TRUYỀN VÀ BẢO MẬT THÔNG TIN

Bài 7:

Mã hóa khoa công khai

VŨ THỊ TRÀ

Nội dung

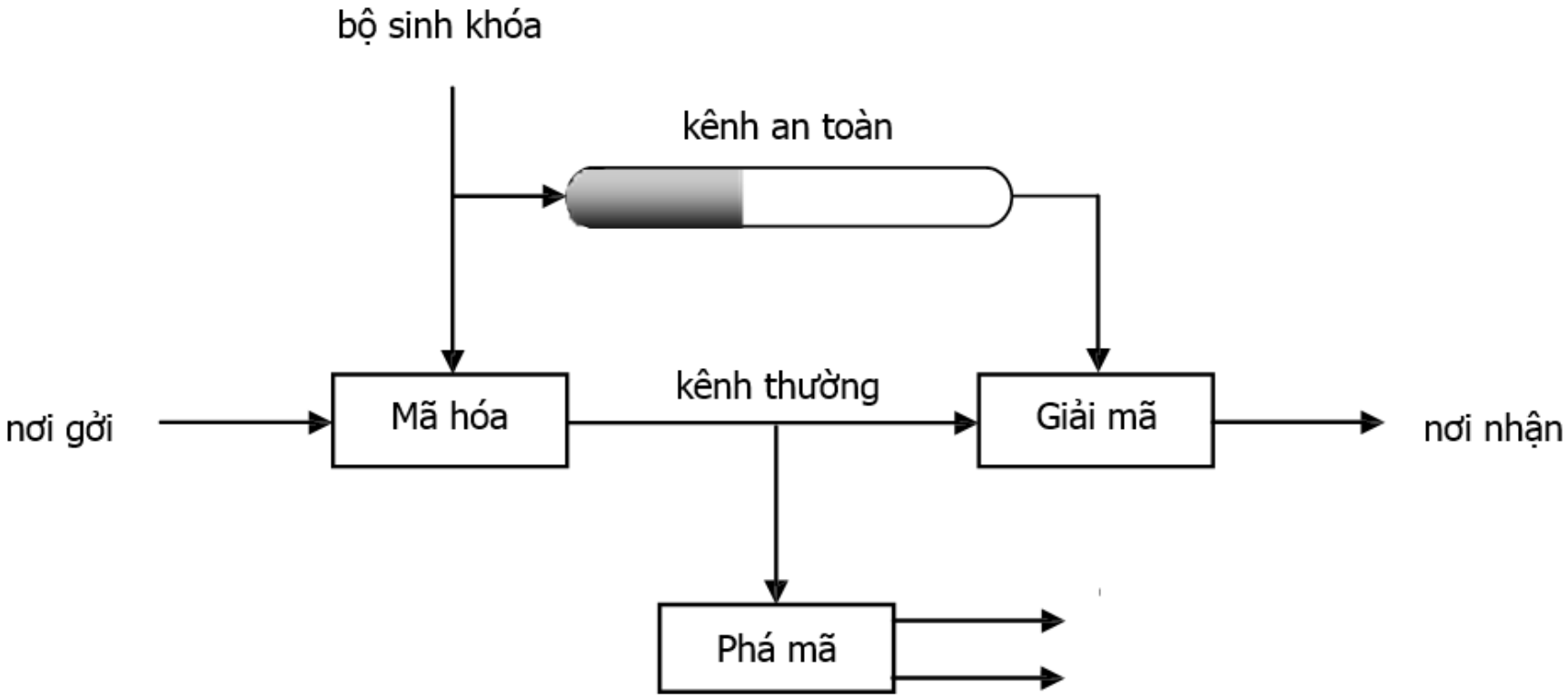
- Lịch sử ra đời mã hóa khóa công khai
- Lý thuyết số
- Mã hóa RSA
- Mô hình bảo mật, chứng thực, không từ chối
- Trao đổi khóa công khai

