

# Chapter 8

## an toàn bảo mật

A note on the use of these PowerPoint slides:

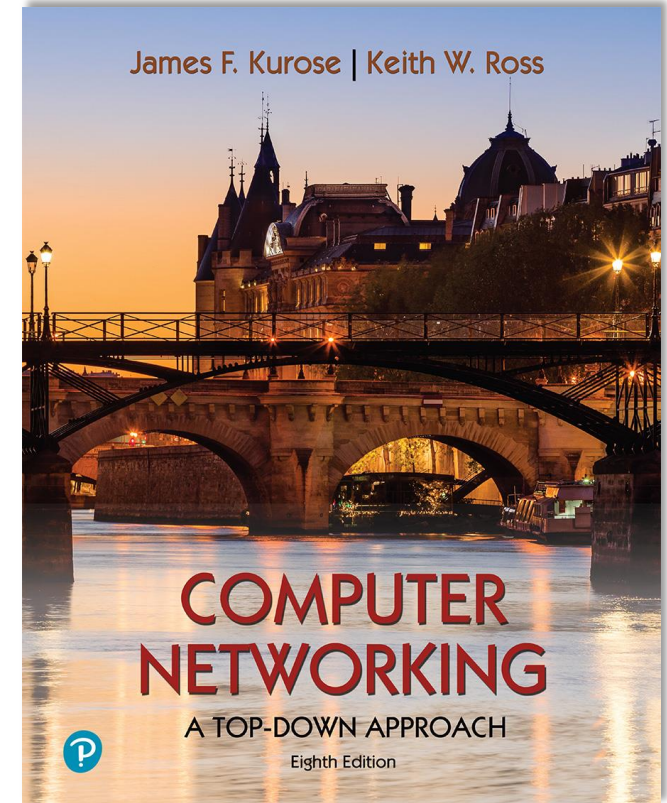
We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

For a revision history, see the slide note for this page.

Thanks and enjoy! JFK/KWR

All material copyright 1996-2020  
J.F Kurose and K.W. Ross, All Rights Reserved



*Computer Networking: A  
Top-Down Approach*

8<sup>th</sup> edition

Jim Kurose, Keith Ross  
Pearson, 2020

# Tổng quan

## Mục tiêu:

- Hiểu nguyên lý bảo mật của mạng:
  - Bảo mật (Mã hóa)
  - Xác thực
  - Toàn vẹn
- An toàn triển khai trong thực tế:
  - Tường lửa và hệ thống phát hiện xâm nhập
  - Bảo mật tầng ứng dụng, vận chuyển, mạng, liên kết

# Chapter 8 nội dung

- **Thế nào là bảo mật**
- Nguyên lý mã hóa
- Toàn vẹn thông điệp và xác thực
- Bảo mật email
- Bảo mật kết nối TCP: TLS
- Bảo mật tầng mạng: IPsec
- Bảo mật trong wireless và mobile networks
- Triển khai: firewalls và IDS



# Bảo mật mạng là gì?

**Confidentiality (bí mật):** chỉ người gửi và nhận biết nội dung

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

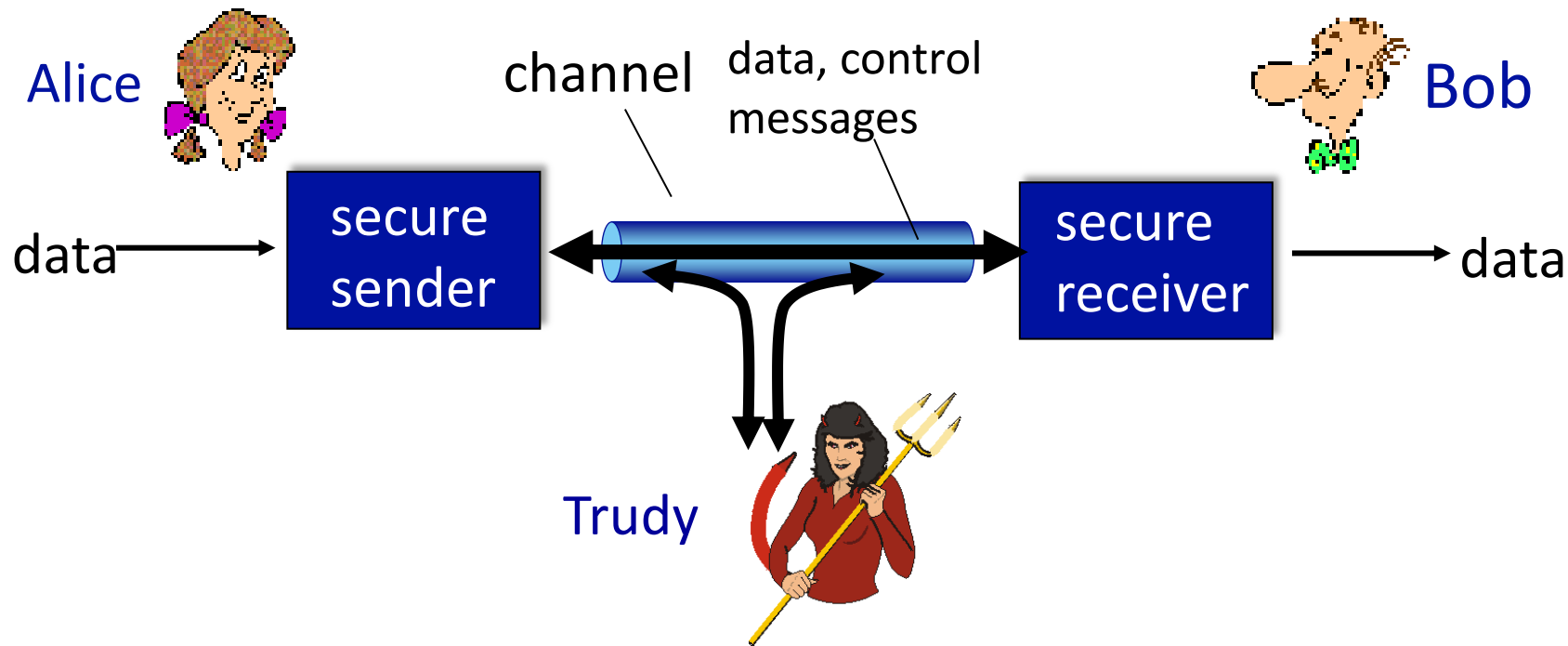
**Authentication (xác thực):** người gửi và người nhận xác thực định danh lẫn nhau

**message integrity (toàn vẹn):** người gửi, người nhận chắc chắn là thông điệp không bị thay đổi

**access and availability (sẵn có):** dịch vụ phải truy cập được và sẵn có

# Bạn và kẻ thù: Alice, Bob, Trudy

- Bob, Alice (lovers!) muốn kết nối bảo mật “securely”
- Trudy (intruder) có thể chặn bắt, xóa, hoặc thêm thông điệp



# Bạn và kẻ thù: Alice, Bob, Trudy

Ai là Bobs và Alices?

- ... Bobs và Alices là có thực!
- Web browser/server đối với giao dịch thương mại(mua bán trực tuyến)
- client/server: ngân hàng trực tuyến
- DNS servers
- Các router BGP trao đổi bảng định tuyến cập nhật

# Có những kẻ xấu!

Q: kẻ xấu sẽ làm cái gì?

A: rất nhiều

- **eavesdrop:** chặn bắt thông điệp
- Chèn thông điệp sai lệch trong kết nối
- **impersonation:** giả mạo địa chỉ nguồn trong gói (hoặc bất kì thông tin gì)
- **hijacking:** chiếm đoạt kết nối bằng cách loại bỏ người gửi hoặc người nhận
- **Tấn công từ chối dịch vụ:** ngăn chặn dịch vụ được thực hiện(e.g., by overloading resources)

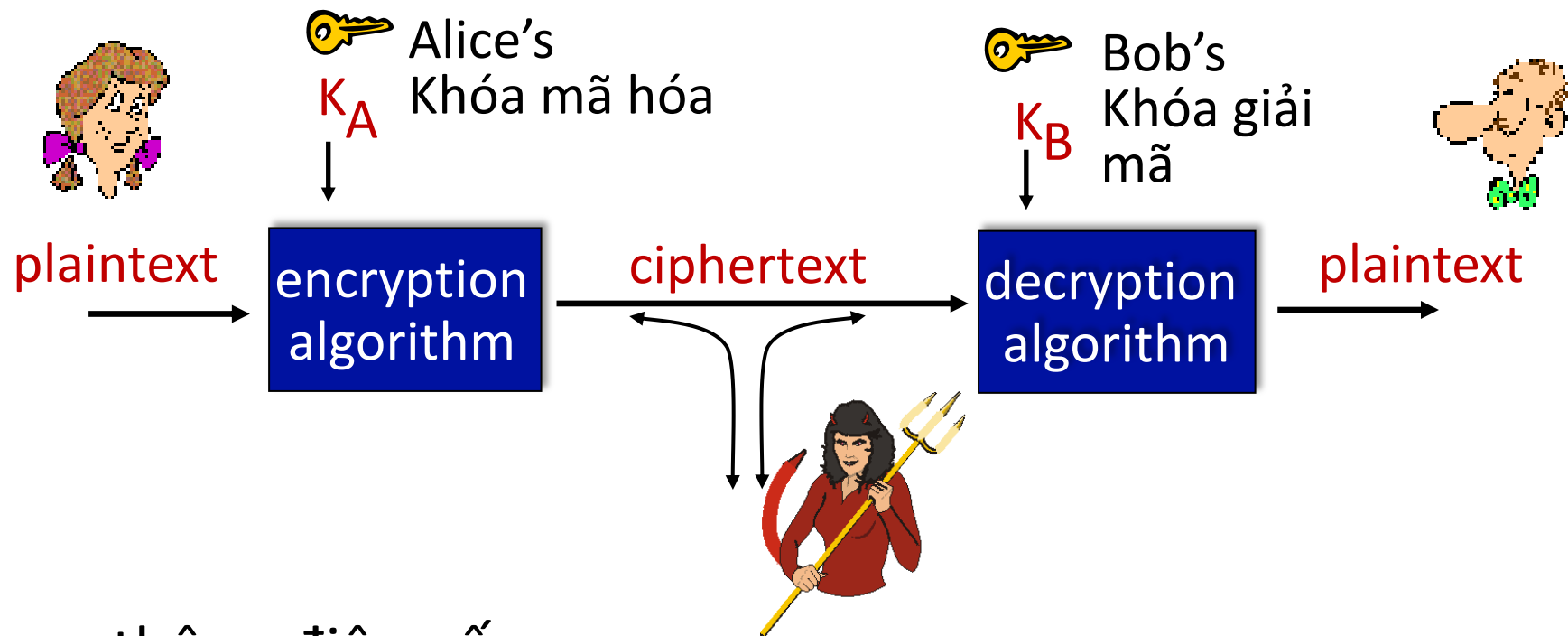
# Chapter 8 outline

- What is network security?
- **Principles of cryptography**
- Message integrity, authentication
- Securing e-mail
- Securing TCP connections: TLS
- Network layer security: IPsec
- Security in wireless and mobile networks
- Operational security: firewalls and IDS





# Ngôn ngữ của mã hóa



$m$ : thông điệp gốc

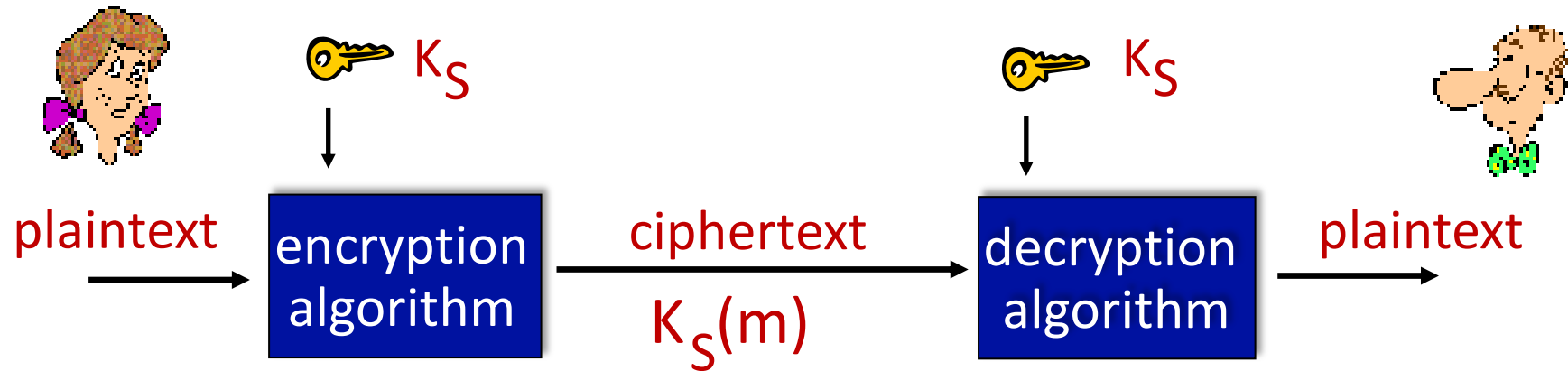
$K_A(m)$ : mã hóa với khóa  $K_A$

$m = K_B(K_A(m))$ : giải mã với khóa  $K_B$

# Phá mã

- **Tấn công bản mã:** kẻ xấu có bản mã, có thể phân tích tìm ra nguyên lý giải mã
- **Hai hướng:**
  - brute force (vét cạn): tìm kiếm thử tất cả các khóa có thể có
  - Phân tích thống kê
- **Tấn công dựa vào bản rõ đã biết:** kẻ xấu có bản rõ và bản mã
  - *e.g.*, tìm ra bảng chữ cái, nguyên lý mã hóa
- **chosen-plaintext attack:** kẻ xấu có thể lấy bản mã từ bản rõ được chúng chọn từ đó biết được lược đồ mã hóa. Ví dụ chúng có thể dụ nạn nhân gửi đoạn text có nội dung chúng yêu cầu, đoạn text này được mã hóa để gửi, chúng có thể so khớp đoạn mã với mã mà chúng đoán.

# Khóa đối xứng



**symmetric key crypto:** Bob and Alice chia sẻ cùng khóa(symmetric) key:  $K$

**Q:** Bob và Alice thống nhất khóa như thế nào?

# Lược đồ mã hóa đơn giản

*Mã hóa thay thế:* thay một thứ này bằng một thứ khác

- Mã hóa đơn giản: thay thế chữ cái này bằng chữ cái khác

plaintext:	abcdefghijklmnopqrstuvwxyz
	↓ ↓
ciphertext:	mnbvcxzasdfghjklpoiuytrewq

e.g.: Plaintext: bob. i love you. alice  
ciphertext: nkn. s gktc wky. mgsbc

🔑 *Khóa mã hóa:* so khớp tập 26 chữ cái với 26 kí tự mã hóa

# Mã hóa đối xứng: DES

## DES: Data Encryption Standard

- US encryption standard [NIST 1993]
- Khóa 56 bit, đầu vào 64 bit dữ liệu mã hóa
- Chia dữ liệu thành các khối 64 bit
- DES giữ bí mật như thế nào?
  - DES Challenge: đoán 56 bit khóa trong 1 ngày dùng vét cạn
  - Phân tích, vi sai...
- Làm DES bảo mật hơn:
  - 3DES: mã hóa 3 lần với 3 khóa khác nhau

# AES: Advanced Encryption Standard

- Thay thế cho DES (Nov 2001)
- Chia dữ liệu thành khối 128 bit blocks
- Khóa là 128, hoặc 192, hoặc 256
- Vết cạn có thể mất 149 ngàn tỉ năm đối với AES

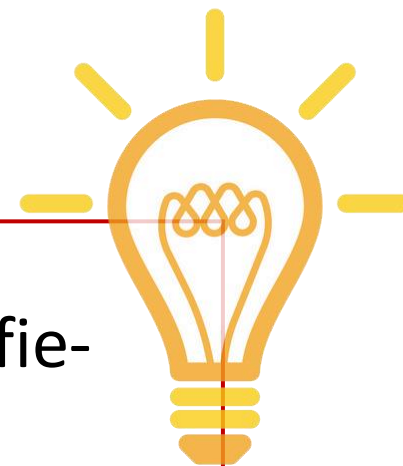
# Mã hóa công khai

## Mã hóa đối xứng:

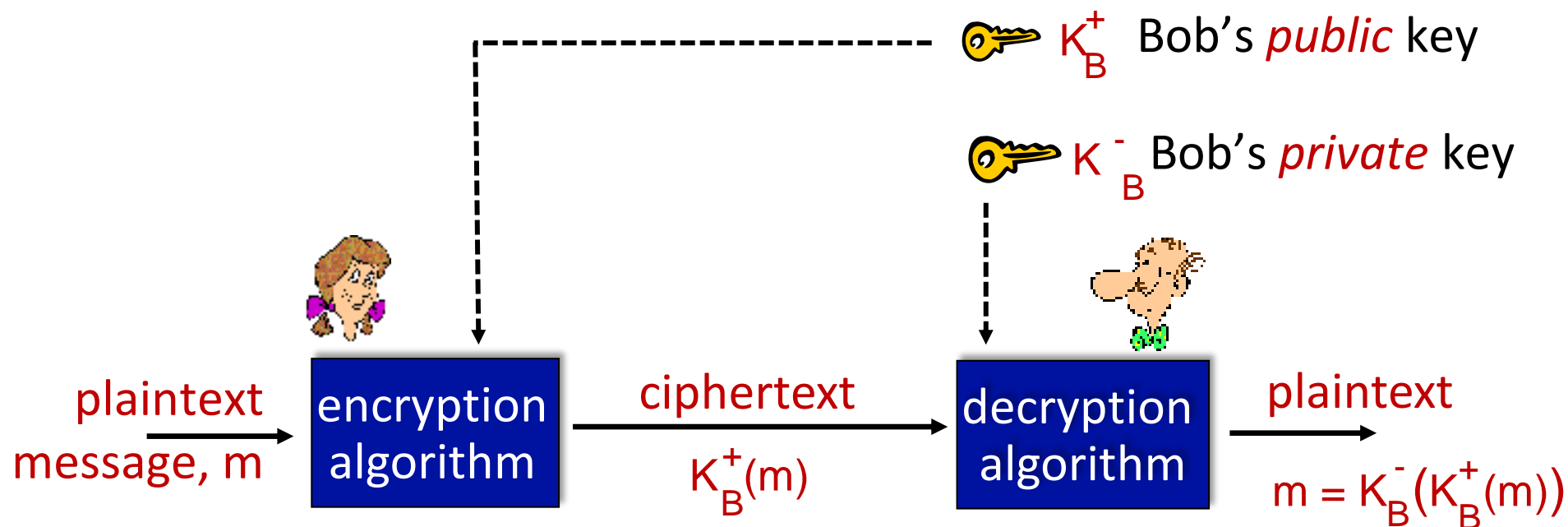
- Người gửi và người nhận cùng biết về khóa bí mật
- Q: làm cách nào thống nhất khóa này lần đầu tiên (chưa bao giờ người nhận và gửi gặp mặt)

