CHƯƠNG 4:
HỆ MÃ HÓA KHÓA CÔNG KHAI PKC – PUBLIC KEY CRYPTOSYTEMs

Chương 4:
Hệ mã hóa khóa công khai

Giới thiệu

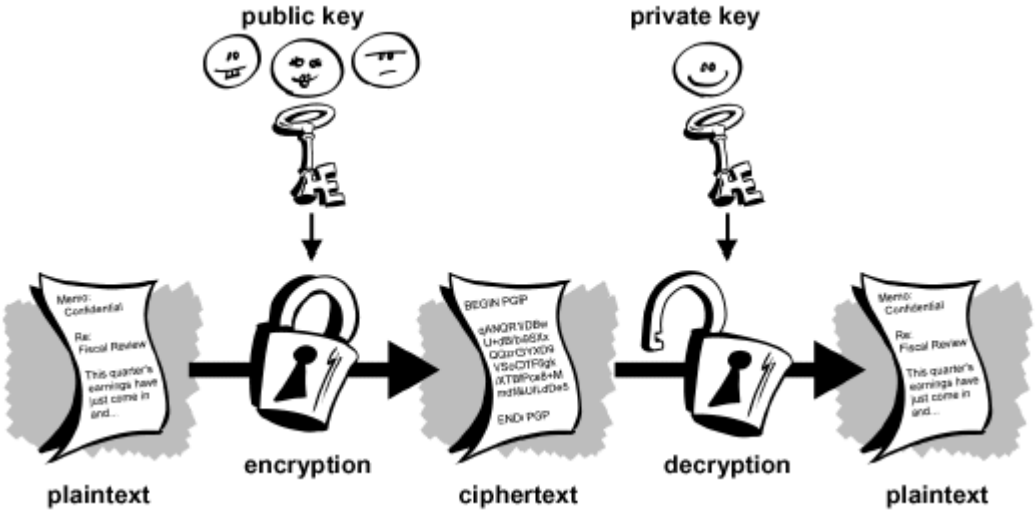
- Ý tưởng về hệ thống mã hóa khóa công khai được Martin Hellman, Ralph Merkle và Whitfield Diffie tại Đại học Stanford giới thiệu vào năm 1976.
- Sau đó, phương pháp Diffie-Hellman của Martin Hellman và Whitfield Diffie đã được công bố.
- Năm 1977, trên báo "*The Scientific American*", nhóm tác giả Ronald Rivest, Adi Shamir và Leonard Adleman đã công bố phương pháp RSA, phương pháp mã hóa khóa công khai nổi tiếng và được sử dụng rất nhiều hiện nay trong các ứng dụng mã hóa và bảo vệ thông tin

Chương 4:
Hệ mã hóa khóa công khai

4.1. Khái niệm hệ mã hóa PKC

Nguyên lý cơ bản của các hệ mã khóa công khai

- Hệ mã khóa công khai là hệ mã dùng 2 khóa:
  - Khóa công khai để mã hóa
  - Khóa bí mật để giải mã



