INTRODUCTION

IT424P–

Assoc. Prof. Noha A. Hikal

Head of IT Dept.

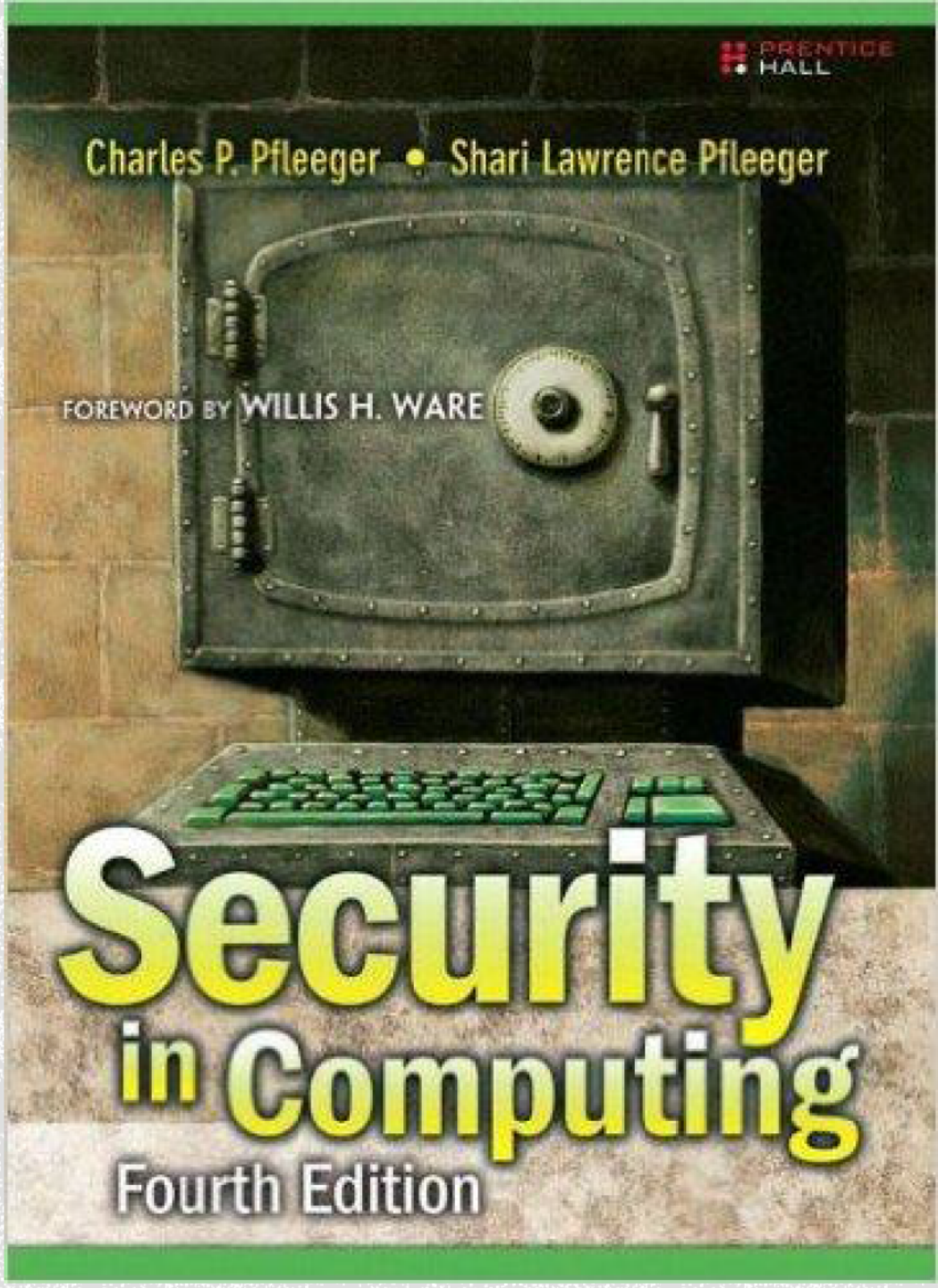
Dr_nahikal@mans.edu.eg

COURSE MATERIAL

- Textbook : Security in Computing by Pfleeger , 4st edition, 2006.

ISDN # 0132390779.

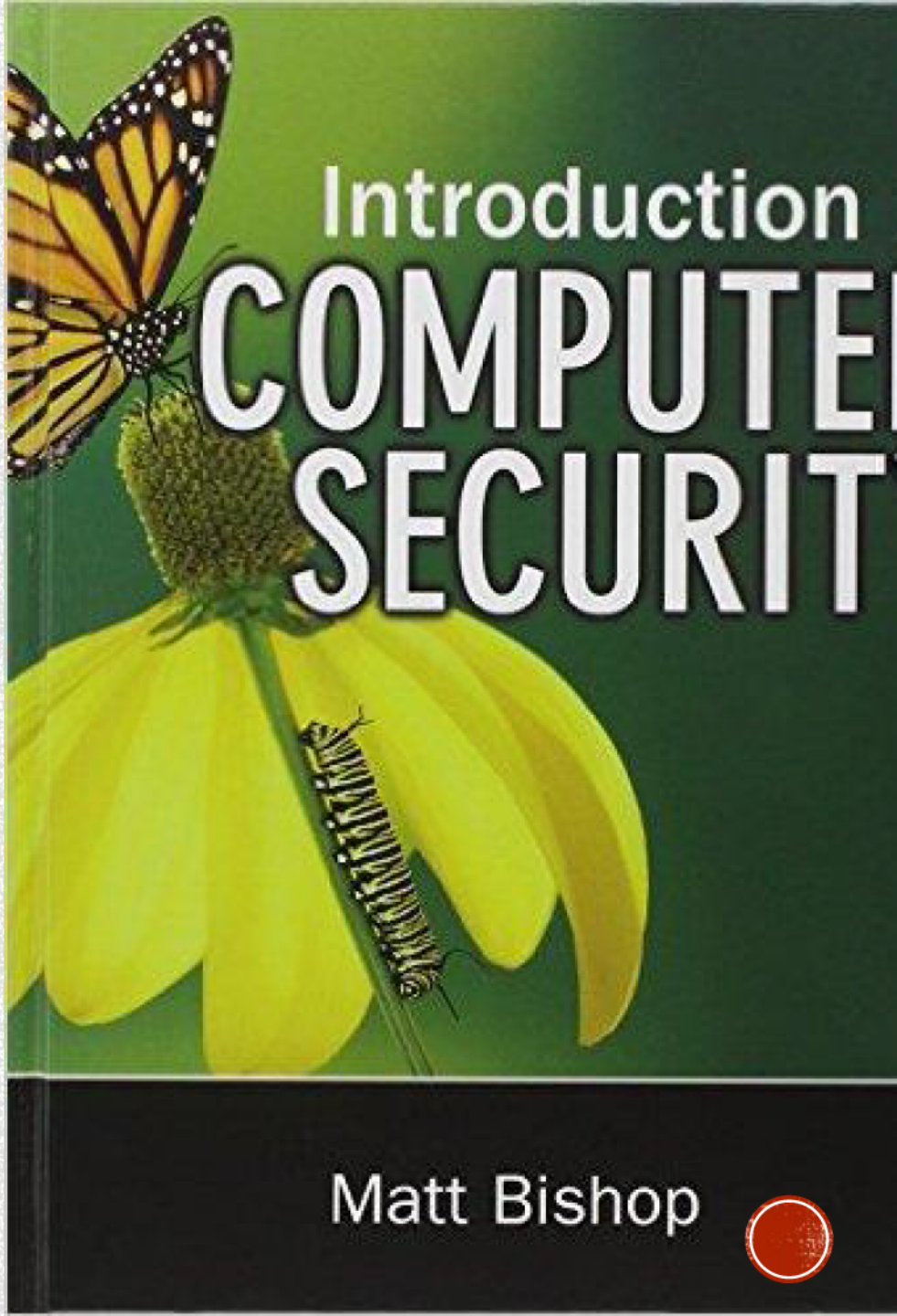
- Lecture notes
- Internet Resource



ADDITIONAL MATERIALS

- Textbook : Introduction to Computer Security by Matt Bishop , 1st edition, 2004.

ISDN #0321247442.



COMPUTER SECURITY

The NIST Computer Security Handbook [NIST95] defines the term computer security as:

- "The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/ data, and telecommunications)."

Hardware:

- Computer
- Devices (disk drives, memory, printer)
- Network gear

Software:

- Operating system
- Utilities (antivirus)
- Commercial applications (word processing, photo editing)
- Individual applications

Data:

- Documents
- Photos
- Music, videos
- Email
- Class projects

BASIC SECURITY COMPONENTS

Confidentiality

- The need for keeping information secrete

Integrity

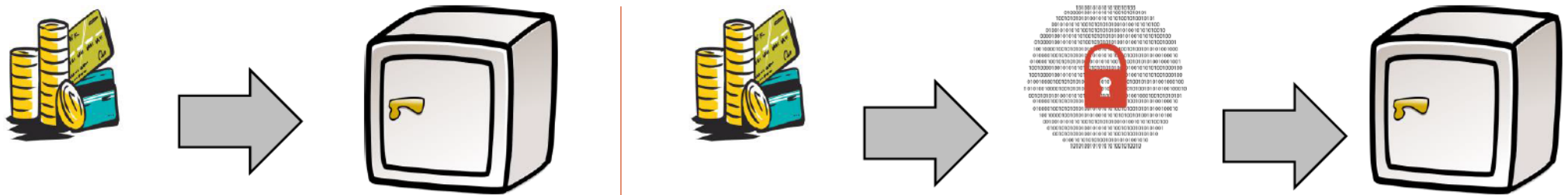
- Trustworthiness of data or resources

Availability

- The ability to use the information or resource.

CONFIDENTIALITY

- *Definition:* is concealment of information or resources.
- The need of keeping the information confident depends on the sensitivity of the information.
  - E.g. Military information vs. industrial information vs. personal information.
- Achieving higher confidentiality is supported by:
  - Access Control: Ensures only authorized personnel have access to the information.
  - Cryptography: When unauthorized access occurred, information is meaningless.



Readable & accessible

Readable & NOT accessible

Readable & accessible

NOT Readable But accessible

NOT Readable NOT accessible

some properties that could mean a failure of data confidentiality:

- An unauthorized person accesses a data item.
- An unauthorized process or program accesses a data item.
- A person authorized to access certain data accesses other data not authorized
- An unauthorized person accesses an approximate data value (for example, not knowing someone's exact salary but knowing that the salary falls in a particular range or exceeds a particular amount).
- An unauthorized person learns the existence of a piece of data (for example, knowing that a company is developing a certain new product or that talks are underway about the merger of two companies).

INTEGRITY

- *Definition:* Trustworthiness of the data or resource. Prevent improper or unauthorized change.
- Integrity includes:
  - Data integrity: Content of the data.
  - Origin integrity : Source of the data using authentication.
- Integrity Mechanism falls into two classes:
  - Prevention: Blocking unauthorized attempt to modify data.
  - Detection: Report when unauthorized modification occurs.
- Evaluating integrity is hard because some attempts seems legit.

© Can Stock Photo

If we say that we have preserved the integrity of an item, we may mean that the item is:

- Precise
- Accurate
- Unmodified
- Modified only in acceptable ways
- Modified only by authorized people
- Modified only by authorized processes
- Consistent
- Internally consistent
- Meaningful and usable

AVAILABILITY

- *Definition:* Ensures that authorized access to data when desired.
- Attempt to block service (System down) or make the service unavailable is called Denial of service attack (DoS).

An object or service is thought to be available if the following are true:

• It is present in a usable form.

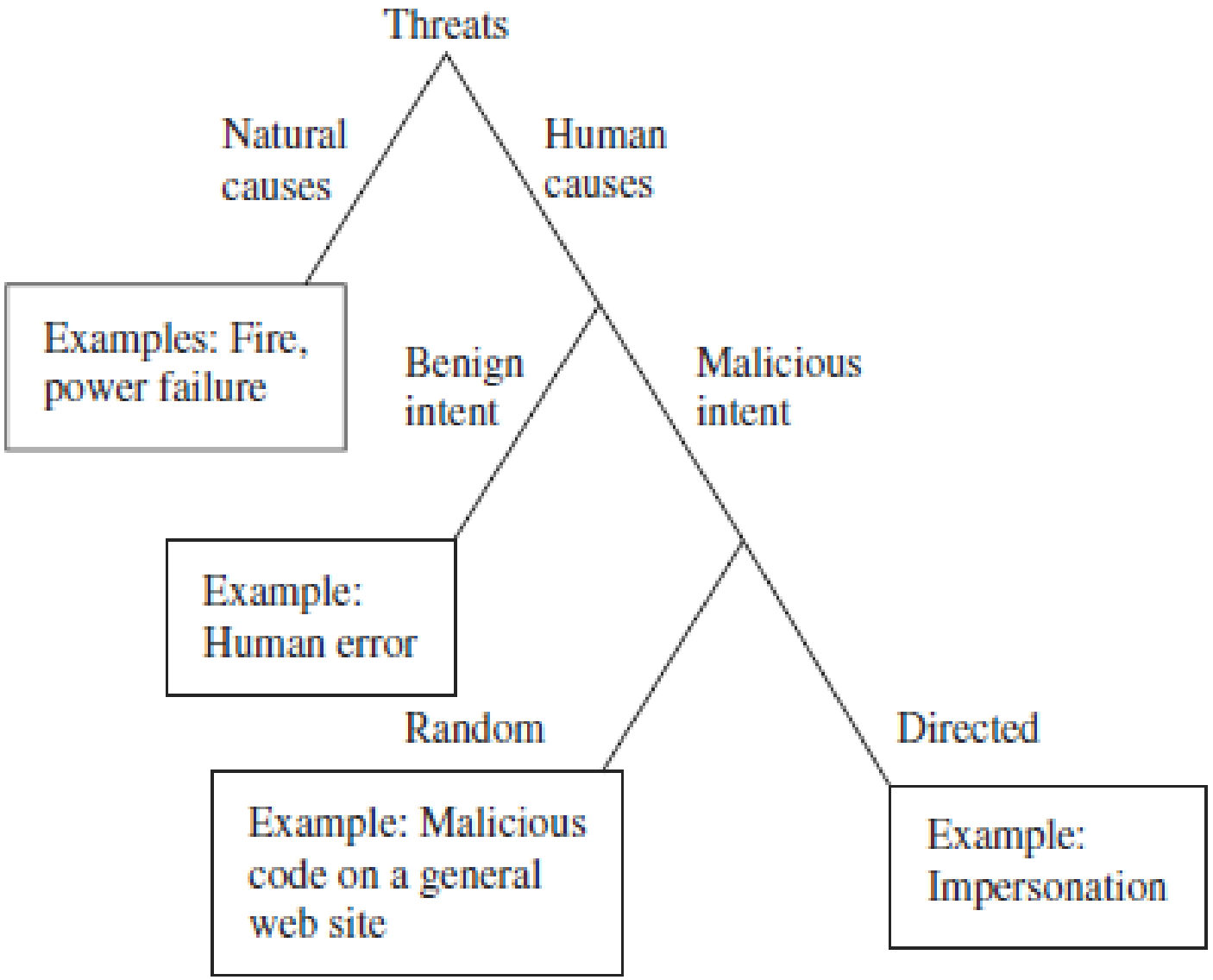
• It has enough capacity to meet the service's needs.

