Chương 3

# MẬT MÃ KHÓA CÔNG KHAI

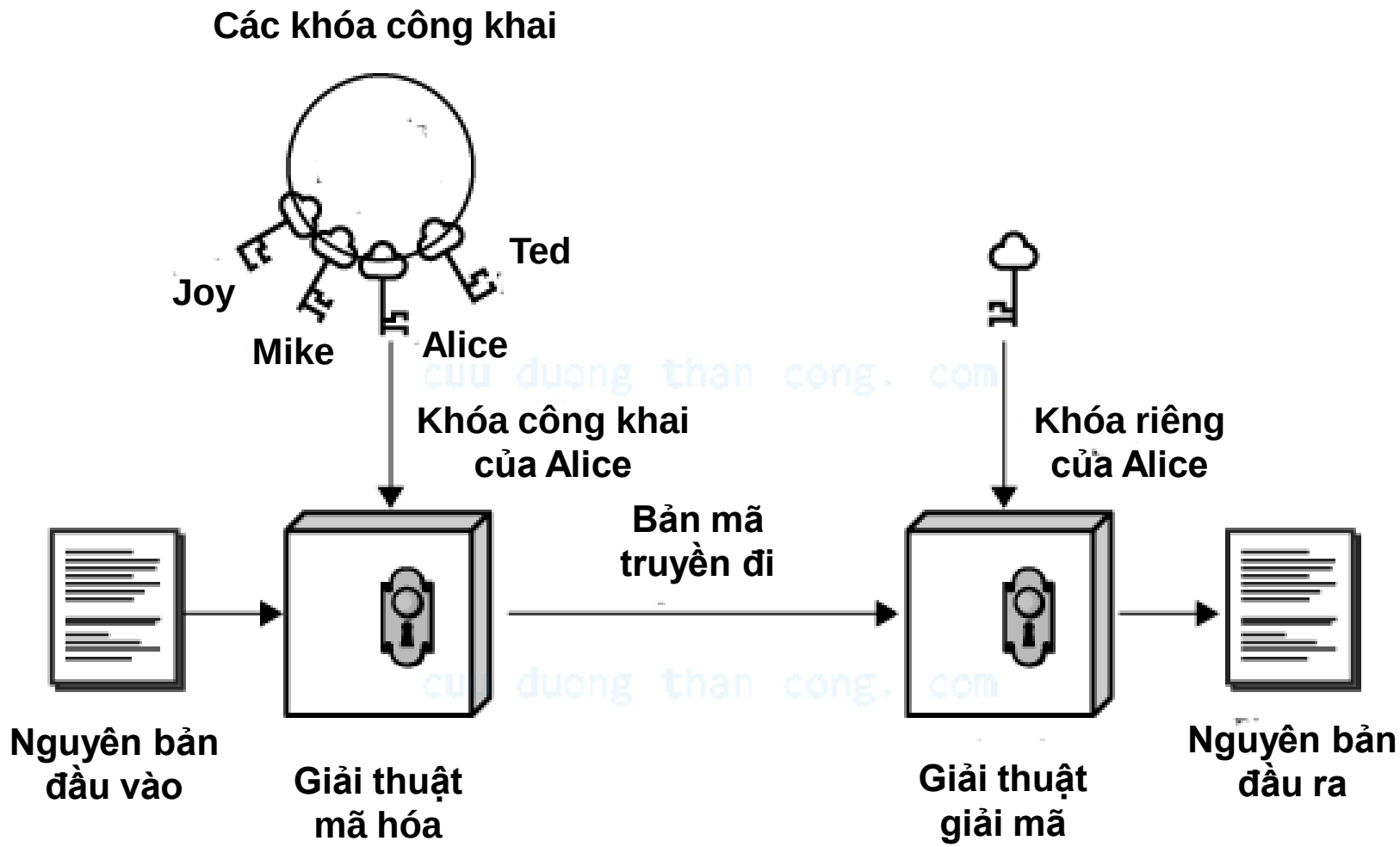
# Giới thiệu

- Những hạn chế của mật mã đối xứng
  - Vấn đề phân phối khóa
    - Khó đảm bảo chia sẻ mà không làm lộ khóa bí mật
    - Trung tâm phân phối khóa có thể bị tấn công
  - Không thích hợp cho chữ ký số
    - Bên nhận có thể làm giả thông báo nói nhận được từ bên gửi
- Mật mã khóa công khai đề xuất bởi Whitfield Diffie và Martin Hellman vào năm 1976
  - Khắc phục những hạn chế của mật mã đối xứng
  - Có thể coi là bước đột phá quan trọng nhất trong lịch sử của ngành mật mã
  - Bổ xung chứ không thay thế mật mã đối xứng

