

Xác thực thông điệp

Nguyễn Linh Giang
Khoa CNTT

Nội dung

- Xác thực thông điệp
- Mã xác thực thông điệp
- Hàm băm

Các tấn công vào hệ thống mạng

- Phát hiện nội dung thông điệp
- Phân tích lưu lượng
- Giả mạo
- Thay đổi nội dung
- Thay đổi thứ tự
- Thay đổi thời gian
- Phủ nhận giao dịch

Các vấn đề xác thực

- Các tiêu chí xác thực thông điệp
 - Thông điệp có nguồn gốc rõ ràng chính xác
 - Nội dung thông điệp toàn vẹn, không bị thay đổi
 - Thông điệp được gửi đúng trình tự và thời điểm
- Mục đích chống lại tấn công chủ động(chống giả mạo, thay đổi dữ liệu ...)
- Các phương pháp xác thực thông điệp
 - Mã hoá thông điệp(1)
 - Sử dụng mã xác thực thông điệp(2)
 - Sử dụng hàm băm(3)

Xác thực bằng cách mã hoá (1)

- Sử dụng mã hoá 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 thô có cấu trúc nhất định
 - Các gói tin được đánh số thứ tự và có mã hoá nên không thể thay đổi trình tự và thời điểm nhận được
- Sử dụng mã hoá 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

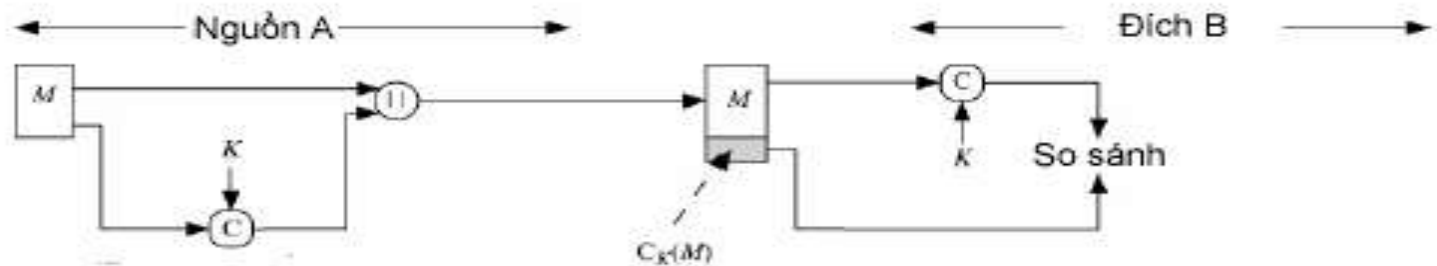
Xác thực dùng mã checksum (2)

- 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 hoá nhưng không cần giải ngược

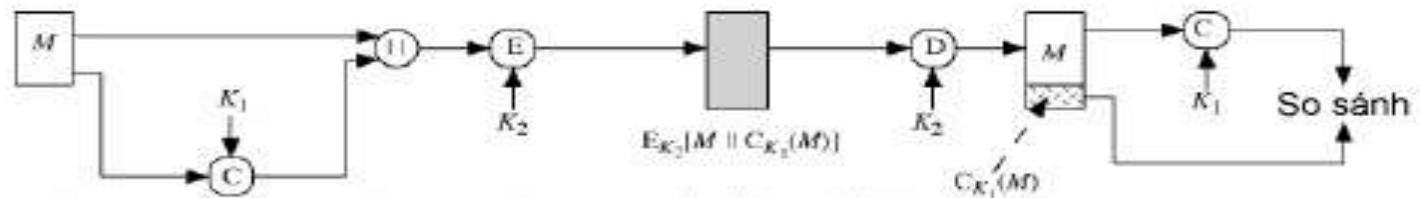
Checksum (tiếp..)

- 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

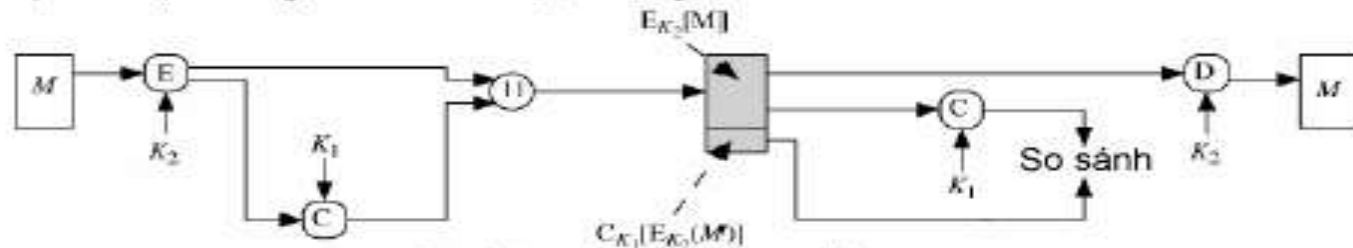
Checksum(tiếp..)



