Bài 4.
Quản lý và phân phối khóa

ONE LOVE. ONE FUTURE.

1

Nội dung

- Giới thiệu chung về giao thức mật mã
- Các giao thức trao đổi khóa đối xứng
- Các giao thức trao đổi khóa công khai
- Hạ tầng khóa công khai(PKI)

2

1. Giới thiệu chung về giao thức mật mã

ONE LOVE. ONE FUTURE.

3

3

Giao thức mật mã là gì?

- Chúng ta đã biết về "mật mã" và các ứng dụng của nó:
  - Bảo mật
  - Xác thực
- Nhưng chúng ta cần biết "Sử dụng mật mã như thế nào?"
  - Hệ mật mã an toàn chưa đủ để làm cho quá trình trao đổi thông tin an toàn
  - Cần phải tính đến các yếu tố, cá nhân tham gia không trung thực
- Giao thức là một chuỗi các bước thực hiện mà các bên phải thực hiện để hoàn thành một tác vụ nào đó.
  - Bao gồm cả quy cách biểu diễn thông tin trao đổi
- Giao thức mật mã: giao thức sử dụng các hệ mật mã để đạt được các mục tiêu an toàn bảo mật

4

Các đặc tính của giao thức mật mã

- Đặc tính của giao thức truyền thông:
  - Các bên tham gia phải hiểu về các bước thực hiện giao thức
  - Các bên phải đồng ý tuân thủ chặt chẽ các bước thực hiện
  - Không có sự nhập nhằng trong giao thức
  - Giao thức phải định nghĩa đầy đủ hành động cho mọi tình huống có thể
- ***Đặc tính riêng cho giao thức mật mã: Không bên nào thu được nhiều lợi ích hơn so với thiết kế của giao thức*** tối thiểu hóa quyền

Các yêu cầu của giao thức mật mã

kẻ tấn công có dữ liệu trong quá khứ nhưng ko thể tấn công vào các phiên trong tương lai

- Forward Secrecy: Dữ liệu của các phiên đã thực hiện trong quá khứ không thể được sử dụng để tấn công thỏa hiệp vào phiên trong tương lai

đảm bảo an toàn các phiên trong tương lai

- Backward Secrecy: Dữ liệu của các phiên đã thực hiện vẫn được đảm bảo an toàn nếu các phiên trong tương lai bị tấn công thỏa hiệp.

đảm bảo an toàn các phiên trong quá khứ

-> nếu phiên hiện tại bị tấn công -> chỉ phiên đó bị mất an toàn, ko ảnh hưởng đến các phiên khác

- Với giao thức phân phối khóa:
  - Perfect Forward Secrecy: Khóa ngắn hạn(short-term key) vẫn phải an toàn nếu trong tương lai khóa dài hạn (long-term key) không còn an toàn

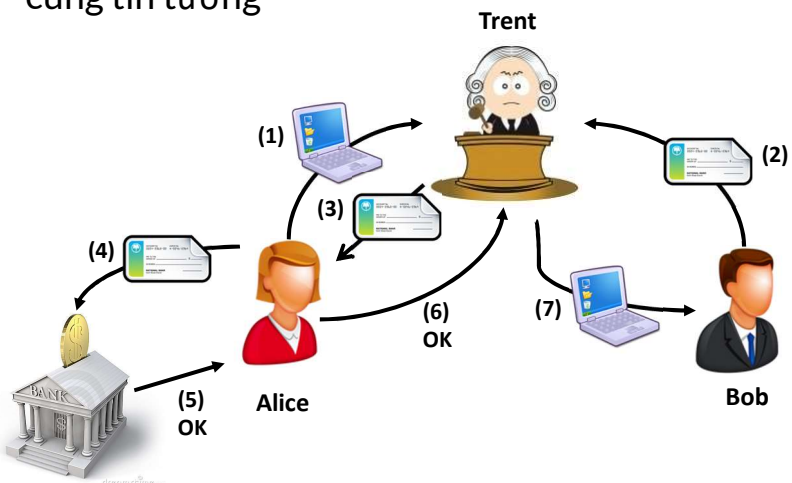
khóa ngắn hạn: dùng trong 1 hoặc vài phiên
khóa dài hạn: dùng trong nhiều phiên

Giao thức có trọng tài(Trusted arbitrator)

- Trọng tài là bên thứ 3 thỏa mãn:
  - Không có quyền lợi riêng trong giao thức
  - Không thiên vị
- Các bên cần tin tưởng vào trọng tài
  - Mọi thông tin từ trọng tài là đúng và tin cậy
  - Trọng tài luôn hoàn thành đầy đủ nhiệm vụ trong giao thức
- Ví dụ: Alice cần bán một chiếc máy tính cho Bob, người sẽ trả bằng séc
  - Alice muốn nhận tờ séc trước để kiểm tra
  - Bob muốn nhận máy tính trước khi giao séc