Chương 4:
Hệ mã hóa khóa công khai

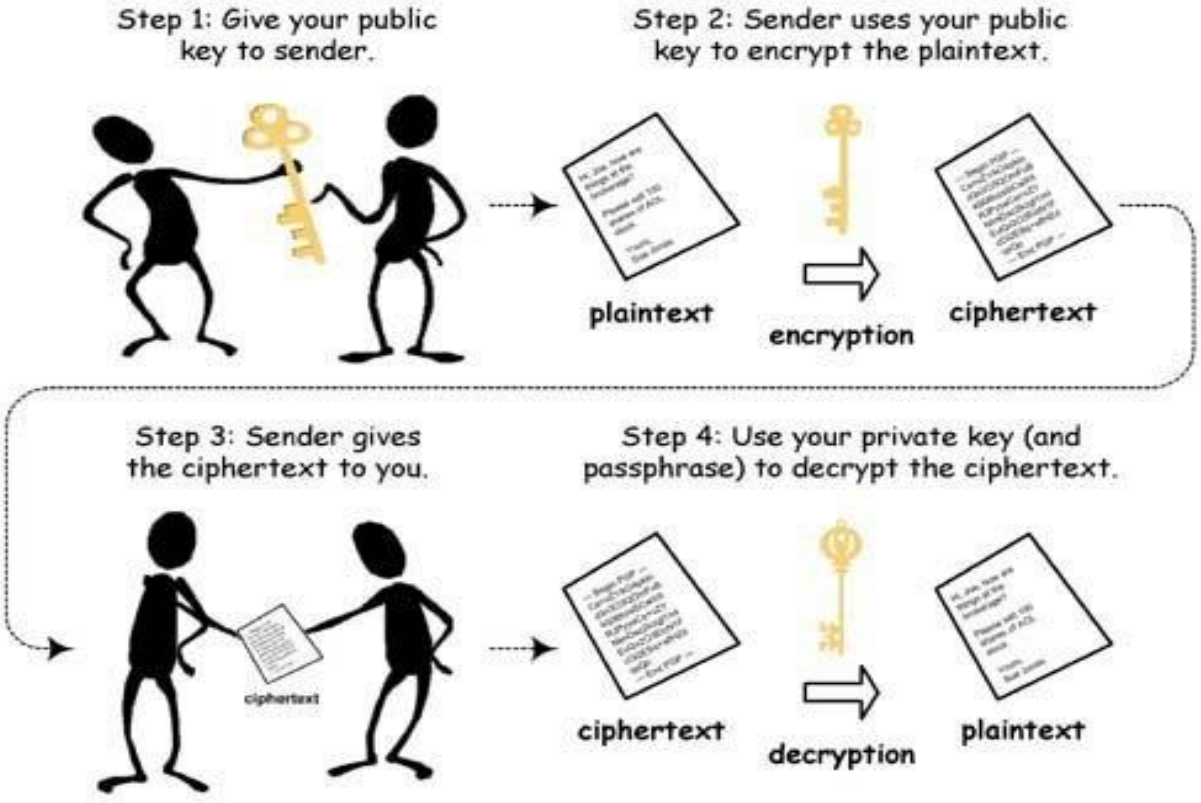
Nguyên lý hoạt động

Trong các hệ mã hóa khóa công khai, **A** và **B** muốn trao đổi thông tin thì sẽ thực hiện theo sơ đồ sau:

- Trong đó **B** sẽ chọn khóa k=(k', k''). **B** sẽ gửi khóa lập mã k' cho **A** (*được gọi là khóa công khai – public key*) qua một kênh bất kỳ và giữ lại khóa giải mã k'' (*được gọi là khóa bí mật – private key*).
- **A** có thể gửi văn bản *M* cho **B** bằng cách lập mã theo một hàm $e_{k'}$ nào đó với khóa công khai k' của **B** trao cho và được bản mã *M'* = $e_{k'}(M)$. Sau đó gửi *M'* cho **B.**
- Đến lược **B** nhận được bản mã *M'* sẽ sử dụng một hàm giải mã $d_{k''}$ nào đó với khóa bí mật k'' để lấy lại bản gốc M= $d_{k''}$(M')

Chương 4:
Hệ mã hóa khóa công khai

Hình vẽ minh họa – Nguyên lý hoạt động



Chương 4:
Hệ mã hóa khóa công khai

4.2. Giới thiệu một số giải thuật PKC

- Trapdoor Knapsack
- RSA
- Elgama

Chương 4:
Hệ mã hóa khóa công khai

4.2. Giới thiệu một số giải thuật PKC

- Trapdoor Knapsack
- RSA
- Elgama

Chương 4:
Hệ mã hóa khóa công khai

4.2.1. Hệ mã Trapdoor Knapsack (Merkle – Hellman)

Trapdoor Knapsack dựa trên bài toán đóng thùng. Năm 1978, hai nhà toán học Merkle – Hellman đã đề xuất một thuật toán mã hóa PKC dựa trên bài toán ĐÓNG THÙNG như sau:

- Cho một tập hợp các số dương a_i, $1 \leq i \leq n$ và 1 số T dương. Hãy tìm 1 tập hợp chỉ số $S \subset \{1, 2,...,n\}$ sao cho: $\sum_{i \in S} a_i = T$

Bài toán này là một bài toán khó, theo nghĩa là chưa tìm được thuật toán nào tốt hơn là thuật toán thử-vét cạn.

- Thời gian xử lý vét cạn có thể tỉ lệ lũy thừa theo kích thước input n

Chương 4:
Hệ mã hóa khóa công khai

Trapdoor Knapsack

VD: (a_1, a_2, a_3, a_4) = (2, 3, 5, 7) và T=7. Hỏi có bao nhiêu trường hợp nhặt ra trong tập a_i để tổng giá trị bằng T?

