

**Telematics Course :**  
**Computer Networks and the Internet**  
(Winter Semester 2002/03)

Prof. Dr. Dieter Hogrefe  
Dipl.-Inform. Michael Ebner  
Dr. Xiaoming Fu

Telematics group  
University of Göttingen, Germany

**Computer and Network Security**

Handout 9, Computer Networks and the  
Internet, Telematics Course

Credits:

- James Kurose, Keith Ross, Computer Networking(2nd Ed.), Addison-Wesley, 2002
- Henning Schulzrinne, Columbia University

Further reading:

- William Stallings, Cryptography and Network Security: Principles and Practice (3rd Ed.), Prentice-Hall, 2003

2

**Outline**

- **General**
  - What to Protect
  - Where to Put the Protection
  - Host and Network Based Security
- **Cryptography**
  - Symmetric Crypto Systems, DES
  - Asymmetric Crypto Systems, RSA
- **Authentication**
- **Integrity**
  - Digital Signature
- **Key Distribution, Certificates**
- **Access Control: firewalls**
- **Email Security, PGP**
- **Security of the Web and Transport Layer (SSL/TLS)**
- **IPSec**

3

**Security Concerns**

**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

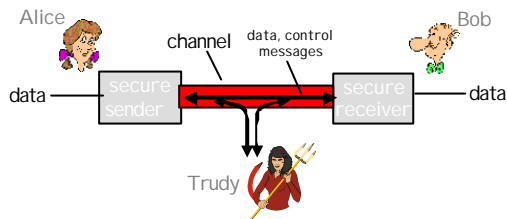
**Message Integrity:** sender, receiver want to ensure  
message not altered (in transit, or afterwards) without  
detection

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

4

### 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



5

### 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?

6

### There are bad guys out there!

**Q:** What can a “bad guy” do?

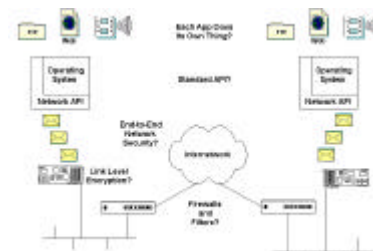
**A:** a lot!

- **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)

more on this later .....

7

### Where to Put the Protection?



8

## Host/OS Based Security

- Key idea: protect the *DATA*
  - End hosts are in control of data
  - Users are in control of end hosts
  - Users can and often will do dumb things, + hijackers!
  - Result: very difficult to protect all hosts
- Approaches:
  - OS software integrity (most attacks on non-patched OS)
  - user-level access control (AAA, tokens)
  - block unneeded services (finger, ftp, DNS)
  - path encryption via IPsec
  - device-level access control (MAC, IP, DNS) in servers, routers, Ethernet switches
  - e.g., host firewalling (such as TCP wrappers, IP chains)

9

## Network Based Security

- Should augment host based security
- Useful for
  - Protecting groups of users from others
  - Prohibiting certain types of network usage
  - Controlling traffic flow
- Difficult to inspect traffic
  - encryption can hide bad things
  - Tunneling (eg., NAT) can mislead you

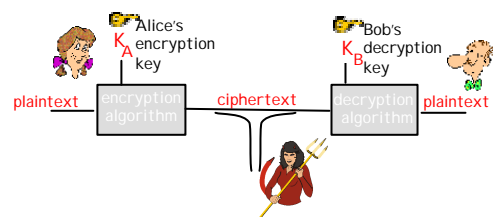
10

## Outline

- General
  - What to Protect
  - Where to Put the Protection
  - Host and Network Based Security
- Principles of Cryptography
  - Symmetric Crypto Systems, DES
  - Asymmetric Crypto Systems, RSA
- Authentication
  - Digital Signature
- Integrity
- Key Distribution, Certificates
- Access Control: firewalls
- Email Security, PGP
- Security of the Web and Transport Layer (SSL)
- IPsec

11

## The language of cryptography



**symmetric key** crypto: sender, receiver keys *identical*  
**public-key** crypto: encryption key *public*, decryption key *secret* (private)

12

### Symmetric key cryptography

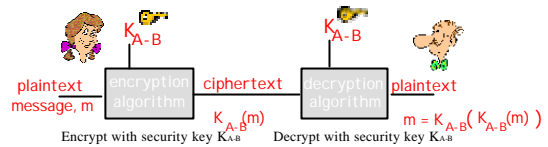
**substitution cipher:** substituting one thing for another  
 - monoalphabetic cipher: substitute one letter for another

plaintext: abcdefghijklmnopqrstuvwxyz  
 ciphertext: mnbcvxzasdfghjklpoiuytrewq

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

13

### Symmetric key cryptography



**symmetric key crypto:** Bob and Alice share know same (symmetric) key:  $K$

- e.g., key is knowing substitution pattern in mono alphabetic substitution cipher
- Q:** how do Bob and Alice agree on key value?

