc được, phụ thuộc vào bản rõ và khóa; nghĩa là với cùng một thông điệp, 2 khóa khác nhau sinh ra 2 bản mã khác nhau
  - **Giải thuật giải mã**

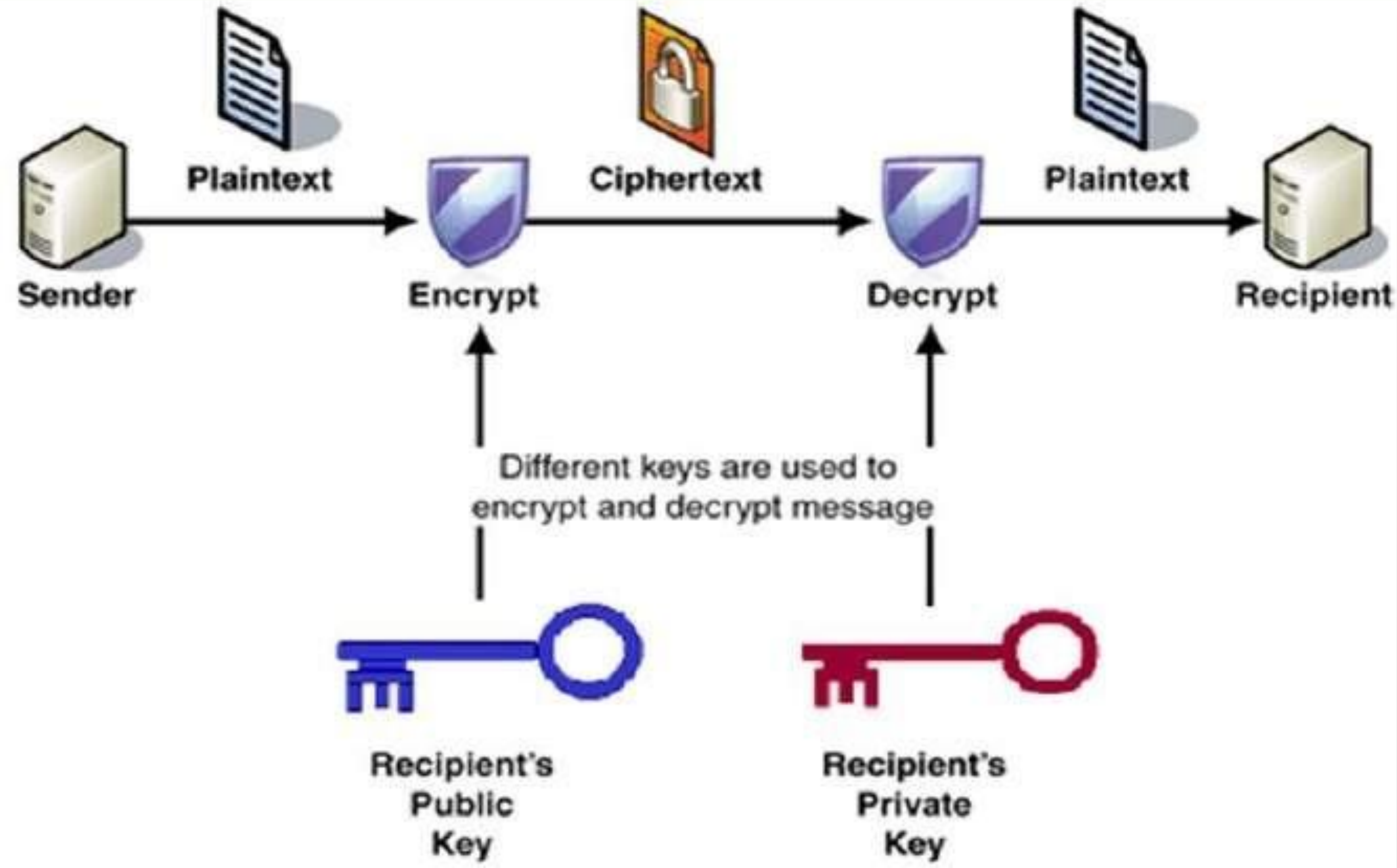
Các bước thực hiện

- Mỗi người dùng tạo một cặp khóa để mã hóa và giải mã
- Mỗi người dùng đăng ký một trong 2 khóa làm **khóa công khai** sao cho mọi người đều có thể truy cập được. Khóa còn lại được giữ **bí mật**.