Lịch sử ra đời mã hóa khóa công khai

Lịch sử ra đời mã hóa khóa công khai

- Mã hóa đối xứng dù rằng đã phát triển từ cổ điển đến hiện đại, vẫn tồn tại:
  - **Vấn đề trao đổi khóa giữa người gửi và người nhận**: Cần phải có một kênh an toàn để trao đổi khóa sao cho khóa phải được giữ bí mật chỉ có người gửi và người nhận biết. Điều này tỏ ra không hợp lý khi mà ngày nay, khối lượng thông tin luân chuyển trên khắp thế giới là rất lớn. Việc thiết lập một kênh an toàn như vậy sẽ tốn kém về mặt chi phí và chậm trễ về mặt thời gian.
  - **Tính bí mật của khóa**: không có cơ sở quy trách nhiệm nếu khóa bị tiết lộ.
- Vào năm 1976 Whitfield Diffie và Martin Hellman đã tìm ra một phương pháp mã hóa khác mà có thể giải quyết được hai vấn đề trên, đó là mã hóa khóa công khai (public key cryptography) hay còn gọi là mã hóa bất đối xứng (asymetric cryptography). Đây có thể xem là một bước đột phá quan trọng nhất trong lĩnh vực mã hóa.

Mô hình mã hóa đối xứng



Lịch sử ra đời mã hóa khóa công khai

- Để khắc phục điểm yếu của mã hóa đối xứng người ta tập trung vào nghiên cứu theo hướng: có phương pháp nào để việc *mã hóa và giải mã dùng hai khóa khác nhau*? Khi đó, $C = E(P, K_1)$ và $P = D(C, K_2)$, chúng ta sẽ có 2 phương án:
  - **PA1:** người nhận giữ bí mật khóa K2, còn khóa K1 thì công khai cho tất cả mọi người biết.
    - ✓ *Đảm bảo tính bảo mật*
    - ✓ *Không đảm bảo tính chứng thực* và *tính không từ chối*
  - **PA2:** người gửi giữ bí mật khóa K1, còn khóa K2 thì công khai cho tất cả mọi người biết.
    - ✓ *Không đảm bảo tính bảo mật*
    - ✓ *Đảm bảo tính chứng thực* và *tính không từ chối*

Lịch sử ra đời mã hóa khóa công khai

- Trong cả hai phương án, một khóa được giữ bí mật chỉ một người biết, còn khóa kia được công khai. Do đó mô hình mã hóa trên được gọi là **mã hóa khóa công khai** (hay **mã hóa bất đối xứng**)
- Nếu kết hợp PA1 và PA2, thì mô hình đề xuất của chúng ta khắc phục được các nhược điểm của mã hóa đối xứng.

Qui ước trong hệ mã hóa khóa công khai

- Để tránh nhầm lẫn với khóa bí mật của các mã đối xứng, khóa bí mật trong mô hình trên được gọi là khóa riêng (private key) và ký hiệu là K_R.
- Khóa công khai (public key) được ký hiệu là K_U.
- Bản rõ được ký hiệu là M, còn bản mã giữ nguyên ký hiệu là C
- Khi đó,
  - **PA1:** $C = E(M, K_U); M = D(C, K_R)$
  - **PA2:** $C = E(M, K_R); M = D(C, K_U)$

Mối liên hệ giữa K_U & K_R

1. $K_U = f(K_R)$ phải bất khả thi về mặt thời gian
2. Nếu **(1)** bị vi phạm thì việc giữ bí mật khóa K_R không còn ý nghĩa vì từ khóa công khai K_U có thể tính được K_R

→ *Sử dùng các hàm một chiều (oneway function) vì hàm nghịch đảo của chúng rất khó thực hiện.*

Ví dụ: việc sinh ra hai số nguyên tố lớn p, q và tính tích N = pq thì thực hiện dễ dàng. Tuy nhiên nếu chỉ cho trước N và thực hiện phân tích N để tìm lại hai số nguyên tố p, q là việc có thể bất khả thi về mặt thời gian. → xem cách thức áp dụng hàm một chiều này để tạo khóa K_R và K_U trong phần mã hóa **RSA**

Lý thuyết số

Lý thuyết số

- Một số khái niệm
- Định lý Fermat
- Phép logarit rời rạc

Phép chia modulo

$$a \ mod \ n = r \quad với \quad a \geq 0; \quad n > 0; \quad 0 \leq r \leq n - 1$$

