CHƯƠNG 4
HỆ MẬT MÃ KHÓA CÔNG KHAI (HỆ MẬT BẤT ĐỐI XỨNG)

4.1. Khái niệm

4.1.1. Vấn đề sử dụng và phân phối khóa

Hệ mật bất đối xứng khắc phục được tính chất phức tạp trong việc phân phối khóa ở hệ mật đối xứng

Cho phép giao tiếp giữa các đối tượng một cách uyển chuyển , dễ dàng.

Sử dụng hai khoá K_p (public key) và K_s (private key) để mã và giải mật

Có hai mode làm việc :

Bảo mật : Mã bằng public key → giải mật bằng private key

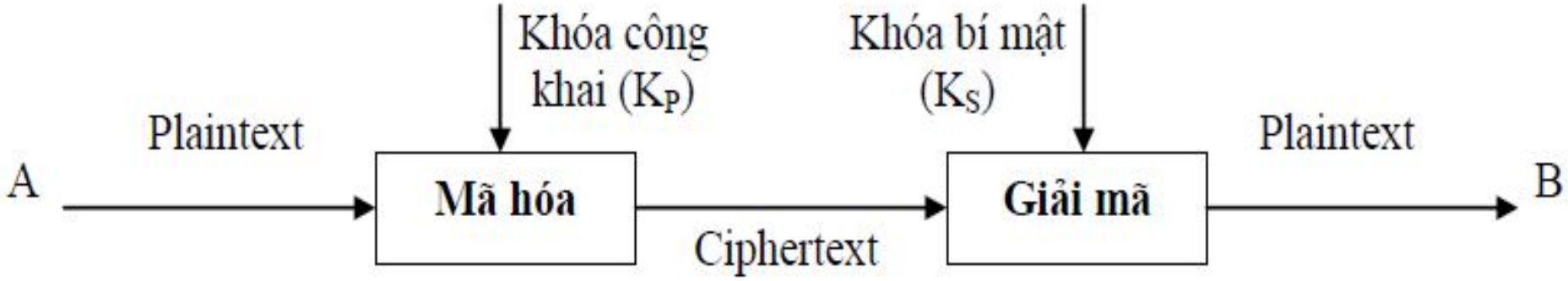
Xác thực : Mã bằng private key →giải mật bằng public key

4.1.2. Các yêu cầu của loại hệ mã PKC

- Việc sinh K_P, K_S phải dễ dàng
- Việc tính $E(K_P, M)$ là dễ dàng
- Nếu có $C = E(K_P, M)$ và K_S thì dễ ràng giải mật .
- Nếu biết K_P thì việc dò tìm K_S là khó
- Rất khó tìm bản rõ từ bản mã nếu không biết khóa .

4.1.3. các mô hình sử dụng PKS

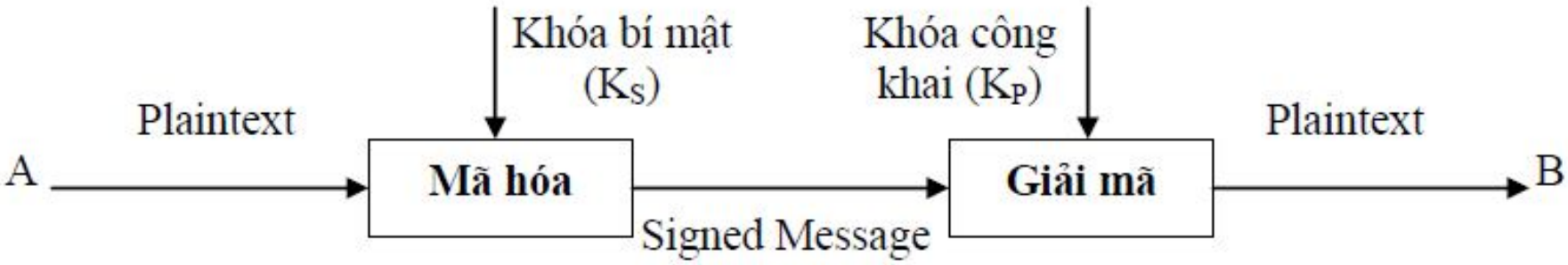
4.1.3.1. Mô hình bảo mật



Hình 4.1: Mô hình sử dụng 1 của các hệ mã khóa công khai PKC

$$\text{Ciphertext} = E(K_P,P)\ ,\quad \text{Plantext} = D(K_S, E(K_P,P))$$

4.1.3.2.Mô hình xác thực



Hình 4.2: Mô hình sử dụng 2 của các hệ mã khóa công khai PKC

$$\text{Ciphertext} = D(K_S, P)\ ,\ \text{Plaintext} = E(K_P, D(K_S, P))$$

4.1.4. Cấu trúc của PKC

- PKC được xây dựng trên các hàm một chiều (one–way functions).
- OWHF $f : X \rightarrow Y$ là hàm nếu biết $x \in X \rightarrow$ dễ dàng tính $y = f(x)$. Nhưng $\forall y \in Y$ việc tìm $x \in X : y = f(x)$, có nghĩa tìm hàm ngược f^{-1} là rất khó.
- Ví dụ : với $P \in \{ P_1, P_2, ..., P_n \}$ thì việc tính $N = P_1 * P_2 * ... * P_n$ là dễ tìm $P_i \in \{P\}$ với N đủ lớn (phân tích ngược – phân rã SNT) là một bài toán khó .
- Trong các hệ mã PKC sử dụng các "trapdoor" giúp cho việc tìm $x : y = f(x)$ dễ dàng . Hàm (trapdoor function): là một hàm một chiều trong đó việc tính f^{-1} là rất nhanh khi chúng ta biết được "trapdoor".

4.1.5.Một số hệ mật mã bất đối xứng thông dụng

- Hệ mã Knapsack (xếp ba lô)
- RSA (Rivest, Adi Shamir, and Leonard Adleman) . RSA dùng để bảo mật và tạo “digital signatures” .
- Diffie-Hellman “Diffie-Hellman key exchange” được sử dụng để truyền khóa mật mã trên kênh công khai , không dùng để mã hoá thông điệp .
- ECC The Elliptic Curve Cryptosystem (ECC) được sử dụng trên các thiết bị nhỏ , ít thông minh như “ cell phones” và “wireless”.
- El Gamal thuật giả dùng để truyền “digital signatures” và “ key exchanges”(Cũng tương tự Diffie-Hellman “. The El Gamal còn được gọi là DSA .

4.2.Hệ mã Knapsack

- Hệ mã knapsack do Merkle và Hellman (năm 1978).

4.2.1. Bài toán xếp ba lô

- Cho M, N và A_1, A_2, ..., A_N là các số nguyên dương
  Hỏi có tồn tại một véc tơ nhị phân $x=(x_1, x_2,..., x_N)$ sao cho:

$$M = \sum_{i-1}^{N} xi * Ai$$

- Vectơ A = (A_1, A_2, ..., A_N) gọi là vectơ "xếp balô"
- Vectơ X = (x_1, x_2, ..., x_N) là vectơ nghiệm.

- Đây là bài toán khó có thời gian là hàm mũ $O(2^N)$.
- Nếu S là dãy siêu tăng thì bài toán trên giải được với thời gian tuyến tính O^N.
- Vector siêu tăng : *Dãy $A=(A_i)$ gọi là siêu tăng nếu với mọi $A_i>\Sigma A_j$ $(j=1,..i-1)$ (tức là phần tử đứng sau lớn hơn tổng các phần tử đứng trước nó)*
- Khi đó bài toán balo được phát biểu như sau:

  Cho M, N và $A'=(A'_1, A'_2, ..., A'_N)$ là một dãy siêu tăng. *Hỏi có tồn tại một véc tơ nhị phân $x=(x_1, x_2,..., x_N)$ sao cho:*

$$M=\Sigma_{i=1} x_i A_i \quad (i=1..N))$$

- Vecto xếp ba lô siêu tăng

  - Một trường hợp riêng đáng quan tâm của bài toán xếp ba lô tổng quát là trường hợp mà $x_i \in \{0, 1\}$. Khi đó ta có bài toán "xếp ba lô" 0, 1.
  - Trong trường hợp vecto $(A_1, A_2, ..., A_N)$ được sắp lại thành $(A'_1, A'_2, ..., A'_N)$ sao cho:

    $\forall i$ ta có : $\sum_{j<i} A'_j < A'_i$ thì vecto $(A_1, A_2, ..., A_N)$ được gọi là vecto xếp balo siêu tăng.
  - Khi $(A'_1, A'_2, ..., A'_N)$ là một vecto "xếp balo" siêu tăng ta có ngay tính chất : $\forall i : M \geq A'$. Do đó việc giải bài toán xếp ba lô 0/1 trở nên dễ dàng hơn rất nhiều.