Như vậy ta có 2 đáp số:

1. S=(1, 3)
2. S=(4)

Chương 4:
Hệ mã hóa khóa công khai

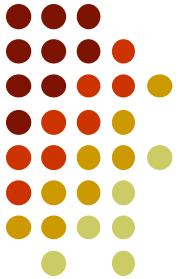
Trapdoor Knapsack

VD: $(a_1, a_2, a_3, a_4, a_5)$ = (2, 3, 5, 7, 12) và T=12. Hỏi có bao nhiêu trường hợp nhặt ra trong tập a_i để tổng giá trị bằng T?

Như vậy ta có 3 đáp số:

1. S = (1, 2, 4)
2. S = (3, 4)
3. S = (5)

Chương 4:
Hệ mã hóa khóa công khai



Trapdoor Knapsack

Từ bài toán đóng thùng này chúng ta sẽ khảo sát các khả năng vận dụng để tạo ra thuật toán mã hoá PKC. Sơ đồ đầu tiên như sau:

- Chọn một vector a = $(a_1, a_2, \ldots, a_n)$ – được gọi là vector mang (cargo vector)
- Với một khối tin X = $(X_1, X_2, \ldots, X_n)$ ta thực hiện phép mã hóa như sau: T = $\sum a_i X_i$ (*)
- Việc giải mã là: Cho mã T, vector mang a, tìm các X_i thỏa mãn (*)

Sơ đồ này thể hiện một hàm one-way với việc sinh mã rất dễ dàng nhưng việc giải mã là rất khó → cơ sở xây dựng một trapdoor

Chương 4:
Hệ mã hóa khóa công khai

Trapdoor Knapsack

- Merkle sử dụng một mẹo là áp dụng một vector mang đặc biệt là vector siêu tăng (super-increasing).
- Thành phần i+1 là lớn hơn tổng giá trị của các thành phần đứng trước nó (từ 1→ i)
- Việc giải mã có thể diễn ra dễ dàng như ví dụ bằng số sau:
  - *Vector mang siêu tăng: a=(1, 2, 4, 8)*
  - *Cho T=11, ta sẽ thấy việc tìm X = (X_1, X_2, …, X_n) sao cho T = $\sum a_i X_i$ là dễ dàng.*
  - *Đặt T_0= T*
  - X_4=1 $T_1 = T_0$-X_4*a_i=3 → (X_1 X_2 X_3 1)
  - X_3=0 $T_2 = T_1$ = 3 → (X_1 X_2 0 1)
  - X_2=1 $T_3 = T_2$ - 2 = 1 → (X_1 1 0 1)
  - X_1=1 → (1 1 0 1)

Chương 4:
Hệ mã hóa khóa công khai

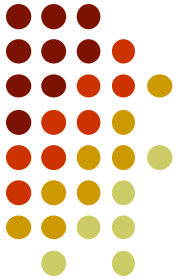
Trapdoor Knapsack

Bài toán được giải quyết dần qua các bước như sau:

- Ở bước i, tổng đích là T_i (tức là phải tìm các a_j để tổng bằng T_i)
- Ta đem so sánh T_i với thành phần lớn nhất trong phần còn lại của vector:
  - *Nếu lớn hơn thì thành phần này được chọn, tức là X_i tương ứng bằng 1*
  - *Ngược lại thì X_i tương ứng bằng 0*
- Sau đó tiếp tục chuyển sang bước sau với $T_{i+1} = T_i - X_i$

Cần chủ động "ngụy trang" vector siêu tăng để chỉ người chủ mới biết còn người ngoài không thể giải mã được.

Chương 4:
Hệ mã hóa khóa công khai



Hệ PKC Merkle – Hellman: Cơ chế ngụy trang

Tạo khóa

- Alice chọn một vector siêu tăng

  a' = (a_1', a_2', …, a_n')

  a' được giữ bí mật tức là một thành phần của khóa bí mật

Sau đó chọn một số nguyên m > $\sum a_i$', gọi là modulo đồng dư và một số nguyên ngẫu nhiên ω, gọi là nhân tử, sao cho nguyên tố cùng nhau với m (gcd(m, ω)=1)

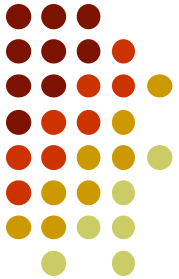
Khóa công khai của Alice sẽ là vector a là tích của a' với nhân tử ω

a = (a_1, a_2, …, a_n)

a_i = ω x a_i' (mod m) với i = 1, 2, 3...n

- Còn khóa bí mật sẽ là (a', m, ω)