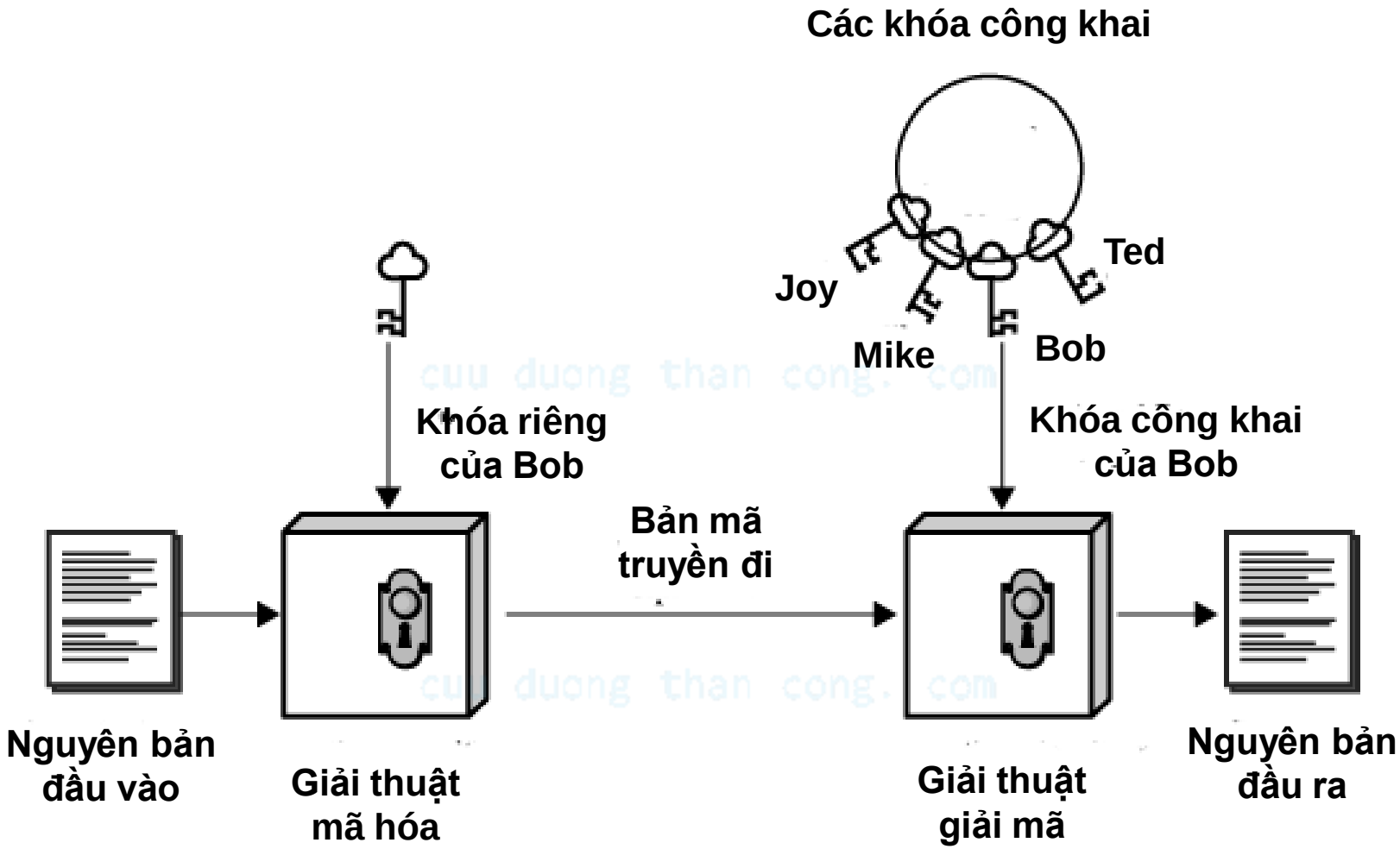
# Đặc điểm mật mã khóa công khai

- Còn gọi là mật mã hai khóa hay bất đối xứng
- Các giải thuật khóa công khai sử dụng 2 khóa
  - Một khóa công khai
    - Ai cũng có thể biết
    - Dùng để mã hóa thông báo và thẩm tra chữ ký
  - Một khóa riêng
    - Chỉ nơi giữ được biết
    - Dùng để giải mã thông báo và ký (tạo ra) chữ ký
- Có tính bất đối xứng
  - Bên mã hóa không thể giải mã thông báo
  - Bên thẩm tra không thể tạo chữ ký