- Thuật giải bài toán xếp balô

For i:=N downto 1 do

Begin

If $M>=a_i$ then

$x_i=1$

else $x_i:=0$;

$C := C - x_i.a_i$;

end;

If C=0 then “bài toán có đáp án là véc tơ x”

else “bài toán không có đáp án”;

4.2.2.Cách xây dựng hệ mã knapsack

1.Chọn 1 vecto siêu tăng A' = (a'_1, a'_2, ..., a'_N),

2. Chọn M > 2 * a'_N, chọn ngẫu nhiên u < M : (u, M) = 1

3.Xây dựng Vecto S = (s_1, s_2, ..., s_N) với s_i = (a'_i * u) mod M

4.Khóa: K_P = (S, M), K_S = (u, u^{-1})

5.Không gian rõ : dãy N bit : P = (x_1, x_2, ..., x_N).

6.Mã hóa : $$C = (\sum_{ki=1}^{N} si * xi) \ mod \ M$$

7.Giải mã: tính C' = C * u^{-1} mod M sau đó giải bài toán xếp ba lô 0/1 với A', C' từ đó tìm được

$$P = (x_1, x_2, \ldots, x_N).$$

Ví dụ Knapsack

Cho hệ mã Knapsack có A' = (2, 3, 6, 12, 25), N = 5, M = 53, u = 46, u^{-1} = 15.

- Hãy tìm các khóa của hệ mã trên
- Mã hóa và giải mã bản mã tương ứng của bản rõ P = (x_1 x_2 x_3 x_4 x_5)= 01001.

a.Tìm khóa : K_p = (S,M) ; S = (s_1, s_2 ,...s_N)= a'$_1$ u , a'$_2$ u...= 2*46, 3*46 , ...25*46 = (39,32,11,22,37) ; M=53 ;
K_s = (u, u^{-1}) = (4,15)

b. Mã hóa : Tính $C = (\sum_{i=1}^{N} xi * si) mod M$

c.Giải mã : Tính $C' = C * u^{-1}$ $Tính\ P = (\sum_{i=1}^{N} c'i * si) mod M$

4.3. Hệ mật RSA

Hệ mã RSA (Rivest, Shamir và Adleman) là thuật toán PKC nổi tiếng và được ứng dụng nhiều trong thực tế nhất.

4.3.1. Định lý RSA

- Cho p,q là hai SNT phân biệt N=pq
- Có một hàm $\varphi = \varphi(n)=(p-1)(q-1)$, $1<e<\varphi$, $(e, \varphi)=1$,
  Tính được : $d \equiv e^{-1} \bmod \varphi$, $1<d< \varphi$,
- Cho một số m : $0 \leq m < N$, và tính $c = m^e \bmod N$

  Thì : $m = c^d \bmod N$

4.3.2. Thuật giải RSA

4.3.2.1.Phát sinh khóa RSA

a. Tính N = p*q và $\varphi = \varphi(n)=(p-1)(q-1)$; (p,q là hai SNT phân biệt đủ lớn .Trong thực tế >100 chữ số).

b. Chọ ngẫu nhiên một số $e \in]1,\varphi[$ thoả $(e,\varphi)=1$.

c. Sử dụng thuật giải Bezout tính số nghịch đảo

$d \in]1,\varphi[= e^{-1} \bmod \varphi$; $ed \equiv 1 \bmod \varphi$ hay $\frac{ed-1}{\varphi} = k$

d. Cặp (e ,N) là khóa công khai (K_p)

Cặp (d,N) là khóa các nhân – khóa bí mật (K_s)

4.3.2.2 Mã hóa và giải mã

1. Mã hóa

   a. Tạo cặp khóa công khai (e,N), và một thông điệp rõ dưới dạng một số nguyên dương m ;

   $m \in [0,N[$, m – văn bản rõ (plaintext).

   b. Tính c

   $c = m^e \bmod N$, c – văn bản mật (ciphertext).

2. Giải mật

   Phục hồi lại văn bản rõ m từ văn bản bảo mật c, ta sử dụng cặp khóa cá nhân (d,N) để tính m;

   $m = c^d \bmod N$.

Ghi chú : RSA sử dụng các sô nguyên tố lớn p,q để việc phân tích N với (N= pq) là vô cùng khó khăn.

4.3.2.3. Độ an toàn của RSA

- Độ an toàn của RSA phụ thuộc vào độ khó của việc tính $\varphi(N)$ .Muốn vậy , cần phân tích N ra thừa số nguyên tố.
- Thuật toán Brent-Pollard là thuật toán phân tích số nguyên tố hiệu quả nhất hiện nay.(Bảng thống kê 4.7)
- Việc sử dụng RSA cần tới các số nguyên tố lớn nên phải có một cơ sở dữ liệu các số nguyên tố.
- Tốc độ RSA chậm do phải tính số lượng lớn các phép nhân. Phép nhân 2 số n bit cần thực hiện $O(n^2)$ phép tính bit. Thuật toán nhân các số nguyên Schonhage – Strassen cho phép nhân 2 số với độ phức tạp là O(n log n)

SỐ CHỮ SỐ HỆ THẬP PHÂN TRONG N	SỐ THAO TÁC BIT ĐỂ PHÂN TÍCH N
20	7.20e+03
40	3.11e+06
60	4.63e+08
80	3.72e+10
100	1.97e+12
120	7.69e+13
140	2.35e+15
160	5.92e+16
180	1.26e+18
200	2.36e+19

Bảng 4.1: Tốc độ của thuật toán Brent-Pollard

- Hiện tượng lộ bản rõ

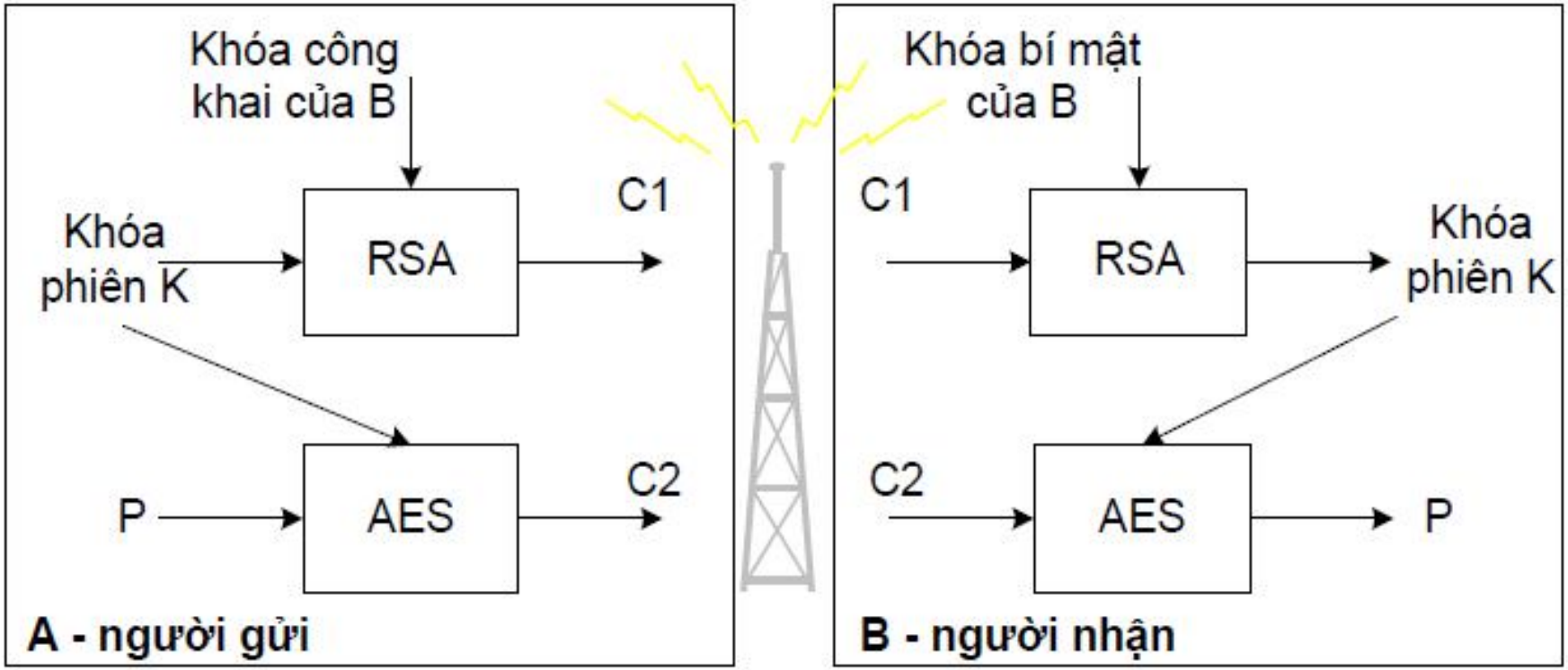
  ✓ Hệ mã RSA có N = p*q = 5*7, e = 17, với m = 6 ta có C = 617 mod N = 6.