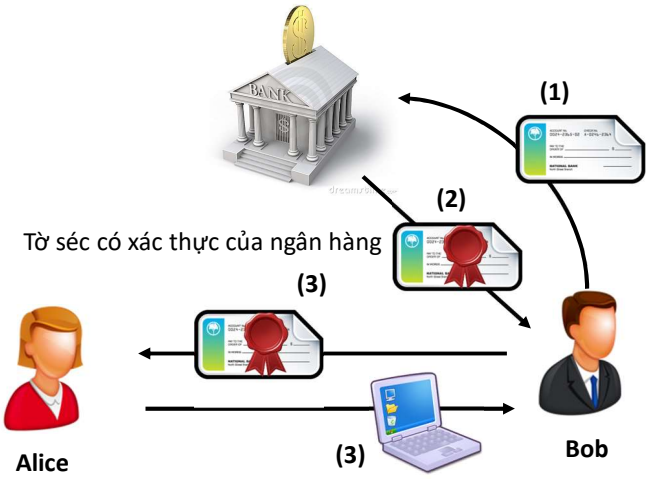
Giao thức có trọng tài – Ví dụ 1

- Alice và Bob tin tưởng vào Trent-Bên thứ 3 mà cả 2 cùng tin tưởng



Giao thức có trọng tài – Ví dụ 2

- Alice tin tưởng vào ngân hàng mà Bob ủy nhiệm



Giao thức sử dụng trọng tài

- Khi 2 bên đã không tin tưởng nhau, có thể đặt niềm tin vào bên thứ 3 không?
- Tăng chi phí
- Tăng trễ
- Trọng tài trở thành "cổ chai" trong hệ thống
- Trọng tài bị tấn công

Giao thức có người phân xử(*Adjudicated Protocols*)

- Chia giao thức có trọng tài thành 2 giao thức:
  - Giao thức không cần đến trọng tài, có thể thực hiện bất kỳ khi nào 2 bên muốn
  - Giao thức cần người phân xử: chỉ sử dụng khi có tranh chấp
- Hãy xem xét lại giao dịch trong ví dụ trên với giải pháp mới này!

(1) Alice và Bob thỏa thuận hợp đồng

(2) Hai bên cùng ký xác thực vào hợp đồng

(3) Thực hiện giao dịch

Nếu có tranh chấp:

(4) Alice trình bằng chứng cho người phán xử

(5) Bob trình bằng chứng cho người phán xử

Giao thức tự phân xử(*Self-Enforcing Protocols*)

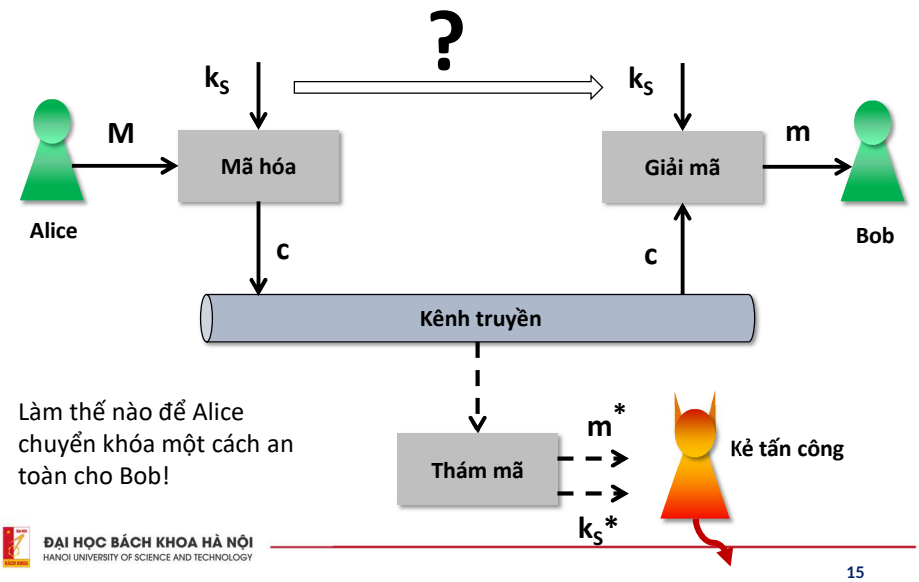
- Không cần đến bên thứ 3
- Giao thức có cơ chế để một bên có thể phát hiện sự gian lận của bên còn lại
- Thông thường giao thức là rất phức tạp -> thời gian dài
  - Ví dụ: Các giao thức trong mạng Bitcoin
- Không phải tình huống nào cũng có thể tìm ra giao thức như vậy

Các dạng tấn công vào giao thức mật mã

- Có thể lợi dụng các điểm yếu trong:
  - Hệ mật mã
  - Các bước thực hiện
- Tấn công thụ động: nghe trộm ko làm thay đổi luồng hd của giao thức
- Tấn công chủ động: can thiệp vào giao thức
  - Chèn thông điệp
  - Thay thế thông điệp
  - Sử dụng lại thông điệp
  - Giả mạo một trong các bên

2. Các giao thức phân phối khóa bí mật

Hãy xem lại sơ đồ bảo mật sử dụng mật mã KĐX



15

Giao thức phân phối khóa không tập trung

- Khóa chính: k_M đã được A và B chia sẻ an toàn
  - Làm thế nào?
  - Khóa chính được sử dụng để trao đổi khóa phiên K_S
- Khóa phiên k_S: sử dụng để mã hóa dữ liệu trao đổi
- Giao thức 1.1 -> cơ chế ngày làm vc đầu tiên
  (1) A → B: ID_A
  (2) B → A: $E(k_M, ID_B||k_S)$
- Giao thức này đã đủ an toàn chưa?
  - Tấn công nghe lén
  - Tấn công thay thế
  - Tấn công giả mạo
  - Tấn công phát lại ko chống đc -> ko đảm bảo đc tính chất forward secrecy
    Km bị lộ -> ko đảm bảo an toàn Ks -> ko đảm bảo tính chất PFS

16

Tấn công phát lại vào giao thức 1.1

- C là kẻ tấn công đã thu thập được khóa k_s^{old} và bản tin số (2) khi phân phối khóa này.
- C thay thế bản tin số (2) của phiên hiện tại bằng bản tin cũ:

