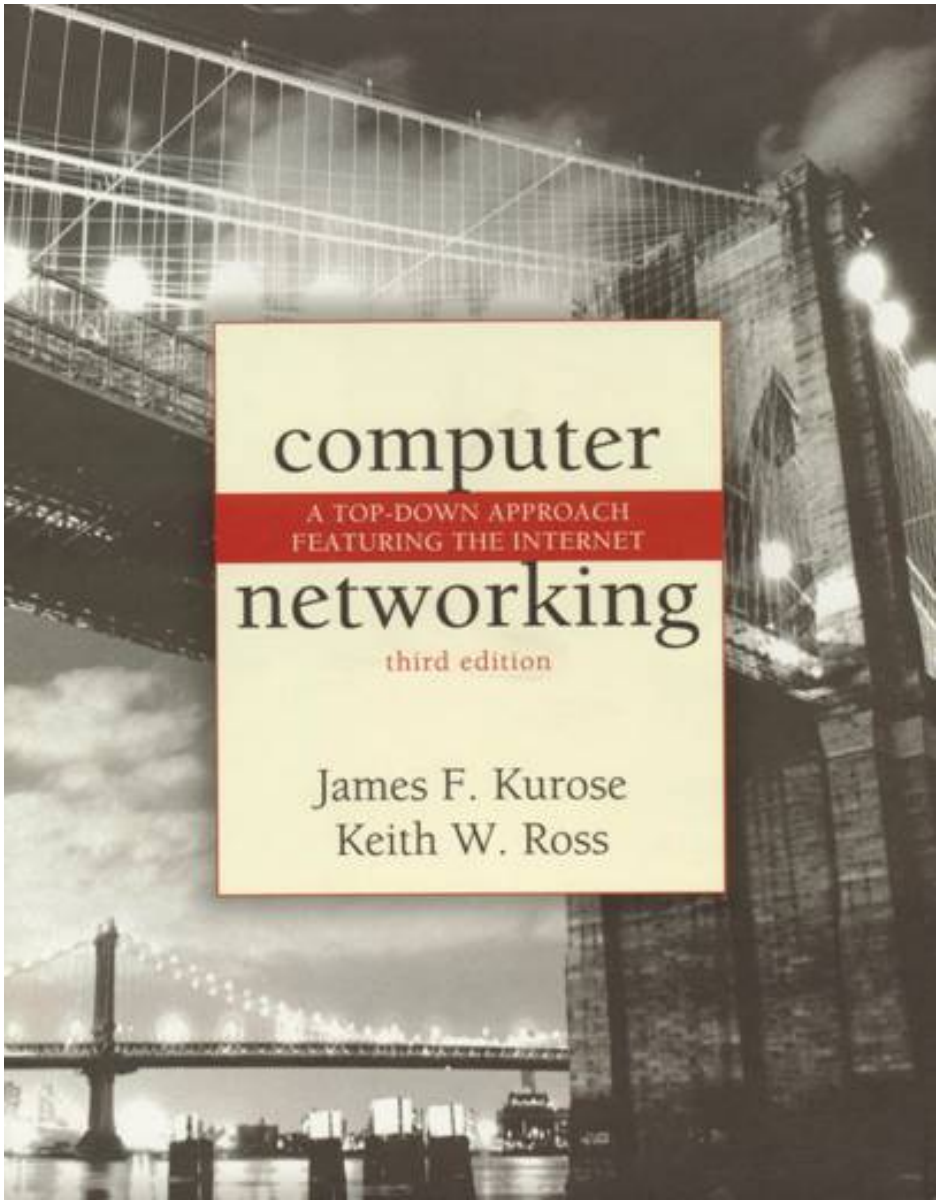


Chương 7
Bảo mật mạng

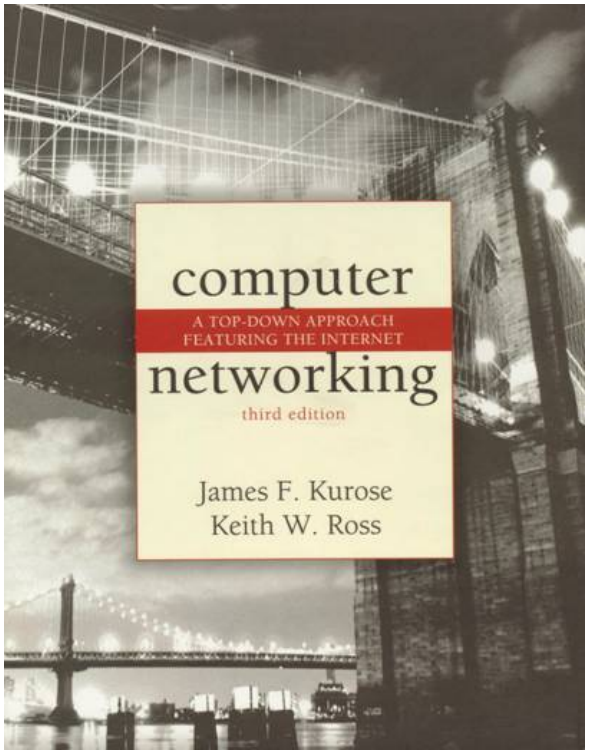
Computer Networking: A Top Down Approach Featuring the Internet, 3rd edition.
Jim Kurose, Keith Ross
Addison-Wesley, July 2004.

Slide này được biên dịch sang tiếng Việt theo sự cho phép của các tác giả

Chương 7: Bảo mật mạng

Mục tiêu:

- hiểu các nguyên lý của bảo mật mạng:
  - mật mã
  - chứng thực
  - tính toàn vẹn
  - khóa phân bố
- bảo mật trong thực tế:
  - các firewall
  - bảo mật trong các lớp application, transport, network, link



7.1 Bảo mật mạng là gì?

Bảo mật mạng là gì?

Sự bảo mật: chỉ có người gửi, người nhận mới "hiểu" được nội dung thông điệp

- người gửi mã hóa thông điệp
- người nhận giải mã thông điệp

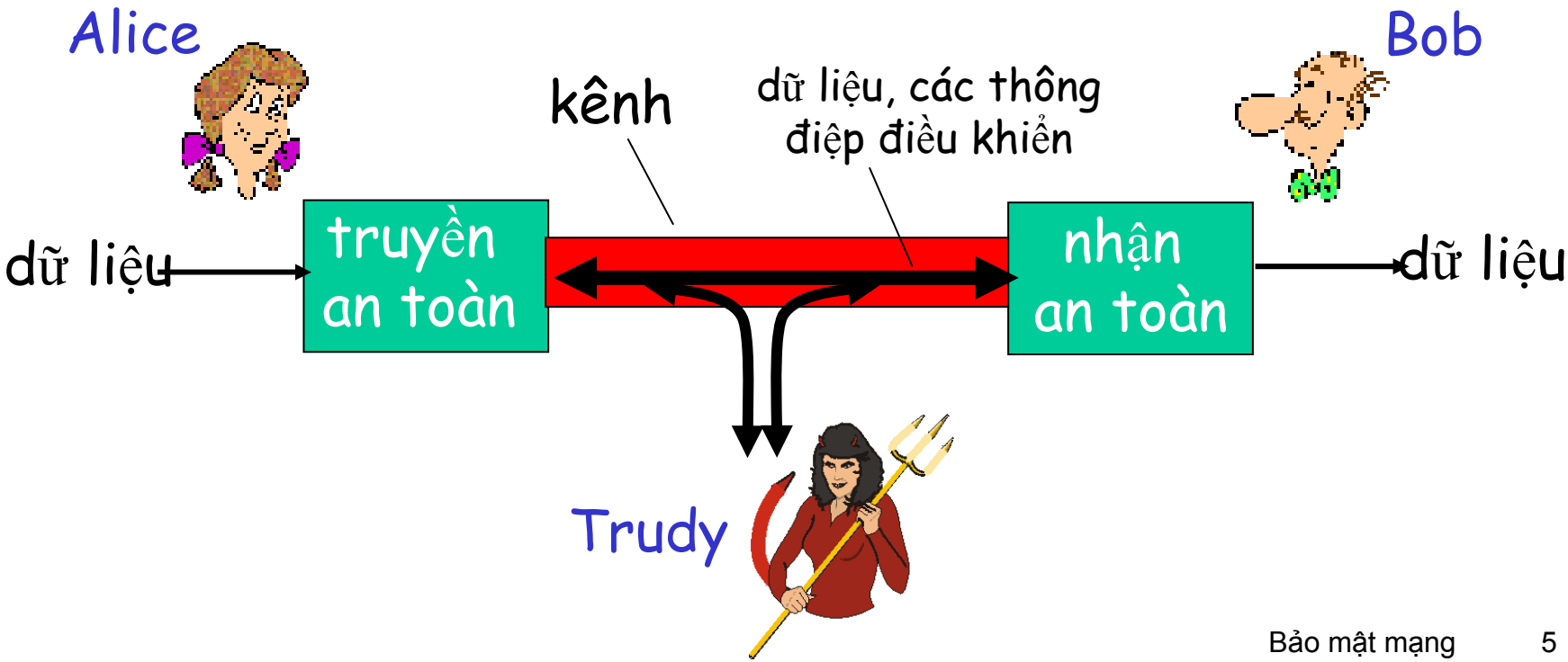
Chứng thực: người gửi, người nhận xác định là nhận ra nhau

Sự toàn vẹn thông điệp: người gửi, người nhận muốn bảo đảm thông điệp không bị thay đổi (trên đường truyền hoặc sau khi nhận) mà không bị phát hiện

Truy cập & tính sẵn sàng: các dịch vụ phải có khả năng truy cập và sẵn sàng đối với các user

Bạn và kẻ thù: Alice, Bob, Trudy

- Bob, Alice (bạn bè) muốn truyền thông "an toàn"
- Trudy (kẻ xâm nhập) có thể ngăn chặn, xóa, thêm các thông điệp



Bob, Alice có thể là những ai?

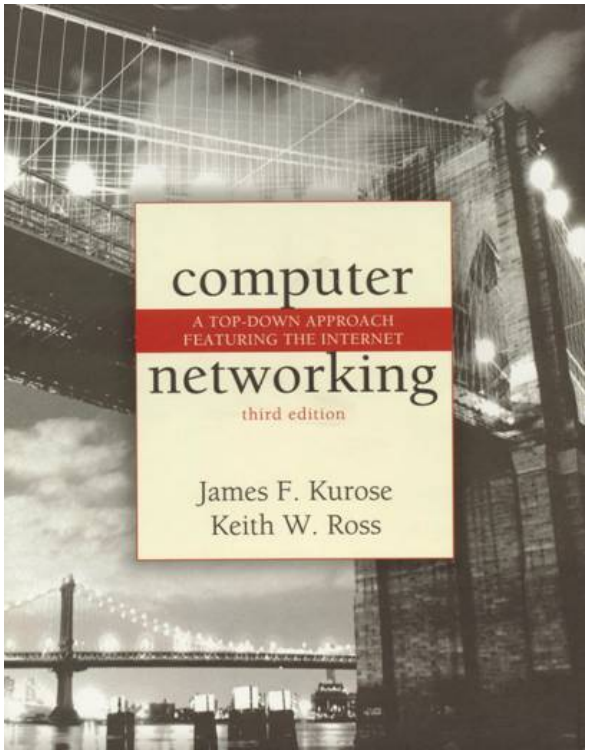
- trình duyệt Web/server cho các giao dịch điện tử
- client/server ngân hàng trực tuyến
- DNS servers
- các router trao đổi thông tin cập nhật bảng routing
- .v.v.

Bạn và kẻ thù

Hỏi: Kẻ xấu có thể làm những việc gì?

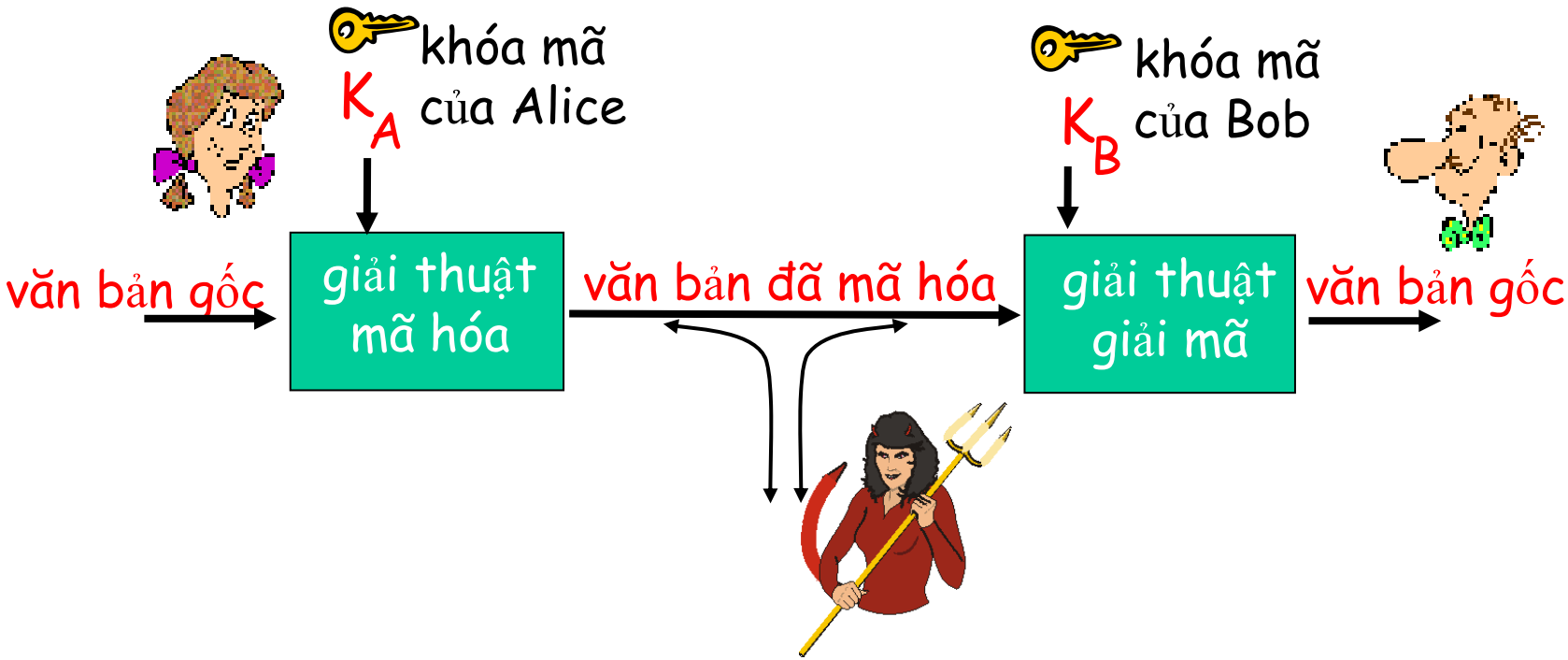
Đáp: rất nhiều!

- *nghe lén:* ngăn chặn các thông điệp
- kích hoạt *chèn* các thông điệp vào trong kết nối
- *giả danh:* có thể giả mạo địa chỉ nguồn trong gói (hoặc bất kỳ trường nào trong đó)
- *cướp:* "tiếp tục" kết nối hiện hành nhưng thay người gửi hoặc người nhận bằng chính họ
- *từ chối dịch vụ:* dịch vụ hiện tại bị người khác dùng (đồng nghĩa quá tải)
- .v.v.



7.2 Các nguyên lý mã hóa

Ngôn ngữ mã hóa



khóa đối xứng: khóa bên gửi và bên nhận giống nhau

khóa công cộng: khóa mã chung, khóa giải mã bí mật (riêng)

Mã hóa khóa đối xứng

mật mã thay thế: thay thứ này thành thứ khác

- mã hóa ký tự đơn: thay thế từng ký tự một

văn bản gốc: abcdefghijklmnopqrstuvwxyz

văn bản đã mã hóa: mnbvcxzasdfghjklpoiuytrewq

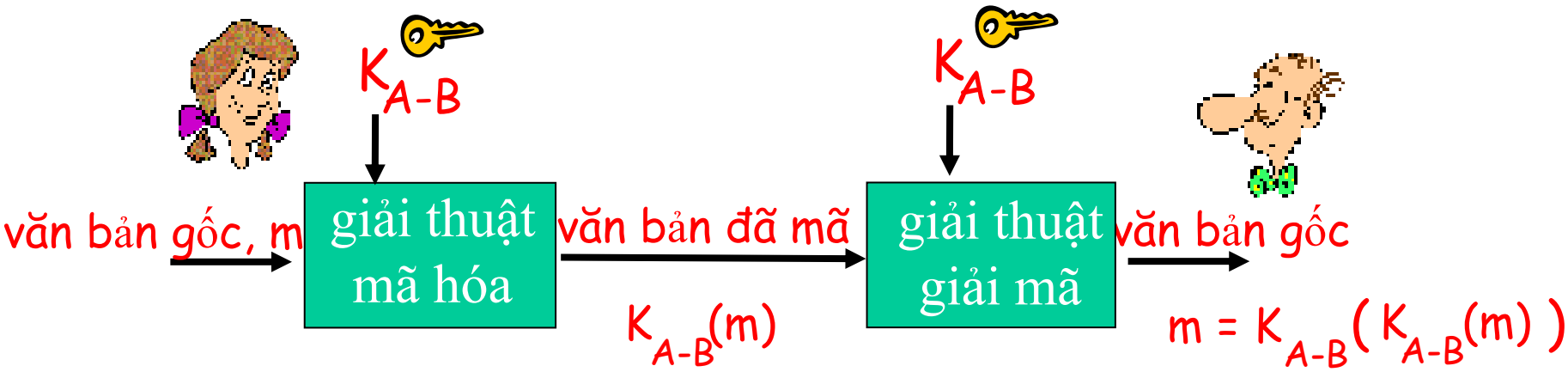
ví dụ: văn bản gốc: bob. i love you. alice

mã hóa thành: nkn. s gktc wky. mgsbc

Hỏi: Bẻ khóa kiểu mã hóa đơn giản này dễ không?