14

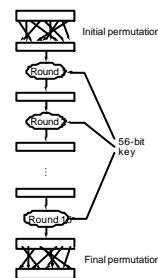
### Symmetric key crypto: DES

#### DES: Data Encryption Standard

- US encryption standard [NIST 1993]
- 56-bit symmetric key, 64-bit plaintext input
- How secure is DES?
  - DES Challenge: 56-bit-key-encrypted phrase ("Strong cryptography makes the world a safer place") decrypted (brute force) in 4 months
  - no known "backdoor" decryption approach
- making DES more secure:
  - use three keys sequentially (3-DES) on each datum
  - use cipher-block chaining

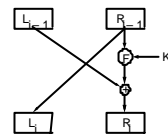
15

- 64-bit key (56-bits + 8-bit parity)
- 16 rounds



**DES operation**  
 initial permutation  
 16 identical "rounds" of function application, each using different 48 bits of key  
 final permutation

- Each Round



16

## 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")?

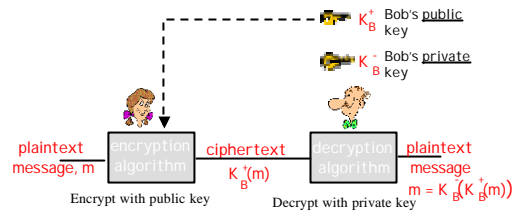
### public key cryptography

- 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



17

## Public key cryptography



18

## Public key encryption algorithms

Requirements:

- need  $K_B^+(\cdot)$  and  $K_B^-(\cdot)$  such that  $K_B^-(K_B^+(m)) = m$
- given public key  $K_B^+$  it should be impossible to compute private key  $K_B^-$

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

19

## RSA: Choosing keys

- Choose two large prime numbers  $p, q$ . (e.g., 1024 bits each)
- Compute  $n = pq$ ,  $z = (p-1)(q-1)$
- Choose  $e$  (with  $e < n$ ) that has no common factors with  $z$ . ( $e, z$  are "relatively prime").
- Choose  $d$  such that  $ed-1$  is exactly divisible by  $z$  (in other words:  $ed \bmod z = 1$ ).
- Public key is  $(n, e)$ . Private key is  $(n, d)$ .

$K_B^+$        $K_B^-$

20

## RSA: Encryption, decryption

0. Given  $(n, e)$  and  $(n, d)$  as computed above
1. To encrypt bit pattern,  $m$ , compute  
 $c = m^e \bmod n$  (i.e., remainder when  $m^e$  is divided by  $n$ )
2. To decrypt received bit pattern,  $c$ , compute  
 $m = c^d \bmod n$  (i.e., remainder when  $c^d$  is divided by  $n$ )

Magic happens!

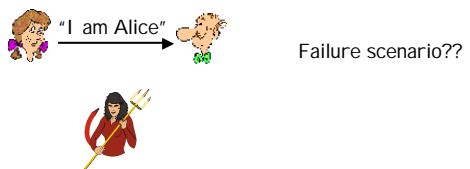
$$m = (\underbrace{m^e \bmod n}_c)^d \bmod n$$

## Outline

- General
  - What to Protect
  - Where to Put the Protection
  - Host and Network Based Security
- Cryptography
  - Symmetric Crypto Systems, DES
  - Asymmetric Crypto Systems, RSA
- Authentication
- Integrity
  - Digital Signature
- Key Distribution, Certificates
- Access Control: firewalls
- Email Security, PGP
- Security of the Web and Transport Layer (SSL)
- IPSec

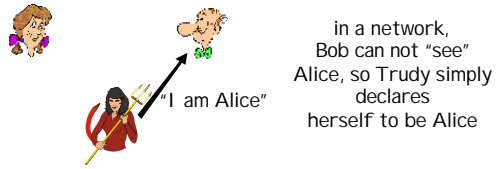
## Authentication

**Goal:** Bob wants Alice to "prove" her identity to him  
**Protocol ap1.0:** Alice says "I am Alice"



## Authentication

**Goal:** Bob wants Alice to "prove" her identity to him  
**Protocol ap1.0:** Alice says "I am Alice"



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

Alice's IP address | "I am Alice"

Failure scenario??