a) Xác thực thông báo



b) Xác thực thông báo và bảo mật; MAC gắn vào bản thô



c) Xác thực thông báo và bảo mật; MAC gắn vào bản mã

Tại sao dùng MAC

- Chỉ cần xác thực, không cần mã hoá tốn thời gian và tài nguyên
 - Thông báo 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

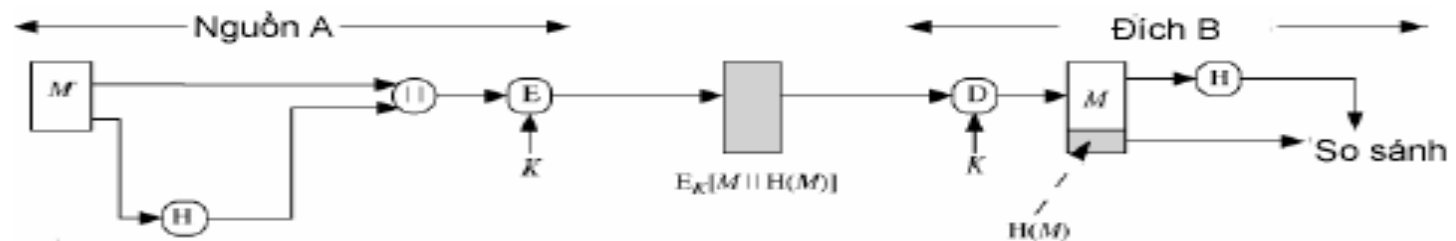
Xác thực dùng hàm băm(3)

- 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 xác thực của thông điệp
- Đảm bảo tính toàn vẹn của thông điệp: bất kỳ một sự thay đổi nhỏ nào trong M cũng tạo ra sự thay đổi trong 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 $M2$ sao cho $H(M1)=H(M2)$
- Rất khó tìm được cặp $(M1, M2)$ sao cho $H(M1)=H(M2)$

