Nhập môn An toàn thông tin

PGS. Nguyễn Linh Giang

Bộ môn Truyền thông và Mạng máy tính

Bài toán xác thực

Nội dung

- Bài toán xác thực.
- Tấn công vào hệ xác thực
- Các phương pháp xác thực thông điệp
  - Mã xác thực thông điệp
  - Hàm băm
- Chữ ký số

Bài toán xác thực

- Các yêu cầu của bài toán xác thực
  - Điểm lại các dạng tấn công
    - Tấn công vào tính riêng tư:
      - Giải mật: giải mật nội dung thông điệp.
      - Phân tích luồng truyền tải: xác định mẫu thông điệp, xác định tần suất trao đổi thông điệp, định vị, xác định chức năng các trạm, định vị
      - Dạng tấn công thụ động.
      - Đảm bảo tính riêng tư: ngăn chặn bằng mật mã và làm nhiễu thông tin.

Bài toán xác thực

- Tấn công vào tính xác thực:
  - Trá hình: đưa ra các thông điệp vào hệ thống với tên giả mạo.
  - Thay đổi nội dung thông điệp: phá huỷ tính toàn vẹn.
  - Thay đổi trình tự trao đổi thông điệp: tấn công vào giao thức.
  - Thay đổi theo tiến trình thời gian: làm trễ hoặc phát lại thông điệp.
  - Từ chối dịch vụ: từ chối gửi hoặc nhận thông điệp: sử dụng chữ ký điện tử.
  - Xác thực:
    - Xác thực các bên trao đổi thông điệp.
    - Làm rõ nguồn gốc thông điệp.
    - Xác định tính toàn vẹn thông điệp – xác thực nội dung
    - Xác thực phiên làm việc
    - Chống phủ nhận.

Bài toán xác thực

- Các tiêu chuẩn xác thực
  - Xác thực chủ thể tham gia vào trao đổi thông tin
  - Thông điệp có nguồn gốc;
  - Nội dung thông điệp toàn vẹn, không bị thay đổi trong quá trình truyền tin (xác thực nội dung thông điệp);
  - Thông điệp được gửi đúng trình tự và thời điểm (xác thực phiên);
- Mục đích của bài toán xác thực:
  - Chống lại các tấn công chủ động:
    - Chống giả mạo;
    - Thay đổi nội dung dữ liệu;
    - Thay đổi trình tự trao đổi thông tin (hoạt động của các giao thức).
- Các phương pháp xác thực và chống giả mạo:
  - Mã hoá thông điệp;
  - Sử dụng mã xác thực thông điệp;
  - Sử dụng hàm băm;
  - Sử dụng các giao thức xác thực

Bài toán xác thực

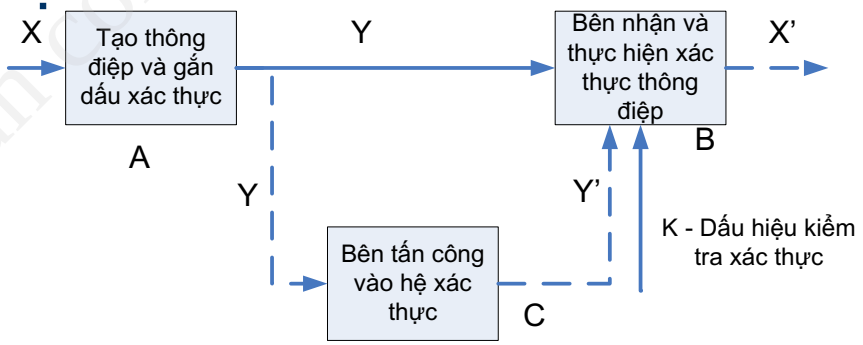
- Các hàm xác thực
  - Các cơ chế xác thực được thực hiện trên hai mức:
    - Mức thấp: trong hệ thống phải có các hàm chức năng cho phép kiểm tra tính xác thực của chủ thể và thông điệp:
      - Hàm tạo các giá trị đặc trưng xác thực chủ thể và thông điệp.
    - Mức cao:
      - Sử dụng các hàm xác thực trong các giao thức xác thực.
      - Cho phép thẩm định tính xác thực của chủ thể và thông điệp.

Bài toán xác thực

- Các dạng hàm xác thực:
  - Mã hoá thông điệp: sử dụng hàm mã hoá để xác thực dựa vào việc sở hữu khoá bí mật.
  - Mã xác thực thông điệp: tạo ra mã xác thực thông điệp độ dài cố định bằng phương pháp mã hoá.
  - Hàm băm xác thực thông điệp: tạo mã băm của thông điệp với độ dài cố định.
  - Chữ ký số: tạo dấu hiệu đặc trưng xác định duy nhất chủ thể.
  - Các phương pháp tạo sinh các dấu hiệu xác thực
  - Các giao thức xác thực

Tấn công vào hệ xác thực
Xác thực và xác thực hoàn hảo

- Vấn đề giả mạo và xác thực
  - Vấn đề: tồn tại hay không phương pháp xác thực hoàn hảo chống lại giả mạo !?



  - Các kịch bản tấn công vào hệ xác thực:
    - Kịch bản 1:
      - A tạo bản tin X, gắn dấu hiệu xác thực, được bản tin Y và gửi Y cho B
    - Kịch bản 2:
      - Giữa A và B không có phiên làm việc.
      - C tạo ra văn bản Y' , giả mạo A và gửi cho B

Lý thuyết xác thực Simmons
Xác thực và xác thực hoàn hảo

- Xác suất tấn công giả mạo
  - P_s: xác suất tấn công thành công bằng thay thế;
  - P_i: xác suất tấn công thành công bằng mạo danh;
  - Xác suất giả mạo thành công: $P_D=\max(P_i, P_s)$
  - Khoá K: thông tin tham gia vào quá trình xác thực
  - N_X: số lượng thông điệp gốc X_i sao cho

    $$P\{X = X_i\} \neq 0$$

  - N_K: số lượng các dấu hiệu xác thực K_L: $P\{K = K_L\} \neq 0$
  - N_Y: số lượng văn bản được gắn dấu hiệu xác thực Y_j, sao cho $P\{Y = Y_j\} \neq 0$

Xác thực bằng cách mã hoá

- Sử dụng phương pháp mật mã khoá đối xứng
  - Thông điệp gửi từ đúng nguồn vì chỉ có người gửi biết khoá bí mật dùng chung
  - Nội dung không thể bị thay đổi vì văn bản rõ có cấu trúc nhất định
  - Các gói tin được đánh số thứ tự và có mã hoá nén không thể thay đổi trình tự và thời điểm nhận được
- Sử dụng phương pháp mật mã khoá công khai
  - Không chỉ xác thực thông điệp mà còn tạo chữ ký số
  - Phức tạp và mất thời gian hơn mã hoá đối xứng

