Chương 7

GIAO THỨC BẢO MẬT VÀ PHÂN PHỐI KHÓA

Giáo viên: Lê Quốc Anh

Nội dung

1. Phân phối khóa đối xứng
2. Phân phối khóa trong hệ mã khóa công khai
3. Phân phối khóa mật sử dụng mật mã khóa công khai
4. Thỏa thuận trao đổi khóa
    1. Mô hình trao đổi khóa Diffie-Hellman
5. Phân phối khóa công khai
    1. Chứng chỉ X509

1. Phân phối khóa đối xứng

- Mã hóa khóa đối xứng hiệu quả hơn mã hóa khóa bất đối xứng đối với việc mã hóa các thông điệp lớn. Tuy nhiên mã hóa khóa đối xứng cần một khóa chia sẻ giữa hai tổ chức.
- Một người cần trao đổi thông điệp bảo mật với N người, thì người đó cần N khóa khác nhau. Vậy N người giao tiếp với N người khác thì cần tổng số là N*(N-1) khóa

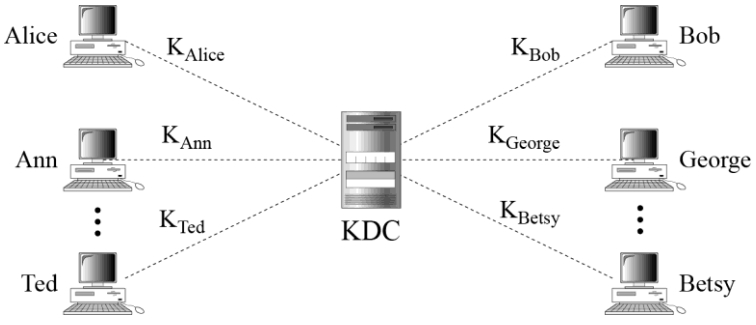
→ số khóa không chỉ là vấn đề, mà phân phối khóa là một vấn đề khác.

→Độ tin cậy của một hệ thống mật mã phụ thuộc vào công nghệ phân phối khóa (*key distribution technique*).

Slide 3

Key-Distribution Center: KDC

- Để giảm số lượng khóa, mỗi người sẽ thiết lập một khóa bí mật chia sẻ với KDC



- Làm thế nào để Alice có thể gửi một thông điệp bảo mật tới Bob

Slide 4

Key-Distribution Center: KDC

- Quá trình xử lý như sau:

1. Alice gửi 1 yêu cầu đến KDC để nói rằng cô ta cần một khóa phiên (session secret key) giữa cô ta và Bob.
2. KDC thông báo với Bob về yêu cầu của Alice
3. Nếu Bob đồng ý, một session key được tạo giữa 2 bên.

- Khóa bí mật này được dùng để chứng thực Alice và Bob với KDC và ngăn chặn Eve giả mạo một trong hai.

Slide 5

Key-Distribution Center: KDC

Flat Multiple KDCs

- Khi số lượng người dùng KDC tăng, hệ thống trở nên khó quản lý và một bottleneck sẽ xảy ra.

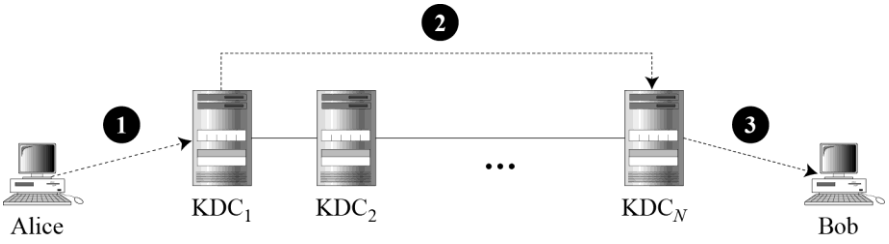
→ chúng ta có nhiều KDCs, chia thành các domain. Mỗi domain có thể có một hoặc nhiều KDCs

- Alice muốn gửi thông điệp bí mật tới Bob, mà Bob thuộc vào domain khác, thì Alice liên lạc với KDC của cô ta mà trong đó tiếp tục liên lạc với KDC trong domain của Bob.
- Hai KDCs như vậy thì được gọi là Flat multiple KDCs

Slide 6

Key-Distribution Center: KDC

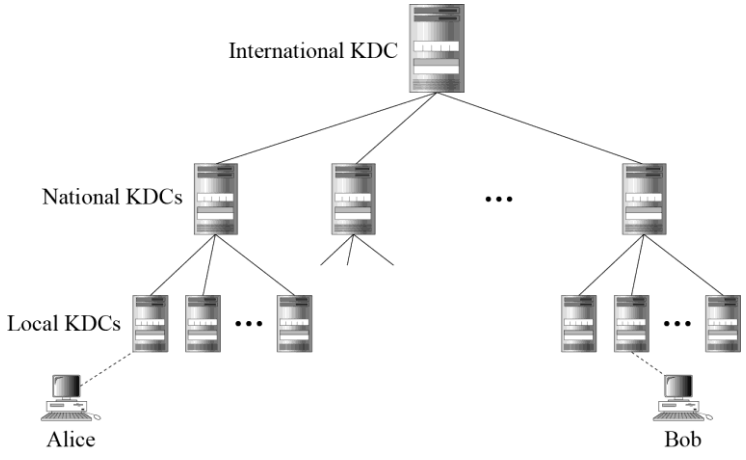
- **Flat Multiple KDCs**



Slide 7

Key-Distribution Center: KDC

- **Hierarchical Multiple KDCs**



Slide 8

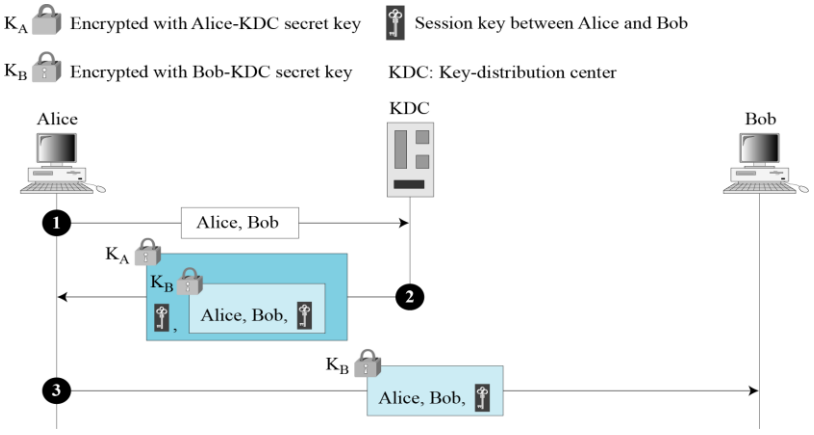
Khóa phiên (Session Keys)

- KDC tạo khóa bí mật cho mỗi thành viên, khóa bí mật này chỉ có thể dùng giữa thành viên và KDC, chứ không dùng giữa hai thành viên
- Nếu muốn dùng giữa hai thành viên, KDC tạo một ***session key*** giữa hai thành viên, sử dụng khóa của họ với trung tâm.
- ***Khóa phiên giữa hai thanh viên chỉ được dùng một lần*** (sau giao tiếp kết thúc thì khóa phiên cũng không còn tác dụng)

Slide 9

Khóa phiên (Session Keys)

- Một giao thức đơn giản sử dụng một KDC



- Giao thức này có thể bị tấn công phát lại ở bước 3

Slide 10

Khóa phiên (Session Keys)

1. Alice → KDC: $\{ID_A||ID_B\}$
2. KDC → Alice: $\{E(K_S||E(ID_A||ID_B||K_S,K_B),K_A\}$
3. Alice → Bob: $\{E(ID_A||ID_B||K_S,K_B)\}$

- Tấn công xen giữa (**Man in the middle**): https://vi.wikipedia.org/wiki/T%E1%BA%A5n_c%C3%B4ng_xen_gi%E1%BB%AFa
- Giao thức truyền thông = Giao thức trong đó các bước thực hiện là trao đổi thông tin
- Giao thức mật mã = Giao thức truyền thông + Mật mã học

Slide 11

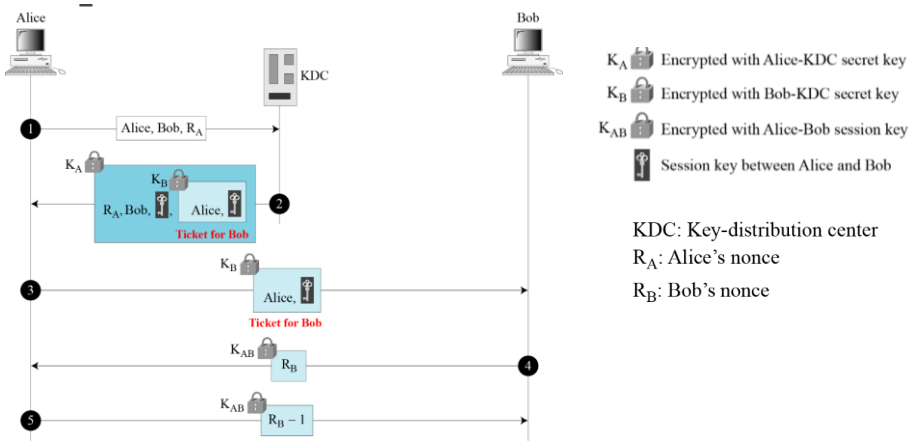
Khóa phiên (Session Keys)

- Một số vấn đề gặp phải:
  - Tại bước 1: Khi Alice gửi yêu cầu tạo khóa phiên tới KDC có thể bị tấn công "**Man in the middle**"
  - Tại bước thứ 3 kể xấu tấn công Bob theo hình thức phát lại thông điệp "**Replay attack**"

Slide 12

Khóa phiên (Session Keys)

- Giao thức Needham-Schroeder (nền tảng của nhiều giao thức khác)



Khóa phiên (Session Keys)

1. Alice → KDC: $\{ID_A||ID_B||R_A\}$
2. KDC → Alice: $\{E(R_A || ID_B || K_S|| E(ID_A||K_S,K_B),K_A\}$
3. Alice → Bob: $\{E(ID_A||K_S,K_B)\}$
4. Bob → Alice: $\{E(R_B ,K_s)\}$
5. Alice → Bob: $\{E(R_B-1 ,K_s)\}$

Khóa phiên (Session Keys)

- Một số vấn đề gặp phải:
  - k_S là 1 khóa phiên, và chỉ có hiệu lực trong phiên làm việc hiện tại; khi đã kết thúc phiên làm việc, thì k_S sẽ mất hiệu lực và vô giá trị cho phiên làm việc tiếp theo.
  - Nếu một người có ý đồ xấu lấy được toàn bộ thông tin trao đổi của Alice và Bob trước đây và có k_S thì có thể tấn công theo phương thức phát lại thông điệp tại bước 3. Khi đã có k_S rồi, Eve này chắc chắn sẽ **đáp ứng** được **thách thức** mà Bob đặt ra ở bước 4 của giao thức (**Challenge - Response protocol**)

    3. Alice → Bob: $\{E(ID_A||K_S,K_B)\}$
    4. Bob → Alice: $\{E(R_B ,K_s)\}$
    5. Alice → Bob: $\{E(R_B\text{-}1 ,K_s)\}$

Slide 15

Giao thức Needham - Schroeder dạng cải tiến.