• It is making clear progress, and, if in wait mode, it has a bounded waiting time.

• The service is completed in an acceptable period of time. There is a timely response to our request.

• Resources are allocated fairly so that some requesters are not favored over others.

THREATS


Threats
Natural causes
Human causes
Examples: Fire, power failure
Benign intent
Malicious intent
Example: Human error
Random
Directed
Example: Malicious code on a general web site
Example: Impersonation

Threats to hardware:

Usually, it is the concern of a small staff of computer center professionals. Hardware threats can be: *Involuntary*: (like accidents; pouring water, food, etc). *Voluntary*: in which some actually wishes to do harm to the computer (bombs ,fires, ,theft, shorting out circuit boards)

-*Threats of software*:

The concern of all programmers and analysts who create and modify programs:

Software deletion.

Software theft.

Software modification (either to cause the program fails during execution or fails in some special circumstances (logic bomb) or to cause it to do some unintended task.

The category of software modification includes:

- Trojan horse: a program that overtly does one thing while covertly doing another

-Virus : a specific type of Trojan horse, that can be used to spread infection from one computer to another.

-Trapdoor: a program that has a secret entry point.

-Information leaks: that makes information accessible

to unintended people or programs.

Threats of data(the concern of general public, so data attack is a more widespread and serious problem than either hardware or software) examples:

- Confidential data leaked to a competitor may narrow a competitive edge.
- Data incorrectly modified can cost human lives

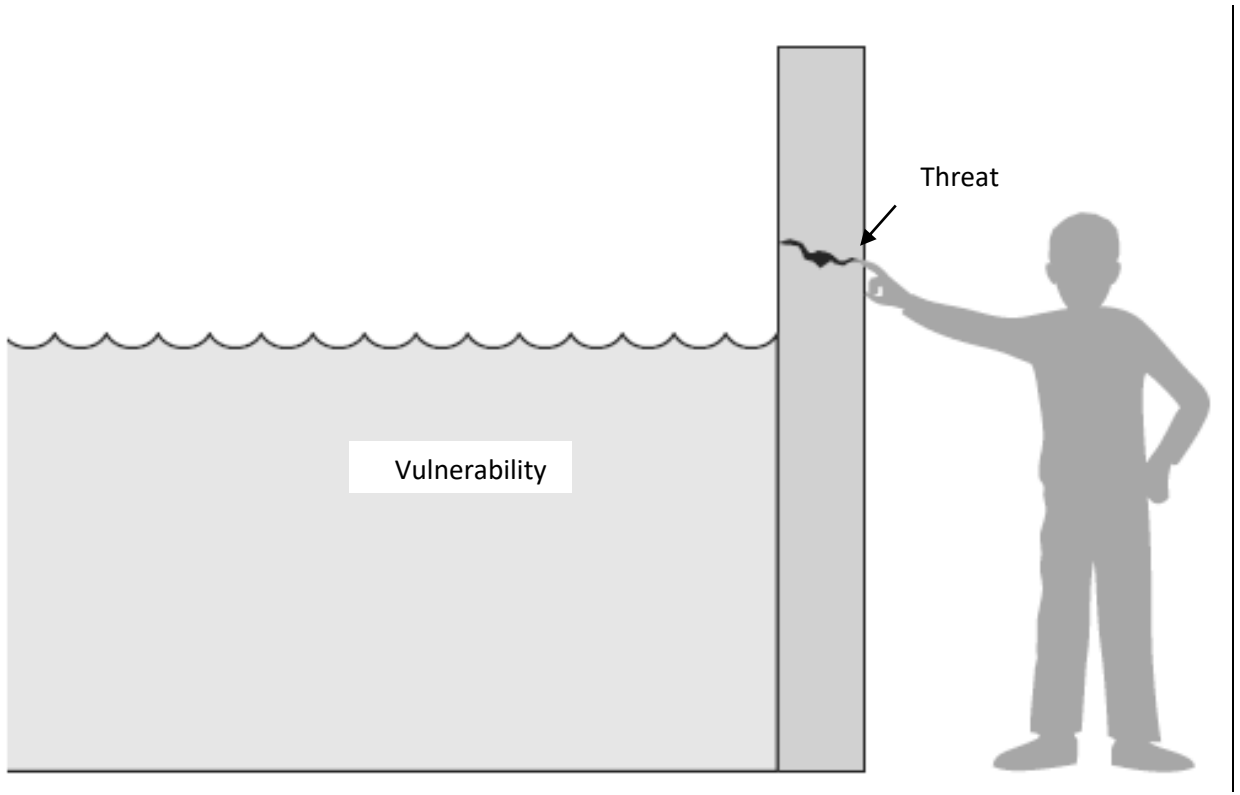
The qualities of data security;

1. **Confidentiality** – (preventing unauthorized disclosure): data can be gathered by tapping wires planting bugs in output devices, from trashes, monitoring electromagnetic radiation, bribing key employees, inferring on data point from other values.
2. **Data integrity –(**preventing unauthorized modification) modifying or making a new data requires understanding the technology by which data stored transmitted and it's format ,and this might be done by using malicious programs Example(**salami attack)**
3. **Availability (**preventing denial of authorized access)

Other exposed assets

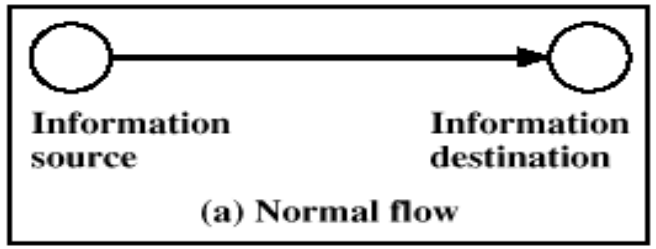
- **Storage media** (effective security plans consider adequate backups of data and physical protection for the media contains these backups.
- **Networks** – a collections of software, hardware , and data and this simply multiply the problem of security.
- **Access to computer equipment (the** intruder may steal computer time just to do computing and he can destroy software or data and this may lead to the denial of the service to a legitimate user .
- **Key People** (if only one person knows how to use or maintain a particular program –trouble can arise if he gets sick, has an accident or leaves ,
  - disgruntled employees can cause serious damage
  - Trusted individuals should be selected carefully

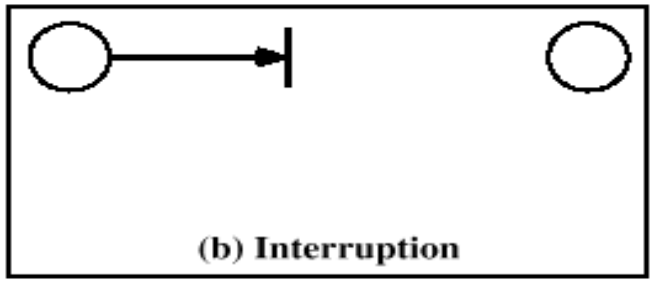
- **Vulnerability:** it refers to a weakness in the computer system (i,e; policy, design, implementation, or procedure) that can be exploited to cause harm or loss.
- **Threats:** A threat to a computing system is a set of circumstances that has the potential to cause loss or harm.

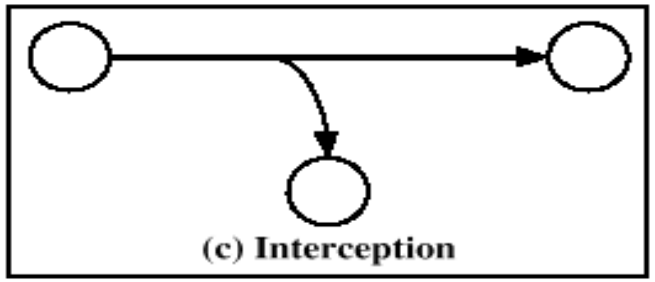


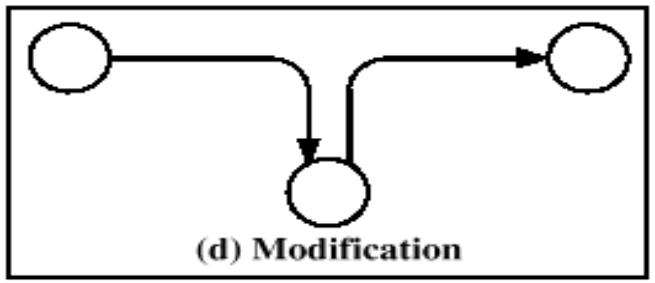
A THREAT IS BLOCKED BY CONTROL OF A VULNERABILITY

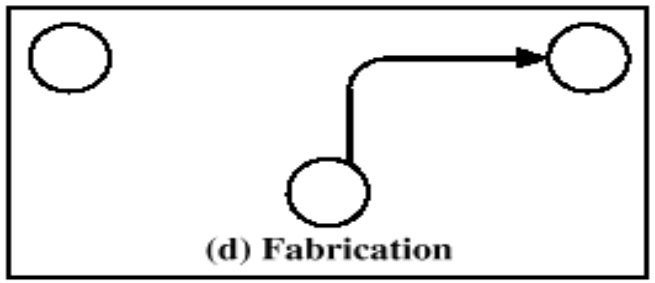
ACTIONS OF THREATS











Threats to the security of a computing system

- **Interruption:** an asset of the system becomes lost, unavailable or unusable (example: Destruction of hardware, erasure of program or data or malfunction of an OS file manager .
- **Interception:** means that some unauthorized party (person, program) has gain access to an asset (example: illicit copying of program or data files, or wiretapping to obtain data in a network**.)**
- **Modification:** Example(changing the values in a database modifying a program so that it performs an additional computation, or modifying data being transmitted by the network).
- **Fabrication :** counterfeit objects on a computing system. (adding records to an existing data base or insertion of spurious transactions to a network communication system.

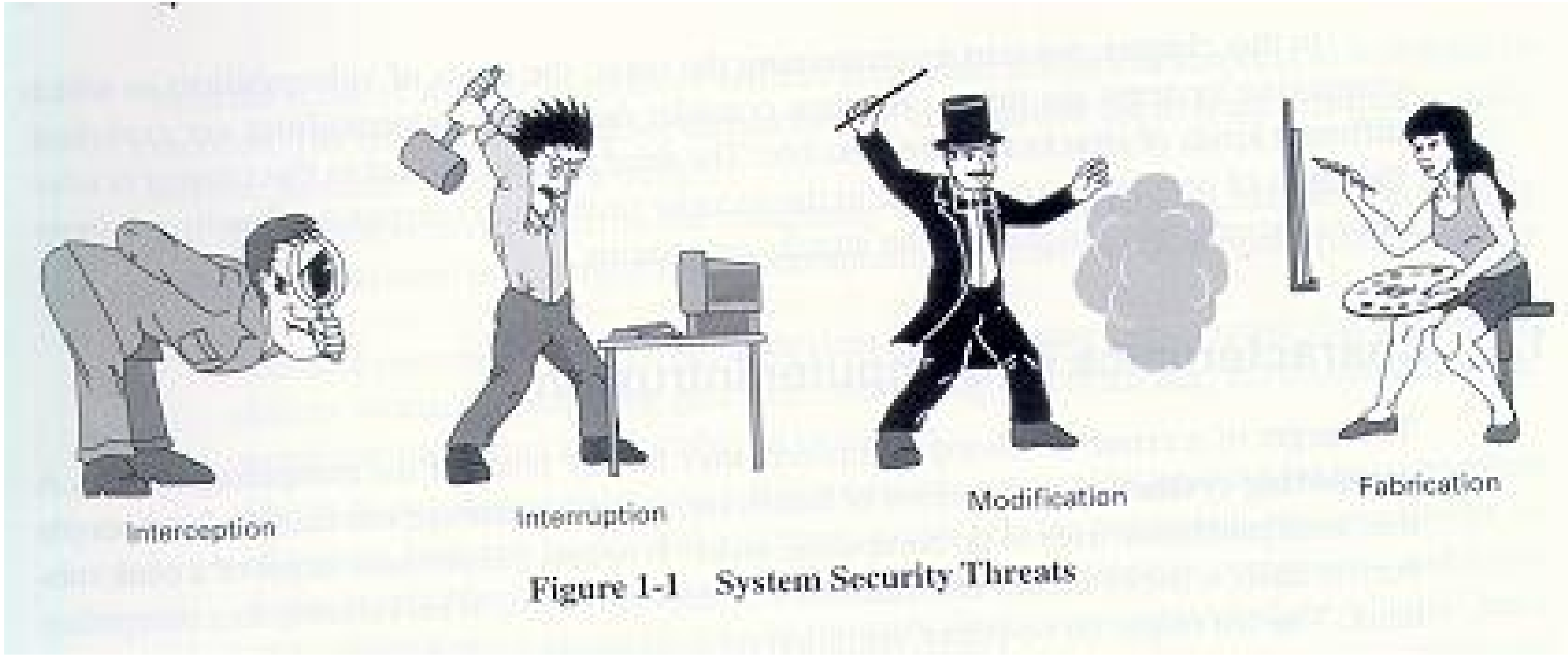
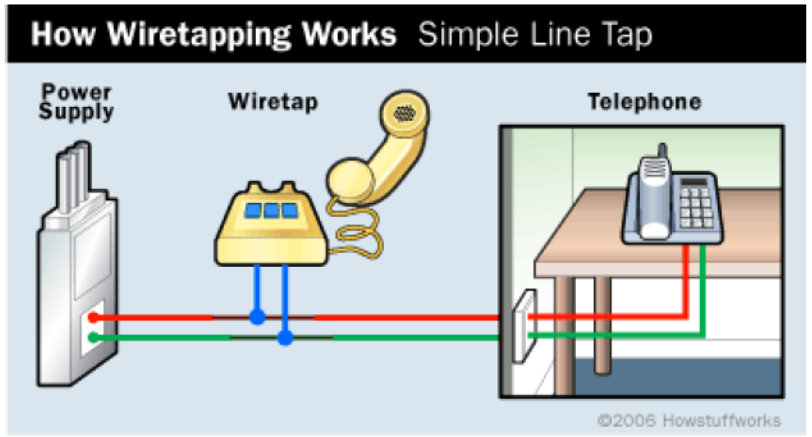