## Mã hóa công khai

- *Cách tiếp cận khác*[Diffie-Hellman76, RSA78]
- Người gửi, nhận không cần phải chia sẻ khóa bí mật
- *Khóa công khai ai biết cũng được*
- *Khóa bí mật chỉ người nhận biết*



# Mã hóa công khai



**Wow** – khóa công khai đã cách mạng hóa mã hóa 2000 năm tuổi(chỉ có khóa đối xứng)



# Giải thuật mã hóa khóa công khai

Yêu cầu

① need  $K_B^+(\cdot)$  and  $K_B^-(\cdot)$  such that

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

② given public key  $K_B^+$ , it should be impossible to compute private key  $K_B^-$

**RSA:** Rivest, Shamir, Adelson algorithm

# Dựa vào toán học modulo

- $x \bmod n$  = phần còn lại của  $x$  khi chia bởi  $n$

- Biết:

$$[(a \bmod n) + (b \bmod n)] \bmod n = (a+b) \bmod n$$

$$[(a \bmod n) - (b \bmod n)] \bmod n = (a-b) \bmod n$$

$$[(a \bmod n) * (b \bmod n)] \bmod n = (a*b) \bmod n$$

- Vì vậy

$$(a \bmod n)^d \bmod n = a^d \bmod n$$

- Ví dụ:  $x=14$ ,  $n=10$ ,  $d=2$ :

$$(x \bmod n)^d \bmod n = 4^2 \bmod 10 = 6$$

$$x^d = 14^2 = 196 \quad x^d \bmod 10 = 6$$

# RSA:

- Thông điệp: thường là các bit
- Các bit có thể được biểu diễn bằng số nguyên
- Vì vậy mã hóa một thông điệp là tương đương với mã hóa một số
- Ví dụ:
  - $m = 10010001$ . thông điệp tương đương với 145.
  - Để mã hóa  $m$ , cần mã hóa số nguyên tương ứng

# Giải thuật RSA

- 1) Chọn hai số nguyên tố lớn  $p$  và  $q$  và tính  $N = pq$ . Cần chọn  $p$  và  $q$  sao cho:  
 $M < 2^{i-1} < N < 2^i$ . Với  $i = 1024$  thì  $N$  là một số nguyên dài khoảng 309 chữ số.
- 2) Tính  $n = (p - 1)(q - 1)$
- 3) Tìm một số  $e$  sao cho  $e$  nguyên tố cùng nhau với  $n$
- 4) Tìm một số  $d$  sao cho  $e \cdot d \equiv 1 \pmod{n}$  ( $d$  là nghịch đảo của  $e$  trong phép modulo  $n$ )
- 5) Hủy bỏ  $n, p$  và  $q$ . Chọn khóa công khai  $K_U$  là cặp  $(e, N)$ , khóa riêng  $K_R$  là cặp  $(d, N)$
- 6) Việc mã hóa thực hiện theo công thức:
  - Theo phương án 1, mã hóa bảo mật:  $C = E(M, K_U) = M^e \pmod{N}$
  - Theo phương án 2, mã hóa chứng thực:  $C = E(M, K_R) = M^d \pmod{N}$
- 7) Việc giải mã thực hiện theo công thức:
  - Theo phương án 1, mã hóa bảo mật:  $\bar{M} = D(C, K_R) = C^d \pmod{N}$
  - Theo phương án 2, mã hóa chứng thực:  $\bar{M} = D(C, K_U) = C^e \pmod{N}$

# Sinh khóa RSA

- Sinh khóa:
  - Chọn  $p, q$  là hai số nguyên tố
  - Tính  $n = p \times q$  ,  $\Phi(n) = (p-1) \times (q-1)$
  - Chọn  $e$  sao cho  $\text{UCLN}(\Phi(n), e) = 1$  ;  $1 < e < \Phi(n)$
  - Tính  $d$  sao cho  $(e \times d) \bmod \Phi(n) = 1$ .
  - Khóa công khai :  $K_U = (e, n)$
  - Khóa riêng :  $K_R = (d, n)$
- Mã hóa :  $C = M^e \bmod n$
- Giải mã:  $M = C^d \bmod n$

# Một cách khác RSA

```
1. Lựa chọn e từ 3,5,17,257,655373,5,17,257,65537
2.repeat
3.  p ← genprime(k/2)
4.until (p mod e)≠1
5.repeat
6.  q ← genprime(k - k/2)
7.until (q mod e)≠1
8.N ← pq ( $M < 2^{k-1} < N < 2^k$ )
9.n ← (p-1)(q-1)
10.d ← modinv(e, n)
11.return (N,e,d)
```

# RSA: encryption, decryption

0. cho  $(n, e)$  và  $(n, d)$  tính toán ở trên

1. để mã hóa  $m (< n)$ , tính

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

2 để giải mã mã  $c$  nhận được

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

magic happens!

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

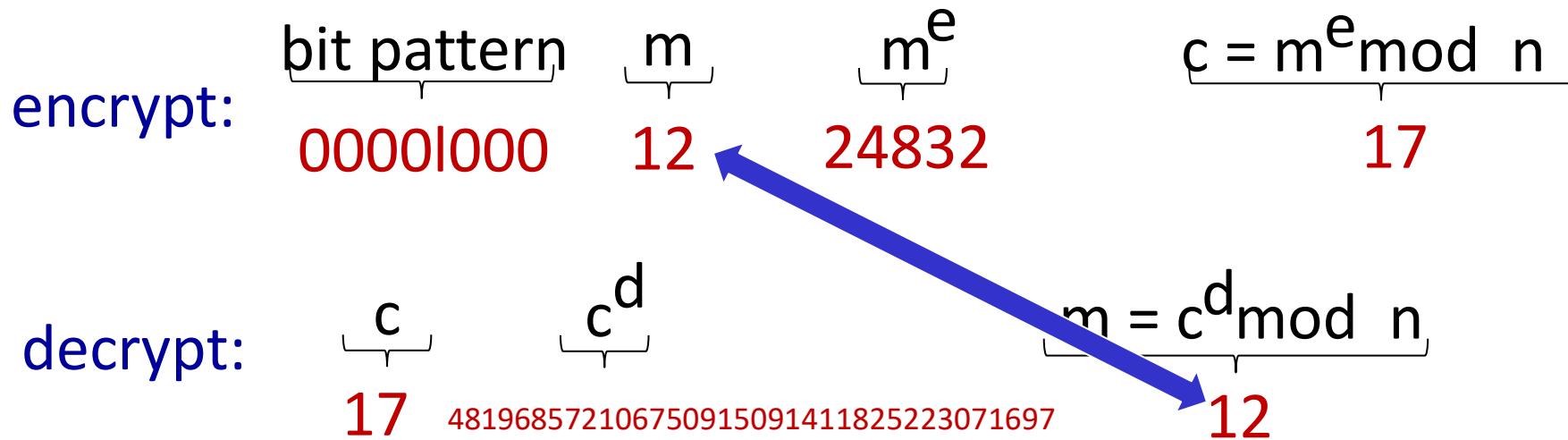
# RSA example:

Bob chooses  $p=5$ ,  $q=7$ . Then  $n=35$ ,  $z=24$ .

$e=5$  (so  $e$ ,  $z$  relatively prime).

$d=29$  (so  $ed-1$  exactly divisible by  $z$ ).

encrypting 8-bit messages.





# Why does RSA work?

- must show that  $c^d \bmod n = m$ , where  $c = m^e \bmod n$
- fact: for any  $x$  and  $y$ :  $x^y \bmod n = x^{(y \bmod z)} \bmod n$ 
  - where  $n = pq$  and  $z = (p-1)(q-1)$
- thus,
$$\begin{aligned}c^d \bmod n &= (m^e \bmod n)^d \bmod n \\&= m^{ed} \bmod n \\&= m^{(ed \bmod z)} \bmod n \\&= m^1 \bmod n \\&= m\end{aligned}$$

# RSA: thuộc tính quan trọng

Quan trọng:

$$\underbrace{K_B^-(K_B^+(m))}_{\text{Sử dụng khóa công khai trước}} = m = \underbrace{K_B^+(K_B^-(m))}_{\text{Sử dụng khóa bí mật trước, khóa công khai sau}}$$

Sử dụng khóa  
công khai trước  
khóa bí mật sau

Sử dụng khóa bí  
mật trước, khóa  
công khai sau

*Kết quả là như nhau*

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

Vì theo tính chất của modulo:

$$\begin{aligned}(m^e \bmod n)^d \bmod n &= m^{ed} \bmod n \\ &= m^{de} \bmod n \\ &= (m^d \bmod n)^e \bmod n\end{aligned}$$

# Tại sao RSA bảo mật?

- Nếu biết khóa công khai  $(n, e)$ . Khó để đoán ra  $d$ ?
- Cần phải phân tích  $n$  thành hai thừa số  $p$  và  $q$ 
  - Nếu  $n$  đủ lớn thì việc phân tích là bất khả thi

# RSA trong thực tế: khóa phiên

- Độ phức tạp của giải thuật RSA là một vấn đề lớn
- DES nhanh hơn RSA 100 lần
- Sử dụng khóa công khai để thiết lập phiên kết nối bảo mật sau đó thiết lập khóa thứ 2, cho việc mã hóa dữ liệu to  
establish secure connection, then establish second key – symmetric session key – for encrypting data

## Khóa phiên, $ks$

- Bob and Alice dùng RSA để trao đổi khóa đối xứng  $ks$
- Một khi cả hai có khóa  $ks$ , họ sử dụng khóa đối xứng để mã hóa

# Chapter 8 outline

- What is network security?
- Principles of cryptography
- **Authentication**, message integrity
- Securing e-mail
- Securing TCP connections: TLS
- Network layer security: IPsec
- Security in wireless and mobile networks
- Operational security: firewalls and IDS



# Authentication-xác thực

**Goal:** Bob muốn Alice chứng minh cô ấy là cô ấy

**Protocol ap1.0:** Alice nói “I am Alice”



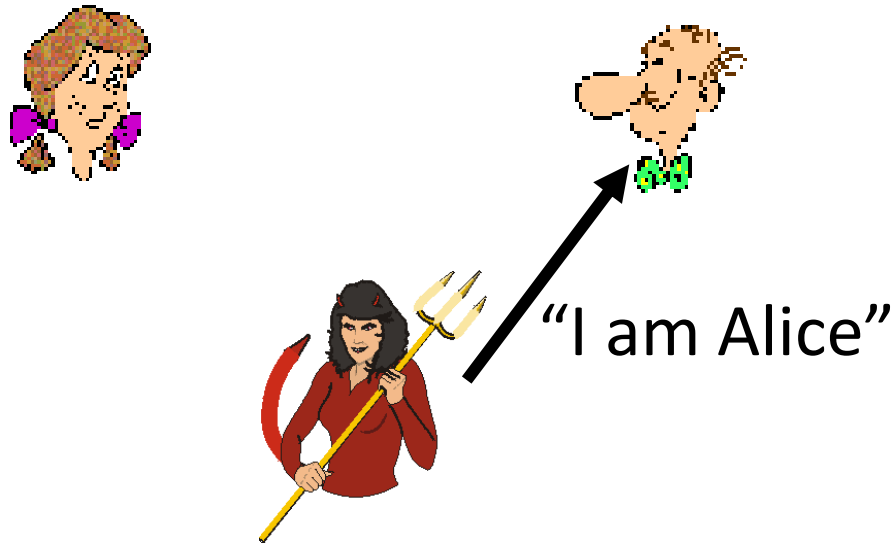
*failure scenario??*



# Authentication

**Goal:** Bob muốn Alice chứng minh cô ấy là cô ấy

**Protocol ap1.0:** Alice nói “I am Alice”



*Trong mạng Bob  
không thể nhìn  
thấy Alice nên  
Trudy dễ dàng giả  
mạo*

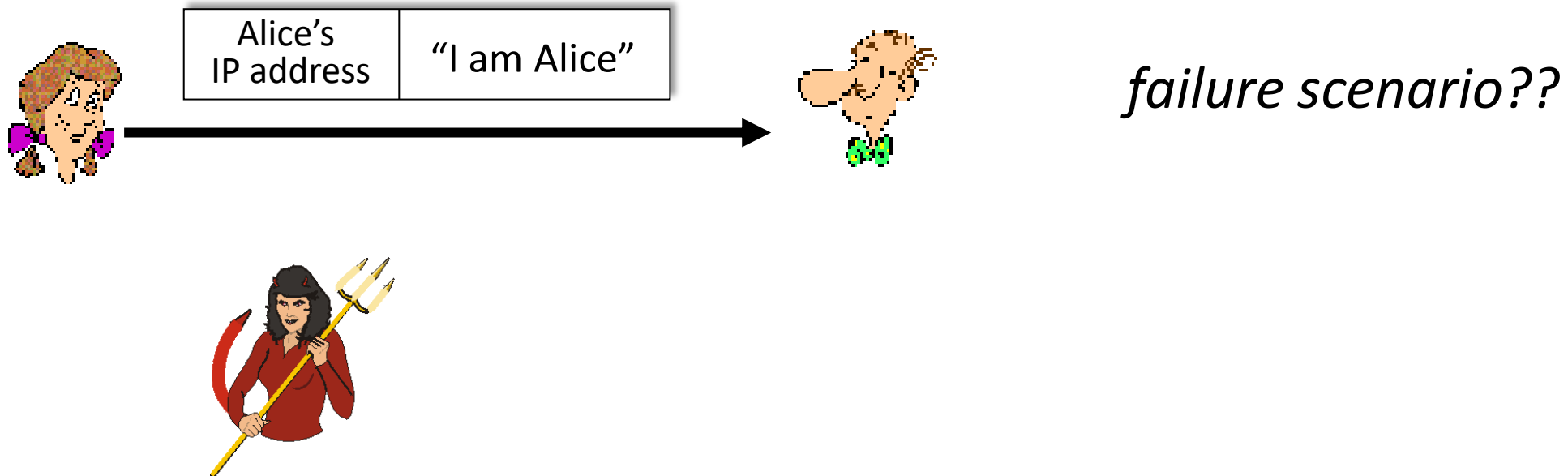




# Authentication: thử tiếp theo

**Goal:** Bob muốn Alice chứng minh cô ấy là cô ấy

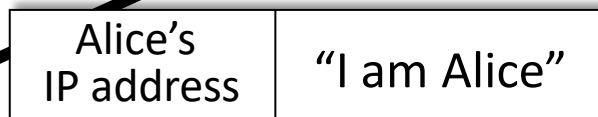
**Protocol ap2.0:** Alice nói “I am Alice” trong gói IP chứa địa chỉ của cô ấy