B→C→A: $E(k_M, ID_B \,||\, k_s^{old})$

→A bị đánh lừa dùng khóa k_s^{old}

→A → B: $E(k_s^{old}, secret)$

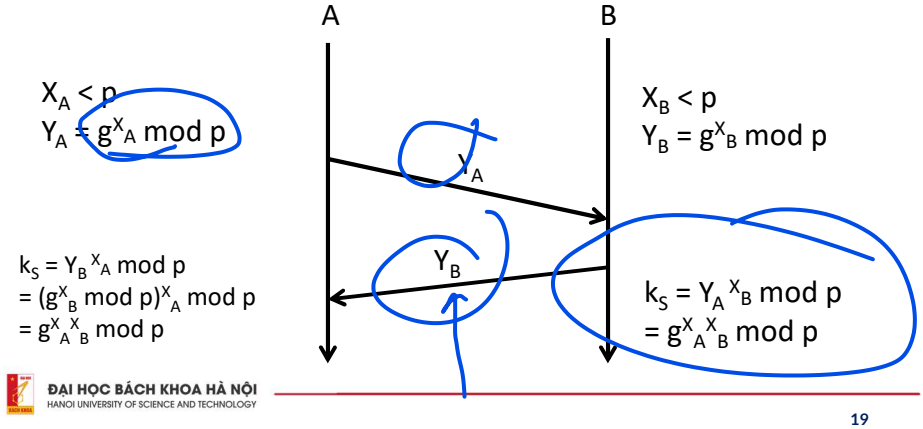
→A cần phải xác định khóa trong bản tin số (2) là khóa mới của phiên hiện tại mà B trao chuyển

Giao thức phân phối khóa không tập trung – Giao thức 1.2

- Sử dụng các yếu tố chống tấn công phát lại (replay attack)
  (1) A → B: $ID_A \,||\, N_1$
  (2) B → A: $E(k_M, ID_B \,||\, k_S \,||\, N_1 \,||\, N_2)$
  (3) A → B: A kiểm tra N_1 và gửi $E(k_S, N_2)$
  (4) B kiểm tra N_2
- E: Hàm mã hóa có xác thực
- N_1, N_2: Giá trị nonce (dùng 1 lần)
- Xem xét việc thỏa mãn các yêu cầu của giao thức mật mã
- Hạn chế của phân phối khóa không tập trung?

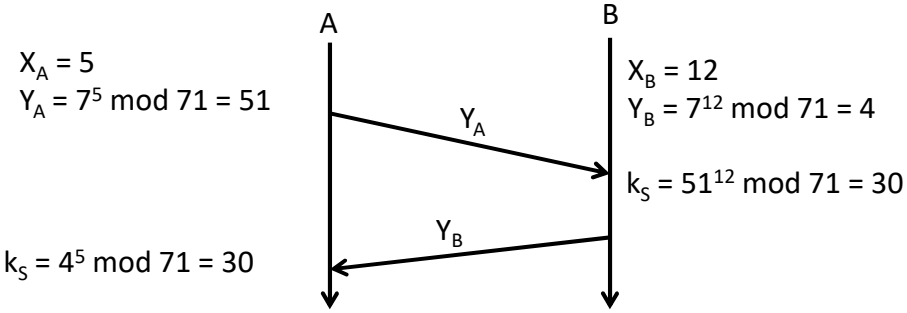
Sơ đồ trao đổi khóa Diffie-Hellman

- Alice và Bob cùng chia sẻ một khóa nhóm (p, g). công khai
  Trong đó
  - p là một số nguyên tố
  - 1< g < p thỏa mãn: $(g^i \bmod p) \neq (g^j \bmod p)\ \forall\ 1 \leq i \neq j < p$

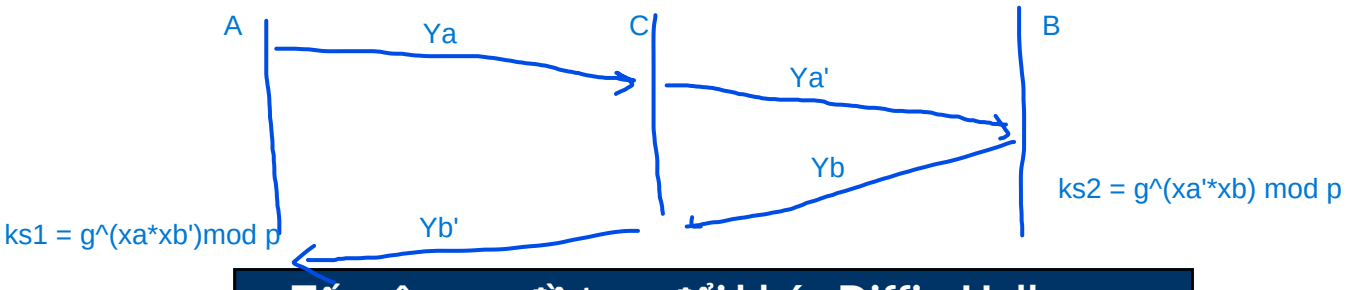


Ví dụ

- Khóa nhóm: p = 71, g= 7
  - Hãy tự kiểm tra điều kiện thỏa mãn của g
- A chọn $X_A = 5$, tính $Y_A = 7^5 \bmod 71 = 51$

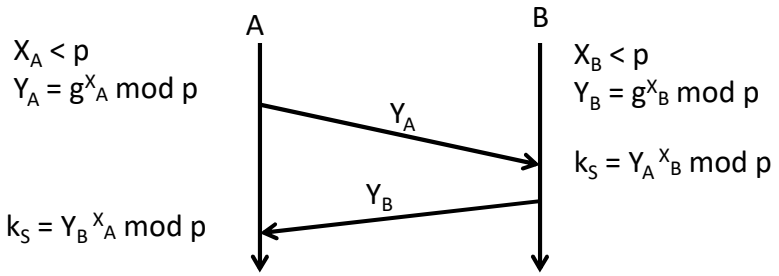


Sơ đồ này đạt những yêu cầu nào?