- brute force (khó như thế nào?)
- khác?

Mã hóa khóa đối xứng



khóa đối xứng: Bob và Alice cùng biết (đối xứng) khóa: K_{A-B}

- vd: khóa được biết trong mẫu mã hóa ký tự đơn
- Hỏi: Làm thế nào Bob và Alice thỏa thuận giá trị khóa?

Mã hóa khóa đối xứng: DES

DES: Data Encryption Standard

- chuẩn mã hóa của Hoa Kỳ [NIST 1993]
- khóa đối xứng 56-bit, văn bản gốc vào 64-bit
- Bảo mật trong DES như thế nào?
  - chưa có cách tiếp cận "backdoor-cửa sau" để giải mã
- làm cho DES bảo mật hơn:
  - dùng 3 khóa tuần tự (3-DES) trong mỗi datum
  - dùng cơ chế liên kết khối mã

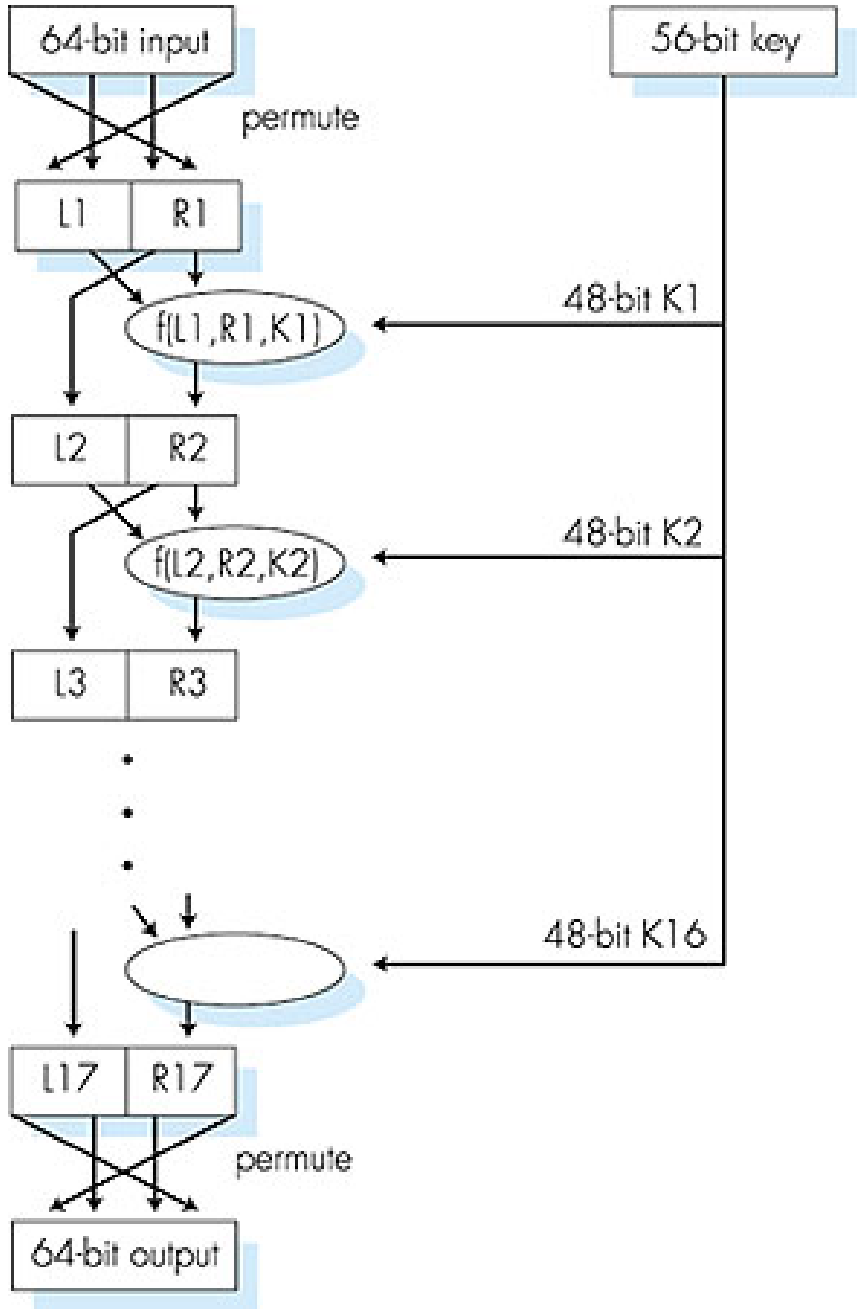
Mã hóa khóa đối xứng: DES

DES hoạt động

hoán vị đầu tiên

16 vòng giống nhau, mỗi vòng dùng khóa 48 bit khác nhau

hoán vị cuối cùng



AES: Advanced Encryption Standard

- chuẩn NIST khóa đối xứng mới (tháng 11-2001) thay thế cho DES
- dữ liệu xử lý từng khối 128 bit
- các khóa 128, 192, hoặc 256 bit
- giải mã brute force (thử sai) tốn 1s với DES, tốn 149 tỷ tỷ năm với AES

Mã hóa khóa công cộng

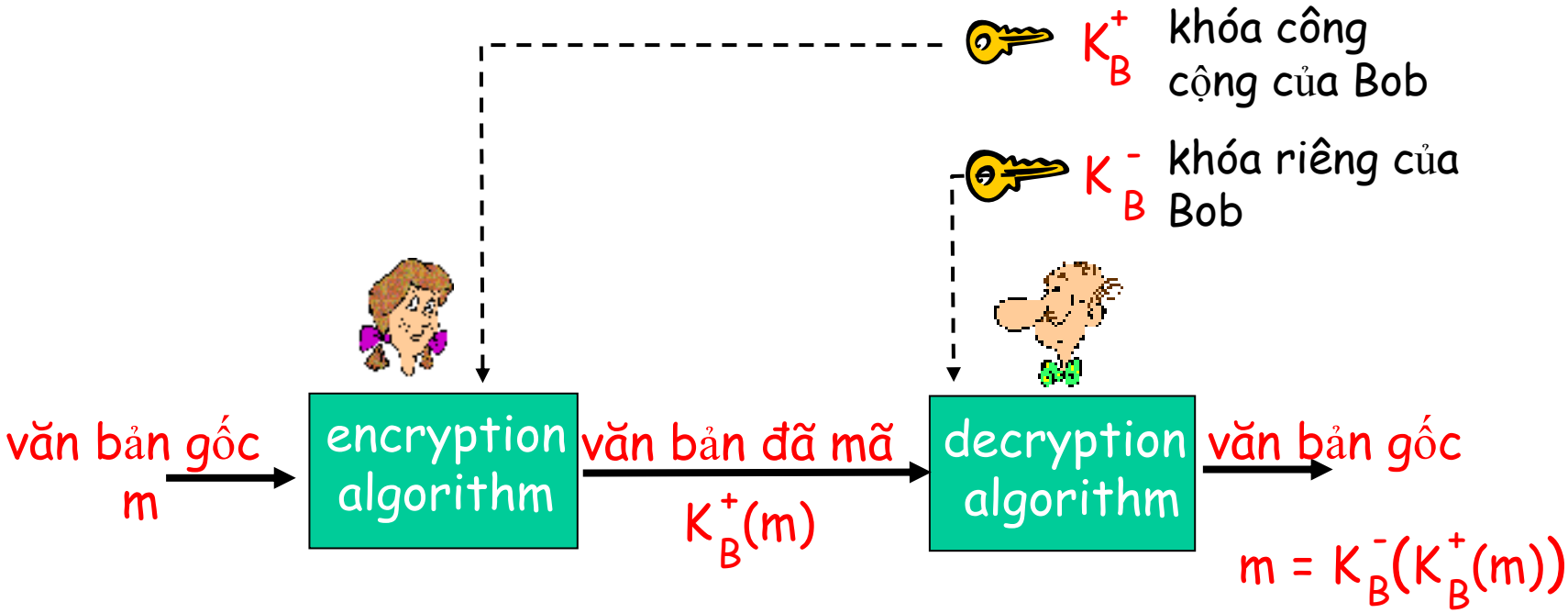
khóa đối xứng

- yêu cầu người gửi, người nhận phải biết khóa công cộng
- Hỏi: làm sao biết khóa công cộng đó trong lần đầu tiên (đặc biệt với những người chưa bao giờ gặp trước)?

Mã hóa khóa công cộng

- tiếp cận khác hoàn toàn
- người gửi, người nhận không chia sẻ khóa công cộng
- khóa công cộng cho mọi người đều biết
- khóa giải mã riêng chỉ có người nhận biết

Mã hóa khóa công cộng



Giải thuật mã hóa khóa công cộng

Yêu cầu:

1. cần $K_B^+(\cdot)$ và $K_B^-(\cdot)$ như sau:

$$K_B^-(K_B^+(m)) = m$$

2. cho khóa công cộng K_B^+, nó phải không thể tính toán ra được khóa riêng K_B^-

giải thuật RSA: Rivest, Shamir, Adelson

RSA: chọn các khóa

1. Chọn 2 số nguyên tố lớn p, q (mỗi số có thể đến 1024 bit)

2. Tính $n = pq$, $z = (p-1)(q-1)$

3. Chọn e (với $e<n$) sao cho không có thừa số chung với z. (e, z là nguyên tố cùng nhau).

4. Chọn d sao cho $ed-1$ chia hết cho z.
   (nói cách khác: $ed \bmod z = 1$).

5. *khóa công cộng* là (n,e). *khóa riêng* là (n,d).

K_B^+ K_B^-

RSA: mã hóa, giải mã

0. Cho (n,e) và (n,d) theo tính toán ở trên

1. Để mã hóa mẫu bit, m, tính:

$$c = m^e \bmod n$$

2. Để giải mã mẫu bit, c, tính:

$$m = c^d \bmod n$$

$$m = (\underbrace{m^e \bmod n}_{c})^d \bmod n$$

RSA ví dụ:

Bob chọn *p=5, q=7*. Thì *n=35, z=24*.

e=5 (vì *e, z* nguyên tố cùng nhau).

d=29 (vì *ed-1* chia hết cho z)

mã hóa:

ký tự	m	m^e	$c = m^e \bmod n$
l	12	1524832	17

giải mã:

c	c^d	$m = c^d \bmod n$	ký tự
17	481968572106750915091411825223071697	12	l

RSA: Tại sao là $m = (m^e \bmod n)^d \bmod n$

Kết quả của lý thuyết "số hữu ích": Nếu p,q nguyên tố và $n = pq$, thì:

$$x^y \bmod n = x^{y \bmod (p-1)(q-1)} \bmod n$$

$$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n$$

$$= m^{ed \bmod (p-1)(q-1)} \bmod n$$

(dùng lý thuyết số ở trên)

$$= m^1 \bmod n$$

(chọn ed sao cho chúng chia cho $(p-1)(q-1)$ dư 1)

$$= m$$

RSA: đặc tính quan trọng khác

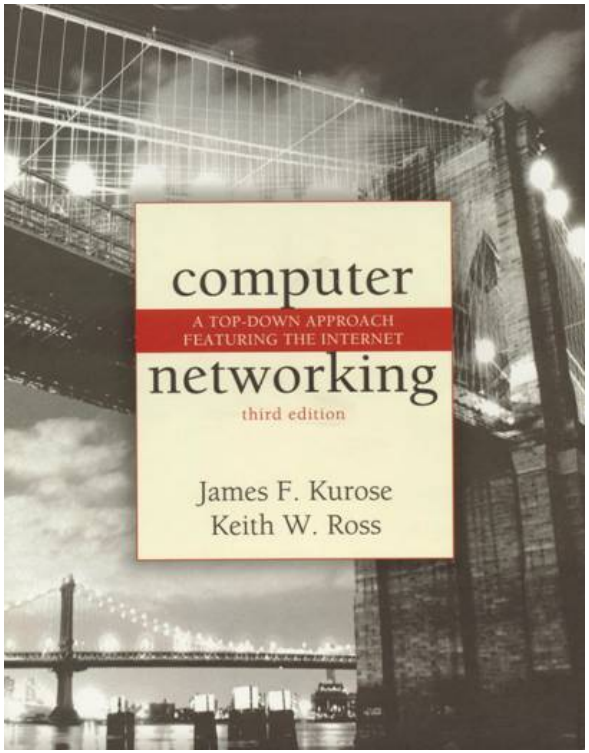
Đặc tính sau đây sẽ rất hữu ích:

$$K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))$$

đầu tiên dùng khóa công cộng, sau đó dùng khóa riêng

đầu tiên dùng khóa riêng, sau đó dùng khóa công cộng

Kết quả giống nhau



7.3 Sự chứng thực

Sự chứng thực

Mục tiêu: Bob muốn Alice "chứng thực" nhân dạng của cô đối với anh ta

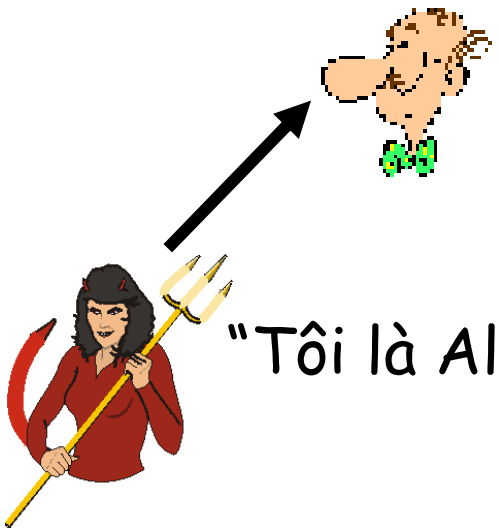
Giao thức ap1.0: Alice nói "Tôi là Alice"

Thất bại sẽ xảy ra??

Sự chứng thực

Mục tiêu: Bob muốn Alice "chứng thực" nhân dạng của cô đối với anh ta

Giao thức ap1.0: Alice nói "Tôi là Alice"

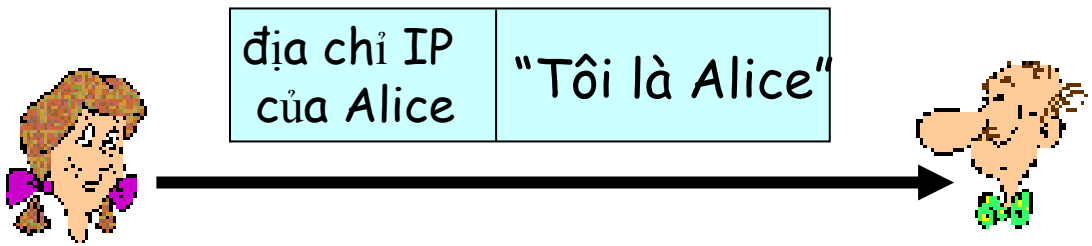


"Tôi là Alice"

Trong mạng, Bob không "nhìn thấy" Alice, vì thế đơn giản Trudy tuyên bố cô ta là Alice

Sự chứng thực: thử lại

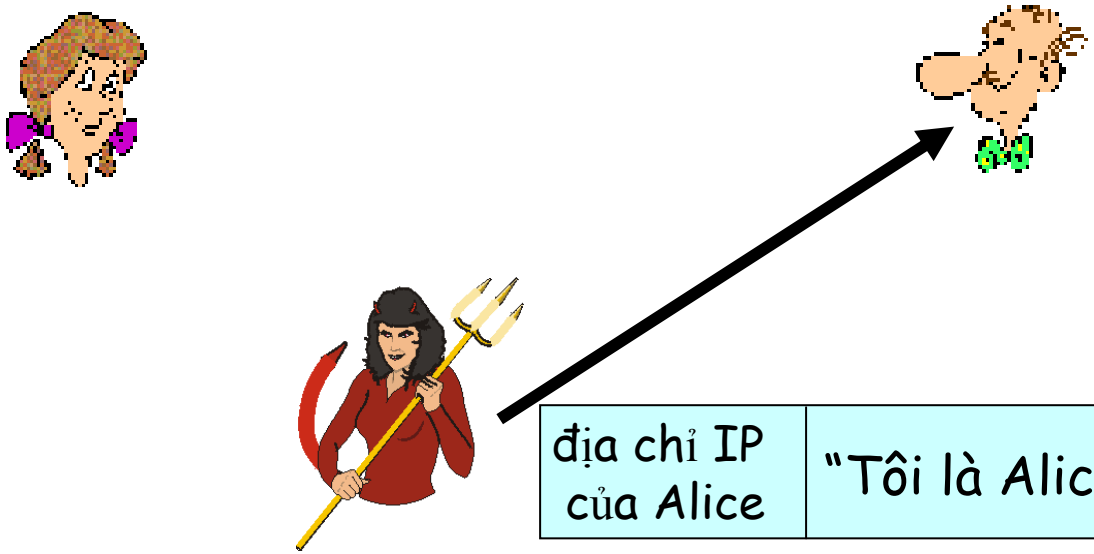
Giao thức ap2.0: Alice nói "Tôi là Alice" trong 1 gói IP chứa địa chỉ IP nguồn của cô ấy



Thất bại sẽ xảy ra??