# Authentication: thử tiếp theo

**Goal:** Bob muốn Alice chứng minh cô ấy là cô ấy

**Protocol ap2.0:** Alice nói “I am Alice” trong gói IP chứa địa chỉ cô ấy

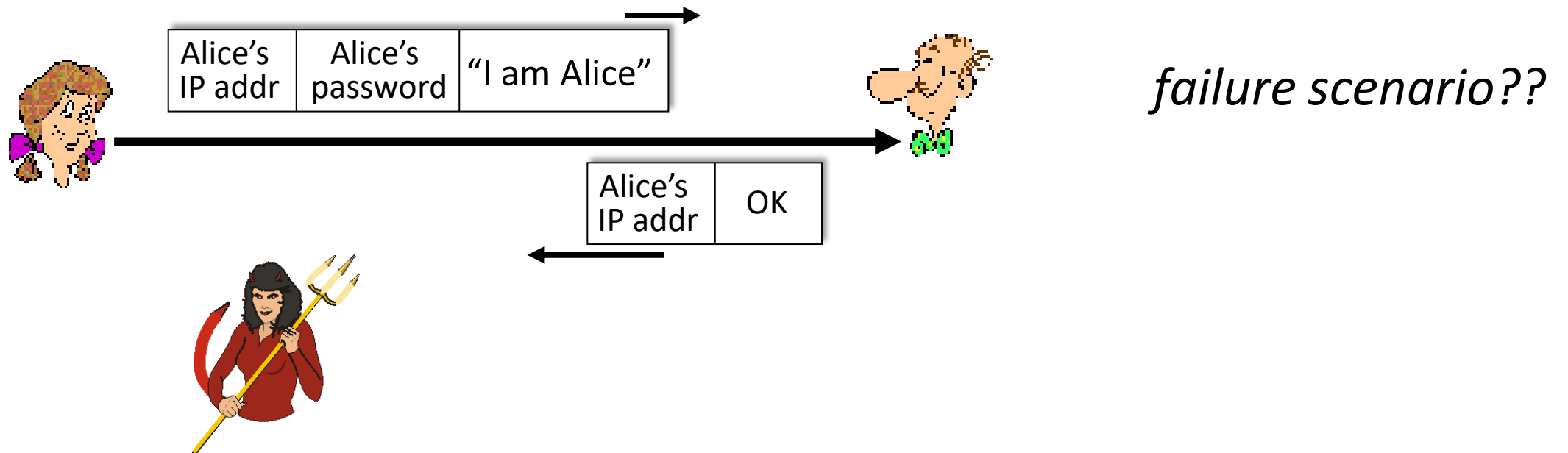


*Trudy có thể  
giả mạo gói tin chứa địa chỉ của Alice*

# Authentication: lần thử thứ 3

**Goal:** Bob muốn Alice chứng minh cô ấy là cô ấy

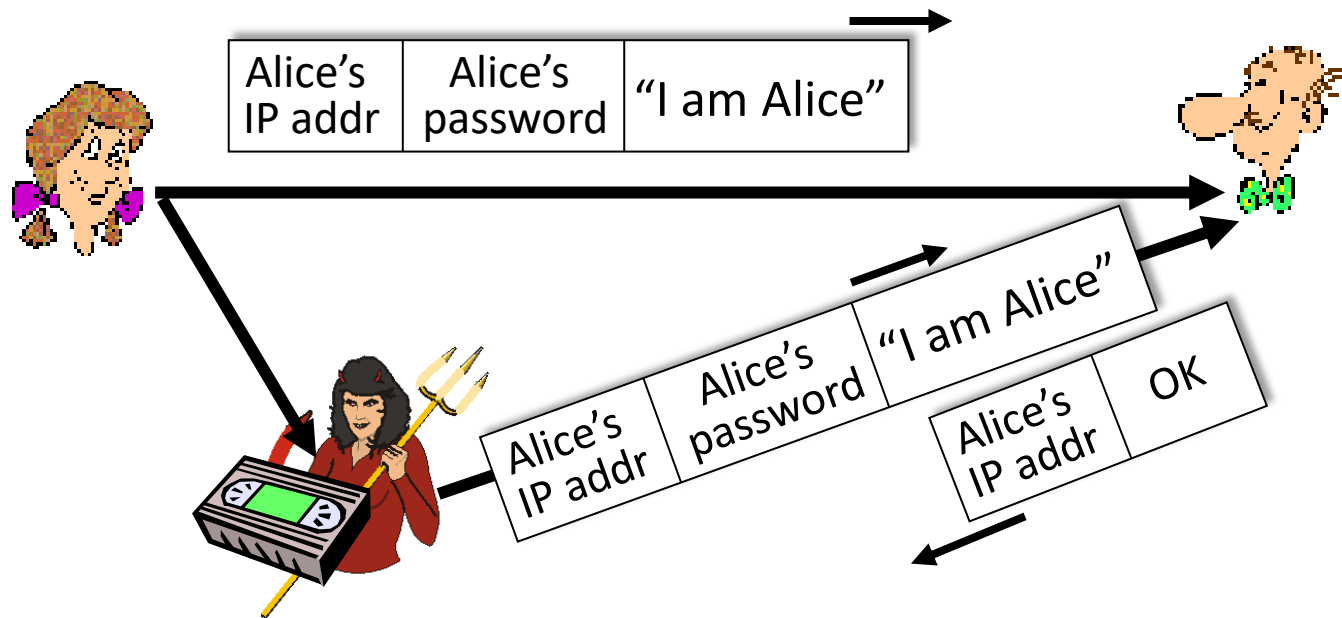
**Protocol ap3.0:** Alice nói “I am Alice” và gửi mật khẩu bí mật chứng minh cô ấy là cô ấy



# Authentication: lần thử thứ 3

**Goal:** Bob muốn Alice chứng minh cô ấy là cô ấy

**Protocol ap3.0:** Alice nói "I am Alice" và gửi mật khẩu bí mật chứng minh cô ấy là cô ấy

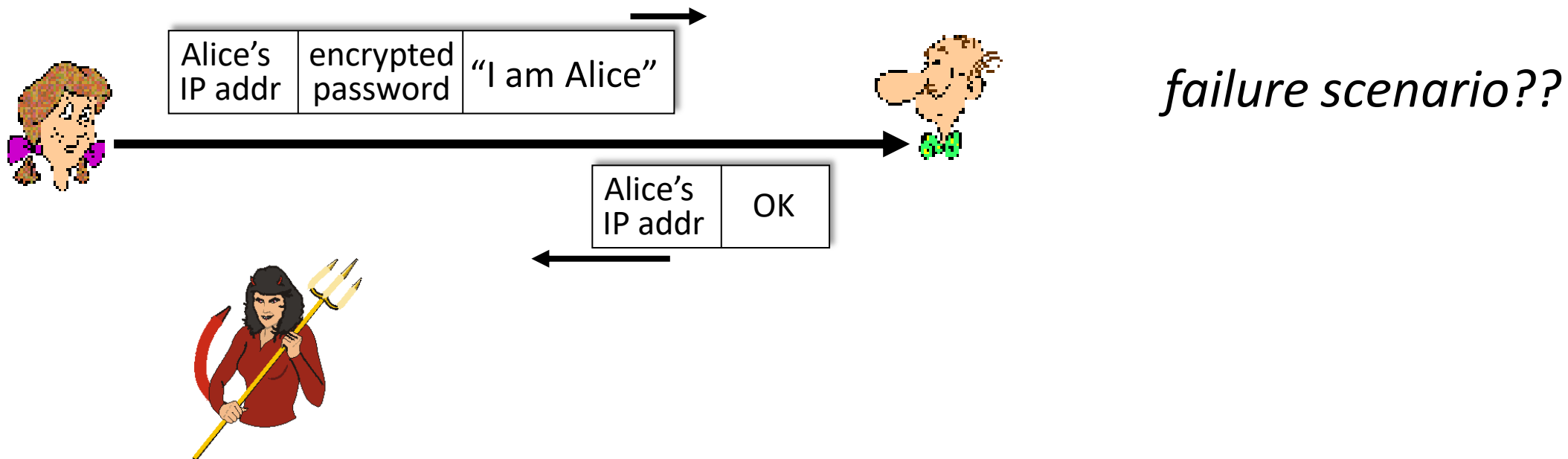


*playback attack:*  
Trudy ghi lại gói tin  
và sau đó gửi lại  
cho Bob

# Authentication: a modified third try

**Goal:** Bob muốn Alice chứng minh cô ấy là cô ấy

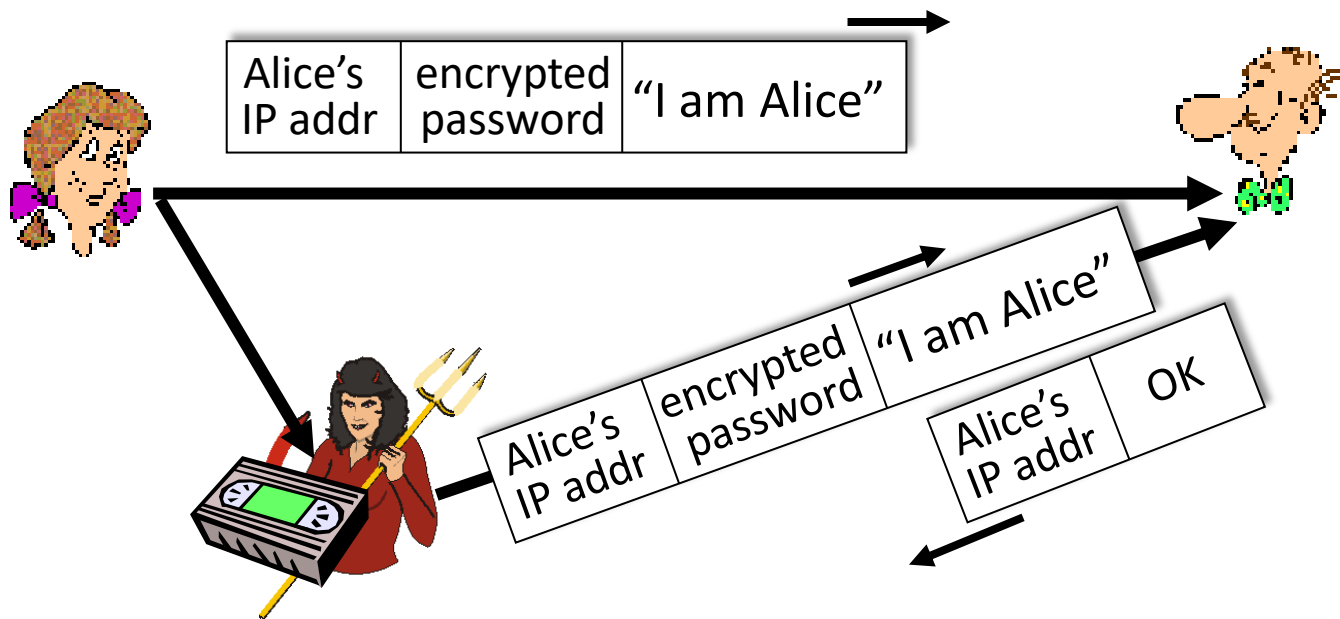
**Protocol ap3.0:** Alice nói "I am Alice" và gửi mật khẩu bí mật chứng minh cô ấy là cô ấy



# Authentication: a modified third try

**Goal:** Bob muốn Alice chứng minh cô ấy là cô ấy

**Protocol ap3.0:** Alice nói "I am Alice" và gửi mật khẩu bí mật chứng minh cô ấy là cô ấy



**playback attack:**  
*Trudy ghi lại gói tin và sau đó gửi lại cho Bob*

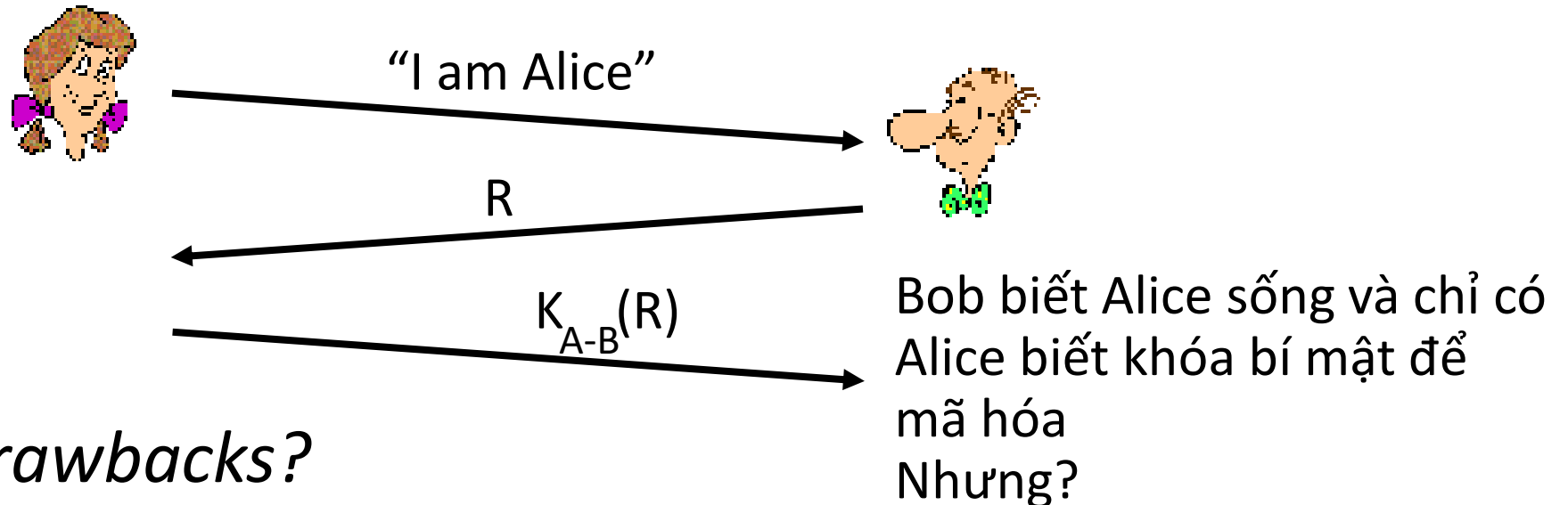
# Authentication: lần thử thứ 4

**Goal:** tránh tấn công playback attack

**nonce:** number (R) chỉ được dùng 1 lần duy nhất

**protocol ap4.0: chứng minh** Alice “sống”, Bob gửi Alice nonce, R

- Alice phải gửi lại R được mã hóa với khóa bí mật

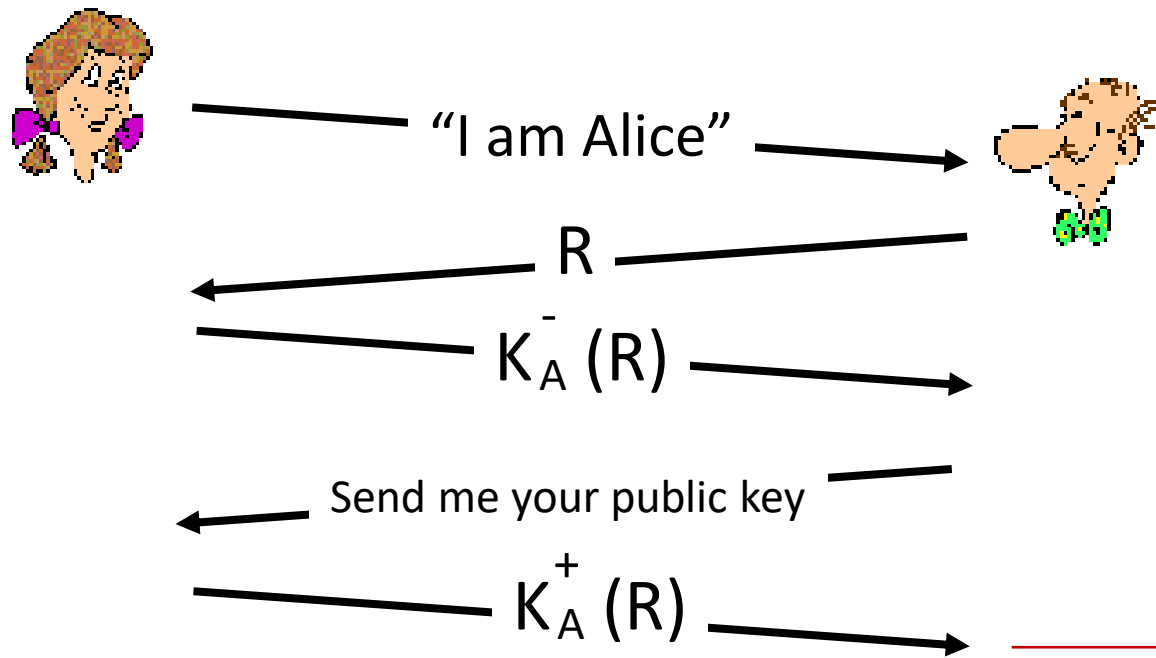


*Failures, drawbacks?*

# Authentication: ap5.0

ap4.0 sử dụng khóa đối xứng, có thể dung mã hóa khóa công khai?

**ap5.0:** sử dụng khóa công khai



Bob tính

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

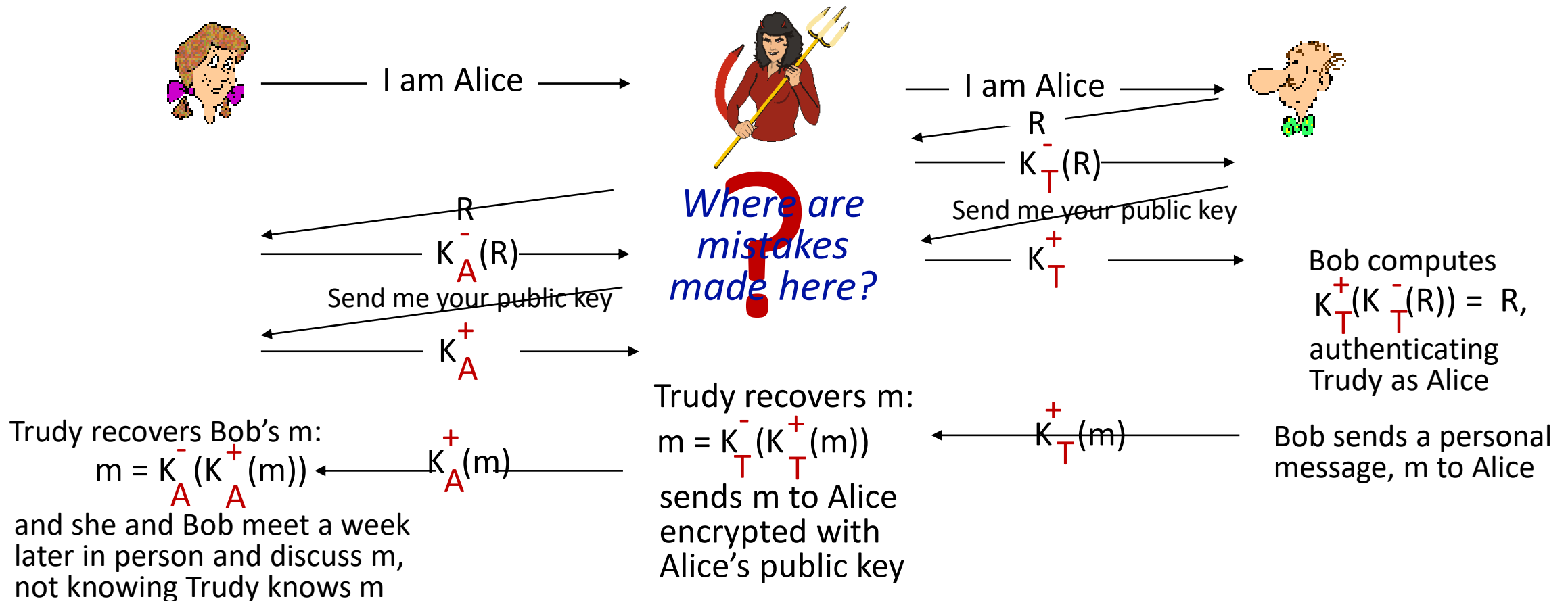
Và biết chỉ có Alice có khóa bí mật để mã hóa  $R$ :