Tấn công sơ đồ trao đổi khóa Diffie-Hellman

- Nhắc lại sơ đồ: không đảm tính toàn vẹn, xác thực



- Kịch bản tấn công man-in-the-middle
  - C sinh 2 cặp khóa (X'_A, Y'_A) và (X'_B, Y'_B)
  - Tráo khóa Y_A bằng Y'_A, Y_B bằng Y'_B
  - Hãy suy luận xem tại sao C có thể biết được mọi thông tin A và B trao đổi với nhau

Giao thức phân phối khóa tập trung

- Sử dụng bên thứ 3 được tin cậy – KDC (Key Distribution Centre):
  - Sinh khóa bí mật k_S
  - Phân phối k_S tới A và B
- A và KDC đã chia sẻ một khóa bí mật k_A, B và KDC đã chia sẻ một khóa bí mật k_B
  - Làm thế nào? sử dụng cơ chế ngày làm vc đầu tiên

Giao thức phân phối khóa tập trung-Giao thức 2.1

(1) A → KDC: ID_A || ID_B

(2) KDC → A: E(k_A, k_S || ID_A || ID_B || E(k_B, ID_A || k_S))

(3) A giải mã (2), thu được k_S

(4) A → B: E(k_B, ID_A || k_S). B giải mã, thu được k_S

(5) A ↔ B: E(k_S, Data)

- E: Mã hóa có xác thực
- Hãy xem xét tính an toàn của giao thức này?
  - Tấn công nghe lén
  - Tấn công thay thế
  - Tấn công giả mạo
  - Tấn công phát lại ko có cơ chế xác định phiên -> ko chống đc tấn công phát lại

x forward secrecy:
backward: đảm bảo nếu ks là ngẫu nhiên
x PFS

Giao thức 2.2 (Needham-Schroeder)

(1) A → KDC: ID_A || ID_B || N_1

(2) KDC → A: E(k_A, k_S || ID_A || ID_B || N_1 || E(k_B, ID_A || k_S))

(3) A giải mã, kiểm tra N_1 thu được k_S

(4) A → B: E(k_B, ID_A || k_S) ← B giải mã, thu được k_S

(5) B → A: E(k_S, N_2) ← A giải mã, có được N_2, tính f(N_2)

(6) A → B: E(k_S, f(N_2)) ← B giải mã kiểm tra f(N_2)

(7) A ↔ B: E(k_S, Data)

E: mã hóa có xác thực

N_1, N_2: giá trị dùng 1 lần (nonce)

f(x): hàm biến đổi sao cho f(x) ≠ x với mọi x(Tại sao cần?)

- Hãy xem xét lại tính an toàn của giao thức này!

chống lại tấn công phản xạ: bên tấn công phản hồi lại chính bản tin thách thức

Giả định tấn công vào (4)

- (4) A → C → B: $E(k_B, ID_A \,||\, k_S^{old})$

B giải mã và có khóa k_s^{old}

(5) B → A: $E(k_s^{old}, N_2)$

A giải mã $D(k_s, E(k_s^{old}, N_2)) = N'_2 \neq N_2$

(6)A→B: $E(k_s, f(N'_2))$

B giải mã $D(k_s^{old}, E(k_s, f(N'_2))) \neq f(N'_2) \neq f(N_2)$

→ B từ chối phiên truyền tin

Giao thức 2.3 (Denning)

(1) A → KDC: $ID_A \,||\, ID_B$

(2) KDC → A: $E(k_A, k_S \,||\, ID_A \,||\, ID_B \,||\, \mathbf{T} \,||\, E(k_B, ID_A \,||\, k_S \,||\, \mathbf{T}))$

(3) A giải mã bản tin (2), kiểm tra T, thu được k_S

(4) A → B: $E(k_B, ID_A \,||\, k_S \,||\, \mathbf{T})$ ← B giải mã, kiểm tra T

(5) B → A: $E(k_S, N_1)$

(6) A → B: $E(k_S, f(N_1))$ ← B giải mã, kiểm tra N_1

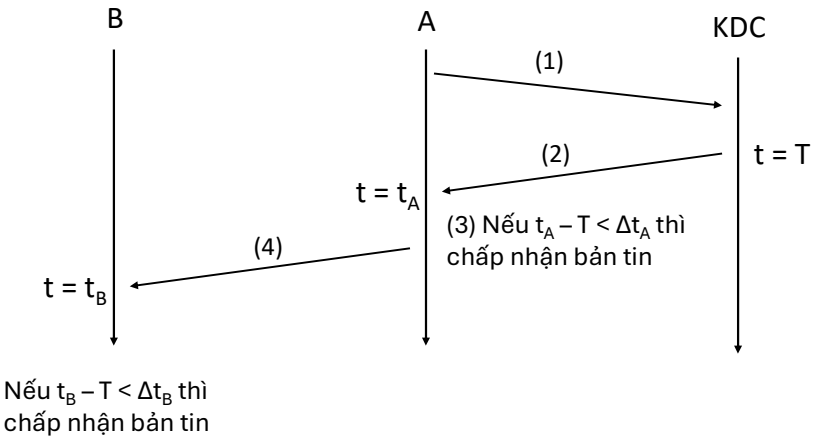
(7) A ↔ B: $E(k_S, Data)$

E: mã hóa có xác thực

T: nhãn thời gian (time stamp) là thời điểm KDC phát bản tin

- Kiểm tra tính an toàn của sơ đồ này khi mất đồng bộ đồng hồ của các bên

Giao thức 2.3 (Denning) – Kiểm tra T



Nếu $t_B - T < \Delta t_B$ thì chấp nhận bản tin

Giao thức 2.4 (Kehne)