Sự chứng thực: thử lại

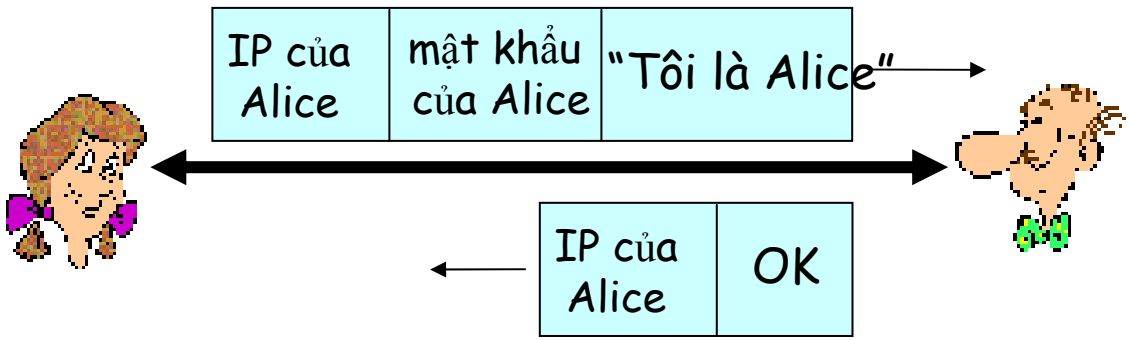
Giao thức ap2.0: Alice nói "Tôi là Alice" trong 1 gói IP chứa địa chỉ IP nguồn của cô ấy



Trudy sẽ tạo 1 gói giả mạo địa chỉ IP của Alice

Sự chứng thực: thử lại

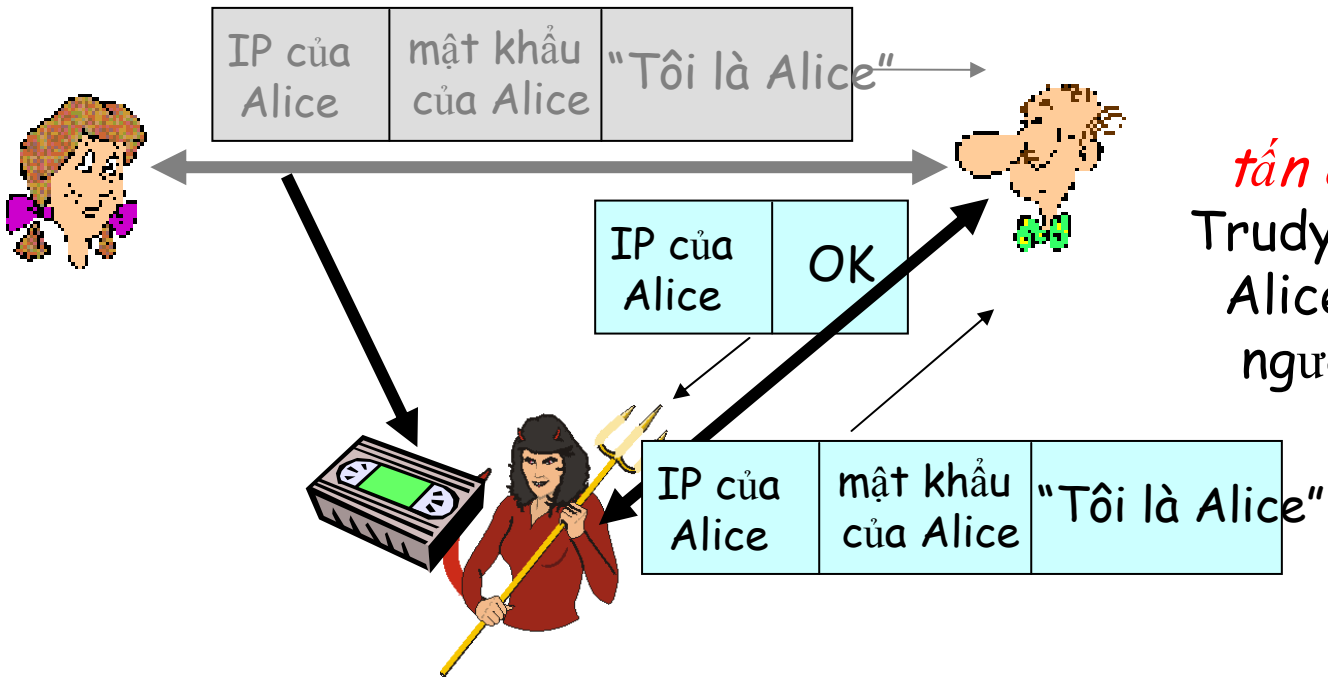
Giao thức ap3.0: Alice nói "Tôi là Alice" và gửi mật khẩu của cô ấy để "chứng thực" nó.



Thất bại sẽ xảy ra??

Sự chứng thực: thử lại

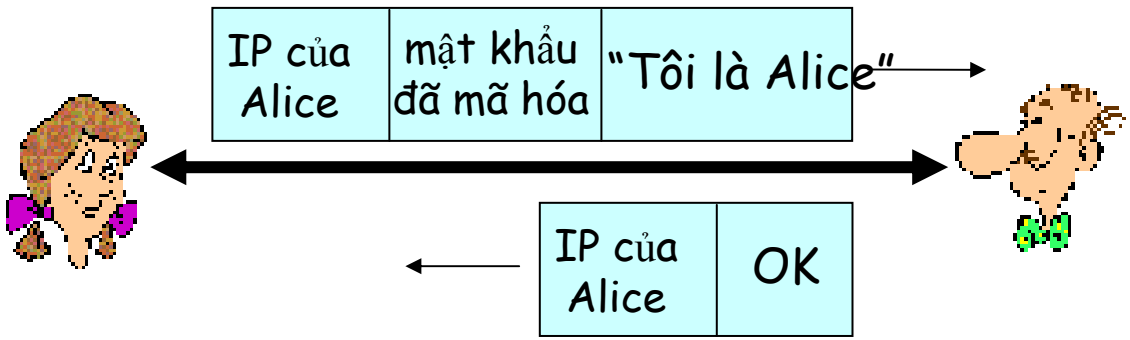
Giao thức ap3.0: Alice nói "Tôi là Alice" và gửi mật khẩu của cô ấy để "chứng thực" nó.



tấn công ngược lại: Trudy ghi nhớ gói của Alice và sau đó gửi ngược lại cho Bob

Sự chứng thực: thử lại

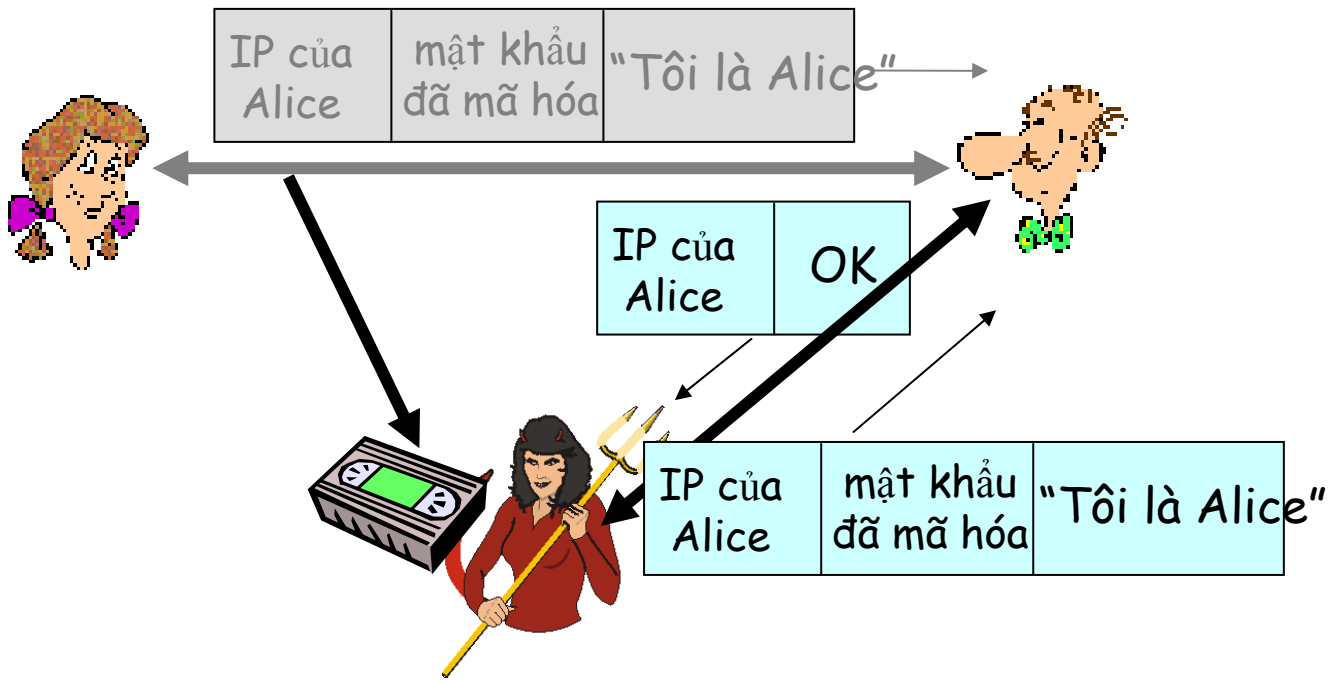
Giao thức ap3.1: Alice nói "Tôi là Alice" và gửi mật khẩu bí mật *đã được mã hóa* để "chứng thực" nó.



Thất bại sẽ xảy ra??

Sự chứng thực: thử lại

Giao thức ap3.1: Alice nói "Tôi là Alice" và gửi mật khẩu bí mật *đã được mã hóa* để "chứng thực" nó.



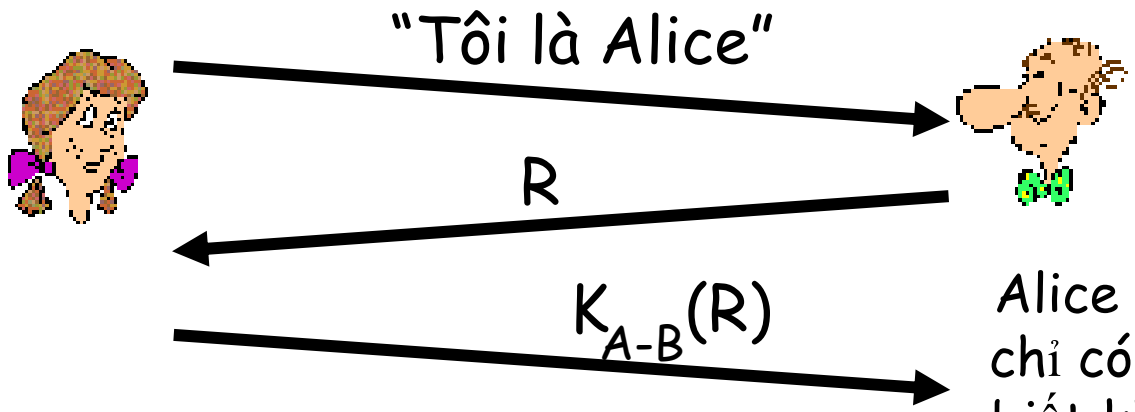
cơ chế ghi nhớ và phát lại vẫn có thể hoạt động!

Sự chứng thực: thử lại

Mục tiêu: tránh cơ chế tấn công kiểu phát lại

Phương pháp: số (R) chỉ dùng 1 lần trong thời gian hoạt động

ap4.0: để chứng thực Alice "sống", Bob gửi Alice số R. Alice phải trả về R đã được mã hóa với khóa bí mật đã chia sẻ



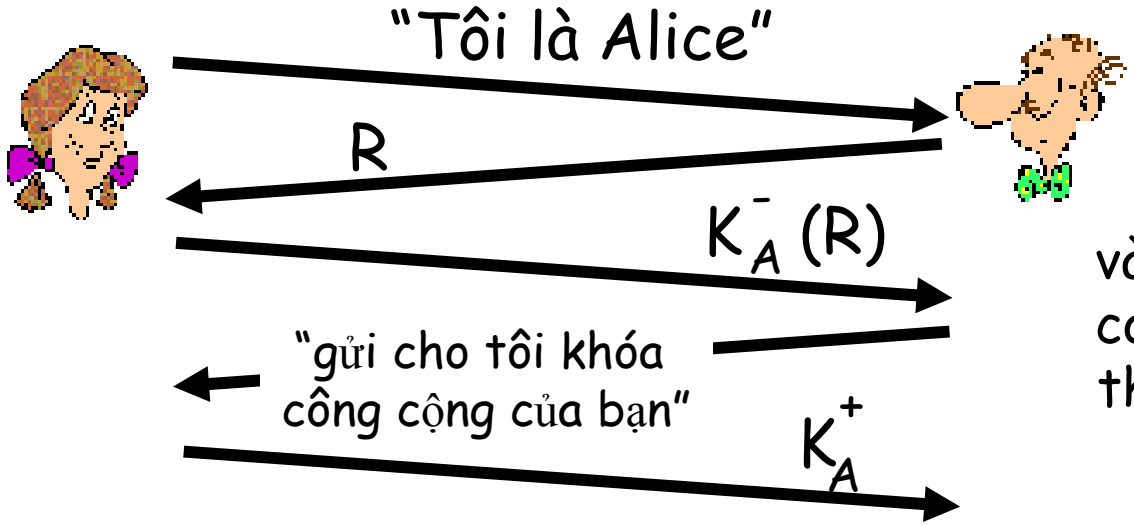
Thất bại, không thuận lợi?

Sự chứng thực: ap5.0

ap4.0 yêu cầu chia sẻ khóa đối xứng

- chúng ta có thể chứng thực dùng các kỹ thuật khóa công cộng?

ap5.0: dùng 1 lần, mã hóa khóa công cộng



Bob tính toán

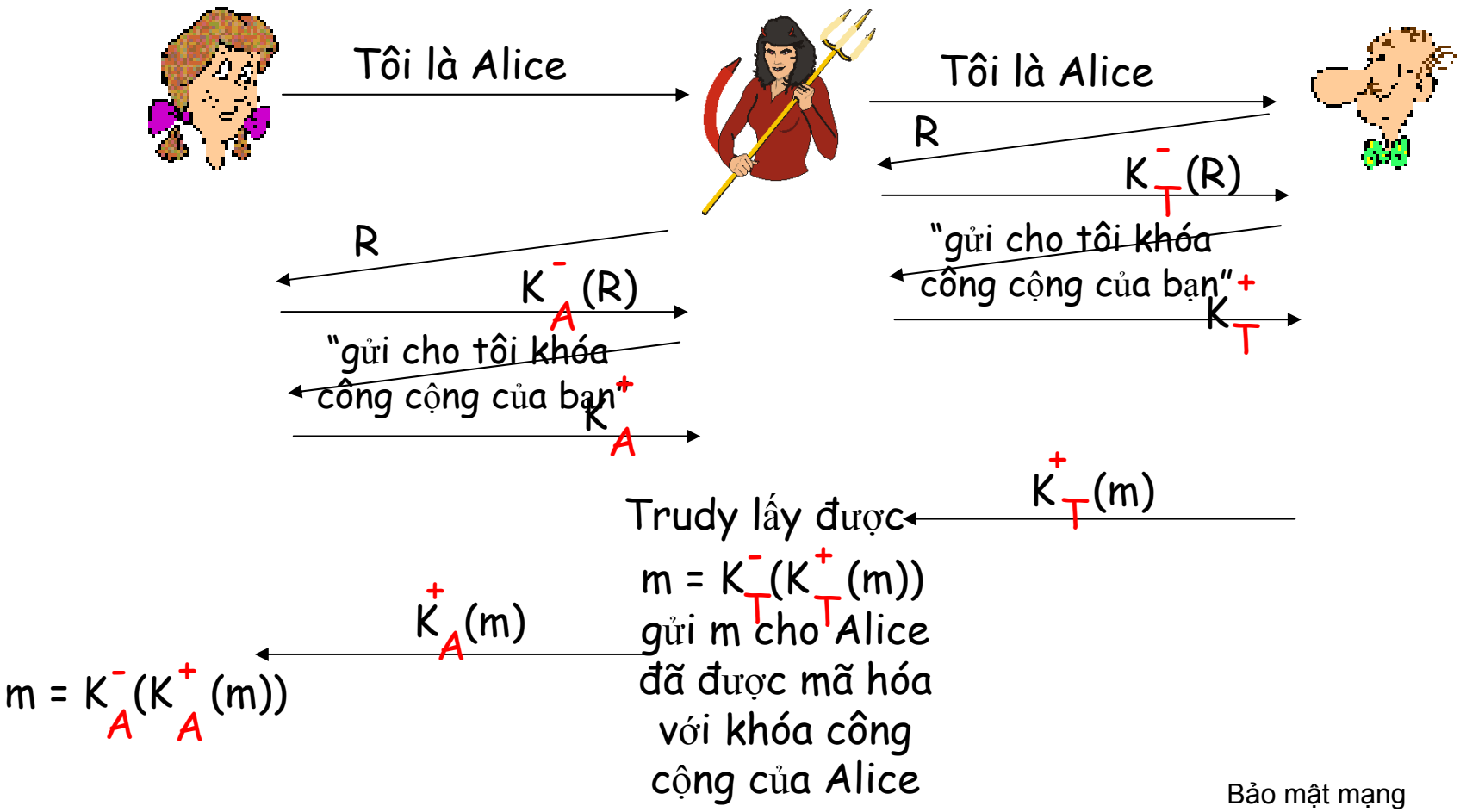
$K_A^+(K_A^-(R)) = R$

và biết chỉ có Alice mới có thể có khóa riêng, vì thế R đã mã hóa phải là

$K_A^+(K_A^-(R)) = R$

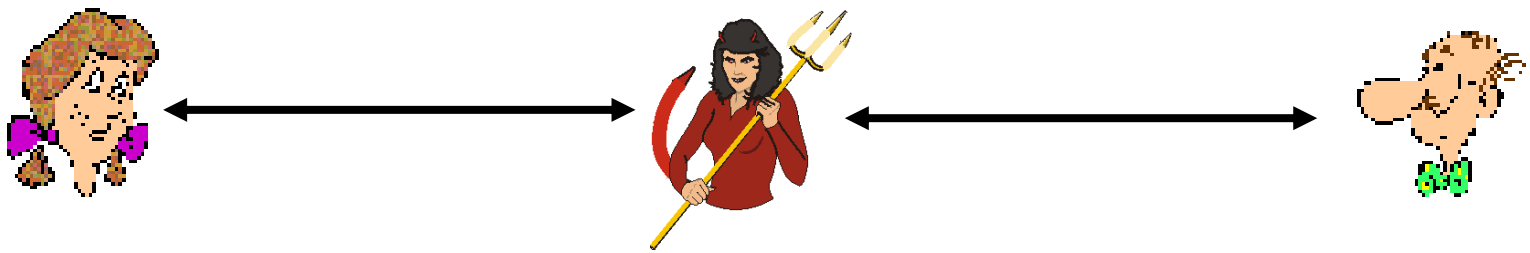
ap5.0: lỗ hổng bảo mật

Kẻ tấn công ẩn: Trudy giả làm Alice (đối với Bob) và như Bob (đối với Alice)