- ✓ Phép chia modulo phân hoạch tập số tự nhiên N thành n lớp tương đương đồng dư với r = {0, 1, …, n-1}

Một số tính chất của phép chia modulo

1) $(a+b) \bmod n = [(a \bmod n) + (b \bmod n)] \bmod n$
2) $(a-b) \bmod n = [(a \bmod n) - (b \bmod n)] \bmod n$
3) $(axb) \bmod n = [(a \bmod n) \times (b \bmod n)] \bmod n$

Một số khái niệm khác

- **Ước số**
  - Nếu $a \bmod n = 0$ (hay $a \equiv 0 \bmod n$), thì n là ước số của a
  - Ước chung lớn nhất (USCLN) của 2 số: $gcd(a,b)$
- **Số nguyên tố**
  - p được gọi là số nguyên tố nếu p chỉ chia hết cho 1 và chính nó.
- **Số nguyên tố cùng nhau**
  - Hai số nguyên a,b được gọi là nguyên tố cùng nhau nếu $gcd(a,b)=1$. Ký hiệu $a \perp b$

Phần tử nghịch đảo của phép chia modulo

- Nếu hai số nguyên *a* và *n* là nguyên tố cùng nhau, thì tồn tại *w* sao cho:

$$a.w \equiv 1 \ mod \ n$$

Khi đó, *w* là phần tử nghịch đảo của *a* trong phép module cho *n*, kí hiệu a^{-1}

Định lý Fermat

- Nếu *p* là số nguyên tố và *a* là số nguyên không chia hết cho *p* thì $a^{p-1} \equiv 1 \ mod \ p$

Phép logarit rời rạc

- Gọi *y* là giá trị nguyên trong phép lũy thừa modulo với *n*

$$y = a^x \bmod n$$

với *a, x, n, y* là các số nguyên

- Nếu biết *y, a, n,* muốn tìm lại *x* ta dùng hàm logarith (hay còn gọi logarith rời rạc)

$$x = dlog_{a,n}\, y$$

→ Việc tính logarith được minh chứng là tốn kém về thời gian, được xem là bất khả thi với *a* và *n* là các số lớn

→ Phép lũy thừa modulo được xem là *hàm một chiều*

Mã hóa RSA

Mã hóa RSA

- 1977 tại học viên MIT, nhóm các tác giả Ron Rivest, Adi Shamir và Len Adleman đã xây dựng RSA – pp mã hóa khóa công khai theo khối. Trong đó M và C là các số nguyên trong khoảng 0 đến 2^i với i là số bít của khối.
- Kích thước thường dùng của i là 1024 bít
- RSA sử dụng hàm một chiều phân tích một số thành thừa số nguyên tố

Nguyên tắc thực hiện

RSA dùng phép lũy thừa modulo

1. Chọn hai số nguyên tố lớn p và q và tính $N = pq$. Cần chọn p và q sao cho: $M < 2^{i-1} < N < 2^i$. Với $i = 1024$ thì N là một số nguyên dài khoảng 309 chữ số.
2. Tính $n = (p-1)(q-1)$
3. Tìm một số e sao cho e nguyên tố cùng nhau với n
4. Tìm một số d sao cho: $ed \equiv 1 \bmod n$ (d là nghịch đảo của e trong phép modulo n)
5. Hủy bỏ n, p và q. Chọn khóa công khai K_U là cặp (e, N), khóa riêng K_R là cặp (d, N)

Nguyên tắc thực hiện

6. Việc mã hóa thực hiện theo công thức:
   - PA 1, mã hóa bảo mật: $C = E(M, K_U) = M^e \bmod N$
   - PA 2, mã hóa chứng thực: $C = E(M, K_R) = M^d \bmod N$
7. Việc giải mã thực hiện theo công thức:
   - PA 1, mã hóa bảo mật: $M = D(C, K_R) = C^d \bmod N$
   - PA 2, mã hóa chứng thực: $M = D(C, K_U) = C^e \bmod N$