Figure 1-1 System Security Threats

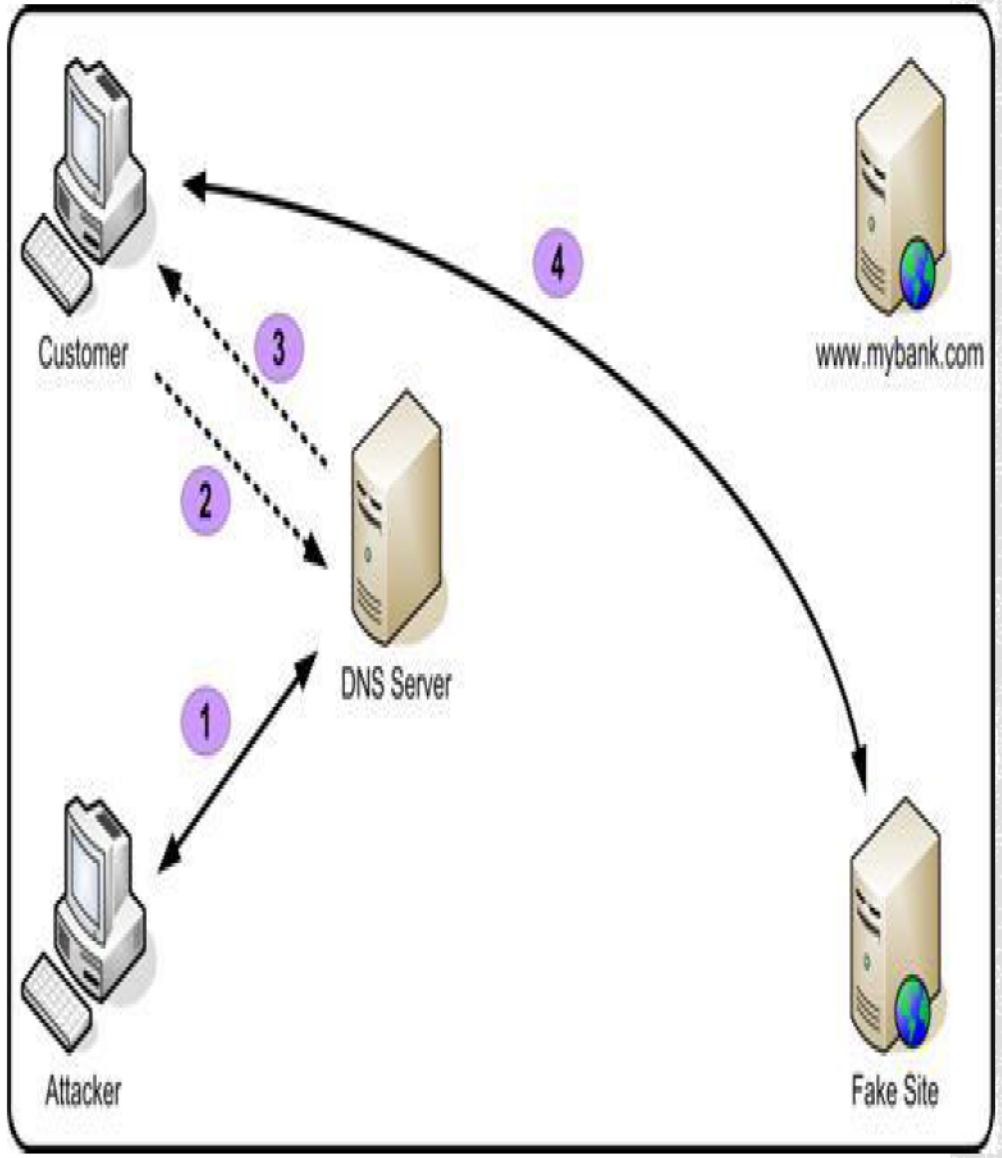


INTERCEPTING ATTACKS: SNOOPING

- Unauthorized interception of information.
- Confidentiality attack.
- Passive attack : No action beyond listing and monitoring activities.
- Example: wiretapping

ATTACKS: MODIFICATION(ALTERATION)

- Unauthorized change to the information.
- CIA attack.
- It may start as deception: giving system false data (e.g. engine hot). Then system might go on recovery mode (Full access), then you have control over the system (usurpation and disruption).
- This is active attack.
- Example: man-in-the-middle attack.



ATTACKS: MASQUERADING OR SPOOFING

- Impersonation of one entity by another.
- Integrity attack.
- Could be passive or active attack.
- Example: Bank website

MODIFICATION ATTACKS: REPUDIATION OF ORIGIN

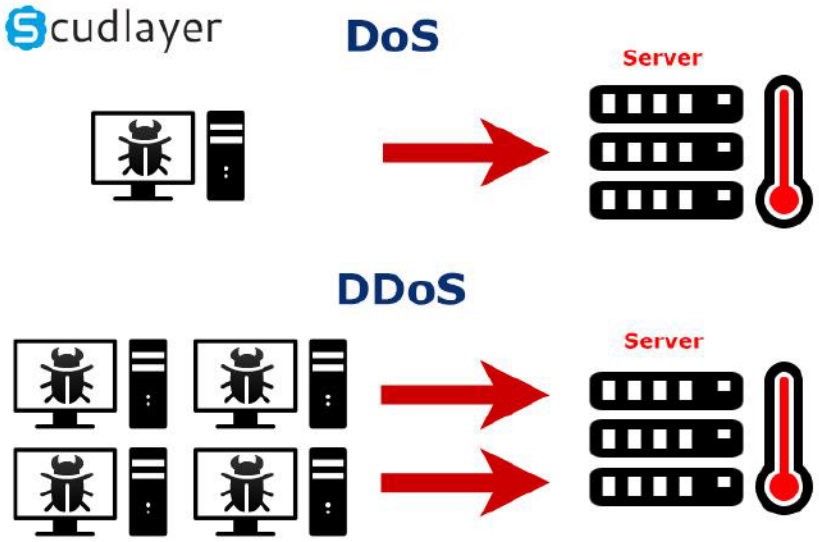
- False denial that an entity sent/created something,.
- Integrity attack.
- Example: Pizza Order

ATTACKS: DENIAL OF RECEIPT

- A false denial that an entity received some information or message.
- Example: order expensive good only.

ATTACKS: DELAY

- Temporary inhibition of service
- Availability attack.
- Example: head of line queue.



ATTACKS: DELAY

- A long-Term inhibition of service
- Availability attack.
- Example: head of line queue.

TO ACHIEVE GOALS OF SECURITY

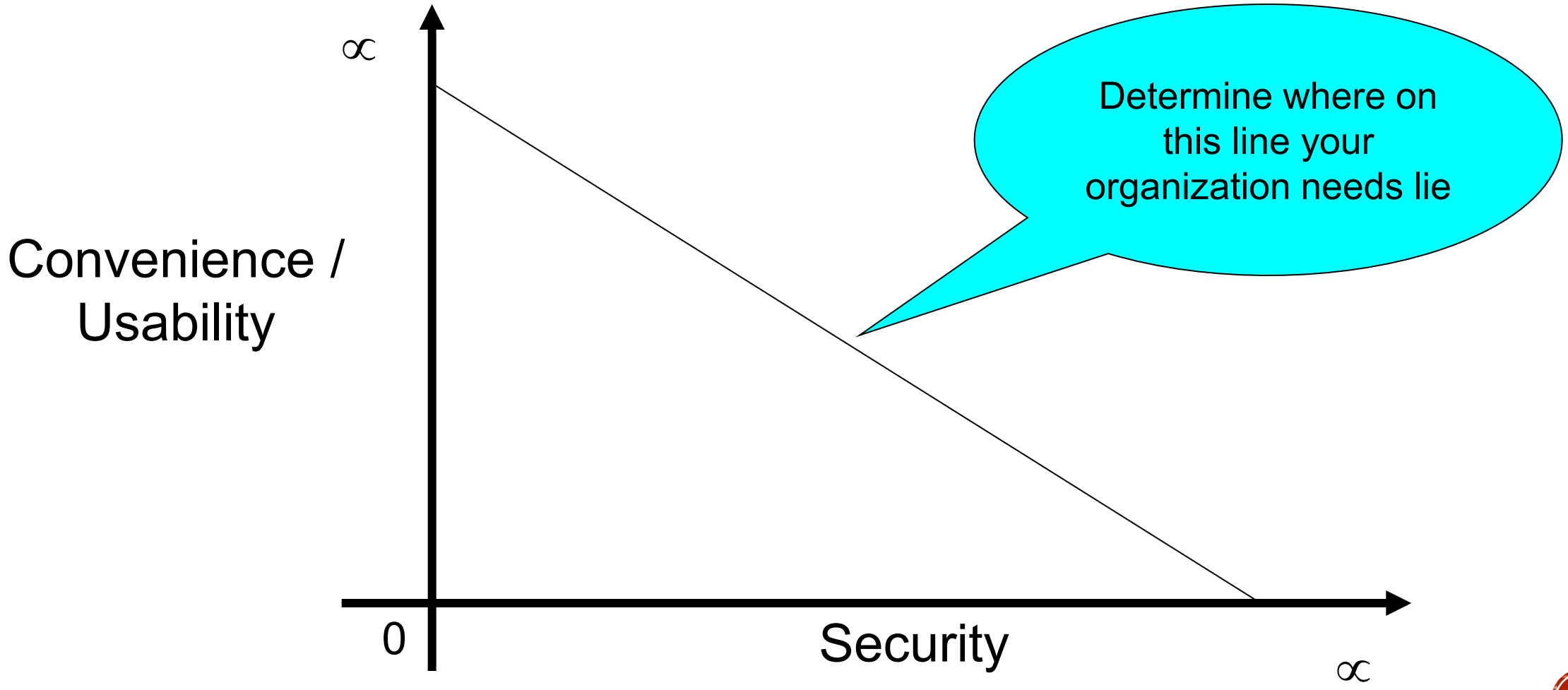
- The attack will fail.
- Involves implementation of mechanisms that users cannot override.
- Must be implemented correctly & unalterable way.
- Attackers must not be able to beat this mechanism

- It is useful when an attack can not be prevented.
- An indication of the effectiveness of prevention mechanism.
- Accept that an attack will occur, the goal is to determine that an attack on the way.

- A complex process because attacks are different.
- Two ways to recover:

1. Stop the attack and to assess & repair and damage caused by the attacker.
2. Recover the system to function.

USABILITY AND SECURITY



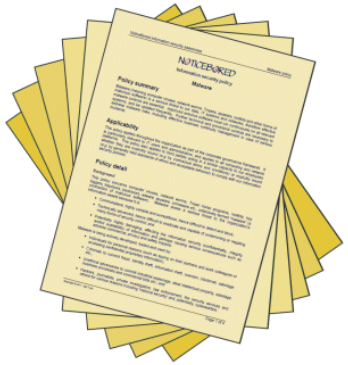
Methods of defense continued

3. **Hardware controls (**hardware or smartcard implementation of encryption to locks limitation access, to theft protection, to circuit boards that control access to disk drivers in PCs.)
4. **Polices** examples:
   - **Frequent changing passwords**
   - **Legal controls**
   - **Ethical controls**
   - **Training and administration**
5. **Physical controls:** include (locks on doors, guards at entry points, backup copies of important software and data and physical site planning that reduces the risk of natural disasters

FACTORS THAT AFFECT THE EFFECTIVENESS OF CONTROLS

1. **Awareness of problem** (people using controls must be convinced of the need for security)
2. **Likelihood of use** (no control is effective until it is used)
3. **Overlapping controls** (several different controls may be used)
4. **Periodic review (continuous efforts to improve the methods of defense)**

POLICY AND MECHANISM



- **Security Policy:** is the statement of what is, and what is not allowed.
- Policy can be presented in mathematical form.
  - Example: Binary matrices of what is allowed and what is not.
- Usually policies are written in **English** language.

POLICY AND MECHANISM (CONT.)

- **Security Mechanism:** is the method or tool or procedure for enforcing a security policy.
- Mechanism can be nontechnical.
  - Example: Presenting your ID (Physically) to change your bank phone number.
- Mechanism Can be Technical.
  - Example: You must change your password every 6 months.

OPERATIONAL ISSUES

- It is a trade off between cost, implementation and mechanism verses policy and mechanism .
- **Cost-Benefit Analysis:**
- The total cost of the system determines how much many to invest in security.
- If data cost less than the cost of protecting it, then investing in security mechanism is cost-effective.
- **Risk analysis:**
  - Risk is a function of environment : Attacker from overseas can not attack an offline system.
  - Risk changes with time: Amazon down-time during Christmas is catastrophic.
  - Risk is remote but still exist: companies allow internet access to some computers not all. Because it is acceptable to some.

NEXT

PEOPLE IN THIS COURSE

GOOD PEOPLE

Alice

Bob

BAD PERSON

Oscar

CRYPTOGRAPHY

Computer Security – IT424

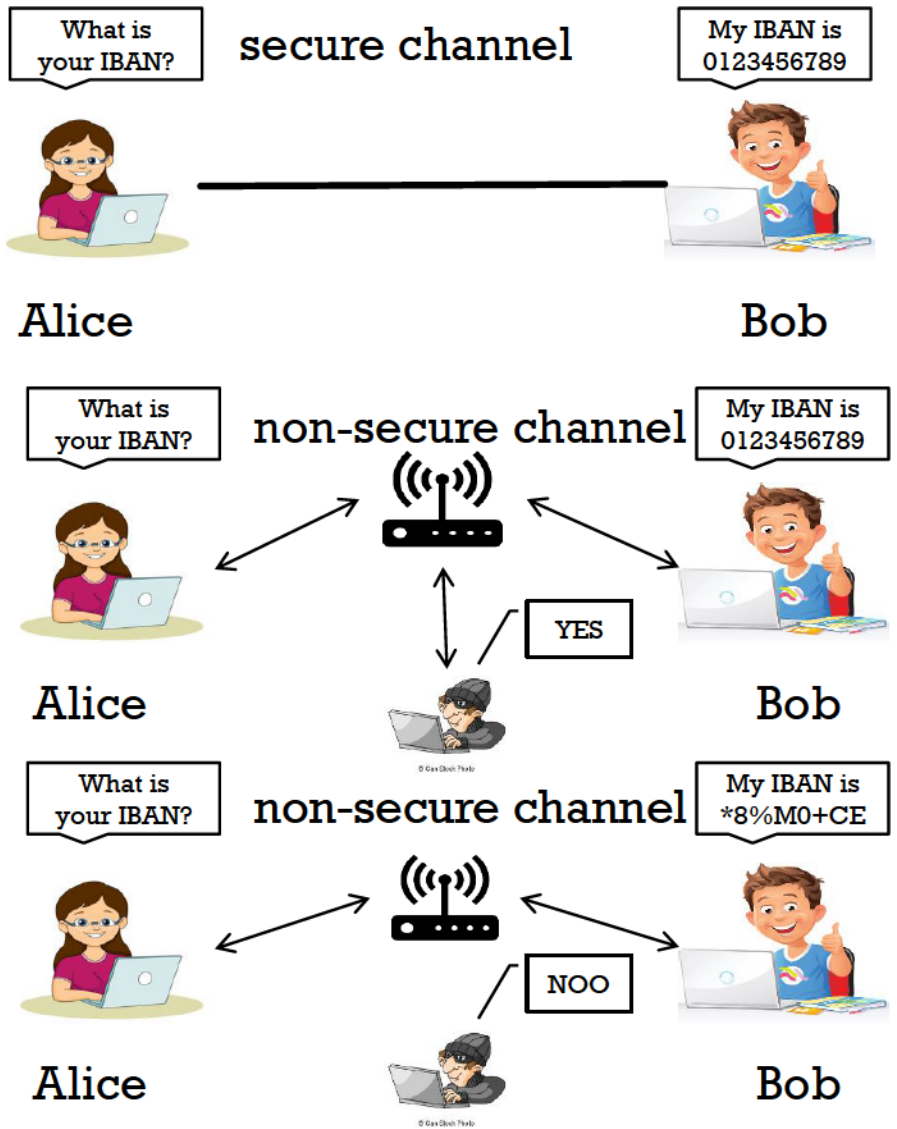
Dr. Noha Hikal

WHY ENCRYPTION?