Các bước thực hiện

- **Ví dụ:**
  - An muốn gửi Bình một thông điệp bí mật → An mã hóa thông điệp bằng khóa công khai của Bình
  - Khi Bình nhận được thông điệp → giải mã thông điệp bằng khóa bí mật của mình
  - Ngoài Bình, không người nào có khả năng giải mã vì chỉ có Bình có khóa để giải mã

Các bước thực hiện

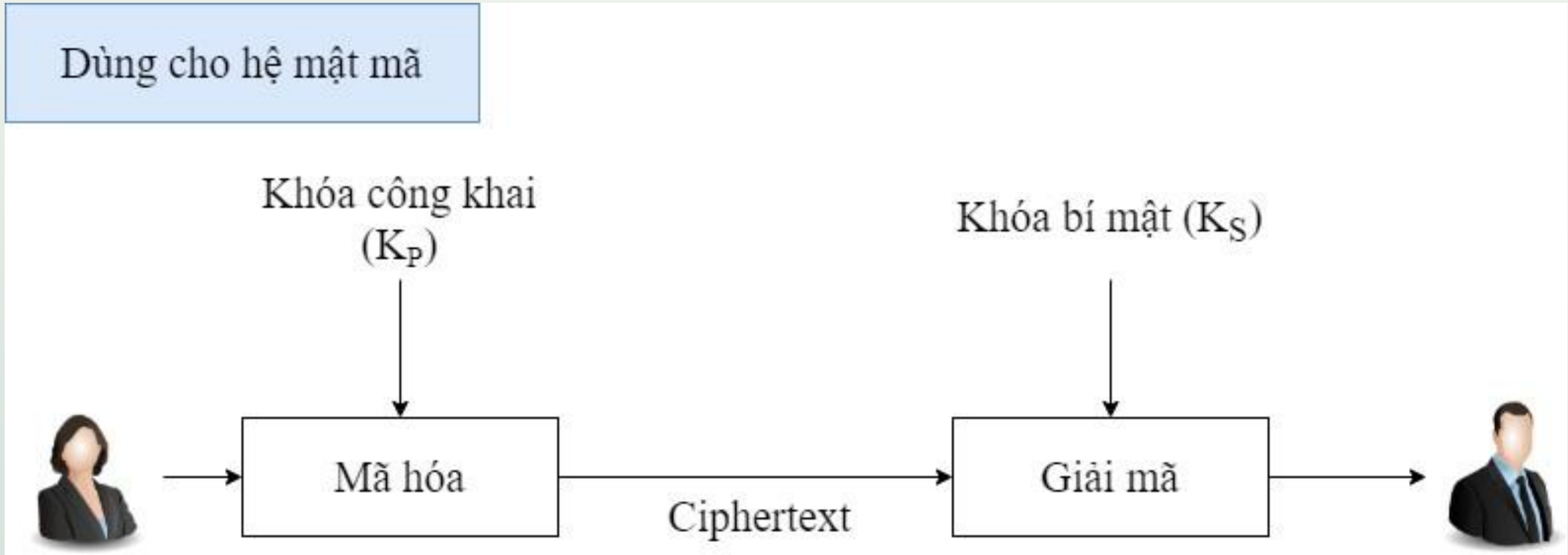


Sơ đồ mã hóa

❖ Sơ đồ mã hóa bất đối xứng (dùng cho mã hóa)