# Mã hóa khóa công khai



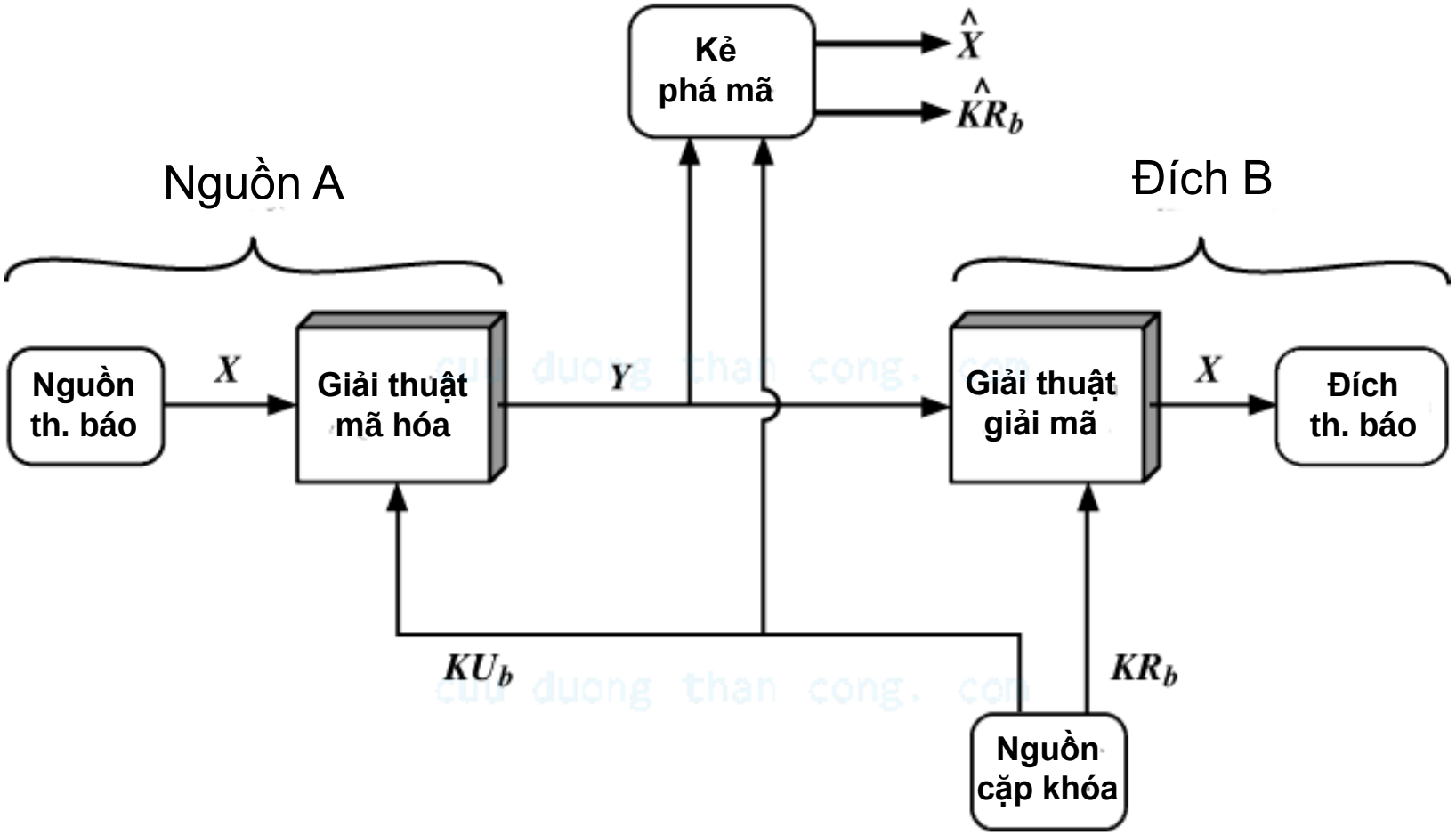
# Xác thực



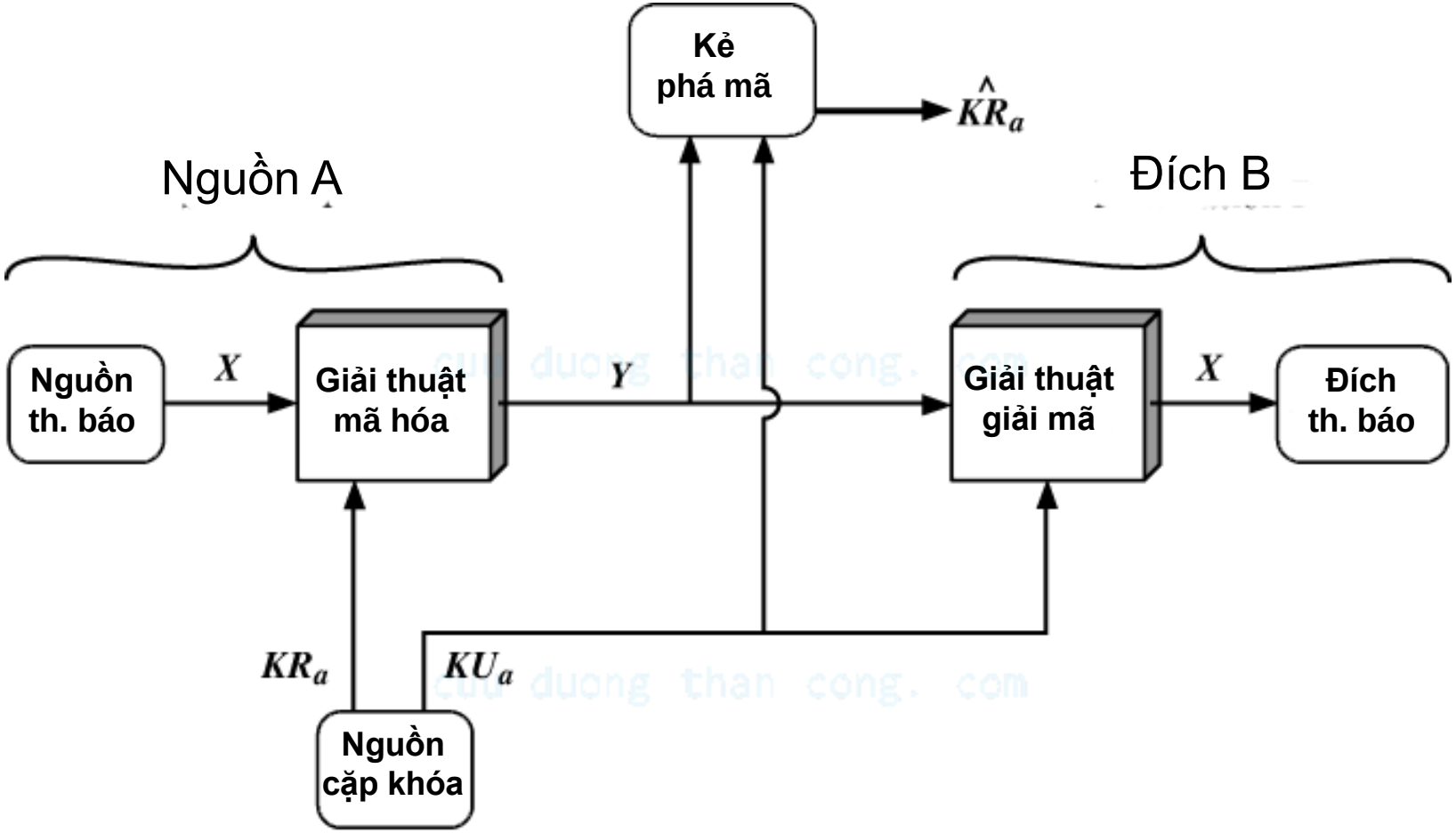
# Ứng dụng mật mã khóa công khai

- Có thể phân ra 3 loại ứng dụng
  - Mã hóa/giải mã
    - Đảm bảo sự bí mật của thông tin
  - Chữ ký số
    - Hỗ trợ xác thực văn bản
  - Trao đổi khóa
    - Cho phép chia sẻ khóa phiên trong mã hóa đối xứng
- Một số giải thuật khóa công khai thích hợp cho cả 3 loại ứng dụng; một số khác chỉ có thể dùng cho 1 hay 2 loại