Xác thực bằng phương pháp mã hóa

- Xác thực: chống giả mạo
  - Xây dựng các dấu hiệu đặc trưng cho đối tượng cần xác thực:
    - Đối tượng cần xác thực:
      - Chủ thể tham gia vào quá trình trao đổi thông tin: nguồn gốc thông tin từ các nguồn được xác thực.
      - Nội dung thông tin trao đổi: không bị sửa đổi trong quá trình trao đổi – tính nguyên bản của thông tin.
      - Xác thực phiên trao đổi thông tin: giao thức trao đổi, trật tự hoạt động của giao thức, thời gian trao đổi thông tin, dấu hiệu phiên.
    - Dấu hiệu: dùng các phương pháp mã hóa để tạo dấu hiệu xác thực: dùng các thuật toán mật mã.

Xác thực bằng phương pháp mã hóa

- Quá trình xác thực
  - Tạo dấu hiệu đặc trưng từ đối tượng.
    - Bằng cách sử dụng các phương pháp mật mã.
    - Tính bền vững của dấu hiệu: khi thay đổi nội dung cần xác thực hoặc thay đổi dấu hiệu: hệ thống xác thực phát hiện dễ dàng.
  - Dấu hiệu được gắn kèm đối tượng trong quá trình trao đổi thông tin
  - Bên nhận sẽ tính toán lại dấu hiệu từ nội dung thông tin
  - So sánh dấu hiệu vừa tính được với dấu hiệu gửi kèm.
  - Nếu trùng khớp: dấu hiệu được xác thực; Ngược lại: không được xác thực.

Xác thực dùng mã xác thực thông điệp (MAC - checksum)

- Dùng mã xác thực thông điệp (MAC Message Authentication Code)
- Là khối có kích thước nhỏ cố định gắn vào thông điệp tạo ra từ thông điệp đó và khóa bí mật chung
- Bên nhận thực hiện cùng giải thuật trên thông điệp và khoá để so xem MAC có chính xác không
- Giải thuật tạo MAC giống giải thuật mã hóa nhưng không cần giải mã

Xác thực dùng mã xác thực thông điệp (MAC - checksum)

- MAC = $C_K(M)$
  - M: là bản tin
  - K: là khoá mật được chia sẻ chỉ bởi người gửi và người nhận;
  - $C_K(M)$: là một hàm xác thực, cho kết quả là một xâu ký tự có độ dài cố định;

Xác thực dùng mã xác thực thông điệp (MAC - checksum)

- Có thể có nhiều thông điệp có cùng chung MAC
  - Nhưng nếu biết 1 thông điệp và MAC, rất khó tìm ra một thông điệp khác cùng MAC
  - Các thông điệp có cùng xác suất tạo ra MAC
- Đáp ứng 3 tiêu chuẩn xác thực

Mã hoá bản tin và cách tấn công của đối phương

- Mã hoá bản tin
  - Đối xứng
  - Không đối xứng
- Sự an toàn của thuật toán phụ thuộc độ dài bit của khoá
- Với 1 lần tấn công
  - 2^k lần thử cho khoá k bit

Mã hoá bản tin và cách tấn công của đối phương

- Ví dụ tấn công
  - Đối phương biết bản mật C (Ciphertext)
    - $P_i = D_{Ki}(C)$ cho tất cả khoá K_i
    - Đến khi P_i khớp với bản rõ P (Plaintext)
- Đối với CheckSum
  - MAC n bit → 2^n CheckSum tạo ra
  - N bản tin áp dụng ($N>>2^n$)
  - Khóa K bit → 2^k khóa tạo ra

Ví dụ tấn công vào MAC

- Giả sử: size(K) > size (MAC) (k>n)
- Match (so khớp): là bản M_i tạo ra gần khớp với bản M_1
- Dùng cách tấn công vét cạn (brute-force)

Ví dụ tấn công vào MAC

- Tấn công MAC bằng cách lặp lại:
  - Vòng 1:
    - Cho: M_1, $MAC_1 = C_K(M_1)$
    - Tính: $M_i = C_{Ki}(MAC_1)$ cho tất cả khoá
    - Số các so khớp tạo ra $\approx 2^{k-n}$
  - Vòng 2:
    - Cho: M_2, $MAC_2 = C_K(M_2)$
    - Tính $M_i = C_{Ki}(MAC_2)$ cho khoá còn lại.
    - Số cách so khớp tạo ra $\approx 2^{k-2n}$
  - …

Ví dụ tấn công vào MAC

- Kết quả:
  - Nếu k = a*n → mất a vòng để tìm ra
  - Nếu k < n thì ngay vòng 1 tạo ra luôn sự so khớp.
  - Ví dụ
    - Nếu một khoá kích thước k=80 bit
    - CheckSum kích thước là n=32 bit
    - Thì vòng 1 sẽ tạo ra khoảng 2^{48} khóa Vòng 2 sẽ thu hẹp xuống còn 2^{16} khóa