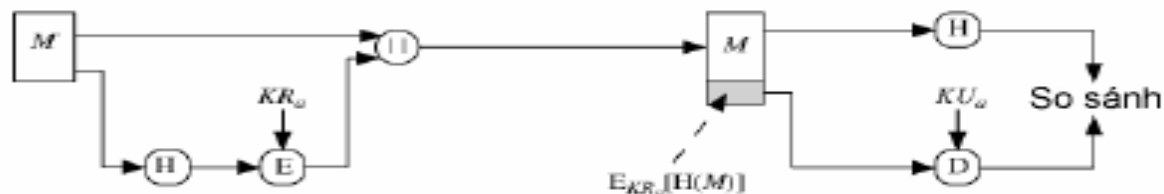
Các cơ chế khi dùng hàm băm



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

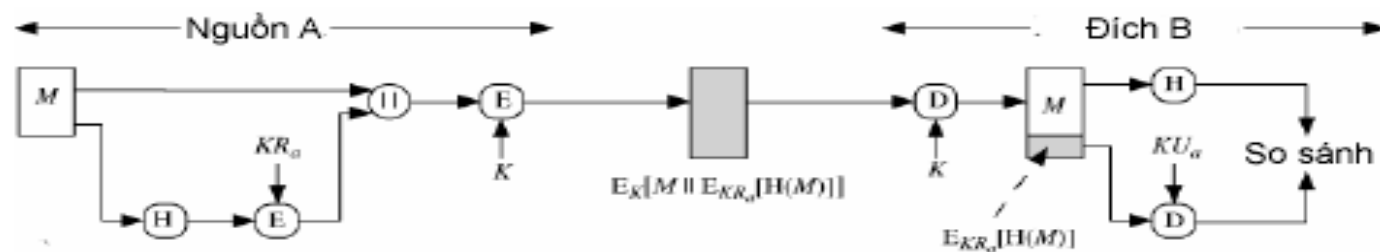


b) Xác thực thông báo; mã băm được mã hóa sử dụng phương pháp đối xứng

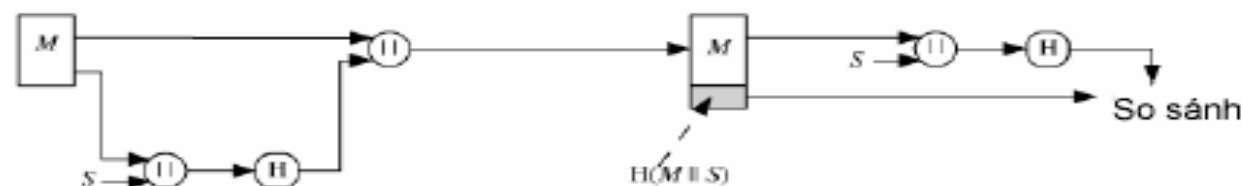


c) Xác thực thông báo; mã băm được mã hóa sử dụng phương pháp khóa công khai

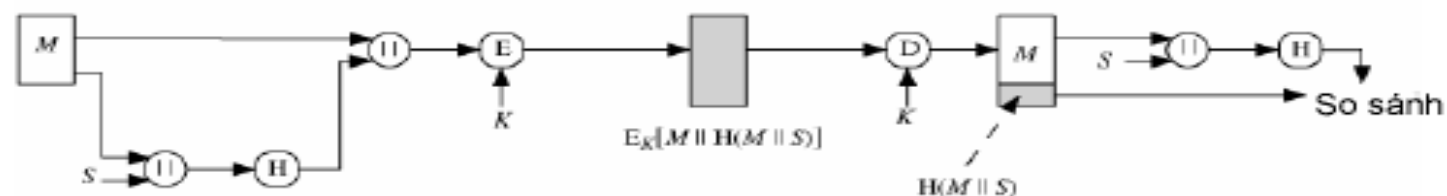
Các cơ chế khi dùng hàm băm(tiếp..)



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

Phần 2: Ciphergraphic CheckSum và HASH FUNCTION



Phần 2: Thuật toán CheckSum và Hash Function

- Mật mã CheckSum
- HASH FUNCTION

Ciphergraphic CheckSum

- Chính là MAC
- Được gắn vào cuối bản tin như một "dấu vân tay"
- Người nhận xác thực bằng cách tính lại MAC

Ciphergraphic CheckSum

- $MAC = C_K(M)$
 - M: là một biến (độ dài bản tin)
 - K: là 1 khoá mật được chia sẻ chỉ bởi người gửi và người nhận
 - $C_K(M)$: là một hàm xác thực độ dài cố định

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)