  ✓Hệ mã RSA có N = p*q = 109*97, e = 865, với mọi m ta đều có m^e mod N = M.

  ✓Với hệ mã RSA có N = p*q và e bất kỳ, số lượng bản rõ bị lộ mã hóa sẽ là (1 + (e-1, p-1))*(1 + (e-1, q-1)).

  • Trong thực tế RSA thường được sử dụng với các thông điệp có kích thước nhỏ (secsion key), và thường sử dụng lai ghép với các hệ mật đối xứng (DES,AES…)

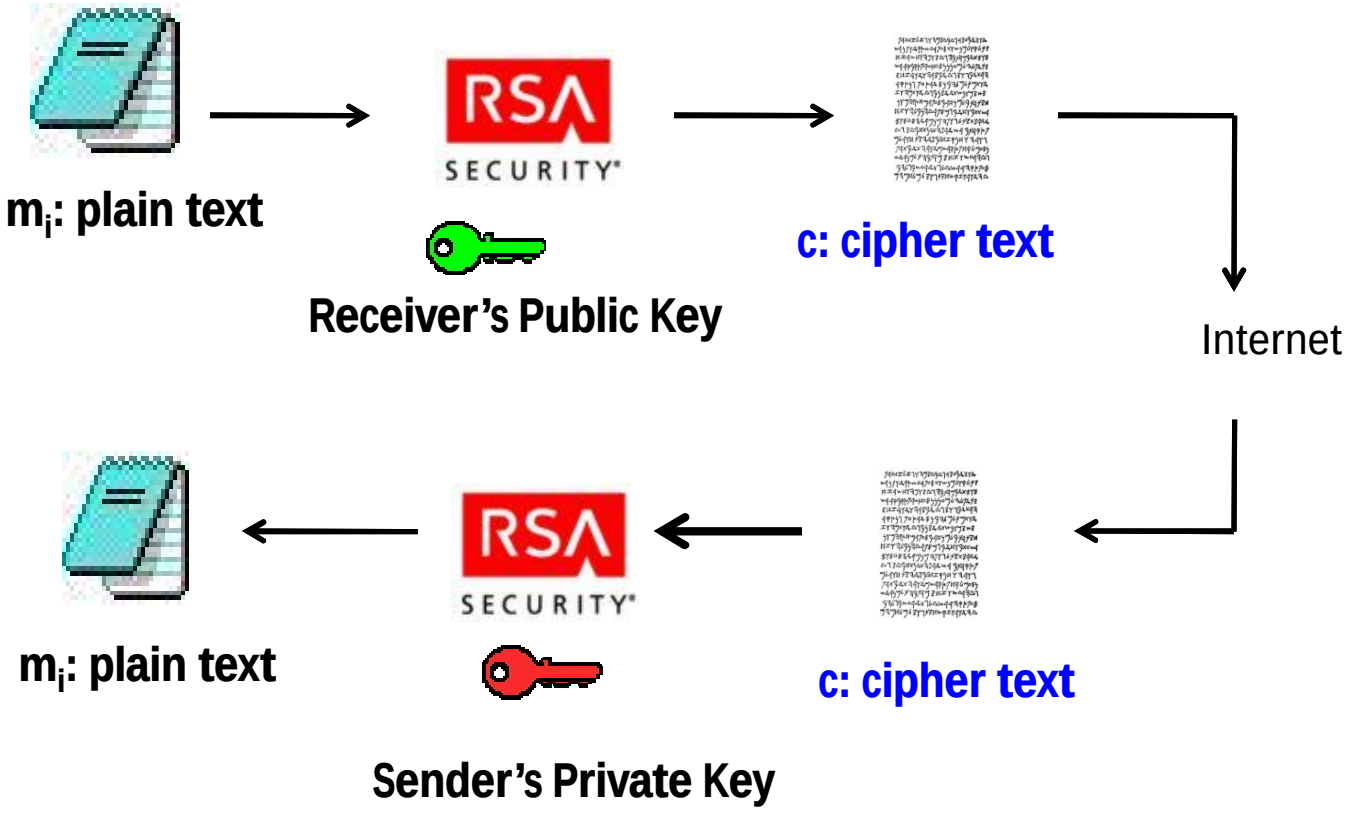
Sơ đồ lai của RSA với hệ mật đối xứng



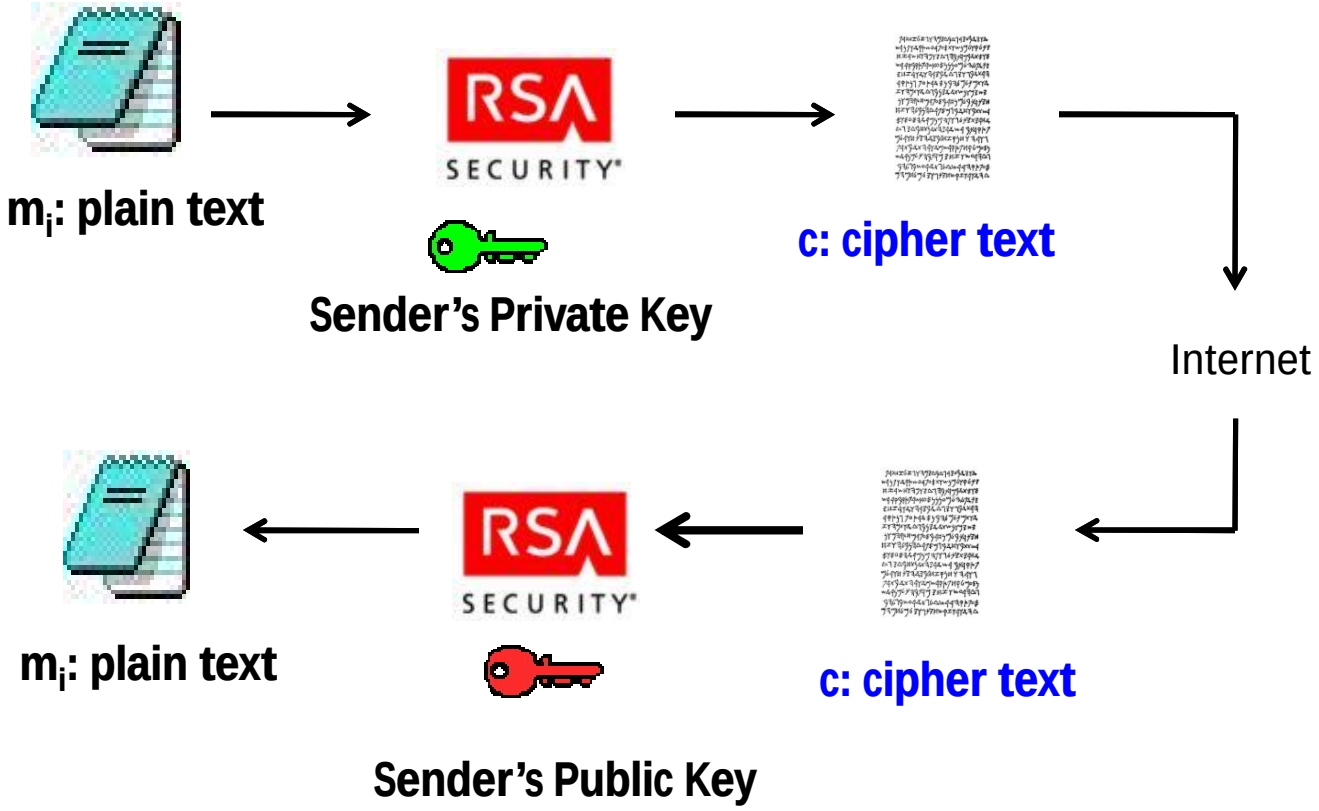
Hình 4.3: Mô hình ứng dụng lai ghép RSA với các hệ mã khối

4.3.2.4. Ứng dụng của RSA

a. Bảo mật thông điệp : Sử dụng khoá công khai của bên nhận để mã , khoá riêng của bên nhận để giải mã



b. Xác thực thông điệp : Dùng khoá cá nhân của bên gửi để mã , khoá công khai của bên gửi để giải mã



4.3.2.5. Phạm vi ứng dụng RSA

- Mạng hành chính công , E-Business , E-Goverment
- Kinh doanh thương mại điện tử : Thanh toán điện tử,bảo mật các dữ liệu điện tử,chứng thực chữ ký điện tử. . .
- Đào tạo ,thi cử từ xa,bảo mật dữ liệu tuyển sinh.
- Ngân hàng thương mại : Giao dịch, thanh toán qua mạng.
- Xuất nhập cảnh
- . . . . . .