Chương 4:
Hệ mã hóa khóa công khai



Mã hóa (sinh mã):

Khi Bob muốn gửi một thông báo X= $\{x_1, x_2, \ldots, x_n\}$ cho Alice, anh ta tính mã theo công thức:

$$T = \sum a_i x_i$$

Giải mã:

Alice nhận được T và giải mã như sau:

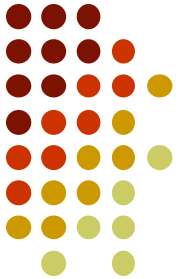
Để bỏ lớp ngụy trang cô ta trước hết tính ω^{-1} (là giá trị nghịch đảo của ω, tức là $\omega.\omega^{-1}$= 1 mod m, rồi tính

$$T' = T \times \omega^{-1} \pmod{m}$$

Alice biết rằng $T' = a' . X$ nên cô ta có thể dễ dàng giải ra được X theo siêu tăng a'

Chú thích: ở đây ta có $T' = T \times \omega^{-1} = \sum a_i X_i \omega^{-1} = \sum a_i' \omega X_i \omega^{-1}$

Chương 4:
Hệ mã hóa khóa công khai



Bài tập:

Cho hệ mã Knapsack có A’ = (2, 3, 6, 12, 25), n=5, m=53, ω = 46, ω^{-1} = 15

a) Hãy tìm các khóa của hệ mã trên

b) Mã hóa và giải mã bản mã tương ứng của bản rõ M = 01001

Chương 4:
Hệ m

Khóa công khai của Alice sẽ là vector a là tích của a' với nhân tử ω

$a = (a_1, a_2, \ldots, a_n)$

$a_i = \omega \times a_i' \pmod m$ với i = 1, 2, 3...n

Còn khóa bí mật sẽ là (a', m, ω)

Mã hóa (sinh mã):

Khi Bob muốn gửi một thông báo $X = \{x_1, x_2, \ldots, x_n\}$ cho Alice, anh ta tính mã theo công thức:

$$T = \sum a_i x_i \bmod m$$

Bài giải:

a) Hãy tìm các khóa của hệ mã trên

Khóa công khai

$a = (a_1, a_2, \ldots, a_n) = (2, 3, 6, 12, 25)' \times \omega$

$a_i = \omega \times a_i' \pmod m = (39, 32, 11, 22, 37)$

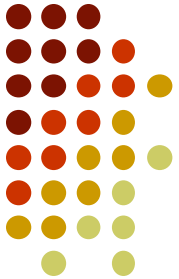
Khóa bí mật: (a', m, ω)

b) Mã hóa bản rõ M = 01001

$$T = \sum a_i X_i = 0*39 + 1*32 + 0*11 + 0*22 + 1*37$$

$$= 69 \bmod 53 = 16$$

Chương 4:
Hệ mã hóa khóa công khai



Bài giải:

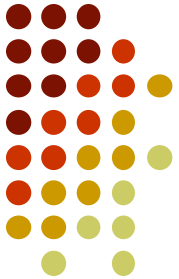
Giải mã: với a'=(2, 3, 6, 12, 25), $\omega^{-1} = 15$,

T' = 16*15 mod 53 = 28

$\rightarrow$

$$X5 = 1$$
$$X4 = 0$$
$$X3 = 0$$
$$X2 = 1$$
$$X1 = 0$$
$$M = 01001$$

Chương 4:
Hệ mã hóa khóa công khai



Trong trường hợp mã hóa

M = 1 0 1 0 1 0: tự tìm a, a', m, ω, ω^{-1}

Chương 4:
Hệ mã hóa khóa công khai

Độ an toàn của Trapdoor – Knapsack

Brute Force Attack

- Với những kẻ không biết trapdoor (a', m, ω), giải mã đòi hỏi phải tìm kiếm vét cạn qua 2^n khả năng của X

Sự đổ vỡ của giải pháp dùng Knapsack (1982-1984)

- Shamir – Adleman đã chỉ ra chỗ yếu của giải pháp này bằng cách đi tìm một cặp (ω', m') sao cho nó có thể biến đổi ngược a về a' (từ Public key về Private key).
- Năm 1984, Brickell tuyên bố sự đổ vỡ của hệ thống Knapsack với dung lượng tính toán khoảng 1 giờ máy Cray -1, với 40 vòng lặp chính và cỡ 100 trọng số.