 - $P_i = D_{K_i}(C)$ cho tất cả khoá K_i
 - Đến khi P_i khớp với bản rõ P (Plaintext)
- Đối với CheckSum
 - MAC n bit $\rightarrow 2^n$ CheckSum tạo ra
 - N bản tin áp dụng ($N \gg 2^n$)
 - Khóa K bit $\rightarrow 2^k$ khóa tạo ra

Ví dụ tấn công vào MAC

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

- Tấn công MAC bằng cách lặp lại:

- Vòng 1:

- Cho: $M_1, MAC_1 = C_K(M_1)$
 - Tính: $M_i = C_{K_i}(MAC_1)$ 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 $M_i = C_{K_i}(MAC_2)$ cho khoá còn lại.
 - Số cách so khớp tạo ra $\approx 2^{k-2n}$

- ...

- Kết quả:

- Nếu $k = a \cdot n \rightarrow$ 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.

- 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(bản tin,Checksum) mới.

Thuật toán mật mã CheckSum

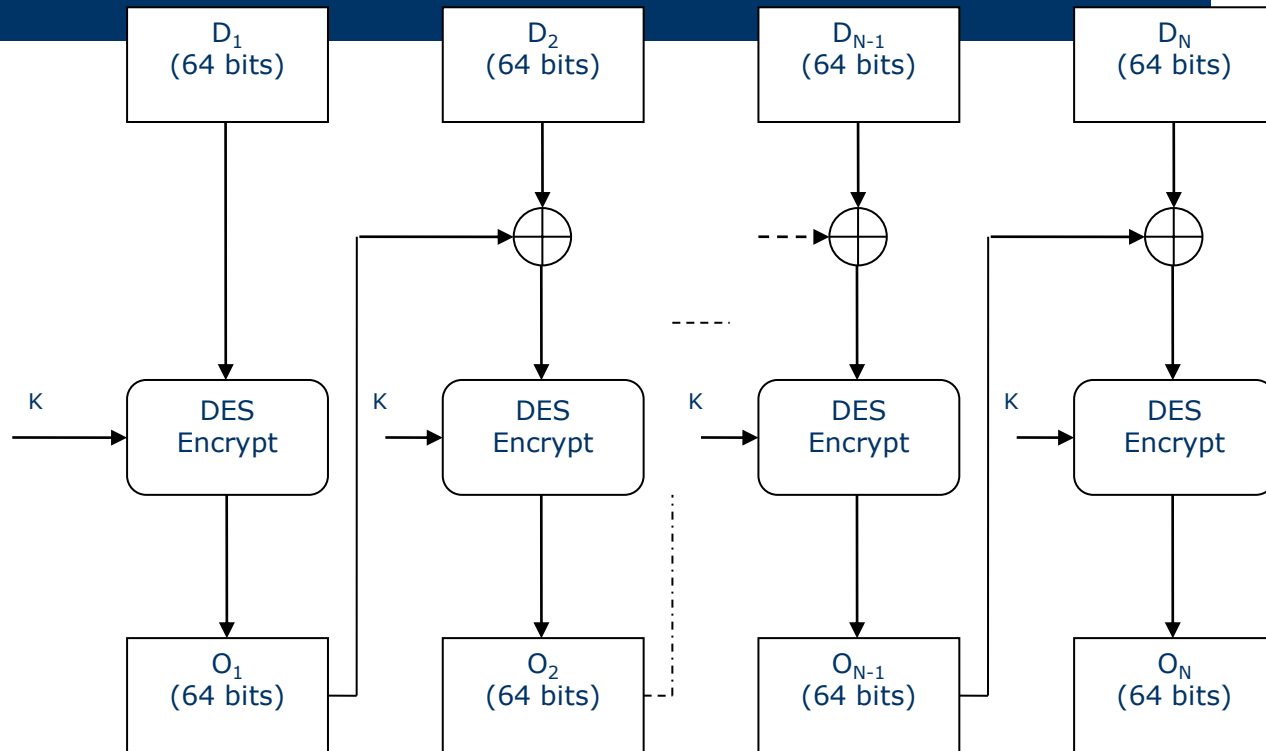
- Đánh giá sự an toàn của hàm mật mã CheckSum
 - Cần quan tâm đến các loại tấn công mà có thể được áp dụng
 - Từ đó trang bị cho thuật toán và đưa ra các yêu cầu cho hàm