$$Ciphertext = E(K_p, Plaintext)$$

$$Plaintext = D(K_s, E(K_p, Plaintext))$$

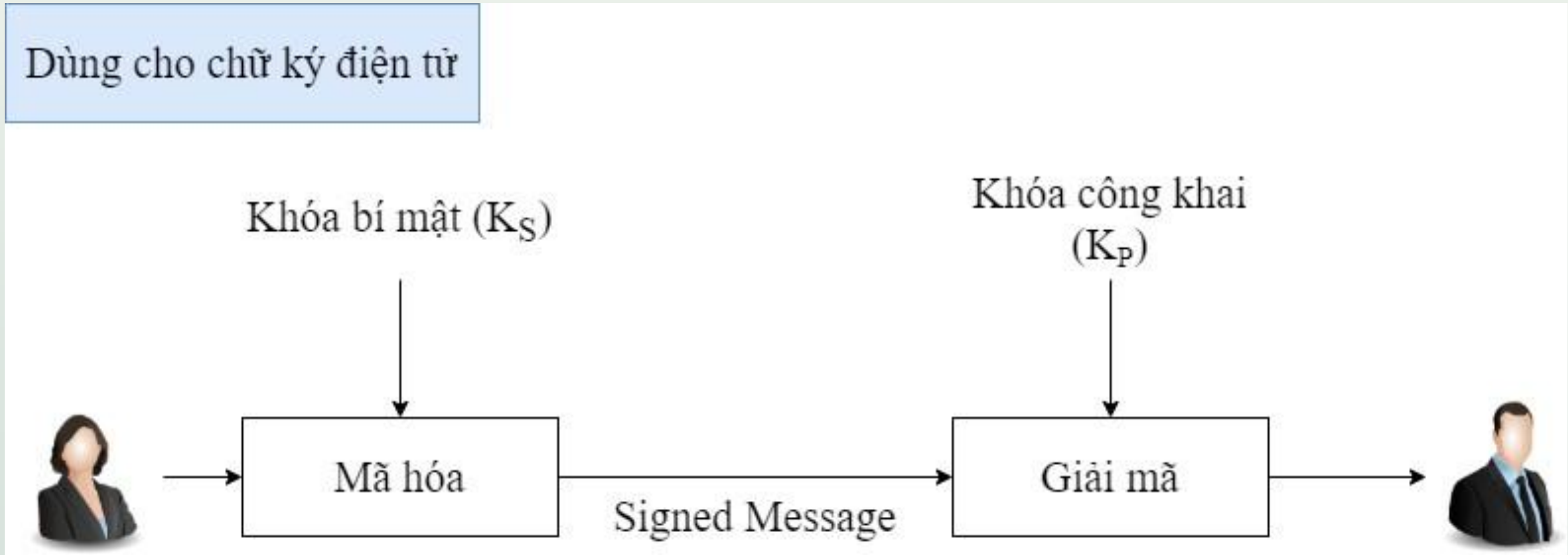


Sơ đồ mã hóa

❖ Sơ đồ mã hóa bất đối xứng (dùng chữ ký điện tử)

$$SignedM = E(K_S, M_{hash})$$

$$M_{hash} = D(K_P, E(K_S, M_{hash}))$$



Các yêu cầu

- Dễ dàng tính được cặp khóa công khai K_p và bí mật K_s
- Dễ dàng tính được bản mã với bản rõ và khóa công khai cho trước $C = E(K_p, P)$
- Dễ dàng tính được bản rõ từ bản mã và khóa bí mật cho trước $P = D(K_s, C) = D(K_s E(K_p, P))$

Các yêu cầu

- Không thể tính K_S từ K_P
- Không thể tính được bản rõ P từ khóa K_P và bản mã cho trước
- Mã hóa và giải mã được thực hiện theo một trong hai quá trình $P = D(K_S, E(K_P, P)) = D(K_P, E(K_S, P))$

