ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Cryptography

1

What is network security?

Confidentiality: only sender, intended receiver should "understand" message contents
- sender encrypts message
- receiver decrypts message

Authentication: sender, receiver want to confirm identity of each other

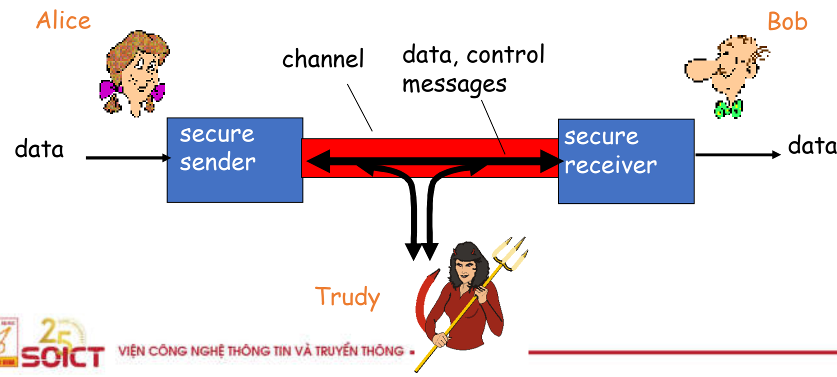
Message Integrity: sender, receiver want to ensure message not altered (in transit, or afterwards) without detection

Access and Availability: services must be accessible and available to users

2

Friends and enemies: Alice, Bob, Trudy

- well-known in network security world
- Bob, Alice (lovers!) want to communicate “securely”
- Trudy (intruder) may intercept, delete, add messages



Who might Bob, Alice be?

- … well, *real-life* Bobs and Alices!
- Web browser/server for electronic transactions (e.g., on-line purchases)
- on-line banking client/server
- DNS servers
- routers exchanging routing table updates
- other examples?

There are bad guys (and girls) out there!

Q: What can a "bad guy" do?

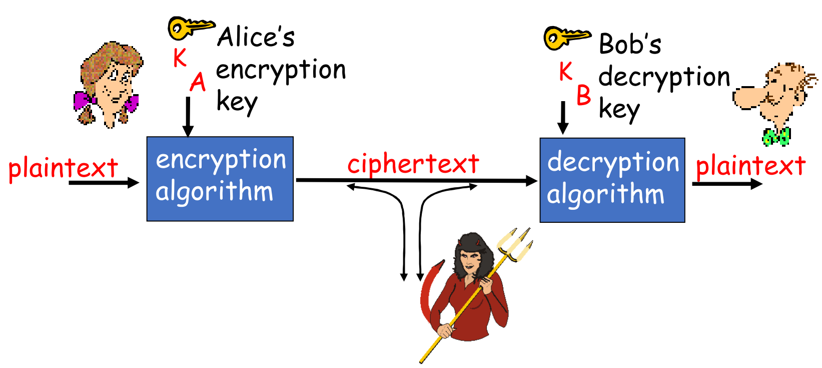
A: a lot!

- **eavesdrop**: intercept messages
- **actively insert** messages into connection
- **impersonation**: can fake (spoof) source address in packet (or any field in packet)
- **hijacking**: "take over" ongoing connection by removing sender or receiver, inserting himself in place
- **denial of service**: prevent service from being used by others (e.g., by overloading resources)

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

5

The language of cryptography



symmetric key crypto: sender, receiver keys *identical*

public-key crypto: encryption key *public*, decryption key *secret* (private)

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

6

Symmetric key cryptography

substitution cipher: substituting one thing for another

- monoalphabetic cipher: substitute one letter for another

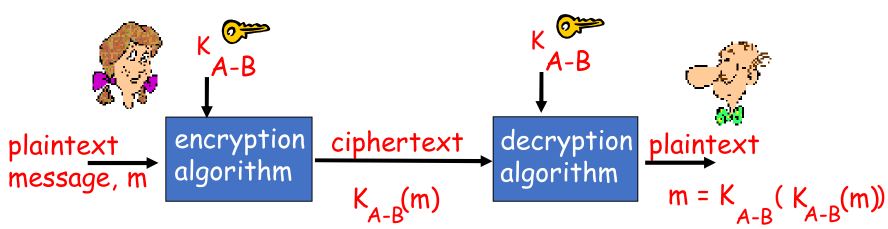
```
plaintext:   abcdefghijklmnopqrstuvwxyz

ciphertext:  mnbvcxzasdfghjklpoiuytrewq
```

E.g.: `ciphertext: nkn. s gktc wky. mgsbc`

`Plaintext: bob. i love you. alice`

7

Symmetric key cryptography



symmetric key crypto: Bob and Alice share know same (symmetric) key: K_{A-B}

- e.g., key is knowing substitution pattern in mono alphabetic substitution cipher