(1) A → B: ID_A || N_A

(2) B → KDC: ID_B || N_B || E(k_B, ID_A || N_A || T_B)

(3) KDC → A: E(k_A, ID_B || N_A || k_S) || E(k_B, ID_A || k_S || T_B) || N_B

A giải mã, kiểm tra N_A → chấp nhận k_S nếu N_A hợp lệ

(4) A →B: E(k_B, ID_A || k_S || T_B) || E(k_S, N_B)

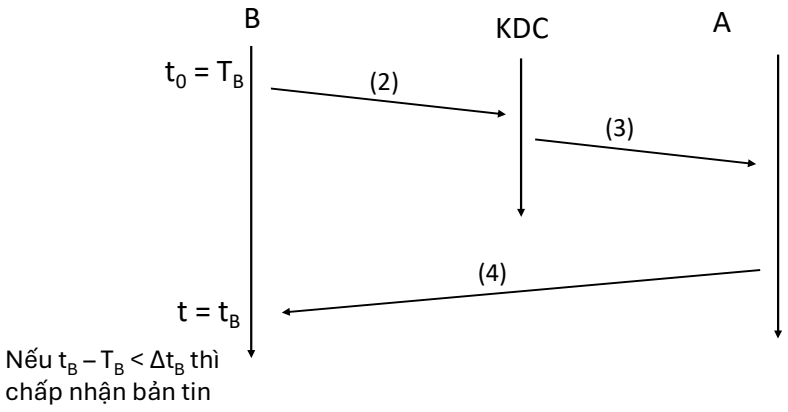
B giải mã, kiểm tra T_B → chấp nhận k_S nếu T_B hợp lệ

→ giải mã với k_S, biết $N_B^{(4)}$ và so sánh với $N_B^{(2)}$ → xác nhận được khóa k_S mà A dùng giống B đang có

T_B: Thời điểm B gửi bản tin số (2)

- Vì sao việc sử dụng nhãn thời gian T_B của B tốt hơn nhãn thời gian T của KDC trong giao thức 2.3

Giao thức 2.4 (Kehne) – Kiểm tra T_B



Nếu $t_B - T_B < \Delta t_B$ thì chấp nhận bản tin

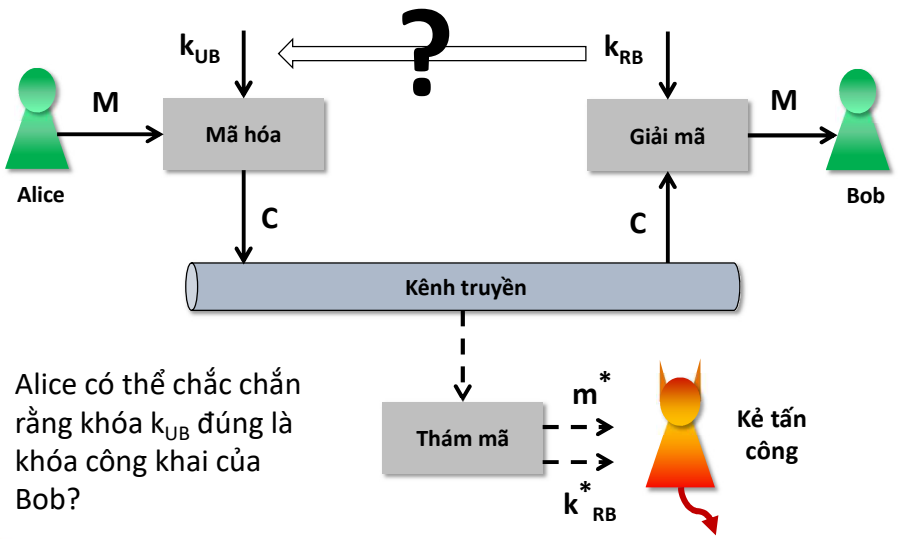
29

3. Các giao thức phân phối khóa công khai

ONE LOVE. ONE FUTURE.

30

Hãy xem lại sơ đồ bảo mật sử dụng mật mã KCK



31

Phân phối khóa không tập trung

- A và B gặp nhau trong ngày làm việc đầu tiên và trao đổi khóa k_{UA} và k_{UB} trực tiếp
- Làm cách nào để phân khóa công khai của C tới A và B?
- Nếu A và C đã trao đổi khóa công khai một cách tin cậy:
  - Trao chuyển khóa dựa trên tin cậy ngang hàng → danh sách tin cậy (Trusted List)
  - A→B: $k_{UC} \,||\, sig(k_{RA}, ID_C \,||\, k_{UC})$

32

Phân phối khóa tập trung: Giao thức 4.1

- Sử dụng bên thứ 3 tin cậy – PKA (Public Key Authority)
  - Có cặp khóa (k_{UPKA}, k_{RPKA}) sử dụng trong lâu dài
- Ngày làm việc đầu tiên:
  - PKA Nhận các khóa công khai k_{UA} của A và k_{UB} của B
  - A và B nhận khóa công khai k_{UPKA} của PKA
- Giao thức 4.1

ko cần đảm bảo tính bí mật nhưng phải đảm bảo tính xác thực

(1) A → PKA: ID_A || ID_B

(2) -> (2)old

(2) PKA → A: ID_B || k_{UB} || sig(k_{RPKA}, ID_B || k_{UB})