25

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

Alice's IP address | "I am Alice"

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

26

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

Alice's IP addr | Alice's password | "I'm Alice"

Alice's IP addr | OK

Failure scenario??

27

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

Alice's IP addr | Alice's password | "I'm Alice"

Alice's IP addr | OK

Alice's IP addr | Alice's password | "I'm Alice"

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

28

### Authentication: yet another try

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

Failure scenario??

29

### 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!**

30

### 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

Failures, drawbacks?

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

31

### Authentication: ap5.0

ap4.0 requires shared symmetric key

- can we authenticate using public key techniques?

**ap5.0:** use nonce, public key cryptography

Bob computes  $K_A^+(K_A^-(R)) = R$  and knows only Alice could have the private key, that encrypted R such that  $K_A^+(K_A^-(R)) = R$

32

### ap5.0: security hole

**Man-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!

33

### Outline

- General
  - What to Protect
  - Where to Put the Protection
  - Host and Network Based Security
- Cryptography
  - Symmetric Crypto Systems, DES
  - Asymmetric Crypto Systems, RSA
- Authentication
  - **Digital Signature**
- Key Distribution, Certificates
- Access Control: firewalls
- Email Security, PGP
- Security of the Web and Transport Layer (SSL)
- IPSec

34

### Message Integrity Approaches

- Digital signature using RSA
  - special case of a message integrity where the code can only have been generated by one participant
  - compute signature with private key and verify with public key
- Keyed MD5
  - sender:  $m + \text{MD5}(m + k) + E(k, \text{private})$
  - receiver
    - recovers random key using the sender's public key
    - applies MD5 to the concatenation of this random key message
- MD5 with RSA signature
  - sender:  $m + E(\text{MD5}(m), \text{private})$
  - receiver
    - decrypts signature with sender's public key
    - compares result with MD5 checksum sent with message

35

### Digital Signatures

**Cryptographic technique analogous to hand-written signatures.**

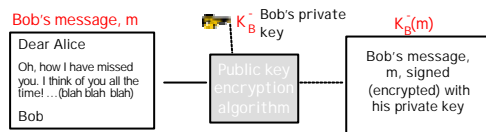
- sender (Bob) digitally signs document, establishing he is document owner/creator.
- **verifiable, nonforgeable:** recipient (Alice) can prove to someone that Bob, and no one else (including Alice), must have signed document

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  $m$ .

38

## Outline

- General
  - What to Protect
  - Where to Put the Protection
  - Host and Network Based Security
- Cryptography
  - Symmetric Crypto Systems, DES
  - Asymmetric Crypto Systems, RSA
- Authentication
- Integrity
  - Digital Signature
- Key Distribution, Certificates
- Access Control: firewalls
- Email Security, PGP
- Security of the Web and Transport Layer (SSL/TLS)
- IPSec

39

## 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?

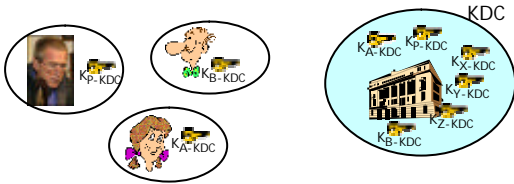
### Solution:

- trusted certification authority (CA)

40

### Key Distribution Center (KDC)

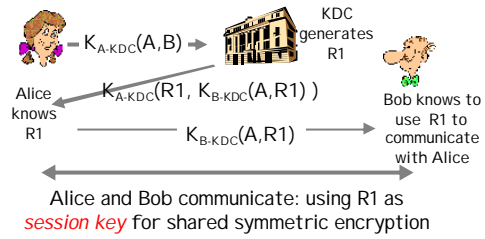
- Alice, Bob need shared symmetric key.
- **KDC**: server shares different secret key with *each* registered user (many users)
- Alice, Bob know own symmetric keys,  $K_{A-KDC}$   $K_{B-KDC}$ , for communicating with KDC.



41

### Key Distribution Center (KDC)

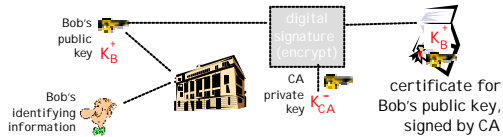
**Q:** How does KDC allow Bob, Alice to determine shared symmetric secret key to communicate with each other?