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



# Authentication: ap5.0 – vẫn có lỗ hổng!

**man (or woman) in the middle attack:** Trudy đóng vai Alice với Bob, và Bob với Alice



# Chapter 8 outline

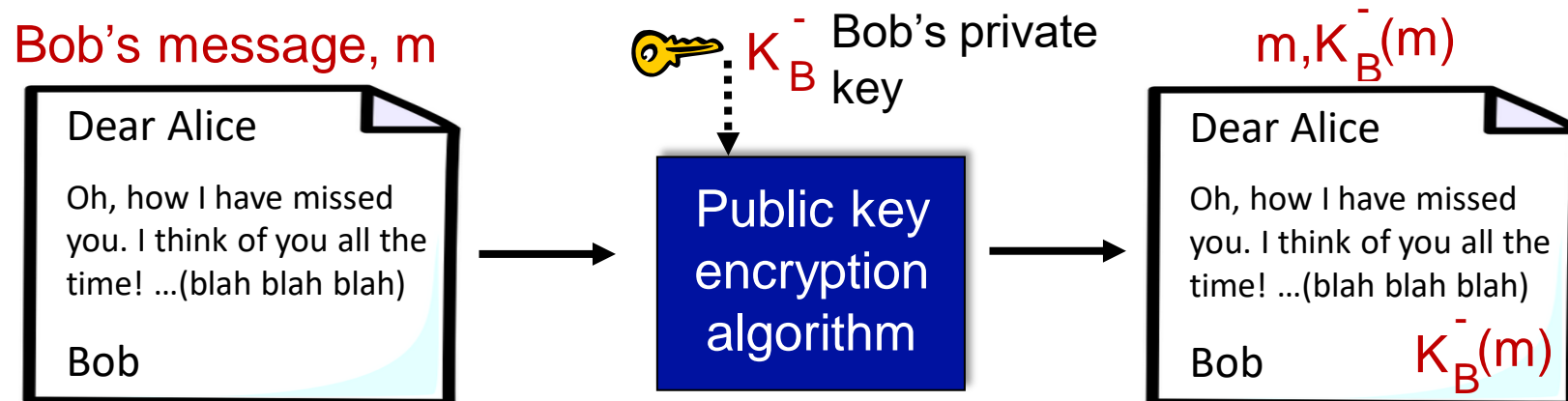
- What is network security?
- Principles of cryptography
- Authentication, **message integrity**
- Securing e-mail
- Securing TCP connections: TLS
- Network layer security: IPsec
- Security in wireless and mobile networks
- Operational security: firewalls and IDS



# Chữ kí điện tử

Tương tự như chữ kí tay-không từ chối:

- Bob gửi tài liệu do anh ấy kí: anh ta là người sở hữu tài liệu
- *Xác thực, không từ chối*: người nhận (Alice) có thể chứng minh không ai khác ngoài Bob kí tài liệu này
- **simple digital signature for message  $m$ :**
  - Bob dùng khóa bí mật mã hóa thông điệp  $m$ ,  $K_B^-(m)$



# Chữ kí điện tử

- Alice nhận thông điệp  $m$ , với chữ kí:  $m, K_B^-(m)$
- Alice kiểm tra  $m$  được kí bởi Bob bằng việc kiểm tra khóa công khai  $K_B^+(m)$  sau đó kiểm tra  $K_B^+(K_B^-(m)) = m$ .
- If  $K_B^+(K_B^-(m)) = m$ , thì việc kí  $m$  phải được kí bằng khóa bí mật của Bob

## Alice biết được:

- Bob kí  $m$
- Bob kí  $m$  không phải  $m'$

## Không từ chối:

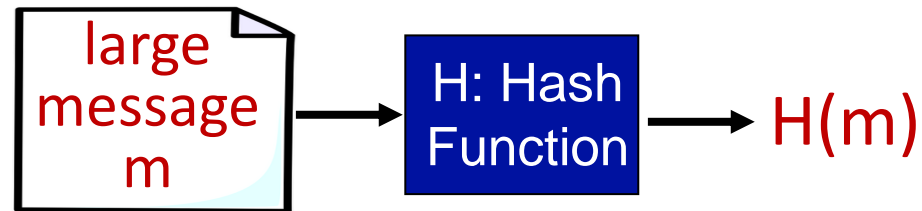
- ✓ Alice có thể lấy  $m$ , và bản đã kí  $K_B^-(m)$  để chứng minh Bob đã kí

# Thông điệp số

Quá đắt đỏ để mã hóa thông điệp dài bằng khóa công khai

**goal:** chiều dài cố định, dễ dàng số hóa “fingerprint”

- Băm  $m$ , được thông điệp số cố định,  $H(m)$



## Thuộc tính hàm băm:


- Thông điệp số cố định(fingerprint)
- Cho thông điệp số là  $x$ , không thể tìm được  $m$  có  $x = H(m)$
- *Many to one*

# Internet checksum: mã hóa băm yếu

Internet checksum có thuộc tính như hàm băm

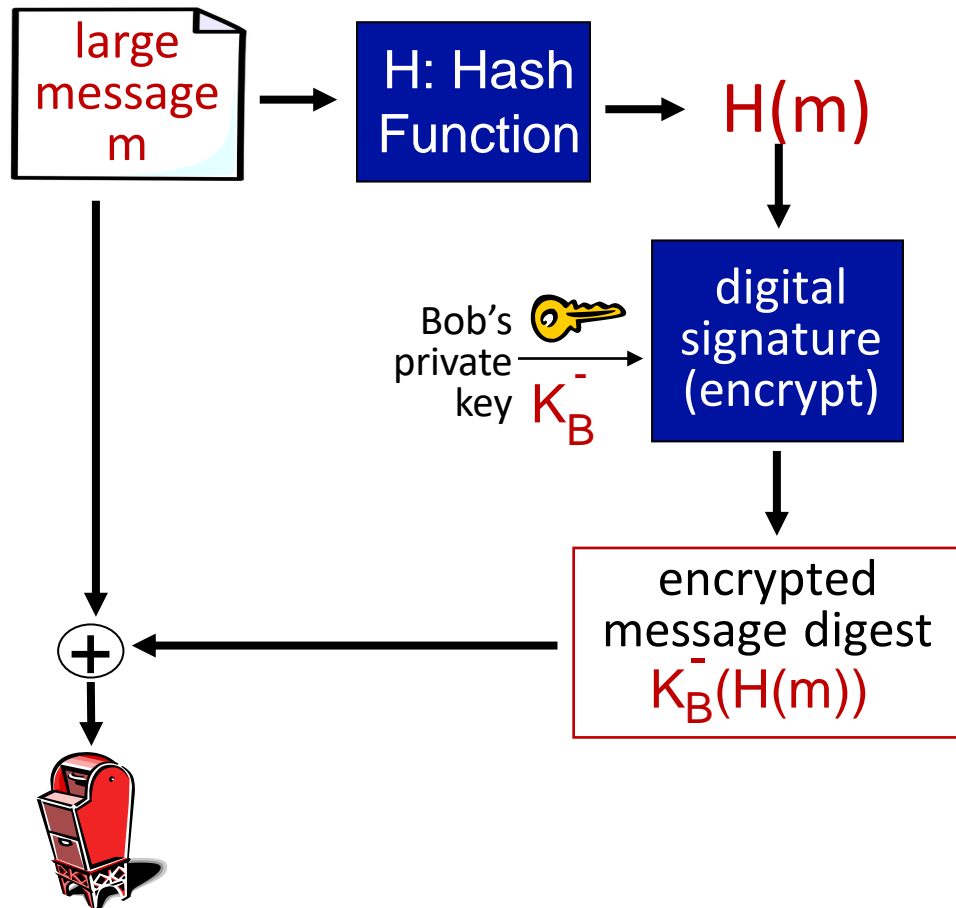
- Tạo thông điệp có chiều dài cố định 16 bit
- Many to one

Cho thông điệp với mảng băm đã biết, dễ dàng tìm thông điệp khác có cùng mảng băm:

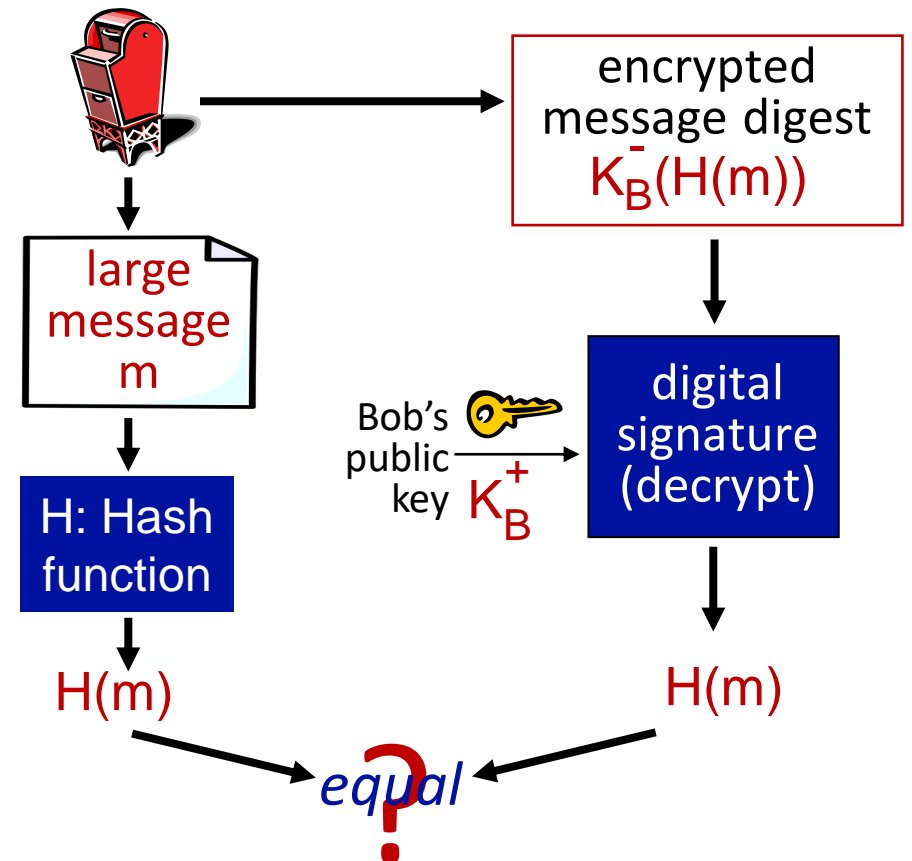
<u>message</u>	<u>ASCII format</u>		<u>message</u>	<u>ASCII format</u>
I O U 1	49 4F 55 31		I O U <u>9</u>	49 4F 55 <u>39</u>
0 0 . 9	30 30 2E 39		0 0 . <u>1</u>	30 30 2E <u>31</u>
9 B O B	39 42 D2 42		9 B O B	39 42 D2 42
<hr/>			<hr/>	
B2 C1 D2 AC		 <i>different messages</i> <i>but identical checksums!</i>	B2 C1 D2 AC	

# Digital signature = signed message digest

Bob gửi thông điệp số được kí:



Alice kiểm tra chữ kí, tính toán  
vẹn dữ liệu đã kí:



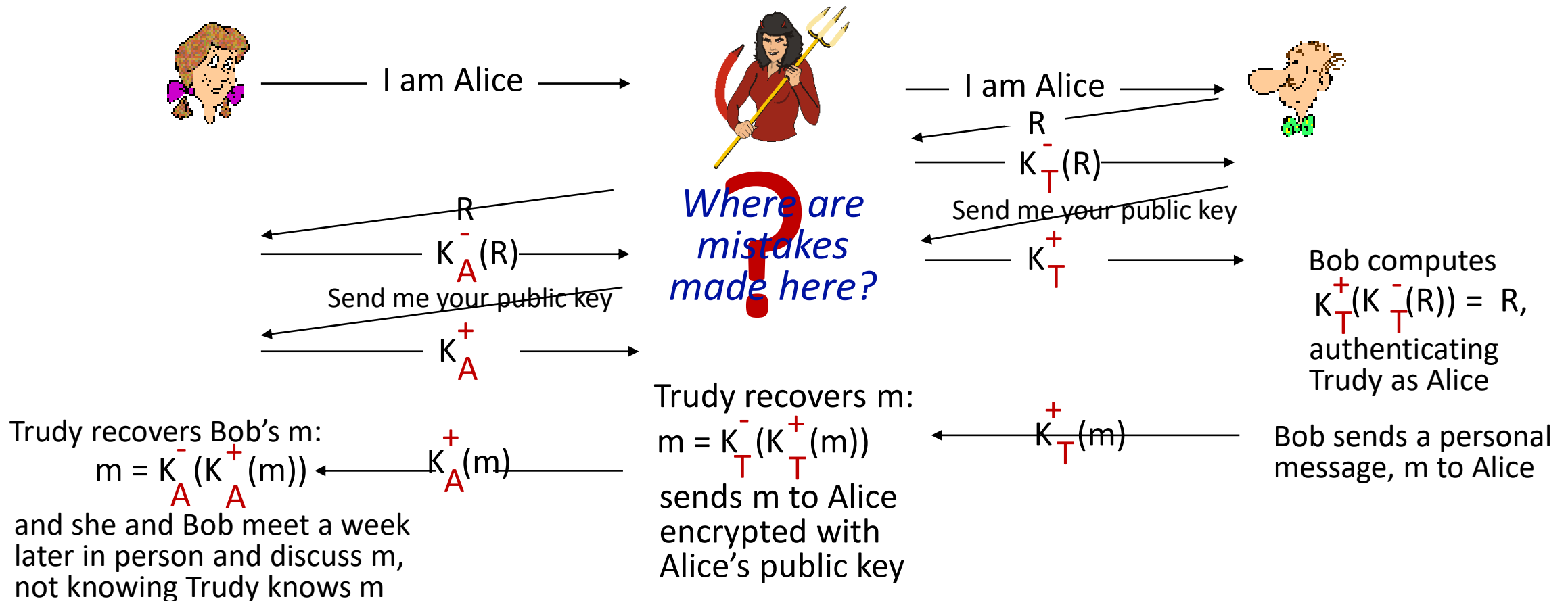
# Giải thuật băm

- MD5 được dùng rộng rãi(RFC 1321)
  - Tính toán thông điệp số 128 bit qua 4 bước.
  - Với thông điệp số  $x$  là 128 bit, khó có thông điệp  $m$  có cùng  $x$
- SHA-1 cũng được dùng
  - US standard [NIST, FIPS PUB 180-1]
  - 160-bit thông điệp số



# Authentication: ap5.0 – let's fix it!!

**Recall the problem:** Trudy poses as Alice (to Bob) and as Bob (to Alice)



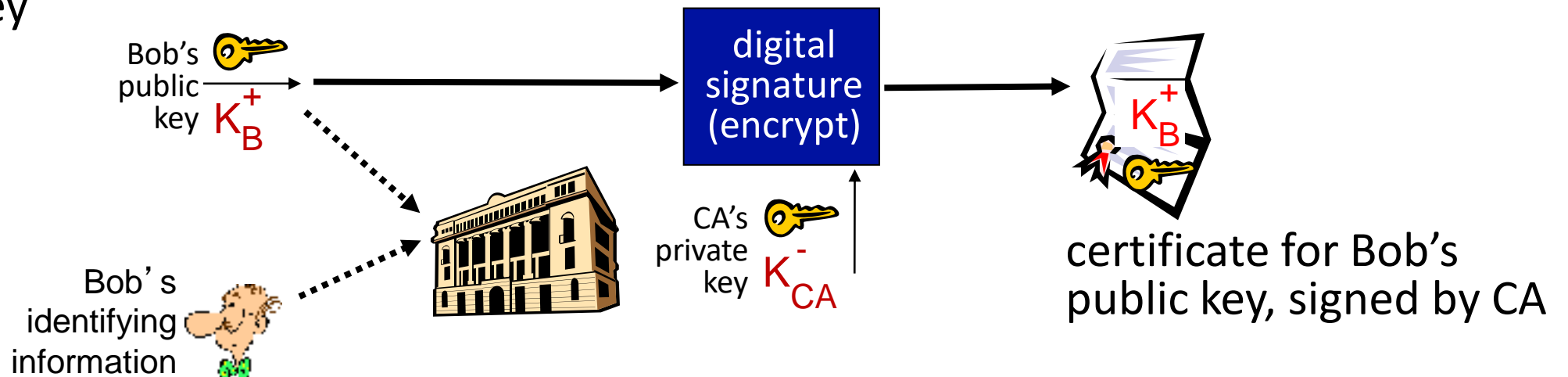
# Cần chứng thực khóa công khai

- motivation: Trudy chơi khăm Bob
  - Trudy tạo email:  
*Dear Pizza Store, Please deliver to me four pepperoni pizzas. Thank you, Bob*
  - Trudy kí với khóa bí mật của cô ta
  - Trudy gửi order tới Pizza Store
  - Trudy gửi Pizza Store khóa công khai và nói nó là khóa công khai của Bob
  - Pizza Store kiểm tra chữ kí; gửi bánh tới Bob
  - Bob không thích pepperoni