Assuming that Alice and Bob are communicating with each other:
Alice would like to have Bob's IBAN number.
Three scenarios might occur:

1. Communicating over a secure channel
2. Communicating over a non-secure channel in plaintext.
3. Communicating over a non-secure channel in encrypted text.

Next week we will go over encryption in greater details.



PROBLEMS ADDRESSED BY ENCRYPTION

Assuming that, a **sender** (S) sends a message to a **recipient** (R) over a **transmission medium** (T).

If an **outsider** (O) wants to access the message, we call him intruder (or interceptor).

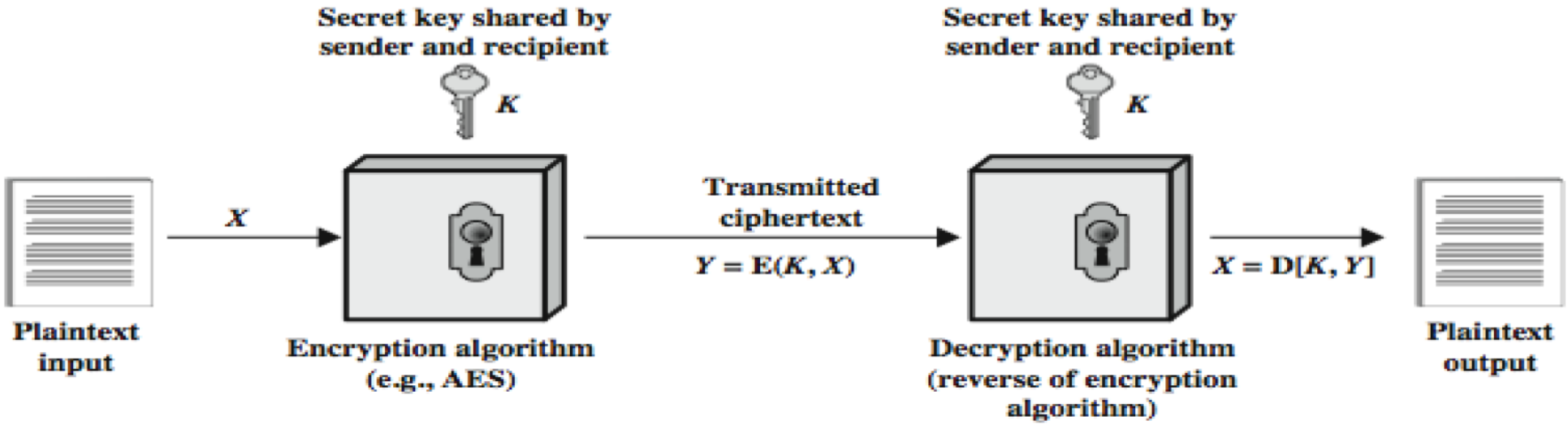
The outsider might take one (or more) of the following actions:

- Block the message from reaching R. (Affecting availability)
- Intercept the message by reading or listening. (affecting confidentiality)
- Modify the message. (Affecting integrity)
- Fabricate the message. (affecting the integrity)

Some Basic Terminology

- **plaintext** - original message
- **ciphertext** - coded message
- **cipher** - algorithm for transforming plaintext to ciphertext
- **key** - info used in cipher known only to sender/receiver
- **encipher (encrypt)** - converting plaintext to ciphertext
- **decipher (decrypt)** - recovering plaintext from ciphertext
- **cryptography** - study of encryption principles/methods
- **cryptanalysis (codebreaking)** - study of principles/ methods of deciphering ciphertext *without* knowing key
- **cryptology** - field of both cryptography and cryptanalysis

Symmetric Cipher Model



Requirements

- two requirements for secure use of symmetric encryption:
  - a strong encryption algorithm
  - a secret key known only to sender / receiver
- mathematically have:

  $Y = E(K, X) = E_K(X) = \{X\}_K$

  $X = D(K, Y) = D_K(Y)$
- assume encryption algorithm is known
  - Kerckhoff's Principle: security in secrecy of key alone, not in obscurity of the encryption algorithm
- implies a secure channel to distribute key
  - Central problem in symmetric cryptography

Cryptography

- can characterize cryptographic system by:
  - type of encryption operations used
    - substitution
    - transposition
    - product
  - number of keys used
    - single-key or private
    - two-key or public
  - way in which plaintext is processed
    - block
    - stream

Cryptanalysis

- objective to recover key not just message
- general approaches:
  - cryptanalytic attack
  - brute-force attack
- if either succeed all key use compromised

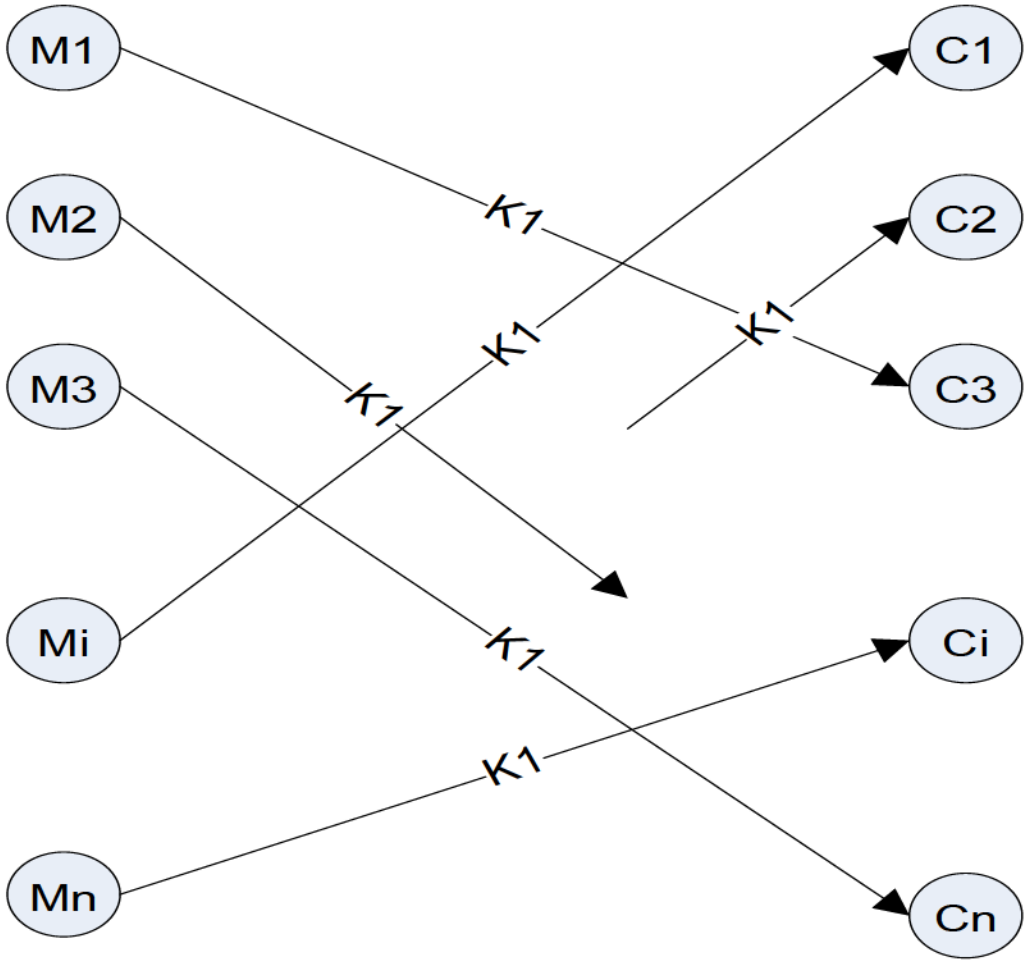
Cryptanalytic Attacks

- **ciphertext only**
  - only know algorithm & ciphertext, is statistical, can identify plaintext
- **known plaintext**
  - know/suspect plaintext & ciphertext
- **chosen plaintext**
  - select plaintext and obtain ciphertext
- **chosen ciphertext**
  - select ciphertext and obtain plaintext
- **chosen text**
  - select plaintext or ciphertext to en/decrypt

Cipher Strength

- **unconditional security**
  - no matter how much computer power or time is available, the cipher cannot be broken since the ciphertext provides insufficient information to uniquely determine the corresponding plaintext
- **computational security**
  - given limited computing resources (e.g. time needed for calculations is greater than age of universe), the cipher cannot be broken

Encryption Mappings

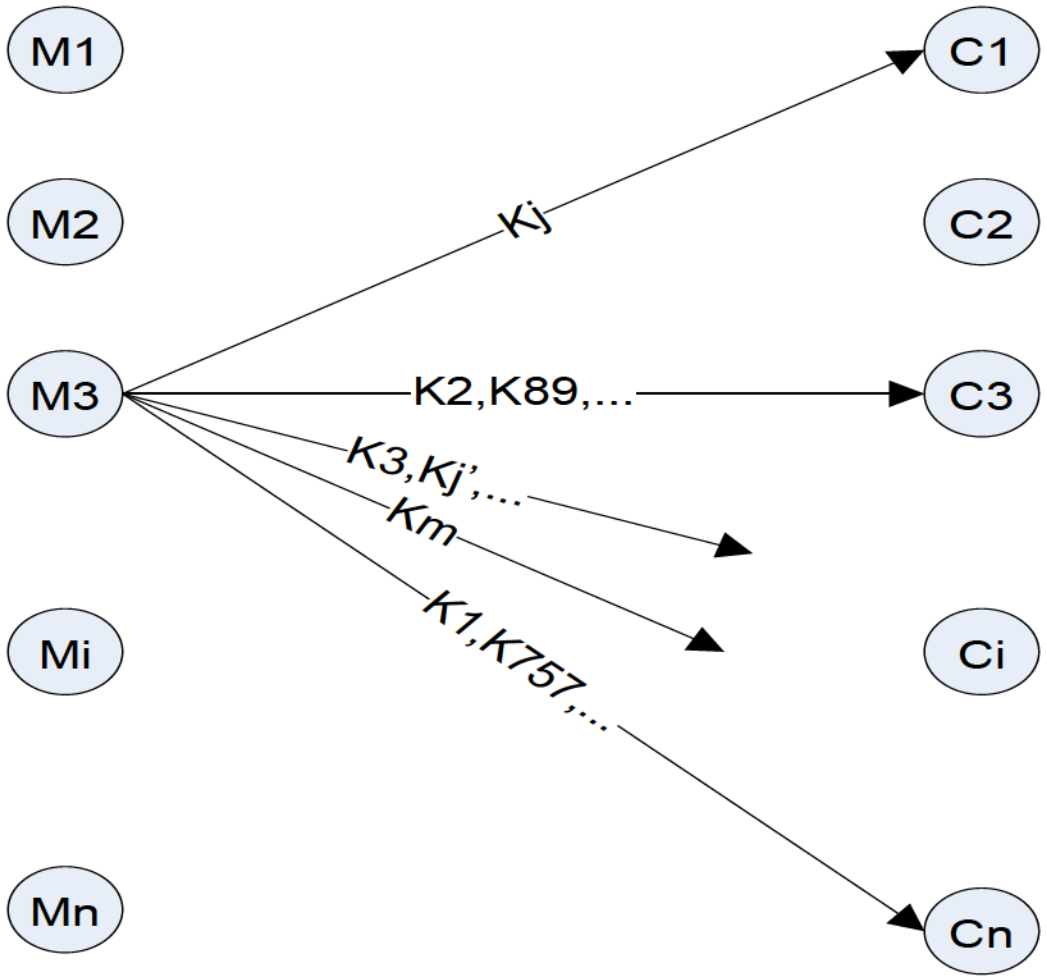


M=set of all plaintexts

C=set of all ciphertexts

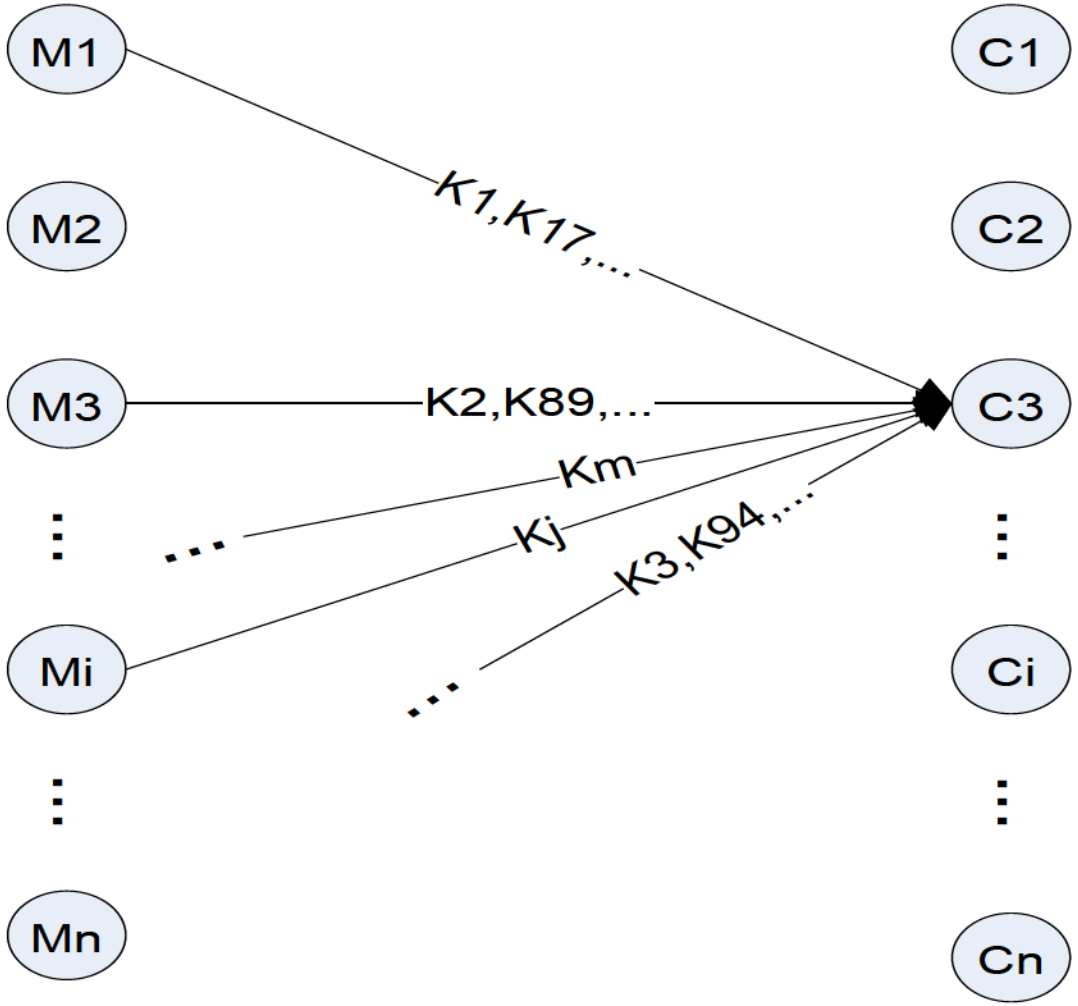
- **A given key (k)**
  - Maps any message Mi to some ciphertext E(k,Mi)
  - Ciphertext image of Mi is unique to Mi under k
  - Plaintext pre-image of Ci is unique to Ci under k
- **Notation**
  - key k and Mi in M, ∃! Cj in C such that E(k,Mi) = Cj
  - key k and ciphertext Ci in C, ∃! Mj in M such that E(k,Mj) = Ci
  - $E_k(.)$ is "one-to-one" (injective)
  - If |M|=|C| it is also "onto" (surjective), and hence bijective.