Lý thuyết liên quan: số học đồng dư

- Số học đồng dư
  - a **mod** n
  - a ***op*** b **mod** n với ***op*** = +, −, $*$, /, mũ
- Ví dụ
  - $40\ mod\ 6\ \ =?$
  - $5+2\ mod\ 6\ \ =?$
  - $9-4\ mod\ 3\ \ =?$
  - $5*3\ mod\ 6\ \ =?$
  - $4/2\ mod\ 3\ \ =?$
  - $2^4\ mod\ 6\ \ =?$

Thủ tục bình phương

- Dựa vào tính chất
  - $a * b \ mod\, n = \big((a\, mod\, n) * (b\, mod\, n)\big)\, mod\, n$
- Tính a^{25}
  - $a^{25_{(10)}} = a^{11001_{(2)}}$
  - $a^{11001_{(2)}} = a^{10000_{(2)} + 1000_{(2)} + 1_{(2)}}$?
  - $a^{10000_{(2)}+1000_{(2)}+1_{(2)}} = a^{10000_{(2)}} * a^{1000_{(2)}} * a^{1_{(2)}}$?
  - $a^{10000_{(2)}} * a^{1000_{(2)}} * a^{1_{(2)}} = a^{2^4_{(10)}} * a^{2^3_{(10)}} * a^{2^0_{(10)}}$?

Thủ tục bình phương

ModExp1(a, b, s)

- Input:
  - 3 số nguyên dương a, b, s sao cho a < s
  - b[n – 1] . . . b[1]b[0] là biểu diễn nhị phân của b, n = $[\log_2 b]$
- Output: $a^b \ mod \ s$

```
p[0] = a mod s
for i = 1 to n - 1
      p[i] = p[i - 1]^2 mod s
r = 1
for i = 0 to n - 1
      if b[i] = 1 then r = r * p[i] mod s
return r
```

Thủ tục bình phương: Bài tập

- Tính $6^{73} \, mod \, 100$:
  - Với $73 = 1001001_2$ (7 bit)
  - Tính
    - $p[0] = 6 \, mod \, 100 = 6$
    - $p[1] = p[0]^2 \, mod \, 100 = 6^2 \, mod \, 100 = 36$
    - $p[2] = p[1]^2 \, mod \, 100 = 36^2 \, mod \, 100 = 96$
    - $p[3] = p[2]^2 \, mod \, 100 = 96^2 \, mod \, 100 = 16$
    - $p[4] = p[3]^2 \, mod \, 100 = 16^2 \, mod \, 100 = 56$
    - $p[5] = p[4]^2 \, mod \, 100 = 56^2 \, mod \, 100 = 36$
    - $p[6] = p[5]^2 \, mod \, 100 = 36^2 \, mod \, 100 = 96$

Thủ tục bình phương: Bài tập

- Tính $6^{73}\ mod\ 100$
  - Có $b[6], b[3], b[0]$ là 1
  - b[0]: $r = r\ *\ p[0]\ mod\ 100 = 1 * 6\ mod\ 100 = 6$
  - b[3]: $r = r\ *\ p[3]\ mod\ 100 = 6 * 16\ mod\ 100 = 96$
  - b[6]: $r = r\ *\ p[6]\ mod\ 100 = 96 * 96\ mod\ 100 = 16$
- Vậy $6^{73}\ mod\ 100 = 16$

Thủ tục bình phương: Ví dụ

- Tính $12^{78} \ mod\ 25$
- Tính $15^{81} \ mod\ 50$
- Tính $8^{67} \ mod\ 10$
- Tính $25^{67} \ mod\ 70$