8

Public Key Cryptography

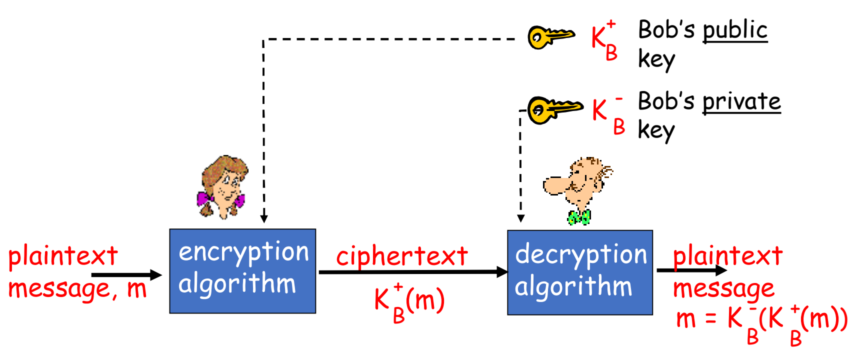
symmetric key crypto
- requires sender, receiver know shared secret key
- Q: how to agree on key in first place (particularly if never "met")?

public key cryptography
- radically different approach [Diffie-Hellman76, RSA78]
- sender, receiver do *not* share secret key
- *public* encryption key known to *all*
- *private* decryption key known only to receiver

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

9

Public key cryptography



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

10

Public key encryption algorithms

Requirements:

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

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

11

Modular Arithmetic (Z_n)

Definition: $a \equiv b \pmod n \Leftrightarrow n \mid (b - a)$

Alternatively, $a = qn + b$

Properties (equivalence relation)

- $a \equiv a \pmod n$ [Reflexive]
- $a \equiv b \pmod n \Rightarrow b \equiv a \pmod n$ [Symmetric]
- $a \equiv b \pmod n$ and $b \equiv c \pmod n \Rightarrow a \equiv c \pmod n$ [Transitive]

Definition: An equivalence class mod n

$$[a] = \{ x : x \equiv a \pmod n \} = \{ a + qn \mid q \in Z \}$$

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG 12

12

The Euclidean Algorithm

Definition. Integer division with remainder
1. $a = qb+r \quad 0 \le r < b$
2. $b|a$ if $a = qb$

Definition: Greatest Common Divisor
1. $g = \gcd(a,b)$.
2. $g|a$ and $g|b$.
3. If $e|a$ and $e|b$ then $e|g$

The Euclidean Algorithm provides and efficient method for computing gcd(a,b)
1. $\gcd(a,0) = a$
2. $\gcd(a,b) = \gcd(b, a \bmod b)$

Bezout's Identity

There exist integers x, y such that
$$ax+by = \gcd(a,b)$$

Proof.
Let $\Lambda = \{ax+by,\ x,y \in Z\}$ and let $d \in Z$ have smallest abs value.
Claim $d = \gcd(a,b)$.

Note that $a,b \in \Lambda \Rightarrow a+b$ and $s \in Z,\ a \in \Lambda \Rightarrow sa \in \Lambda$

$a = qd + r,\ 0 \le r < d$ and $r = a - qd \in \Lambda \Rightarrow r = 0$ and $d|a$. Similarly $d|b$.

If $e|a$ and $e|b$ then $e|(ax+by) \Rightarrow e|d$.

The Extended Euclidean Algorithm

Let $a_1 = a$, $a_2 = b$, $a_3, \ldots, a_{n+1}$ be a remainder sequence defined by

$a_i = q_i a_{i+1} + a_{i+2}$, $0 \le a_{i+2} < a_{i+1}$ for $i=3,\ldots,$with $a_{n+2} = 0$

Definition. Cosequences

$x_1 = 1$, $x_2 = 0$, $x_{i+2} = x_i - q_i\, x_{i+1}$

$y_1 = 0$, $y_2 = 1$, $y_{i+2} = y_i - q_i\, y_{i+1}$

Then $ax_i + by_i = a_i$ and in particular $\mathbf{ax_{n+1} + by_{n+1} = a_{n+1} = gcd(a,b)}$

15

Extended Euclidean Algorithm - example

- **8 mod 11**

Step 0:	11= 8(1) + 3	3 = 11 − 8(1)
Step 1:	8 = 3(2) + 2	2 = 8 − 3(2)
Step 2:	2(1) + 1	1 = 3 − 2(1)
Step 3:	2 = 1(2)	

$x = -4$, $y = 3$

$1 \equiv 8(-4) \bmod 11$

$1 \equiv 8(7) \bmod 11$

$$1 = 3 - 2(1)$$
$$1 = 3 - (8 - 3(2))(1) = 3 - (8 - (3(2)) = 3(3) - 8$$
$$1 = (11 - 8(1))(3) - 8 = 11(3) - 8(4) = 11(3) + 8(-4)$$