- Giả sử đối phương biết được hàm CheckSum C nhưng không biết K.
- Hàm CheckSum nên có những đặc tính sau:
 - Nếu đối phương biết M và $C_K(M)$, hàm C nên làm cho đối phương không thể tìm được 2 bản tin M và M' mà $C_K(M)=C_K(M')$.

Đặc tính CheckSum

- Đặc tính (tiếp):
 - $C_K(M)$ nên có phân bố ngẫu nhiên đều khi tạo ra các bản tin M, M' khi đó xác suất $C_K(M)=C_K(M')$ là trong đó n là số bit của CheckSum \rightarrow ngăn tấn công vét cạn
 - Tính $M' = f(M)$. Ví dụ, hàm $f()$ có thể là hàm đảo ngược của 1 hay nhiều bit đã chỉ rõ. Trong trường hợp này xác suất $[C_K(M)=C_K(M')]=2^{-n}$
 \Rightarrow hàm băm tốt hơn

Mật mã CheckSum dựa trên DES



HASH FUNCTION

- Một hàm băm được tạo bởi hàm H có dạng :
$$h = H(M)$$
 - h : giá trị băm
 - M : là một biến (độ dài bản tin)
 - $H(M)$: hàm băm có giá trị hàm có độ dài cố định
- h được gắn vào bản tin tại lúc gửi khi thông điệp được coi là chính xác.

HASH FUNCTION

- Người nhận xác thực lại thông điệp bằng cách tính toán lại giá trị băm
- Bởi vì hàm băm chính nó không có bảo mật dẫn đến cần thêm một số phương tiện được yêu cầu thêm để bảo vệ các giá trị băm

Các yêu cầu cho hàm băm

- H có thể được áp dụng cho 1 khối dữ liệu (block data) với kích thước bất kỳ
- H tạo ra giá trị băm có độ dài cố định
- $H(x)$ có các quan hệ đơn giản để tính toán với x , để ta có thể triển khai được cả trên phần mềm và phần cứng trong thực tế
- Với bất kỳ mã **m** được cho, nó không thể làm việc tính toán được để tìm x sao cho $H(x) = m$
- Với bất kỳ block x đã cho, nó không thể làm việc tính toán được để tìm $y \neq x$ với $H(y) = H(x)$
- Không thể tìm được bất kỳ cặp (x, y) nào thoả mãn $H(x) = H(y)$ (Hàm $H()$ là hàm song ánh)

Các yêu cầu cho 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 cho 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).

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 \Rightarrow tạo ra 1 giá trị băm n bit

Hàm băm XOR

- Thực hiện phép XOR bit-by-bit
- Có thể biểu diễn như sau:

- $C_i = b_{i1} \oplus b_{i2} \oplus \dots \oplus b_{im}$

- Trong đó:

C_i : bit thứ i của mã băm ($i=1..n$)