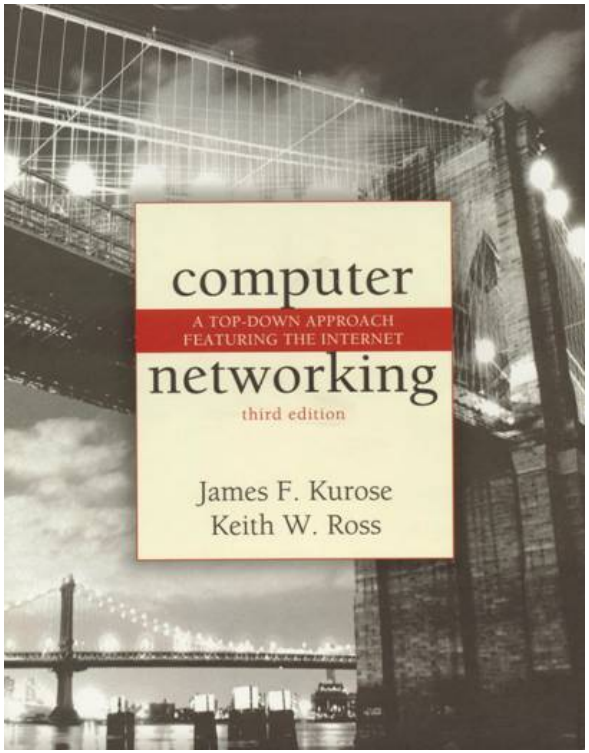
ap5.0: lỗ hổng bảo mật

Kẻ tấn công ẩn: Trudy giả làm Alice (đối với Bob) và như Bob (đối với Alice)



Khó khăn để phát hiện:

- Bob nhận mọi thứ mà Alice gửi và ngược lại
- vấn đề là Trudy cũng nhận tất cả các thông điệp như thế!



7.4 Sự toàn vẹn

Chữ ký số

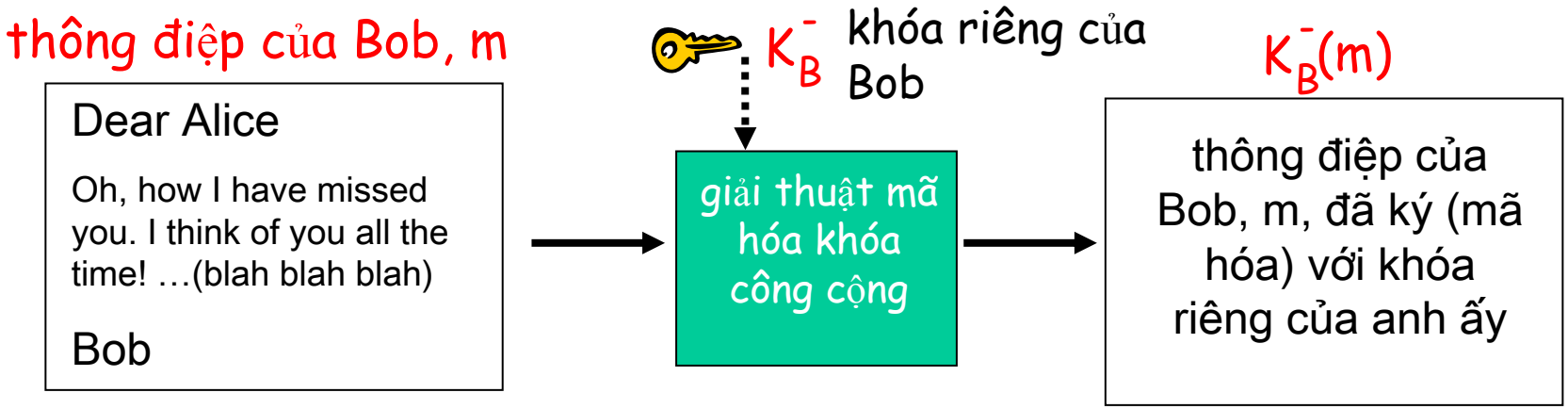
Kỹ thuật mã hóa tương tự như các chữ ký bằng tay.

- người gửi (Bob) đánh dấu (số hóa) tài liệu, thiết lập thuộc tính anh ấy là người sở hữu/tạo lập tài liệu.
- có thể kiểm tra, không thể làm giả: người nhận (Alice) có thể chứng thực với người khác là chỉ có Bob chứ ngoài ra không có ai (kể cả Alice) đã ký trên tài liệu đó.

Chữ ký số

Chữ ký số đơn giản cho thông điệp m:

- Bob ký m bằng cách mã hóa với khóa riêng của anh ấy K_B^-, tạo thông điệp "đã được ký", $K_B^-(m)$



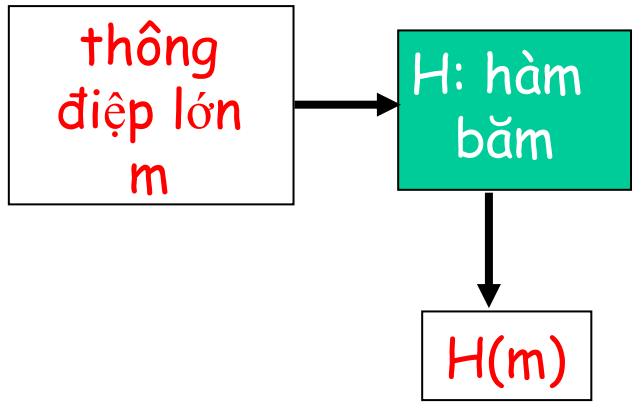
Chữ ký số (tt)

- Giả sử Alice nhận được m, với chữ ký số hóa là $K_B^-(m)$
- Alice kiểm tra m đã được ký bởi Bob bằng cách áp dụng khóa công cộng của Bob K_B^+ cho $K_B^-(m)$ sau đó kiểm tra $K_B^+(K_B^-(m)) = m$.
- Nếu $K_B^+(K_B^-(m)) = m$, bất cứ ai đã ký m phải dùng khóa riêng của Bob

  Alice kiểm tra:
  - ✓ Bob đã ký m.
  - ✓ Không có ai khác đã ký m.
  - ✓ Bob đã ký m và không ký m'.

  Không thể bác bỏ:
  - ✓ Alice có thể giữ m, và chữ ký $K_B^-(m)$ để chứng thực rằng Bob đã ký m.

Phân loại thông điệp



Tính toán các thông điệp dài có chi phí đắt

Mục tiêu: "dấu tay" số hóa có kích thước cố định, dễ tính toán được

- áp dụng hàm băm H vào m, tính được phân loại thông điệp kích thước cố định, *H(m)*.

Các đặc tính hàm băm:

- nhiều-một
- sinh ra phân loại thông điệp kích thước cố định ("dấu tay")
- cho phân loại thông điệp x, không thể tính toán để tìm m dùng x = H(m)

Internet checksum: hàm băm

Internet checksum có một số đặc tính của hàm băm:

- ✓ sinh ra các phân loại độ dài cố định (tổng 16 bit) của thông điệp
- ✓ là nhiều-một

nhưng với thông điệp và giá trị băm cho trước, dễ dàng tìm được thông điệp khác có cùng giá trị băm:

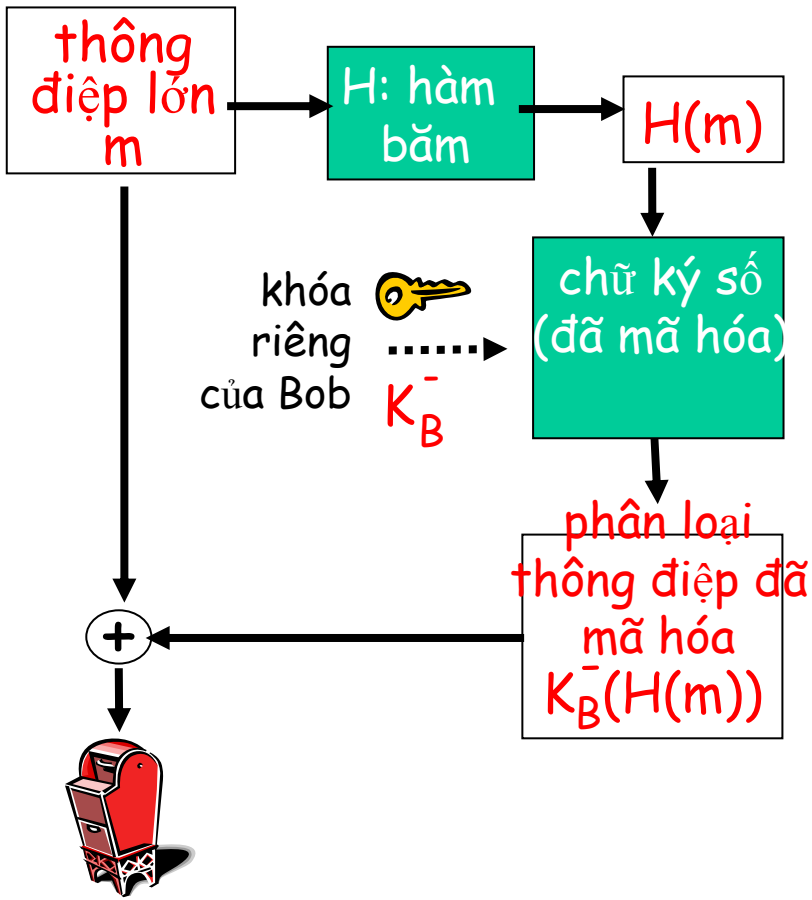
thông điệp	ASCII format
I O U 1	49 4F 55 31
0 0 . 9	30 30 2E 39
9 B O B	39 42 4F 42
	B2 C1 D2 AC

thông điệp	ASCII format
I O U 9	49 4F 55 39
0 0 . 1	30 30 2E 31
9 B O B	39 42 4F 42
	B2 C1 D2 AC

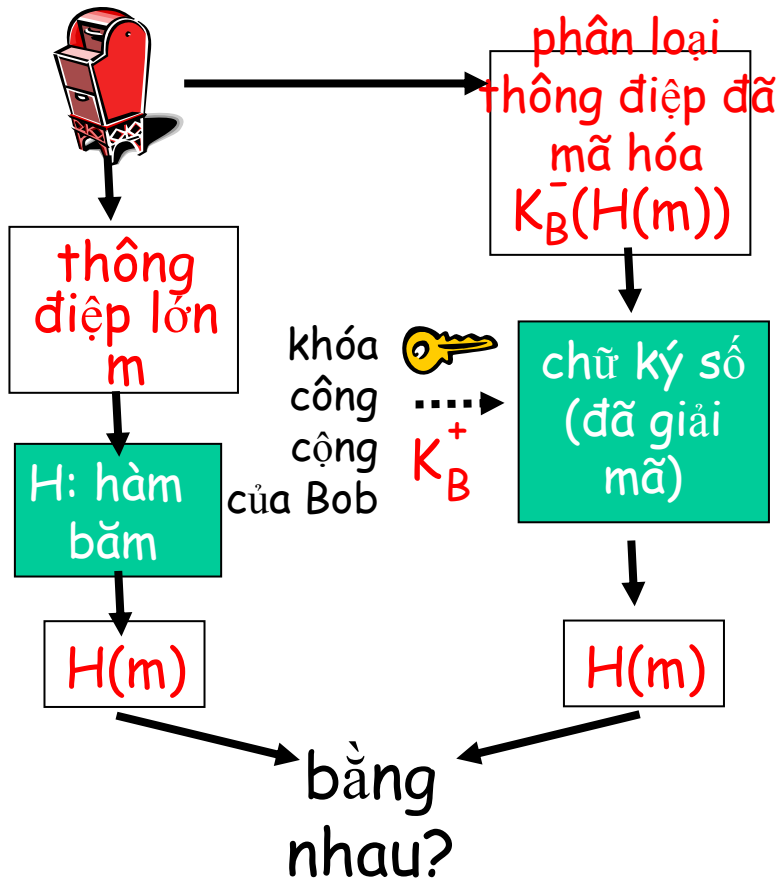
các thông điệp khác nhưng checksum giống nhau!

chữ ký số = phân loại thông điệp đã ký

Bob gửi thông điệp đã ký số hóa:

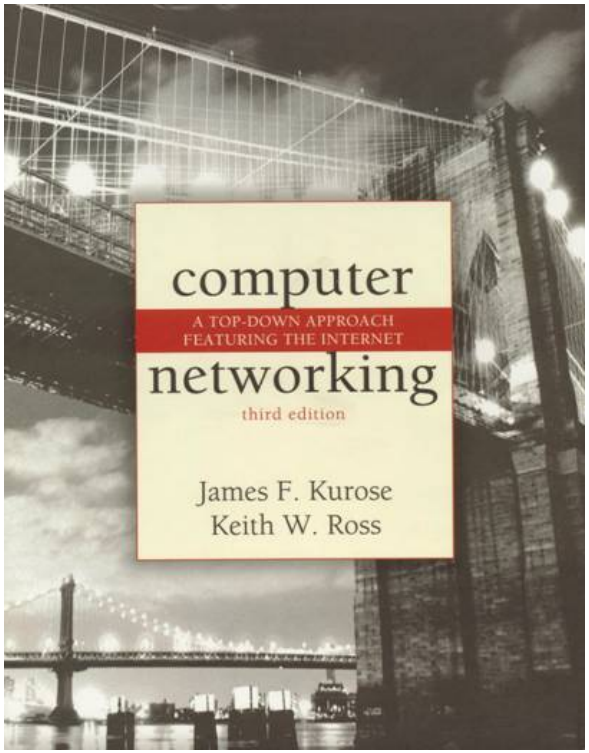


Alice kiểm tra chữ ký và sự toàn vẹn của thông điệp đã ký bằng số hóa:



Các giải thuật hàm băm

- hàm băm MD5 sử dụng rất phổ biến (RFC 1321)
  - tính phân loại thông điệp 128 bit trong tiến trình 4 bước
  - với chuỗi x có 128 bit bất kỳ, khó khăn để xây dựng thông điệp m sao cho băm MD5 của nó bằng với x
- SHA-1 cũng được dùng.
  - chuẩn của Mỹ [NIST, FIPS PUB 180-1]
  - phân loại thông điệp 160-bit



7.5 Khóa phân bố và chứng chỉ

Các trung gian được tin cậy

Vấn đề khóa đối xứng:

- Làm thế nào 2 thực thể cùng thiết lập khóa bí mật trên mạng?

Giải pháp:

- Trung tâm phân bố khóa (key distribution center-KDC) được tin cậy – hoạt động trung gian giữa các thực thể

Vấn đề khóa công cộng:

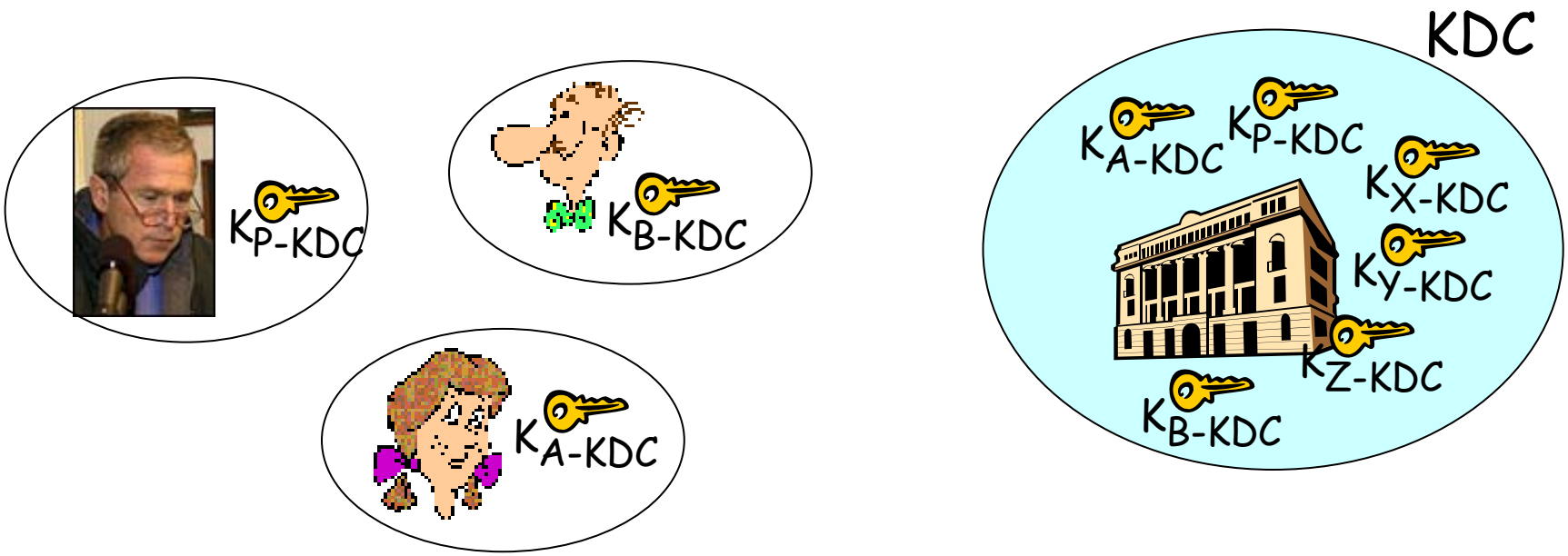
- Khi Alice lấy được khóa công cộng của Bob (từ web site, email, đĩa), làm sao biết khóa công cộng của Bob chứ không phải của Trudy?