Ví dụ RSA

Mã hóa RSA với kích thước khóa 6 bít

1. Chọn p = 11 và q = 3, do đó N = pq = 33 ($2^5 = 32 < 33 < 64 = 2^6$)
2. n = (p-1)(q-1) = 20
3. Chọn e = 3 nguyên tố cùng nhau với n
4. Tính nghịch đảo của e trong phép modulo n được d = 7 (3x7 = 21)
5. Khóa công khai K_U= (e, N) = (3, 33). Khóa bí mật K_R= (d, N)= (7, 33)

Ví dụ RSA

PA 1

6. Mã hóa bản rõ M = 15:

$C = E(M, K_U) = M^e \bmod N = 15^3 \bmod 33 = 9$

(vì $15^3 = 3375 = 102 \times 33 + 9$)

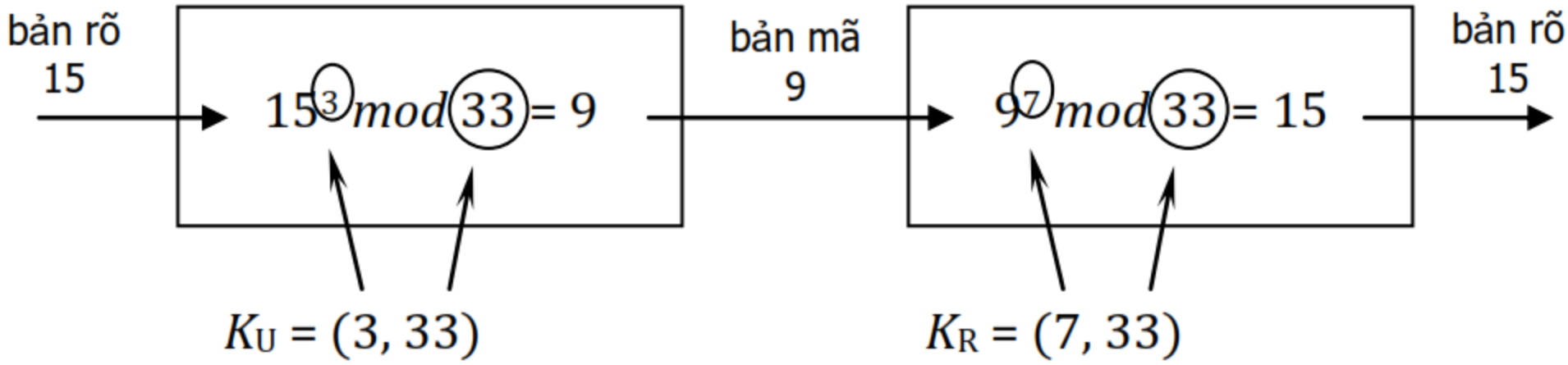
7. Giải mã bản mã C = 9:

$M = D(C, K_R) = C^d \bmod N = 9^7 \bmod 33 = 15$

(vì $9^7 = 4\,782\,696 = 144\,938 \times 33 + 15$)

Ví dụ RSA

PA 1



Ví dụ RSA

PA 2

6. Mã hóa bản rõ M = 15:

$C = E(M, K_R) = M^d \bmod N = 15^7 \bmod 33 = 27$

(vì $15^7 = 170\,859\,375 = 5\,177\,556 \times 33 + 27$)

7. Giải mã bản mã C = 27:

$M = D(C, K_U) = C^e \bmod N = 27^3 \bmod 33 = 15$

(vì $27^3 = 19\,683 = 596 \times 33 + 15$)

Phép sinh số

- Thuật toán Miller-Rabin: kiểm tra số nguyên tố
- Thuật toán Euclid mở rộng: kiểm tra tính nguyên tố cùng nhau của e và n

  → *(Phụ lục 2)*

Độ an toàn của RSA

- Năm 1977, các tác giả của RSA đã treo giải thưởng cho ai phá được RSA có kích thước của N vào khoảng 428 bít, tức 129 chữ số. Các tác giả này ước đoán phải mất 40 nghìn triệu triệu năm mới có thể giải được.
- Năm 1994, câu đố này đã được giải chỉ trong vòng 8 tháng