4.3.4. Hệ mã Difie-Henman

- Được sử dụng trong các cơ chế phân phối khóa trong hệ mật đối xứng.

a. Tạo khóa

- Ta có p là số nguyên tố ($p \epsilon Z_p$) .
- Giả sử $\alpha \in Z_p$ là một số nguyên thuỷ (primitive element)
- Các giá trị p và α được công bố công khai trên mạng.
- UID thông tin định danh hợp lệ cho từng user U trên mạng ("tên"," e-mail address"," telephone number"...)
- Từng "user U,V" có một số mũ a_u, a_V với $(0 \leq a_u, a_V \leq p-2)$, và tính giá trị b_U, b_V công khai tương ứng :

$$b_U = \alpha^{a}{}_{u} \bmod p \text{ và}$$

$$b_V = \alpha^{a}{}_{v} \bmod p$$

- Khoá chung $K_{u,v}$ được tính $K_{u,v} = \alpha^{a}{}_{u}, {}^{a}{}_{v} \, mod p$

b. Thuật giải

- Input : *p* SNT và α primitive element $\in Z^*_p$ → *truyền công khai trên mạng*

  Từng "*user U,V*" có một số mũ a_u ,a_v với *:*

  $$(0 \le a_u , a_v \le p-2),$$

- Output *:*

  Hai bên cùng tính $b_u = \alpha^{a_u} \bmod p$ và $b_v = \alpha^{a_v} \bmod p$

  Hai bên gửi cho nhau : b_u và b_v.

  1. Bên V tính : $K_{U,V} = \alpha^{a_u , a_v} \bmod p = b_u^{a_v} \bmod p$

     Dùng b_U từ U cùng với giá trị mật a_u

  2. Bên U tính : $K_{U,V} = \alpha^{a_u , a_v} \bmod p = b_v^{a_u} \bmod p$

     Dùng b_V gửi từ V cùng với giá trị mật a_v

c. Ví dụ Diffie- Hellman

- Giả sử p = 25307 và α = 2 biết công khai (p là SNT và α là số nguyên thuỷ gốc modulo p).
- User U Chọn a_U = 3578. Tính

$$b_U = \alpha^{a_U} \bmod p$$
$$= 2^{3578} \bmod 25307$$
$$= 6113,$$

Dùng để chứng nhận U

- User V chọn a_V = 19956. Tính

$$b_V = \alpha^{a_V} \bmod p$$
$$= 2^{19956} \bmod 25307$$
$$= 7984,$$

Dùng để chứng nhận V

Ví dụ Diffie- Hellman (tiếp)

- User U tính khoá của mình

$$\begin{aligned} K_{U,V} &= b_V{}^{a_U} \bmod p \\ &= 7984^{3578} \bmod 25307 \\ &= 3694, \end{aligned}$$

- User V tính khoá của mình

$$\begin{aligned} K_{U,V} &= b_U{}^{a_V} \bmod p \\ &= 6113^{19956} \bmod 25307 \\ &= 3694. \end{aligned}$$

4.3.5. Hệ mã El Gamal (1985)

- Là một biến thể của sơ đồ Diffie – Hellman.
- Tính an toàn dựa trên tính khó giải của bài toán logarit rời rạc.
- Nhược điểm chính: kích thước thông tin sau khi mã hóa sẽ tăng gấp đôi so với thông tin gốc.
- Giống các hệ mã khóa công khai khác , El Gamal làm việc với tốc độ thấp (việc với các số nguyên lớn),
- Cần bộ nhớ lớn dành cho việc lưu trữ các khóa .
- Với hệ mã El Gamal chúng ta cần gấp đôi bộ nhớ để chứa bản mã so với các hệ mã khác

4.3.5.1. Mã hóa

- Chọn pϵ Z_p và α <P (α là một phần tử nguyên thủy ϵ Z^*_p) và x ϵ Z_N (x là của người nhận, bí mật) , tính:

  $y = \alpha^x \bmod p$

- Thông điệp rõ M (M ϵ Z_P)
- Chọn ngẫu nhiên k < p và tính khóa mã hóa K:

  $K = y^k \bmod p$

- Sau đó tính cặp bản mã:

  $C1 = \alpha^k \bmod p$

  $C2 = K.M \bmod p$

- Gửi bản mã C = (C1, C2) đi (chú ý là sau đó k sẽ bị huỷ).

4.3.5.2.Giải mã

- Để giải mã thông điệp đầu tiên ta cần tính lại khóa mã hóa thông điệp K:

  $K = C_1^x \bmod p = \alpha^{k.x} \bmod p$

- Sau đó tính M bằng cách giải phương trình :

  $M = C_2 \, . \, K^{-1} \bmod p$

- Việc giải mã bao gồm việc tính lại khóa tạm thời K (rất giống với mô hình của Diffie – Hellman). Khóa công khai của hệ mã là (p, α, y), khóa bí mật là x.

4.3.5.4. Ví dụ El Gamal

- Cho hệ mã El Gamal có P = 97,α= 5, x = 58.

  Tìm khóa của hệ mã trên.

  Mã hóa bản rõ M = 3 với k được chọn bằng 36.
- Tính $y = 5^{58} \bmod 97 = 44$, từ đó suy ra $K_P = (P, \alpha, y) = (97, 5, 44)$ và $K_S = (58)$.
- Để mã hóa thông điệp M = 3 ta tính khóa $K = 44^{36} \bmod 97 = 75$ sau đó tính:

  $C_1 = 5^{36} = 50 \bmod 97 = 50$

  $C_2 = 75.3 \bmod 97 = (75 \bmod 97 * 3 \bmod 97) \bmod 97 = 31$
- Vậy bản mã thu được là C = (50, 31).

4.3. Public Key Infrastructure (PKI)

4.3.1. Khái niệm

- *Public Key Infrastructure (PKI)* cung cấp giải pháp tổng thể bảo vệ thông điệp và hiện thực những nội dung đã thảo luận trên đây.Sự cần thiết một hệ thống tổng hợp hỗ trợ cho e-commerce, giao dịch an toàn và bảo mật thông tin là những lợi ích do PKI mạng lại.

- Mục đích : PKI thiết lập một hạ tầng thông tin an toàn thông suốt cho mọi nhà cung cấp,các hệ thống và mạng . PKI là một môi trường làm việc chứ không phải là một công nghệ đặc biệt. Phát triển PKI độc lập với việc phát triển phần mềm và các ứng dụng khác

Khái niệm (tiếp)

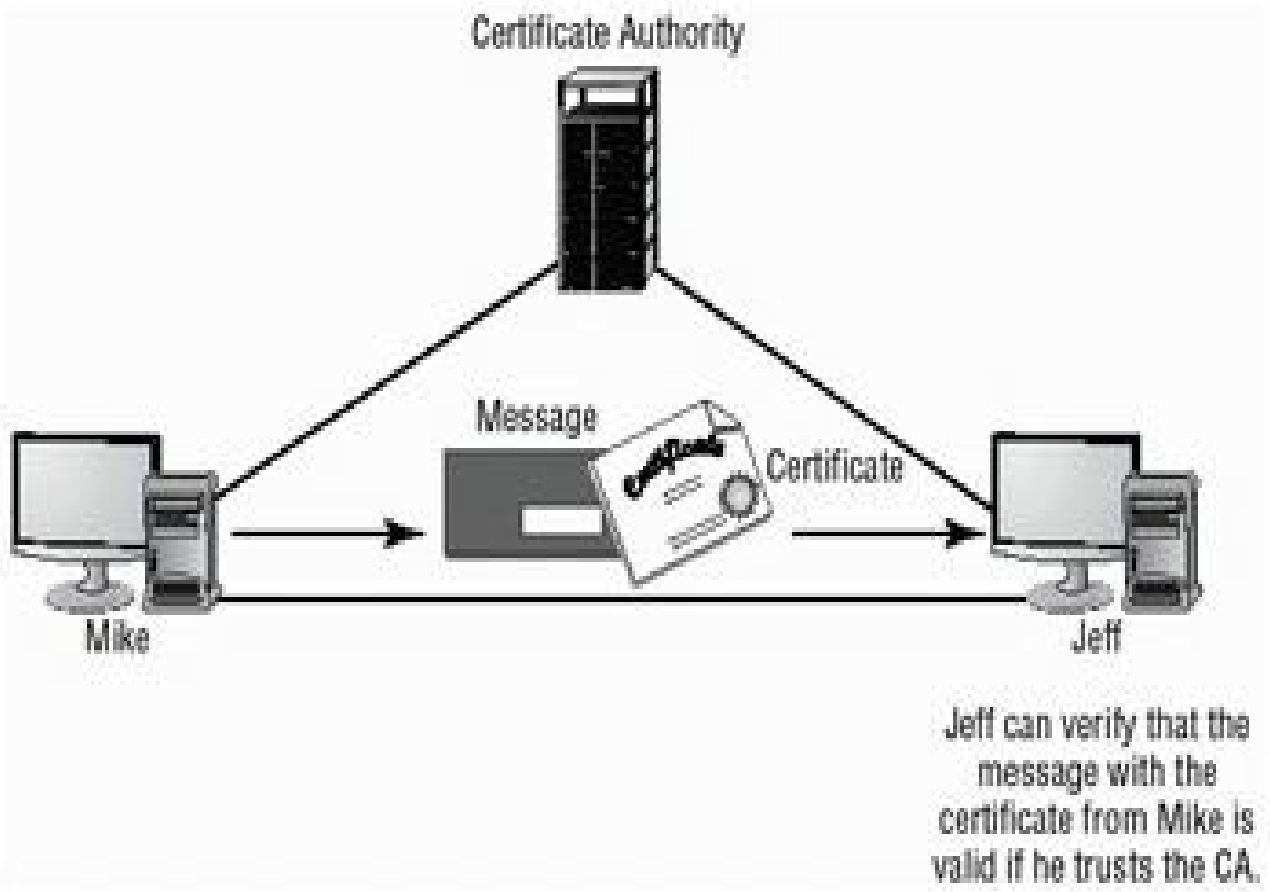
- PKI là hệ thống khoá kép (two-key system). Thông điệp được mã bởi “public key” và giải mật bằng “ private key”.Nếu muốn mã và gửi thông điệp mã cho ai đó , ta phải yêu cầu người đó gửi “ public key “ của họ và ta dùng “ public key” để mã và gửi thông điệp đã mã đi . Bên nhận sẽ dùng khoá “ private key “ của mình để giải mật.