16

Extended Euclidean Algorithm - practice

- *a = 1914, b = 899*
- *a = 102, b = 38*
- *a = 81, b = 57*
- *a =* 42823*, b = 6409*

x = 8, y = -17

x = 3, y = -8

x = 10, y = -7

x = -22, y = 147

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

17

Extended Euclidean Algorithm - example

- $240x \equiv 1 \pmod{17}$

$1 = 240x + 17y$

$240 = 17 \cdot 14 + 2$ (a)
$17 = 2 \cdot 8 + 1$ (b)

x = -8, y = 133

x = 9 as $-8 \equiv 9 \pmod{17}$

(a) → $2 = 240 - 17 \cdot 14$
(b) → $1 = 17 - 2 \cdot 8$
→ $1 = 17 - (240 - 17 \cdot 14) \cdot 8 = 17 \cdot (1 + 132) - 240 \cdot 8 = \mathbf{240 \cdot (-8)} + 17 \cdot 133$

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

18

Fermat's Theorem

If $a \neq 0 \in Z_p$, then $a^{p-1} \equiv 1 \pmod p$

More generally, if $a \in Z_p$, then $a^p \equiv a \pmod p$

Proof: Assume that $a \neq 0 \in Z_p$. Then
$a * 2a * \ldots (p-1)a = (p-1)! * a^{p-1}$

Also, since $a*i \equiv a*j \pmod p \Rightarrow i \equiv j \pmod p$, the numbers a, 2a, …, (p-1)a are distinct elements of Z_p. Therefore, they are equal to 1,2,…,(p-1) and their product is equal to (p-1)! mod p. This implies that

$$(p-1)! * a^{p-1} \equiv (p-1)! \pmod p \Rightarrow a^{p-1} \equiv 1 \pmod p.$$

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG 19

19

Fermat's Theorem - example

Calculate 4^73 mod 11

- 4^10 = 1 mod 11 1048576 = 1 + 95325*11
- 73 = 7*10 + 3
- 4^73 = 4^(7*10+3)
- = 4^3 * (4^10)^7
- = 64 * 1 (mod 11)
- = 9 mod 11

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

20

Euler phi function

- Definition: phi(n) = #{a: 0 < a < n and gcd(a,n) = 1}
- Properties:
  - $\varphi(p) = p-1$, for prime p.
  - $\varphi(p^e) = (p-1)*p^{(e-1)}$
  - $\varphi(m*n) = \varphi(m)*\varphi(n)$ for gcd(m,n) = 1.
  - $\varphi(p*q) = (p-1)*(q-1)$

- Examples:

  - $\varphi(15) = \varphi(3)*\varphi(5) = 2*4 = 8.$ = #{1,2,4,7,8,11,13,14}
  - $\varphi(9) = (3-1)*3^{(2-1)} = 2*3 = 6$ = #{1,2,4,5,7,8}

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG 21

21

Euler's Identity

- The number of elements in Z_n that have multiplicative inverses is equal to phi(n).

- Theorem: Let $(Z_n)^*$ be the elements of Z_n with inverses (called units). If $a \in (Z_n)^*$, then

$$\mathbf{a^{\varphi(n)} \equiv 1 \pmod{n}}$$

Proof. The same proof presented for Fermat's theorem can be used to prove this theorem.

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG 22

22

RSA: another important property

The following property will be *very* useful later:

$$\underbrace{K_B^-(K_B^+(m))}_{\text{use public key first, followed by private key}} = m = \underbrace{K_B^+(K_B^-(m))}_{\text{use private key first, followed by public key}}$$

Result is the same!

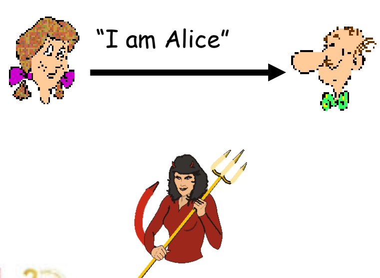
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

23

Authentication

Goal: Bob wants Alice to “prove” her identity to him

Protocol ap1.0: Alice says “I am Alice”



Failure scenario??

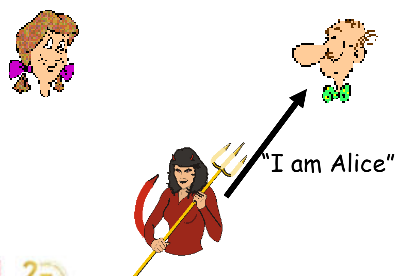
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

24

Authentication

Goal: Bob wants Alice to “prove” her identity to him

Protocol ap1.0: Alice says “I am Alice”



in a network,
Bob can not “see” Alice, so
Trudy simply declares
herself to be Alice

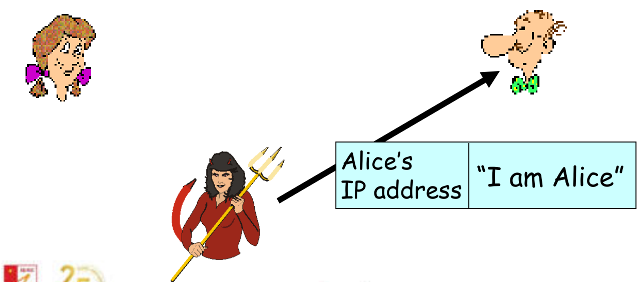
Authentication: another try

Protocol ap2.0: Alice says “I am Alice” in an IP packet containing her source IP address

Failure scenario??