(3) A → B: E(k_{UB}, N_1) N1: ngẫu nhiên sử dụng 1 lần

nhạy cảm trc tấn công phát lại

(4) B → PKA: ID_B || ID_A

(5) -> (5)old

(5) PKA → B: ID_A || kU_A || sig(k_{RPKA}, ID_A || k_{UA})

(6) B → A: E(k_{UA}, N_1)

sig(): hàm tạo chữ ký số

-> kiểm tra N1 nếu đúng -> có khóa công khai hợp lệ

Giao thức 4.1

Giao thức 4.2

(1) A → PKA: ID_A || ID_B || T_1 thời điểm A gửi yêu cầu

(2) PKA → A: ID_B || k_{UB} || T_1 || sig(k_{RPKA}, ID_B || k_{UB} || T_1) kiểm tra chữ ký và nhãn thời gian

(3) A → B: E(k_{UB}, N_1)

(4) B → PKA: ID_B || ID_A || T_2

(5) PKA → B: ID_A || k_{UA} || T_2 || sig(k_{RPKA}, ID_A || k_{UA} || T_2)

(6) B → A: E(k_{UA}, N_1)

T_1, T_2: nhãn thời gian chống tấn công phát lại

sig(): hàm tạo chữ ký số -> xác thực

- Giao thức này có hạn chế gì?

  - tải trên PKA cao (mọi truy vấn khóa đều đến PKA)

Giao thức 4.2

Giao thức 4.3

- Bên thứ 3 được tin cậy – CA(Certificate Authority)
  - Có cặp khóa (k_{UCA}, k_{RCA})
  - Phát hành chứng chỉ số cho khóa công khai của các bên có dạng

Cert = ID || k_U || Time || sig(k_{RCA}, ID || k_U || Time)

ID: định danh của thực thể

k_U: khóa công khai của thực thể đã được đăng ký tại CA

Time: Thời hạn sử dụng khóa công khai. Thông thường có thời điểm bắt đầu có hiệu lực và thời điểm hết hiệu lực.

sig(): hàm tạo chữ ký số

37

A thay đổi sao -> làm sao để B biết ?

Giao thức 4.3 (tiếp)

- Cấp phát chứng chỉ số cho A

(1) A → CA: ID_A || k_{UA} || $Time_A$

(2) CA → A: $Cert_A$= ID_A || k_{UA} || $Time_A$ || sig(k_{RCA}, ID_A || k_{UA} || $Time_A$)

- Cấp phát chứng chỉ số cho B

(1) B → CA: ID_B || k_{UB} || $Time_B$

(2) CA → B: $Cert_B$= ID_B || k_{UB} || $Time_B$ || sig(k_{RCA}, ID_B || k_{UB} || $Time_B$)

- A và B trao đổi chứng chỉ số

(5) A → B: $Cert_A$

(6) B → A: $Cert_B$

38

Giao thức 4.3 (tiếp)

Hạ tầng khóa công khai PKI

- Public Key Infrastructure
- Hệ thống bao gồm phần cứng, phần mềm, chính sách, thủ tục cần thiết để tạo, quản lý và lưu trữ, phân phối và thu hồi các chứng chỉ số
- Chứng chỉ số: văn bản điện tử chứng thực khóa công khai
- Các thành phần:
  - RA(Registration Authority): Chứng thực thông tin đăng ký
  - CA(Certification Authority): Phát hành và quản lý chứng chỉ số
  - CR(Certificate Reposiroty): Lưu trữ, chứng thực chứng chỉ số
  - EE(End-Entity): đối tượng sử dụng chứng chỉ số

thành phần chính

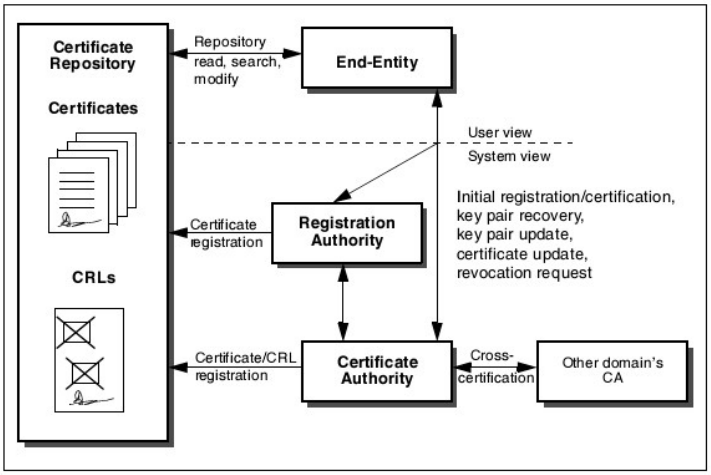
module tiếp nhận đk

phát hành và quản lý chứng chỉ cho khóa công khai

kho lưu giữ chứng trỉ đã phát hành

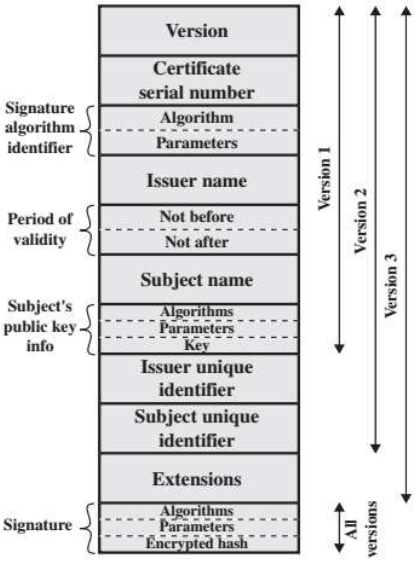
thực thể đầu cuối

Các thành phần của PKI



41

Chứng chỉ số X.509



42

Chứng chỉ số X.509

- Version: phiên bản của chứng chỉ số
- Số serial của chứng chỉ số (tối đa 20 byte) định danh chứng trỉ
- Algorithm: Thuật toán chữ ký số được CA sử dụng để ký
- Issuer: Thông tin cơ quan cấp chứng chỉ số định danh CA
  - C: Quốc gia
  - CN: Tên giao dịch của CA
  - DN: Tên định danh
  - O: Tên tổ chức phát hành
  - ST: Tên đơn vị hành chính trực thuộc trung ương
- Validity: Thời gian hiệu lực của chứng chỉ số
  - Not Before: Ngày bắt đầu có hiệu lực
  - Not after: Ngày hết hiệu lực