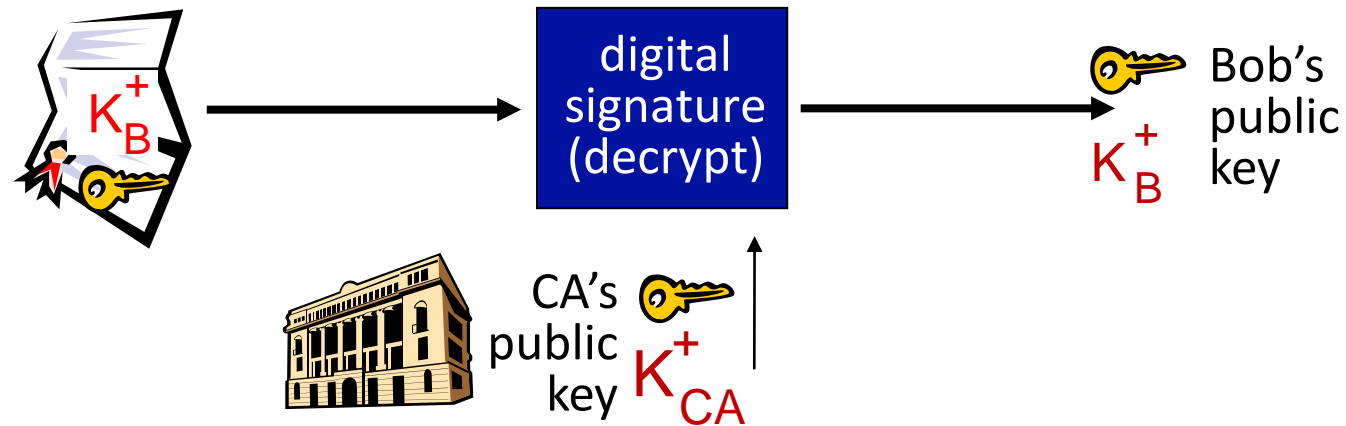
# Chứng nhận khóa công khai(CA)

- **certification authority (CA):** chứng thực khóa công khai gắn với một thực thể cụ thể
- Thực thể (person, website, router) đăng kí khóa công khai để chứng minh khóa công khai là của thực thể
  - CA tạo định danh gắn E tới khóa công khai của E
  - Chứng thực chứa khóa công khai được kí bởi CA: CA nói “this is E’s public key”



# Chứng nhận khóa công khai(CA)

- khi Alice muốn Bob's public key:
  - Lấy chứng nhận của Bob(Bob or elsewhere)
  - Áp khóa công khai của CA vào chứng nhận của Bob biết được Bob's public key



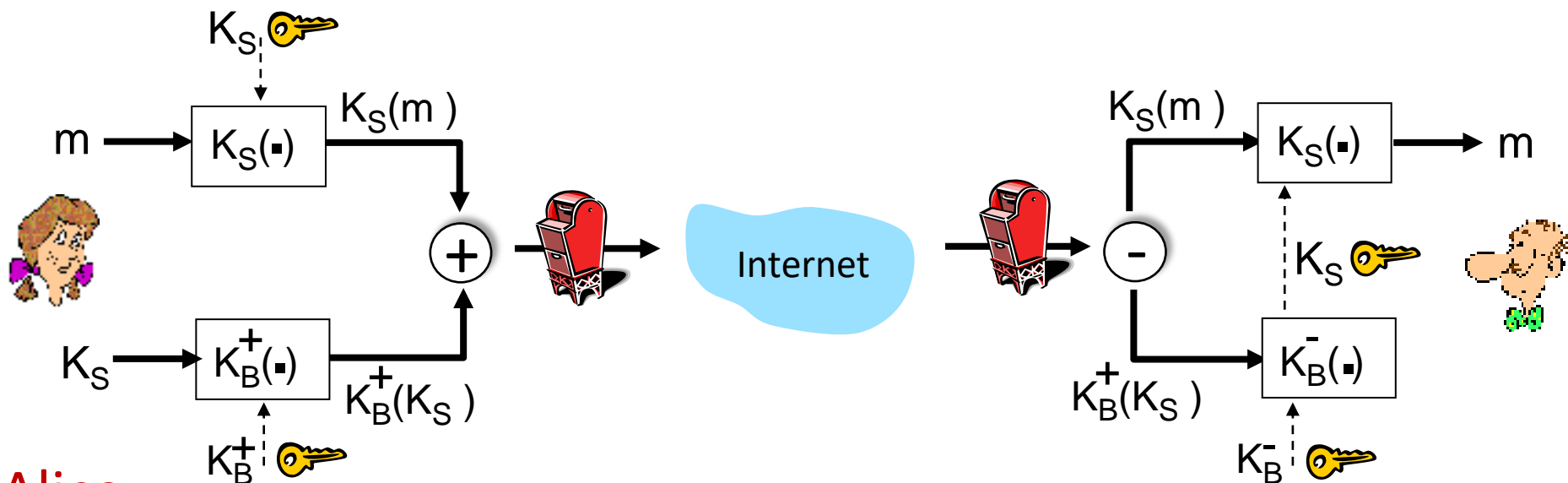
# Chapter 8 outline

- What is network security?
- Principles of cryptography
- Authentication, message integrity
- **Securing e-mail**
- Securing TCP connections: TLS
- Network layer security: IPsec
- Security in wireless and mobile networks
- Operational security: firewalls and IDS



# Secure e-mail: bí mật

Alice muốn gửi e-mail bí mật,  $m$ , tới Bob.

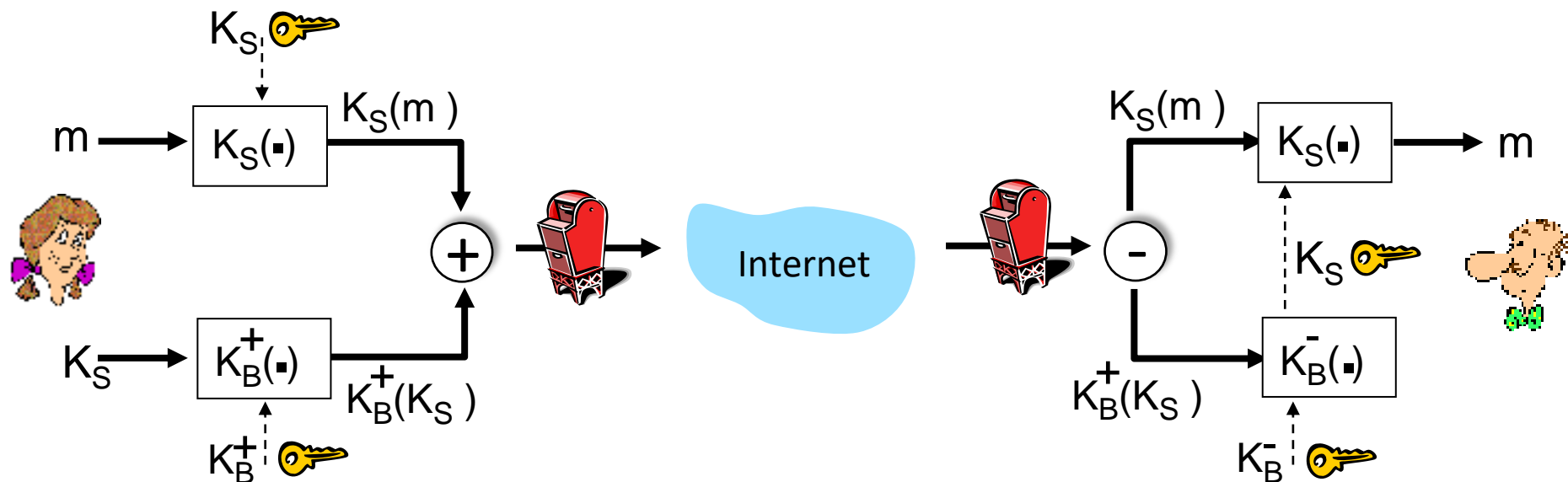


**Alice:**

- Tạo khóa đối xứng,  $K_S$
- Mã hóa thông điệp với  $K_S$  (for efficiency)
- Mã  $K_S$  với khóa công khai của Bob
- Gửi  $K_S(m)$  và  $K_B^+(K_S)$  tới Bob

# Secure e-mail: confidentiality (more) (bí mật)

Alice muốn gửi e-mail bí mật ,  $m$ , tới Bob.

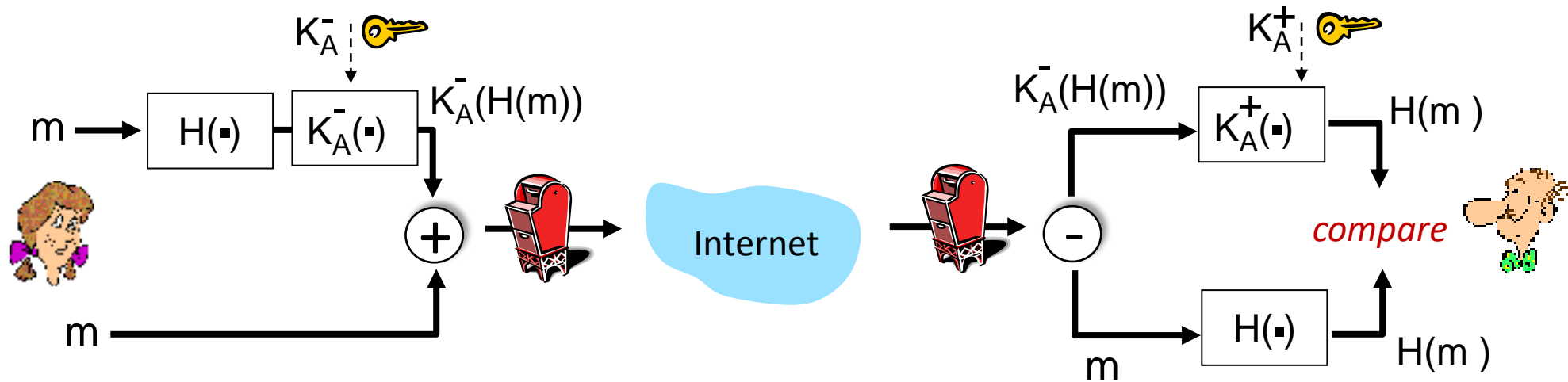


**Bob:**

- Sử dụng khóa bí mật để giải mã và khôi phục  $K_S$
- Dùng  $K_S$  để giải mã thông điệp

# Secure e-mail: integrity, authentication

Alice muốn gửi tới Bob, thông điệp toàn vẹn, xác thực

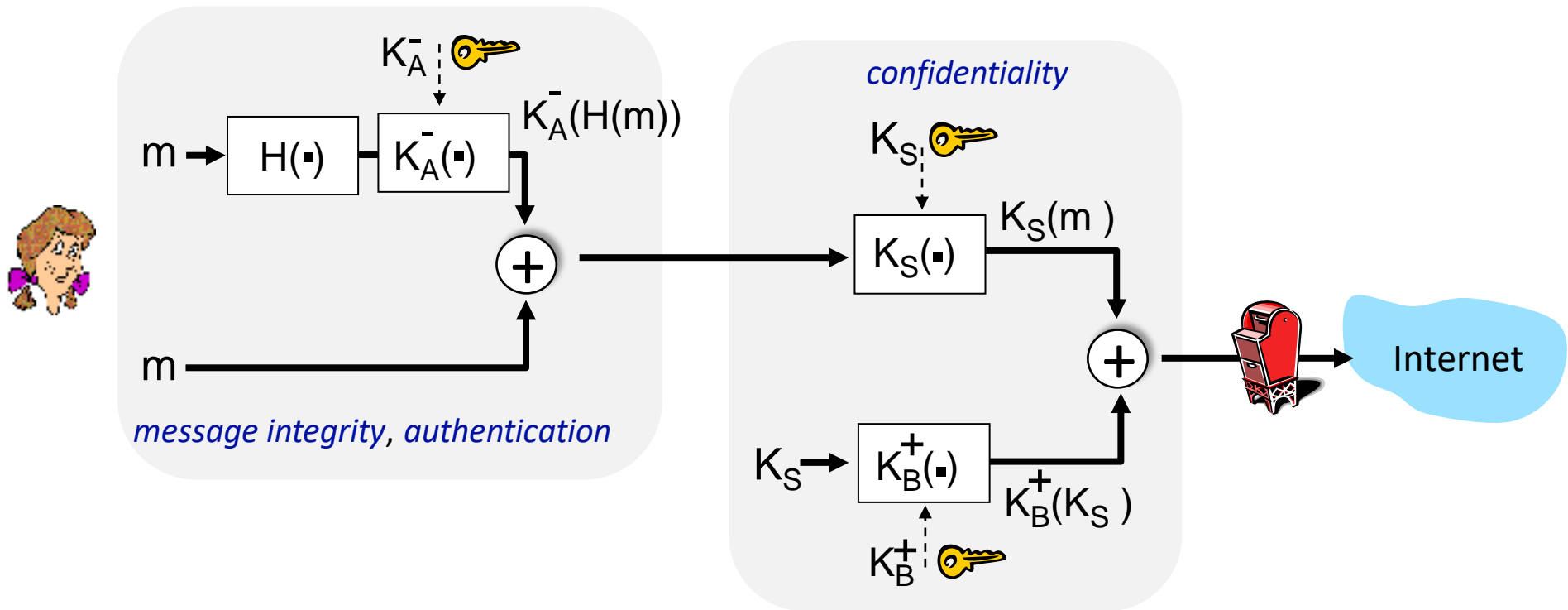


- Alice kí mảng băm thông điệp số với khóa bí mật như vậy có toàn vẹn và xác thực
- Gửi thông điệp(ban đầu) và chữ kí số



# Secure e-mail: integrity, authentication

Alice gửi thông điệp tới Bob *confidentiality, message integrity, authentication*



**Alice sử dụng ba khóa:** her private key, Bob's public key, new symmetric key

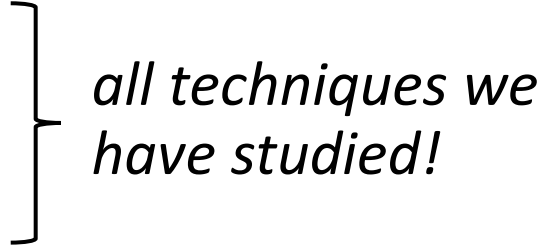
*Bob sẽ nhận như thế nào và kiểm tra ra  
sao*

# Chapter 8 outline

- What is network security?
- Principles of cryptography
- Authentication, message integrity
- Securing e-mail
- **Securing TCP connections: TLS**
- Network layer security: IPsec
- Security in wireless and mobile networks
- Operational security: firewalls and IDS



# Transport-layer security (TLS)

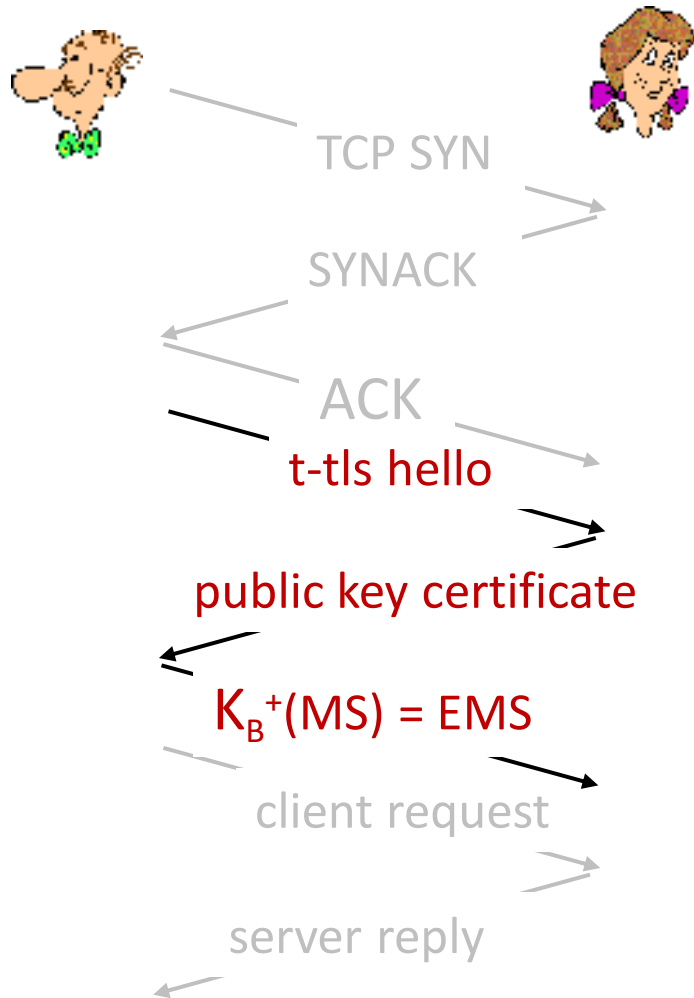
- Được dùng rộng rãi để bảo vệ kết nối tầng transport layer
  - Hỗ trợ cho toàn bộ trình duyệt web và máy chủ web: https (port 443)
- Cung cấp:
  - **confidentiality**: mã hóa đối xứng
  - **integrity**: băm bản mã
  - **authentication**: mã hóa công khai

*all techniques we have studied!*
- Lịch sử:
  - early research, implementation: secure network programming, secure sockets
  - secure socket layer (SSL) deprecated [2015]
  - TLS 1.3: RFC 8846 [2018]

# Transport-layer security: cần cái gì?

- Giả sử xây dựng, *t-tls*, để hiểu về TLS!
- Ta có:
  - **handshake**: Alice, Bob sử dụng chứng thực, khóa bí mật để xác thực lẫn nhau, trao đổi hoặc tạo khóa bí mật chung
  - **key derivation**: Alice, Bob sử dụng khóa bí mật chung để tạo khóa khác
  - **data transfer**: truyền dữ liệu nối tiếp nhau
    - Không phải giao dịch một lần
  - **connection closure**: thông điệp cụ thể để đóng kết nối

# t-tls: initial handshake



## t-tls handshake phase:

- Bob thiết lập kết nối TCP với Alice
- Bob kiểm tra Alice là Alice
- Bob gửi Alice một master secret key (MS), được dùng tạo ra tất cả các khóa khác cho phiên TLS
- Vấn đề:
  - Cần 3 RTT trước khi truyền dữ liệu đi (tăng độ trễ)

# t-tls: cryptographic keys

- Dùng khóa khác nhau
- four keys:
  - 🔑  $K_c$  : encryption key for data sent from client to server
  - 🔑  $M_c$  : MAC key for data sent from client to server
  - 🔑  $K_s$  : encryption key for data sent from server to client
  - 🔑  $M_s$  : MAC key for data sent from server to client
- keys được tạo từ hàm sinh khóa(KDF)
  - Từ khóa master secret có thể thêm dữ liệu ngẫu nhiên tạo thêm khóa mới