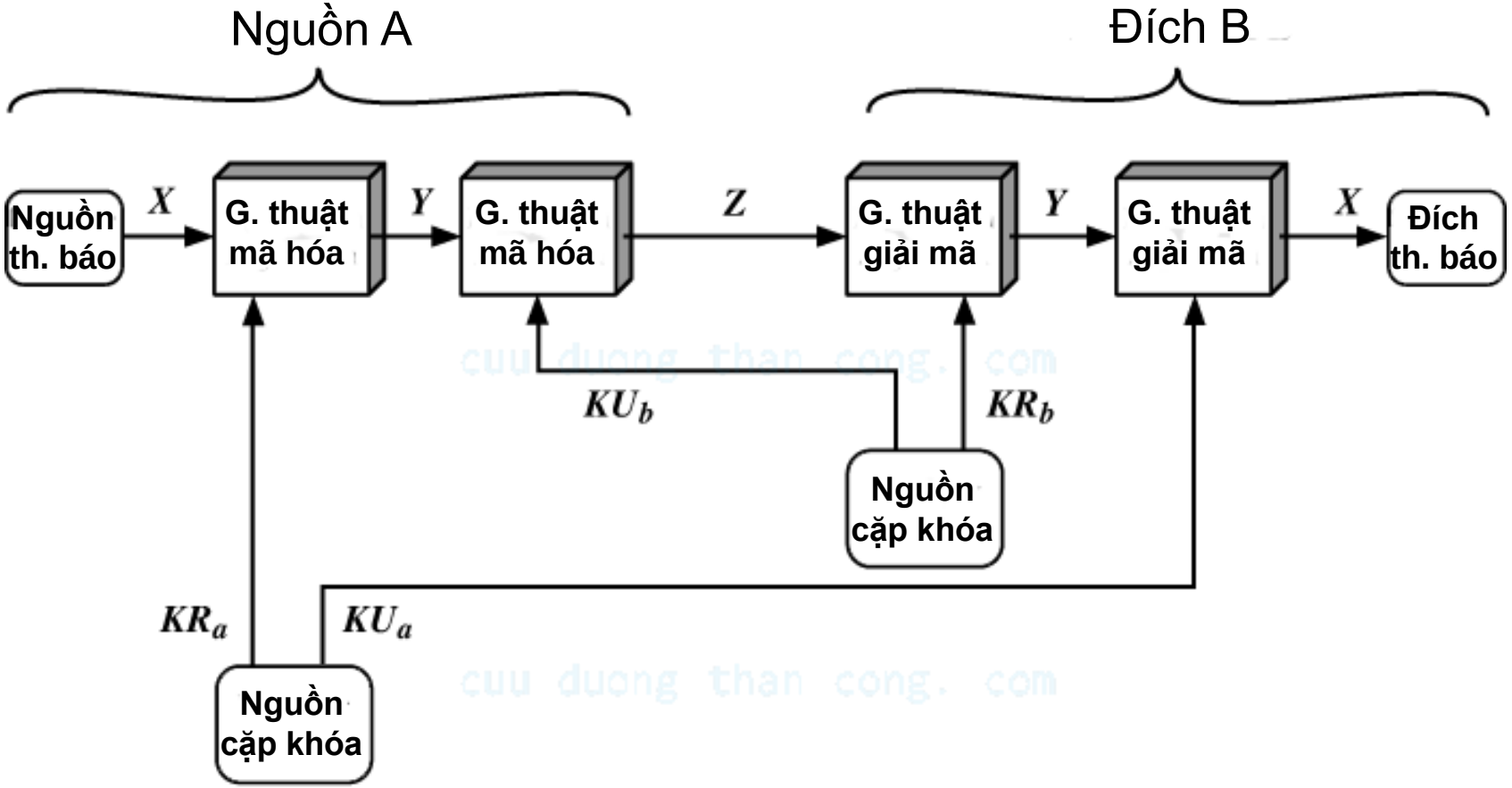
# Mô hình đảm bảo bí mật



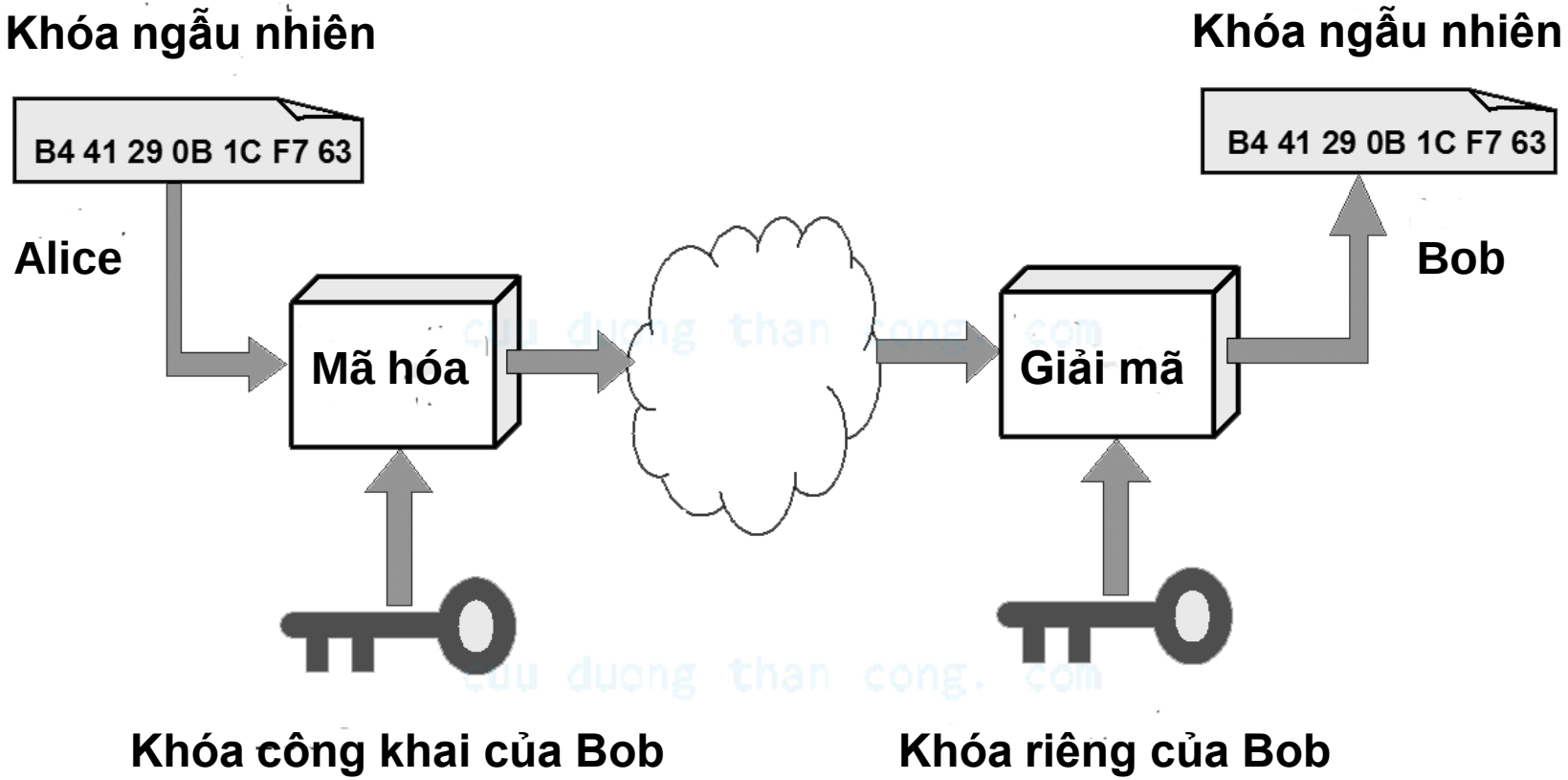
# Mô hình xác thực



# Mô hình kết hợp



# Trao đổi khóa



# Các điều kiện cần thiết

- Bên B dễ dàng tạo ra được cặp ($KU_b$, $KR_b$)
- Bên A dễ dàng tạo ra được $C = E_{KU_b}(M)$
- Bên B dễ dàng giải mã $M = D_{KR_b}(C)$
- Đối thủ không thể xác định được $KR_b$ khi biết $KU_b$
- Đối thủ không thể xác định được M khi biết $KU_b$ và C
- Một trong hai khóa có thể dùng mã hóa trong khi khóa kia có thể dùng giải mã
  - $M = D_{KR_b}(E_{KU_b}(M)) = D_{KU_b}(E_{KR_b}(M))$
  - Không thực sự cần thiết

# Hệ mã hóa RSA

- Đề xuất bởi Ron Rivest, Adi Shamir và Len Adleman (MIT) vào năm 1977
- Hệ mã hóa khóa công khai phổ dụng nhất
- Mã hóa khối với mỗi khối là một số nguyên < n
  - Thường kích cỡ n là 1024 bit ≈ 309 chữ số thập phân
- Đăng ký bản quyền năm 1983, hết hạn năm 2000