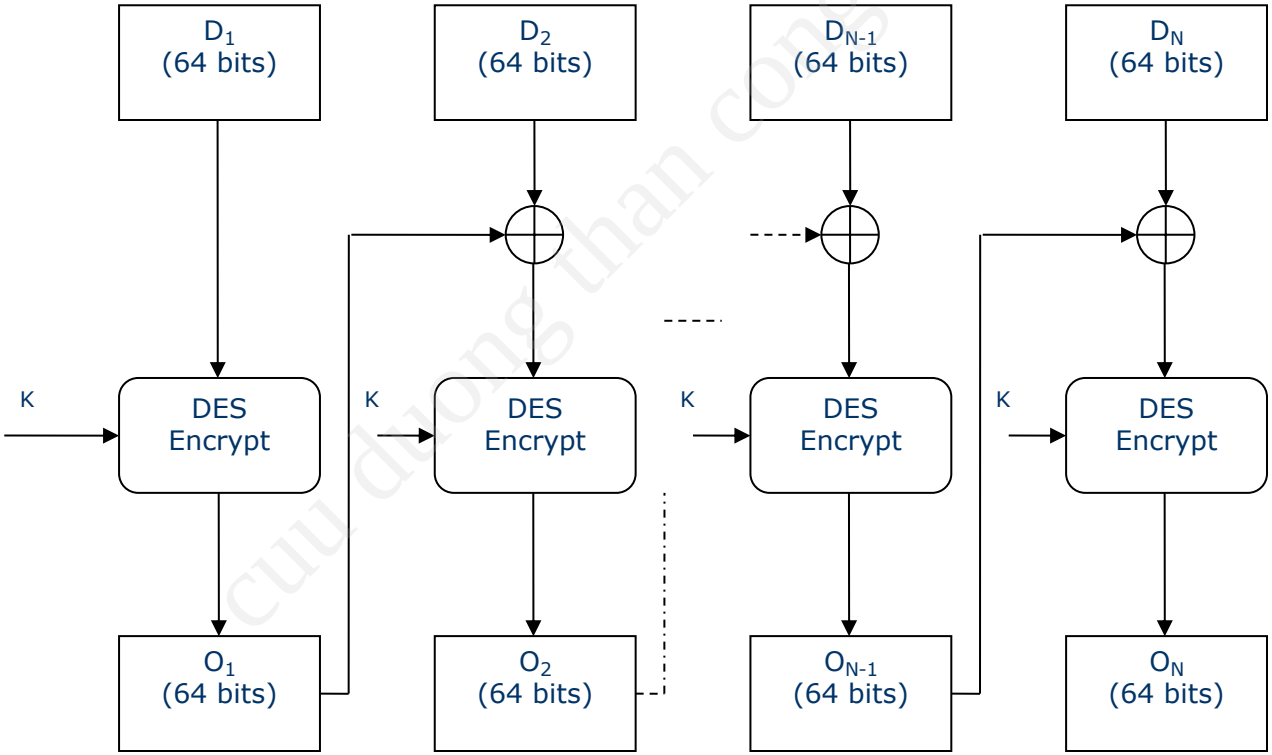
    Vòng 3 sẽ tạo chỉ 1 khoá đơn, và đó chính là khoá được dùng bởi người gửi.

Ví dụ tấn công vào MAC

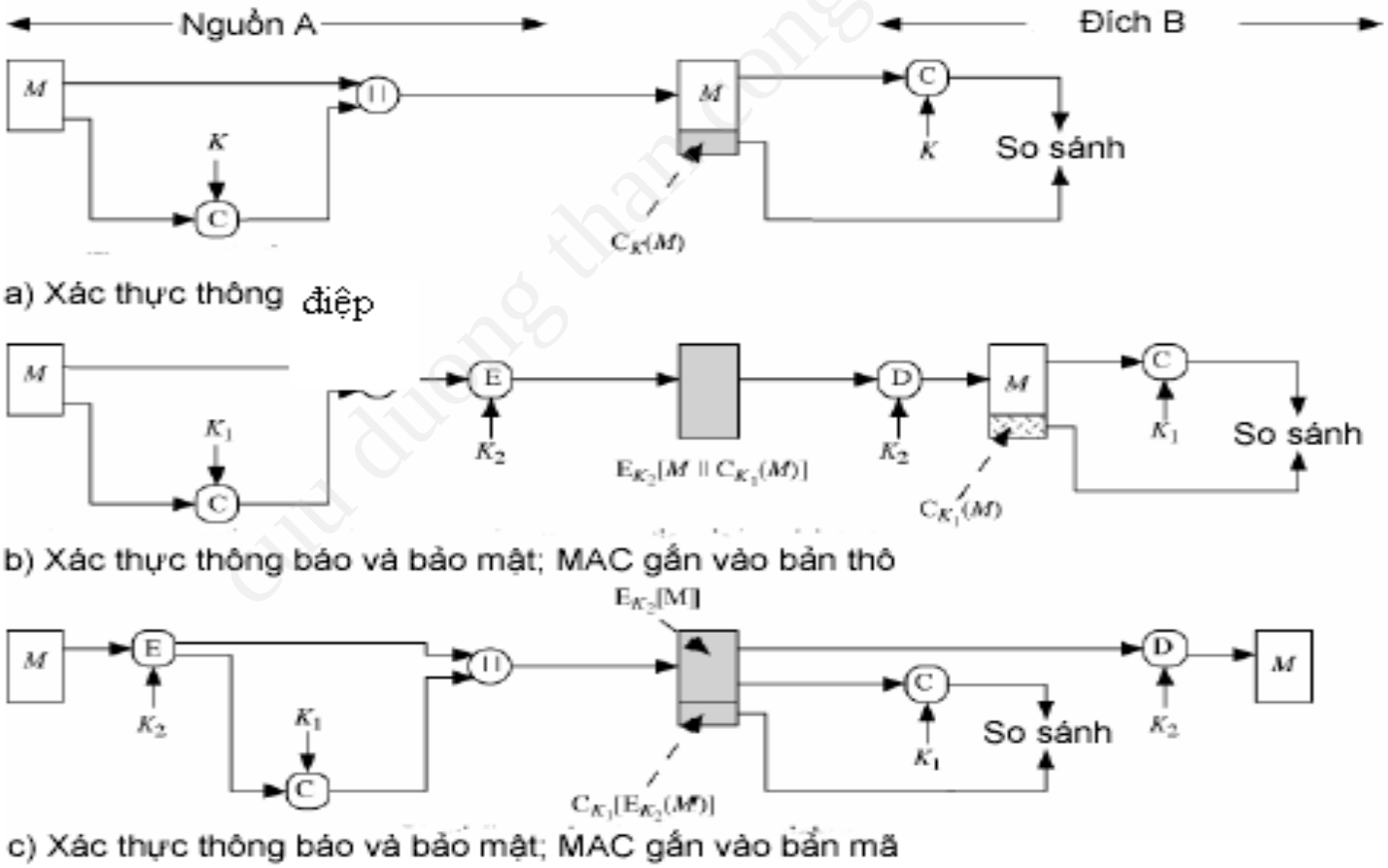
- Tồn tại khả năng có nhiều khoá thoả mãn việc so khớp

  ⇨ Đối phương có thể thực hiện cùng một kiểm tra trên một cặp(bảntin,CheckSum) mới.