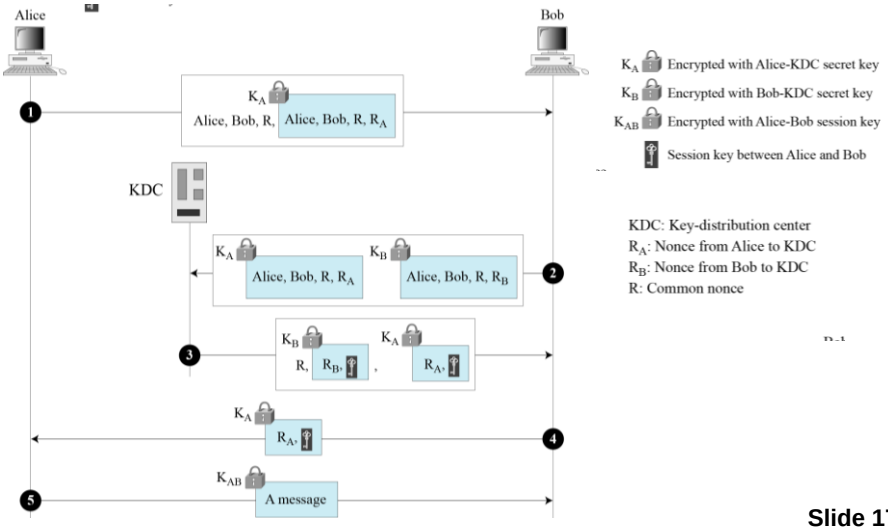
1. Alice → KDC: $\{ID_A||ID_B||R_A\}$
2. KDC → Alice: $\{E(R_A||ID_B||K_S||E(ID_A||\mathbf{T}||K_S,K_B),K_A\}$
3. Alice → Bob: $\{E(ID_A||\mathbf{T}||K_S,K_B)\}$
4. Bob → Alice: $\{E(R_B ,K_s)\}$
5. Alice → Bob: $\{E(R_B\text{-}1 ,K_s)\}$

- Tại bước 2 của giao thức cải tiến KDC đặt thêm một timestamp trong hộp mã hóa.
- Tại bước 3: Alice sẽ chuyển hộp con đó cho Bob. Bob sẽ mở ra và đối chiếu với đồng hồ hiện tại của mình. Nếu sự chênh lệch thời gian vượt quá ngưỡng cho phép, Bob sẽ hiểu rằng đã có kẻ Replay attack và sẽ reject.

Slide 16

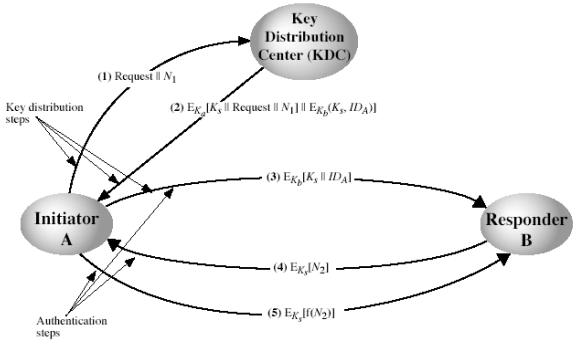
Khóa phiên (Session Keys)

- Giao thức Otway-Rees



Slide 17

Các giả thiết



- Kịch bản giả thiết rằng mỗi đầu cuối chia xẻ một khóa chủ duy nhất với KDC.
- A muốn thiết lập một liên kết logic với B để truyền dữ liệu.
- A có khóa chủ K_a chỉ A và KDC biết.
- B có khóa chủ K_b chỉ B và KDC biết.

Slide 18

Các bước tạo khóa phiên

1. A gửi yêu cầu đến KDC để nhận được khoá phiên nhằm thực hiện truyền thông với B.

- Bản tin gồm định danh của A, B và một định danh duy nhất N_1 cho phiên truyền gọi là nonce (nhãn thời gian, biến đếm, số ngẫu nhiên).
- Đối phương rất khó để xác định nonce.

Các bước tạo khóa phiên

2. KDC trả lời yêu cầu bằng một tin tức, được mã hoá với việc sử dụng khoá K_a. Người duy nhất có thể nhận và đọc được tin tức này đó chính là A và bởi vậy A có thể tin tưởng rằng tin tức đã được gửi từ KDC.

Các bước tạo khóa phiên

- Tin tức có hai thông tin được chờ đợi với A.
  - Khoá phiên dùng một lần K_s, nó sẽ được sử dụng làm khoá phiên để liên lạc
  - Tin tức nguyên bản đã gửi bao gồm nonce để A có khả năng đối chiếu câu trả lời phù hợp với câu đã hỏi .
- Trong tin tức, cũng bao gồm hai thông tin chờ đợi với B:
  - Khoá phiên dùng một lần K_s, nó sẽ được sử dụng làm khoá phiên để liên lạc.
  - Định danh của A (ID_A).

Slide 21

Các bước tạo khóa phiên

3. A lưu giữ khoá phiên K_s để dùng cho phiên liên lạc, và gửi về phía B một thông tin đã nhận được từ trung tâm (đó là thông tin EK_b [$K_s \| ID_A$]).

- Người sử dụng B biết được khoá phiên K_s và biết được thông tin nhận được đã được gửi từ KDC (bởi vì thông tin đó đã được mã hoá bằng K_b) .

4. Phía B gửi cho phía A một nonce mới N_2, nó được mã hoá bằng khoá phiên vừa nhận được.

5. Nhờ khoá phiên K_S, A trả lời lại $f(N_2)$ cho B, ở đây là hàm được thực hiện bằng biến đổi nào đó của N_2 (chẳng hạn bổ sung thêm đơn vị).

Slide 22

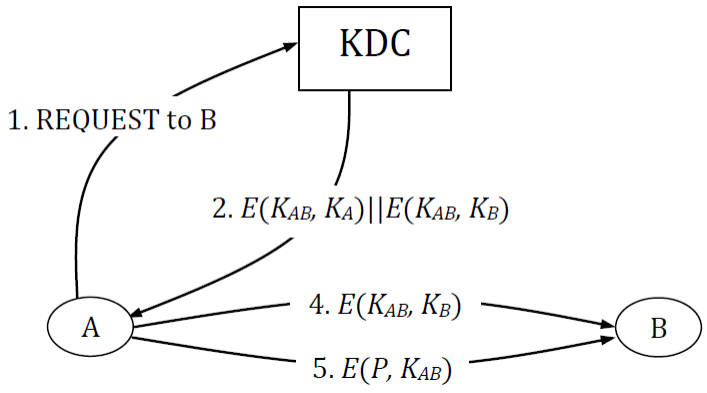
Nhận xét

- Các bước 4, 5 đảm bảo với B, tin tức là nguyên bản mã không bị tái tạo lại.
- Bước 1, 2, 3 → phân phối khóa.
- Bước 3, 4, 5 → Xác thực.

Slide 23

Định danh và trao đổi khóa phiên dùng mã hóa đối xứng với KDC

- Xét mô hình trao đổi khóa phiên



Slide 24

Định danh và trao đổi khóa phiên dùng mã hóa đối xứng với KDC

- Mô hình trên có thể bị tấn công replay attack. Ví dụ, Trudy có thể replay bước 4 mà B vẫn nghĩ là A gửi và B tiếp tục dùng *KAB* này làm khóa phiên. Dựa trên cơ sở đó Trudy tiếp tục replay bước 5.

Slide 25

Định danh và trao đổi khóa phiên dùng mã hóa đối xứng với KDC

- Do đó giao thức Needham/Schroeder được sửa lại như sau:

  1) A → B: $ID_A \,||N_A$

  2) B →KDC: $ID_B||N_B||E(ID_A||N_A, K_B)$

  3) KDC → A: $E(ID_B||N_A||K_S, K_A)||\ E(ID_A||\ K_S, K_B)||\ N_B$

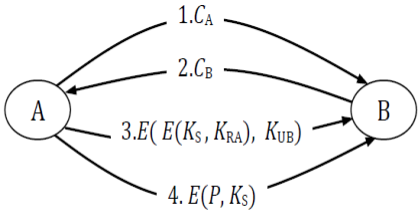
  4) A → B: $E(ID_A||K_S, K_B)||\ E(N_B, K_S)$

  5) A → B: $E(P, K_S)$

Slide 26

Định danh và trao đổi khóa phiên dùng mã hóa khóa công khai

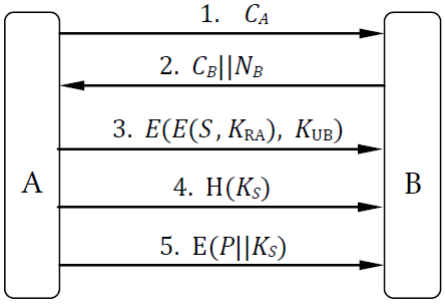
- Trong mô hình bên, Trudy có thể replay bước 3 mà B vẫn nghĩ là A gửi và B tiếp tục dùng *KS* này làm khóa phiên. Dựa trên cơ sở đó Trudy tiếp tục replay bước 4.



Slide 27

Định danh và trao đổi khóa phiên dùng mã hóa khóa công khai

- Dựa trên cơ sở đó Trudy tiếp tục replay bước 4. Ở đây áp dụng một cơ chế challenge/response khác để chống replay như sau:



Slide 28

Định danh và trao đổi khóa phiên dùng mã hóa khóa công khai

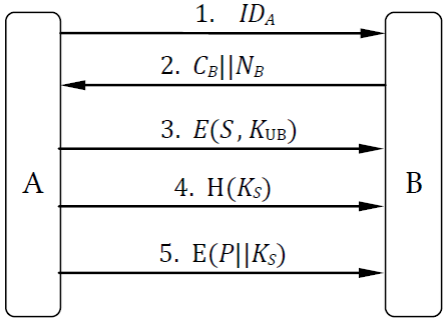
Mô tả:

- *Bước 1*: A gửi chứng chỉ *CA* cho B.
- *Bước 2*: B gửi chứng chỉ *CB* và nounce *NB* cho A.
- *Bước 3*: A chọn một *tiền khóa phiên S* và tính được khóa phiên *KS* = H(*S*||*NB*). A gửi chứng thực và bảo mật *S* cho B. B cũng tính khóa phiên *KS*.
- *Bước 4*: A gửi giá trị hash H(*KS*) cho B, B kiểm tra giá trị hash này với giá trị hash do B tự tính. Nếu khớp, B biết được rằng bước 3 không thể bị replay attack. Giả sử Trudy replay bước 3 nhưng không biết S, vậy Trudy không tính được *KS* tương ứng với *N*B mới của Bob, từ đó Trudy cũng không thể tính được H(*KS*). Do đó Trudy không thể replay bước 4 mà không bị Bob phát hiện.
- *Bước 5*: A và B tiến hành trao đổi dữ liệu.

Slide 29

Định danh và trao đổi khóa phiên dùng mã hóa khóa công khai

- Bài tập: Xét giao thức sau:



Slide 30

Định danh và trao đổi khóa phiên dùng mã hóa khóa công khai

Bài tập:

a) B có thể chắc chắn A là người ứng với IDA không? Nếu Trudy mạo danh A sử dụng IDA thì B có phát hiện được không? Giải thích

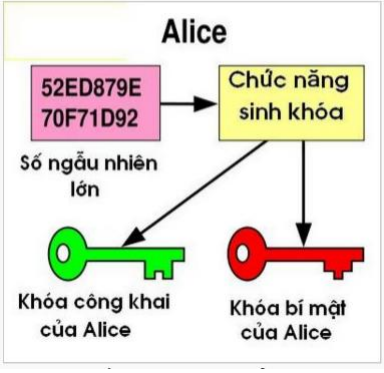
b) Giả sử A có password để định danh với B, B lưu trữ giá trị hash password của A. Hãy sửa giao thức trên để B có thể định danh được A.

2. Phân phối khóa trong các hệ mật khóa công khai

- Một trong các vai trò chính của mật mã công khai là giải quyết vấn đề phân phối khóa.
- Có hai hướng chính sử dụng mật mã khóa công khai:
  - Phân phối các khóa công khai.
  - Sử dụng mật mã khóa công khai để phân phối khóa bí mật.

2.1. Phân phối khóa công khai

- Một số công nghệ được đề xuất:
  - Công bố công khai khoá.
  - Catalog khoá công khai.
  - Trung tâm ủy quyền khoá công khai.
  - Chứng chỉ khoá công khai.



Slide 33

2.1. Phân phối khóa công khai

- Khóa được công bố công khai.
- Bất kì ai cũng có thể gửi khóa tới bất kì người khác.