4.3.2 . CA – Certificate Authorities

- ✓ Tổ chức thứ ba- Certificate Authorities (CA) quản lý " public keys" và cấp phát giấy chứng nhận (cirtificates) để kiểm tra tính hợp lệ của thông điệp từ bên phát đến.
- ✓ CA là một phần của PKI (Public Key Infrastructures) .

Certificate Authorities

- *Certificate authority (CA)* Tổ chức có quyền cấp chứng thực (*certificates)*

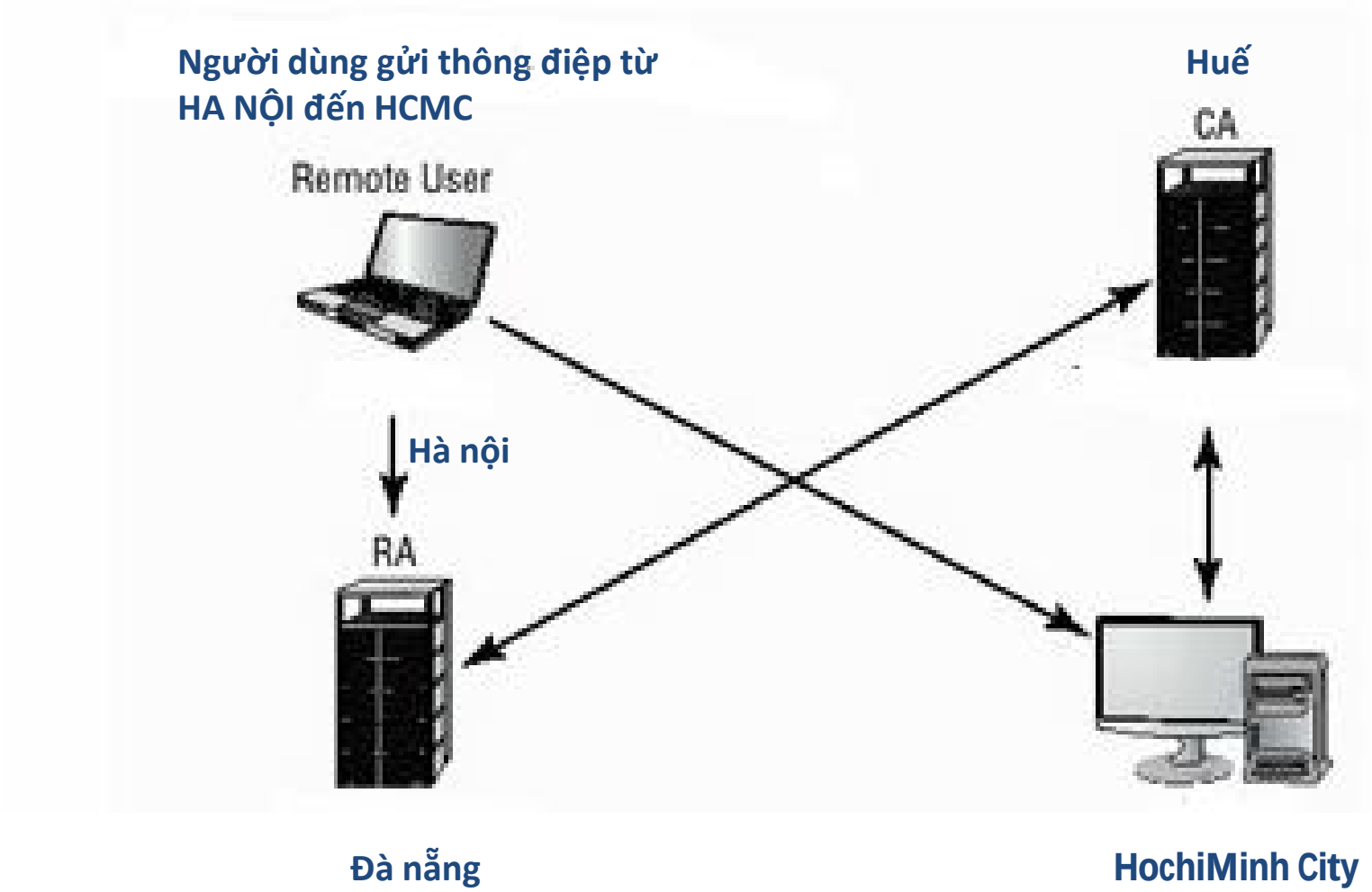


4.3.3. RAs and LRAs

4.3.3.1. Registration authority (RA)

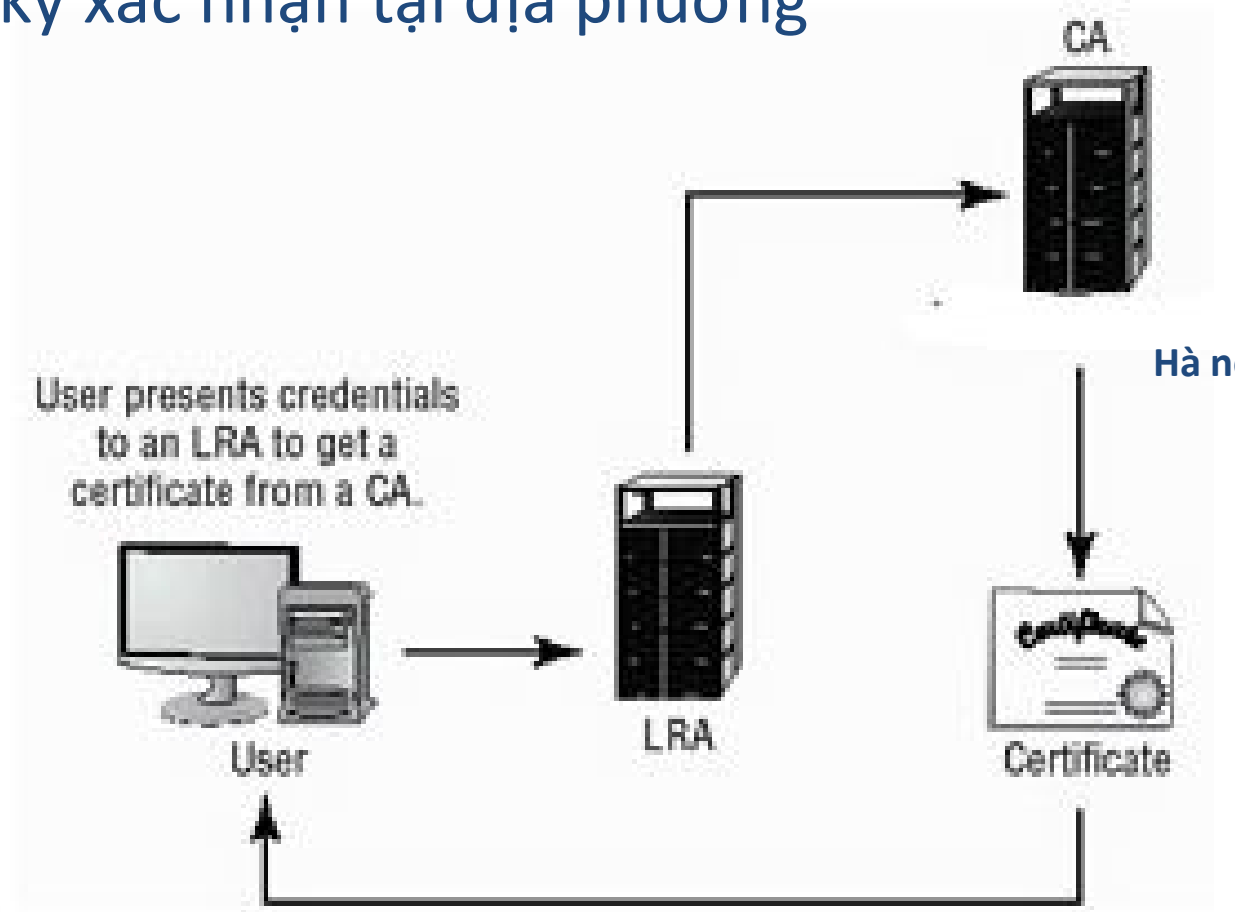
Chia sẻ bớt một phần công việc của CA. Hệ thống RA làm việc như một người trung gian trong quá trình. RA phân phối khoá , chấp nhận đăng ký cho CA và xác minh định danh. RA không cấp chứng chỉ.Đây là trách nhiệm của CA.