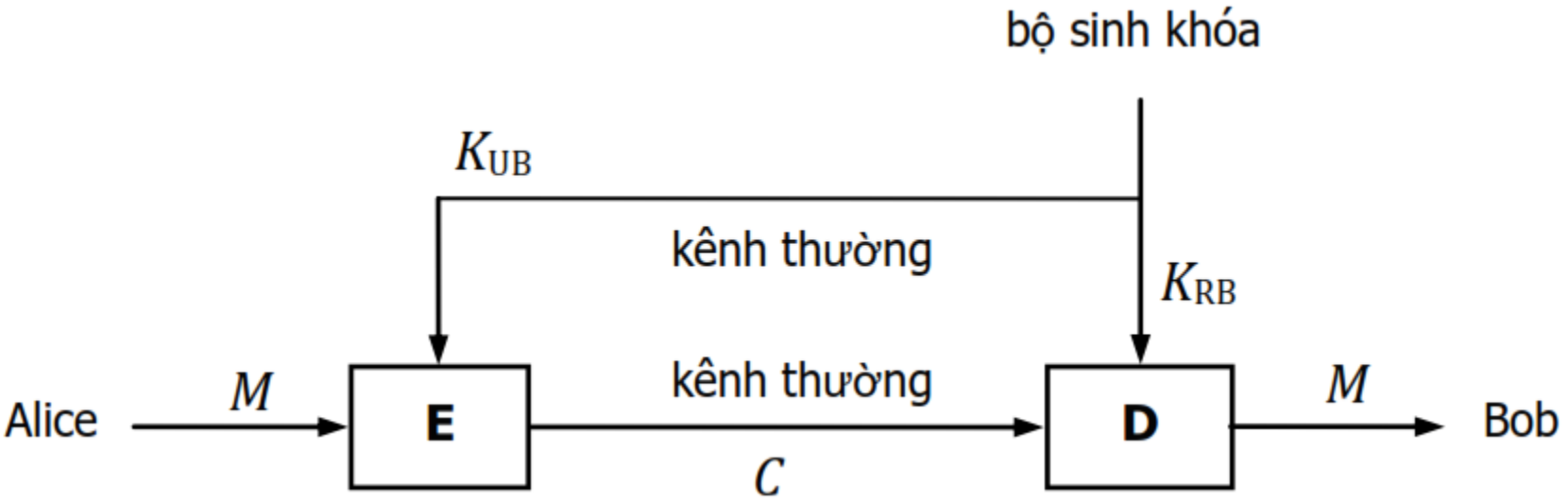
Bảng liệt kê các mốc phá mã RSA

Số chữ số của N	*Số bít*	*Năm phá mã*	*Thuật toán*
100	322	1991	Quadratic sieve
110	365	1992	Quadratic sieve
120	398	1993	Quadratic sieve
129	428	1994	Quadratic sieve
130	431	1996	GNFS
140	465	1999	GNFS
155	512	1999	GNFS
160	530	2003	Lattice sieve
174	576	2003	Lattice sieve
200	633	2005	Lattice sieve