m : Số Block n -bit của Input

b_{ij} : bit 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

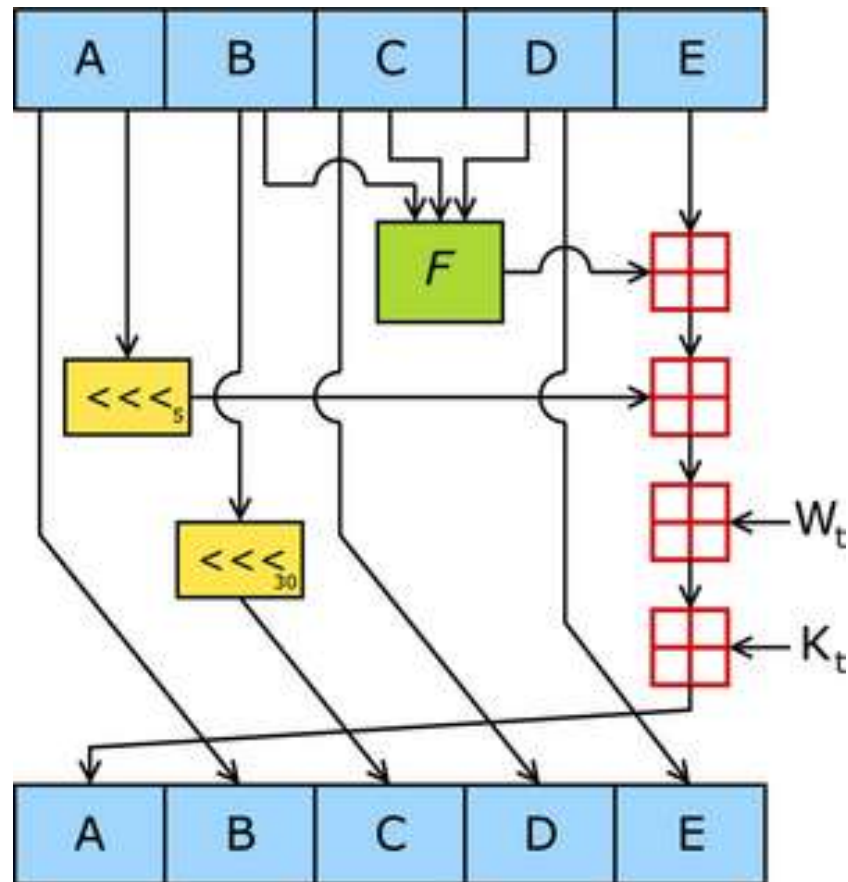
SHA-1 (Secure Hash Algorithm -1)

- 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 chỉ 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

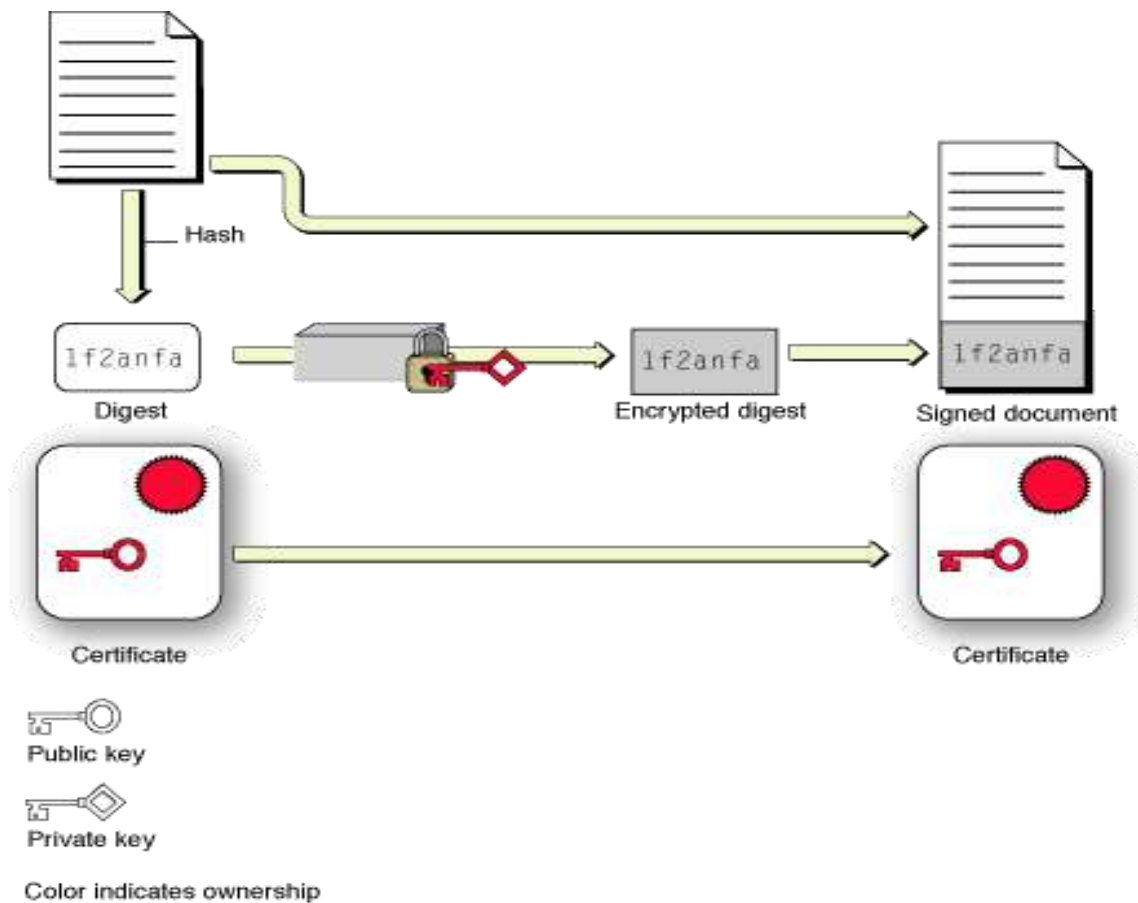
Chữ ký trực tiếp

- 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.