Registration authority (RA)



4.3.3.2. Local registration authority (LRA)

Đăng ký xác nhận tại địa phương



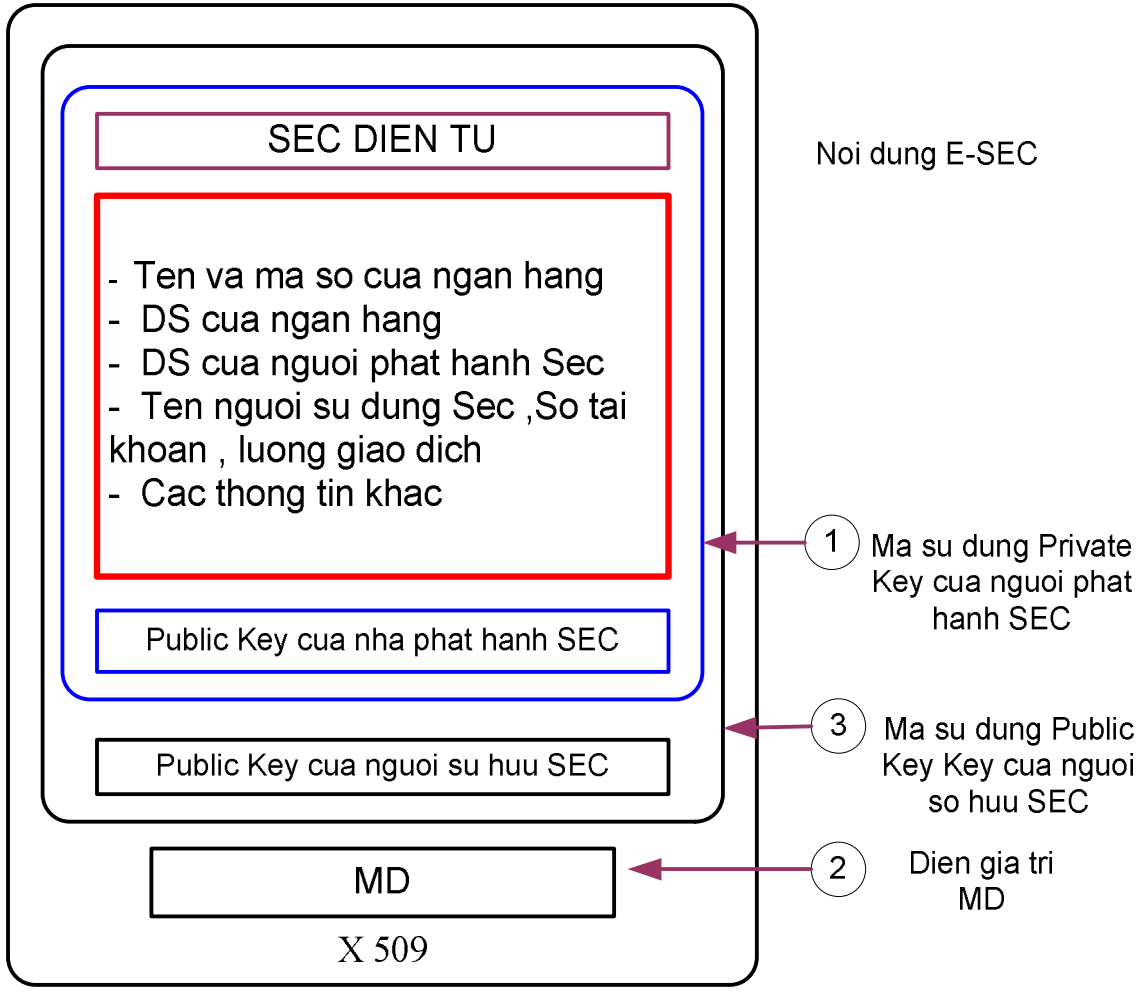
4.3.4. Certificates (Chứng chỉ)

4.3.4.1 Nội dung của chứng chỉ - chuẩn x-509

Version	V3
Serial Number	1234 D123 4567
Signature Algorithm	Md2RSA
Issuer	Sample Certificate
Valid from:	Sunday, September 8, 2005
Valid to:	Sunday, September 15, 2005
Subject	Mr. Your Name Here, Myco
Public Key	Encrypted Value of Key
Extensions	Subject Type = End Entity
Signature Algorithm Signature	sha1 Encrypted Data

Digital Signature Area

Fields of a Simple X.509 Certificate



Sử dụng khóa công khai để chứng thực séc điện tử theo chuẩn X 509

4.3.4.2 Certificate Policies

- ✓ *Certificate policies* xác định "chứng chỉ" được dùng để làm gì?
- ✓ *Certificate policies* quy định các chứng chỉ được cấp và sử dụng như thế nào.
- ✓ CA cần có policy để phân chia trách nhiệm hoặc xác nhận các điểm CA khác .

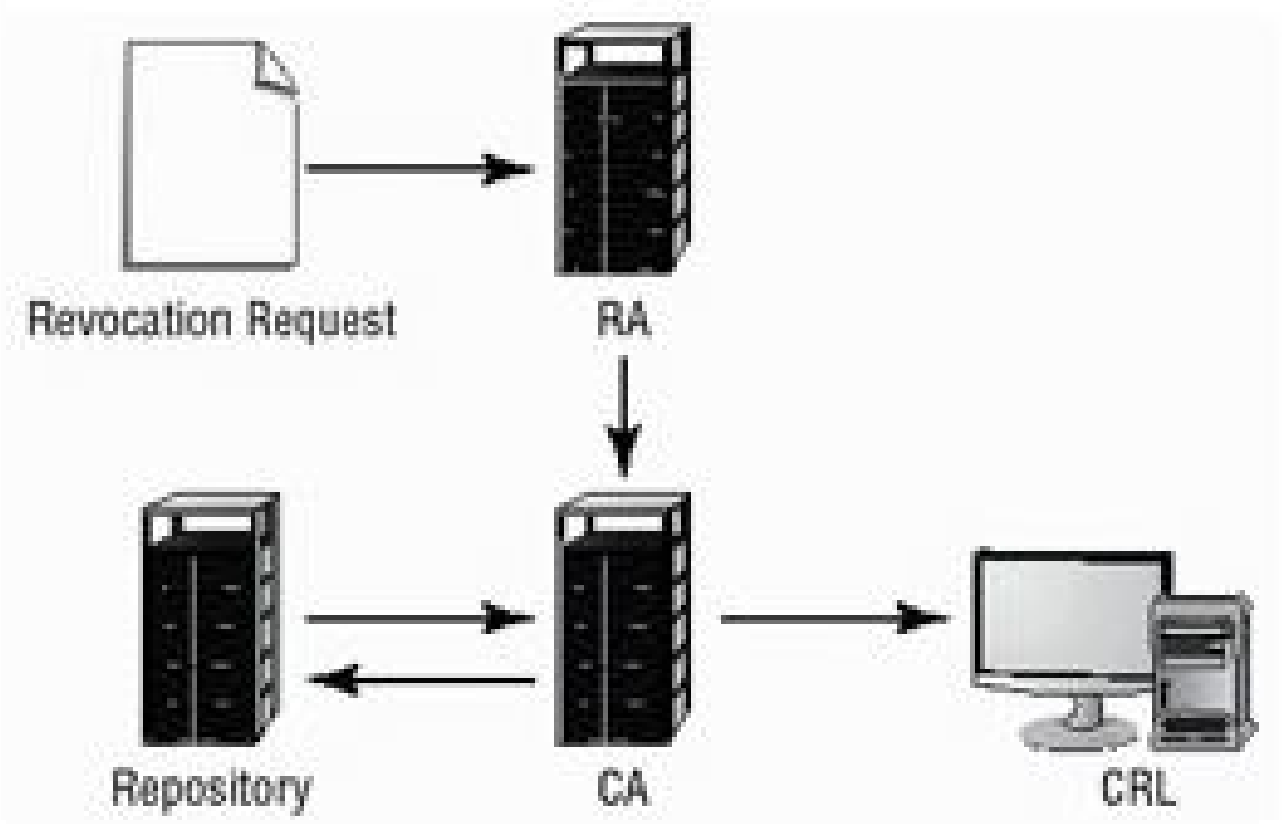
4.3.4.3 Certificate Practice Statements

- ✓ *Certificate practice statement (CPS)* : CA cấp phát chứng chỉ và thực thi các chính sách của mình. Đây là những tài liệu chi tiết giúp cho các chính sách của CA có hiệu lực.