Mật mã CheckSum dựa trên DES



Xác thực dùng mã xác thực thông điệp (MAC - checksum)



Xác thực dùng mã xác thực thông điệp (MAC - checksum)

- Chỉ cần xác thực, không cần mã hoá tốn thời gian và tài nguyên
  - Thông điệp hệ thống
  - Chương trình máy tính
- Tách riêng bảo mật và xác thực sẽ khiến tổ chức linh hoạt hơn
  - Chẳng hạn mỗi chức năng ở 1 tầng riêng
- Cần đảm bảo tính toàn vẹn của dữ liệu trong suốt thời gian tồn tại, không chỉ trong lúc lưu chuyển
  - Vì thông điệp có thể bị thay đổi sau khi giải mã

Xác thực dùng hàm băm

- Tạo ra hàm băm có kích thước xác định từ thông điệp đầu vào(không cần khoá): h=H(M)
- Hàm băm không cần giữ bí mật
- Giá trị băm gắn kèm với thông điệp để đảm bảo tính toàn vẹn của thông điệp
- Bất kỳ một sự thay đổi nhỏ nào trong thông điệp M cũng tạo ra sự thay đổi trong mã băm h

Các yêu cầu đối với hàm băm

- Có thể áp dụng với thông điệp M với độ dài bất kỳ
- Tạo ra giá trị băm h có độ dài cố định
- H(M) dễ dàng tính được với bất kỳ M nào
- Từ h rất khó tìm được M sao cho h=H(M): tính một chiều
- Từ M1 rất khó tìm được M_2 sao cho $H(M_1)=H(M_2)$
- Rất khó tìm được cặp (M_1,M_2) sao cho $H(M_1)=H(M_2)$

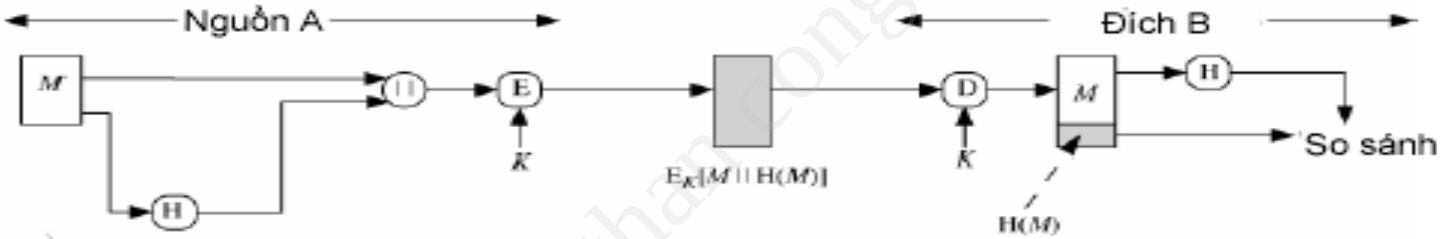
Các yêu cầu đối với hàm băm

- Đặc điểm 4 là đặc điểm ”1 chiều ” (one - way). Nó tạo ra 1 mã cho bản tin nhưng không thể tạo ra 1 bản tin cho 1 mã
- Đặc điểm 5 đảm bảo:
  - 1 bản tin thay thế khi bị băm không cùng giá trị băm với bản tin đã cho là
  - Bảo vệ lại sự giả mạo khi sử dụng 1 mã băm được mã hóa.

Các yêu cầu đối với hàm băm

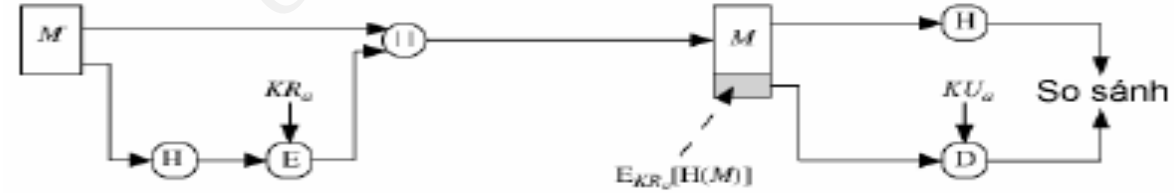
- Một hàm băm mà thoả mãn các đặc điểm từ 1→5 trong danh sách trên thì vẫn bị coi là 1 hàm băm kém. Nếu đặc điểm 6 được thoả mãn, nó mới được coi là một hàm băm tốt.
- Đặc điểm 6 bảo vệ bản tin khỏi một lớp các tấn công tinh vi như tấn công ngày sinh (birthday attack).

Xác thực dùng hàm băm



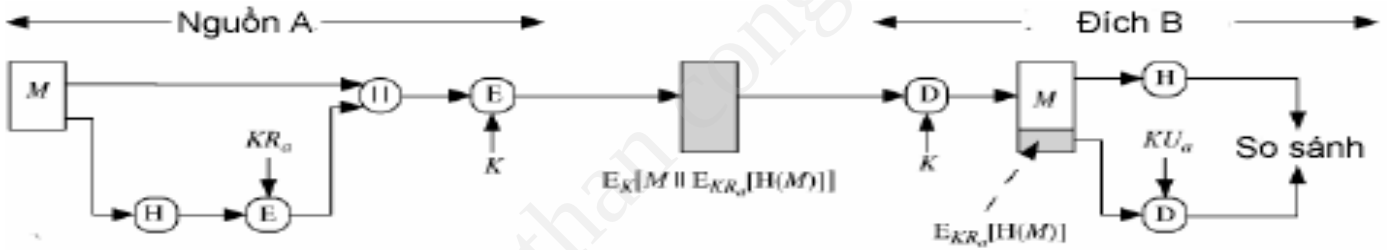
a) Xác thực thông báo và bảo mật; mã băm gắn vào bản thô

b) Xác thực thông báo; mã băm được mã hóa sử dụng phương pháp đối xứng



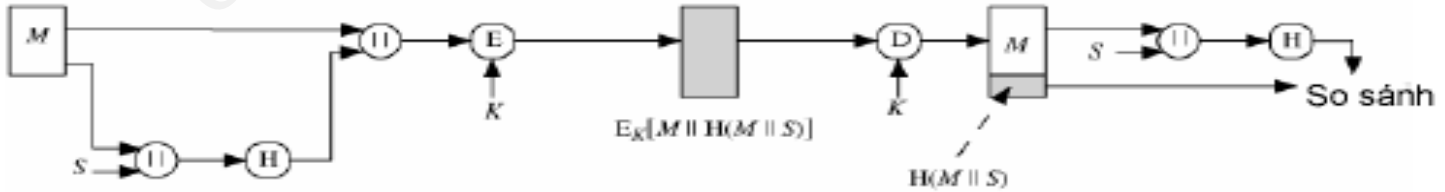
c) Xác thực thông báo; mã băm được mã hóa sử dụng phương pháp khóa công khai