Giải pháp:

- nơi cấp chứng chỉ (certification authority-CA) được tin cậy

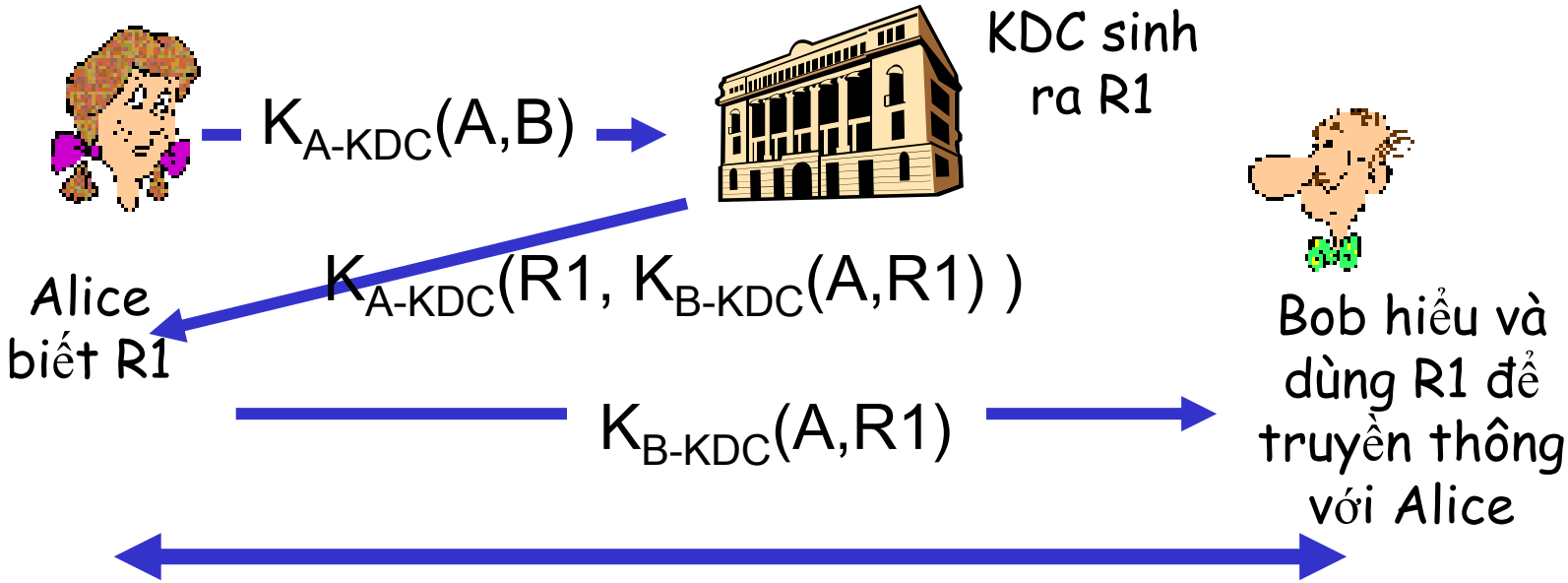
Key Distribution Center (KDC)

- Alice, Bob cần khóa đối xứng được chia sẻ.
- KDC: chia sẻ khóa bí mật khác nhau với mỗi người dùng đã đăng ký
- Alice, Bob biết các khóa đối xứng của họ, $K_{A\text{-}KDC}$ $K_{B\text{-}KDC}$, để truyền thông với KDC.



Key Distribution Center (KDC)

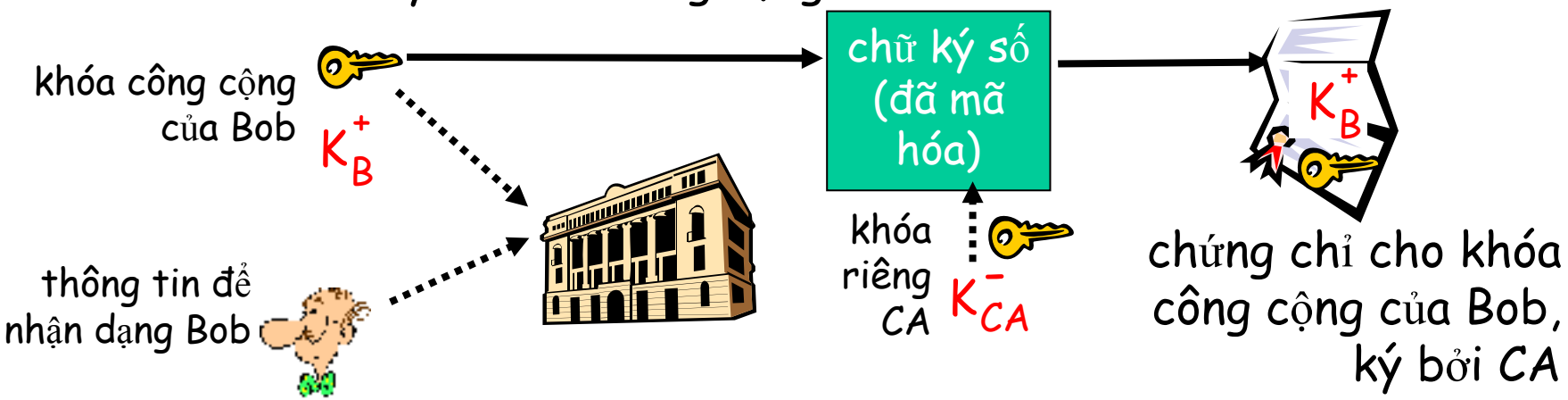
Hỏi: Làm sao KDC cho phép Bob, Alice xác định khóa bí mật đối xứng để truyền thông với nhau?



Alice và Bob truyền thông: dùng R1 như *khóa phiên làm việc* để chia sẻ mã hóa đối xứng

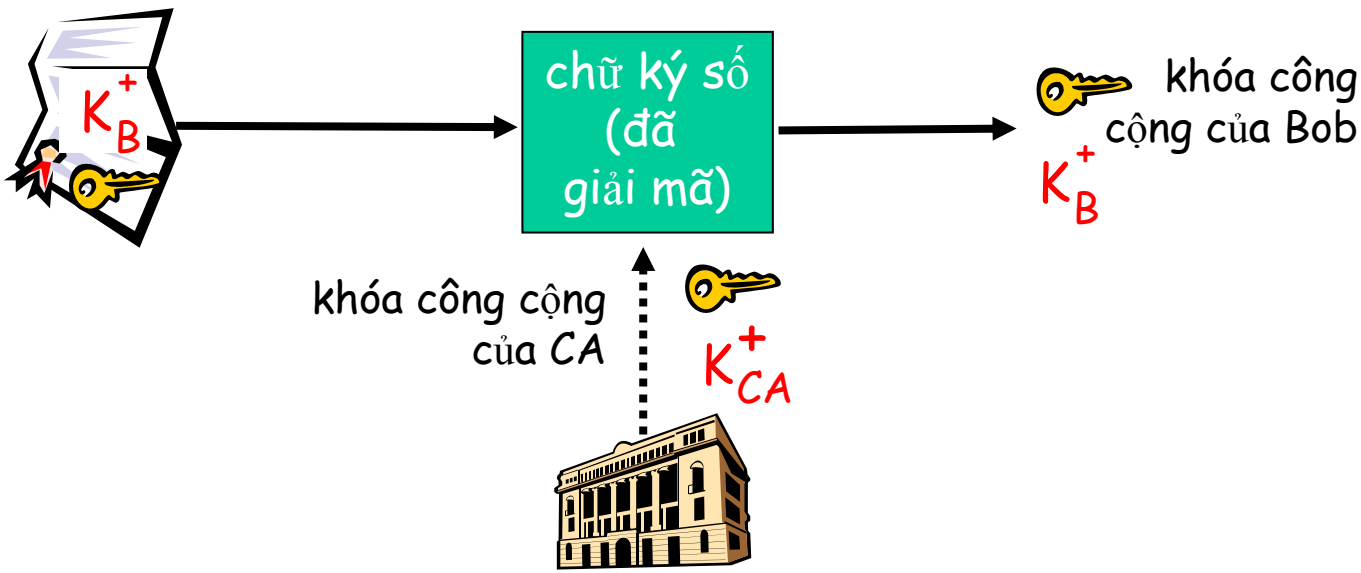
Cấp chứng chỉ

- Certification authority (CA): gắn kết khóa công cộng với thực thể E nào đó.
- E (người, router) đăng ký khóa công cộng của họ với CA.
  - E cung cấp "bằng chứng để nhận dạng" cho CA.
  - CA tạo ra chứng chỉ ràng buộc E với khóa công cộng của nó.
  - chứng chỉ chứa khóa công cộng của E được ký số bởi CA – CA nói "đây là khóa công cộng của E"



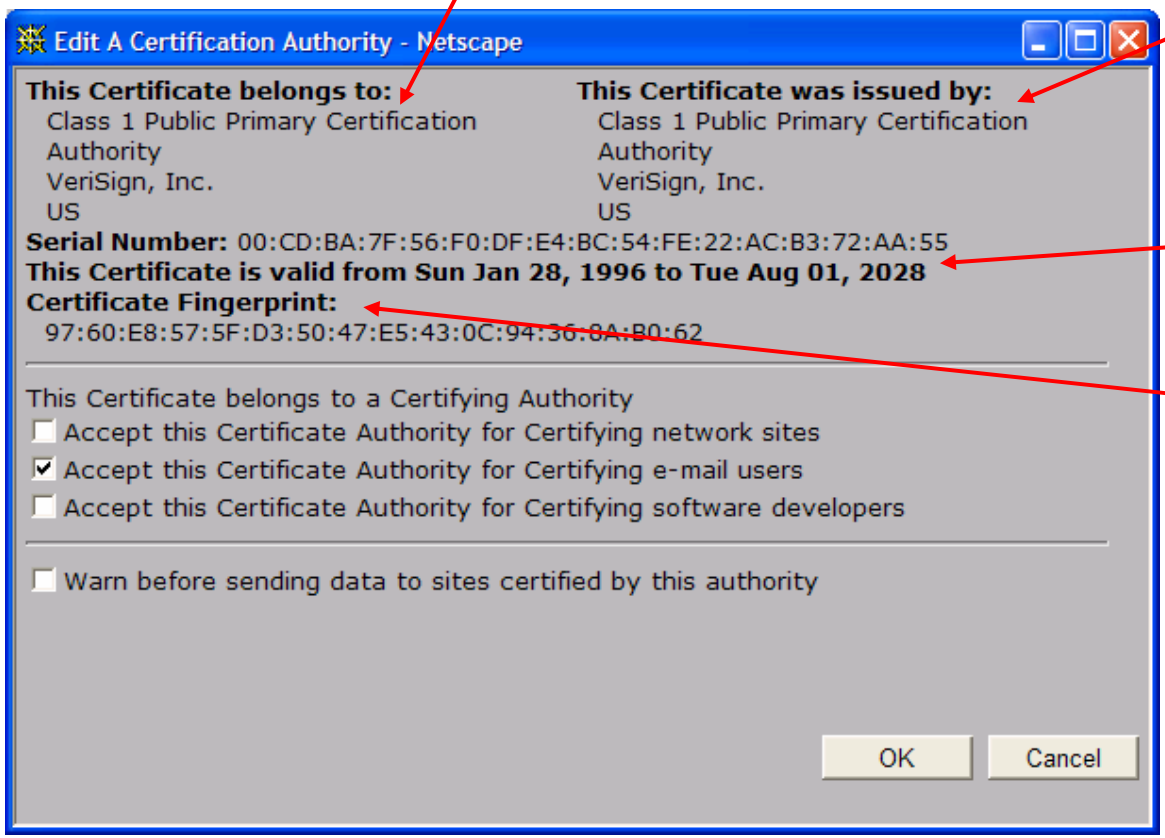
Cấp chứng chỉ

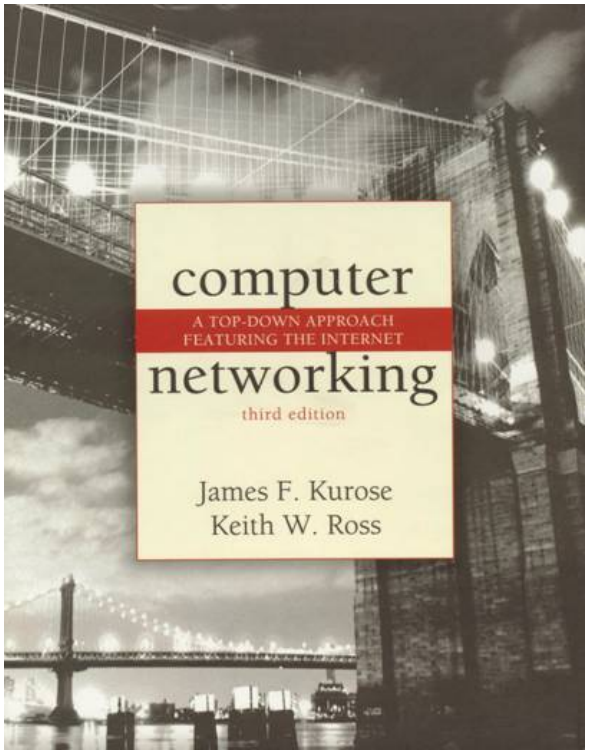
- Khi Alice muốn lấy được khóa công cộng của Bob:
  - lấy chứng chỉ của Bob (ở Bob hoặc nơi nào đó)
  - áp dụng khóa công cộng của CA cho chứng chỉ của Bob, lấy được khóa công cộng của Bob



Mỗi chứng chỉ chứa:

- Số thứ tự (duy nhất)
- thông tin về người sở hữu chứng chỉ, bao gồm giải thuật và chính giá trị khóa (không hiển thị ra)
- thông tin về người phát hành chứng chỉ
- ngày kiểm tra tính hợp lệ
- chữ ký số bởi người phát hành chứng chỉ



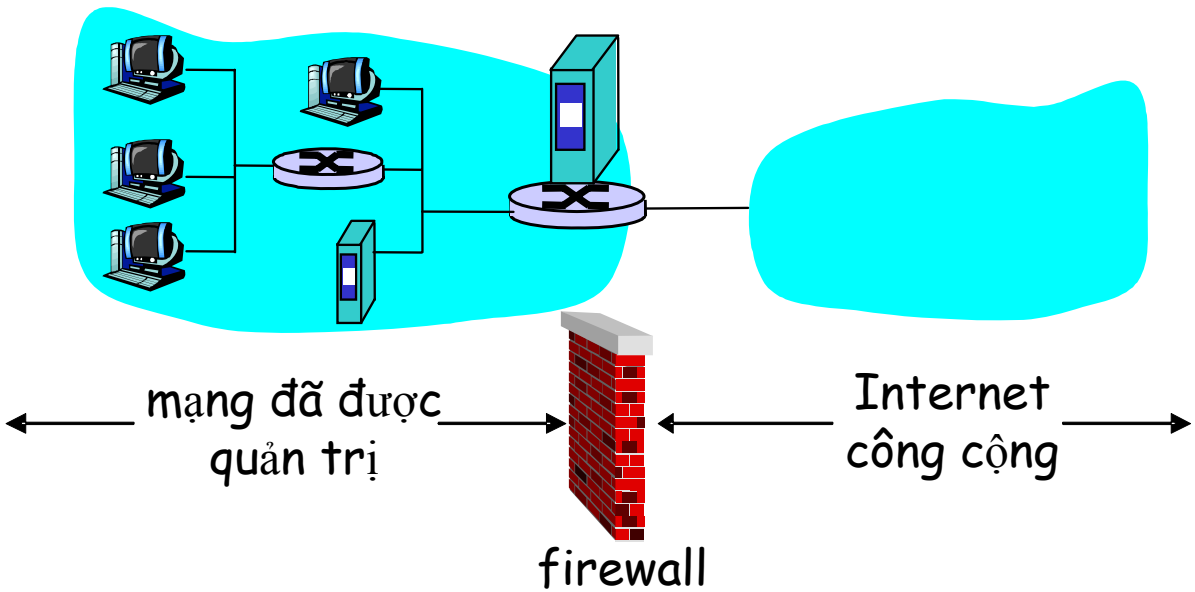


7.6 Điều khiển truy cập: các firewall

Các Firewall-Tường lửa

firewall

cô lập mạng nội bộ của tổ chức với Internet, cho phép một số gói được truyền qua, ngăn chặn các gói khác



Firewall: Tại sao phải dùng?

ngăn chặn các cuộc tấn công từ chối dịch vụ denial of service (DoS):

- SYN flooding: kẻ tấn công thiết lập nhiều kết nối TCP "ảo", không còn tài nguyên cho các kết nối "thật"

ngăn chặn việc sửa đổi/truy cập bất hợp pháp các dữ liệu nội bộ.