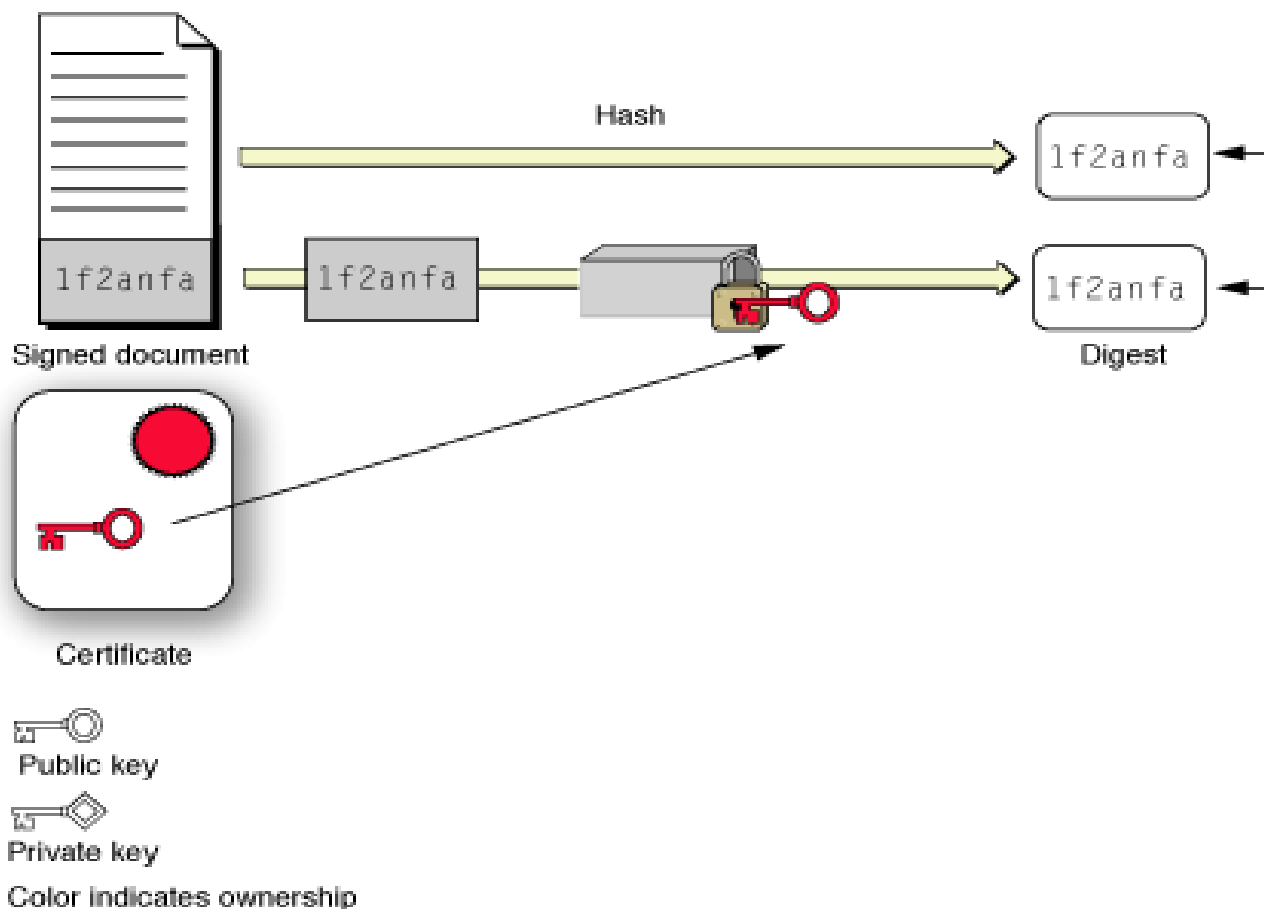
Chữ ký phân xử

- 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



Chứng thực chữ ký



Chứng chỉ số

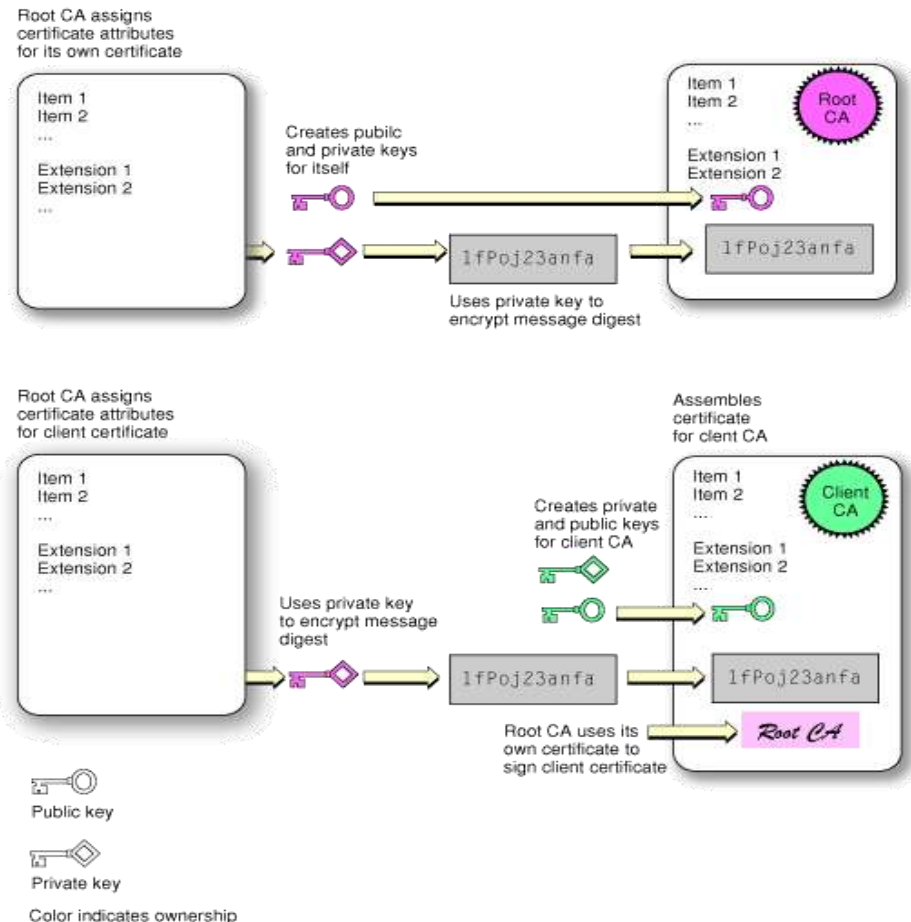
- Để chứng thực được chữ ký số bắt buộc người nhận phải có khoá chung của người gửi.
- Bản chất cặp khoá này không liên hệ với thuộc tính của người sử dụng → cần có cơ chế để liên kết chúng với người dùng → các chứng chỉ
- Các chứng chỉ được cung cấp bởi CA

Các thông tin trong chứng chỉ

- Phiên bản
- Số serial
- Nhà cung cấp chứng chỉ
- Người giữ chứng chỉ
- Thời gian hợp pháp của chứng chỉ
- Các thuộc tính
- Chữ ký điện tử của nhà cung cấp
- Khoá công khai của người sở hữu chứng chỉ
- Thuật toán băm dùng để tạo chữ ký.