4.3.4.4 Thu hồi / huỷ chứng chỉ (Certificate Revocation)

- *Certificate revocation* là qua trình thu hồi chứng chỉ trước khi nó hết hiệu lực , hết hạn.
- Thực hiện nhờ *"Certificate revocation list" – danh sách thu hồi chứng chỉ (CRL)* hoặc sử dụng giao thức *"online certificate status protocol" (OCSP).*

Thu hồi / huỷ chứng chỉ (Certificate Revocation)



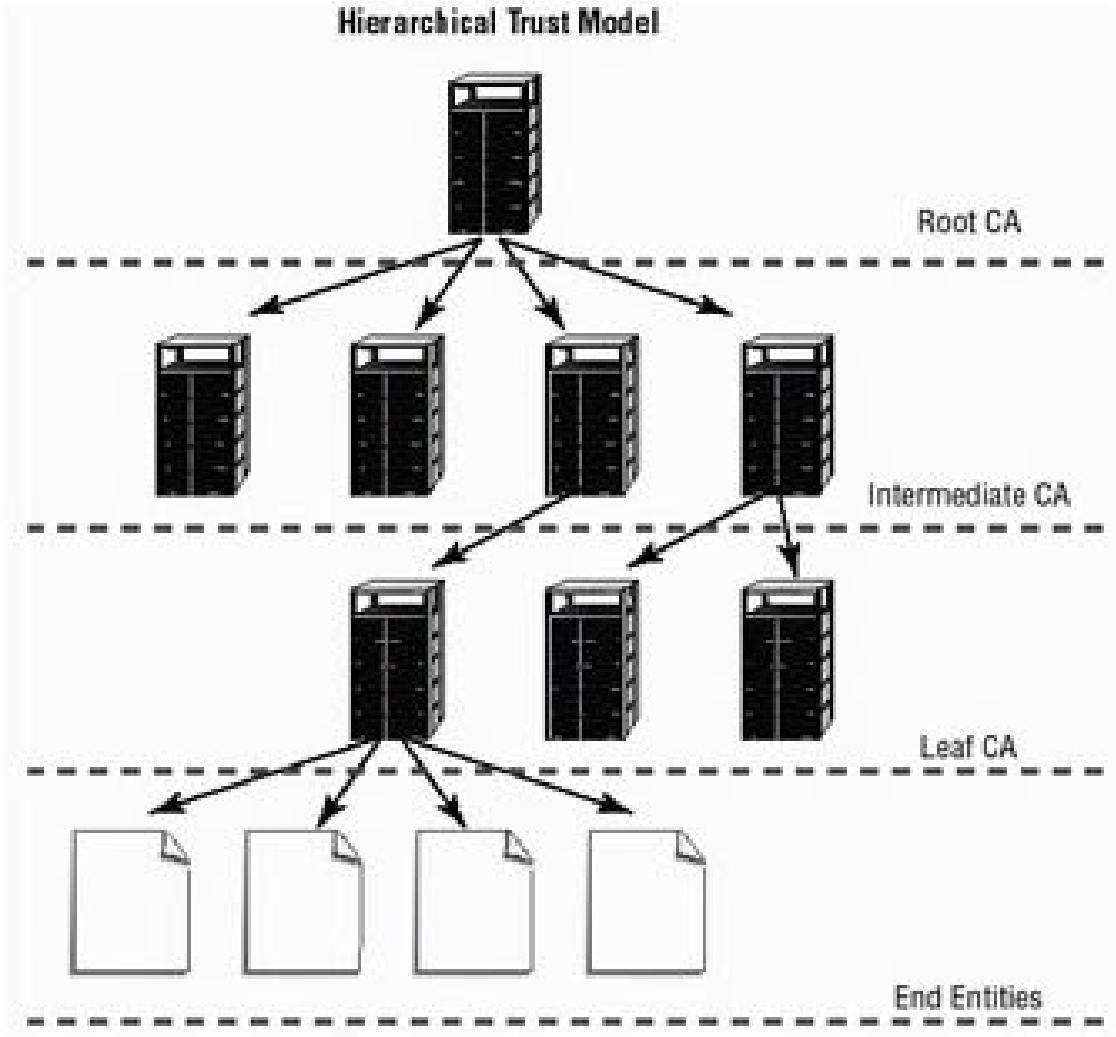