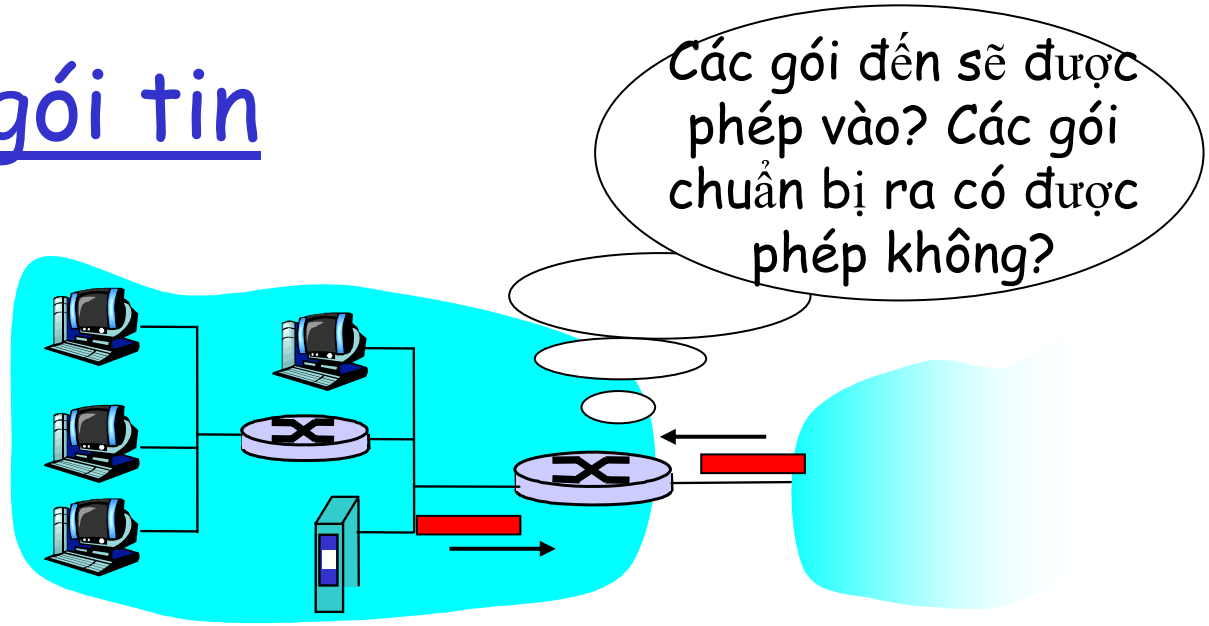
- vd: kẻ tấn công thay thế trang chủ của CIA bằng trang nào đó

chỉ cho phép các truy cập hợp pháp vào bên trong mạng (tập hợp các host/user được chứng thực)

2 kiểu firewall:

- mức ứng dụng
- lọc gói tin

Lọc gói tin



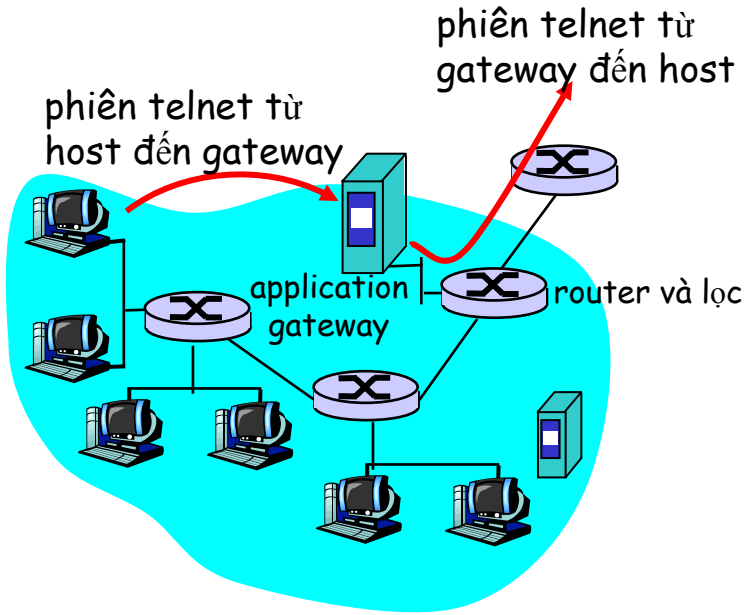
- mạng nội bộ kết nối với Internet thông qua router firewall
- router lọc từng gói một, xác định chuyển tiếp hoặc bỏ các gói dựa trên:
  - địa chỉ IP nguồn, địa chỉ IP đích
  - các số hiệu port TCP/UDP nguồn và đích
  - kiểu thông điệp ICMP
  - các bit TCP SYN và ACK

Lọc gói tin

- Ví dụ 1: chặn các datagram đến và đi với trường giao thức IP = 17 và port nguồn hoặc đích = 23.
  - Tất cả các dòng UDP đến/đi và các kết nối telnet đều bị chặn lại.
- Ví dụ 2: chặn các đoạn Block TCP với ACK=0.
  - Ngăn chặn các client bên ngoài tạo các kết nối TCP với các client bên trong, nhưng cho phép các client bên trong kết nối ra ngoài.

Các ứng dụng gateway

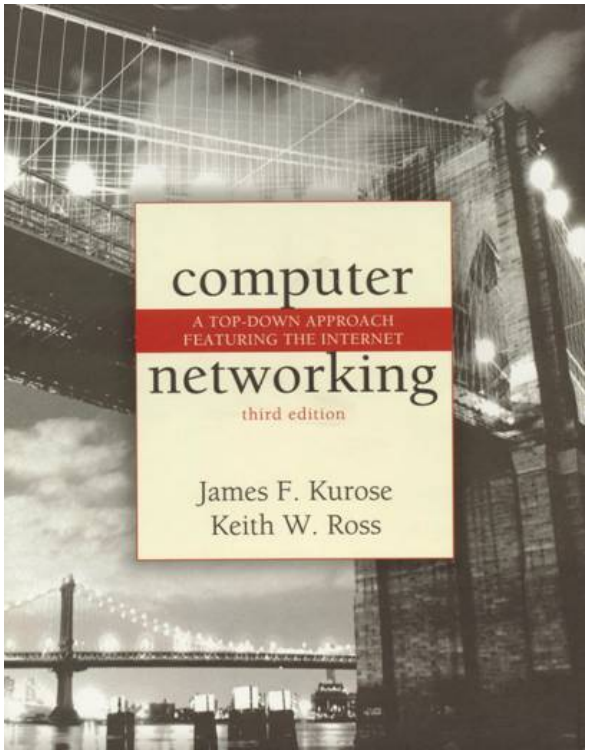
- Lọc các gói trên dữ liệu ứng dụng cũng như các trường IP/TCP/UDP.
- Ví dụ: cho phép chọn các user bên trong được telnet ra ngoài.



1. yêu cầu tất cả các user phải telnet thông qua gateway
2. với các user đã được cấp phép, gateway thiết lập kết nối với host đích. gateway tiếp vận dữ liệu giữa 2 kết nối.
3. Router lọc và chặn tất cả các kết nối telnet không xuất phát từ gateway.

Các hạn chế của các firewall và gateway

- giả mạo IP: router không thể biết dữ liệu có thực sự đến từ nguồn tin cậy hay không
- nếu nhiều ứng dụng cần đối xử đặc biệt, mỗi cái sở hữu gateway riêng…
- phần mềm client phải biết cách tiếp xúc với gateway.
  - ví dụ: phải thiết lập địa chỉ IP của proxy trong trình duyệt Web
- các lọc thường dùng tất cả hoặc không có chính sách nào dành cho UDP
- sự cân bằng: mức độ truyền thông với bên ngoài và sự an toàn
- nhiều site bảo vệ mức cao vẫn phải chịu đựng sự tấn công



7.7 Các loại tấn công và cách phòng chống

Các mối đe dọa bảo mật Internet

Phương thức:

- trước khi tấn công: tìm hiểu các dịch vụ đã hiện thực/hoạt động trên mạng
- Dùng `ping` để xác định các host nào có địa chỉ trên mạng
- Quét port: liên tục thử thiết lập các kết nối TCP với mỗi port (xem thử chuyện gì xảy ra)

Biện pháp đối phó?

Các mối đe dọa bảo mật Internet

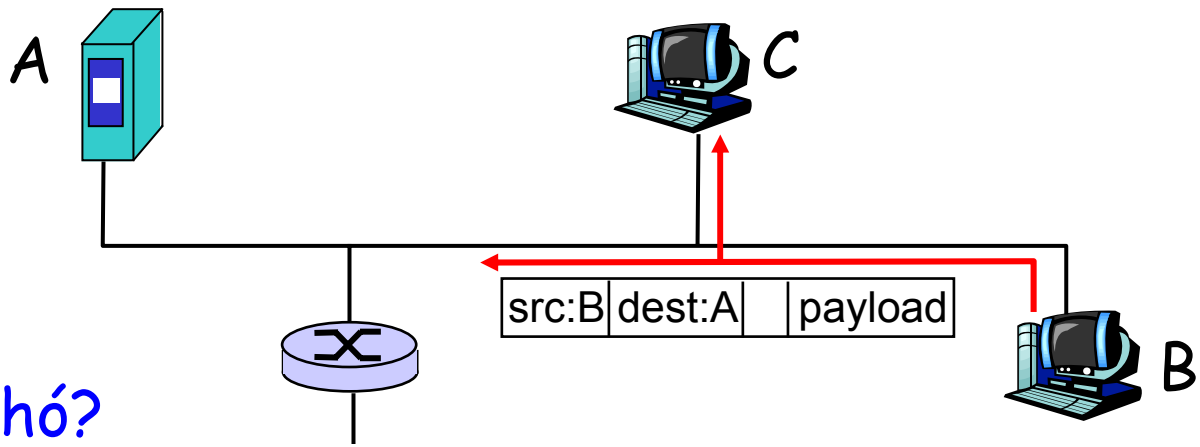
Biện pháp đối phó

- ghi nhận lưu thông vào mạng
- quan tâm hành vi nghi ngờ (các địa chỉ IP, port bị quét liên tục)

Các mối đe dọa bảo mật Internet

nghe ngóng gói:

- phương tiện truyền thông phổ biến
- NIC promiscuous (hỗn tạp) đọc tất cả các gói chuyển qua nó
- có thể đọc tất cả các dữ liệu được mã hóa (như mật khẩu)
- ví dụ: C nghe ngóng các gói của B

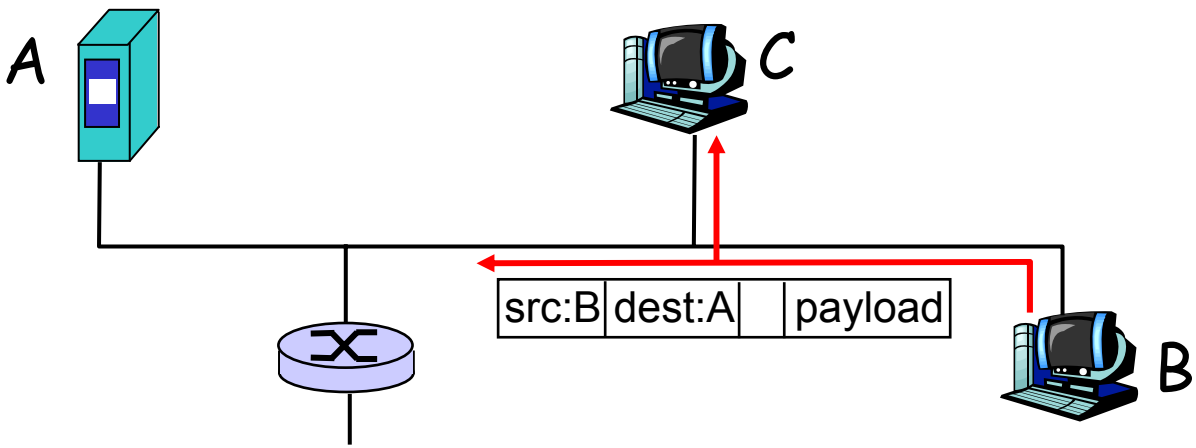


Biện pháp đối phó?

Các mối đe dọa bảo mật Internet

Packet sniffing: Biện pháp đối phó

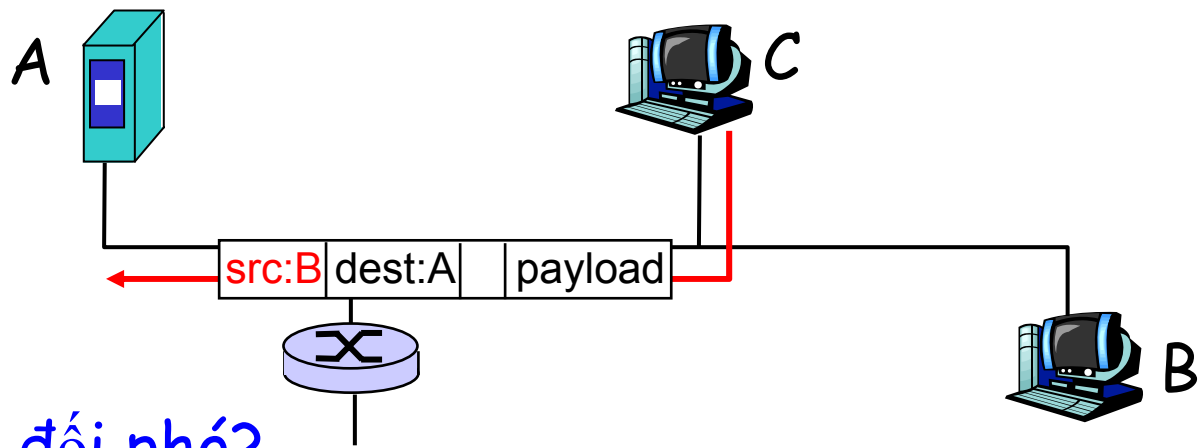
- tất cả các host trong tổ chức chạy phần mềm kiểm tra định kỳ thử host có ở chế độ promiscuous
- 1 host mỗi đoạn của phương tiện truyền thông



Các mối đe dọa bảo mật Internet

IP Spoofing (giả mạo IP):

- có thể sinh ra các gói IP "thô" trực tiếp từ ứng dụng, gán giá trị bất kỳ vào trường địa chỉ IP nguồn
- bên nhận không thể xác định nguồn bị giả mạo
- ví dụ: C giả mạo là B

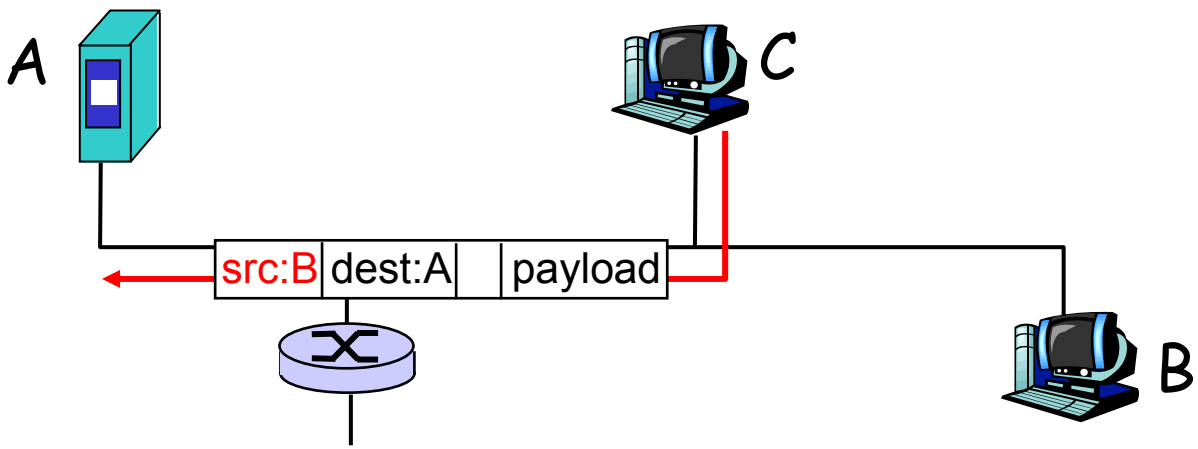


Biện pháp đối phó?