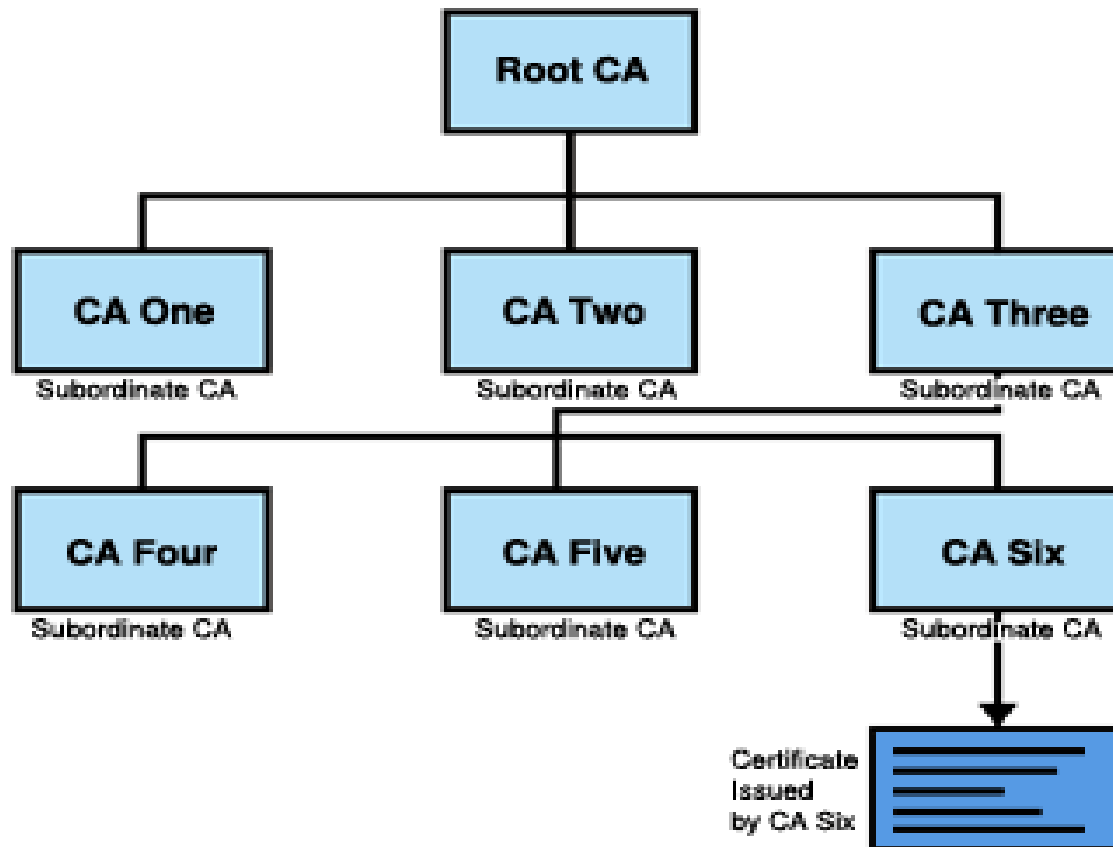
Tạo chứng chỉ

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

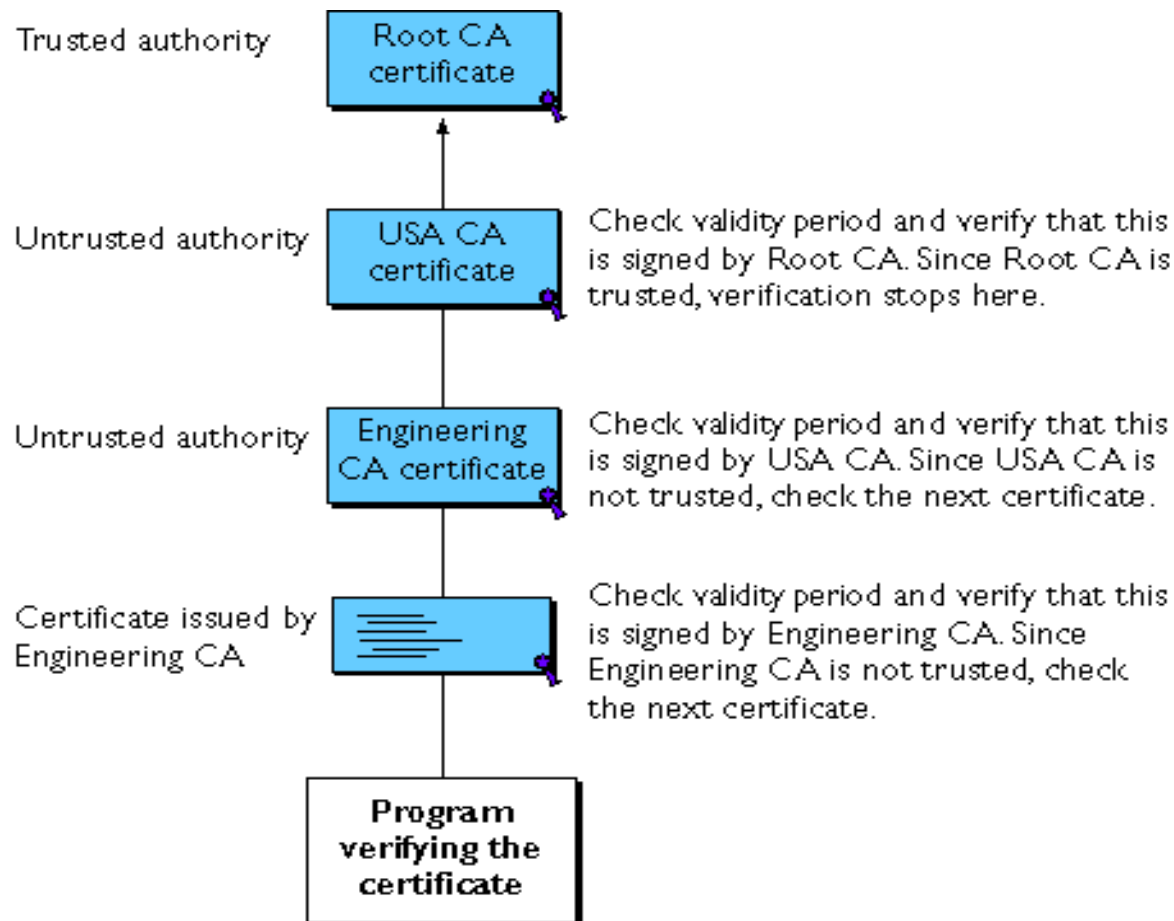


Cấu trúc phân cấp của CA

A Hierarchy of Certificate Authorities



Xác thực chuỗi chứng chỉ



Các giao thức xác thực

- Xác thực lẫn nhau
- Các phương pháp mã hoá đối xứng
- Phương pháp mã hoá khoá công khai

Xác thực lẫn nhau

- Tại đây, chúng ta chỉ xem xét vấn đề phân phối khoá
- Tồn tại 2 vấn đề :
 - Tính tin cậy : ngăn chặn hiện tượng giả mạo và thoả hiệp khoá phiên
 - Tính hợp thời: 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 đoán 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 đối xứng

- 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 dụng 1 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