Encryption Mappings (2)



- **A given plaintext (Mi)**
  - Mi is mapped to *some* ciphertext E(K,Mi) by every key k
  - Different keys may map Mi to the same ciphertext
  - There may be some ciphertexts to which Mi is never mapped by any key
- **Notation**
  - key k and Mi in M, ∃! ciphertext Cj in C such that E(k,Mi) = Cj
  - It is possible that there are keys k and k' such that E(k,Mi) = E(k',Mi)
  - There may be some ciphertext Cj for which ∃ key k such that E(k,Mi) = Cj

Encryption Mappings (3)



- **A ciphertext (Ci)**
  - Has a unique plaintext pre-image under each k
  - May have two keys that map the same plaintext to it
  - There may be some plaintext Mj such that no key maps Mj to Ci
- **Notation**
  - key k and ciphertext Ci in C, ∃! Mj in M such that E(k,Mj) = Ci
  - There may exist keys k, k' and plaintext Mj such that E(k,Mj) = E(k',Mj) = Ci
  - There may exist plaintext Mj such that ∃ key k such that E(k,Mj) = Ci

Encryption Mappings (4)

- Under what conditions will there always be some key that maps some plaintext to a given ciphertext?
- If for an intercepted ciphertext c_j, there is some plaintext m_i for which there does not exist any key k that maps m_i to c_j, then the attacker has learned something
- If the attacker has ciphertext c_j and known plaintext m_i, then many keys may be eliminated

Brute Force Search

- always possible to simply try every key
- most basic attack, exponential in key length
- assume either know / recognise plaintext

Key Size (bits)	Number of Alternative Keys	Time required at 1 decryption/μs	Time required at 10^6 decryptions/μs
32	$2^{32} = 4.3 \times 10^9$	2^{31} μs = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} μs = 1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} μs = 5.4×10^{24} years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} μs = 5.9×10^{36} years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	2×10^{26} μs = 6.4×10^{12} years	6.4×10^6 years

TERMINOLOGY

Encryption *(E)* (encoding - enciphering): The process of encoding a message so that its meaning is not obvious.

Decryption*(D)*(decoding - deciphering): Is the reverse process.

Cryptosystem: A system that encrypts/decrypts.

Plaintext *(P)*: The original form of a message.

Cipher text *(C)*: the encrypted form of a message.

NOTATION

Processing a plaintext *(P)* using the encryption rule *(E)*, results on cipher text *(C)*, and denoted by:

$$C = E(P)$$

The reverse process is denoted by:

$$P = D(C)$$

Question:

What is D(E(P))?

Classical Substitution Ciphers

- where letters of plaintext are replaced by other letters or by numbers or symbols
- or if plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns

CAESAR CIPHER

- Caesar cipher is one of the oldest encryption algorithms.
- It was created by Julius Caesar (100 – 44 BC).
- The idea is simple:
  1. Index every character from 0 to 25
  2. When encrypting, shift every character by 3 (revise it when decrypting)
- Caesar agreed with his people to use the encryption key = 3 (K = 3).

CAESAR CIPHER

Example: Encrypt the word (KEY) using Caesar cipher ?

The corresponding index (*i*) for each letter in the plaintext is { 10, 4, 24 } respectively.

The mathematical formula is : *(i + 3) mod 26.*

The corresponding index (*i*) for each letter in the cipher text is { 13, 7, 1 } respectively.

The corresponding cipher text is (NHB).

0	1	2	3	4	5	6	7	8	9	10	11	12
A	B	C	D	E	F	G	H	I	J	K	L	M
13	14	15	16	17	18	19	20	21	22	23	24	25
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

SHIFT CIPHER

- Caesar cipher is a special case of the shift cipher when the key K = 3.
- Shift cipher is a generalized formula and the value varies between 0 to 26.
- Shift cipher is an example of a symmetric cryptosystem.
- EX:

meet me after the toga party

PHHW PH DIWHU WKH WRJD SDUWB

Playfair Cipher

- not even the large number of keys in a monoalphabetic cipher provides security
- one approach to improving security was to encrypt multiple letters
- the **Playfair Cipher** is an example
- invented by Charles Wheatstone in 1854, but named after his friend Baron Playfair

Playfair Key Matrix

- a 5X5 matrix of letters based on a keyword
- fill in letters of keyword (sans duplicates)
- fill rest of matrix with other letters
- eg. using the keyword MONARCHY

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Encrypting and Decrypting

- plaintext is encrypted two letters at a time
  1. if a pair is a repeated letter, insert filler like 'X'
  2. if both letters fall in the same row, replace each with letter to right (wrapping back to start from end)
  3. if both letters fall in the same column, replace each with the letter below it (wrapping to top from bottom)
  4. otherwise each letter is replaced by the letter in the same row and in the column of the other letter of the pair

Playfair Example

- Message = Move forward
- Plaintext = mo ve fo rw ar dx
  - Here x is just a filler, message is padded and segmented
- mo -> ON; ve -> UF; fo -> PH, etc.
- Ciphertext = ON UF PH NZ RM BZ

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Transposition Ciphers

- now consider classical **transposition** or **permutation** ciphers
- these hide the message by rearranging the letter order
- without altering the actual letters used
- can recognise these since have the same frequency distribution as the original text

Rail Fence cipher

- write message letters out diagonally over a number of rows
- then read off cipher row by row
- eg. write message out as:

```
m e m a t r h t g p r y
  e t e f e t e o a a t
```

- giving ciphertext

MEMATRHTGPRYETEFETEOAAT

Row Transposition Ciphers

- is a more complex transposition
- write letters of message out in rows over a specified number of columns
- then reorder the columns according to some key before reading off the rows

```
Key: 4312567  p: attack postponed until two
Column Out 4 3 1 2 5 6 7
Plaintext: a t t a c k p
           o s t p o n e
           d u n t i l t
           w o a m x y z
Ciphertext: TTNAAPTMTSUOAODWCOIXKNLYPETZ
```


Block Transposition Ciphers

- arbitrary block transposition may be used
- specify permutation on block
- repeat for each block of plaintext

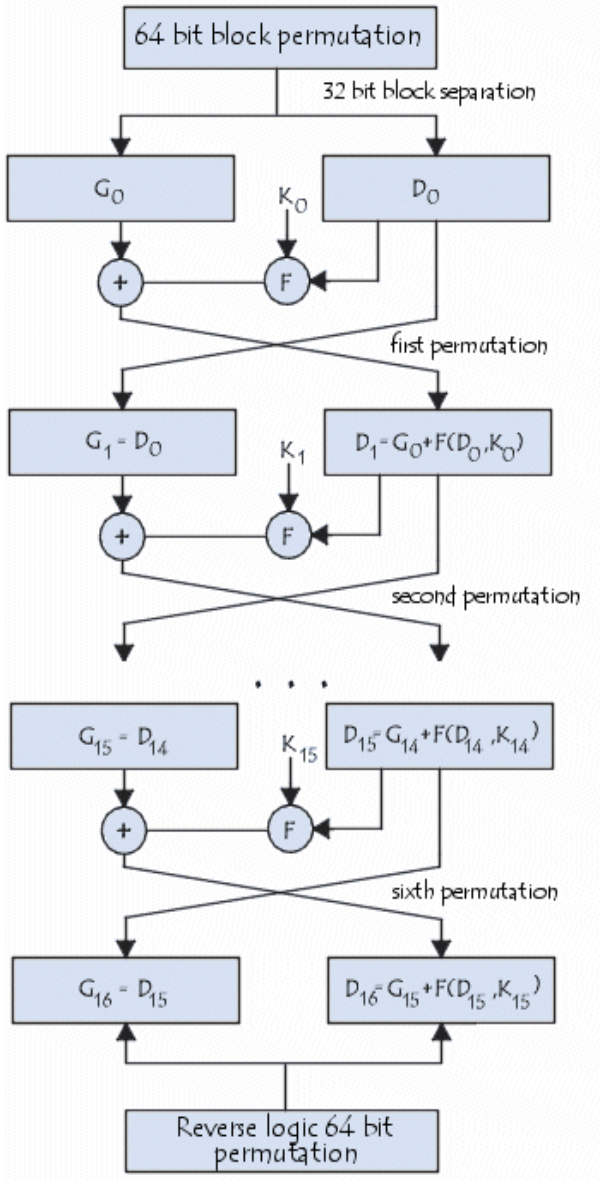
```
Key: 4931285607
Plaintext:  attackpost poneduntil twoamxyzab

Ciphertext: CTATTSKPAO DLEONIDUPT MBAWOAXYTZ
```

Product Ciphers

- ciphers using substitutions or transpositions are not secure because of language characteristics
- hence consider using several ciphers in succession to make harder, but:
  - two substitutions make a more complex substitution
  - two transpositions make more complex transposition
  - but a substitution followed by a transposition makes a new much harder cipher
- this is bridge from classical to modern ciphers

DATA ENCRYPTION STANDARD (DES)

- It was developed in 1970 by IBM.
- DES is a block ciphering algorithm.
- DES is a symmetric encryption algorithm.
- DES is complex encryption algorithm that uses: substitution and transposition.
- DES starts by chopping each message into 64bits (8 bytes).
- The key is (also) 64 bits + 8 bits for check digits = 56 bits.
- 16 iterations are involved.
- In 1977, Diffie and Hellman argued that 56-bits key is short.
- In 1997, 3500 of parallel PC was able to break a DES cipher in 4 months.



DOUBLE DES

- In 1998, a US$ 200,000 "DES cracker" machine was invented to crack any DES in 4 days.
- Later, DES is deciphered in few hours.
- Double Key DES was invented later followed by Triple DES.
- Double DES uses the following method:

$$E(k_2, E(k_1, m))$$

- In theory, double DES multiply the difficulty of breaking the code.
- Merkle and Hellman double DES proved that using double 56bits key results in similar result as if we were using single 57bits key.
- Therefore, DES adds no security.

TRIPLE DES

- The cipher texted is produced using 3DES as follows:
$$E(k_3, E(k_2, E(k_1, m))$$
- The key produced is roughly equivalent to 112-bit key.
- Two-Key 3DES is invented using the following formula:
$$E(k_1, D(k_2, E(k_1, m))$$
- The key produced is roughly equivalent to 80-bit key.
- A comparison between all DES is available in the book in page 97
- Breaking double DES requires 2000 years
- Breaking a 3DES is significantly longer.
- In 1995, Advanced Encryption Slandered (AES) was introduced.
- In 2001, AES became the standard encryption for all federal government.

Steganography

- an alternative to encryption
- hides existence of message
  - using only a subset of letters/words in a longer message marked in some way
  - using invisible ink
  - hiding in LSB in graphic image or sound file
  - hide in "noise"
- has drawbacks
  - high overhead to hide relatively few info bits
- advantage is can obscure encryption use

ASYMMETRIC ENCRYPTION

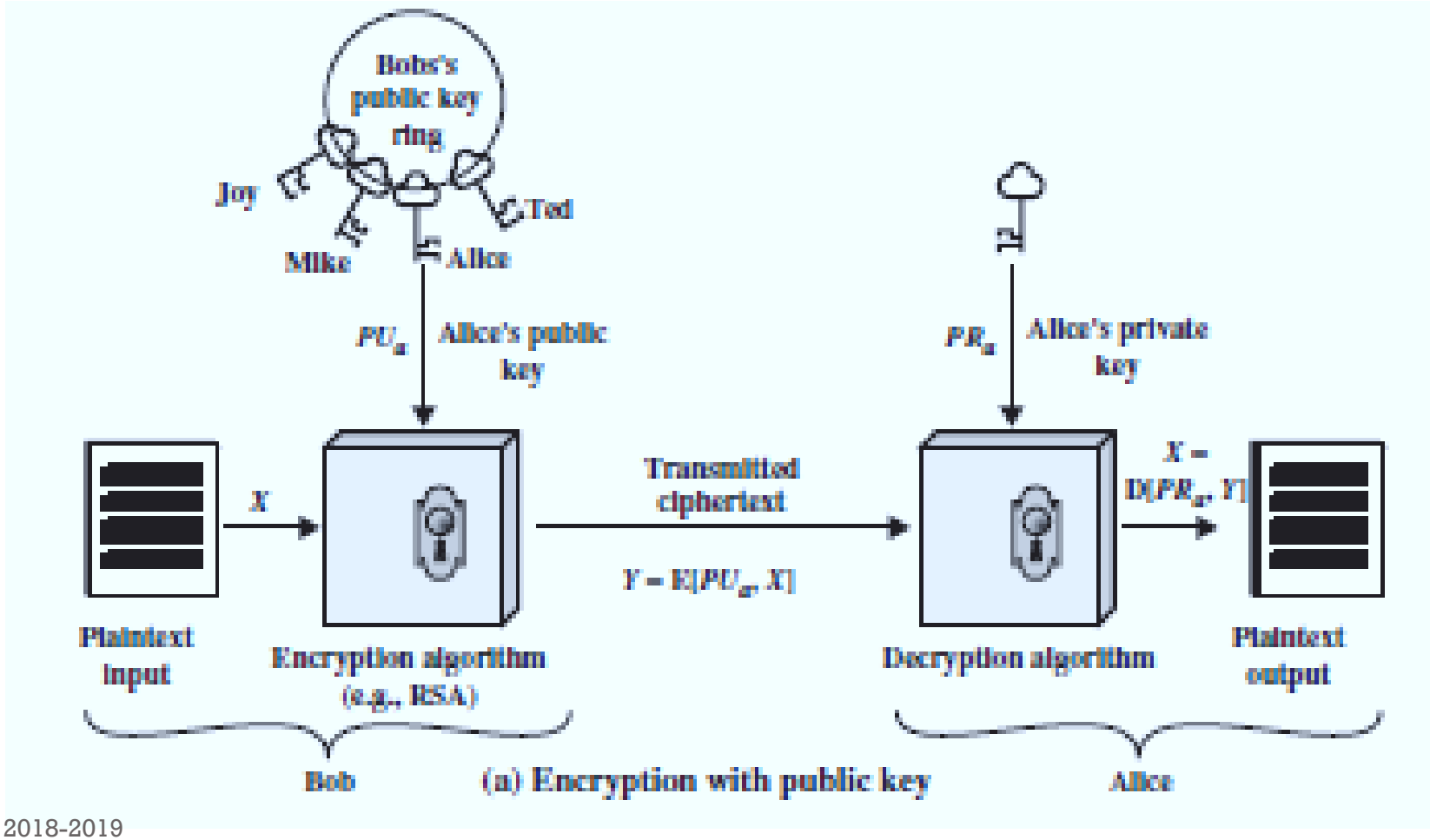


Table 9.3 Applications for Public-Key Cryptosystems

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No

CHARACTERISTICS OF GOOD CIPHERS

In 1949 Claude Shannon [SHA49] proposed characteristics of a good cipher.