# t-tls: mã hóa dữ liệu

- recall: TCP truyền dữ liệu theo kiểu luồng
- Q: cần mã hóa luồng dữ liệu trong TCP socket?
  - A: MAC đi đâu? If cuối cùng, thông điệp toàn vẹn nếu tất cả dữ liệu được nhận, đóng kết nối
  - solution: chia luồng thành các khối record
    - Mỗi record mang 1 MAC, sử dụng  $M_c$
    - Người nhận có thể xử lý từng record
- t-tls record mã hóa dùng khóa đối xứng,  $K_c$ , truyền tới TCP:

$K_c$  ( 

<i>length</i>	<i>data</i>	<i>MAC</i>
---------------	-------------	------------

 )

# t-tls: encrypting data (more)

- Có thể tấn công luồng dữ liệu?
  - *re-ordering*: man-in middle chặn bắt thông điệp TCP segments và thay đổi thứ tự (điều chỉnh sequence #s trong header TCP không được mã hóa)
  - *Truyền lại*
- Giải pháp:
  - dùng TLS sequence numbers (data, TLS-seq-# incorporated into MAC)
  - Sử dụng nonce



# t-tls: đóng kết nối

- truncation attack:
  - Kẻ tấn công giả vờ đóng kết nối
  - Một trong hai bên sẽ tưởng là ít dữ liệu hơn thực tế
- **solution:** kiểu record báo đóng mở
  - type 0 for data; type 1 for close
- MAC tính toán data, type, sequence #

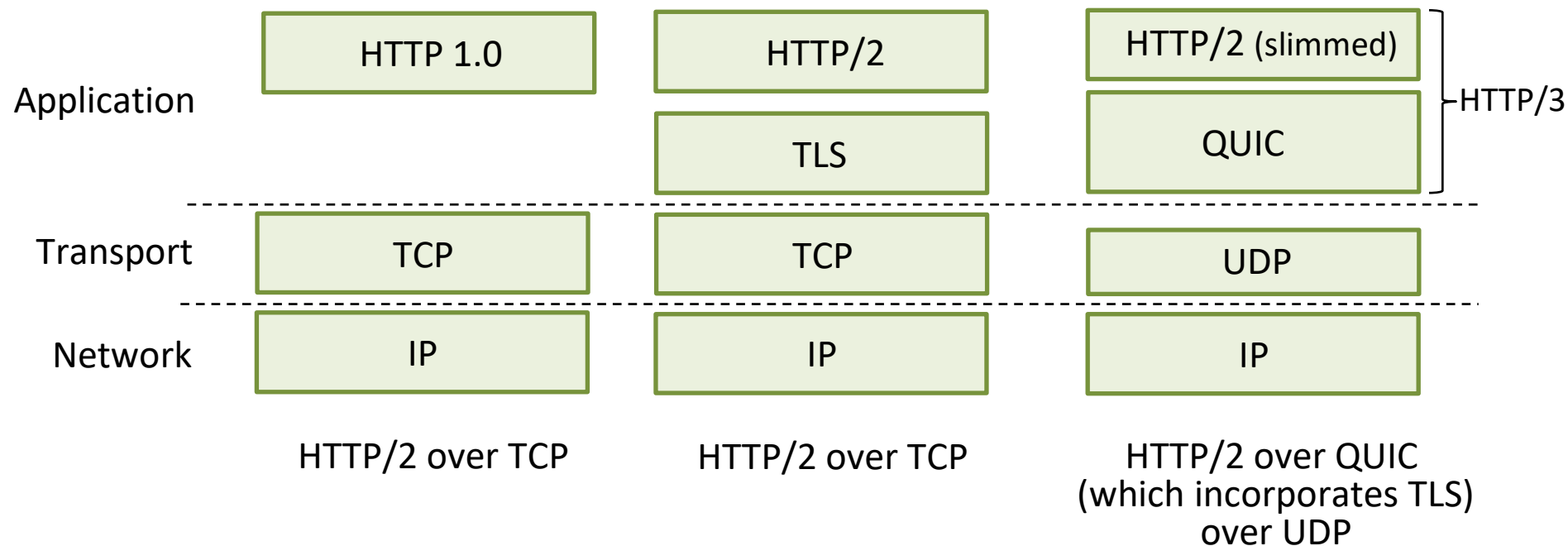
$K_c$  ( 

<i>length</i>	<i>type</i>	<i>data</i>	<i>MAC</i>
---------------	-------------	-------------	------------

 )

# Transport-layer security (TLS)

- TLS cung cấp API cho bất kì ứng dụng nào có thể sử dụng
- an HTTP view of TLS:



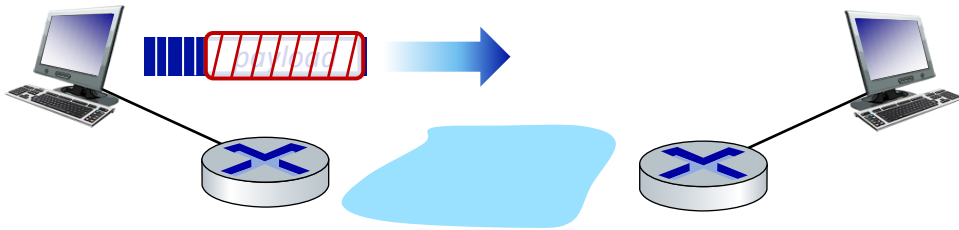
# Chapter 8 outline

- What is network security?
- Principles of cryptography
- Authentication, message integrity
- Securing e-mail
- Securing TCP connections: TLS
- **Network layer security: IPsec**
- Security in wireless and mobile networks
- Operational security: firewalls and IDS



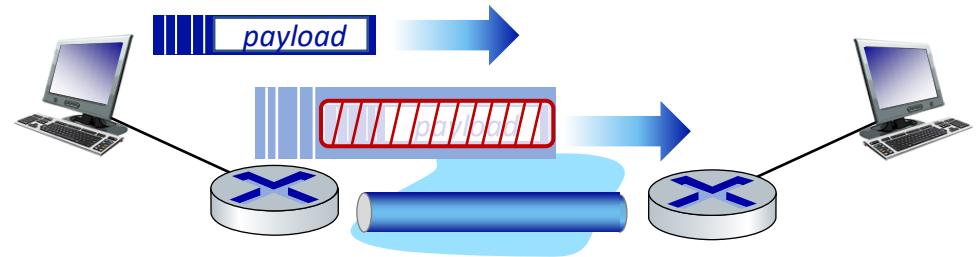
# IP Sec

- Hỗ trợ encryption, authentication, integrity cho tầng mạng
  - Cả lưu lượng dữ liệu và lưu lượng điều khiển (e.g., BGP, DNS messages)
- Hai chế độ:



## transport mode:

- *only* datagram *payload* is encrypted, authenticated



## tunnel mode:

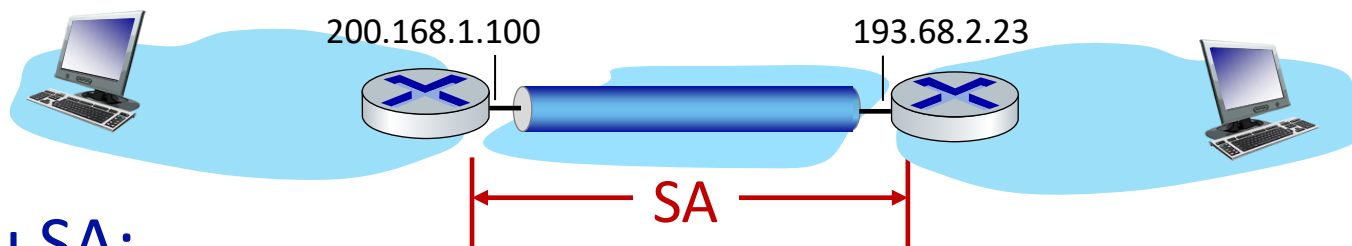
- entire datagram is encrypted, authenticated
- encrypted datagram encapsulated in new datagram with new IP header, tunneled to destination

# Two IPsec protocols

- Authentication Header (AH) protocol [RFC 4302]
  - Cung cấp xác thực nguồn, toàn vẹn nhưng không bí mật
- Encapsulation Security Protocol (ESP) [RFC 4303]
  - Cung cấp: authentication nguồn, data integrity, *and confidentiality*
  - Được dùng rộng rãi hơn AH

# Security associations (SAs)

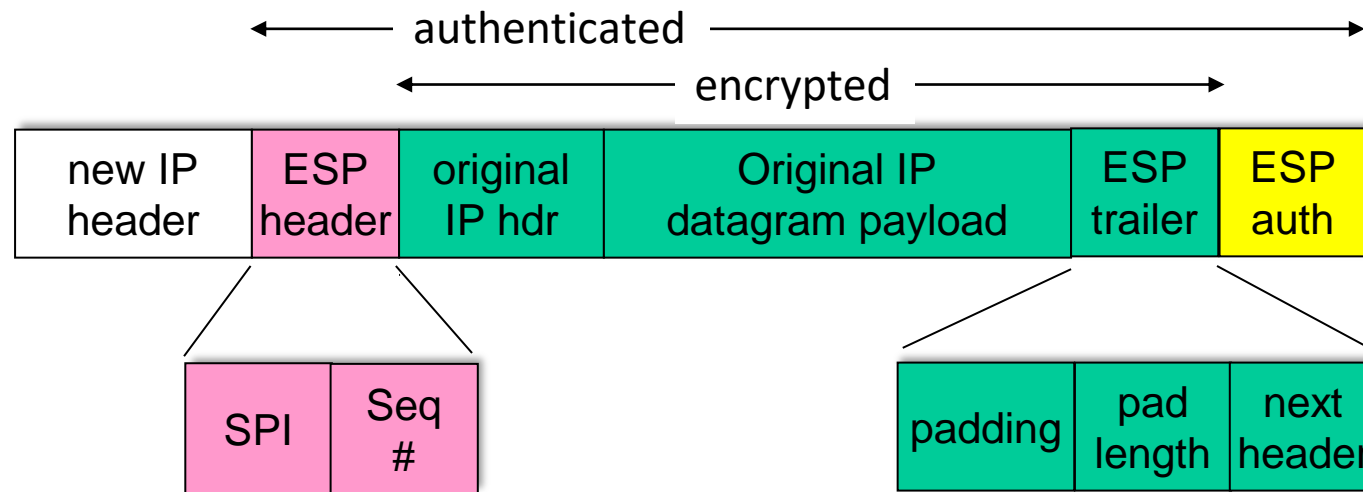
- Trước khi truyền dữ liệu, **security association (SA)** thiết lập giữa thực thể gửi và nhận (directional)
- Kết thúc, thực thể nhận duy trì trạng thái thông tin SA
  - recall: TCP có thể lưu trạng thái kết nối
  - IP không hướng kết nối; IPsec hướng kết nối!



## R1 lưu SA:

- 32-bit identifier: *Security Parameter Index (SPI)*
- origin SA interface (200.168.1.100)
- destination SA interface (193.68.2.23)
- Kiểu mã hóa được dùng
- Khóa mã hóa
- Kiểu kiểm tra tính toàn vẹn
- Khóa xác thực

# IPsec datagram



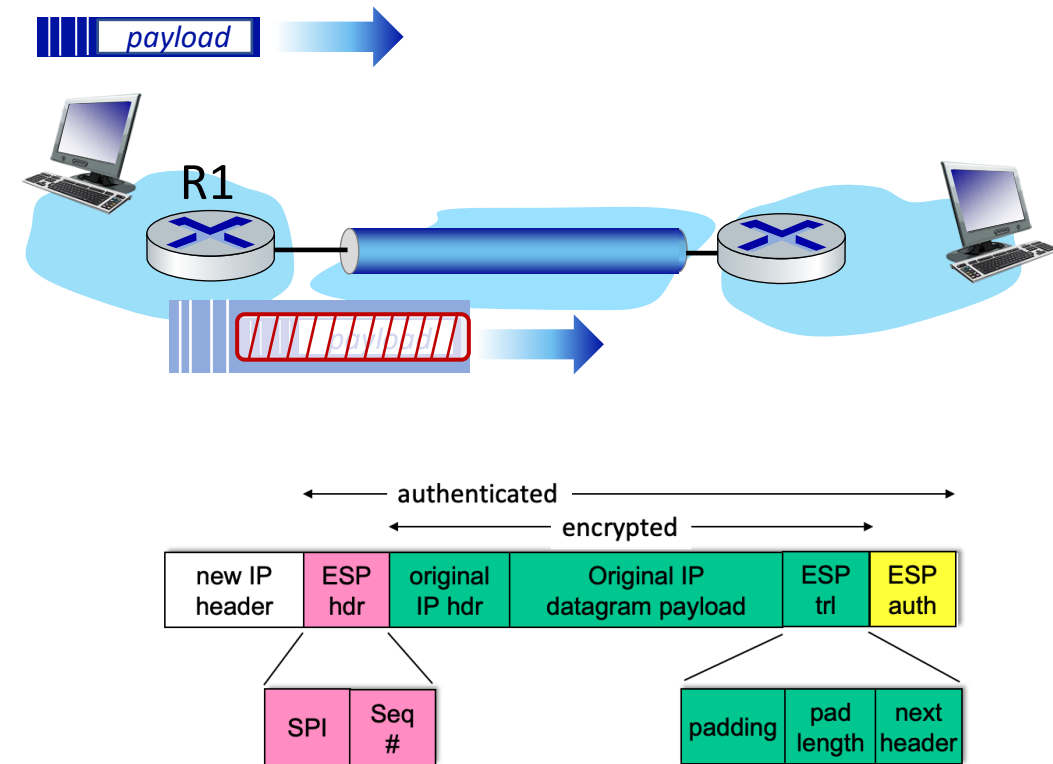
*tunnel mode  
ESP*

- ESP trailer: thêm vào khối mã hóa
- ESP header:
  - SPI, thực thể nhận biết làm cái gì
  - sequence number, chống tấn công phát lại
- MAC trong ESP xác thực các trường đã tạo dùng khóa bí mật dùng chung

# ESP tunnel mode: actions

## Tại R1:

- thêm ESP trailer đến datagram
- Mã hóa sử dụng giải thuật và khóa quy định trong SA
- thêm ESP header phía trước của kết quả mã hóa
- Tạo MAC dùng thuật toán mã hóa và khóa trong SA
- thêm MAC dạng *payload*
- Tạo IP header mới với các trường mới để đi qua đường hầm tới bên kia





# IPsec sequence numbers

- Với SA mới, người gửi khởi tạo seq. # là 0
- Mỗi lần datagram được gửi trên SA:
  - Người gửi tăng seq # counter
  - Đặt vào trường seq # field
- Mục tiêu:
  - Ngăn chặn kẻ tấn công sniff và phát lại packet
  - Loại bỏ việc trùng lặp
- Phương thức:
  - Đích kiểm tra sự trùng lặp
  - Không lưu vết toàn bộ gói đã nhận; thay vào đó sử dụng window

# Cơ sở dữ liệu IPsec

## Security Policy Database (SPD)

- policy: người gửi cần biết nếu có dùng Ipsec
- policy lưu trong **security policy database (SPD)**
- Cần biết SA nào dùng
  - Có thể dùng: source and destination IP address; protocol number

*SAD: “how” to do it*

## Security Assoc. Database (SAD)

- **Điểm cuối lưu trạng thái SA trong security association database (SAD)**
- Khi gửi IPsec datagram, R1 truy cập SAD để quyết định xử lý datagram như thế nào
- khi IPsec datagram tới R2, R2 kiểm tra SPI trong IPsec datagram, đánh chỉ mục SAD với SPI, xử lý datagram

*SPD: “what” to do*

# Tóm tắt: IPsec services



Trudy ngồi giữa R1, R2. không biết các khóa

- Liệu cô ta có biết nội dung gốc của datagram?  
Địa chỉ nguồn, đích, giao thức tầng transport, địa chỉ cổng?
- Có phá hiện lật bit không?
- Đóng giả R1?
- Phát lại datagram?

# IKE: Internet Key Exchange

- *previous examples*: thiết lập IPsec SAs trong các IPsec endpoints:

*Example SA:*

SPI: 12345

Source IP: 200.168.1.100

Dest IP: 193.68.2.23

Protocol: ESP

Encryption algorithm: 3DES-cbc

HMAC algorithm: MD5

Encryption key: 0x7aeaca...

HMAC key: 0xc0291f...

- Không thể thiết lập tay các khóa trong trường hợp VPN với hơn 100 endpoints
- Thay vào đó dùng **IPsec IKE (Internet Key Exchange)**

# IKE: PSK and PKI

- Xác thực
  - pre-shared secret (PSK) hoặc
  - PKI (public/private keys and certificates).
- PSK:
  - Chạy IKE để xác thực lẫn nhau, tạo ra IPsec SAs (một trong mỗi hướng), bao gồm mã hóa và khóa xác thực.
- PKI: các bên bắt đầu với public/private key pair, chứng thực
  - Chạy IKE để xác thực, chứa IPsec SAs (một cho mỗi hướng).
  - Tương tự bắt tay trong SSL.