- An ninh vì chi phí phân tích thừa số của một số nguyên lớn là rất lớn

# Tạo khóa RSA

- Mỗi bên tự tạo ra một cặp khóa công khai - khóa riêng theo các bước sau :
  - Chọn ngẫu nhiên 2 số nguyên tố đủ lớn p [redacted]q
  - Tính n = pq
  - Tính [redacted]) = (p-1)(q-1)
  - Chọn ngẫu nhiên khóa mã hóa e sao cho 1 < e < [redacted]) và gcd(e, [redacted])) = 1
  - Tìm khóa giải mã d ≤ n thỏa mãn e.d ≡ 1 mod [redacted])
- Công bố khóa mã hóa công khai KU = {e, n}
- Giữ bí mật khóa giải mã riêng KR = {d, n}
  - Các giá trị bí mật p và q bị hủy bỏ

# Thực hiện RSA

- Để mã hóa 1 thông báo nguyên bản M, bên gửi thực hiện
  - Lấy khóa công khai của bên nhận KU = {e, n}
  - Tính $C = M^e \bmod n$
- Để giải mã bản mã C nhận được, bên nhận thực hiện
  - Sử dụng khóa riêng KR = {d, n}
  - Tính $M = C^d \bmod n$
- Lưu ý là thông báo M phải nhỏ hơn n
  - Phân thành nhiều khối nếu cần