Authentication: another try

Protocol ap2.0: Alice says "I am Alice" in an IP packet containing her source IP address

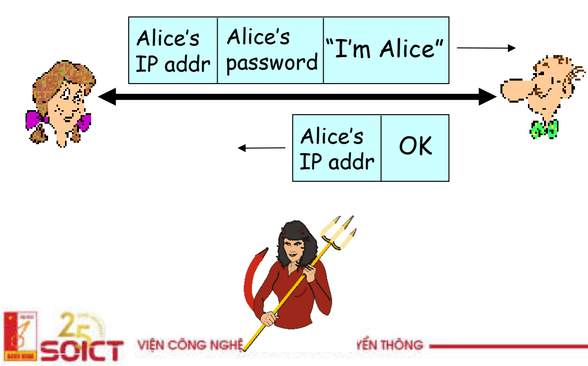


Trudy can create a packet "spoofing" Alice's address

27

Authentication: another try

Protocol ap3.0: Alice says "I am Alice" and sends her secret password to "prove" it.

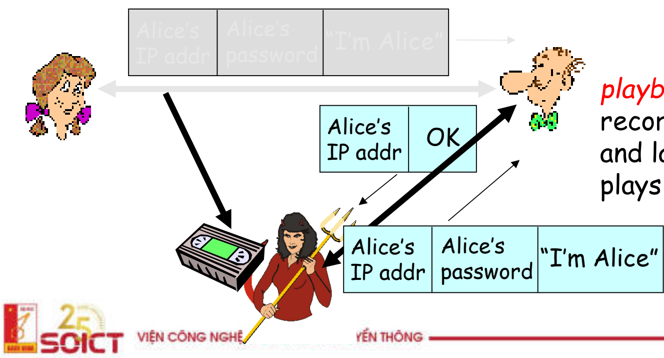


Failure scenario??

28

Authentication: another try

Protocol ap3.0: Alice says "I am Alice" and sends her secret password to "prove" it.



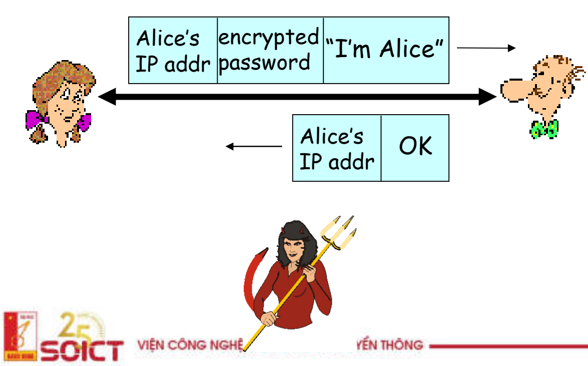
playback attack: Trudy records Alice's packet and later plays it back to Bob

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

29

Authentication: yet another try

Protocol ap3.1: Alice says "I am Alice" and sends her *encrypted* secret password to "prove" it.



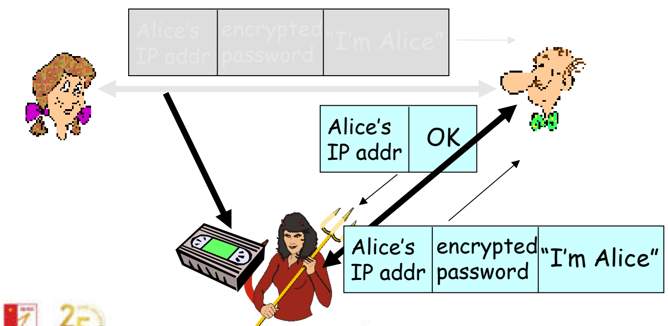
Failure scenario??

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

30

Authentication: another try

Protocol ap3.1: Alice says "I am Alice" and sends her *encrypted* secret password to "prove" it.



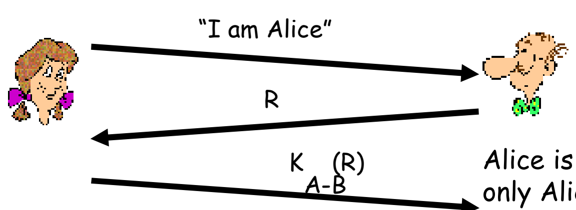
record and playback still works!