Xác thực dùng hàm băm



d) Xác thực bằng mã hóa khóa công khai và bảo mật bằng mã hóa đối xứng

e) Xác thực không cần mã hóa nhờ hai bên chia sẻ một giá trị bí mật chung



f) Xác thực nhờ một giá trị bí mật chung; bảo mật bằng phương pháp đối xứng

So sánh MAC và Hash

- Tương tự hàm MAC nhưng gọi là hash không khoá, MAC là hash có khoá

Các hàm băm đơn giản

- Nguyên tắc hoạt động chung:
  - Input: file, message.. được chia thành chuỗi các block n bit
  - Xử lý đầu vào: mỗi block được xử lý tại 1 thời điểm và lặp lại với các block khác ⇨ tạo ra 1 giá trị băm n bit

Hàm băm XOR

- Thực hiện phép XOR bit-by-bit
- Có thể biểu diễn như sau:
  - $C_i = b_{i1} \oplus b_{i2} \oplus \ldots \oplus b_{im}$
  - Trong đó:

    C_i : bit thứ i của mã băm (i=1..n)

    m: Số Block n-bit của Input

    b_{ij} : bít thứ i của Block j

    $\oplus$: phép toán XOR bit

Hàm băm XOR

- Minh họa:

	Bit 1	**Bit 2**	…….	**Bit n**
Block 1	B_{11}	B_{21}	…….	B_{n1}
Block 2	B_{12}	B_{22}	……	B_{n2}
….	…..	……..	……	……
Block m	B_{1m}	B_{2m}	…….	B_{nm}
Hash Code	C_1	C_2	…….	C_n

Hàm băm RXOR

- Thực hiện: Xoay đi một bit rồi thực hiện phép XOR → tăng tính ngẫu nhiên
- Sơ đồ:
  - Khởi tạo n bit của giá trị băm bằng 0
  - Xử lý mỗi block n-bit thành công là như sau:
    - Xoay giá trị băm hiện tại sang trái 1 bit
    - XOR block với giá trị băm

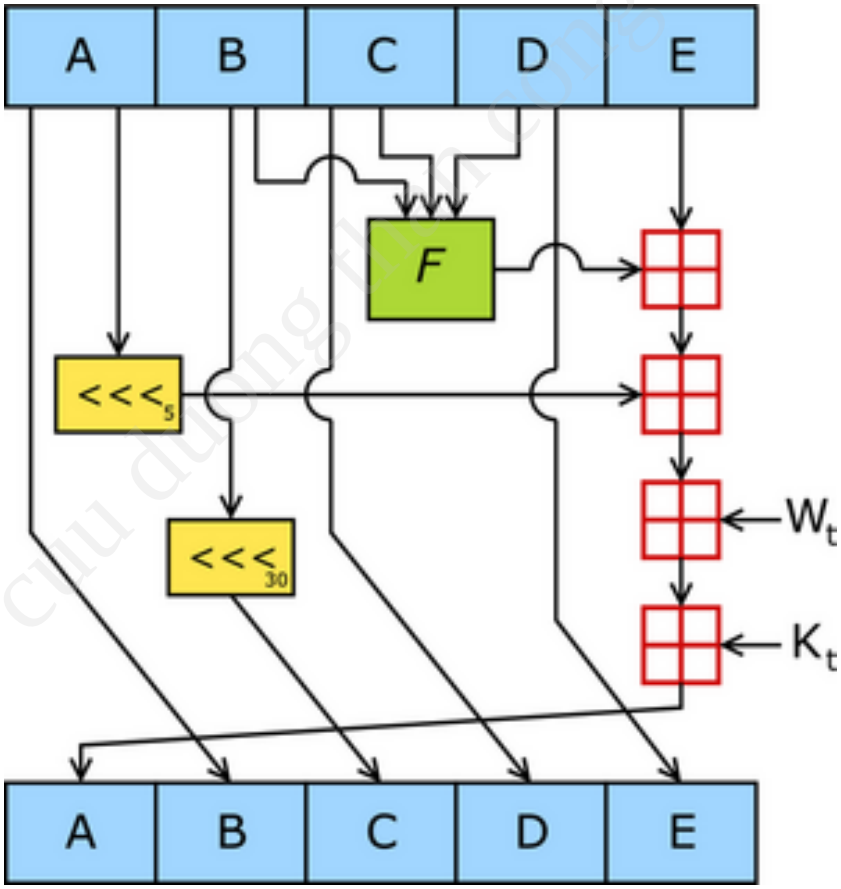
SHA-1 (Secure Hash Algorithm -1)

- Đây là một hàm băm 1 chiều
- Các phiên bản
  - SHA-0: Công bố năm 1993
  - SHA-1:
  - SHA-2: Bao gồm tập hợp SHA-224, SHA-256, SHA-384, và SHA-512
- Chúng được dùng bởi chính phủ Mỹ

SHA-1

- Đặc điểm của hàm:
  - Input: Đầu vào message có size < 2^{64}
    - Chia thành các Block có size = 512 bit
  - Ra: 1 Digest độ dài 160 bit
  - Bảo mật:
    - Không tính toán ra được thông điệp với 1 Digest đã cho
    - Không có 2 thông điệp cùng tạo ra 1 Digest

Sơ đồ hoạt động



Một số kết quả test

- Một số giá trị digest của SHA-1:
  - SHA1("The quick brown fox jumps over the lazy dog") == "2fd4e1c67a2d28fced849ee1bb76e7391b93eb12"
  - SHA1("The quick brown fox jumps over the lazy cog") == "de9f2c7fd25e1b3afad3e85a0bd17d9b100db4b3"
  - SHA1("") == "da39a3ee5e6b4b0d3255bfef95601890afd80709"

Chữ ký số

- Yêu cầu
- Phân loại
- Tạo và chứng thực chữ ký
- Chứng thư số

Yêu cầu

- Dựa trên thông điệp
- Sử dụng thông tin duy nhất thuộc về người gửi → chống giả mạo
- Dễ kiểm tra và nhận dạng
- Phải không thể tính toán để giả mạo được
- Để thoả mãn các yêu cầu trên, người ta thường sử dụng hàm băm.