Slide 34

Ví dụ

- Việc sử dụng PGP (Pretty Good Privacy) rất phổ biến (có sử dụng RSA).
- Khóa công khai được công bố trên USENET, Mailling list.

Tuy nhiên phương pháp này phát sinh vấn đề về chứng thực: Làm thế nào người gửi có thể đảm bảo Kub chính là khóa công khai của người nhận? Người thứ 3 có thể dùng khóa Ku3 và mạo danh người gửi để nói rằng đó là công khai của Bob
→Để khắc phục sử dụng mô hình (Certificate Authority-CA)

Hình 4-4. Trao đổi khóa công khai tự phát

Slide 35

Phân phối khóa không điều khiển

Các bước thực hiện chứng chỉ cho người gửi (Alice):

- Alice gửi định danh ID và khóa công khai KU_A của mình đến trung tâm chứng thực
- Trung tâm chứng thực kiểm tra tính hợp lệ của Alice, ví dụ nếu IDA là 'Microsoft', thì Alice phải có bằng chứng chứng tỏ mình thực sự là công ty Microsoft
- Dựa trên cơ sở đó, trung tâm chứng thực cấp một chứng chỉ Ca để xác nhận rằng khóa công khai KUA đó là tương ứng với IDA. 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.

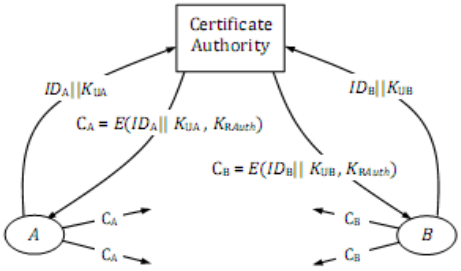
$$C_A = E(ID_A || K_{UA}, K_{RAuth})$$

(|| là phép nối dãy bit)

Slide 36

Phân phối khóa không điều khiển

- Alice công khai chứng chỉ CA
- Bob muốn trao đổi thông tin với Alice thì sẽ giải mã CA bằng khóa công khai của trung tâm chứng thực để có được khóa công khai KUA của Alice. Do đó Bob tin tưởng vào trung tâm chứng thực thì Bob sẽ tin tưởng là KUA là tương ứng với IDA của Alice



Hình 4-5. Trao đổi khóa công khai dùng trung tâm chứng thực

Slide 37

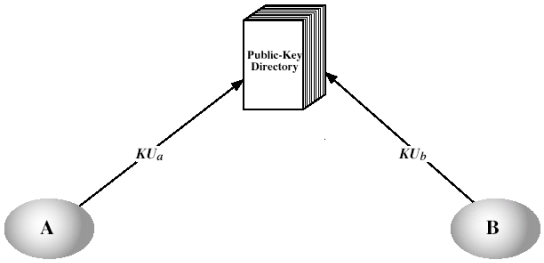
Nhận xét

- Là phương pháp rất đơn giản, thuận tiện.
- Nhưng có một điểm yếu chính → giả mạo khóa công khai của đối tương khác.

Slide 38

2.2. Catalog khóa công khai

- Một hình thức an toàn công bố khóa công khai cao hơn là sử dụng duy trì một catalog động của các khóa công khai.
- Việc phát hành catalog sẽ do một số thực thể hoặc trung tâm tin cậy thực hiện.



Slide 39

Mô tả các thành phần

1. Trung tâm được uỷ quyền lưu giữ catalog dưới dạng các bản ghi (tên, khoá công khai) của mỗi người tham gia.

2. Mỗi một người tham gia phải đăng ký khoá công khai của mình với trung tâm.

   Việc đăng ký phải diễn ra khi có mặt của chính người tham gia, hoặc thông qua một kênh truyền thông an toàn nào đó.

Slide 40

Mô tả các thành phần

3. Bất kì một người tham gia nào cũng có quyền thay đổi khoá công khai mới của mình vào bất kì thời điểm nào.

4. Theo từng chu kì, catalog phải được tái bản có bổ xung.

Hình thức xuất bản có thể tương tự như cuốn danh bạ điện thoại điện tử.

5. Mọi người tham gia có thể được phép xâm nhập vào catalog thường xuyên.

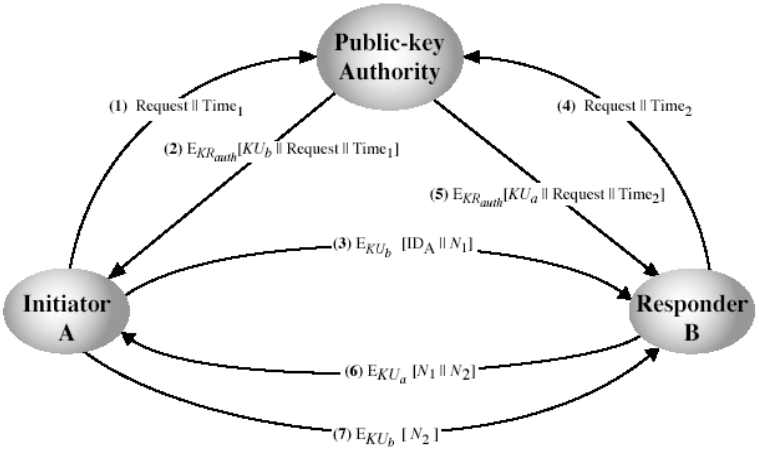
Đối với điều này việc bảo mật và xác thực giữa hai bên là ủy thác.

Nhận xét

- Có tính an toàn cao hơn so với phương pháp công bố công khai không kiểm soát khoá.
- Vẫn có điểm yếu:
  - Nếu đối phương có khóa riêng của trung tâm → giả mạo.

2.3. Trung tâm ủy quyền khóa công khai

- Cung cấp kiểm soát chặt chẽ quá trình phân phối khóa công khai từ catalog.



Slide 43

Mô tả các bước thực hiện

1. A gửi tin tức cùng với điểm dấu ngày tháng/thời gian tới trung tâm tin cậy, yêu cầu được cấp khoá công khai hiện thời của B.

2. Trung tâm trả lời, tin tức được mã hoá bằng khoá riêng của trung tâm KR_{auth}. Tin tức này A có thể giải mã nhờ khoá công khai của trung tâm. Bởi vậy, A có thể tin tưởng rằng tin tức đã được gửi từ trung tâm.

- Bản tin bao gồm:Khoá công khai của B: KU_B. Yêu cầu gốc, để A có thể đối chiếu với yêu cầu đã gửi, từ đó A tin tưởng chắc chắn rằng yêu cầu của mình đã không bị thay đổi trên đường truyền tới trung tâm.
- Điểm dấu thời gian, để A tin tưởng chắc chắn rằng: tin tức này không phải là tin tức cũ của trung tâm, khoá của B là khoá đang lưu hành.

Slide 44

Mô tả các bước thực hiện

3. A lưu giữ khoá công khai của B, và sử dụng nó để mã hoá tin tức để gửi cho B, trong đó phải có định danh của A (ID_A) và nonce (N1), được sử dụng để chỉ rõ tính duy nhất của phiên truyền.

4. B cũng sẽ nhận được khoá công khai của A: KU_A từ trung tâm, tương tự như A đã nhận.

5. Vào thời điểm này, khoá công khai đã được nhận bởi A và B theo thủ tục trên, họ có thể bắt đầu trao đổi tin tức.

Slide 45

Hai hành động bổ sung

6. B gửi tin tức tới A, được mã hoá bằng khoá công khai của A: KU_A, tin tức phải bao gồm nonce của A (N_1) và kèm theo nhãn thời gian mới của B (N_2).

Rõ ràng chỉ có B mới có khả năng giải mã tin tức từ A gửi đến (3), do có N_1 trong tin tức (6) làm cho A tin tưởng rằng B là người nhận được tin tức đã gửi.

7. A gửi quay lại N_2, được mã hoá bằng khoá công khai của B, để chứng tỏ rằng đó chính là A.

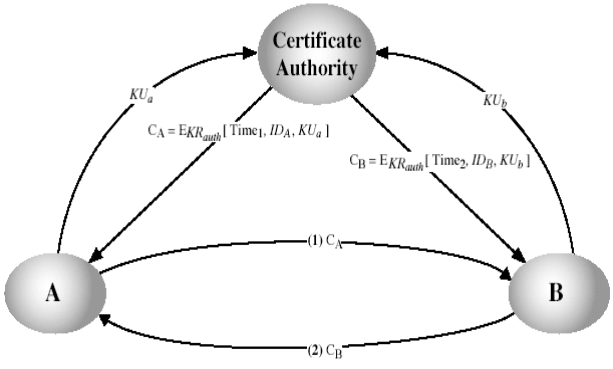
Slide 46

Nhận xét

- Như vậy, trong trường hợp chung đòi hỏi bảy tin tức.
- Tuy nhiên, việc gửi tin tức từ thủ tục đầu tiên đến thủ tục thứ bốn là không thường xuyên,
  - bởi vì cả hai phía đều có thể lưu giữ khoá công khai của nhau để tiếp tục sử dụng.
- Trung tâm ủy quyền có thể bị tắc nghẽn.
- Catalog lưu giữ tên và khóa công khai có thể bị giả mạo.

Slide 47

2.4. Chứng chỉ khóa công khai



Slide 48

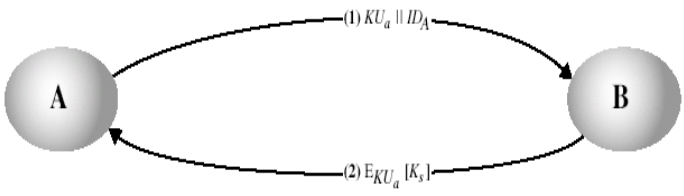
3. Phân phối khóa mật sử dụng mật mã khóa công khai

- Phân phối khóa mật đơn giản.
- Phân phối khóa mật với bảo mật và xác thực.
- Sơ đồ lai ghép.

Slide 49

3.1. Phân phối khóa mật đơn giản

- Sơ đồ được đề xuất bởi Merkle.



Slide 50

Các bước thực hiện

1. A phát sinh cặp khoá công khai và khoá riêng (KU_A, KR_A) và truyền tin tức về phía B, bao gồm KU_A và định danh của A (ID_A).

2. B phát sinh khoá mật K_s và truyền khoá này về phía A, tin tức được mã hoá bằng khoá công khai của A.

3. A tính $D_{KRa}[E_{KUa}[K_s]]$, để khôi phục khoá mật . Bởi vì chỉ có A mới có khả năng giải mã bản tin đó, và chỉ có A, B biết khoá mật K_s.

4. A bỏ khóa KU_a, KR_a, còn B bỏ khóa KU_a.

Slide 51

Nhận xét

- Sau khi kết thúc liên lạc, cả A, B đều vứt bỏ khoá mật K_s.
- Không khảo sát tính đơn giản, thủ tục này quả là thuận lợi.
- Không có khoá nào tồn tại trước mỗi phiên liên lạc, không có khoá nào còn lại sau mỗi phiên liên lạc. Bởi vậy, khả năng bị mất khoá trở nên rất hiếm hoi. Các cuộc liên lạc xem ra có vẻ an toàn.
- Không an toàn khi đối phương có thể chặn tin tức (giữ chậm hoặc thay đổi nội dung) → tấn công theo kiểu **man-in-the-middle attack**.
- Đối phương có thể can thiệp vào phiên truyền thông theo cách sau (không phát hiện được).

Slide 52

Mô tả

1. A phát sinh cặp khoá công khai và khoá riêng (KU_A, KR_A) và truyền tin tức về phía B, bao gồm KU_A và định danh của A (ID_A).

2. Đối phương E chặn tin tức, và tạo nên một cặp khoá giả (KU_e, KR_e) và truyền về phía B, bao gồm KU_e và ID_A.