Bảo mật, chứng thực, không từ chối

Mô hình bảo mật với mã hóa khóa công khai

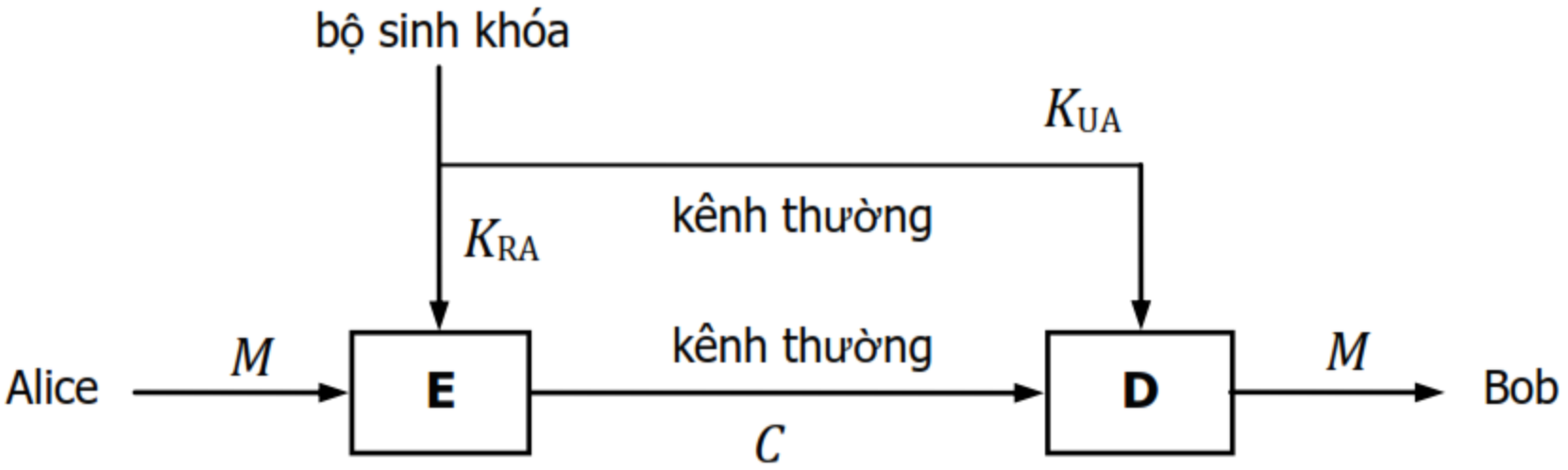
$$C = E(M, K_{UB})$$
$$M = D(C, K_{RB})$$