Phân loại

- Thường được phân làm 2 loại:
  - ✓ Chữ ký trực tiếp
  - ✓ Chữ ký phân xử

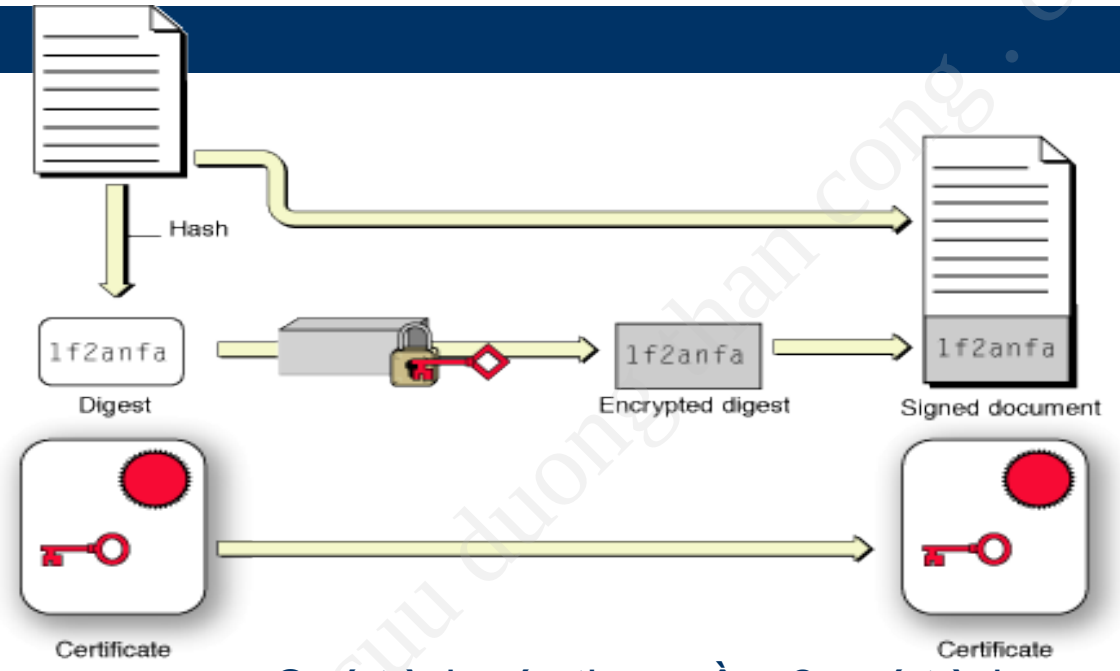
Chữ ký trực tiếp

- Chỉ bao gồm các thành phần truyền thông
- Có thể được tạo ra :
  - Mã hoã toàn bộ bản tin với khoá riêng của người gửi
  - Mã hoá mã băm của bản tin với khoá riêng của người gửi
- Tính hợp lệ của chữ ký phụ thuộc vào việc bảo mật khoá riêng của người gửi.

Chữ ký phân xử

- Hoạt động chung :
  - Mọi bản tin được gửi từ X đến Y phải thông qua A, để kiểm tra nguồn gốc và nội dung của nó
  - Bản tin được ghi lại thời gian rồi được gửi đến B + 1 thông điệp được đảm bảo bởi A.
  - Sự có mặt của A giải quyết vấn đề: X có thể phủ nhận bản tin này

Tạo chữ ký

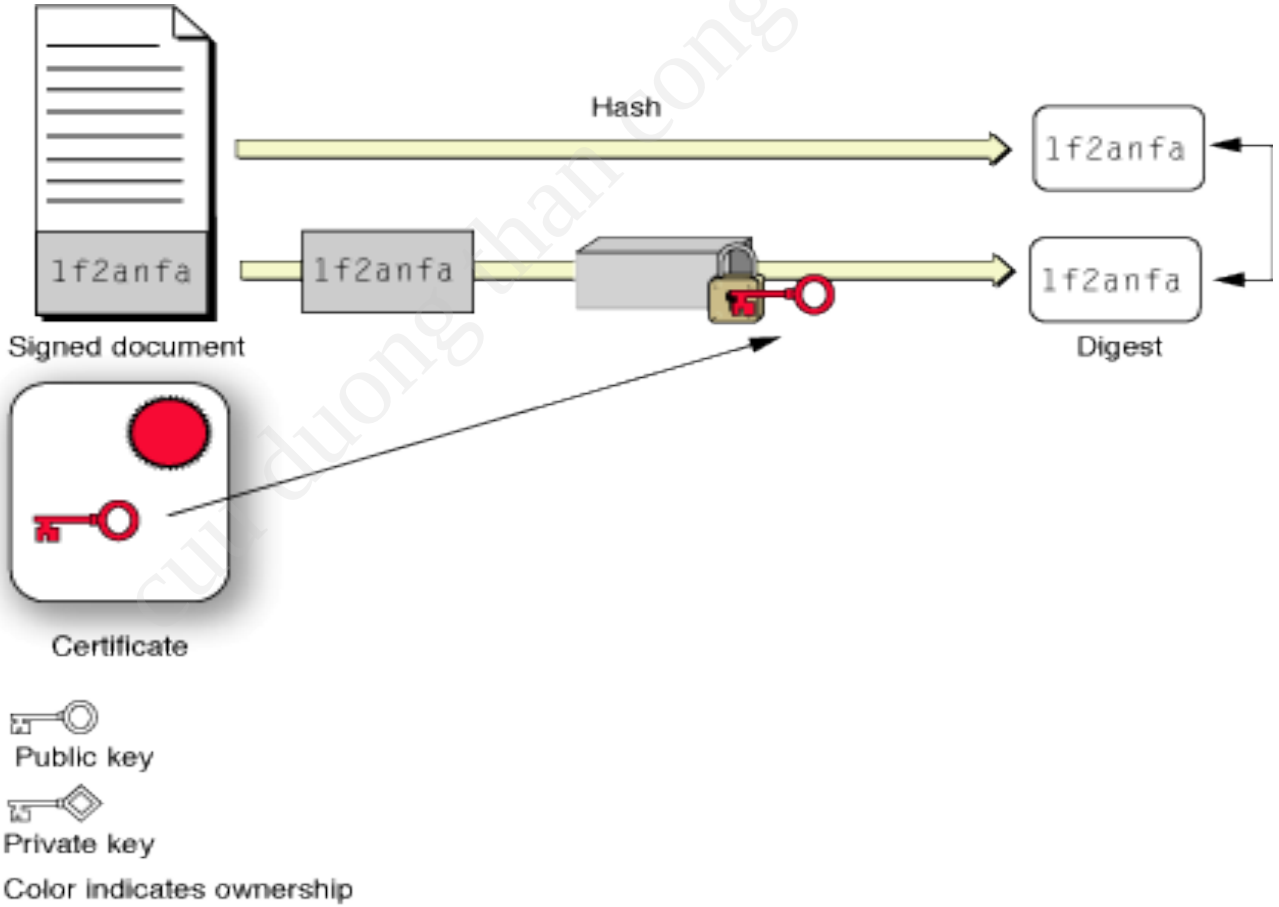


Quá trình xác thực gồm 2 quá trình con:

-Xác thực người gửi: thông qua khoá riêng của người gửi và kiểm chứng khoá riêng bằng khoá công khai của người gửi chứa trong chứng chỉ số

-Xác thực nội dung (tính toàn vẹn của văn bản) thông qua mẫu của thông điệp – mã băm của thông điệp.