# IKE phases

- IKE có hai pha
  - *phase 1*: thiết lập IKE SA cả hai hướng
    - note: IKE SA khác với IPsec SA
    - Tương tự ISAKMP security association
  - *phase 2*: tạo các Sas từ thông tin ISAKMP
- phase 1 có hai mode: aggressive mode và main mode
  - aggressive mode dùng ít thông điệp hơn
  - main mode bảo vệ định danh và linh động hơn

# IPsec summary

- IKE message trao đổi algorithms, secret keys, SPI numbers
- AH/ESP protocol (or both)
  - AH cung cấp integrity, source authentication
  - ESP protocol (with AH) additionally provides encryption
- IPsec peers có thể hai end systems, hai routers/firewalls, hoặc một router/firewall và một end system

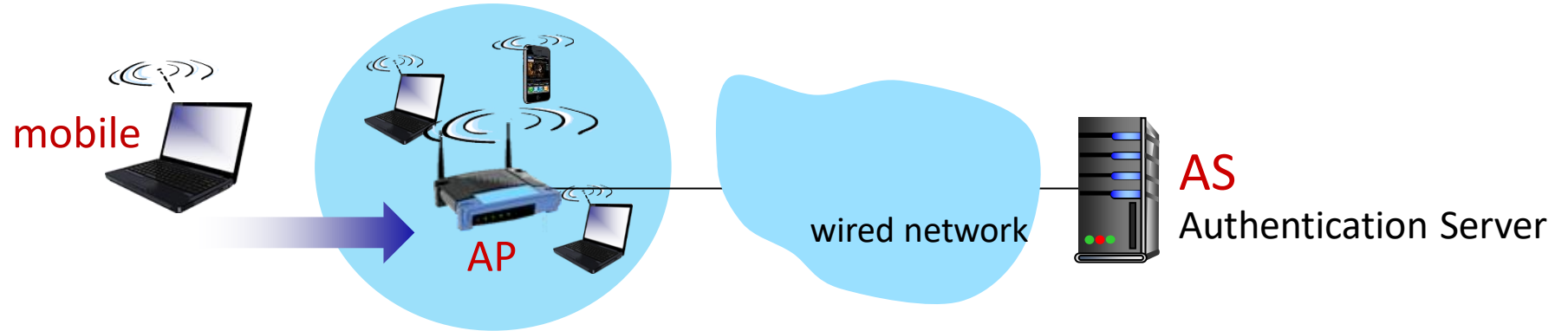
# Chapter 8 outline

- What is network security?
- Principles of cryptography
- Authentication, message integrity
- Securing e-mail
- Securing TCP connections: TLS
- Network layer security: IPsec
- **Security in wireless and mobile networks**
  - 802.11 (WiFi)
  - 4G/5G
- Operational security: firewalls and IDS





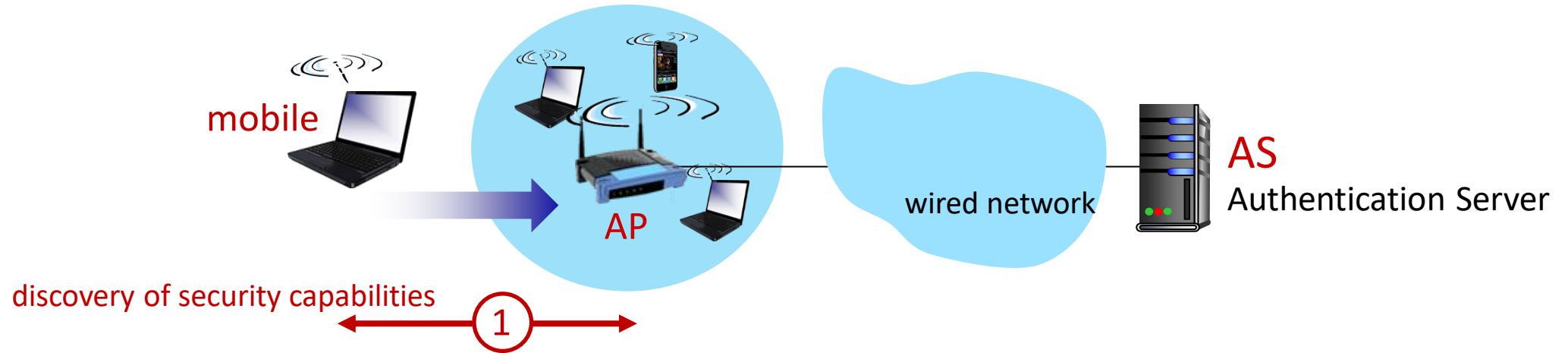
# 802.11: xác thực , mã hóa



Di động đến phải:

- Xác thực với AP để thiết lập kết nối
- Xác thực mạng

# 802.11: authentication, encryption

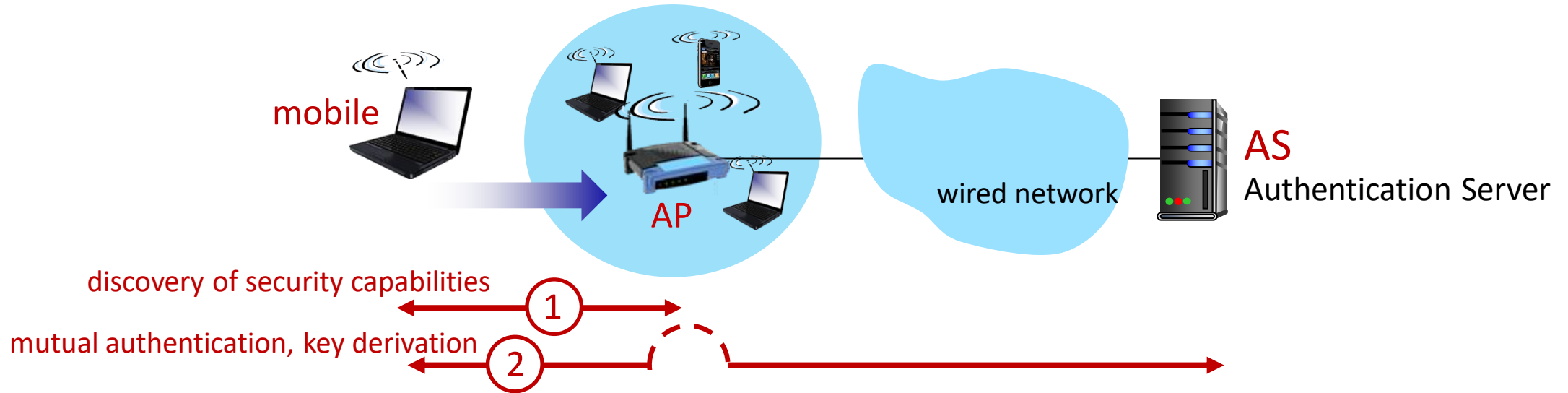


## ① Khám phá khả năng bảo mật:

- AP quảng bá sự hiện diện, form authentication và encryption
- Thiết bị điền vào form authentication, encryption

Mặc dù thiết bị và AP đã trao đổi thông điệp nhưng vẫn chưa xác thực thực sự và không có khóa mã hóa

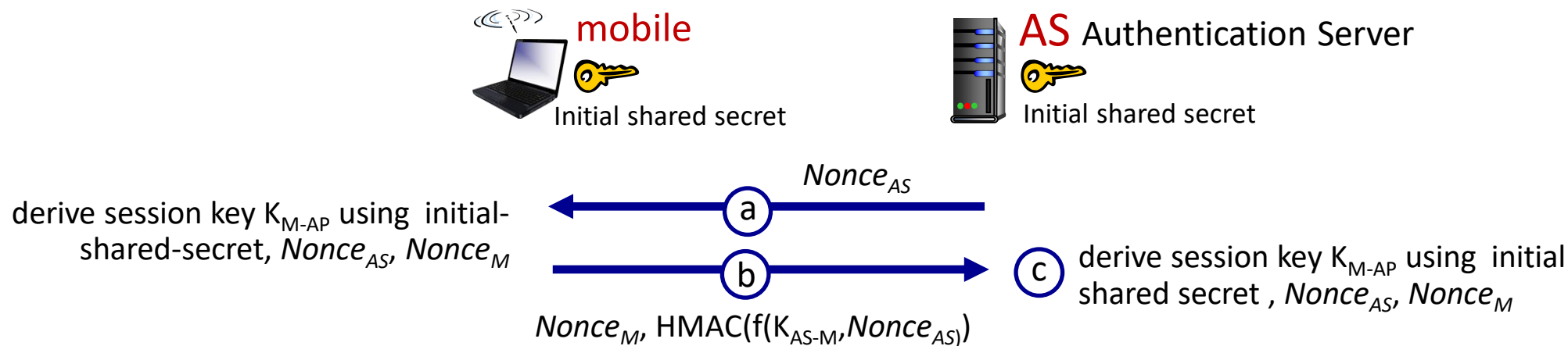
# 802.11: authentication, encryption



## ② Xác thực và chia sẻ khóa đối xứng

- AS, mobile thực sự có hóa bí mật chung (e.g., password)
- AS, mobile dùng khóa bí mật, nonces (chống tấn công phát lại), băm mã hóa (đảm bảo message integrity) để xác thực lẫn nhau
- AS, mobile sử dụng khóa được dẫn xuất

# 802.11: WPA3 handshake



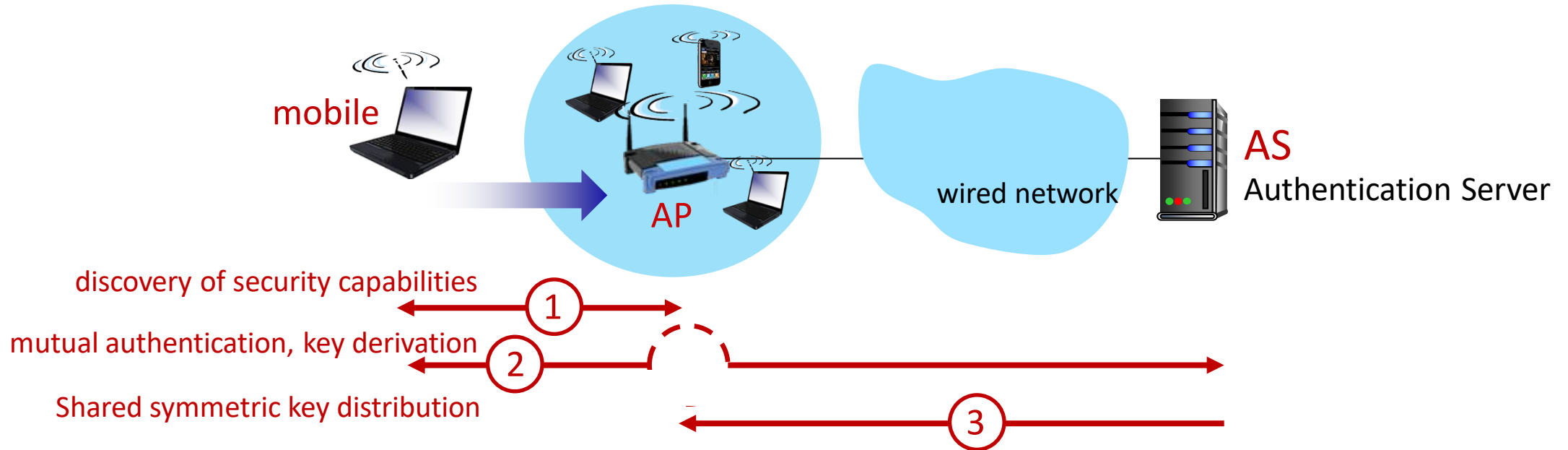
Ⓐ AS tạo ra  $Nonce_{AS}$ , gửi tới mobile

Ⓑ mobile nhận  $Nonce_{AS}$

- Tạo ra  $Nonce_M$
- Tạo khóa đối xứng  $K_{M-AP}$  dùng  $Nonce_{AS}$ ,  $Nonce_M$ , và khởi tạo khóa bí mật chung
- Gửi  $Nonce_M$ , và HMAC-được kí dùng  $Nonce_{AS}$  và khởi tạo khóa bí mật chung

Ⓒ AS phân phối khóa  $K_{M-AP}$

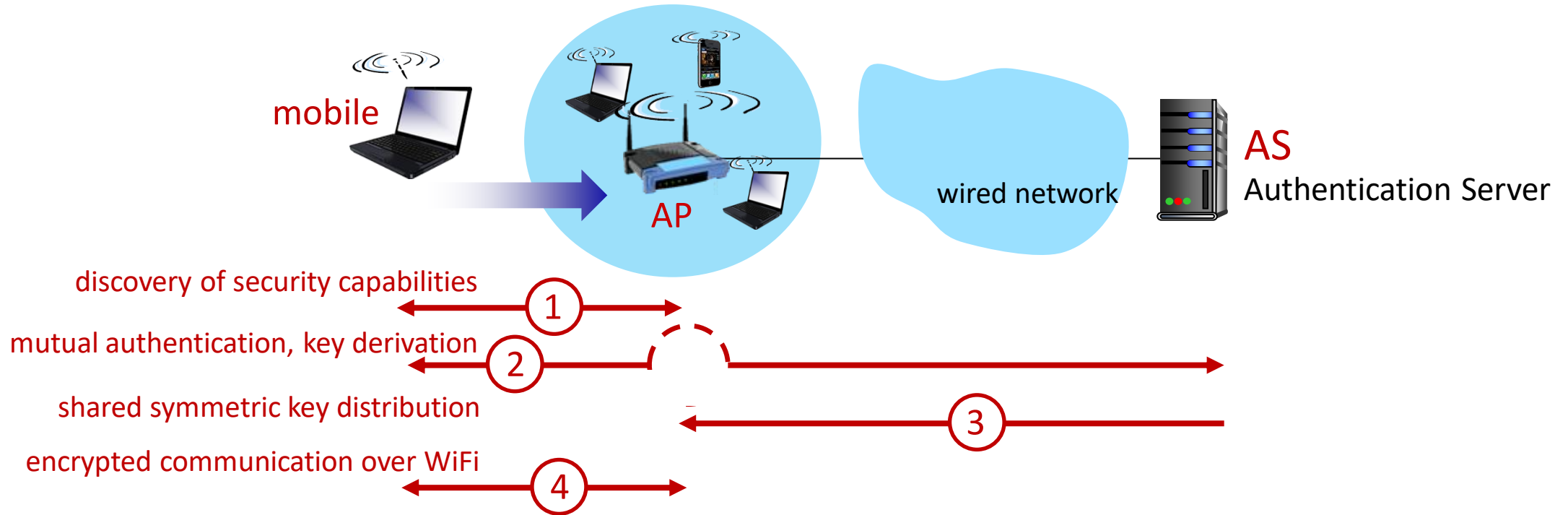
# 802.11: authentication, encryption



## ③ Phân phối khóa(e.g., for AES encryption)

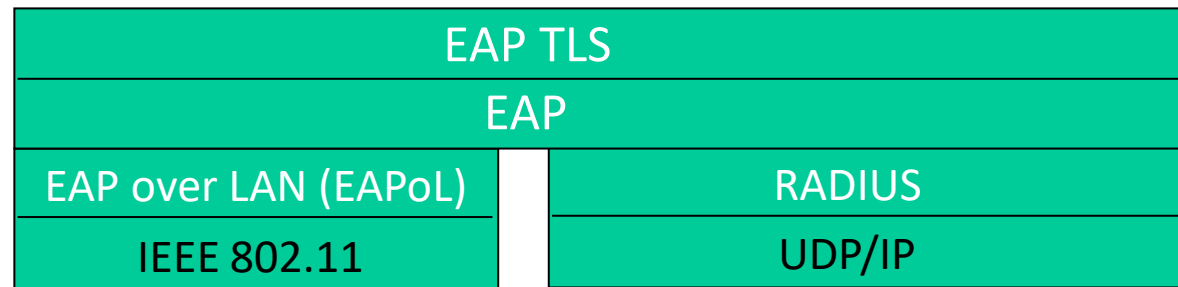
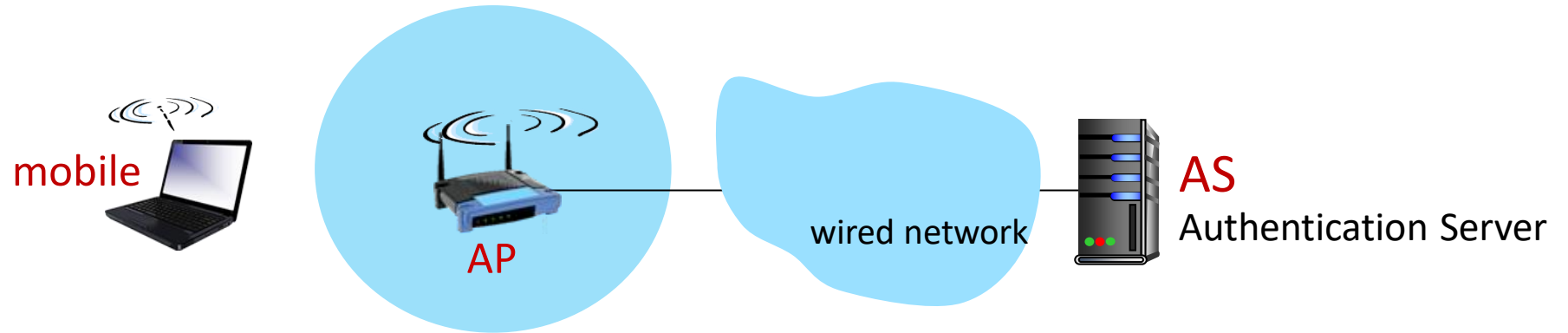
- Cùng khóa được gửi tới mobile, AS
- AS thông tin AP về phiên

# 802.11: authentication, encryption



- ④ Kết nối được mã hóa giữa mobile và remote host thông qua AP
- Cùng khóa được gửi tới mobile, AS
  - AS thông tin AP về phiên

# 802.11: authentication, encryption



- Extensible Authentication Protocol (EAP) [RFC 3748] defines end-to-end request/response protocol between mobile device, AS

# Chapter 8 outline

- What is network security?
- Principles of cryptography
- Authentication, message integrity
- Securing e-mail
- Securing TCP connections: TLS
- Network layer security: IPsec
- **Security in wireless and mobile networks**
  - 802.11 (WiFi)
  - 4G/5G
- Operational security: firewalls and IDS