Chứng chỉ số X.509(tiếp)

- Subject: Thông tin người được cấp chứng thư
  - Các trường con tương tự thông tin tổ chức phát hành
- Subject's Public Key Information: Thông tin khóa công khai
  - Algorithm: Thuật toán tạo khóa
  - Public Key: Giá trị khóa
- Signature: chữ ký số của cơ quan cấp chứng chỉ số
- Issuer UID: định danh của cơ quan cấp chứng chỉ số
- Subject UID: định danh của người được cấp chứng thư định danh toàn cầu (ko có 2 ng trùng nhau trong toàn bộ mạng internet)
- Extensions: Các trường mở rộng khác

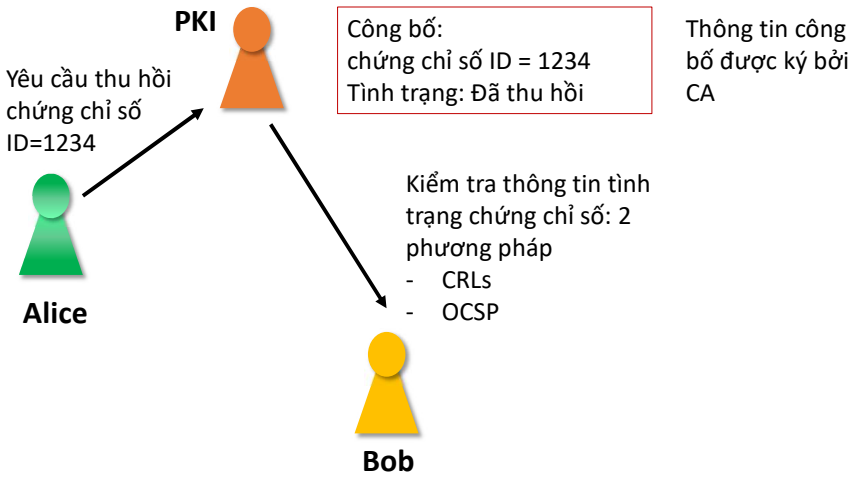
Xác thực chứng chỉ số

Chứng chỉ số cần được kiểm tra tính tin cậy:

- Kiểm tra tên thực thể sử dụng có khớp với tên đăng ký trong chứng chỉ số
- Kiểm tra hạn sử dụng của chứng chỉ số
- Kiểm tra tính tin cậy của CA phát hành chứng chỉ số
- Kiểm tra trạng thái thu hồi chứng chỉ số giải quyết việc A thay đổi khóa
- Kiểm tra chữ ký trên chứng chỉ số để đảm bảo chứng thư không bị sửa đổi, làm giả

Thu hồi chứng chỉ số

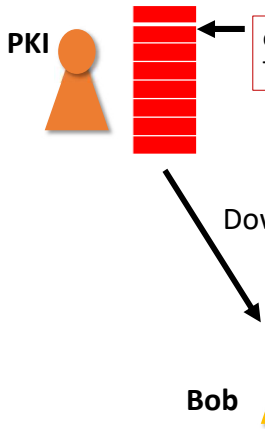
- Thực hiện khi khóa của người dùng mất an toàn



Certificate Revocation List

CRLs

- PKI công bố danh sách chứng chỉ số bị thu hồi. Danh sách này được ký bởi CA



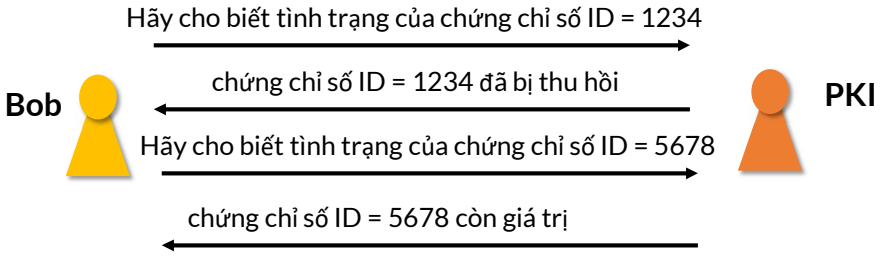
khó có thể cập nhật liên tục do chính sách của PKI

Hạn chế:
- Kích thước CRLs lớn
- Không cập nhập liên tục

tần suất cập nhật chứng trỉ số phụ thuộc vào yêu cầu hệ thống

OCSP

- Dịch vụ kiểm tra trạng thái chứng chỉ số trực tuyến(Online Certificate Status Protocol)



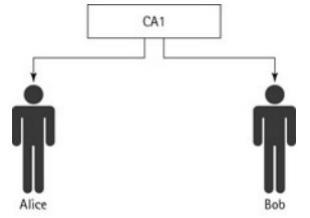
Thông điệp trả lời từ PKI được ký bởi CA