Mô hình không thoái thác với mã hóa khóa công khai

$$C = E(M, K_{RA})$$

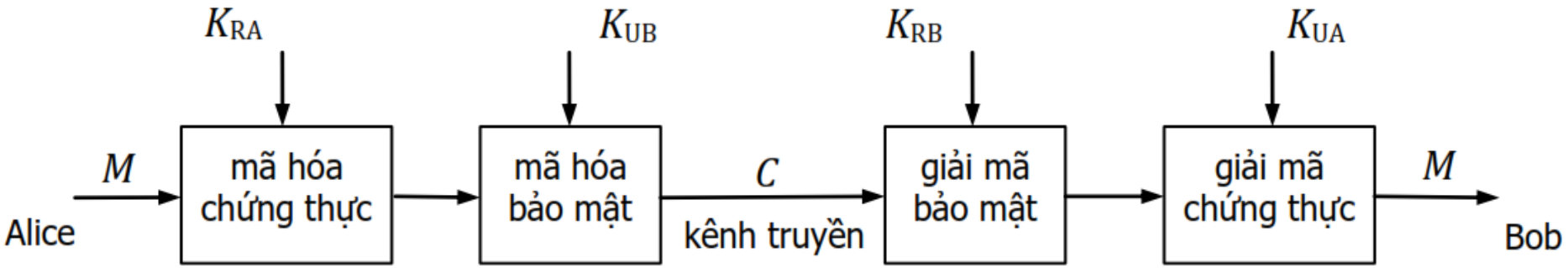
$$M = D(C, K_{UA})$$



Mô hình kết hợp bảo mật, chứng thực và không từ chối

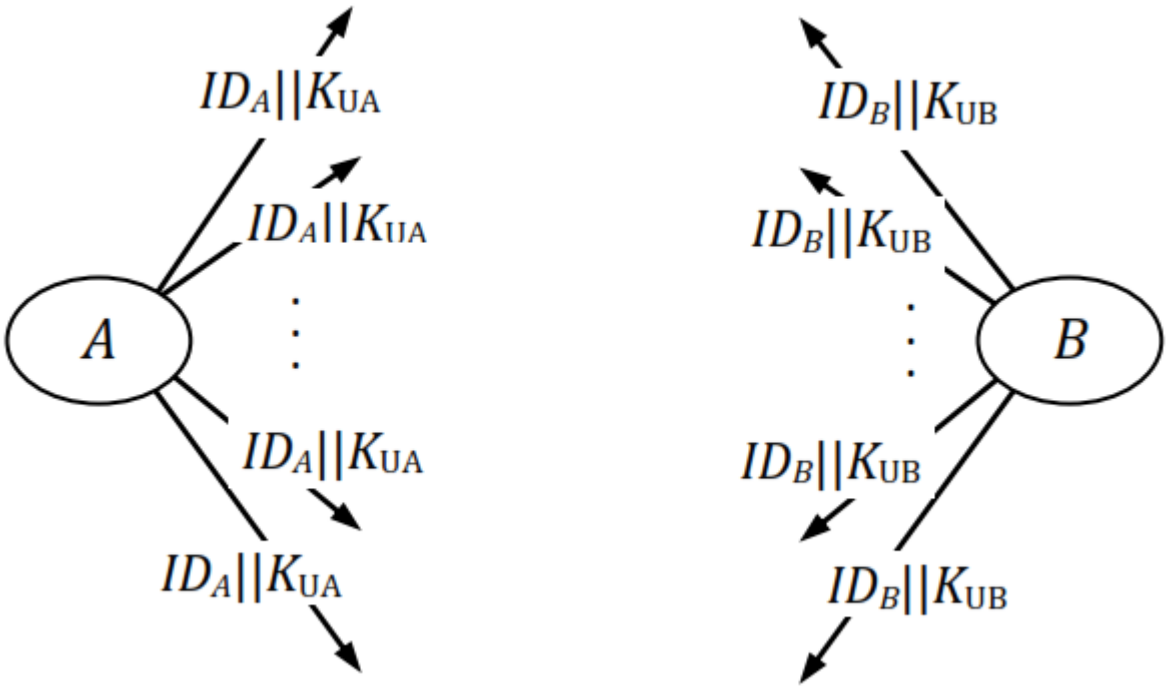
$$C = E(E(M, K_{RA}), K_{UB})$$

$$M = D(D(C, K_{RB}), K_{UA})$$



Trao đổi khóa công khai

Trao đổi khóa công khai tự phát



Trao đổi khóa công khai

- Tuy nhiên ở đây chúng ta lại gặp phải vấn đề về chứng thực. Làm như thế nào mà Alice có thể đảm bảo rằng K_{UB} chính là khóa công khai của Bob? Trudy có thể mạo danh.
- Bob bằng cách lấy khóa K_{UT} của Trudy và nói rằng đó là khóa công khai của Bob. Vì vậy, việc trao đổi khóa công khai theo mô hình trên đặt gánh nặng lên vai của từng cá nhân. Alice muốn gửi thông điệp cho Bob hay bất cứ người nào khác thì phải tin tưởng vào khóa công khai của Bob hay của người đó. Tương tự như vậy cho Bob.

Trao đổi khóa công khai

- Để giảm gánh nặng cho từng cá nhân, một mô hình gọi là „chứng chỉ khóa công khai“ (public-key certificate) được sử dụng. Trong mô hình này có một tổ chức làm nhiệm vụ cấp chứng chỉ được gọi là trung tâm chứng thực (Certificate Authority – CA). Các bước thực hiện cấp chứng chỉ cho Alice như sau:

  1) Alice gửi định danh ID_A và khóa công khai K_{UA} của mình đến trung tâm chứng thực.

  2) Trung tâm chứng nhận kiểm tra tính hợp lệ của Alice, ví dụ nếu ID_A là „Microsoft“, thì Alice phải có bằng chứng chứng tỏ mình thực sự là công ty Microsoft.

Trao đổi khóa công khai

- Để giảm …

  3) Dựa trên cơ sở đó, trung tâm chứng thực cấp một chứng chỉ C_A để xác nhận rằng khóa công khai K_{UA} đó là tương ứng với ID_A. Chứng chỉ được ký chứng thực bằng khóa riêng của trung tâm để đảm bảo rằng nội dung của chứng chỉ là do trung tâm ban hành.

$$C_A = E(ID_A || K_{UA}, K_{RAuth})$$

(|| là phép nối dãy bít)