3. B phát sinh khoá mật K_s và truyền với EKU_e [K_s].

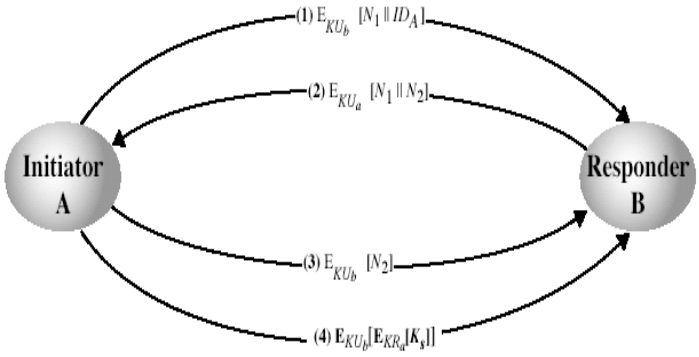
4. E chặn tin tức này và biết K_s khi tính $D_{KRe}[E_{KUe}[K_s]]$.

5. E truyền cho A tin tức $E_{KUa}[K_s]]$.

Nhận xét

- A và B đều biết K_s, nhưng sẽ không biết rằng K_s còn được biết cả bởi đối phương E.
- Thủ tục sẽ hữu dụng chỉ trong trường hợp kênh truyền chỉ có một nguy cơ nghe trộm.
- Mô hình có khả năng chống lại cả tấn công thụ động và chủ động.
- Mô hình được thực hiện với giả thiết A và B đã có khóa công khai của nhau.

3.2. Phân phối khóa mật với sự bí mật và xác thực



Slide 55

Mô tả

- Sử dụng trung tâm phân phối khóa KDC (chia xẻ khóa chủ với người dùng).
- Phân phối khóa phiên bí mật được mã hóa bởi khóa chủ.
- Sử dụng sơ đồ khóa công khai để phân phối khóa chủ.

Slide 56

Các bước thực hiện

1. A sử dụng khóa công khai của B để gửi cho B một văn bản mã, bao gồm định danh của A (ID_A) và nonce (N1).

2. B gửi tin tức cho A, tin tức được mã hoá nhờ KU_A, bao gồm nonce (N_1) và (N_2). Rõ ràng chỉ có B mới có khả năng giải mã tin tức (1), sự có mặt của N_1 trong tin tức (2), thuyết phục A rằng tin tức đã được gửi từ phía B.

Các bước thực hiện

3. A gửi trở lại(N_2), được mã hoá bằng khoá công khai của B, điều đó bảo đảm rằng tin tức là của phía A.

4. A chọn khoá mật K_s và gửi cho B tin tức: $M = E_{KUb}[E_{KRa}[K_s]]$.

Tin tức này được mã hoá bằng khoá công khai của B, để chứng tỏ rằng chỉ có B mới giải mã được nhờ khoá riêng của mình, còn mã hoá bằng khoá riêng của A, xác thực rằng chỉ có A là người đã gửi tin tức đó.

5. B tính DKUa[EKRb[M]], để khôi phục lại được khoá mật Ks.

Giao thức KERBEROS

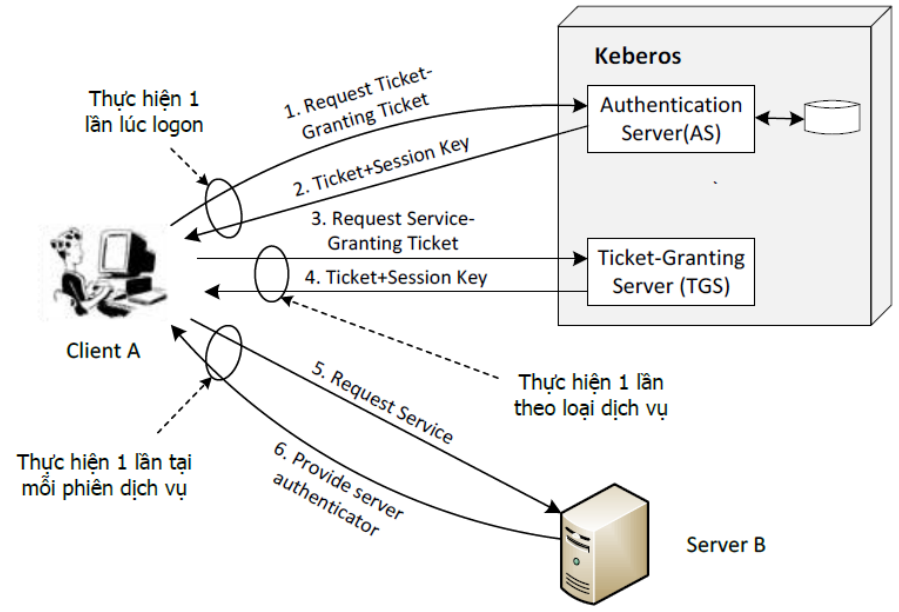
- Kerberos là tên của một hệ dịch vụ phân phối (hay cấp phát) khóa phiên (session) cho từng phiên truyền tin bảo mật theo yêu cầu của người dùng trong một mạng truyền tin
- Kerberos là một giao thức chứng thực. Keberos chỉ dựa trên mã hóa đối xứng
- Ra đời cùng thời điểm với KDC, nhưng đã trở nên thông dụng. (Windows 2000 sử dụng cơ chế Kerberos để chứng thực)
- Đầu tiên được thiết kế tại MIT, nó đã qua nhiều phiên bản khác nhau

Slide 59

Giao thức KERBEROS

- Mục đích của Keberos là để trao đổi khóa phiên, thông qua đó đảm bảo tính bảo mật và tính chứng thực.
- Do nguyên tắc của Keberos dựa trên KDC nên Keberos cũng kế thừa được những ưu điểm của mô hình KDC như tính phi trạng thái

Slide 60



Hình 7-9. Mô hình chứng thực và trao đổi khóa phiên Keberos

Giao thức KERBEROS

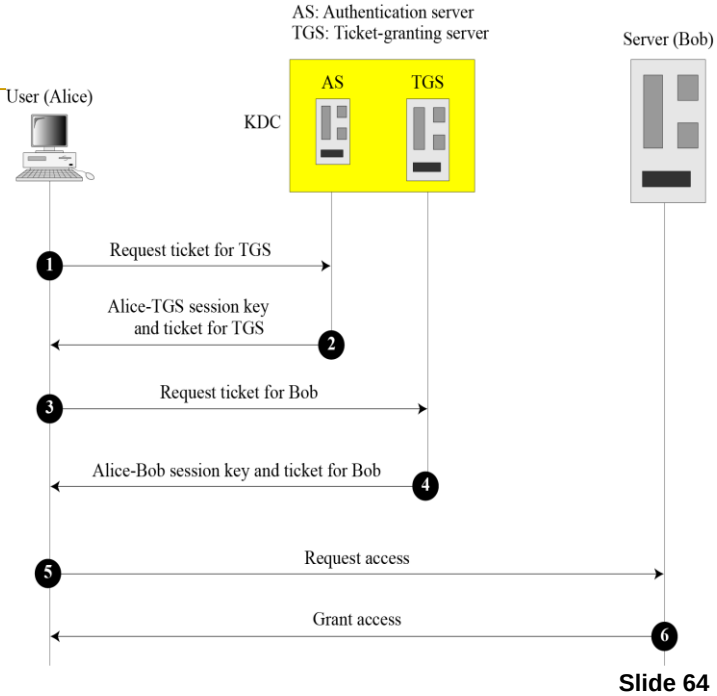
- Trong giao thức Kerberos gồm có:
  - Servers
  - Operation
  - Using Different Servers
  - Kerberos Version 5
  - Realms

Servers

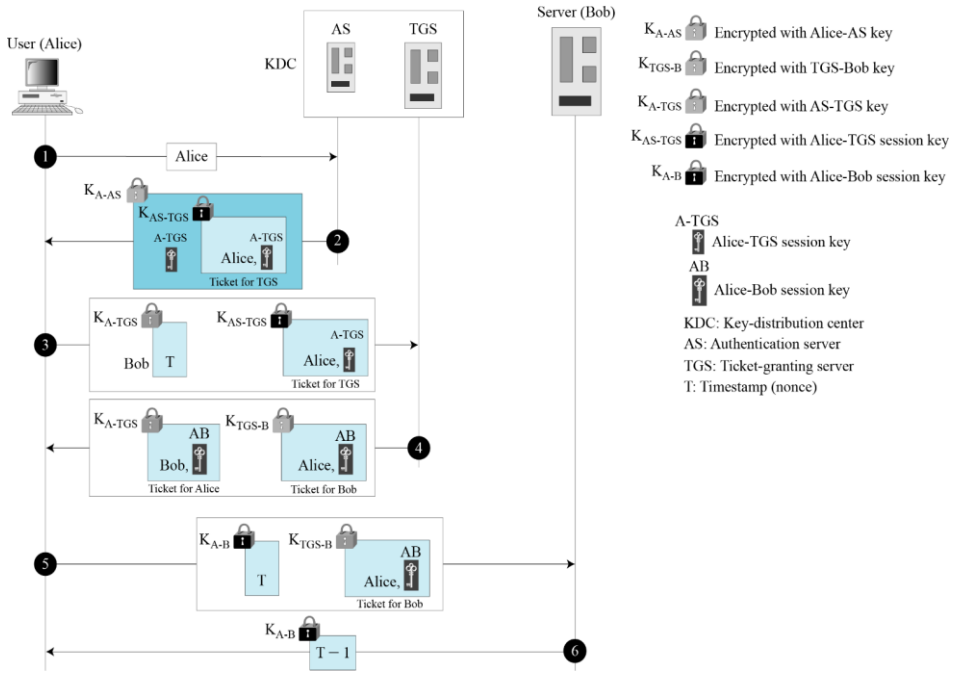
- Authentication Server (chỉ có 1 AS): là KDC trong giao thức Kerberos. AS có nhiệm vụ cung cấp khóa đối xứng cho trao đổi giữa client A và server TGS
- Ticket-granting server (TGS): đóng vai trò là các KDC, có nhiệm vụ cung cấp khóa đối xứng cho trao đổi giữa client A và server dịch vụ B
- Các người sử dụng A cần đăng ký mật khẩu *K*A của mình với Server AS. Các server dịch vụ B đăng ký khóa bí mật *K*B với Server TGS. Server TGS cũng đăng ký khóa bí mật *K*TGS với Server AS
- Real (data) server (của Bob): cung cấp dịch vụ cho người dùng (Alice)

Slide 63

Servers



Slide 64



Slide 65

Using Different Servers

- Nếu Alice cần nhận các dịch vụ từ các servers khác, cô ta chỉ cần lặp lại 4 bước sau cùng.

Slide 66

Realms (lãnh địa)

- Kerberos cho phép sự phân bố toàn cục của các AS và TGS, với mỗi hệ thống được gọi là một realm. Người dùng có thể lấy một ticket cho một local server hoặc remote server.

Slide 67

4. Thỏa thuận trao đổi khóa

- Alice và Bob có thể tạo ra một session key giữa chúng mà không cần dùng một KDC. Phương pháp tạo session-key này được tham chiếu như một symmetric-key agreement.
- Hai phương pháp
  - Diffie-Hellman Key Agreement
  - Station-to-Station Key Agreement

Slide 68

Mô hình trao đổi khóa Diffie-Hellman

- Trao đổi khoá Diffie Hellman là sơ đồ khoá công khai đầu tiên được đề xuất bởi Diffie và Hellman năm 1976 cùng với khái niệm khoá công khai.

- Sau này được biết đến bởi James Ellis (Anh), người đã đưa ra mô hình tương tự năm 1970. Đây là phương pháp thực tế trao đổi công khai các khoá mật. Nó thúc đẩy việc nghiên cứu đề xuất các mã khoá công khai. Sơ đồ được sử dụng trong nhiều sản phẩm thương mại.

Slide 69

Mô hình trao đổi khóa Diffie-Hellman

- Không thể dùng để trao đổi mẩu tin bất kỳ.
- Tuy nhiên nó có thể thiết lập khoá chung.
- Chỉ có hai đối tác biết đến.
- Giá trị khoá phụ thuộc vào các đối tác (và các thông tin về khoá công khai và khoá riêng của họ).
- Dựa trên phép toán lũy thừa trong trường hữu hạn (modulo theo số nguyên tố hoặc đa thức) là bài toán dễ.
- Độ an toàn dựa trên độ khó của bài toán tính logarit rời rạc (giống bài toán phân tích ra thừa số) là bài toán khó.