# Xác thực và mã hóa trong 4G LTE



## ■ Di động phải:

- Xác thực với BS: thiết lập kết nối với 4G wireless link
- Xác thực với mạng

## ■ Khác với WiFi

- Thẻ SIMcard cung cấp định danh toàn cầu, chưa khóa chia sẻ
- Dịch vụ trong các mạng thiết bị di động đi tới tùy thuộc vào các dịch vụ đã trả tiền và đăng kí trong mạng home network

# Authentication, encryption: from 4G to 5G

- **4G:** MME in visited network makes authentication decision
- **5G:** home network provides authentication decision
  - visited MME plays “middleman” role but can still reject
- **4G:** uses shared-in-advance keys
- **5G:** keys not shared in advance for IoT
- **4G:** device IMSI transmitted in cleartext to BS
- **5G:** public key crypto used to encrypt IMSI

# Chapter 8 outline

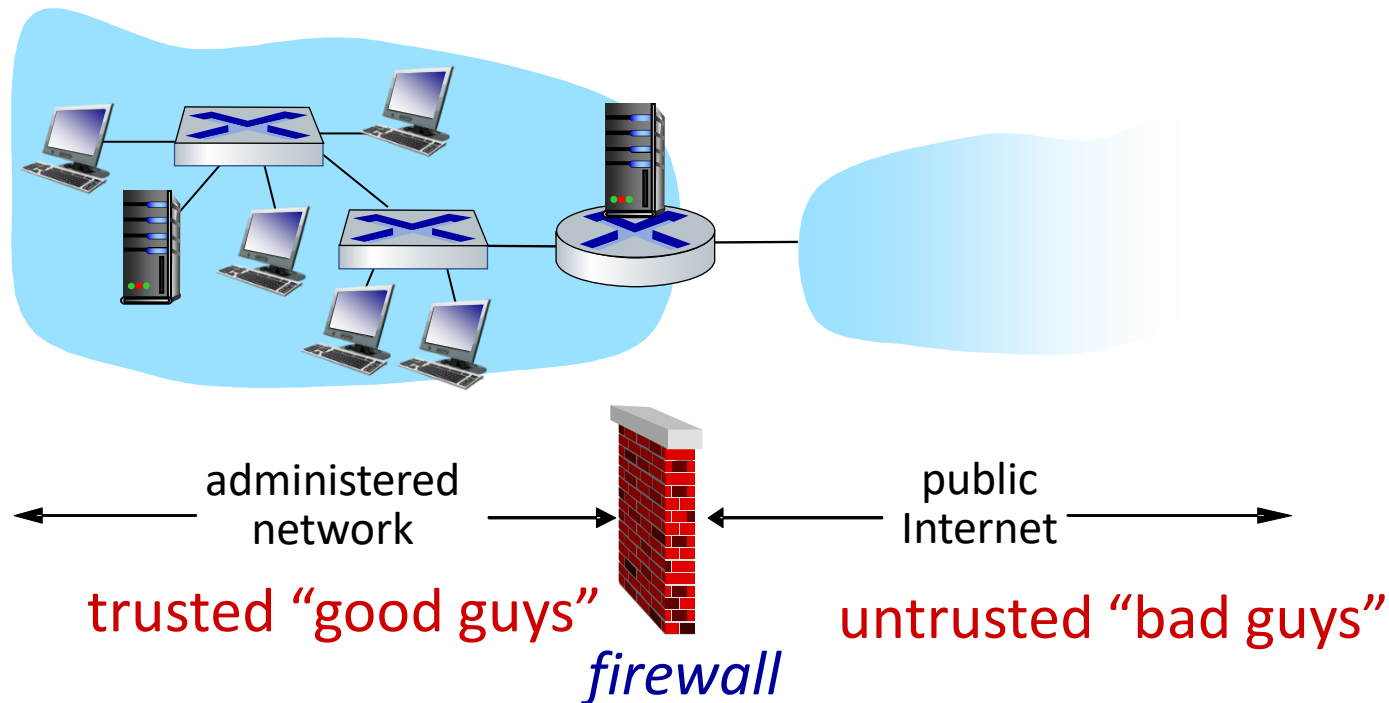
- What is network security?
- Principles of cryptography
- Authentication, message integrity
- Securing e-mail
- Securing TCP connections: TLS
- Network layer security: IPsec
- Security in wireless and mobile networks
- **Operational security: firewalls and IDS**



# Firewalls

## firewall

Cô lập mạng nội bộ với mạng Internet, cho qua hoặc khóa lại lưu lượng tùy vào chính sách



# Firewalls: tại sao

Ngăn chặn tấn công từ chối dịch vụ:

- SYN flooding: kẻ tấn công tạo cùng lúc nhiều yêu cầu kết nối giả mạo

Ngăn chặn truy cập trái luật vào mạng nội bộ

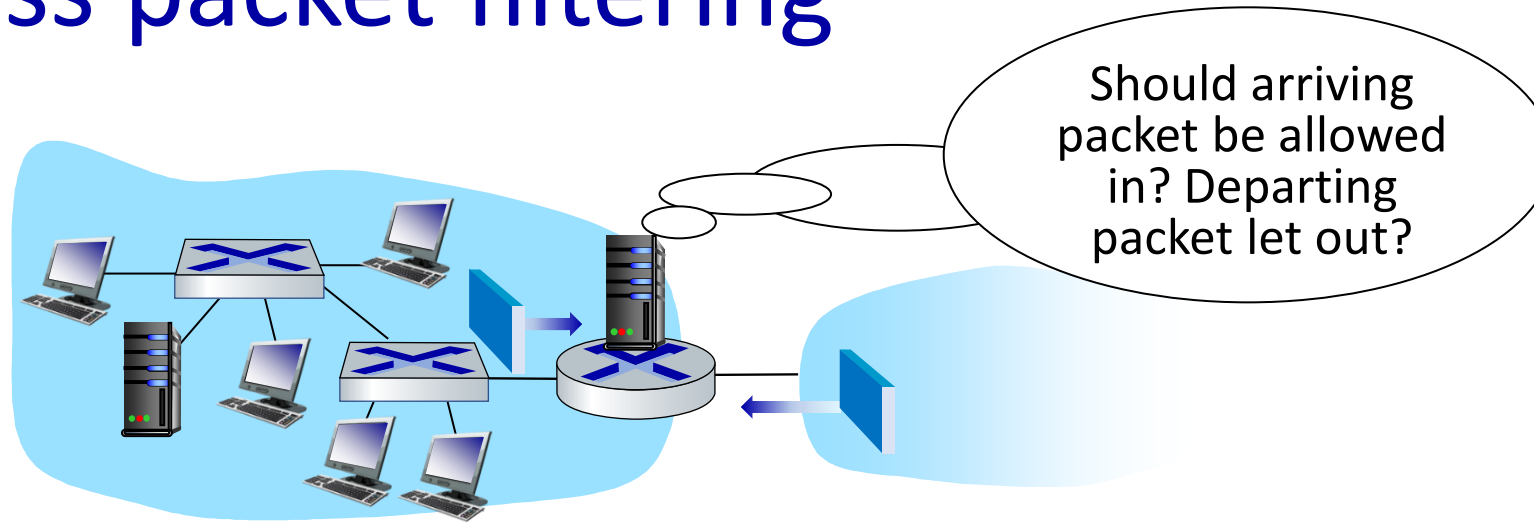
Cho phép các truy cập được cho phép vào mạng nội bộ

- xác thực users/hosts

Ba kiểu firewalls:

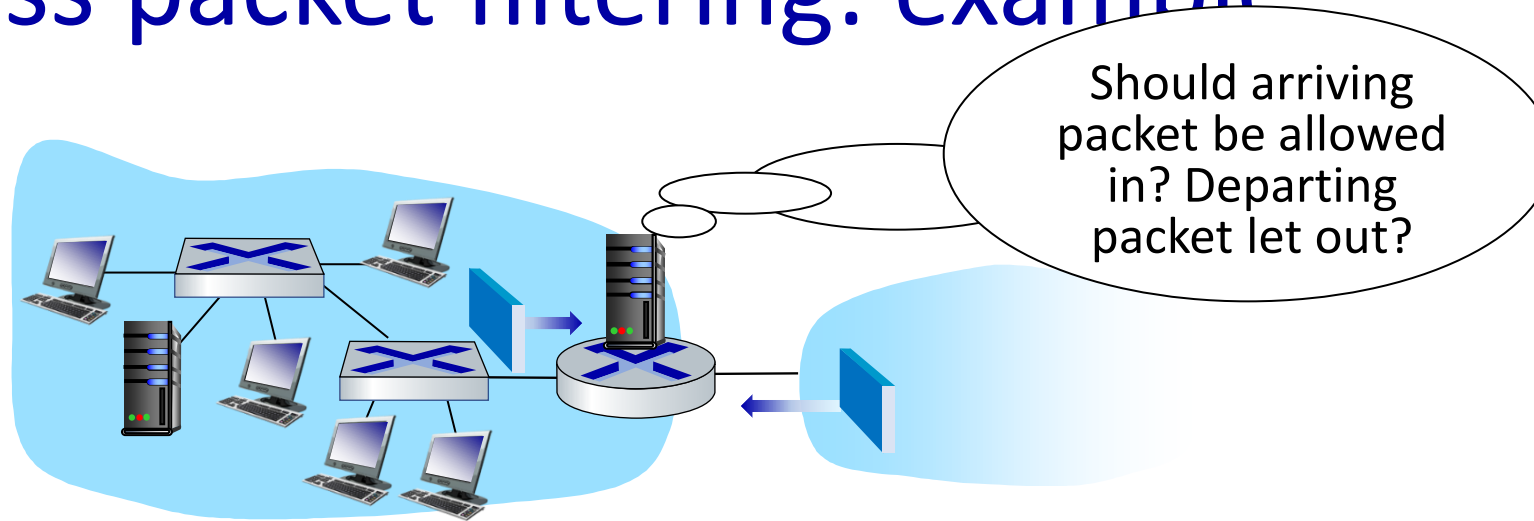
- stateless packet filters
- stateful packet filters
- application gateways

# Stateless packet filtering



- internal network connected to Internet via router **firewall**
- filters **packet-by-packet, luật dựa trên**:
  - source IP address, destination IP address
  - TCP/UDP source, destination port numbers
  - ICMP message type
  - TCP SYN, ACK bits

# Stateless packet filtering: example



- **example 1:** block incoming and outgoing datagrams with IP protocol field = 17 and with either source or dest port = 23
  - **result:** all incoming, outgoing UDP flows and telnet connections are blocked
- **example 2:** block inbound TCP segments with ACK=0
  - **result:** prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside

# Stateless packet filtering: more examples

Policy	Firewall Setting
no outside Web access	drop all outgoing packets to any IP address, port 80
no incoming TCP connections, except those for institution's public Web server only.	drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80
prevent Web-radios from eating up the available bandwidth.	drop all incoming UDP packets - except DNS and router broadcasts.
prevent your network from being used for a smurf DoS attack.	drop all ICMP packets going to a "broadcast" address (e.g. 130.207.255.255)
prevent your network from being tracerouted	drop all outgoing ICMP TTL expired traffic



# Access Control Lists

**ACL:** table of rules, applied top to bottom to incoming packets: (action, condition) pairs: looks like OpenFlow forwarding (Ch. 4)!

action	source address	dest address	protocol	source port	dest port	flag bit
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----
deny	all	all	all	all	all	all

# Stateful packet filtering

## ■ *stateless packet filter*:

- đôi khi cho phép các gói không có ý nghĩa ví dụ cổng 80, Ack bit, không thiết lập kết nối TCP

action	source address	dest address	protocol	source port	dest port	flag bit
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK

## ■ *stateful packet filter*: lưu vết mọi kết nối TCP

- track connection setup (SYN), teardown (FIN): gói đến, gói đi là “có ý nghĩa”
- timeout inactive connections at firewall: không cho phép gói

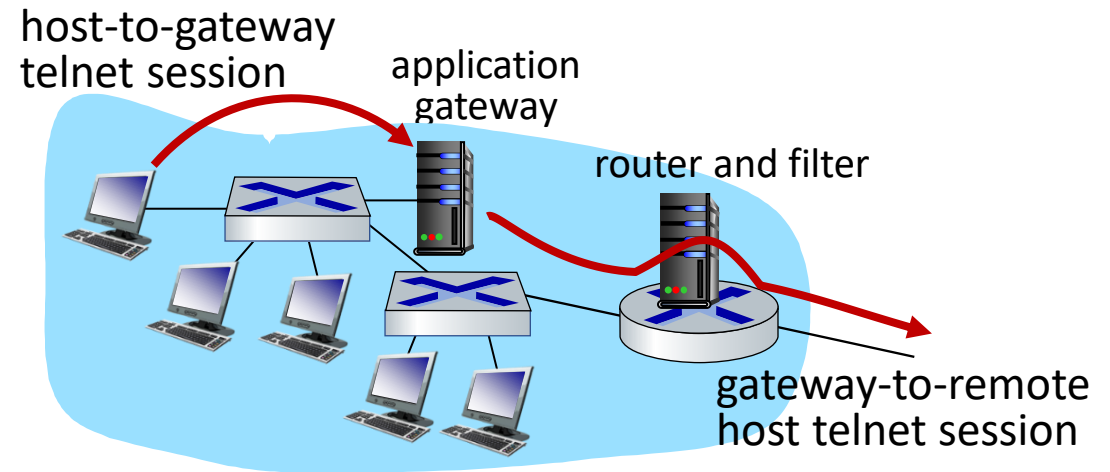
# Stateful packet filtering

ACL được tăng cường để chấp nhận các gói sau khi phải thông qua được bảng kiểm tra trạng thái kết nối

action	source address	dest address	proto	source port	dest port	flag bit	check connection
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any	
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK	X
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---	
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----	X
deny	all	all	all	all	all	all	

# Application gateways

- filter packets tại tầng ứng dụng và IP/TCP/UDP fields.
- *example: cho phép telnet từ trong ra ngoài*



# Hạn chế firewalls, gateways

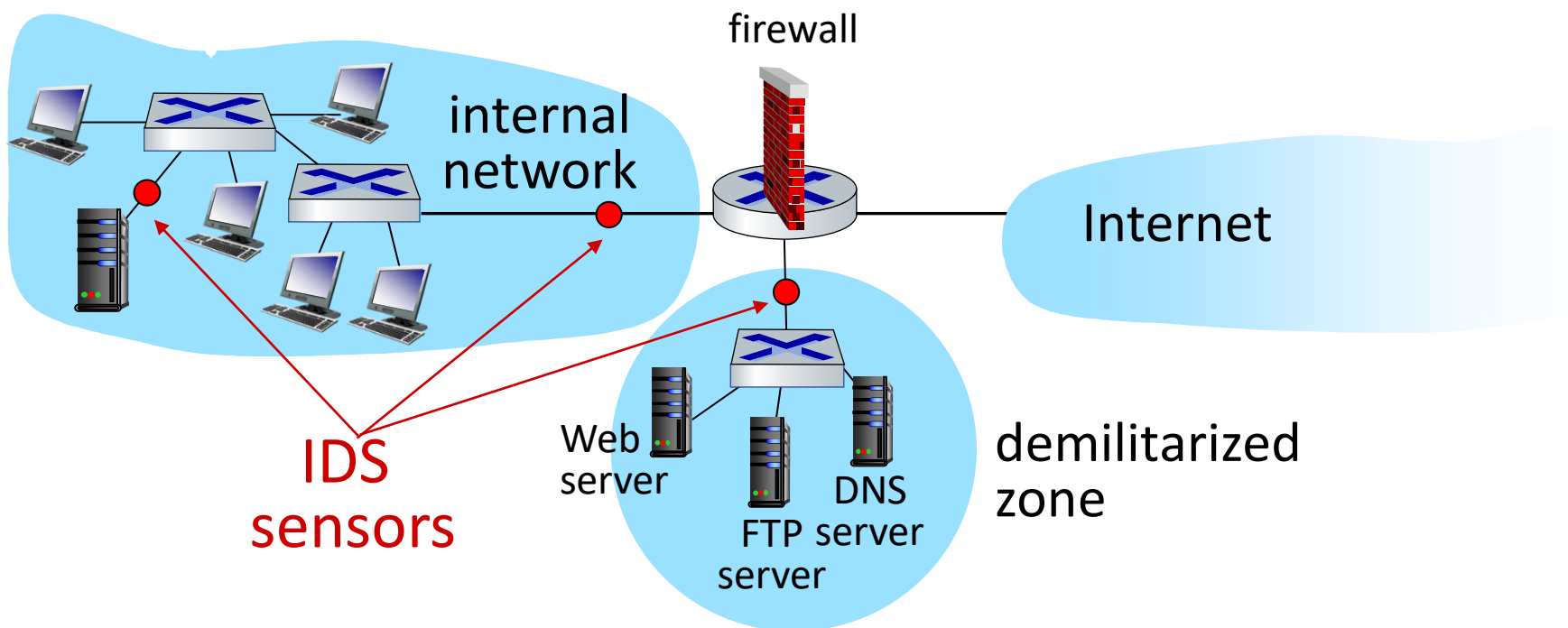
- **IP spoofing:** không phát hiện được giả mạo
- if có nhiều ứng dụng mỗi cái phải có gateway để lọc
- client software phải học cách kết nối với gateway
  - e.g., phải thiết lập IP address của proxy trong Web browser
- filters khó với UDP

# Intrusion detection systems (phát hiện xâm nhập)

- packet filtering:
  - Chỉ xử lý tại TCP/IP headers
- IDS: intrusion detection system
  - **deep packet inspection:** xem nội dung gói tin(kiểm tra các dấu hiệu của gói tin so với cơ sở dữ liệu các dấu hiệu)
  - Kiểm tra được
    - port scanning
    - network mapping
    - DoS attack

# Intrusion detection systems

multiple IDSs: nhiều kiểu triển khai



# Network Security-tóm tắt

## basic techniques.....

- cryptography (symmetric and public key)
- message integrity
- end-point authentication

## .... used in many different security scenarios

- secure email
- secure transport (TLS)
- IP sec
- 802.11, 4G/5G

## operational security: firewalls and IDS