# Vì sao RSA khả thi

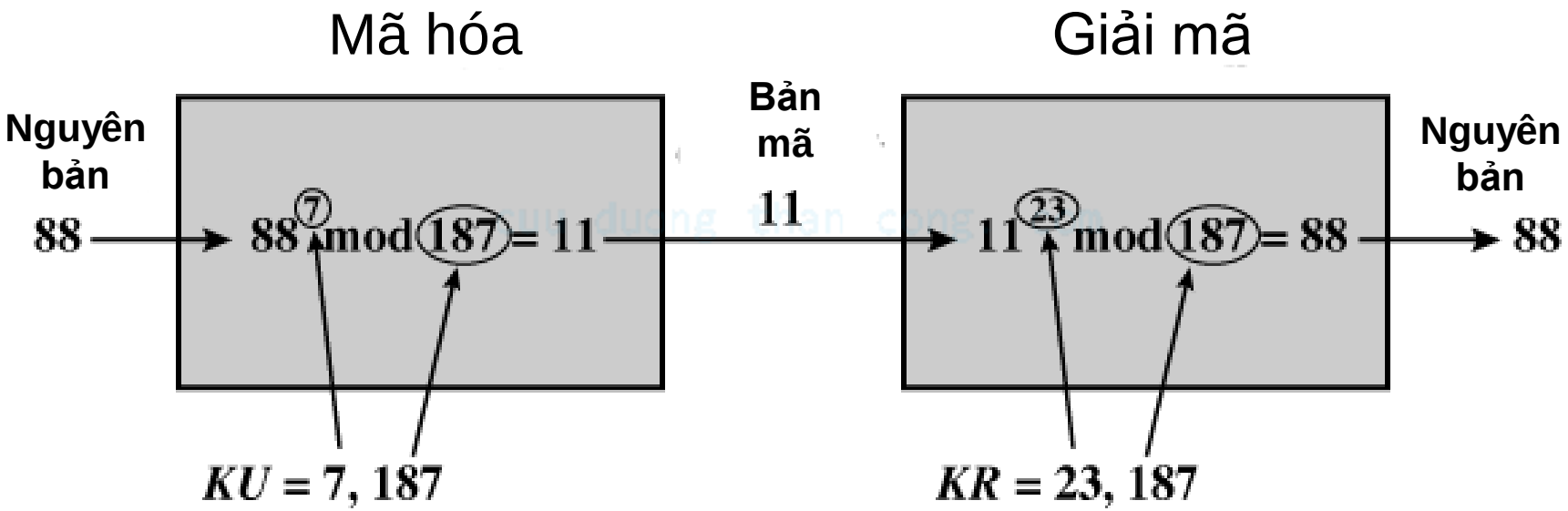
- Theo định lý Euler
  - [redacted], n : gcd(a, n) = 1 [redacted] mod n = 1
  - [redacted]) là số các số nguyên dương nhỏ hơn n và nguyên tố cùng nhau với n
- Đối với RSA có
  - n = pq với p và q là các số nguyên tố
  - [redacted]) = (p - 1)(q - 1)
  - ed ≡ 1 mod [redacted]) [redacted]số nguyên k : ed = k[redacted]) + 1
  - M < n
- Có thể suy ra
  - $C^d \bmod n = M^{ed} \bmod n = M^{k[redacted]) + 1} \bmod n = M \bmod n = M$

# Ví dụ tạo khóa RSA

- Chọn 2 số nguyên tố p = 17 và q = 11
- Tính n = pq = 17 [redacted]11 = 187
- Tính [redacted]) = (p - 1)(q - 1) = 16 [redacted]10 = 160
- Chọn e : gcd(e, 160) = 1 và 1 < e < 160; lấy e = 7
- Xác định d : de ≡ 1 mod 160 và d ≤ 187

  Giá trị d = 23 vì 23 [redacted]7 = 161 = 1 [redacted]160 + 1
- Công bố khóa công khai KU = {7, 187}
- Giữ bí mật khóa riêng KR = {23, 187}
  - Hủy bỏ các giá trị bí mật p = 17 và q = 11

# Ví dụ thực hiện RSA



# Chọn tham số RSA

- Cần chọn p và q đủ lớn
- Thường chọn e nhỏ
- Thường có thể chọn cùng giá trị của e cho tất cả người dùng
- Trước đây khuyến nghị giá trị của e là 3, nhưng hiện nay được coi là quá nhỏ
- Thường chọn e = $2^{16} - 1 = 65535$
- Giá trị của d sẽ lớn và khó đoán

# An ninh của RSA

- Khóa 128 bit là một số giữa 1 và một số rất lớn

  340.282.366.920.938.000.000.000.000.000.000.000.000

- Có bao nhiêu số nguyên tố giữa 1 và số này

  ≈ n / ln(n) = $2^{128}$ / ln($2^{128}$) ≈
  3.835.341.275.459.350.000.000.000.000.000.000.000

- Cần bao nhiêu thời gian nếu mỗi giây có thể tính được $10^{12}$ số

  Hơn 121,617,874,031,562,000 năm (khoảng 10 triệu lần tuổi của vũ trụ)