Authentication: yet another try

Goal: avoid **playback** attack

Nonce: number (R) used only *once –in-a-lifetime*

ap4.0: to prove Alice "live", Bob sends Alice **nonce**, R. Alice must return R, encrypted with shared secret key



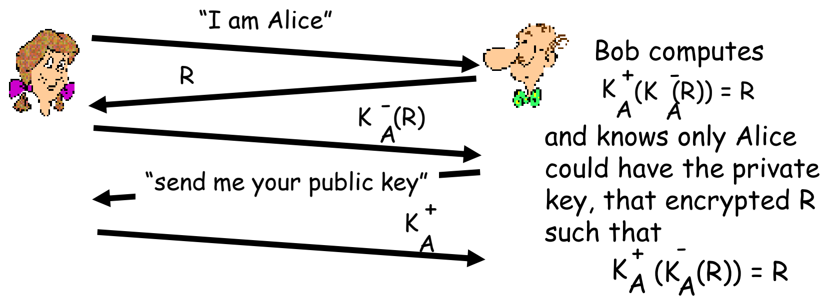
Alice is live, and only Alice knows key to encrypt nonce, so it must be Alice!

Authentication: ap5.0

ap4.0 requires shared symmetric key

- can we authenticate using public key techniques?

ap5.0: use nonce, public key cryptography

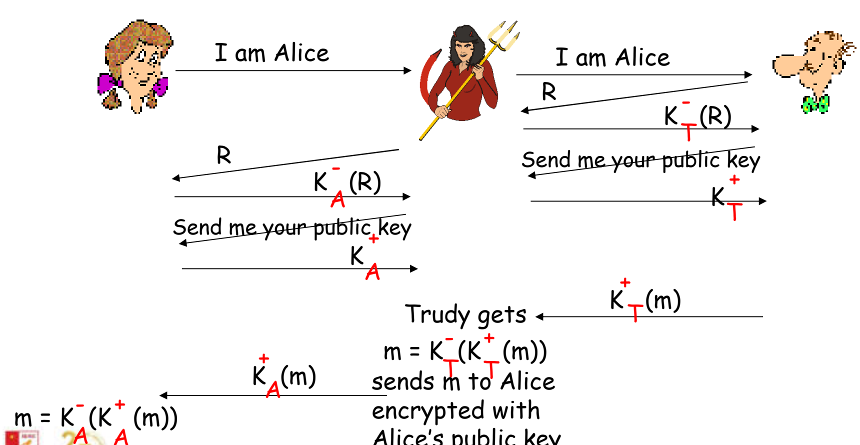


VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

33

ap5.0: security hole

Man (woman) in the middle attack: Trudy poses as Alice (to Bob) and as Bob (to Alice)

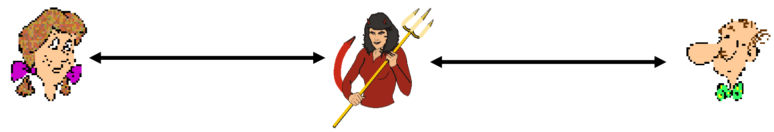


VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

34

ap5.0: security hole

Man (woman) in the middle attack: Trudy poses as Alice (to Bob) and as Bob (to Alice)



Difficult to detect:
- Bob receives everything that Alice sends, and vice versa. (e.g., so Bob, Alice can meet one week later and recall conversation)
- problem is that Trudy receives all messages as well!

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

35

Digital Signatures

Cryptographic technique analogous to hand-written signatures.

- sender (Bob) digitally signs document, establishing he is document owner/creator.
- verifiable, nonforgeable: recipient (Alice) can prove to someone that Bob, and no one else (including Alice), must have signed document

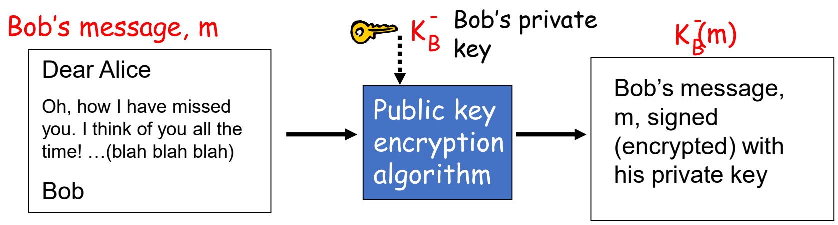
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

36

Digital Signatures

Simple digital signature for message m:

- Bob signs m by encrypting with his private key K_B, creating "signed" message, $K_B^-(m)$



37

Digital Signatures (more)

- Suppose Alice receives msg m, digital signature $K_B^-(m)$
- Alice verifies m signed by Bob by applying Bob's public key K_B^+ to $K_B^-(m)$ then checks $K_B^+(K_B^-(m)) = m$.
- If $K_B^+(K_B^-(m)) = m$, whoever signed m must have used Bob's private key.