- Principle 1
  - The amount of secrecy needed should determine the amount of labor appropriate for encryption and decryption.
- Principle 2
  - The set of keys and the enciphering algorithm should be free from complexity.

CHARACTERISTICS OF GOOD CIPHERS

- Principle 3
  - The implementation of the process should be as simple as possible.
- Principle 4
  - Errors in ciphering should not propagate and cause corruption of further information in the message.
- Principle 5
  - The size of the enciphered text should be no longer than the text of the original message.

KEY ENCRYPTION

Encrypting a plaintext P using cryptosystem E and encryption key K is denoted by:

$$C = E(K,P)$$

The reverse process is denoted by:

$$P = D(K,C) = D(K,E(K,P))$$

This process assume symmetric cryptosystem, where a single-key is used for encryption and decryption.

In an asymmetric cryptosystem, the encryption process is denoted by:

$$C = E(K_E,P)$$

While the decryption process is denoted by:

$$P = D(K_D,C) = D(K_D,E(K_E,P))$$

SYMMETRIC & ASYMMETRIC ENC. SYS.

Symmetric Enc. System

- AKA Private Key or Secrete Key
- Key must remain secrete to ensure authenticity for the source and the content.
- Exchanging Keys is an issue as the number of users increase. For n users:

$$\frac{n(n-1)}{2}$$

Keys required.

Key Distribution is an issue.

Asymmetric Enc. System

- AKA Public key
- Keys are produced together or one is derived from the other one mathematically.
- Key management excel here.

- When keys compromised, a key management is a major issue.

STREAM VS BLOCK CIPHERS

Stream cipher

- Message is encrypted in bits or bytes
- Usable for real time applications.

Block cipher

- Message is broken into fixed size blocks and each block is encrypted.
- Padding is used for short blocks

	Stream	**Block**
Speed of transformation	Fast	Slow
Diffusion	Low	High
Error propagation	Low	High
Padding	No	Yes
Immunity to insertion of symbols	No	Yes

EXAMPLE: DES VS 3-DES

- **Key:** CS433
- **PlainText:** Welcome to Computer Security
- **Ciphertext:**

DES: ja/0eNmjsInpv1rAFJjuKlRtbmtxaYANfaBXVraBiIcEZ/yw8JIpjw==

3-DES: Qe7nFov0rjeDn2RgeEcGDlOWuTZ4JcdAAzxeAuIz5sR6RVNhTafN8A==

AES: RdTue9H1jjUAm7tCGEvCoOXB7J2LokJ9sacjNmI6NGk=

https://www.tools4noobs.com/online_tools/encrypt/

http://aesencryption.net/

KEY MANAGEMENT

- public-key encryption helps address key distribution problems
- have two aspects of this:
  - distribution of public keys
  - use of public-key encryption to distribute secret keys

DISTRIBUTION OF PUBLIC KEYS

- can be considered as using one of:
  - public announcement
  - publicly available directory
  - public-key authority
  - public-key certificates

PUBLIC ANNOUNCEMENT

- users distribute public keys to recipients or broadcast to community at large
  - eg. append PGP keys to email messages or post to news groups or email list
- major weakness is forgery
  - anyone can create a key claiming to be someone else and broadcast it
  - until forgery is discovered can masquerade as claimed user

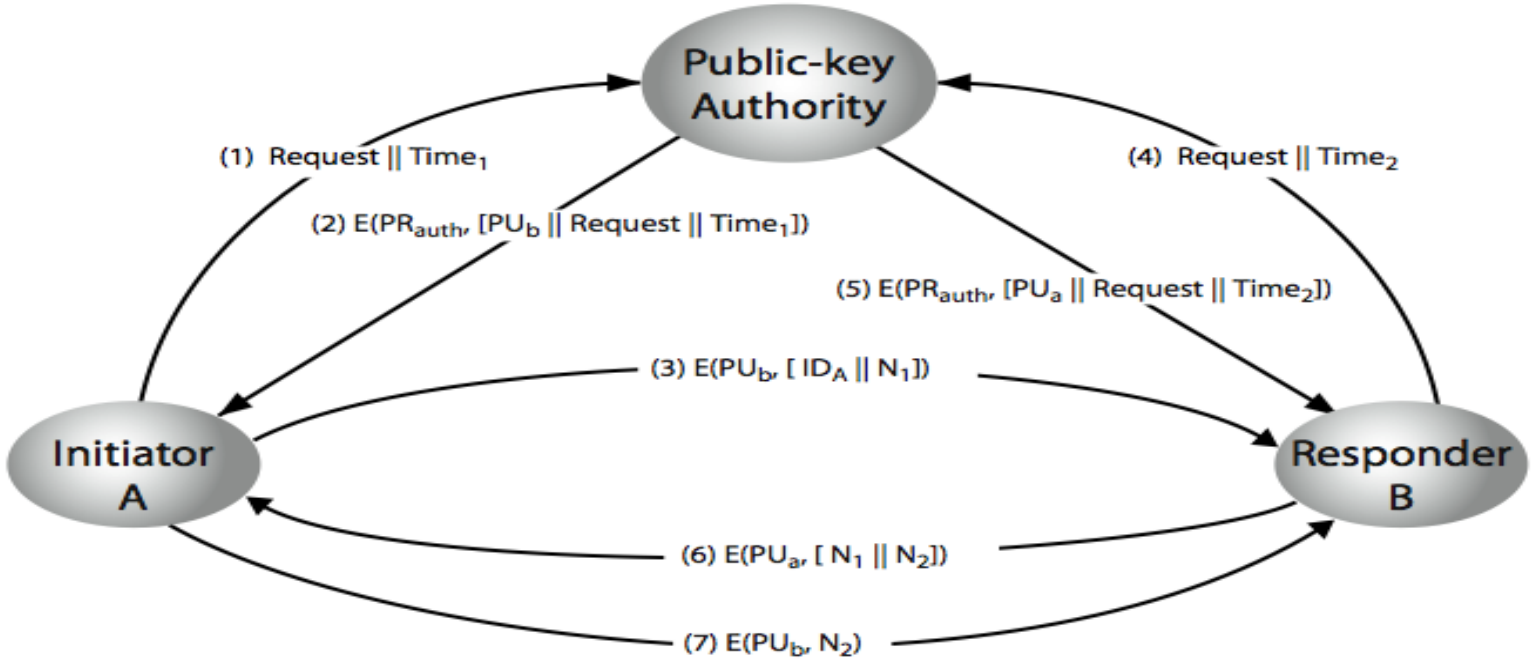
PUBLICLY AVAILABLE DIRECTORY

- can obtain greater security by registering keys with a public directory
- directory must be trusted with properties:
  - contains {name,public-key} entries
  - participants register securely with directory
  - participants can replace key at any time
  - directory is periodically published
  - directory can be accessed electronically
- still vulnerable to tampering or forgery

PUBLIC-KEY AUTHORITY

- improve security by tightening control over distribution of keys from directory
- has properties of directory
- and requires users to know public key for the directory
- then users interact with directory to obtain any desired public key securely
  - does require real-time access to directory when keys are needed

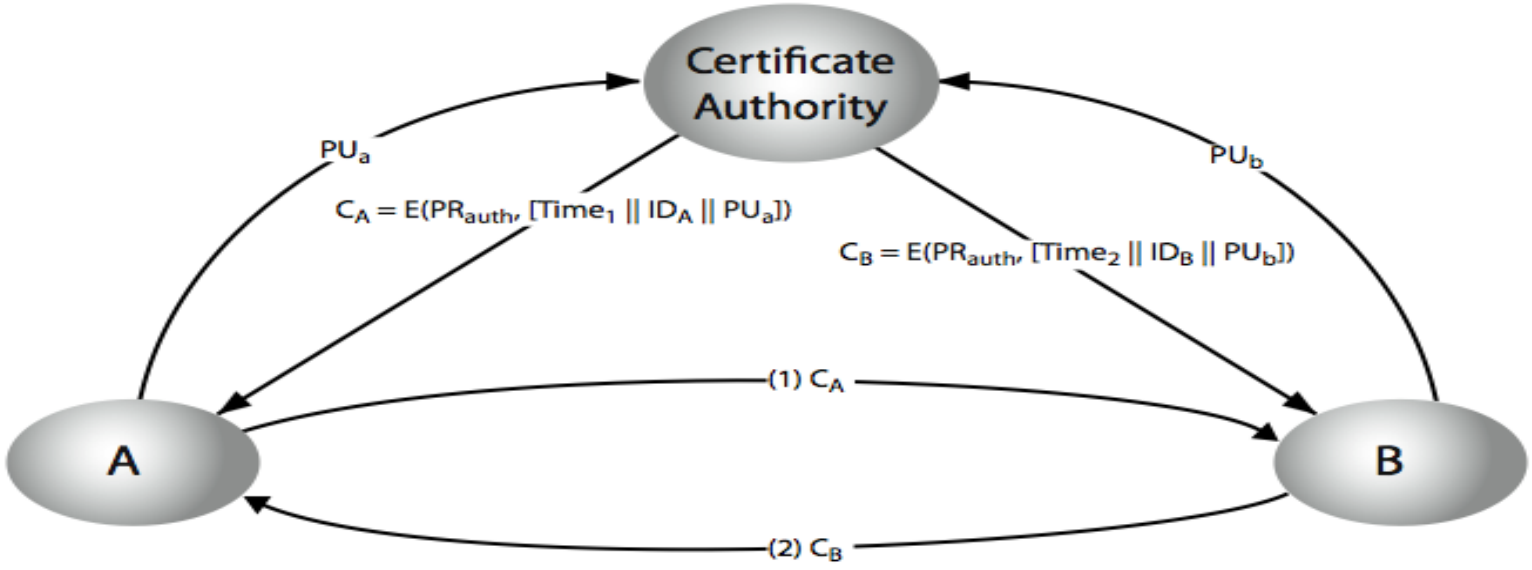
PUBLIC-KEY AUTHORITY



PUBLIC-KEY CERTIFICATES

- certificates allow key exchange without real-time access to public-key authority
- a certificate binds **identity** to **public key**
  - usually with other info such as period of validity, rights of use etc
- with all contents **signed** by a trusted Public-Key or Certificate Authority (CA)
- can be verified by anyone who knows the public-key authorities public-key

PUBLIC-KEY CERTIFICATES



PUBLIC-KEY DISTRIBUTION OF SECRET KEYS

- use previous methods to obtain public-key
- can use for secrecy or authentication
- but public-key algorithms are slow
- so usually want to use private-key encryption to protect message contents
- hence need a session key
- have several alternatives for negotiating a suitable session

Thank you

Programs & Operating system security

Computer Security – IT424

Assoc. Prof. Noha Hikal

Program Security

Topics:

- Programming errors with security implications: buffer overflows, incomplete access control
- Malicious code: viruses, worms, Trojan horses
- Program development controls against malicious code and vulnerabilities: software engineering principles and practices
- Controls to protect against program flaws in execution: operating system support and administrative controls

Program/Code Security (cont.)

- **Protecting programs is at the heart of computer security because they constitute so much of a computing system s.a. (**OS, Device drivers, Network infrastructure, DBMS, … etc)

- **Two important questions:**

1. How do we keep programs free from flaws?
2. How do we protect computing resources against programs that contain flaws?

Bug, Error, Fault, and Failure

- **Bug** in software is a term that can mean many different things depending on context. For example, it can be a mistake in interpreting a requirement or a syntax error in a piece of code.
- **Error** is a human mistake in performing some software activity that may lead to a **fault** in a computer program.
- A **fault** may cause a **failure** (which is a departure from the system's required behavior).
- a fault is an inside view of the system, seen by the developers, whereas a failure is an outside view seen by the user.

Program security flaws

Unfortunately, we do not have techniques to eliminate or address all program security flaws because:

• Program controls apply at the level of the individual program and programmer. When we test a system, we try to make sure that the functionality prescribed in the requirements is implemented in the code (i.e we take a "should do" checklist and verify). However, security is also about preventing certain actions: a "shouldn't do" list.

•Programming and software engineering techniques change and evolve far more rapidly than do computer security techniques.

Types of Flaws

- **Intentional**
  - Malicious
  - Non-malicious

- **Inadvertent (Unintentional)**
  - validation error.
  - domain error: controlled access to data.
  - serialization and aliasing: program flow order.
  - inadequate identification and authentication.
  - boundary condition violation.
  - other exploitable logic errors.

Non-malicious Program Errors

- **Buffer Overflows**

All program and data elements are in memory during execution, sharing space with the operating system, other code, and resident routines. Therefore, the effect of the overflows data depends on where it is go in the memory; It may affect the user data, user code, system data, or system code.

Non-malicious Program Errors (cont.)

- **Incomplete Mediation**

Attackers are exploiting it to cause security problems. eg: changing the price value in a URL.

•http://www.things.com/order.asp?custID=101&part=555A&qy=20&price=10&ship=boat&shipcost=5&total=**205**

•

http://www.things.com/order.asp?custID=101&part=555A&qy=20&price=10&ship=boat&shipcost=5&total=**25**

Non-malicious Program Errors (cont.)

- **Time-of-Check to Time-of-Use Errors**

A delay between the time the access was checked and the time the result of the check was used, a change occurred, invalidating the result of the check.

Example: Changing the file name that checked for a deletion access.

Malicious Code

- By themselves, programs are seldom security threats.
- Most users don't know which programs in addition to their programs are executed or modified, and which files are changed because they usually do not see computer data directly, **malicious people** can make programs serve as vehicles to access and change data and other programs.

Malicious Code (cont.)

When a user install a SW package, or a plug-in from the Internet, or download an application s.a. a Java applet or an ActiveX control while viewing a web site; a lot of programs and data are transferred and a lot of modifications may be made to your existing files, all occurring without your explicit consent or knowledge.

→ **We should worry about malicious code.**

Malicious Code (cont.)

- Malicious code can do anything like any other program (writing data, stopping a running program, erasing a file, …etc).
- Malicious code can do nothing at all right now; it can be planted to lie dormant, undetected, until some event triggers the code to act s.a.:
  - A time or date trigger
  - An interval (eg. after 30 minutes)
  - An event (eg. when a particular program is executed)
  - A condition (eg. when communication occurs on a NW)
  - A count (eg. the fifth time something happens)
  - A random situatio
  - etc.

  OR some combination of these
  OR different things each time

- Malicious code runs under the user's authority, but without the user's permission or even knowledge.

Kinds of Malicious Code

- **Virus** is a program that can replicate itself and pass on malicious code to other non-malicious programs by modifying them.
  - **Transient:** runs when its attached program executes and terminates when its attached program ends.
  - **Resident**: locates itself in memory so that it can remain active even after its attached program ends.
- **Trojan Horse** is malicious code that, in addition to its primary effect, has a second, non-obvious malicious effect.