- An ninh nhưng cần đề phòng những điểm yếu

# Phá mã RSA

- Phương pháp vét cạn
  - Thử tất cả các khóa riêng có thể
    - Phụ thuộc vào độ dài khóa
- Phương pháp phân tích toán học
  - Phân n thành tích 2 số nguyên tố p và q
  - Xác định trực tiếp [redacted]) không thông qua p và q
  - Xác định trực tiếp d không thông qua [redacted])
- Phương pháp phân tích thời gian
  - Dựa trên việc đo thời gian giải mã
  - Có thể ngăn ngừa bằng cách làm nhiễu

# Phân tích thừa số RSA

- An ninh của RSA dựa trên độ phức tạp của việc phân tích thừa số n
- Thời gian cần thiết để phân tích thừa số một số lớn tăng theo hàm mũ với số bit của số đó
  - Mất nhiều năm khi số chữ số thập phân của n vượt quá 100 (giả sử làm 1 phép tính nhị phân mất 1 [redacted])
- Kích thước khóa lớn đảm bảo an ninh cho RSA
  - Từ 1024 bit trở lên
  - Gần đây nhất năm 1999 đã phá mã được 512 bit (155 chữ số thập phân)

# Hệ trao đổi khóa Diffie-Hellman

- Giải thuật mật mã khóa công khai đầu tiên
- Đề xuất bởi Whitfield Diffie và Martin Hellman vào năm 1976
  - Malcolm Williamson (GCHQ - Anh) phát hiện trước mấy năm nhưng đến năm 1997 mới công bố
- Chỉ dùng để trao đổi khóa bí mật một cách an ninh trên các kêch thông tin không an ninh
- Khóa bí mật được tính toán bởi cả hai bên
- An ninh phụ thuộc vào độ phức tạp của việc tính log rời rạc

# Thiết lập Diffie-Hellman

- Các bên thống nhất với nhau các tham số chung
  - q là một số nguyên tố đủ lớn
  - [redacted]à một nguyên căn của q
    - [redacted]mod q, [redacted]mod q,..., [redacted]$^{1}$ mod q là các số nguyên giao hoán của các số từ 1 đến q - 1
- Bên A
  - Chọn ngẫu nhiên làm khóa riêng $X_A < q$
  - Tính khóa công khai $Y_A =$ [redacted]$^{A}$ mod q
- Bên B
  - Chọn ngẫu nhiên làm khóa riêng $X_B < q$
  - Tính khóa công khai $Y_B =$ [redacted]$^{B}$ mod q

# Trao đổi khóa Diffie-Hellman

- Tính toán khóa bí mật
  - Bên A biết khóa riêng $X_A$ và khóa công khai $Y_B$

$$K = Y_B^{X_A} \bmod q$$

  - Bên B biết khóa riêng $X_B$ và khóa công khai $Y_A$

$$K = Y_A^{X_B} \bmod q$$

- Chứng minh

$$Y_A^{X_B} \bmod q = (\blacksquare^{A} \bmod q)^{X_B} \bmod q$$
$$= \blacksquare^{A X_B} \bmod q$$
$$= \blacksquare^{B X_A} \bmod q$$
$$= (\blacksquare^{B} \bmod q)^{X_A} \bmod q$$
$$= Y_B^{X_A} \bmod q$$

# Ví dụ Diffie-Hellman

- Alice và Bob muốn trao đổi khóa bí mật
- Cùng chọn q = 353 và [redacted] = 3
- Chọn ngẫu nhiên các khóa riêng
  - Alice chọn $X_A = 97$, Bob chọn $X_B = 233$
- Tính toán các khóa công khai
  - $Y_A = 3^{97} \bmod 353 = 40$ (Alice)
  - $Y_B = 3^{233} \bmod 353 = 248$ (Bob)
- Tính toán khóa bí mật chung
  - $K = Y_B^{X_A} \bmod 353 = 248^{97} \bmod 353 = 160$ (Alice)
  - $K = Y_A^{X_B} \bmod 353 = 40^{233} \bmod 353 = 160$ (Bob)

# Hạn chế của khóa công khai

- Tốc độ xử lý
  - Các giải thuật khóa công khai chủ yếu dùng các phép nhân chậm hơn nhiều so với các giải thuật đối xứng
  - Không thích hợp cho mã hóa thông thường
  - Thường dùng trao đổi khóa bí mật đầu phiên truyền tin
- Tính xác thực của khóa công khai
  - Bất cứ ai cũng có thể tạo ra một khóa công bố đó là của một người khác
  - Chừng nào việc giả mạo chưa bị phát hiện có thể đọc được nội dung các thông báo gửi cho người kia
  - Cần đảm bảo những người đăng ký khóa là đáng tin