42

### Certification Authorities

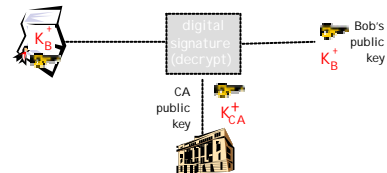
- **Certification authority (CA)**: binds public key to particular entity, E.
- E (person, router) registers its public key with CA.
  - E provides "proof of identity" to CA.
  - CA creates certificate binding E to its public key.
  - certificate containing E's public key digitally signed by CA – CA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



43

### 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



44

### 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

45

### Outline

- General
  - What to Protect
  - Where to Put the Protection
  - Host and Network Based Security
- Cryptography
  - Symmetric Crypto Systems, DES
  - Asymmetric Crypto Systems, RSA
- Authentication
- Integrity
  - Digital Signature
- Key Distribution, Certificates
- Access Control: firewalls
- Email Security, PGP
- Security of the Web and Transport Layer (SSL/TLS)
- IPSec

46

### Firewalls

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

47

### Firewalls: Why

**prevent denial of service attacks:**

- SYN flooding: attacker establishes many bogus TCP connections, no resources left for "real" connections.

**prevent illegal modification/access of internal data.**

- e.g., attacker replaces CIA's homepage with something else

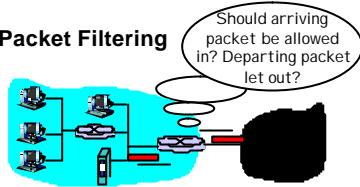
**allow only authorized access to inside network (set of authenticated users/hosts)**

**two types of firewalls:**

- application-level
- packet-filtering

48

## 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

49

## 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.

50

## Example Firewall: ipchains

```
-A input -s 192.168.0.0/255.255.0.0 -d 0.0.0.0/0.0.0.0 -j DENY
-A input -s 172.0.0.0/255.240.0.0 -d 0.0.0.0/0.0.0.0 -j DENY
-A input -s 10.0.0.0/255.0.0.0 -d 0.0.0.0/0.0.0.0 -j DENY
-A input -s 224.0.0.0/224.0.0.0 -d 0.0.0.0/0.0.0.0 -j DENY
-A input -s 0.0.0.0/0.0.0.0 -d a.b.c.d/255.255.255.255 22:22 -p 6 -j ACCEPT
-A input -s 0.0.0.0/0.0.0.0 -d a.b.c.d/255.255.255.255 1024:65535 -p 6 ! -y
-j ACCEPT
```

51

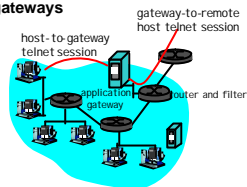
## Example Firewall: Cisco Router Filters

```
access-list 100 deny ip 192.168.0.0 0.0.255.255 any
access-list 100 deny ip 172.0.0.0 0.15.255.255 any
access-list 100 deny ip 10.0.0.0 0.255.255.255 any
access-list 100 deny ip 0.0.0.0 0.255.255.255 any
access-list 100 deny ip 127.0.0.0 0.255.255.255 any
access-list 100 deny ip 224.0.0.0 31.255.255.255 any
access-list 100 deny ip 1.2.0.0 0.0.255.255 any
access-list 100 permit tcp any host 1.2.3.4 eq domain
access-list 100 permit udp any host 1.2.3.4 eq domain
access-list 100 deny tcp any host 1.2.3.5 eq telnet log
access-list 100 deny tcp any host 1.2.3.6 eq syn log
access-list 100 deny ip any host 1.2.3.4
access-list 100 permit ip any 1.2.0.0 0.0.255.255
access-list 100 deny ip any any
```

52

### Application gateways

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



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

53

### Limitations of firewalls and gateways

- IP spoofing:** router can't know if data "really" comes from claimed source
- if multiple app's. need special treatment, each has own app. gateway.
- client software must know how to contact gateway.
  - e.g., must set IP address of proxy in Web browser
- filters often use all or nothing policy for UDP.
- tradeoff: **degree of communication with outside world, level of security**
- many highly protected sites still suffer from attacks.