Các mối đe dọa bảo mật Internet

IP Spoofing: lọc quyền vào

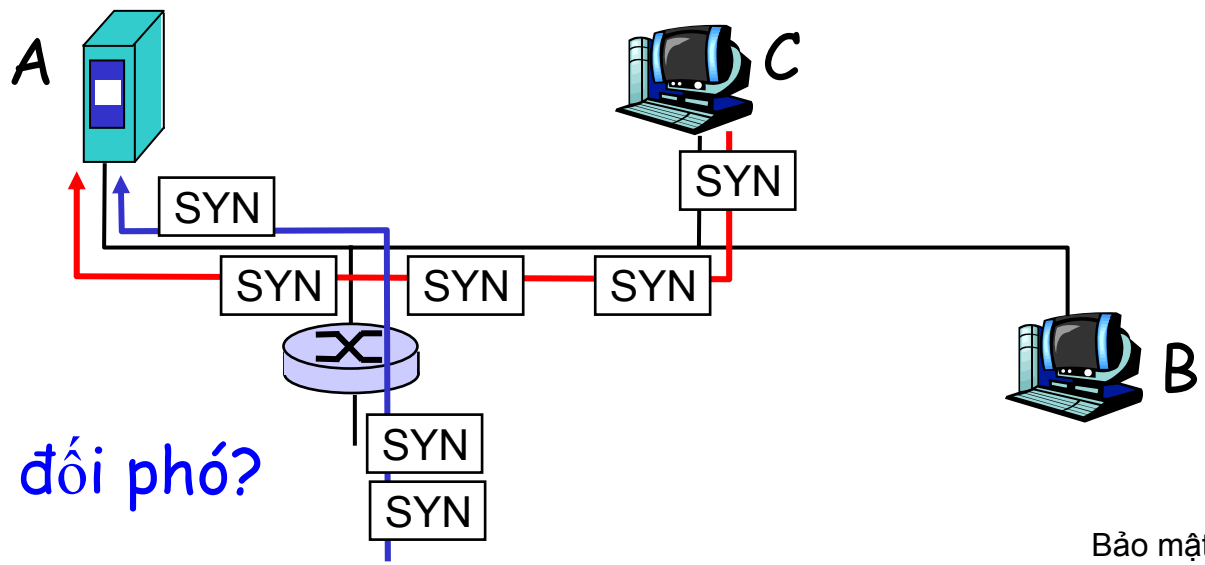
- router sẽ không chuyển tiếp các gói đi với trường hợp các địa chỉ nguồn không hợp lệ
- tuyệt vời, nhưng lọc như thế không thể áp dụng cho tất cả các mạng



Các mối đe dọa bảo mật Internet

Denial of service (DOS):

- gây ra "ngập lụt" bằng các gói sinh ra bởi ý đồ xấu cho bên nhận
- Distributed DOS (DDOS): nhiều nguồn phối hợp làm "ngập lụt" bên nhận
- ví dụ: C và các host ở xa tấn công SYN A

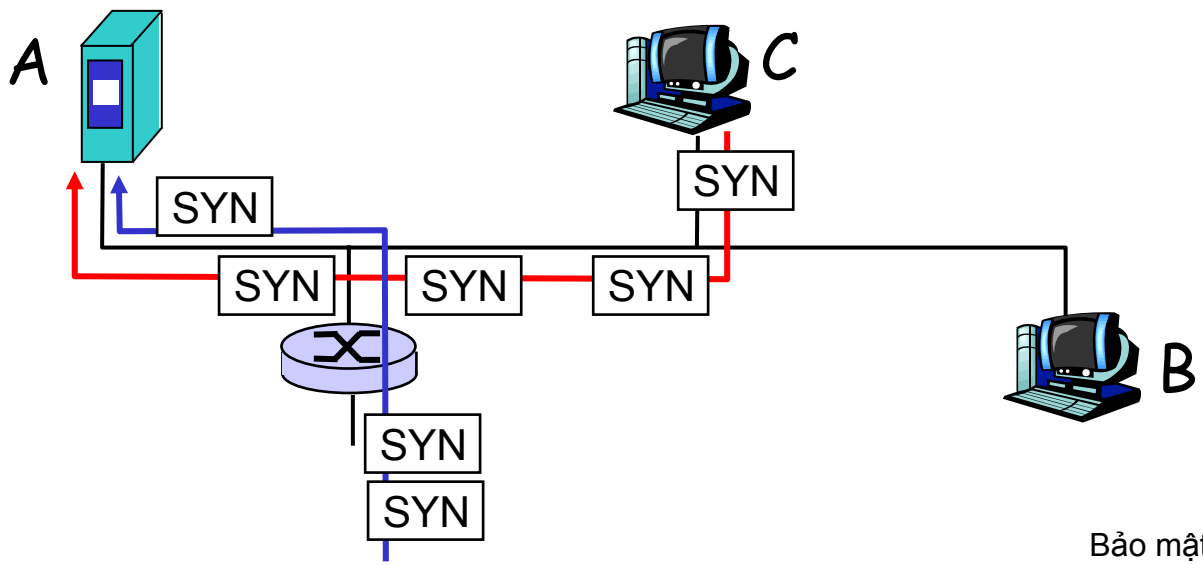


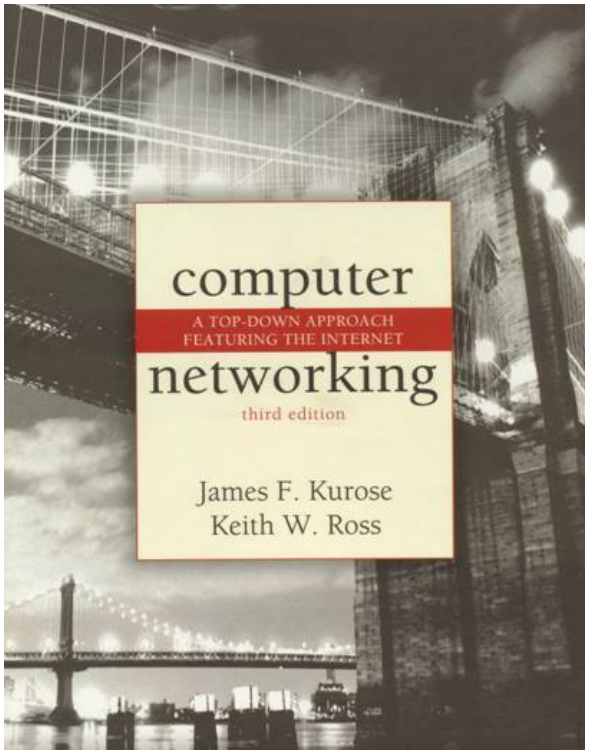
Biện pháp đối phó?

Các mối đe dọa bảo mật Internet

Denial of service (DOS): Biện pháp đối phó?

- lọc ra trước các gói dùng làm "ngập lụt" (ví dụ: SYN)
- theo dõi ngược lại nguồn gây ra "ngập lụt" (cơ chế giống máy phát hiện nói dối của Mỹ)



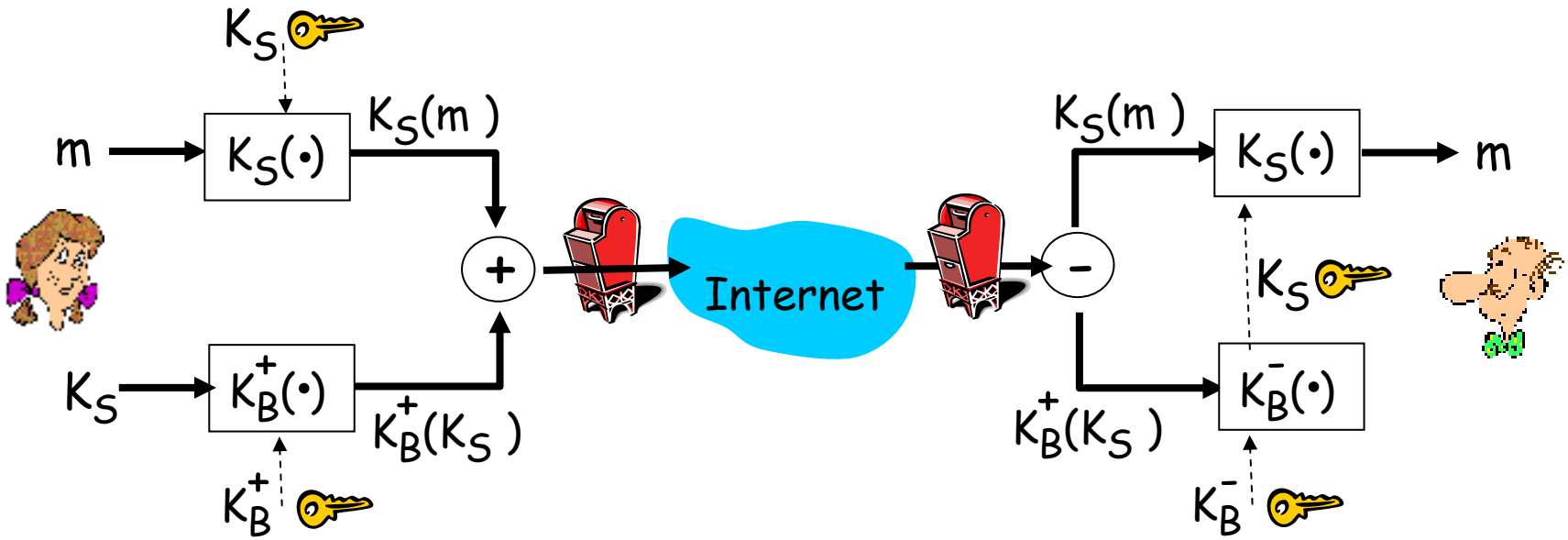


7.8 Bảo mật nhiều lớp

7.7.1. Bảo mật email
7.7.2. Bảo mật các socket
7.7.3. IPsec
7.7.4. Bảo mật trong 802.11

Bảo mật e-mail

- Alice muốn gửi 1 e-mail bí mật, m, đến Bob.

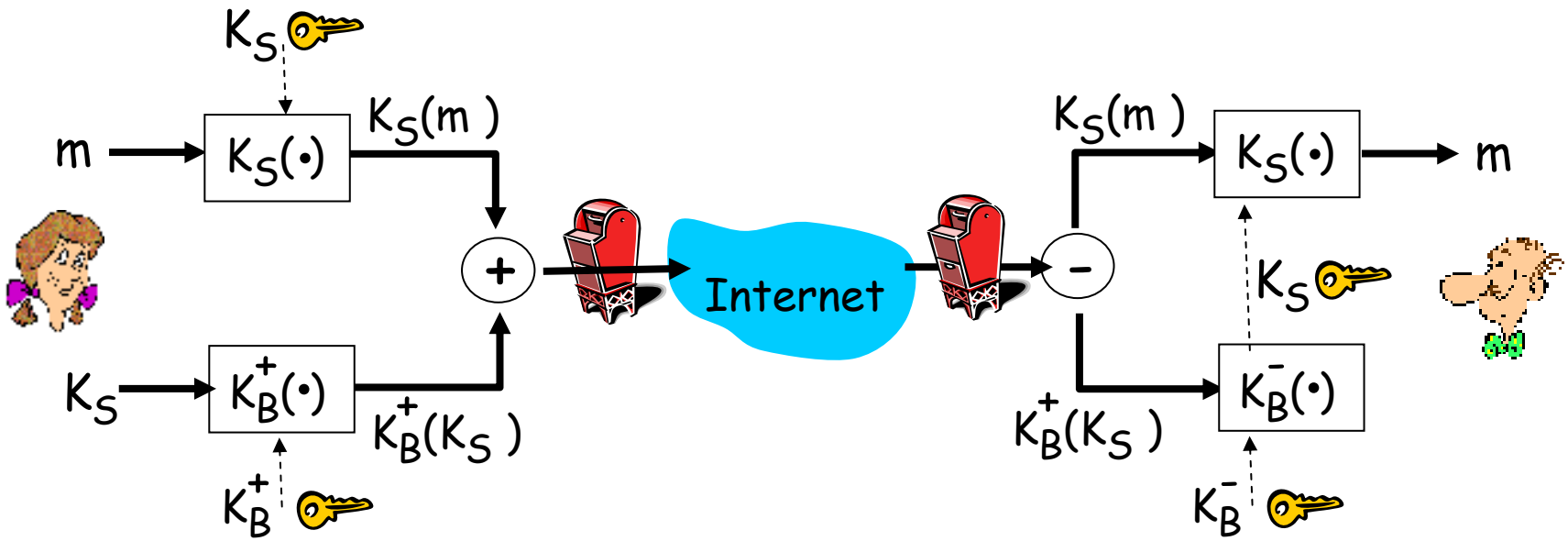


Alice:

- sinh ra khóa riêng đối xứng ngẫu nhiên, K_S.
- mã hóa thông điệp với K_S
- cũng mã hóa K_S với khóa công cộng của Bob.
- gửi cả $K_S(m)$ và $K_B(K_S)$ cho Bob.

Bảo mật e-mail

- Alice muốn gửi 1 e-mail bí mật, m, đến Bob.

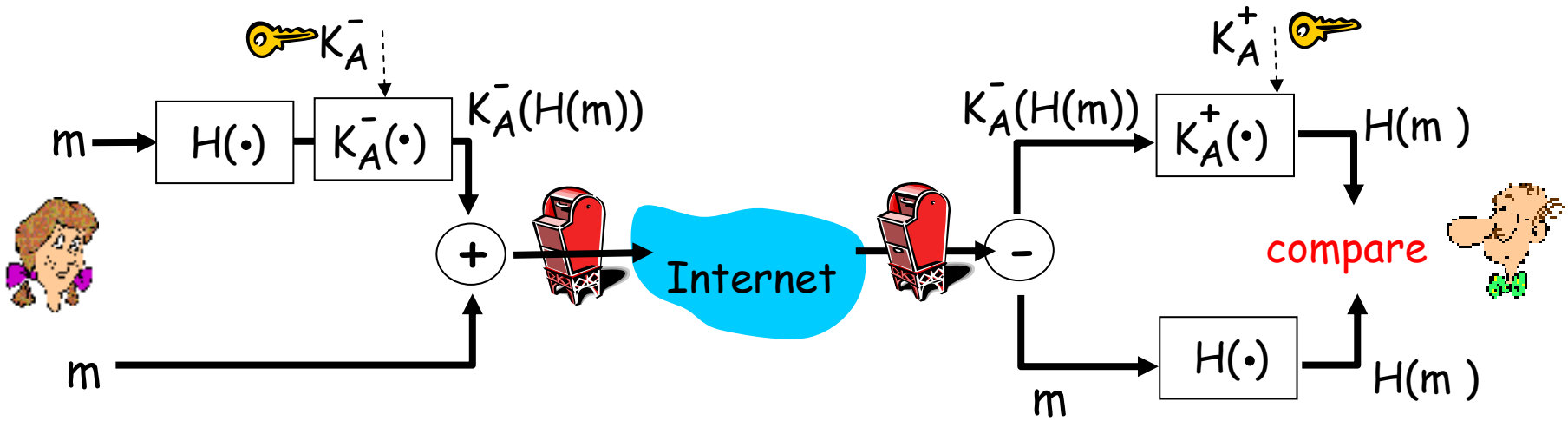


Bob:

- dùng khóa riêng của anh ấy để giải mã và phục hồi K_S
- dùng K_S để giải mã $K_S(m)$ và phục hồi m

Bảo mật e-mail

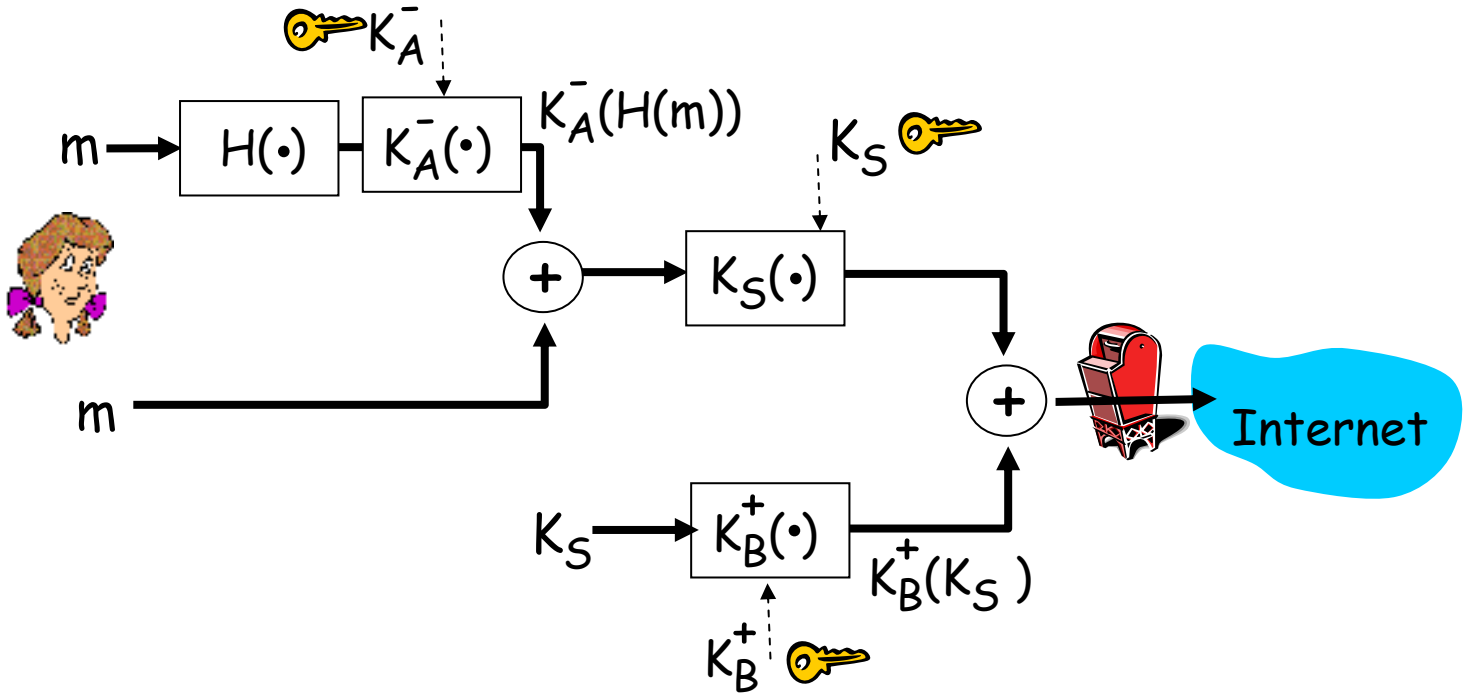
•Alice muốn cung cấp sự toàn vẹn thông điệp chứng thực người gửi.