Alice thus verifies that:

- ✓ Bob signed m.
- ✓ No one else signed m.
- ✓ Bob signed m and not m'.

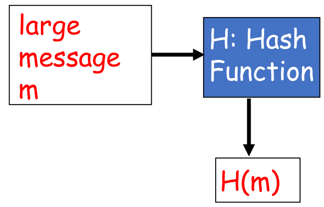
Non-repudiation:

- ✓ Alice can take m, and signature $K_B^-(m)$ to court and prove that Bob signed m.

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

38

Message Digests



Computationally expensive to public-key-encrypt long messages

Goal: fixed-length, easy-to-compute digital "fingerprint"

- apply hash function H to *m*, get fixed size message digest, *H(m)*.

Hash function properties:

- produces fixed-size msg digest (fingerprint)
- given message digest x, computationally infeasible to find m such that x = H(m)

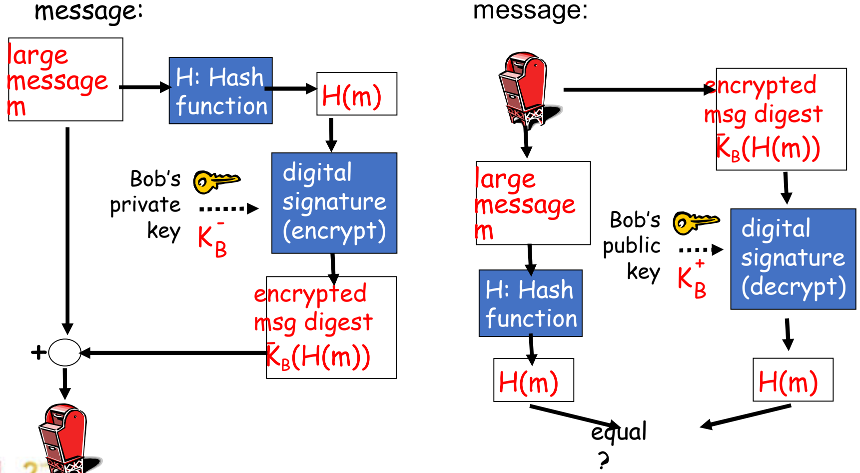
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

39

Digital signature = signed message digest

Bob sends digitally signed message:

Alice verifies signature and integrity of digitally signed message:



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

40

Hash Function Algorithms

- MD5 hash function widely used (RFC 1321)
  - computes 128-bit message digest in 4-step process.
  - arbitrary 128-bit string x, appears difficult to construct msg m whose MD5 hash is equal to x.
- SHA-1 is also used.
  - US standard [NIST, FIPS PUB 180-1]
  - 160-bit message digest

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

41

Trusted Intermediaries

Symmetric key problem:

- How do two entities establish shared secret key over network?

Solution:

- trusted key distribution center (KDC) acting as intermediary between entities

Public key problem:

- When Alice obtains Bob's public key (from web site, e-mail, diskette), how does she know it is Bob's public key, not Trudy's?

Solution:

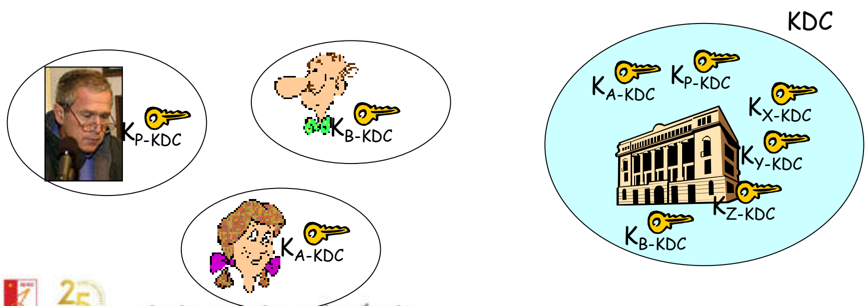
- trusted certification authority (CA)

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

42

Key Distribution Center (KDC)

- Alice, Bob need shared symmetric key.
- KDC: server shares different secret key with *each* registered user (many users)
- Alice, Bob know own symmetric keys, $K_{A\text{-}KDC}$ $K_{B\text{-}KDC}$, for communicating with KDC.