54

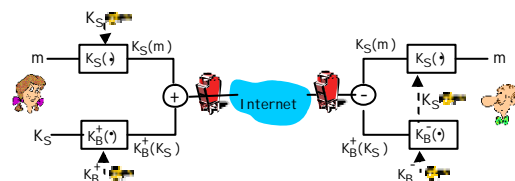
### Outline

- General
  - What to Protect
  - Where to Put the Protection
  - Host and Network Based Security
- Cryptography
  - Symmetric Crypto Systems, DES
  - Asymmetric Crypto Systems, RSA
- Authentication
- Integrity
  - Digital Signature
- Key Distribution, Certificates
- Access Control: firewalls
- Email Security PGP**
- Security of the Web and Transport Layer (SSL)
- IPSec

55

### Secure e-mail

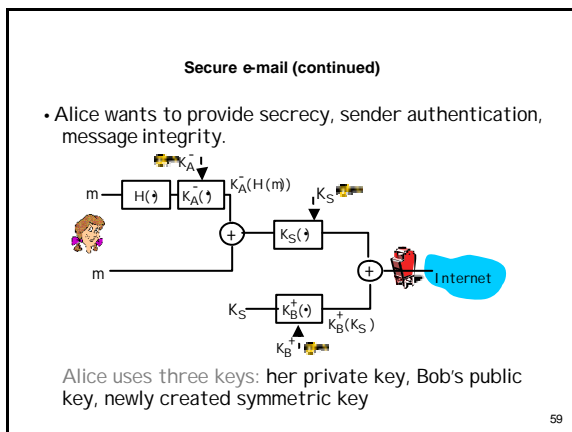
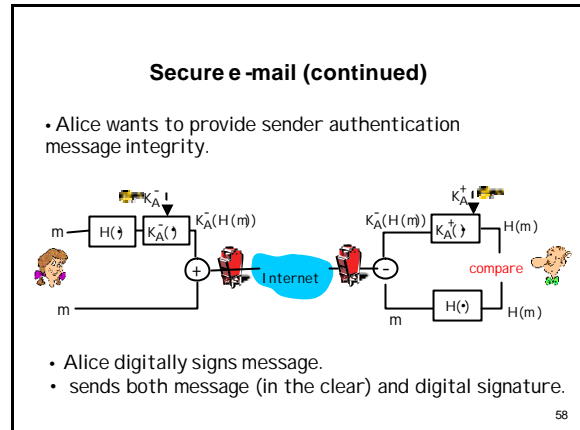
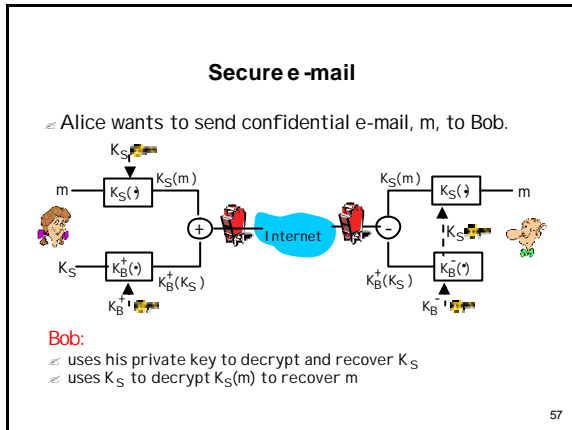
≠ Alice wants to send confidential e-mail,  $m$ , to Bob.



**Alice:**

- ≠ generates random *symmetric* private key,  $K_S$ .
- ≠ encrypts message with  $K_S$  (for efficiency)
- ≠ also encrypts  $K_S$  with Bob's public key.
- ≠ sends both  $K_S(m)$  and  $K_B^+(K_S)$  to Bob.

56



### Pretty good privacy (PGP)

- Internet e-mail encryption scheme, de-facto standard.
- uses symmetric key cryptography, public key cryptography, hash function, and digital signature as described.
- provides secrecy, sender authentication, integrity.
- inventor, Phil Zimmerman, was target of 3-year federal investigation.