Kinds of Malicious Code (cont.)

- **Logic Bomb** is a class of malicious code that "detonates" or goes off when a specified condition occurs.
- **Trapdoor/backdoor** is a feature in a program by which someone can access the program other than by the obvious, direct call, perhaps with special privileges.
- **worm** is a program that spreads copies of itself through a network.

A worm vs. a virus

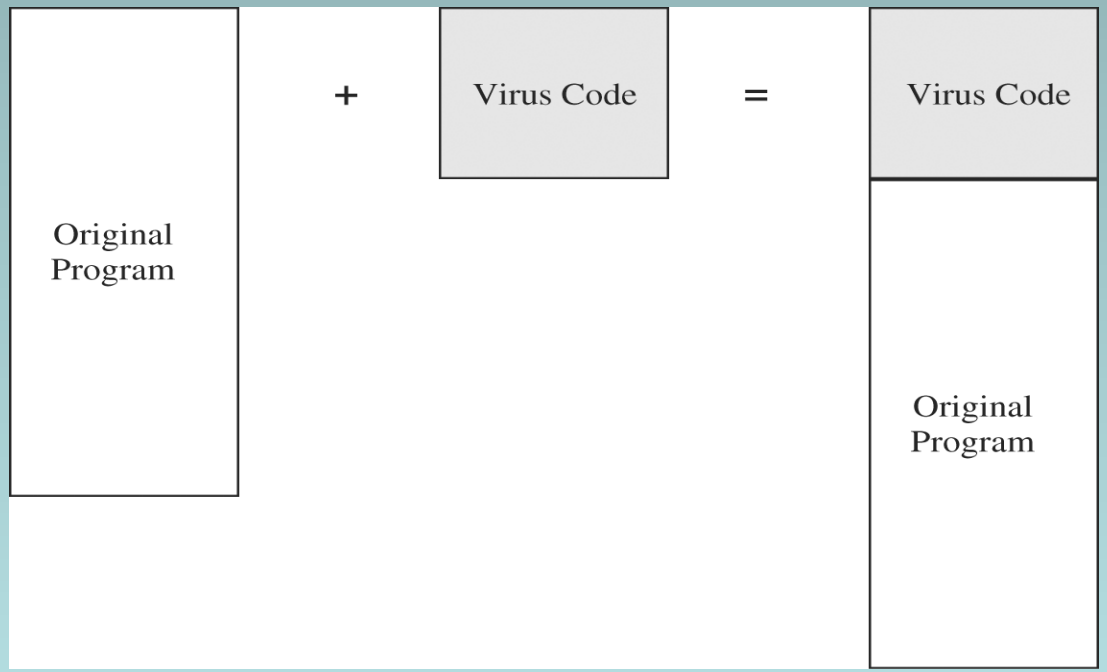
- A worm operates through networks, and a virus can spread through any medium.
- A worm spreads copies of itself as a stand-alone program, whereas the virus spreads copies of itself as a program that attaches to or embeds in other programs.

Summary of Malicious Code

Virus	Attaches itself to program and propagates copies of itself to other programs
Trojan horse	Contains unexpected, additional functionality
Logicbomb	Triggers action when condition occurs
Time bomb	bombTriggers action when specified time occursTrapdoorAllows
Trapdoor	Allows unauthorized access to functionality
Worm	Propagates copies of itself through a network
Rabbit	Replicates itself without limit to exhaust resources

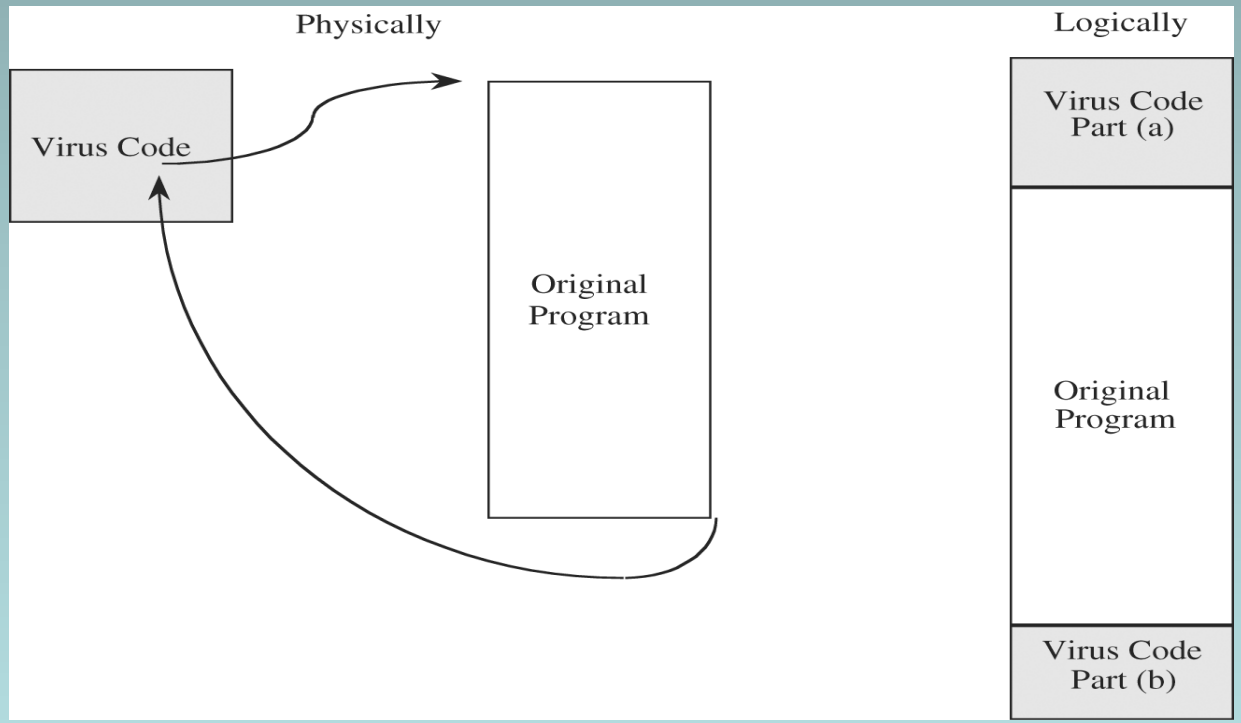
How Viruses Attach

- **Appended Viruses:** A program virus attaches itself to a program; then, whenever the program is run, the virus is activated. This kind of attachment is usually easy to program.



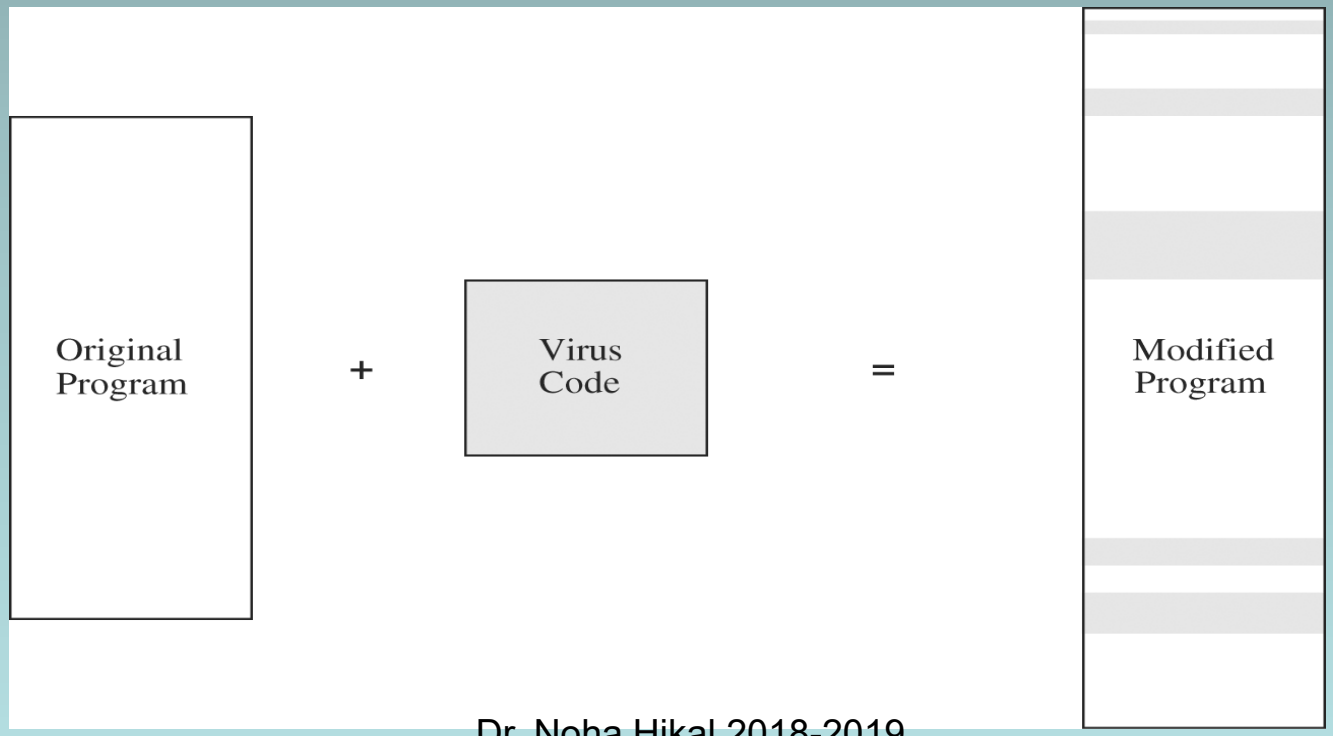
How Viruses Attach (cont.)

- **Viruses That Surround a Program: A** virus that runs the original program but has control before and after its execution.



How Viruses Attach (cont.)

- **Integrated Viruses and Replacements:** A virus replaces some of its target, integrating itself into the original code of the target.



Document Viruses

Document virus is implemented within a formatted document, such as a written document, a database, a slide presentation, a picture, or a spreadsheet.

These documents are highly structured files that contain both data and commands. The commands are part of a rich programming language, including macros, variables and procedures, file accesses, and even system calls.

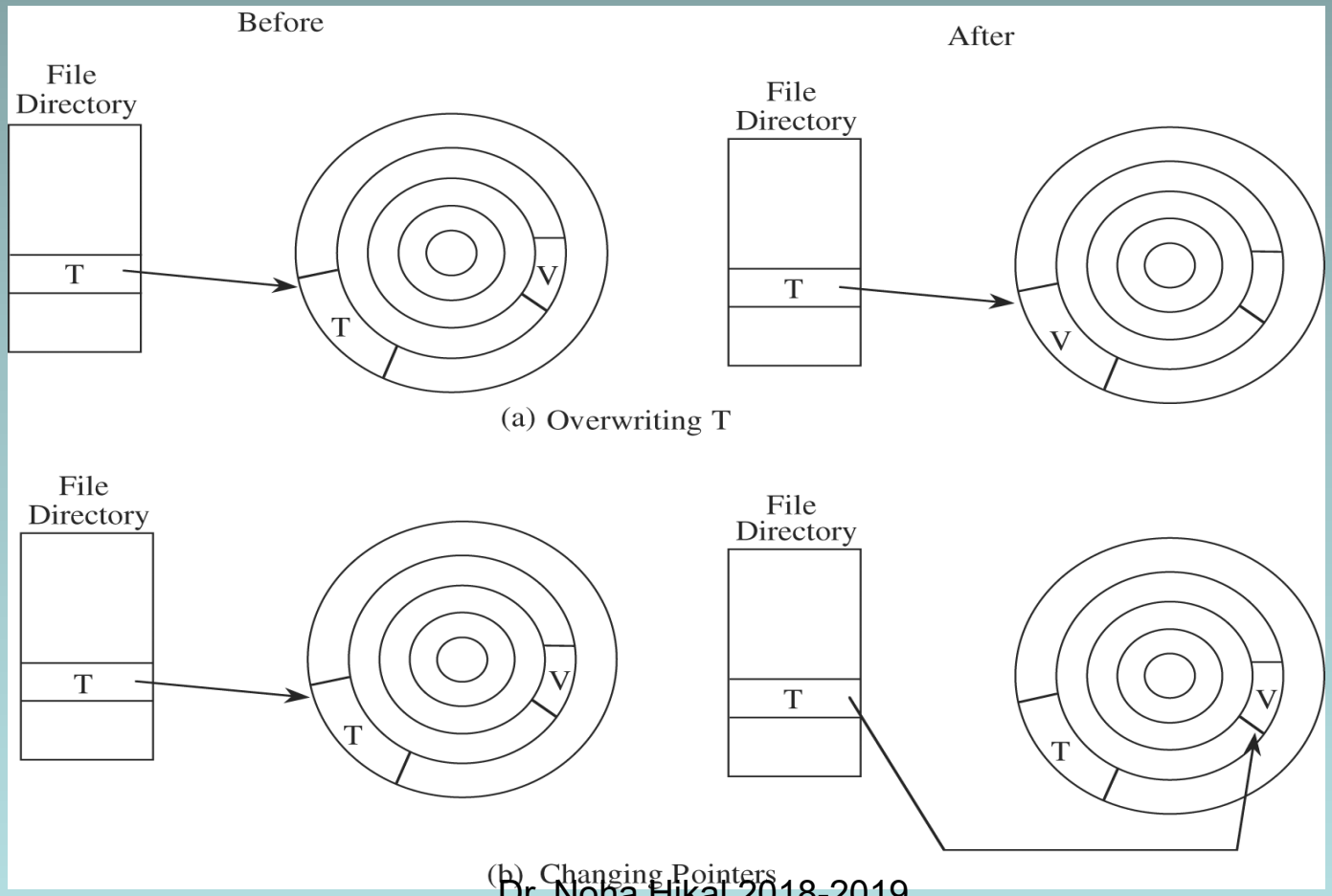
Document Viruses (cont.)

The writer of a document virus uses any of the features of the programming language to perform malicious actions.

The ordinary user usually sees only the content of the document (its text or data), so the virus writer simply includes the virus in the commands part of the document, as in the integrated program virus.

How Viruses Gain Control

- The virus (V) has to be invoked instead of the target (T).



Homes for Viruses

The virus writer may find these qualities appealing in a virus:

- It is hard to detect.
- It is not easily destroyed or deactivated.
- It spreads infection widely.
- It can reinfect its home program or other programs.
- It is easy to create.
- It is machine independent and operating system independent.