Slide 70

Mô hình trao đổi khóa Diffie-Hellman

Giao thức trao đổi khoá giữa A và B:

- A và B thống nhất chọn chung một số nguyên tố q và một phần tử sinh α.

Global Public Elements	
q	prime number
α	$\alpha < q$ *and* α a primitive root of q

Slide 71

Mô hình trao đổi khóa Diffie-Hellman

- Tạo cặp khóa:

User A Key Generation	
Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = \alpha^{X_A} \bmod q$

User B Key Generation	
Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = \alpha^{X_B} \bmod q$

Slide 72

Mô hình trao đổi khóa Diffie-Hellman

- Xác định khóa phiên: Dựa vào số học modulo

Calculation of Secret Key by User A

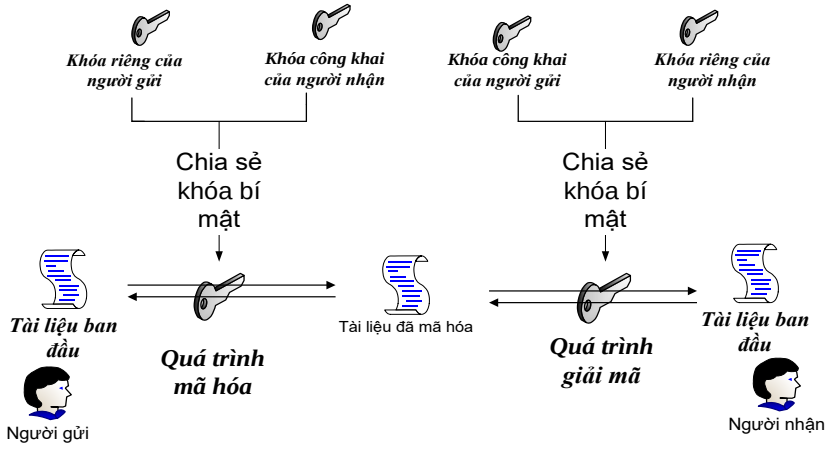
$$K = (Y_B)^{X_A} \bmod q$$

Calculation of Secret Key by User B

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

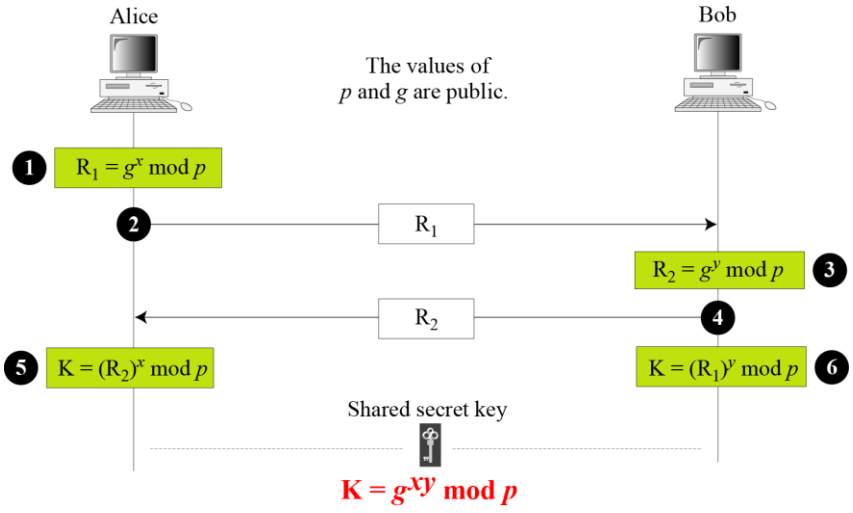
Slide 73

Mô hình truyền tin bí mật sử dụng trao đổi khóa diffie-hellman



Slide 74

Mô hình thỏa thuận trao đổi khóa



Slide 75

Ví dụ

Diffie Hellman Key Exchange

	Alice	Evil Eve	Bob
	Alice and Bob exchange a Prime (P) and a Generator (G) in clear text, such that P > G and G is Primitive Root of P G = 7, P = 11	Evil Eve sees G = 7, P = 11	Alice and Bob exchange a Prime (P) and a Generator (G) in clear text, such that P > G and G is Primitive Root of P G = 7, P = 11
Step 1	Alice generates a random number: X_A X_A=6 (Secret)		Bob generates a random number: X_B X_B=9 (Secret)
Step 2	$Y_A = G^{X_A} \pmod P$ $Y_A = 7^6 \pmod{11}$ $Y_A = 4$		$Y_B = G^{X_B} \pmod P$ $Y_B = 7^9 \pmod{11}$ $Y_B = 8$
Step 3	Alice receives Y_B = 8 in clear-text	Evil Eve sees $Y_A = 4$, $Y_B = 8$	Bob receives Y_A = 4 in clear-text
Step 4	Secret Key = $Y_B^{X_A} \pmod P$ Secret Key = $8^6 \pmod{11}$ **Secret Key = 3**		Secret Key = $Y_A^{X_B} \pmod P$ Secret Key = $4^9 \pmod{11}$ **Secret Key = 3**

Slide 76

Thỏa thuận trao đổi khóa diffie-hellman

- Ví dụ: Giả sử rằng $g = 7$ và $p = 23$. Các bước như sau:

1. Alice chọn x = 3 và tính $R_1 = 7^3 \bmod 23 = 21$.
2. Bob chọn y = 6 và tính $R_2 = 7^6 \bmod 23 = 4$.
3. Alice gửi số 21 cho Bob.
4. Bob gửi số 4 cho Alice.
5. Alice tính Symmetric Key $K = 4^3 \bmod 23 = 18$.
6. Bob tính Symmetric key $K = 21^6 \bmod 23 = 18$.
7. Giá trị của K giống nhau giữa Alice và Bob; $g^{xy} \bmod p = 7^{18} \bmod 35 = 18$.

Slide 77

Thỏa thuận trao đổi khóa diffie-hellman

- Bài tập: Cho số nguyên tố q=353 và α=3
- Chọn các khoá mật ngẫu nhiên: A chọn xA=97, B chọn xB=233. Tính khóa công khai và khóa phiên. Sau đó cho nhận xét
- Tính các khoá công khai:
  - **$YA=3^{97} \bmod 353 = 40$ (A)**
  - **$YB=3^{233} \bmod 353 = 248$ (B)**
- Tính khoá phiên chung:
  - **$K_{AB} = Y_B{}^{x}{}_A \bmod 353 = 248^{97} = 160$ (A)**
  - **$K_{AB} = Y_A{}^{xB} \bmod 353 = 40^{233} = 160$ (B)**

Slide 78

Thỏa thuận trao đổi khóa diffie-hellman

- Ví dụ 2: Chúng ta dùng một chương trình tạo một số nguyên ngẫu nhiên 521-bit (khoảng 159 chữ số). Chúng ta cũng chọn g, x, và y như sau.

p	7646242985634935721824937659550305074763380967269497489235737728609252356666607554236374233096611800333381061947301309504147387009991780436548785807987581
g	2
x	557
y	273

Slide 79

Thỏa thuận trao đổi khóa diffie-hellman

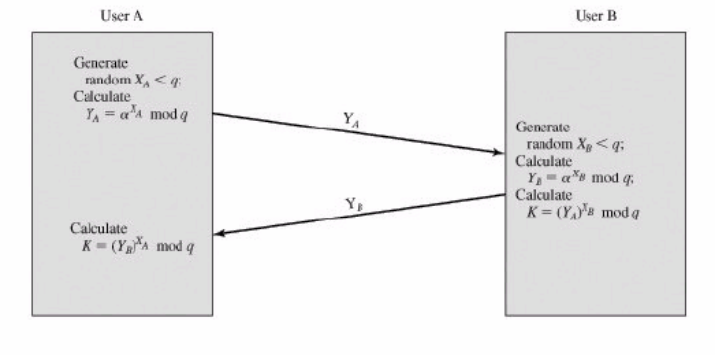
- Giá trị của R_1, R_2, và K là:

R_1	8449202842056655052161729474910350941434336985200126608628636310676736199592808285867008021318592909451402175003199733129458360838219430659660201579553 54
R_2	4352628387092003794707471148955816276363891162621155579751233792185663100114357182083900401818764868417538311653426916302634211067215085896255201288594143
K	1556380006645222905962258275232707652732180469444236785203204001464065008879366512042574267766083279110171530386745612522131516109765842001204086433617740

Slide 80

Thỏa thuận trao đổi khóa diffie-hellman

Tấn công
q = 353; a = 3; Y_A = 40; Y_B= 248
Vét cạn để tìm 2 khóa bằng nhau (với số nhỏ)
Không chống được tấn công Man-in-the-Middle



Slide 81

Thỏa thuận trao đổi khóa diffie-hellman

4. Nếu cho số nguyên tố p = 353 thì a = 3 là một primitive root modulo p. Sử dụng hai số này để xây dựng một hệ thống trao đổi khoá Diffiel-Hellman.
   a. Nếu Alice chọn một private key X_A = 97, giá trị public key Y_A của Alice là?
   b. Nếu Bob chọn một private key X_B = 233, giá trị public key Y_B của Bob là?
   c. Giá trị của khoá bí mật thống nhất giữa cả Alice và Bob là bao nhiêu?

Slide 82

Thỏa thuận trao đổi khóa diffie-hellman

5. Cho p = 13.
   a. Chứng minh rằng a = 2 là một primitive root modulo p. Sử dụng hai tham số này để xây dựng một hệ thống trao đổei khoá Diffie-Hellman.
   b. Nếu public key của Alice là Y_A = 7, giá trị private key X_A của cô ấy là bao nhiêu?
   c. Nếu public key của Bob là Y_B = 11, giá trị private key X_B của anh ấy?

Slide 83

Thỏa thuận trao đổi khóa diffie-hellman

Bảo mật của Diffie-Hellman:

1. Discrete Logarithm Attack

2. Man-in-the-Middle Attack

Slide 84

Thỏa thuận trao đổi khóa diffie-hellman

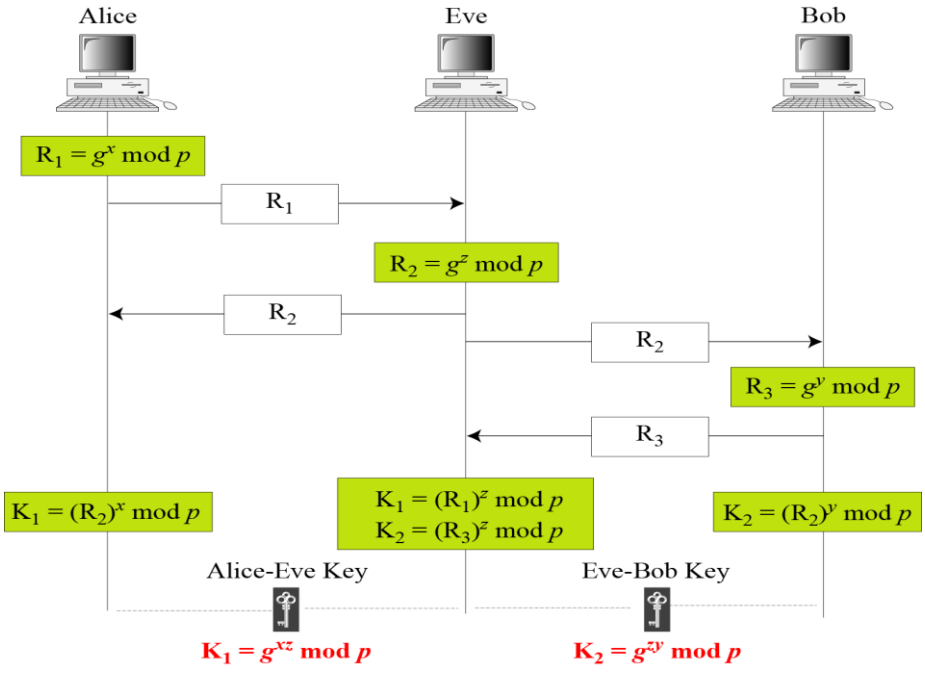
Discrete Logarithm Attack

Eve chặn R_1 và R_2. Nếu cô ta tìm ra được x từ $R_1=g^x \bmod p$ và y từ $R_2=g^y \bmod p$ → có thể tính toán được khóa $K=g^{xy} \bmod p$ → Khóa bí mật này không còn bí mật nữa.