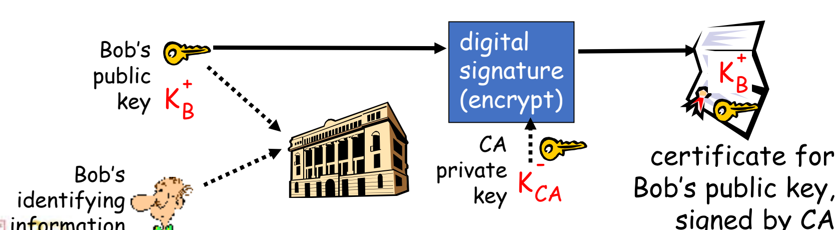


SOICT VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

43

Certification Authorities

- Certification authority (CA): binds public key to particular entity, E.
- E (person, router) registers its public key with CA.
  - E provides "proof of identity" to CA.
  - CA creates certificate binding E to its public key.
  - certificate containing E's public key digitally signed by CA – CA says "this is E's public key"

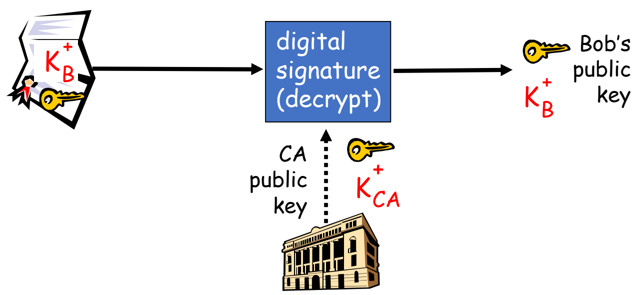


SOICT VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

44

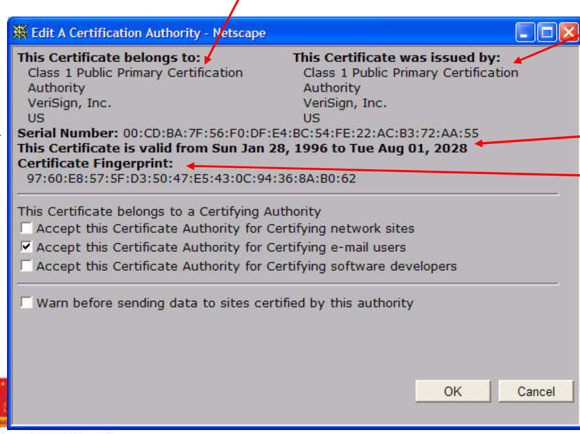
Certification Authorities

- When Alice wants Bob's public key:
  - gets Bob's certificate (Bob or elsewhere).
  - apply CA's public key to Bob's certificate, get Bob's public key



A certificate contains:

- Serial number (unique to issuer)
- info about certificate owner, including algorithm and key value itself (not shown)



- info about certificate issuer
- valid dates
- digital signature by issuer

Secure sockets layer (SSL)

- transport layer security to any TCP-based app using SSL services.
- used between Web browsers, servers for e-commerce (shttp).
- security services:
  - server authentication
  - data encryption
  - client authentication (optional)
- server authentication:
  - SSL-enabled browser includes public keys for trusted CAs.
  - Browser requests server certificate, issued by trusted CA.
  - Browser uses CA's public key to extract server's public key from certificate.
- check your browser's security menu to see its trusted CAs.

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

47

SSL (continued)

Client

Server

Open secure socket

Certificate (CA signed Pub Key)

- Verify CA trusted
- Extract Server Pub Key
- Generate symmetric Session Key
- Encrypt Session Key with Server Pub Key

Encrypted Session Key

- Extract Session Key (using Private Key)

In Java all of this happens behind the scenes!

```
SSLSocket s = (SSLSocket)sslFact.createSocket(host, port);
```

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

48

SSL Observations

- Previous example does not
  - Show how public/private key pairs are generated
    - Manually
  - Enable the Server to authenticate the client
    - Client can use trusted certificate, or another scheme such as passwords
  - Show how the Server receives a signed certificate
    - CA!

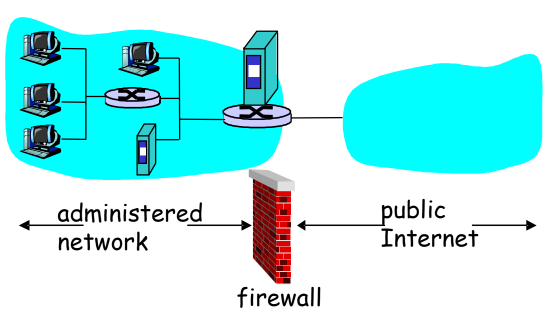
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

49

Firewalls

firewall

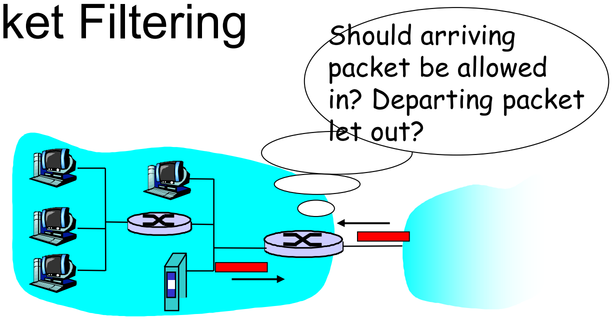
isolates organization's internal net from larger Internet, allowing some packets to pass, blocking others.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

50

Packet Filtering



- internal network connected to Internet via router firewall
- router filters packet-by-packet, decision to forward/drop packet based on:
  - source IP address, destination IP address
  - TCP/UDP source and destination port numbers
  - ICMP message type
  - TCP SYN and ACK bits

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

51

Packet Filtering

- Example 1: block incoming and outgoing datagrams with **IP protocol field = 17** and with either **source** or **dest port = 23**.