Giải thuật Euclide mở rộng

- Giải thuật Euclide
  - Tìm USCLN(a, b)
    - Dựa trên tính chất: nếu a > b thì

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

  - Giải thuật Euclide mở rộng
    - Tính x, y sao cho

$$a * x + b * y = USCLN(a, b)$$

    - Giải quyết bài toán tìm x sao cho

$$a * x = 1 \bmod s$$

Giải thuật Euclide mở rộng

- **Extended-Euclid (a, b)**
- Input: 2 số nguyên dương a, b
- Output: 3 số nguyên x, y, d sao cho

$$d = USCLN(a,b) \text{ và } ax + by = d$$

1. Nếu $b = 0$ thì trả về $(1, 0, a)$
2. Tìm q, r sao cho $a = b * q + r$
3. $(x', y', d) = \boldsymbol{Extended - Euclid(b, r)}$
4. Trả về $(y', x' - q * y', d)$

Giải thuật Euclide mở rộng: Ví dụ

- Dùng Euclide mở rộng tìm USCLN(120, 23)
  - **Bước 1: Extended-Euclid(120, 23)**
    - a = 120, b = 23
    - b không bằng 0.
    - q = floor(120 / 23) = floor(5.21...) = 5
    - r = 120 - 23 * 5 = 120 - 115 = 5
    - Gọi đệ quy: (x', y', d) = **Extended-Euclid(23, 5)**

Giải thuật Euclide mở rộng: Ví dụ

- Dùng Euclide mở rộng tìm USCLN(120, 23)
  - **Bước 2: Extended-Euclid(23, 5)**
    - a = 23, b = 5
    - b không bằng 0.
    - q = floor(23 / 5) = floor(4.6) = 4
    - r = 23 - 5 * 4 = 23 - 20 = 3
    - Gọi đệ quy: (x', y', d) = **Extended-Euclid(5, 3)**

Giải thuật Euclide mở rộng: Ví dụ

- Dùng Euclide mở rộng tìm USCLN(120, 23)
  - **Bước 3: Extended-Euclid(5, 3)**
    - a = 5, b = 3
    - b không bằng 0.
    - q = floor(5 / 3) = floor(1.66...) = 1
    - r = 5 - 3 * 1 = 2
    - Gọi đệ quy: (x'', y'', d) = **Extended-Euclid(3, 2)**

Giải thuật Euclide mở rộng: Ví dụ

- Dùng Euclide mở rộng tìm USCLN(120, 23)
  - **Bước 4: Extended-Euclid(3, 2)**
    - a = 3, b = 2
    - b không bằng 0.
    - q = floor(3 / 2) = floor(1.5) = 1
    - r = 3 - 2 * 1 = 1
    - Gọi đệ quy: (x', y', d) = **Extended-Euclid(2, 1)**

Giải thuật Euclide mở rộng: Ví dụ

- Dùng Euclide mở rộng tìm USCLN(120, 23)
  - **Bước 5: Extended-Euclid(2, 1)**
    - a = 2, b = 1
    - b không bằng 0.
    - q = floor(2 / 1) = 2
    - r = 2 - 1 * 2 = 0
    - Gọi đệ quy: (x', y', d) = **Extended-Euclid(1, 0)**

Giải thuật Euclide mở rộng: Ví dụ

- Dùng Euclide mở rộng tìm USCLN(120, 23)
  - **Bước 5: Extended-Euclid(2, 1)**
    - a = 2, b = 1
    - b không bằng 0.
    - q = floor(2 / 1) = 2
    - r = 2 - 1 * 2 = 0
    - Gọi đệ quy: (x', y', d) = **Extended-Euclid(1, 0)**

Giải thuật Euclide mở rộng: Ví dụ

- Dùng Euclide mở rộng tìm USCLN(120, 23)
  - **Bước 6: Extended-Euclid(1, 0)**
    - a = 1, b = 0
    - Trường hợp cơ sở được kích hoạt!
    - **Return (1, 0, 1)**
      - Vậy, x' = 1, y' = 0, d = 1.
      - **Lưu ý: d = 1 là USCLN của 120 và 23.**