Chứng thực chữ ký



Digital Certificate

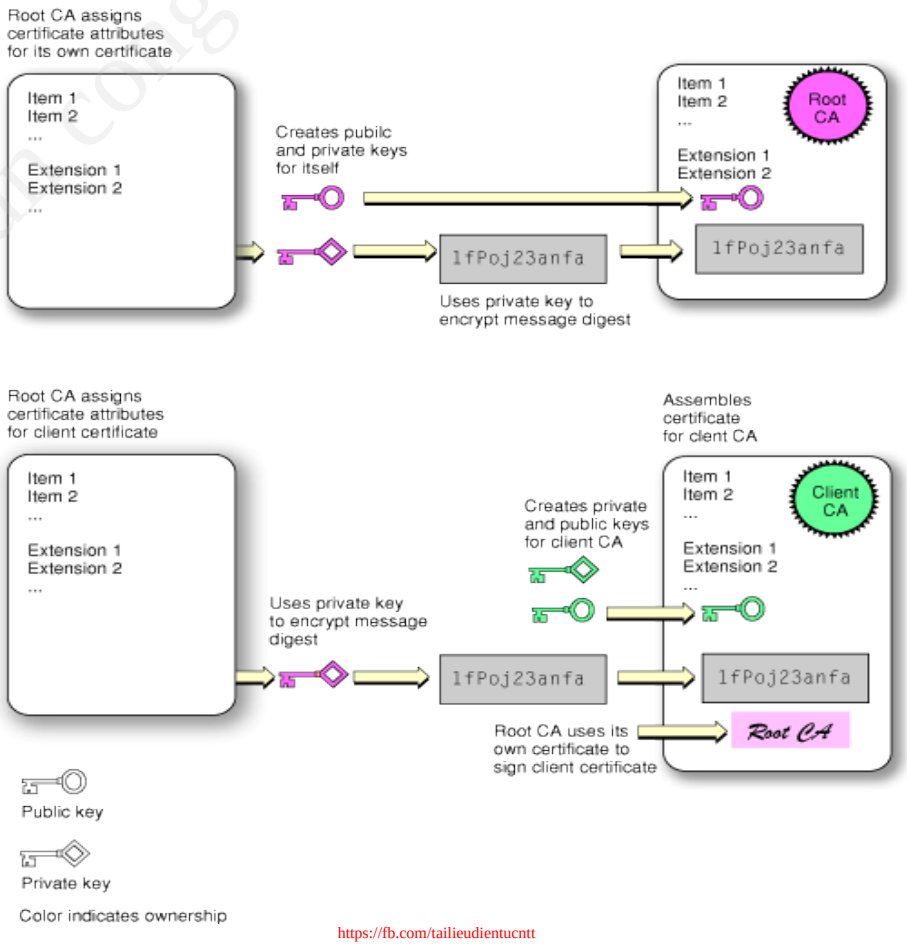
- Để chứng thực được chữ ký điện tử bắt buộc người nhận phải có khoá chung của người gửi.
- Bản chất cặp khoá này không liên hệ với thuộc tính của người sử dụng → cần có cơ chế để liên kết chúng với người dùng → các certificate
- Các Certificate được CA cung cấp

Các thông tin trong Certificate

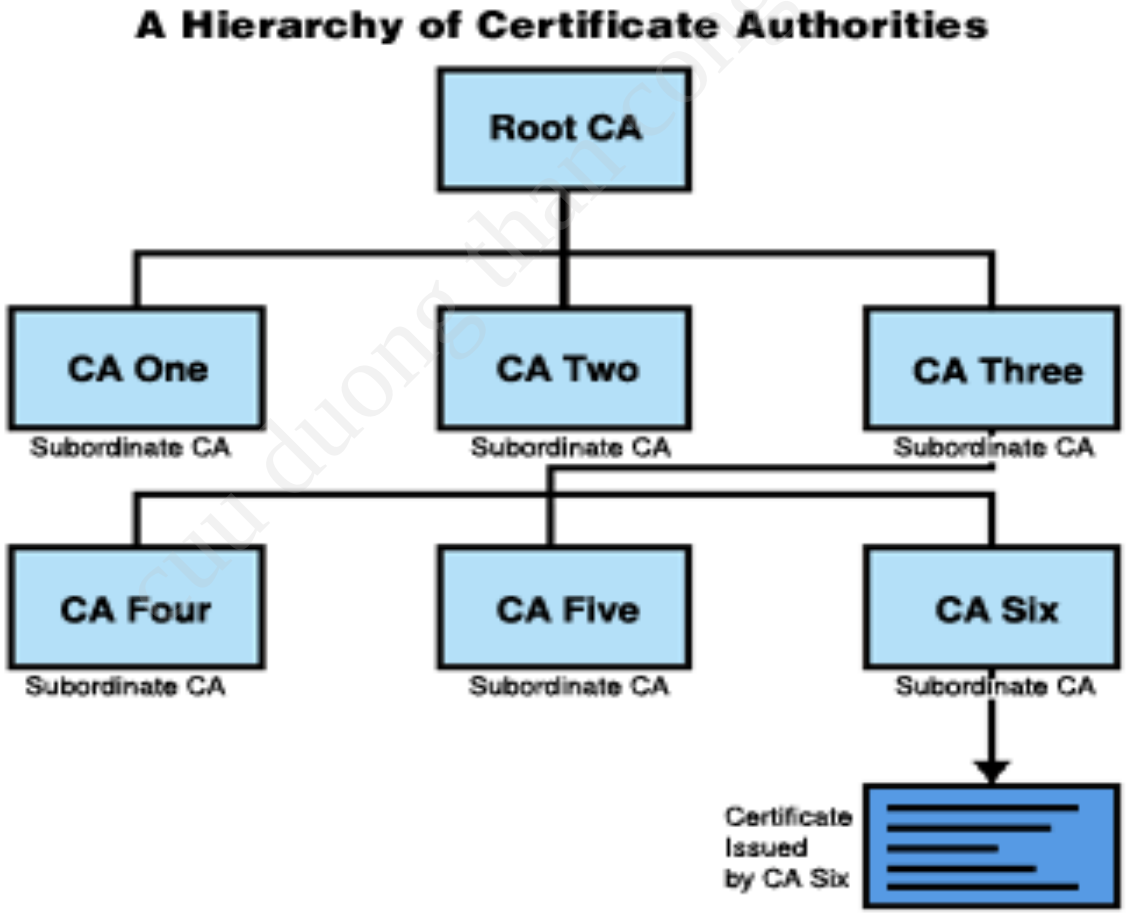
- Phiên bản
- Số serial
- Nhà cung cấp Certifficate
- Người sở hữu Certificate
- Thời gian hiệu lực của Certificate
- Các thuộc tính
- Chữ ký số của nhà cung cấp
- Khoá công khai của người sở hữu Certificate
- Thuật toán băm dùng để tạo chữ ký.