Để an toàn, ta nên chọn:

- Số nguyên tố p phải là rất lớn (hơn 300 chữ số)
- Số p phải được chọn sao cho p-1 có ít nhất một thừa số nguyên tố lớn (nhiều hơn 60 chữ số)
- Phần tử sinh phải được chọn từ nhóm <Zp*,x>
- Bob và Alice phải hủy x và y sau khi tính K; x và y chỉ nên sử dụng một lần

Slide 85



Slide 86

Thỏa thuận trao đổi khóa diffie-hellman

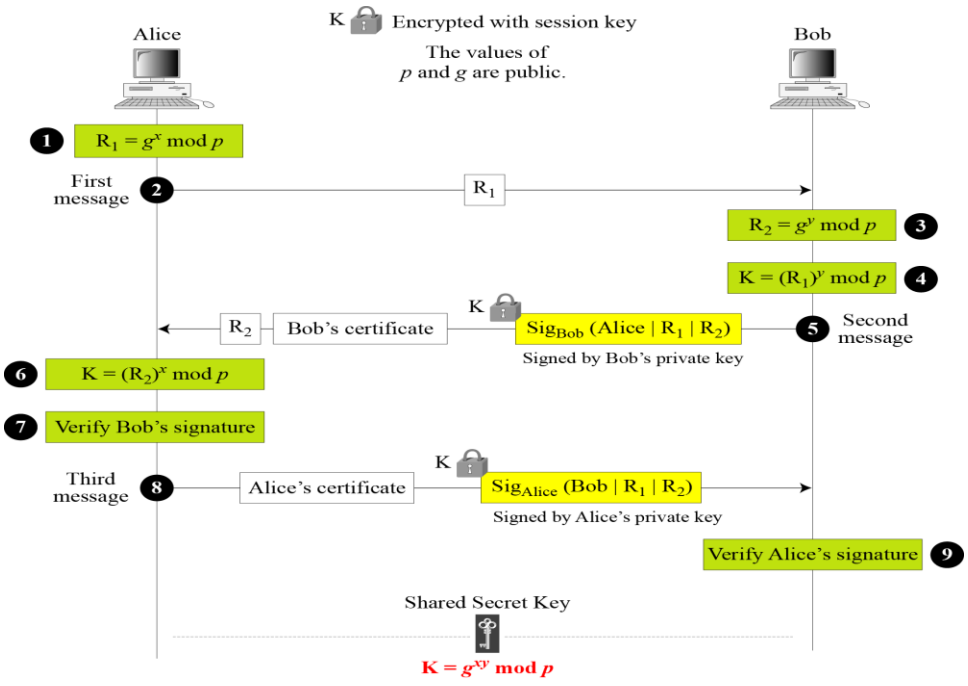
- Giao thức là an toàn đối với việc tấn công thụ động, nghĩa là một người thứ ba dù biết b_A và b_B sẽ khó mà biết được $K_{A,B}$.

Slide 87

Thỏa thuận trao đổi khóa Station-to-Station

- Là một giao thức dựa trên Diffie-Hellman
- Dùng Digtal signature với Public-key Certificates để thiết lập nên session key giữa Alice và Bob

Slide 88



Slide 89

Thỏa thuận trao đổi khóa Station-to-Station

- Giao thức này **ngăn chặn được tấn công man-in-the-middle.**
  - Sau khi chặn R_1, Eve không thể gửi R_2 của cô ta cho Alice và giả bộ nó được gửi đến từ Bob bởi vì Eve không thể giả mạo được Private key của Bob để tạo ra Sinature – Signature không thể được thẩm tra bằng public key của Bob được xác định trong Certificate.
  - Cùng cách tương tự Eve không thể giả private key của Alice để ký thông điệp thứ 3 gửi bởi Alice.

Slide 90

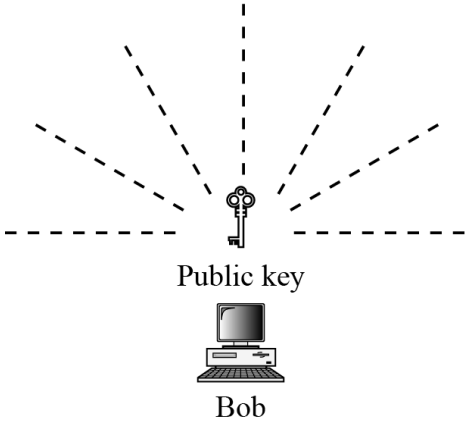
4. PUBLIC-KEY DISTRIBUTION

- Trong mã hóa khóa công khai, mọi người có thể truy xuất đến Public key của mọi người; các Public Key này sẵn sàng được công khai
- Public key, giống như khóa bí mật, cần được phân bố như thể nào cho hữu dụng

Slide 91

4.1 Public Announcement

- Đưa Public Key lên Website, tạp chí
- Không an toàn, nó có thể bị giả mạo



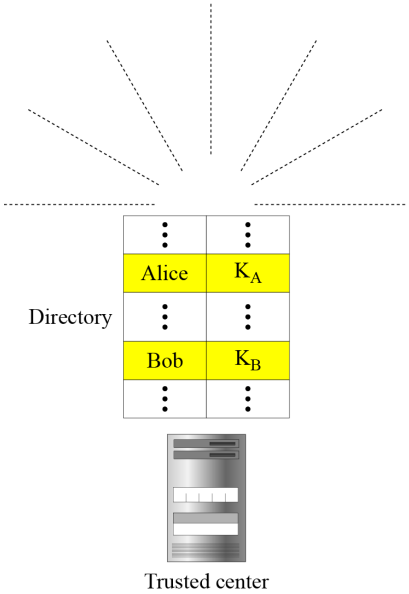
Slide 92

4.2 Trusted Center

- Cách này an toàn hơn, phải có một trung tâm tin cậy lưu giữ lại một danh bạ (Directory) các Public Keys, danh bạ này được cập nhật tự động
- Mỗi user chọn một Private Key (giữ bí mật) và một Public Key (được chuyển để chèn vào danh bạ)
- Trung tâm yêu cầu mỗi user đăng ký và chứng minh identity của cô/anh ta
- Danh bạ có thể được yết thị công khai bởi Trusted Center
- Center cũng đáp trả lại những thẩm tra về Public Key

Slide 93

4.2 Trusted Center



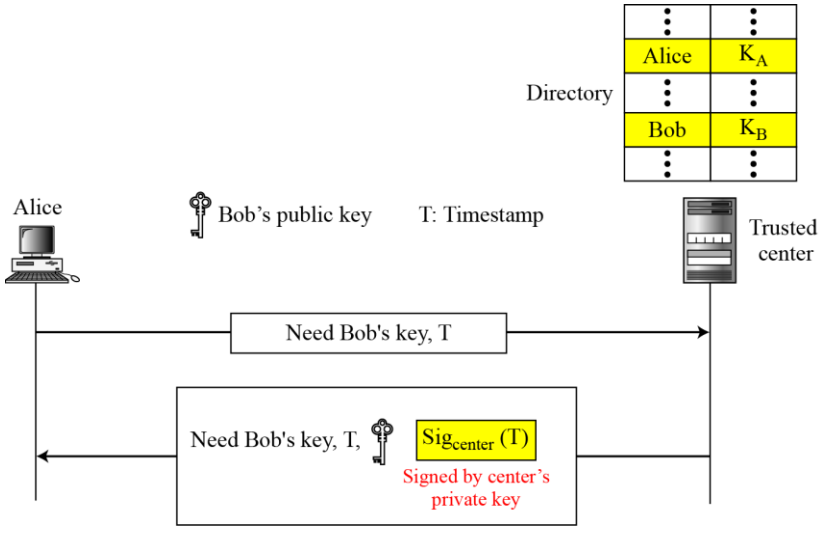
Slide 94

4.3 Controlled Trusted Center

- Mức độ bảo mật cao hơn có thể đạt được nếu được thêm các điều khiển (Control) vào việc phân phối Public Key.
- Bao gồm một Timestamp và được ký bởi người thẩm quyền để ngăn chặn sự nghe lén (interception) và sự hiểu chỉnh response

Slide 95

4.3 Controlled Trusted Center



Slide 96

4.4 Certification Authority

- Các phương pháp trước có thể tạo một tải trọng nặng lên Center nếu số lượng yêu cầu lớn
- Bob muốn 2 điều: (1) mọi người đều biết Pulic Key của anh ta; (2) không một ai chấp nhận một Public Key bị giả mạo
- → đến Certification Authority (cơ quan chứng nhận)- một tổ chức liên bang hoặc quốc gia mà gắn kết Public Key với một thực thể và phát hành một chứng chỉ (Certificate)

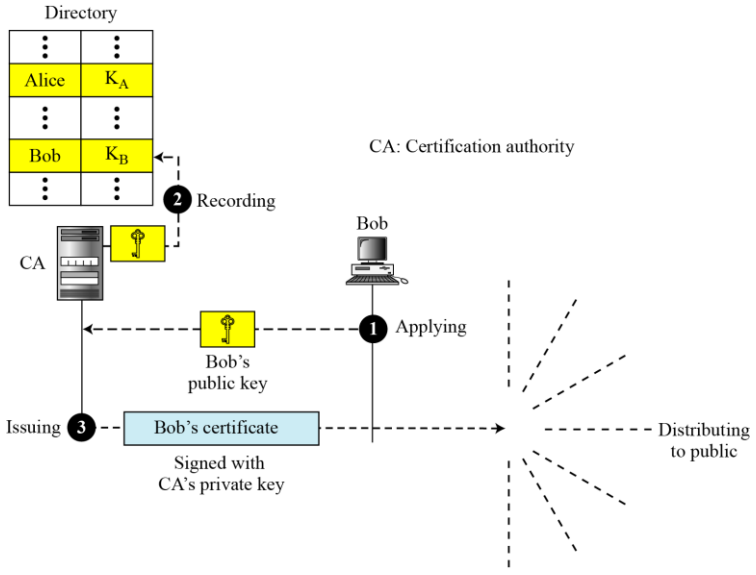
Slide 97

4.4 Certification Authority

- CA có một Public Key của chính nó, ai cũng biết mà không thể giả mạo.
- CA kiểm tra nhận dạng của Bob (dùng một ID hình ảnh với những bằng chứng khác)
- Nó hỏi Public Key của Bob và ghi nó vào một Certificate (để ngăn chặn Certificate giả mạo, CA ký (sign) vào Certificate bằng Private Key của nó.
- Bob có thể đăng Certificate đã được ký.
- Alice sẽ tải Certificate của Bob về và dùng Public Key của Center đề trích ra Public Key của Bob

Slide 98

4.4 Certification Authority



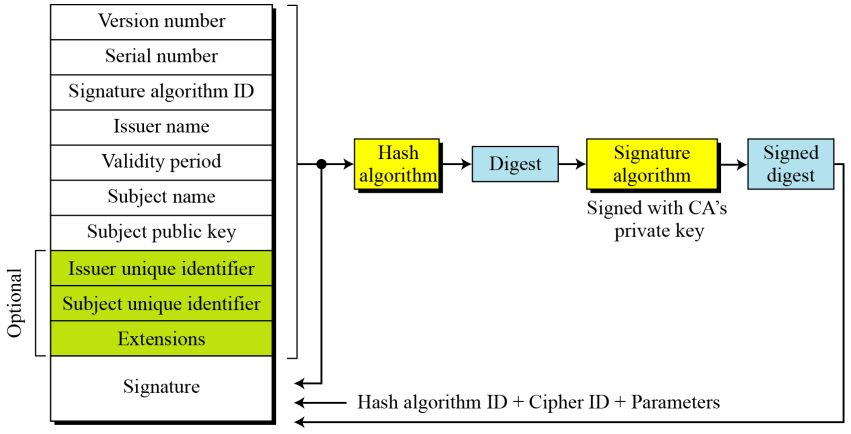
Slide 99

4.5 X.509

- CA đã giải quyết được vấn đề gian lận Public Key, nhưng nó cũng tạo ra một tác động thứ yếu.
  - Mỗi Certificate có thể có một định dạng khác nhau, điều này gây khó khăn nếu người dùng muốn dùng một chương trình tự động để tải và lấy về các Public Key của nhiều người khác nhau
  - → cần một định dạng phổ quát cho Certificate
- ITU thiết kế X.509 – là một cách để mô tả Certificate trong một cách có cấu trúc, nó dùng một giao thức nổi tiếng gọi là ASN.1 (Abstract Syntax Notation 1)