Yêu cầu thu hồi chứng chỉ

4.3.4.5. Mô hình uỷ quyền (Trust Models)

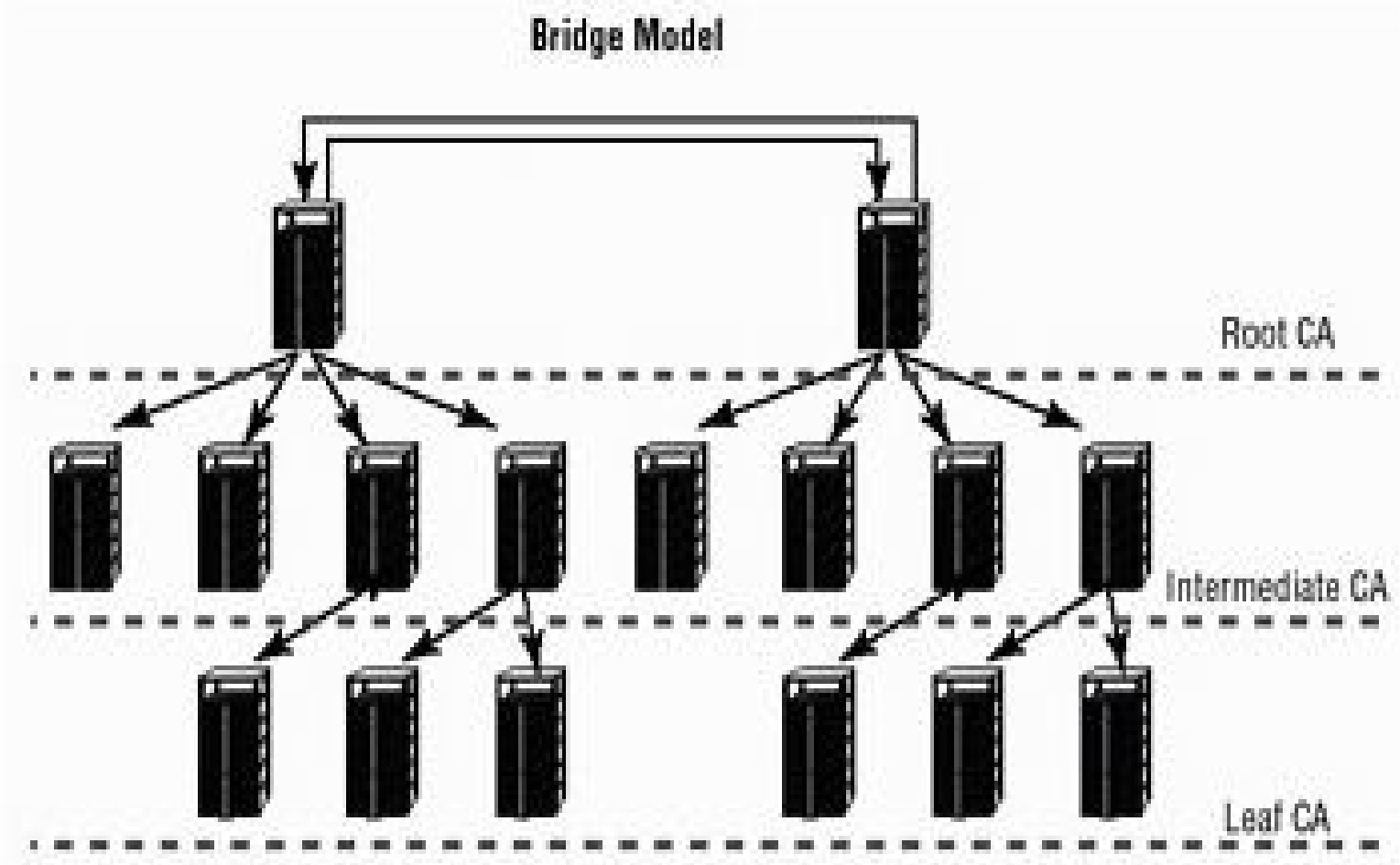
Có bốn mô hình uỷ quyền trên PKI

- Hierarchical
- Bridge
- Mesh
- Hybrid

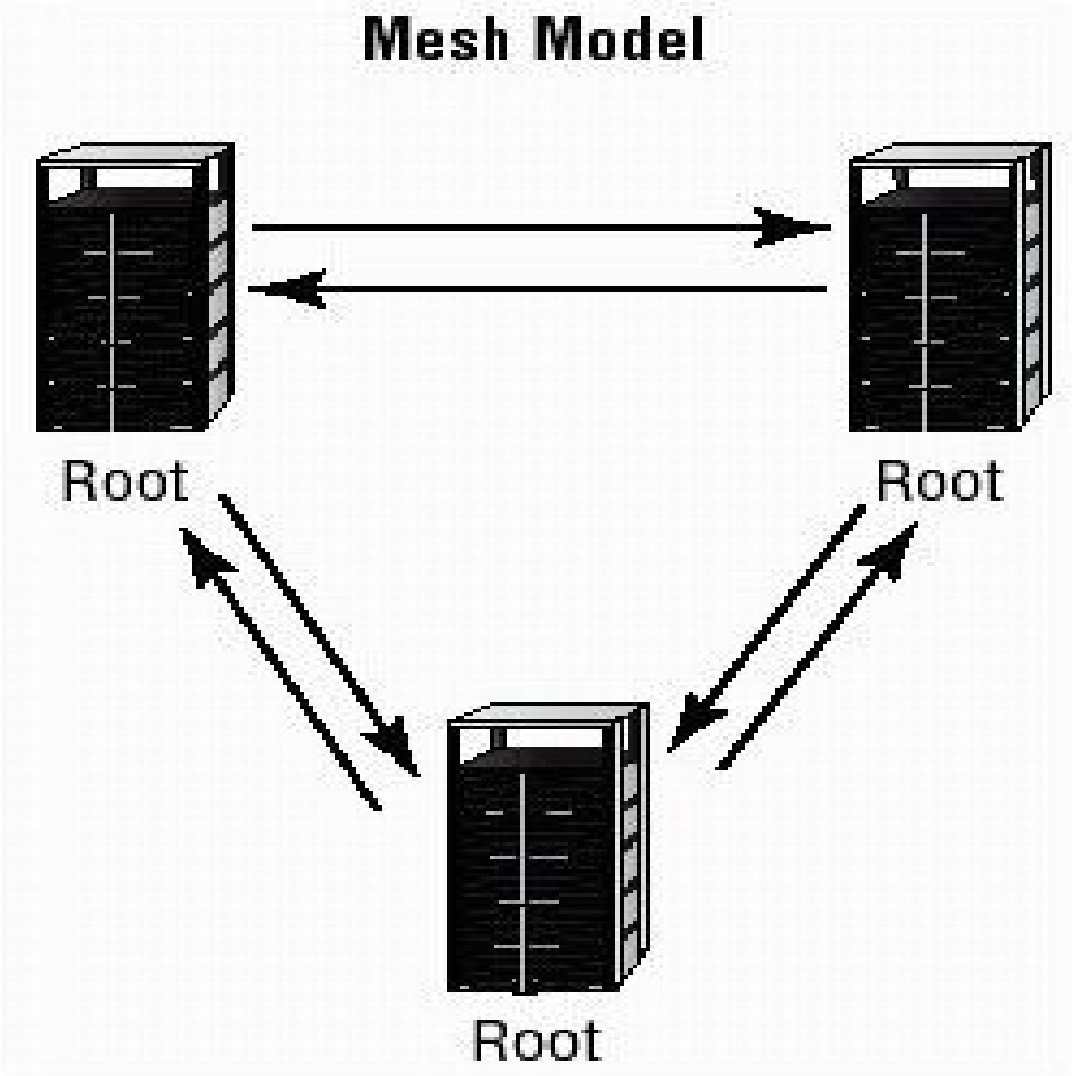
1. Hierarchical



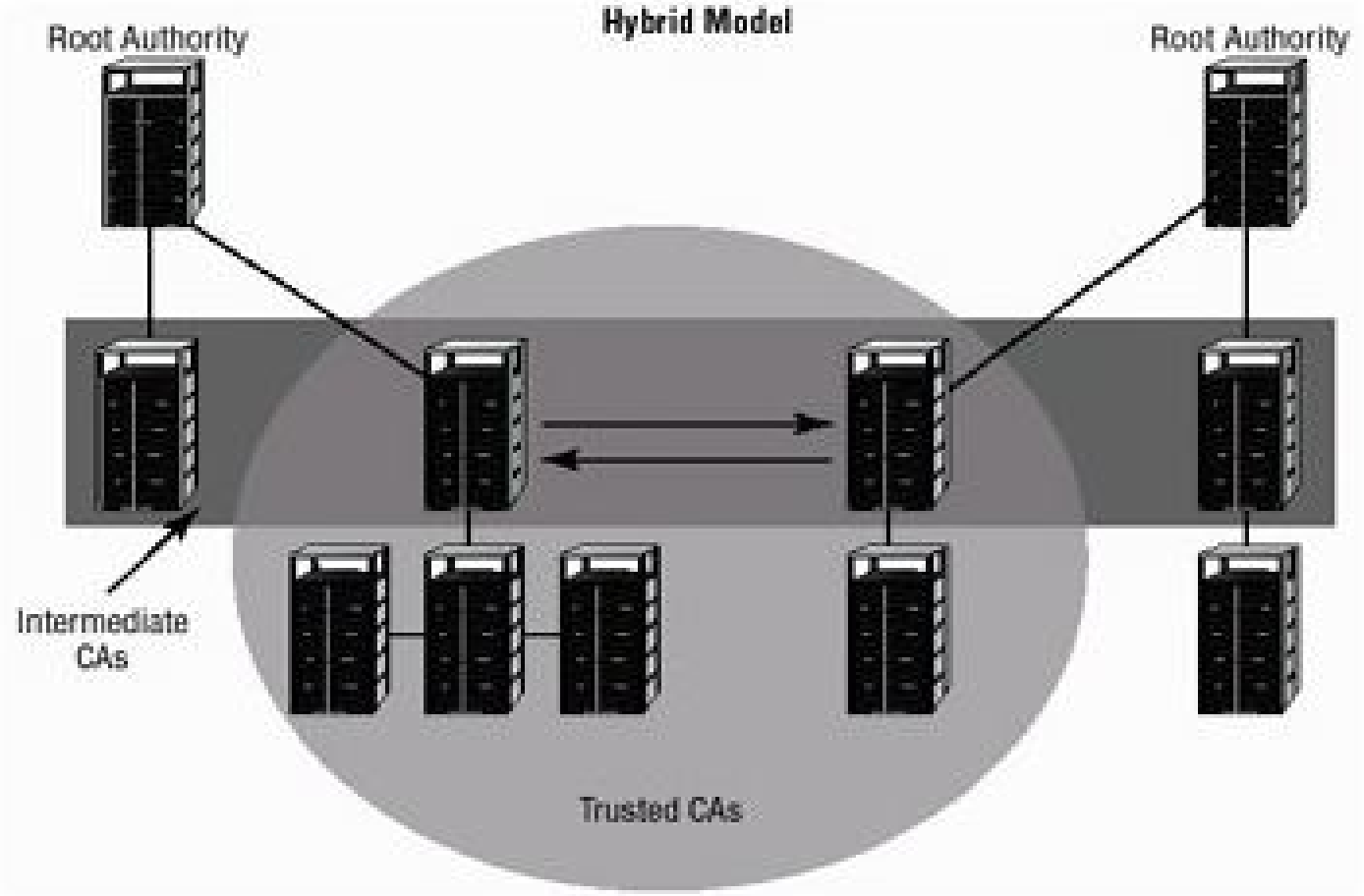
2. Bridge



3. Mesh



4. Hybrit


Hybrid Model
Root Authority
Root Authority
Intermediate CAs
Trusted CAs

Hết chương 4