  All incoming and outgoing UDP flows and telnet connections are blocked.

- Example 2: Block inbound TCP segments with ACK=0.

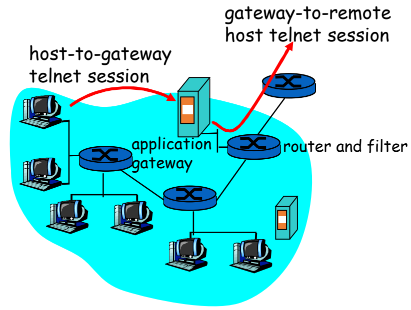
  Prevents **external** clients from making TCP connections with **internal** clients, but **allows** **internal** clients to connect to outside.

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

52

Application gateways

- Filters packets on application data as well as on IP/TCP/UDP fields.
- Example: allow select internal users to telnet outside.



1. Require all telnet users to telnet through gateway.
2. For authorized users, gateway sets up telnet connection to dest host. Gateway relays data between 2 connections
3. Router filter blocks all telnet connections not originating from gateway.

Internet security threats

Mapping:

- before attacking: "case the joint" – find out what services are implemented on network
- Use `ping` to determine what hosts have addresses on network
- Port-scanning: try to establish TCP connection to each port in sequence (see what happens)
- nmap (http://www.insecure.org/nmap/) mapper: "network exploration and security auditing"

Countermeasures?

Internet security threats

Mapping: countermeasures

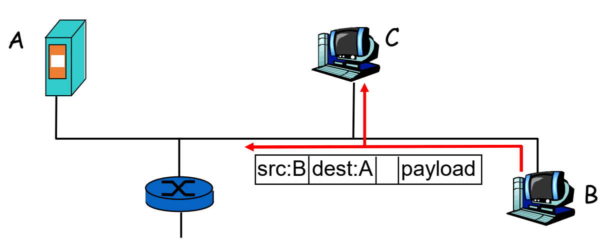
- record traffic entering network
- look for suspicious activity (IP addresses, pots being scanned sequentially)

55

Internet security threats

Packet sniffing:

- broadcast media
- promiscuous NIC reads all packets passing by
- can read all unencrypted data (e.g. passwords)
- e.g.: C sniffs B's packets



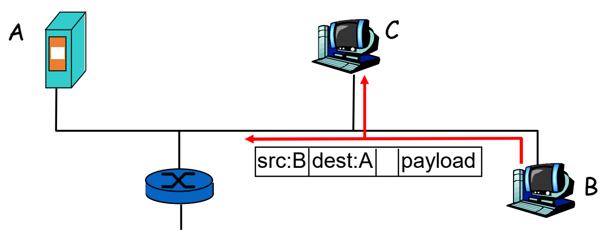
Countermeasures?

56

Internet security threats

Packet sniffing: countermeasures

- all hosts in organization run software that checks periodically if host interface in promiscuous mode.
- one host per segment of broadcast media (switched Ethernet at hub)



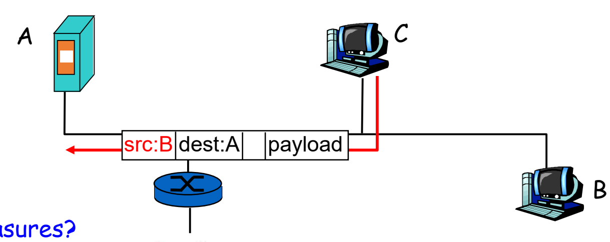
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

57

Internet security threats

IP Spoofing:

- can generate "raw" IP packets directly from application, putting any value into IP source address field
- receiver can't tell if source is spoofed
- e.g.: C pretends to be B



Countermeasures?

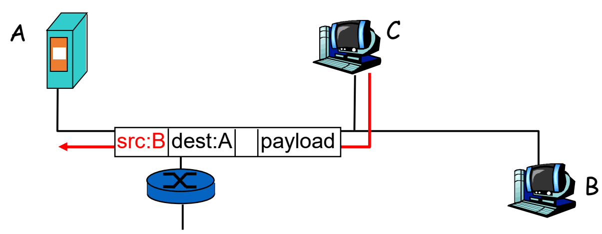
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

58

Internet security threats

IP Spoofing: ingress filtering

- routers should not forward outgoing packets with invalid source addresses (e.g., datagram source address not in router's network)
- great, but ingress filtering can not be mandated for all networks



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

59

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Q&A

soict.hust.edu.vn/ fb.com/groups/soict

60