Slide 100

4.5 X.509



Slide 101

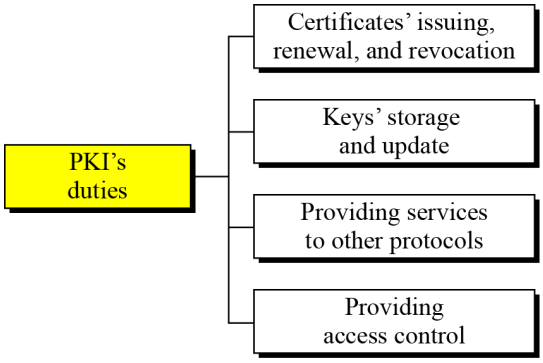
4.5 X.509

- Thay mới (Certificate Renewal)
  - Certificate hết hạn dùng, nếu không có vấn đề thì CA cấp lại một certificate mới trước khi hết hạn
- Thu hồi (Delta Revocation)
  - Private key ứng với Public Key có thể bị thỏa hiệp
  - CA không muốn chứng nhận người dùng nữa
  - Private Key của CA mà có thể dùng thẩm tra Certificate, có thể đã bị thỏa hiệp
  - → CA phát hành danh sách các thu hồi (Certificate Revocation List - CRL)

Slide 102

4.6 Public-Key Infrastructures (PKI)

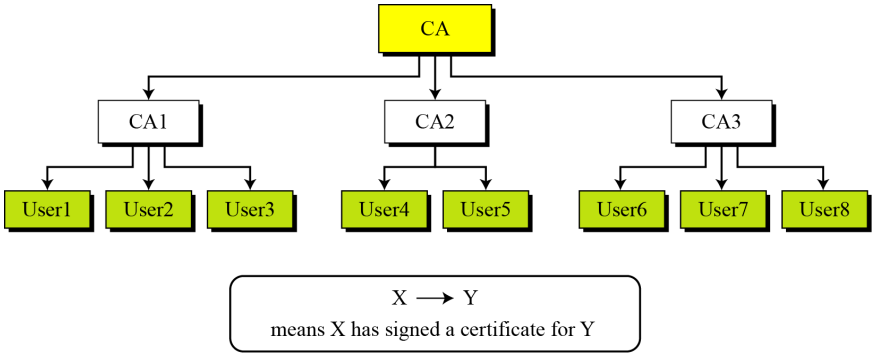
- PKI là một mô hình dùng để tạo, phân phối và thu hồi Certificate dựa trên X.509
- Trách nhiệm của PKI:



Slide 103

4.6 Public-Key Infrastructures (PKI)

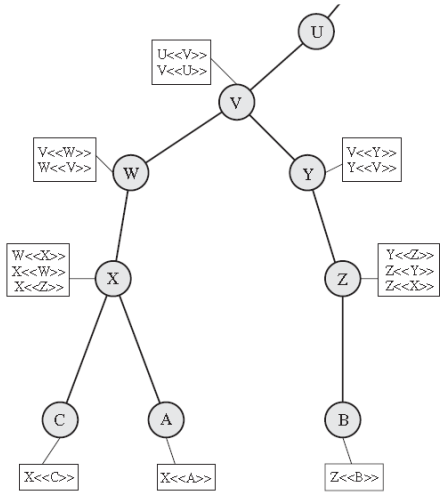
- Mô hình phân cấp PKI



Slide 104

4.6 Public-Key Infrastructures (PKI)

- Ví dụ: Mô hình phân cấp PKI



Slide 105

4.6 Public-Key Infrastructures (PKI)

Ví dụ:

- A muốn có được Certificates của B để lấy Public Key của B

  X<<W>> W<<V>> V<<Y>>Y<<Z>>Z<<B>>

- B muốn có Public Key của A

  Z<<Y>> Y<<V>> V<<W>>W<<X>>X<<A>>

Slide 106

4. Hạ tầng khóa công khai (PKI)

- *"PKI là tập hợp của các công nghệ mật mã, phần mềm, phần cứng chuyên dụng và các dịch vụ cho phép các tổ chức/doanh nghiệp đảm bảo an toàn thông tin liên lạc, định danh và xác thực được người dùng, khách hàng trên các giao dịch qua mạng/Internet"*.

Slide 107

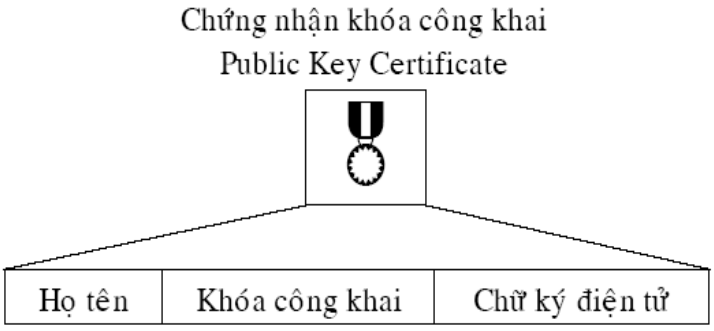
Các thành phần

1. Chứng chỉ khóa công khai: họ tên hoặc định danh của người sở hữu thật sự của khóa, khóa công cộng và chữ ký điện tử giúp xác nhận được tính hợp lệ của hai thành phần này.

2. Hệ thống phân phối khóa tin cậy: sử dụng hệ thống trao đổi thông tin tin cậy để chuyển mã khóa công cộng đến người nhận.

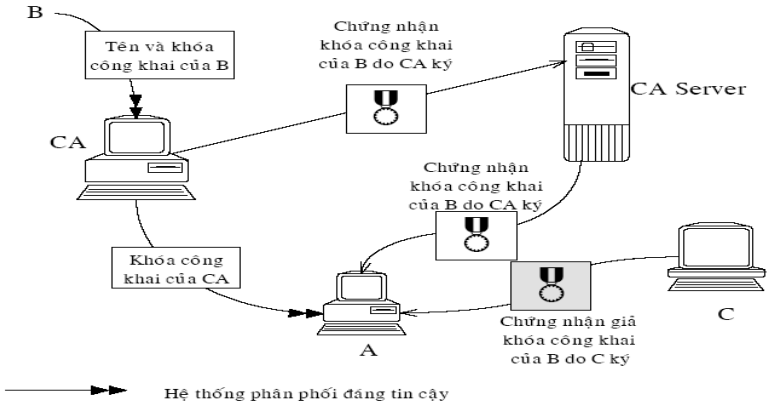
Slide 108

Các *thành phần của một chứng nhận khóa công cộng*



Slide 109

Mô hình Certification Authority đơn giản



Slide 110

Các loại giấy chứng nhận khóa công cộng

- Giấy chứng nhận là một tập tin nhị phân có thể dễ dàng chuyển đổi qua mạng máy tính.
- Tổ chức CA áp dụng chữ ký điện tử của nó cho giấy chứng nhận khóa công cộng mà nó phát hành.

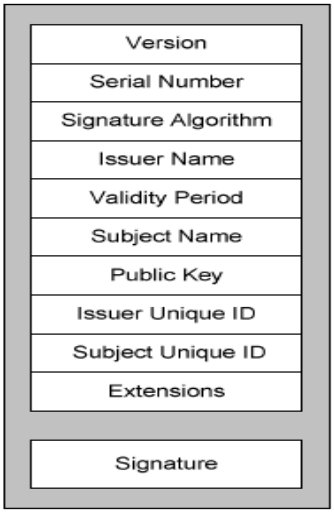
Slide 111

Chứng nhận X.509

- Chứng nhận X.509 là chứng nhận khóa công cộng phổ biến nhất.
- Hiệp hội viễn thông quốc tê (ITU) đã chỉ định chuẩn X.509 vào năm 1988 (phiên bản 1).
- Phiên bản 2 (1993) của chuẩn X.509 được phát hành với 2 trường tên nhận dạng duy nhất được bổ sung.
- Phiên bản 3 (1997) của chuẩn X.509 được bổ sung thêm trường mở rộng.

Slide 112

Phiên bản 3 của chuẩn chứng nhận X.509



Slide 113

Mô tả

- Một chứng nhận khóa công cộng kết buộc một khóa công cộng với sự nhận diện của một người (hoặc một thiết bị).
- Khóa công cộng và tên thực thể sở hữu khóa này là hai mục quan trọng trong một chứng nhận.
- Hầu hết các trường khác trong chứng nhận X.509 phiên bản 3 đều đã được chứng tỏ là có ích.

Slide 114

Mô tả một số trường

- *Signature Algorithm*: Thuật toán chữ ký chỉ rõ thuật toán mã hóa được CA sử dụng để ký giấy chứng nhận.
- *Subject Name*: là một X.500 DN (X.500 Distinguised Name – X.500 DN), xác định đối tượng sở hữu giấy chứng nhận mà cũng là sở hữu của khóa công cộng.

Slide 115

Mô tả một số trường

- *Public key*: Xác định thuật toán của khóa công cộng (như RSA) và chứa khóa công cộng được định dạng tuỳ vào kiểu của nó.
- *Extensions*: Chứa các thông tin bổ sung cần thiết mà người thao tác CA muốn đặt vào chứng nhận. Trường này được giới thiệu trong X.509 phiên bản 3.
- *Signature*: Đây là chữ ký điện tử được tổ chức CA áp dụng.

Slide 116

Chứng nhận PGP

- Giấy chứng nhận X.509 được ký bởi tổ chức CA.
- Trong khi đó, giấy chứng nhận PGP có thể được ký bởi nhiều cá nhân.
  - Mô hình tin cậy của giấy chứng nhận PGP đòi hỏi phải tin tưởng vào những người ký giấy chứng nhận PGP muốn dùng chứ không chỉ tin tưởng vào CA phát hành X.509.

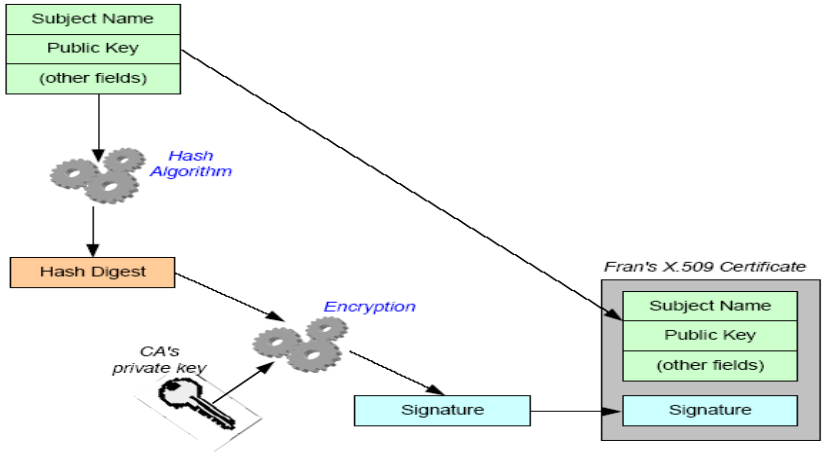
Slide 117

Sự chứng nhận và kiểm tra chữ ký

- Quá trình chứng nhận chữ ký diễn ra theo hai bước.
  - Đầu tiên, các trường của chứng nhận được băm bởi thuật toán cho trước.
  - Sau đó, kết quả xuất của hàm băm, được mã hóa với khóa bí mật của tổ chức CA đã phát hành chứng nhận này.