- Alice ký số trên thông điệp.
- gửi cả thông điệp (dạng rõ ràng) và chữ ký số.

Bảo mật e-mail

• Alice muốn cung cấp sự toàn vẹn thông điệp chứng thực người gửi, sự bí mật



Alice dùng 3 khóa: khóa riêng của cô ấy, khóa công cộng của Bob, khóa đối xứng vừa mới tạo

Pretty good privacy (PGP)

- chuẩn trên thực tế, là lược đồ mã hóa email Internet.
- dùng mã hóa khóa đối xứng, khóa công cộng, hàm băm và chữ ký số như đã trình bày ở trước.
- hỗ trợ đồng nhất, chứng thực người gửi, sự bí mật
- người phát minh: Phil Zimmerman.

A PGP signed message:

```
---BEGIN PGP SIGNED MESSAGE---
Hash: SHA1

Bob:My husband is out of town
   tonight.Passionately yours,
   Alice

---BEGIN PGP SIGNATURE---
Version: PGP 5.0
Charset: noconv
yhHJRHhGJGhgg/12EpJ+lo8gE4vB3mqJ
   hFEvZP9t6n7G6m5Gw2
---END PGP SIGNATURE---
```

Secure sockets layer (SSL)

- bảo mật lớp transport với bất kỳ ứng dụng nào dựa trên TCP dùng các dịch vụ SSL
- dùng giữa trình duyệt Web, các server trong thương mại điện tử
- các dịch vụ bảo mật:
  - chứng thực server
  - mã hóa dữ liệu
  - chứng thực client (tùy chọn)
- chứng thực server:
  - trình duyệt cho phép SSL chứa các khóa công cộng cho các CA được tin cậy
  - trình duyệt yêu cầu chứng chỉ server, phát ra bởi CA được tin cậy
  - trình duyệt dùng khóa công cộng của CA để trích ra khóa công cộng của server từ chứng chỉ
- kiểm tra trong trình duyệt của bạn để thấy các CA được tin cậy

SSL (tt)

Mã hóa phiên làm việc SSL :

- trình duyệt sinh ra *khóa phiên đối xứng*, mã hóa nó với khóa công cộng của server, gửi khóa (đã mã hóa) cho server.
- dùng khóa riêng, server giải mã khóa phiên
- trình duyệt, server biết khóa phiên
  - tất cả dữ liệu gửi vào trong TCP socket (do client hoặc server) được mã hóa bởi khóa phiên.

- SSL: cơ sở của IETF Transport Layer Security (TLS).
- SSL có thể dùng cho các ứng dụng không Web, như IMAP.
- chứng thực client có thể hoàn thành với các chứng chỉ client

IPsec: bảo mật lớp Network

- bảo mật lớp Network:
  - host gửi mã hóa dữ liệu trong IP datagram
  - các đoạn TCP & UDP; các thông điệp ICMP & SNMP.
- chứng thực lớp Network:
  - host đích có thể chứng thực địa chỉ IP nguồn
- 2 giao thức cơ bản:
  - authentication header (AH)
  - encapsulation security payload (ESP)

- với cả AH và ESP, nguồn – đích bắt tay nhau:
  - tạo kênh logic lớp network gọi là một security association (SA)
- mỗi SA theo 1 chiều duy nhất
- duy nhất xác định bởi:
  - giao thức bảo mật (AH hoặc ESP)
  - địa chỉ IP nguồn
  - ID của kết nối 32-bit

Giao thức AH

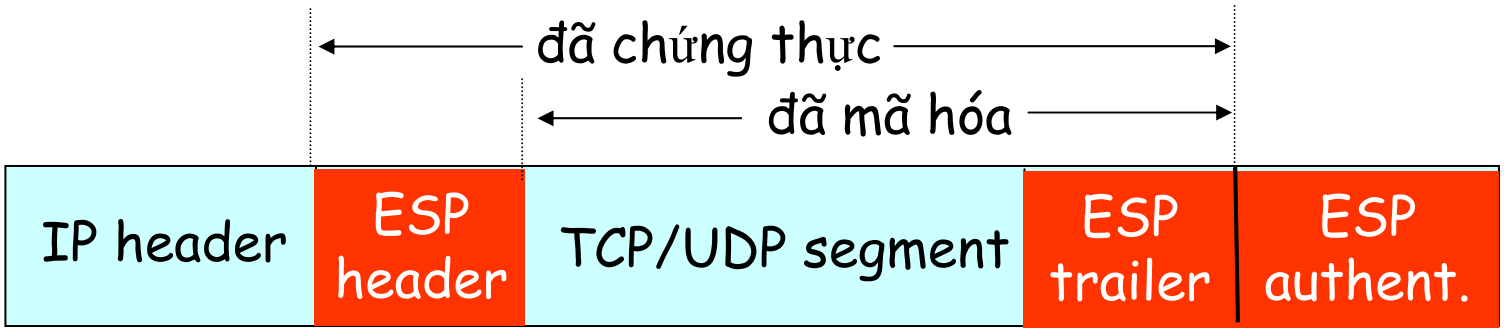
- hỗ trợ chứng thực nguồn, toàn vẹn dữ liệu, không tin cậy
- AH header được chèn vào giữa IP header, trường dữ liệu.
- trường giao thức: 51
- trung gian xử lý các datagram như bình thường

AH header chứa:

- nhân dạng kết nối
- dữ liệu chứng thực: thông điệp đã được ký từ nguồn được tính toán dựa trên IP datagram gốc
- trường header kế tiếp: xác định kiểu của dữ liệu (vd: TCP, UDP, ICMP)

Giao thức ESP

- hỗ trợ toàn vẹn dữ liệu, chứng thực host, tính bí mật
- mã hóa dữ liệu, ESP trailer
- trường header kế tiếp nằm trong ESP trailer.
- trường chứng thực ESP tương tự như của AH
- Protocol = 50.



Bảo mật IEEE 802.11

- *Khảo sát:*
  - 85% việc sử dụng mà không có mã hóa/chứng thực
  - dễ dàng bị phát hiện/nghe ngóng và nhiều loại tấn công khác!
- Bảo mật 802.11
  - mã hóa, chứng thực
  - thử nghiệm bảo mật 802.11 đầu tiên: Wired Equivalent Privacy (WEP): có thiếu sót
  - thử nghiệm hiện tại: 802.11i

Wired Equivalent Privacy (WEP):

- chứng thực như trong giao thức *ap4.0*
  - host yêu cầu chứng thực từ access point
  - access point gửi 128 bit
  - host mã hóa dùng khóa đối xứng chia sẻ
  - access point giải mã, chứng thực host
- không có cơ chế phân bố khóa
- chứng thực: chỉ cần biết khóa chia sẻ

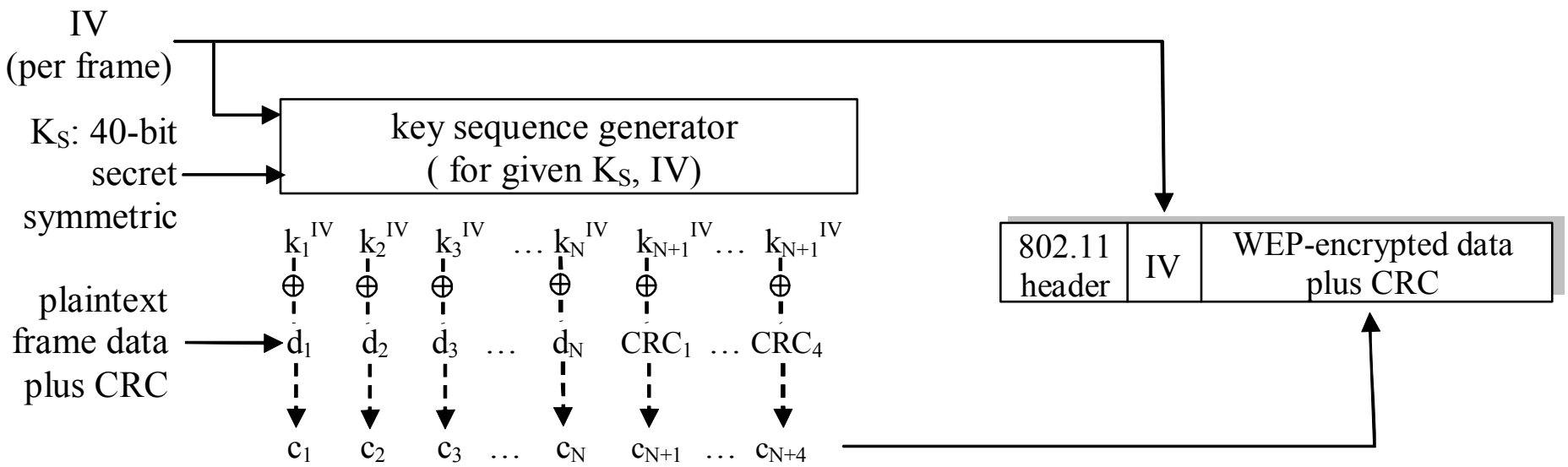
mã hóa dữ liệu WEP

- Host/AP chia sẻ khóa đối xứng 40 bit (bán cố định)
- Host gắn thêm vector 24 bit (initialization vector-IV) để hình thành khóa 64 bit
- khóa 64 bit dùng để sinh ra dòng các khóa, k_i^{IV}
- k_i^{IV} dùng để mã hóa byte thứ i, d_i, trong frame:

$$c_i = d_i \text{ XOR } k_i^{IV}$$

- IV và các byte đã được mã hóa, c_i gửi trong frame

mã hóa 802.11 WEP



mã hóa WEP phía gửi

Bẻ khóa 802.11 WEP

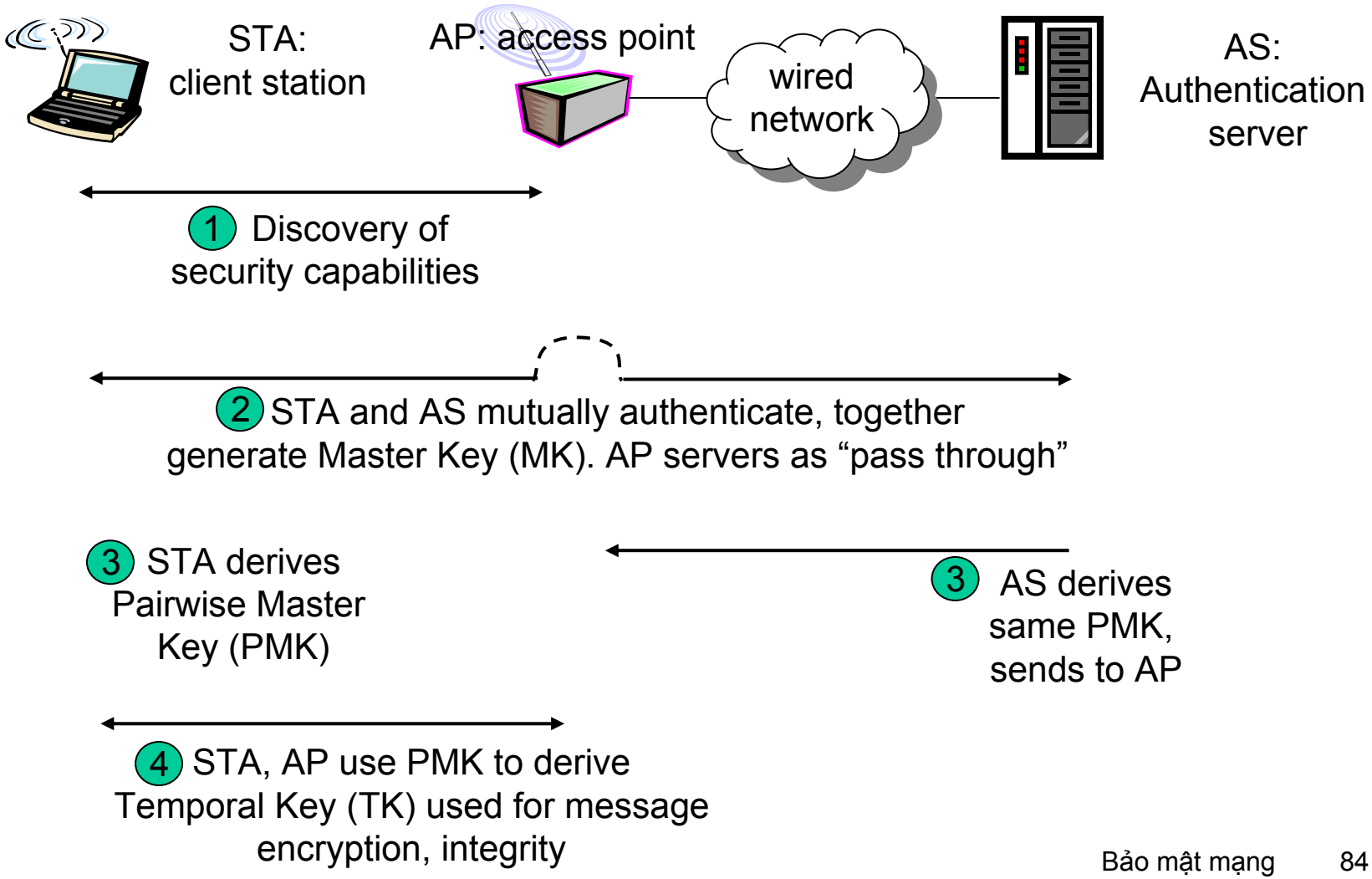
Lỗ hổng bảo mật:

- 24-bit IV, một IV mỗi frame -> phải dùng lại IV
- IV truyền đi với dạng văn bản thô -> phát hiện được việc dùng lại IV
- **Tấn công:**
  - Alice mã hóa văn bản thô cho trước d_1 d_2 d_3 d_4 …
  - Trudy nhìn thấy: $c_i = d_i$ XOR k_i^{IV}
  - Trudy biết c_i d_i, vì thế có thể tính được k_i^{IV}
  - Trudy biết được mã hóa chuỗi khóa k_1^{IV} k_2^{IV} k_3^{IV} …
  - lần dùng IV lại kế tiếp, Trudy có thể giải mã được!

802.11i: cải tiến sự bảo mật

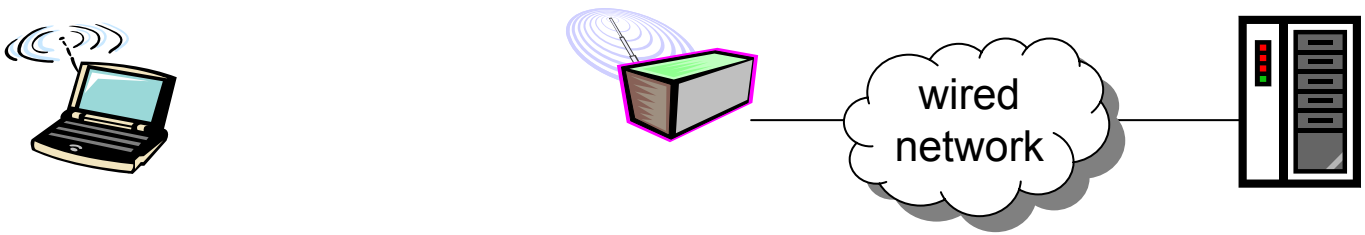
- rất nhiều (và chắc chắn hơn) dạng mã hóa có thể
- hỗ trợ phân bố khóa
- dùng chứng thực server tách riêng khỏi AP

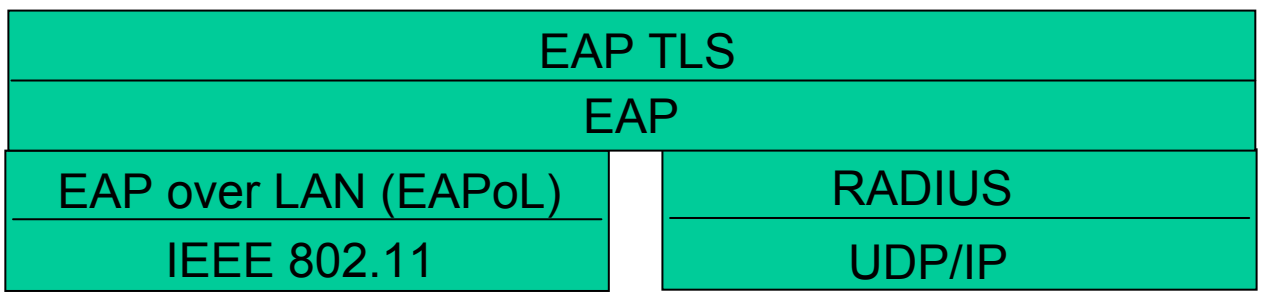
802.11i: 4 giai đoạn hoạt động



EAP: extensible authentication protocol

- EAP được gửi trên các "link" riêng biệt
  - mobile-đến-AP (EAP trên LAN)
  - AP đến server chứng thực (RADIUS trên UDP)





Tổng kết bảo mật mạng

Các kỹ thuật cơ bản.....

- mã hóa (đối xứng và công cộng)
- chứng thực
- toàn vẹn thông điệp
- phân bố khóa

.... sử dụng trong nhiều bối cảnh bảo mật khác nhau

- bảo mật email
- bảo mật vận chuyển (SSL)
- IP sec
- 802.11