Giải thuật Euclide mở rộng: Ví dụ

- Dùng Euclide mở rộng tìm USCLN(120, 23)
  - **Bước 6: Extended-Euclid(1, 0)**
    - a = 1, b = 0
    - Trường hợp cơ sở được kích hoạt!
    - **Return (1, 0, 1)**
      - Vậy, x' = 1, y' = 0, d = 1.
      - **Lưu ý: d = 1 là USCLN của 120 và 23.**

Giải thuật Euclide mở rộng: Ví dụ

- Dùng Euclide mở rộng tìm USCLN(120, 23)
  - **Trở về từ Bước 5: Extended-Euclid(2, 1)**
    - Với (x', y', d) = (1, 0, 1) từ lệnh gọi **Extended-Euclid(1, 0)**
    - x' = y' = 0
    - y' = x' - q * y' = 1 - 2 * 0 = 1
    - **Return (0, 1, 1)**
      - Vậy, x' = 0, y' = 1, d = 1.

Giải thuật Euclide mở rộng: Ví dụ

- Dùng Euclide mở rộng tìm USCLN(120, 23)
  - **Trở về từ Bước 4: Extended-Euclid(3, 2)**
    - Với (x', y', d) = (0, 1, 1) từ lệnh gọi **Extended-Euclid(2, 1)**
    - q = 1 (từ Bước 4)
    - x' = y' = 1
    - y' = x' - q * y' = 0 - 1 * 1 = -1
    - **Return (1, -1, 1)**
      - Vậy, x' = 1, y' = -1, d = 1.

Giải thuật Euclide mở rộng: Ví dụ

- Dùng Euclide mở rộng tìm USCLN(120, 23)
  - **Trở về từ Bước 3: Extended-Euclid(5, 3)**
    - Với (x', y', d) = (1, -1, 1) từ lệnh gọi **Extended-Euclid(3, 2)**
    - q = 1 (từ Bước 3)
    - x' = y' = -1
    - y' = x' - q * y' = 1 - 1 * (-1) = 1 + 1 = 2
    - **Return (-1, 2, 1)**
      - Vậy, x' = -1, y' = 2, d = 1.

Giải thuật Euclide mở rộng: Ví dụ

- Dùng Euclide mở rộng tìm USCLN(120, 23)
  - **Trở về từ Bước 2: Extended-Euclid(23, 5)**
    - Với (x', y', d) = (-1, 2, 1) từ lệnh gọi **Extended-Euclid(5, 3)**
    - q = 4 (từ Bước 2)
    - x' = y' = 2
    - y' = x' - q * y' = -1 - 4 * 2 = -1 - 8 = -9
    - **Return (2, -9, 1)**
      - Vậy, x' = 2, y’ = -9, d = 1.

Giải thuật Euclide mở rộng: Ví dụ

- Dùng Euclide mở rộng tìm USCLN(120, 23)
  - **Kết quả cuối cùng: Trở về từ Bước 1: Extended-Euclid(120, 23)**
    - Với (x', y', d) = (2, -9, 1) từ lệnh gọi Extended-Euclid(23, 5)
    - q = 5 (từ Bước 1)
    - x = y' = -9
    - y = x' - q * y' = 2 - 5 * (-9) = 2 + 45 = 47
    - **Return (-9, 47, 1)**
      - Vậy, x' = -9, y' = 47, d = 1.

Giải thuật Euclide mở rộng: Ví dụ

- Dùng Euclide mở rộng tìm USCLN(120, 23)
  - USCLN(120, 23) = 1
  - Các hệ số x và y là: x = -9, y = 47

    120 * (-9) + 23 * 47 = -1080 + 1081 = 1

Giải thuật Euclide mở rộng: Ví dụ

- Dùng Euclide mở rộng tìm x sao cho
  - $51 * x \mod 100 = 1$
  - $80 * x \mod 79 = 1$
  - 1013 * x mod 1019 = 1