Tạo Certificate

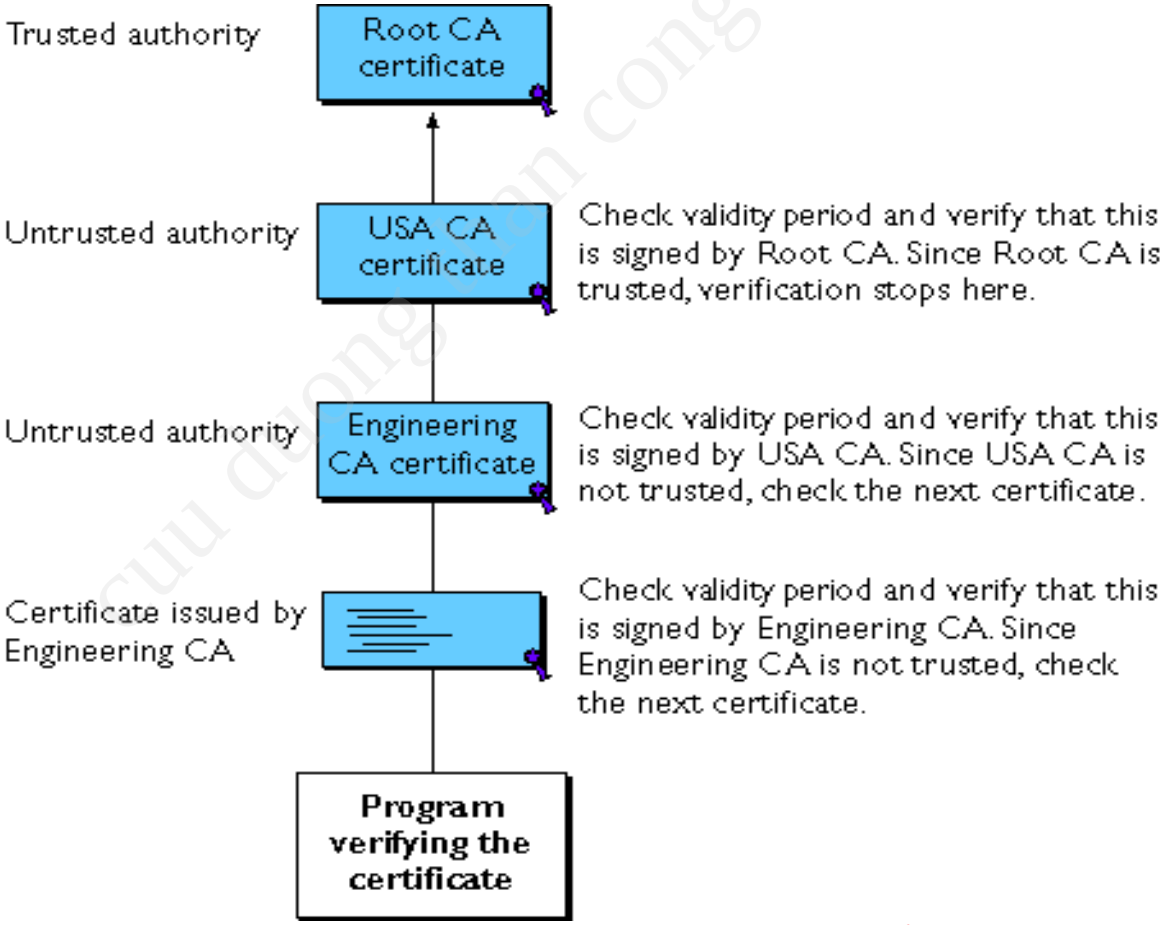
- Các Certificate được tạo ra còn để chứng thực cho bản thân nó
- Các CA có cấu trúc phân cấp
- Minh hoạ quá trình tạo Certificate cho CA gốc và CA mức thấp hơn



Cấu trúc phân cấp của CA



Xác thực chuỗi Certificate



Các giao thức xác thực

- Xác thực hai bên
- Các phương pháp mã hoá cổ điển
- Phương pháp mã hoá khoá công khai

Xác thực hai bên

- Tại đây, chúng ta chỉ xem xét vấn đề quản lý phân phối khoá
- Tồn tại 2 vấn đề :
  - Tính tin cậy : ngăn chặn hiện tượng giả mạo và tấn công vào khoá phiên
  - Xác định thời điểm: chống lại kiểu tấn công replay

Phương pháp chống replay

- 2 phương pháp:
  - Timestamp: gắn 1 timestamp vào bản tin --> yêu cầu đồng bộ
  - Challenge/Response: A sẽ gửi đến B 1 nonce và đợi trả lời của B. Nếu có chứa giá trị nonce chính xác thì mới bắt đầu gửi bản tin

Đánh giá 2 phương pháp

- Timestamp: không áp dụng cho các ứng dụng hướng kết nối
  - Yêu cầu đồng bộ giữa các tiến trình đồng hồ
  - Cơ hội tấn công thành công sẽ tăng lên nếu có 1 khoảng thời gian không đồng bộ
  - Tính luôn thay đổi và không dự đoàn trước được của các độ trễ trong mạng
- Challenge/Response: không áp dụng cho các ứng dụng không hướng kết nối
  - Yêu cầu bắt tay trước khi truyền thông không kết nối
  - Phương pháp tốt nhất: tạo sự đồng bộ giữa đồng hồ ở mỗi bên

Phương pháp mã hoá kinh điển

- Sử dụng 1 trung tâm phân phối khoá tin cậy(KDC)
- Mỗi bên chia sẻ 1 khoá mật với KDC:khoá chính
- KDC sẽ sinh ra các khoá phiên: sử dụng1 trên kết nối giữa 2 bên
- KDC còn chịu trách nhiệm phân phối các khoá phiên sử dụng khoá chính để bảo vệ quá trình phân phối khoá

Mã hoá khoá công khai

- Phương pháp này đảm bảo là mỗi bên đều lưu trữ khoá công khai hiện thời của bên còn lại
- Tất cả các phương pháp trên vẫn tồn tại những điểm thiếu sót
- Có nhiều phương pháp:
  - Denny
  - Woo và Law