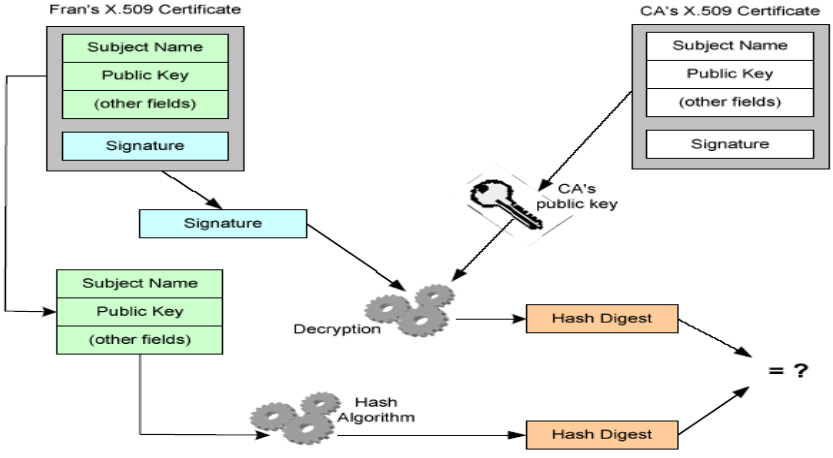
Slide 118

Quá trình ký chứng nhận



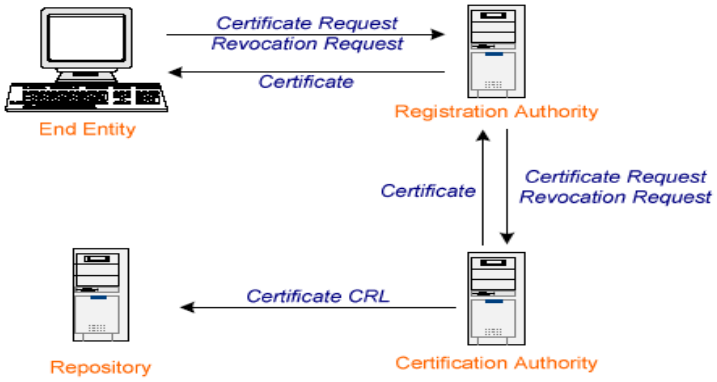
Slide 119

Quá trình kiểm tra chứng nhận



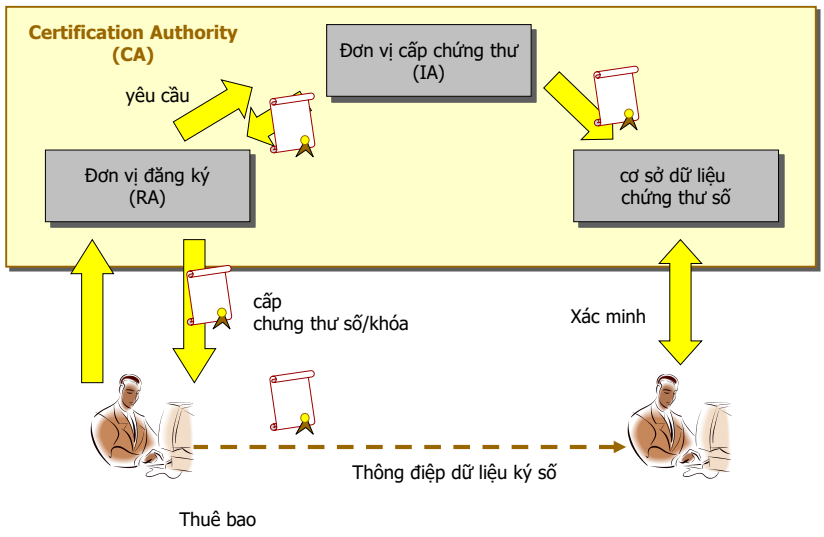
Slide 120

Các thành phần của một cở sở hạ tầng khóa công cộng



Slide 121

Mô hình cơ bản



Slide 122

Tổ chức chứng nhận – Certificate Authority (CA)

- Tổ chức CA là một thực thể quan trọng duy nhất trong X.509 PKI. (Public key Infrastructure).
- Tổ chức CA có nhiệm vụ phát hành, quản lý và hủy bỏ các giấy chứng nhận.

Slide 123

Mô tả

- Để thực hiện nhiệm vụ phát hành giấy chứng nhận của mình, CA nhận yêu cầu chứng nhận từ khách hàng.
- Sau đó, tổ chức CA tạo ra nội dung chứng nhận mới cho khách hàng và ký nhận cho chứng nhận đó.
- Nếu CA có sử dụng nơi lưu trữ chứng nhận thì nó sẽ lưu giấy chứng nhận mới được tạo ra này ở đó.

Slide 124

Tổ chức đăng ký chứng nhận – Registration Authority (RA)

- Một RA là một thực thể tùy chọn được thiết kế để chia sẻ bớt công việc trên CA.
- Một RA không thể thực hiện bất kỳ một dịch vụ nào mà tổ chức CA của nó không thực hiện được

Slide 125

Mô tả

- Các nhiệm vụ chính của RA có thể được chia thành các loại:
  - Các dịch vụ chứng nhận.
  - Các dịch vụ kiểm tra.

Slide 126

Nhận xét

- Một RA hoạt động như là một xử lý ngoại vi của CA.
- Một RA chỉ nên phục vụ cho một CA. Trong khi đó, một CA có thể được hỗ trợ bởi nhiều RA.

Slide 127

Kho lưu trữ chứng nhận – Certificate Repository (CR)

- Một kho chứng nhận là một cơ sở dữ liệu chứa các chứng nhận được phát hành bởi một CA.
- Kho có thể được tất cả các người dùng của PKI.

Slide 128

Chu trình quản lý giấy chứng nhận

- ***Khởi tạo***
- ***Yêu cầu về giấy chứng nhận***
- ***Tạo lại chứng nhận***
- ***Hủy bỏ chứng nhận***
- ***Lưu trữ và khôi phục khóa***

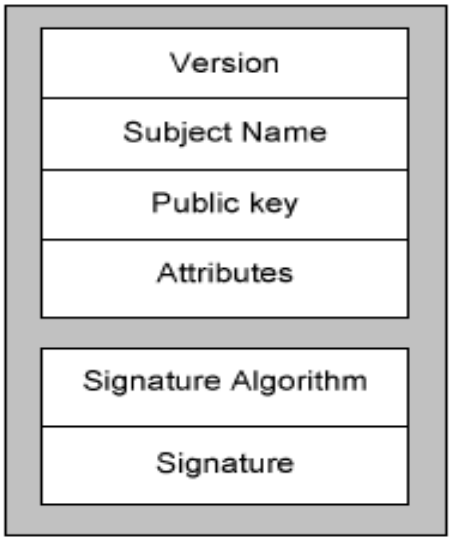
Slide 129

Yêu cầu về giấy chứng nhận

- Hầu hết các CA sử dụng một trong hai phương thức tiêu chuẩn của yêu cầu chứng nhận : PKCS #10 và CRMF.

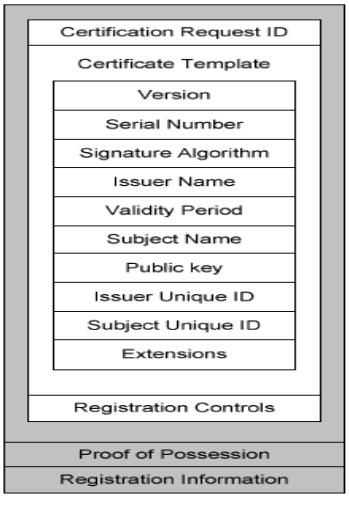
Slide 130

Mẫu yêu cầu chứng nhận theo chuẩn PKCS#10



Slide 131

Định dạng thông điệp yêu cầu chứng nhận theo RFC 2511



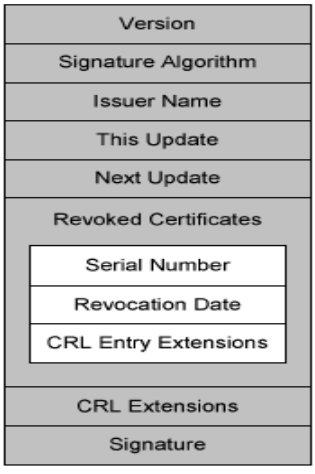
Slide 132

Hủy bỏ chứng nhận

- Certificate Revocation List (CRL) là cách đầu tiên và thông dụng nhất để phổ biến thông tin hủy bỏ.
- CRL chứa thông tin thời gian nhằm xác định thời điểm tổ chức CA phát hành nó.
- CA ký CRL với cùng khóa bí mật được dùng để ký các chứng nhận.

Slide 133

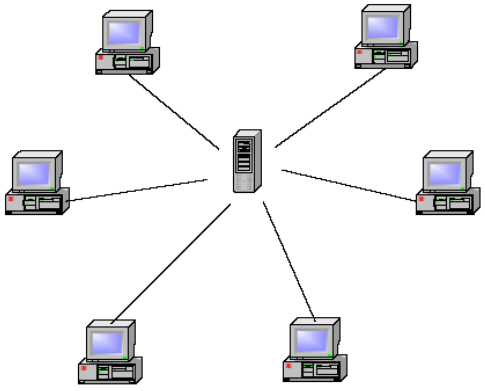
Phiên bản 2 của định dạng danh sách chứng nhận bị hủy



Slide 134

Các mô hình CA

- Mô hình tập trung



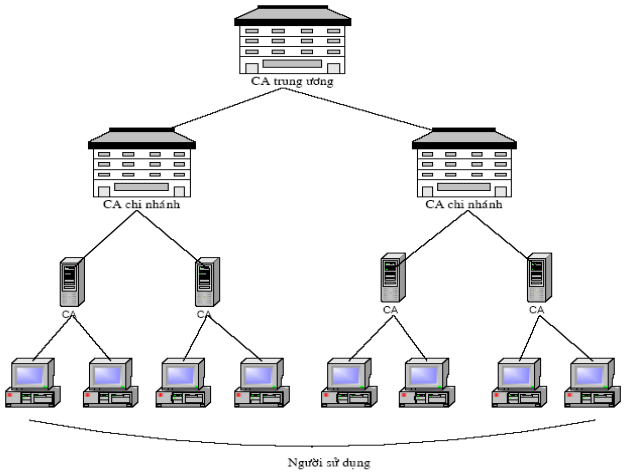
Slide 135

Nhận xét

- Tất cả mọi chứng nhận khóa công cộng đều được ký tập trung bởi tổ chức CA và có thể được xác nhận bằng khóa công cộng của CA.
- Khuyết điểm chính của mô hình này là hiện tượng "nút cổ chai" tại trung tâm.

Slide 136

Mô hình phân cấp



Slide 137

Nhận xét

- Tổ chức CA được phân ra thành nhiều cấp, tổ chức CA ở cấp cao hơn sẽ ký vào chứng nhận khóa công cộng của các tổ chức CA con trực tiếp của mình.
- Một chứng nhận khóa công cộng của người sử dụng sẽ được ký bởi một tổ chức CA cục bộ.

Slide 138

Câu hỏi và bài tập

1. Liệt kê các cách mà secret key có thể được phân phối cho hai bên giao tiếp.
2. Khóa của Distribution Center là gì?
3. Cho biết trách nhiệm của một KDC
4. Session Key là gì và chỉ ra một KDC có thể tạo một session key giữa Alice và Bob
5. Kerberos là gì và tên server của nó; mô tả trách nhiệm của từng Server.
6. Thế nào là tấn công Man-in-the-middle
7. Giao thức Station-to-Station là gì, cho biết mục đích của nó
8. CA là gì, mối quan hệ của nó vỡ mã hóa khóa công khai

Slide 139

Trân trọng cảm ơn!

Slide 140