Hệ thống mã hóa RSA

- Được xây dựng tại học viện MIT năm 1977
- Đặt theo tên của các tác giả: Ron Rivest, Adi Shamir và Len Adleman
- Mã hóa khối và sử dụng hàm một chiều phân tích một số thành thừa số nguyên tố
- Để đảm bảo an toàn, khuyến nghị sử dụng khóa 2048 bit hoặc lớn hơn 3072 bit trong tương lai (khởi đầu là 1024 bit)

Hệ thống mã hóa RSA

- Mã hóa và giải mã được tính theo công thức

$$C = P^e \bmod n$$

$$P = C^d \bmod n$$

- Các yêu cầu
  - Có thể tìm được các giá trị e, d, n sao cho $P^{e.d} = P \pmod n$ với mọi P < n
  - Dễ dàng tính được P^e và C^d với mọi P < n
  - Không thể tính được d từ e và n

Hệ thống mã hóa RSA

❖ Thuật toán sinh khóa RSA

1. Chọn 2 số nguyên tố lớn p và q
2. Tính $n = p * q$
3. Tính $m = \varphi(n) = (p-1).(q-1)$
4. Tìm một số e sao cho e là nguyên tố cùng nhau với m → $UCLN(e, m) = 1$
5. Tìm một số d sao cho $(e * d)\ mod\ m = 1$
6. Kết quả: khóa công khai $P_K = \{e, n\}$, khóa bí mật

$$S_K = \{d, n\}$$

Hệ thống mã hóa RSA

- Thuật toán brute-force để tìm d (d nhỏ)

  Function Compute_d(e, phi_n):

  for i trong khoảng (1, 1000) do

  $$x \leftarrow \big((i * phi\_n) + 1\big)/e$$

  $$y \leftarrow (e * x)\ \%\ phi\_n$$

  $$if\ y = 1\ return\ x$$

- Nếu d lớn thì dùng thuật toán $Extended - Euclid(e, phi\_n)$ với d là x trong $e.x + phi\_n.y = 1$

Hệ thống mã hóa RSA: Ví dụ

- Cho hệ mã RSA có $p = 11, q = 47$ và $e = 3$
  - Tìm khóa công khai và khóa bí mật
  - Sau đó mã hóa P = 26

Hệ thống mã hóa RSA: Ví dụ

- Cho hệ mã RSA có $p = 11, q = 47$ và $e = 3$
  - Tìm khóa công khai và khóa bí mật

1. $p = 11, q = 47$
2. $n = p * q = 517$
3. $m = \varphi(n) = (11-1).(47-1) = 460$
4. $1 < e < 460$ và $(3, 460)$ là nguyên tố cùng nhau
5. Dùng brute force: $d = 307$

Hệ thống mã hóa RSA: Ví dụ

- Cho hệ mã RSA có p = 11, q = 47 và e = 3
  - Tìm khóa công khai và khóa bí mật

$$P_K = (e, n) = (3, 517)$$

$$S_K = (d, n) = (307, 517)$$

  - Sau đó mã hóa P = 26

$$C = P^e \, mod \, n = 26^3 \, mod \, 517 = 515$$

Hệ thống mã hóa RSA: Ví dụ

- Cho hệ mã RSA có $p = 7, q = 19$ và $e = 5$
  - Tìm khóa công khai và khóa bí mật
  - Sau đó mã hóa P = 6
- Cho hệ mã RSA có $p = 17, q = 23$ và $e = 7$
  - Tìm khóa công khai và khóa bí mật.
  - Sau đó mã hóa P=50

Hệ thống mã hóa RSA: Ví dụ

- Cho hệ mã RSA có $p = 61, q = 53$
  - Tìm khóa công khai và khóa bí mật
  - Sau đó mã hóa P = 123
- Cho hệ mã RSA có $p = 43$ và $q = 59$
  - Tìm khóa công khai và khóa bí mật
  - Sau đó mã hóa P=150

Hệ thống mã hóa RSA: Ví dụ

- Cho hệ mã RSA có $p = 7, q = 11$.
- Giả sử An dùng khóa công khai $P_k = (17, 77)$.
  - Tìm khóa bí mật.
  - Biết rằng ký tự từ A - Z biểu diễn bằng số nguyên từ 0 - 25, dấu cách được biểu diễn bằng số 26.
  - Bảo gửi An mẩu tin “HELLO WORD”. Bản mã tương ứng là gì?