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+1o8gE4vB3mqJ
hFEvZP9t6n7G6m5Gw2
---END PGP SIGNATURE---
```

60

## Outline

- General
  - What to Protect
  - Where to Put the Protection
  - Host and Network Based Security
- Cryptography
  - Symmetric Crypto Systems, DES
  - Asymmetric Crypto Systems, RSA
- Authentication
- Integrity
  - Digital Signature
- Key Distribution, Certificates
- Access Control: firewalls
- Email Security, PGP
- Security of the Web and Transport Layer (SSL)
- IPSec

61

## Web Security

- authentication: basic, digest
- often supplemented by *cookies*
- access control via network addresses
- multi-layered:
  - SHTTP (secure HTTP) = just for HTTP (shttp://)  
CommerceNet, Mosaic
  - SSL (? TLS) = generic for TCP (https://)  
implementation: SSLeay
  - IP security: host-to-host

62

## Web vulnerabilities

### Risks:

1. revealing private information on server
2. intercept of client information (credit card records)
3. information about host  $\not\leq$  breakin
4. execute programs, denial of service
5. server log privacy

### Information Leakage:

- Altavista search for etc/passwd
- directory listings
- chroot
- soft links
- file ownership  $\not\leq$  local protection
- web access
- cgi-bin

63

## HTTP access control - principles

- client doesn't know which method
- client attempts access (GET, PUT, ...) normally
- server returns
  - HTTP/1.0 401 Unauthorized
  - WWW-Authenticate: Basic realm="WallyWorld"
- realm: protection space
- client tries again with
  - Authorization: Basic base64(user:password)
- passwords in the clear  $\not\leq$  not secure
- repeat cycle on each access

64

## Web Server Access Configuration

```
http://hoohoo.ncsa.uiuc.edu/docs/tutorials/user.html
For NCSA httpd, Apache not .htaccess per directory or global:
AuthType Basic
AuthUserFile /etc/passwd
AuthName "Private information"
<Limit GET>
order deny,allow
require user hgs
deny from all
allow from .ncsa.uiuc.edu
</Limit>

Global configuration file access.conf:
<Directory /full/path/to/protected/directory>
AuthName name.of.your.server
AuthType Basic
AuthUserFile /usr/local/etc/httpd/conf/passwd
<Limit GET POST>
require user foo
</Limit>
</Directory>
```

65

## Secure sockets layer (SSL)

- **transport layer security to any TCP-based app using SSL services.**
- Standardized as TLS (Transport-layer Security)
- used between Web browsers, servers for e-commerce (**shhttp**).
- 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.

66

## SSL (continued)

- **Encrypted SSL session:**
- Browser generates **symmetric session key**, encrypts it with server's public key, sends encrypted key to server.
- Using private key, server decrypts session key.
- Browser, server know session key
  - All data sent into TCP socket (by client or server) encrypted with session key.
- SSL: basis of IETF Transport Layer Security (TLS).
- SSL can be used for non-Web applications, e.g., IMAP.
- Client authentication can be done with client certificates.

67

## Outline

- General
  - What to Protect
  - Where to Put the Protection
  - Host and Network Based Security
- Cryptography
  - Symmetric Crypto Systems, DES
  - Asymmetric Crypto Systems, RSA
- Authentication
- Integrity
  - Digital Signature
- Key Distribution, Certificates
- Access Control: firewalls
- Email Security, PGP
- Security of the Web and Transport Layer (SSL)
- **IPSec**

68

## IPsec: Network Layer Security

- **Network layer secrecy:**
  - sending host encrypts the data in IP datagram
  - TCP and UDP segments; ICMP and SNMP messages.
- **Network layer authentication**
  - destination host can authenticate source IP address
- **Two principle protocols:**
  - authentication header (AH) protocol
  - encapsulation security payload (ESP) protocol
- **For both AH and ESP, source, destination handshake:**
  - create network-layer logical channel called a security association (SA)
- **Each SA unidirectional.**
- **Uniquely determined by:**
  - security protocol (AH or ESP)
  - source IP address
  - 32-bit connection ID

69

## Authentication Header (AH) Protocol

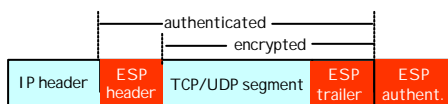
- provides source authentication, data integrity, no confidentiality
  - AH header inserted between IP header, data field.
  - protocol field: 51
  - intermediate routers process datagrams as usual
- AH header includes:**
- connection identifier
  - authentication data: source- signed message digest calculated over original IP datagram.
  - next header field: specifies type of data (e.g., TCP, UDP, ICMP)

IP header AH header data (e.g., TCP, UDP segment)

70

## ESP Protocol

- provides secrecy, host authentication, data integrity.
- data, ESP trailer encrypted.
- next header field is in ESP trailer.
- ESP authentication field is similar to AH authentication field.
- Protocol = 50.



71

## Network Security (summary)

### Basic techniques.....

- cryptography (symmetric and public)
- authentication
- message integrity
- key distribution

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

- secure email and web
- secure transport (SSL/TLS)
- IPsec

72