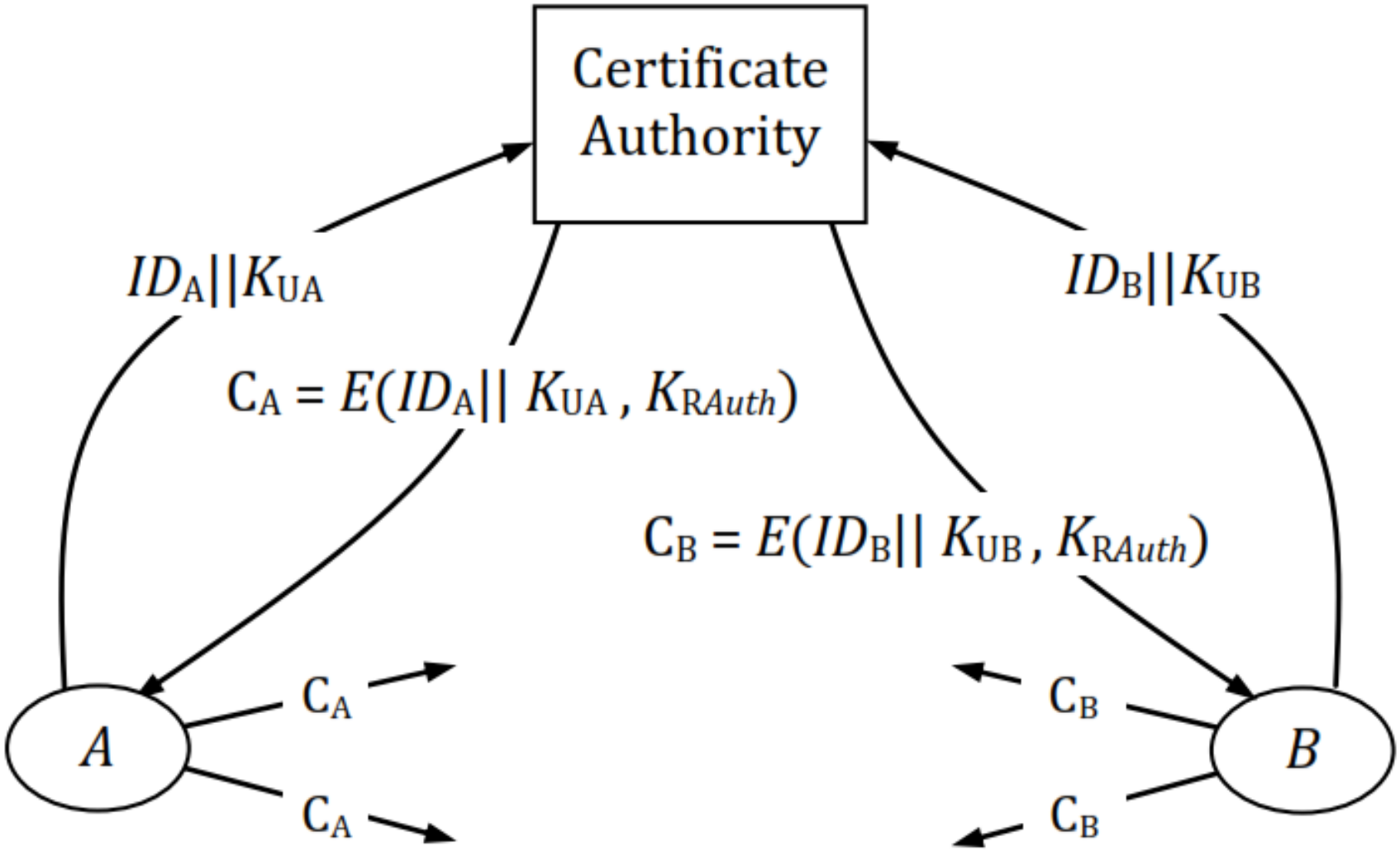
  4) Alice công khai chứng chỉ C_A

Trao đổi khóa công khai

- Để giảm …

  5) Bob muốn trao đổi thông tin với Alice thì sẽ giải mã C_A bằng khóa riêng của trung tâm chứng thực để có được khóa công khai K_{UA} của Alice. Do đó nếu Bob tin tưởng vào trung tâm chứng thực thì Bob sẽ tin tưởng là K_{UA} là tương ứng với ID_A, tức tương ứng với Alice.

Trao đổi khóa công khai dùng TT chứng thực



Dùng khóa công khai để trao đổi khóa bí mật