Hệ thống mã hóa RSA

- Để tiện cho việc giao dịch trên mạng có sử dụng truyền tin mật, người ta lưu trữ các khóa công khai của các người dùng tại một điểm công cộng
- Độ an toàn của RSA dựa vào độ phức tạp của bài toán phân tích một số nguyên dương cho trước n thành hai thừa số nguyên tố p và q

Hệ thống mã hóa RSA

- Lựa chọn p, q
  - Đảm bảo rằng bài toán phân tích thừa số nguyên số PTTSNT(n) thật sự khó
  - Tránh tình trạng p, q rơi vào những trường hợp đặc biệt → bài toán dễ dàng

    Ví dụ: p - 1 có các thừa số nguyên tố nhỏ
  - p, q phải có độ dài tối thiểu 512 bit p, q xấp xỉ nhau

Hệ thống mã hóa RSA

- Lựa chọn e
  - e nhỏ nhất có thể
  - e không quá nhỏ để tránh bị tấn công theo dạng "low exponent"
- Lựa chọn d
  - d không quá nhỏ ($d < \frac{n}{4}$) để tránh tấn công dạng "low decryption"

Hệ mã hóa Elgamal

- Được T. ElGamal giới thiệu vào năm 1984 dựa trên cơ sở ý tưởng từ Diffie-Hellman
- Được sử dụng trong việc mã hóa dữ liệu, chữ ký số, trao đổi khóa
- Tính an toàn dựa trên tính khó giải của bài toán Logarit rời rạc

Hệ mã hóa Elgamal

❖ Thuật toán sinh khóa (bên nhận và bên gửi)

- Chọn một số nguyên tố lớn p (thường có độ dài từ 1024 đến 2048 bit) và hai số nguyên ngẫu nhiên ε và a, cả hai đều nhỏ hơn p

$$y = \varepsilon^{a} (mod\, p)$$

- Khóa công khai được lấy là (p, ε, y)
- Khóa bí mật là a

Hệ mã hóa Elgamal

❖ Thuật toán sinh mã (bên gửi)

➢ Chọn giá trị $k(k < p)$ và tính toán khóa

$$K = y^k mod\, p$$

➢ Tính cặp mã, trong đó P là bản rõ

$$C_1 = \varepsilon^k mod\, p$$

$$C_2 = K.Pmod\, p$$

➢ Cặp (C_1, C_2) được gửi đi, đồng thời k bị hủy đi

Hệ mã hóa Elgamal

- Thuật toán giải mã (bên nhận)
  - Nhận được cặp mã (C_1, C_2) thực hiện các bước Khôi phục bản rõ

$$P = \frac{C_2}{C_1^a} mod\ p$$

với $\left((C_1^a)^{-1}\right) mod\ p = \left(C_1^{p-a-1}\right) mod\ p$

Hệ mã hóa Elgamal: Ví dụ

- Trước khi bắt đầu truyền tin, An chọn $p = 97$, chọn ngẫu nhiên $\varepsilon = 5, a = 58$.
- Bình muốn gửi cho An một tài liệu mật $P = 3$, Bình chọn ngẫu nhiên $k = 36$

Hệ mã hóa Elgamal

- Ưu điểm:
  - Độ an toàn của mã hóa bất đối xứng cao
  - Cung cấp được tính chứng thực, toàn vẹn dữ liệu
  - Thuận tiện phân phối khóa
- Hạn chế:
  - Xử lý chậm hơn so với mã hóa đối xứng
  - Gặp khó khăn nếu mất khóa bí mật
  - Phức tạp trong vấn đề tìm số nguyên tố và ngẫu nhiên hợp lý


Thank You