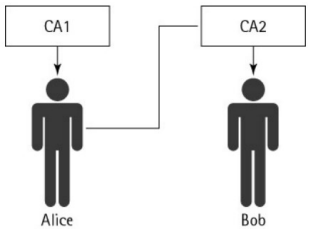
Kiến trúc PKI

- Kiến trúc PKI rất đa dạng, tương ứng theo mô hình hoạt động của mỗi tổ chức
- Các kiến trúc PKI sau được phân loại dựa trên số lượng CA, tổ chức và mối quan hệ giữa chúng:
  - Kiến trúc đơn CA (Single CA)
  - Kiến trúc PKI xí nghiệp (Enterprise PKI)
  - Kiến trúc PKI lai (Hybrid PKI)

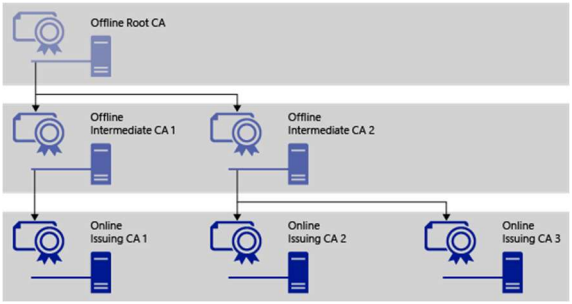
Kiến trúc đơn CA

- Chỉ sử dụng 1 CA trong hệ thống PKI
- Đơn giản, phù hợp với hệ thống nhỏ
- Không có khả năng mở rộng
- Mô hình danh sách tin cậy: 2 thực thể sử dụng chứng chỉ số được phát hành bởi 2 CA khác nhau
  - Mỗi CA có danh sách các CA mà nó tin cậy
  - Mỗi CA phải nằm trong danh sách tin cậy của CA còn lại
  - Hạn chế: Luôn đòi hỏi phải đồng bộ. Ví dụ: 1 CA ngừng hoạt động



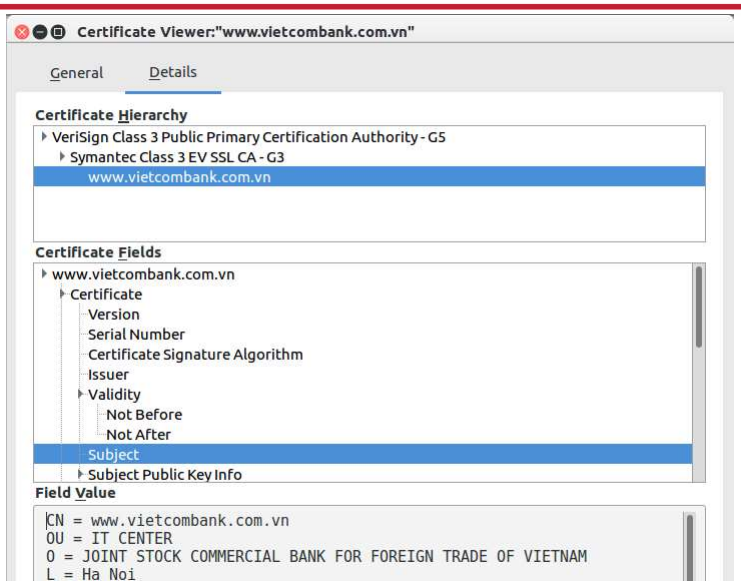


Kiến trúc PKI phân cấp



- Mỗi CA chứng thực cho tất cả các CA cấp dưới của nó
- Dễ dàng mở rộng
- Yêu cầu: Root CA cần được giữ an toàn tuyệt đối(thông thường Root CA luôn nằm ở phân vùng mạng offline)

Chứng chỉ số trong kiến trúc PKI phân cấp



Chuỗi xác thực

- Một chứng thư được phát hành bởi hệ thống PKI phân cấp cần được chứng thực theo một chuỗi hướng từ nút gốc tới nút lá trong cây phân cấp
- Ví dụ: Một chứng thư trong kiến trúc phân cấp

"I'm because I say so!"

"I'm because says so"

"I'm because says so"

Chuỗi xác thực

"I'm because I say so!"

"I'm because says so"

"I'm because says so"

Chuỗi xác thực từ chối chứng chỉ số nếu có bất kỳ bước nào cho kết quả xác thực thất bại

Kết luận

- Hệ thống có nguy cơ mất an toàn ngay cả khi chúng ta sử dụng hệ mật mã tốt nếu không có một giao thức quản lý và phân phối khóa an toàn
- Hãy sử dụng các giao thức tiêu chuẩn: IPSec, TLS, IEEE802.11x, Keberos,…
- Mật mã phải gắn liền với xác thực

Một số lưu ý

- Đảm bảo tính bí mật:
  - Khóa bí mật
  - Khóa cá nhân
  - Các giá trị chia sẻ bí mật khác
- Đảm bảo tính toàn vẹn, xác thực:
  - Khóa bí mật
  - Khóa công khai
  - Thông tin sinh khóa
- Kiểm tra tính hợp lệ của các tham số nhóm
- Kiểm tra tính hợp lệ của khóa công khai
- Kiểm tra quyền sở hữu khóa cá nhân

Một số lưu ý

- Không kết thúc ngay giao thức khi có 1 lỗi xảy ra
  - Làm chậm thông báo lỗi
- Chỉ sử dụng các giao thức tiêu chuẩn
- Thông báo lỗi không nêu cụ thể nguyên nhân lỗi
- Không sử dụng khóa giống nhau cho cả 2 chiều truyền tin
- Không sử dụng khóa giống nhau cho 2 mục đích mã mật và xác thực
- Cần chống lại tấn công vào quá trình thỏa thuận của giao thức

MS Point-to-Point Encryption

- Sử dụng mật mã RC4
- Hoạt động của giao thức