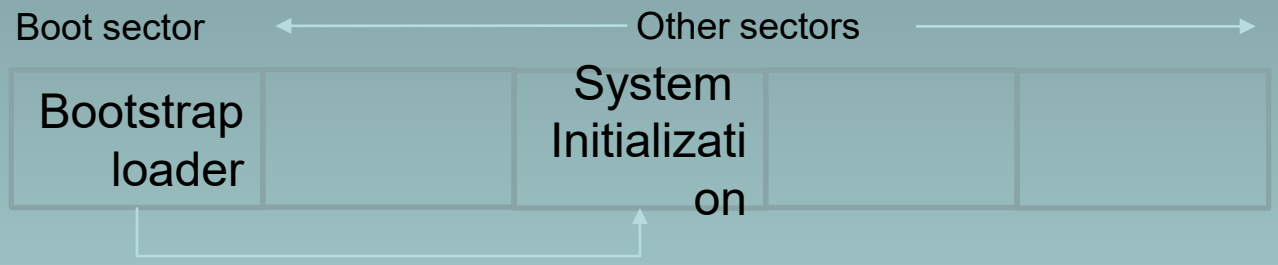
***** Few viruses meet all these criteria**

Homes for Viruses (cont.)

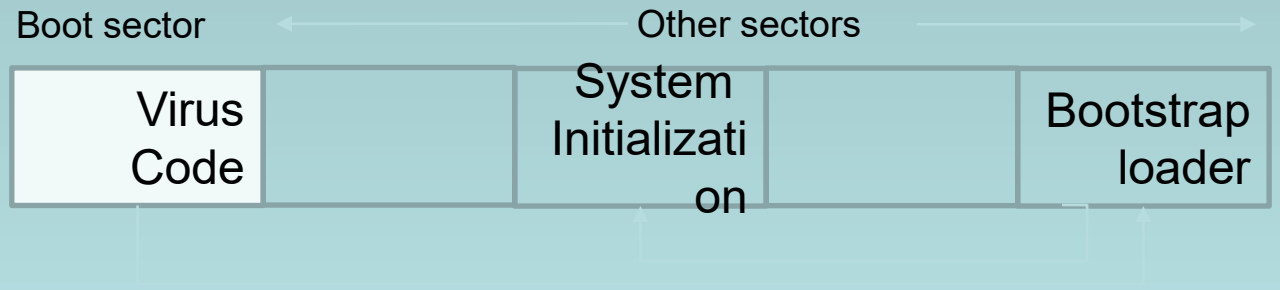
- **One-Time Execution:** The majority of viruses today execute only once, spreading their infection and causing their effect in that one execution.
- **Boot Sector Viruses:** Change control on the OS instructions, when a computer is started.
- **Memory-Resident Viruses:** Virus writers also like to attach viruses to resident code because the resident code is activated many times while the machine is running.
- **Other Homes for Viruses:** application programs and libraries.

Boot sector viruses

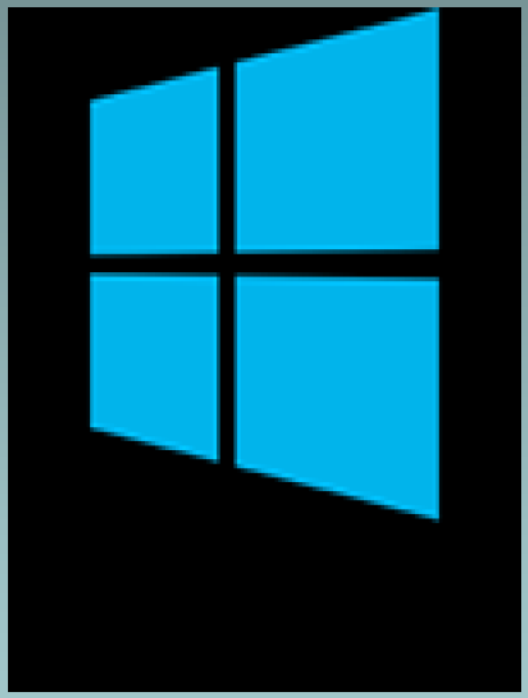
Before Infection



After Infection



- Understanding booting process in computers
- Every time a resident code runs, the virus runs too.
- Once activated, the virus infects uninfected file.
- Also, when activated, removable devices can be infected.
- A virus can infect operating system table of program runs.



Memory-resident viruses

Dr. Noha Hikal 2018-2019

Virus Signatures

A virus cannot be completely invisible. Code must be stored somewhere, and the code must be in memory to execute. Moreover, the virus executes in a particular way, using certain methods to spread. Each of these characteristics yields a telltale pattern, called a **signature.**

Virus scanner looks for the virus signature to detect its existence.

polymorphic virus changes its appearance.

Prevention of Virus Infection

- The only way to prevent the infection of a virus is not to receive executable code from an infected source.

- **Some technique that help:**
  - Use only commercial software acquired from reliable, well-established vendors.
  - Test all new software on an isolated computer.

Prevention of Virus Infection (cont.)

- Open attachments only when you know them to be safe.
- Make a recoverable system image and store it safely.
- Make and retain backup copies of executable system files.
- Use virus detectors (often called virus scanners) regularly and update them daily.

Controls Against Program Threats
Developmental Controls

- Modularity, Encapsulation, and Information Hiding.
- Peer Reviews.
- Independent Testing.
- Good design.
- Configuration Management.
- Proofs of Program Correctness.

Modularity, Encapsulation, and Information Hiding

- Modularization is the process of dividing a task into subtasks on a functional and logical basis.
- **Advantages**: Maintenance, Reuse, Understandability, Correctness, Testing.
- **Characteristics of a Module**: Unity, Smallness, Simplicity, Independence.

Peer Reviews

- Each team member has a clear design document.
- Team members review each others' code.
- All team members recognize that the product belongs to the group.

Independent Testing

- To certify the correctness of a program, not necessarily to assign blame for errors
- Independent test teams are more effective to test a program than the programmer
- Independent testing increases the likelihood that a test will expose the effect of a hidden feature

Configuration Management

- To guard against the inadvertent loss of a version of a program.
- To manage the parallel development of several similar versions of one program.
- To provide facilities for controlled sharing of modules that combine to form one system.
- Security Advantages:
  - Protects against unintentional threats.
  - Guards against malicious ones.

Proofs of Program Correctness

- Program verification is a technique to demonstrate formally the “correctness” of a specific program.
- Program statements are translated into logical statements about the contribution of that statement to the logical flow of the program.

Proofs of Program Correctness (cont.)

- Program correctness proofs are hindered by:
  - Program translation is error prone.
  - The logical engines are slow.
  - Proofs of correctness have not been consistently and successfully applied to large production systems.

AUTHENTICATION & OPERATING SYSTEM SECURITY

IT424- Computer Security

Assoc. Prof. Noha Hikal

OPERATING SYSTEMS

- An operating system allows different users to access different resources in a shared mannar.
- The operating system needs to control this sharing and provide an interface to allow this access
- Identification and authentication are required for this access control

HISTORY

- Operating systems evolved as a way to allow multiple users use the same hardware
- OS makes resources available to users if required by them and permitted by some policy
- OS also protects users from each other
  - Attacks, mistakes, resource overconsumption
- Even for a single-user OS, protecting a user from him/herself is a good thing

PROTECTION IN GENERAL-PURPOSE OS

- **OS system has two goals:**
  - Controlling shared access.
  - Implementing an interface to allow that access.
- **Underneath those goals are support activities:**
  - Identification and authentication
  - Naming
  - Filing objects
  - Scheduling
  - Communication among processes
  - Reusing objects.

*** Each of them has security implications.

PROTECTION IN GENERAL-PURPOSE OS

- Simple supporting a single task at a time.
- Complex supporting multiuser and multitasking.

*** Naturally, security considerations increase as OS become more complex.

PROTECTED OBJECTS

- The rise of multiprogramming meant that several aspects of a computing system required protection:
  - Memory.
  - Sharable I/O devices, such as disks.
  - Serially reusable I/O devices, such as printers and tape drives.
  - Sharable programs and sub-procedures.
  - Networks.
  - Sharable data.

SECURITY METHODS OF OS

- The basis of protection is separation: keeping one user's objects separate from other users. That separation in an OS can occur in several ways:
  - ***Physical separation***, in which different processes use different physical objects. it is easy to implement, but expensive and inefficient
  - ***Temporal separation***, in which processes having different security requirements are executed at different times.

SECURITY METHODS OF OS (CONT.)

- ***Logical separation***, in which users operate under the illusion that no other processes exist, as when an operating system constrains a program's accesses so that the program cannot access objects outside its permitted domain.
- ***Cryptographic separation***, in which processes conceal their data and computations in such a way that they are unintelligible to outside processes. (It is more Complex.)

*** A combination of these separations is possible.

SEPARATION AND SHARING

- Separation is half of the answer. We want to separate users and their objects, but we also want to be able to provide sharing for some of those objects. OS can support separation and sharing in several ways, offering protection at any of several levels.

- **Do not protect**. OS with no protection is appropriate when sensitive procedures are being run at separate times.

SEPARATION AND SHARING (CONT.)

- ***Isolate***. When OS provides isolation, different processes running concurrently are unaware of the presence of each other. Each process has its own address space, files, and other objects.
- ***Share all or share nothing***. With this form of protection, the owner of an object declares it to be public or private.

SEPARATION AND SHARING (CONT.)

Share via access limitation. With protection by access limitation, the OS checks the allowability of each user's potential access to an object. That is, access control is implemented for a specific user and a specific object. The OS acts as a guard between users and objects, ensuring that only authorized accesses occur.

SEPARATION AND SHARING (CONT.)

- ***Share by capabilities***. An extension of limited access sharing, this form of protection allows dynamic creation of sharing rights for objects.
- ***Limit use of an object***. This form of protection limits not just the access to an object but the use made of that object after it has been accessed.

ACCESS CONTROL

In general, access control has three goals:

- Check every access: Else OS might fail to notice that access has been revoked
- Enforce least privilege: Grant program access only to smallest number of objects required to perform a task
  - Access to additional objects might be harmless under normal circumstances, but disastrous in special cases
- Verify acceptable use: Limit types of activity that can be performed on an object
  - E.g., for integrity reasons

USER AUTHENTICATION

- Computer systems often have to identify and authenticate users before authorizing them
  - Identification: Who are you?
  - Authentication: Prove it!
- Identification and authentication is easy among people that know each other
  - For your friends, you do it based on their face or voice
- More difficult for computers to authenticate people sitting in front of them
- Even more difficult for computers to authenticate people accessing them remotely

USER AUTHENTICATION

- OS bases much of its protection on identifying a user of the system.
- Authentication mechanisms use any of three qualities to confirm a user's identity:
  - Something the user knows (Passwords or PIN)
  - Something the user has, (cards or physical keys).
  - Something the user is (biometrics identity),
  - Something about the user's (Location or time)

*** Two or more forms can be combined for more solid authentication.

PASSWORDS AS AUTHENTICATORS

- PWs are widely used for authentication.
- Other information can be used in addition to PW, s.a. access time and terminal.
- PWs suffer from some difficulties of use:
  - **Loss**. Depending on how the passwords are implemented, it is possible that no one will be able to replace a lost or forgotten password. If the user loses the password, a new one must be assigned.
  - **Use**. Supplying a password for each access to a file can be inconvenient and time consuming.

PASSWORDS AS AUTHENTICATORS (CONT.)

- **Disclosure**. If a password is disclosed to an unauthorized individual, the file becomes immediately accessible. If the user then changes the password, all other legitimate users must be informed of the new password.

- **Revocation**. To revoke one user's access right to a file, someone must change the password, thereby causing the same problems as disclosure.

ATTACKS ON PASSWORDS

- Here are some ways you might be able to determine a user's password:
  - Try all possible passwords.
  - Try frequently used passwords.
  - Try passwords likely for the user.
  - Search for the system list of passwords.
  - Ask the user.
  - Rainbow table
  - Social engineering

EXHAUSTIVE ATTACK

- **Exhaustive** or **brute force attack**, the attacker tries all possible passwords.
- The number of possible passwords depends on the implementation of the system.
- If passwords are words consisting of the 26 characters A..Z and can be of any length from 1 to 8 characters, then the system as a whole has **$26^1 + 26^2 + \ldots + 26^8$ = about $5 * 10^{12}$** possible passwords.

ENCRYPTED PASSWORD FILE

- **To foil an intruder seeking passwords in plain sight, encrypt them:**
- Conventional encryption, but still some user(s) can decrypt them.

- Hash function/one-way ciphers.
  A problem: what if two users choose the same pw and one of them see the hash value of the other user?
  A solution: add to the pw a unique generated number to the pw before hashing.

GUIDELINES FOR PW SELECTION CRITERIA

- Use characters other than just AZ.
- Choose long passwords.
- Avoid actual names or words.
- Choose an unlikely password.
- Change the password regularly.
- Don't write it down.
- Don't tell anyone else.

BIOMETRICS: AUTHENTICATION NOT USING PASSWORDS

- Biometrics are biological authenticators, based on some physical characteristic of the human body:
  - Fingerprints
  - Hand and face geometry
  - Retina
  - Voice
  - Handwriting
  - Walking pattern
- Advantage over pw: a biometric cannot be lost, stolen, forgotten, forged and is always available.

PROBLEMS WITH BIOMETRICS

- Biometrics are relatively new, and some people find their use intrusive.
- Biometric recognition devices are costly.
- All biometric readers use sampling and establish a threshold for when a match is close enough to accept.
- Biometrics can become a single point of failure.

PROBLEMS WITH BIOMETRICS (CONT.)

- Although equipment is improving, there are still false readings.
- The speed at which a recognition must be done limits accuracy.
- Although we like to think of biometrics as unique parts of an individual, forgeries are possible.

AUTHENTICATION PROCEDURE

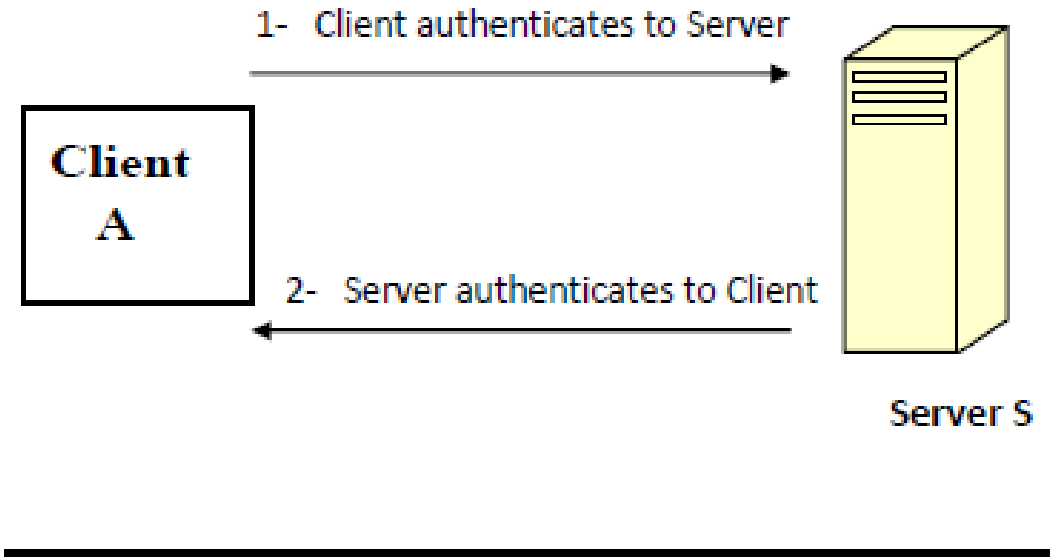


Figure 2: Two –Party Authentication

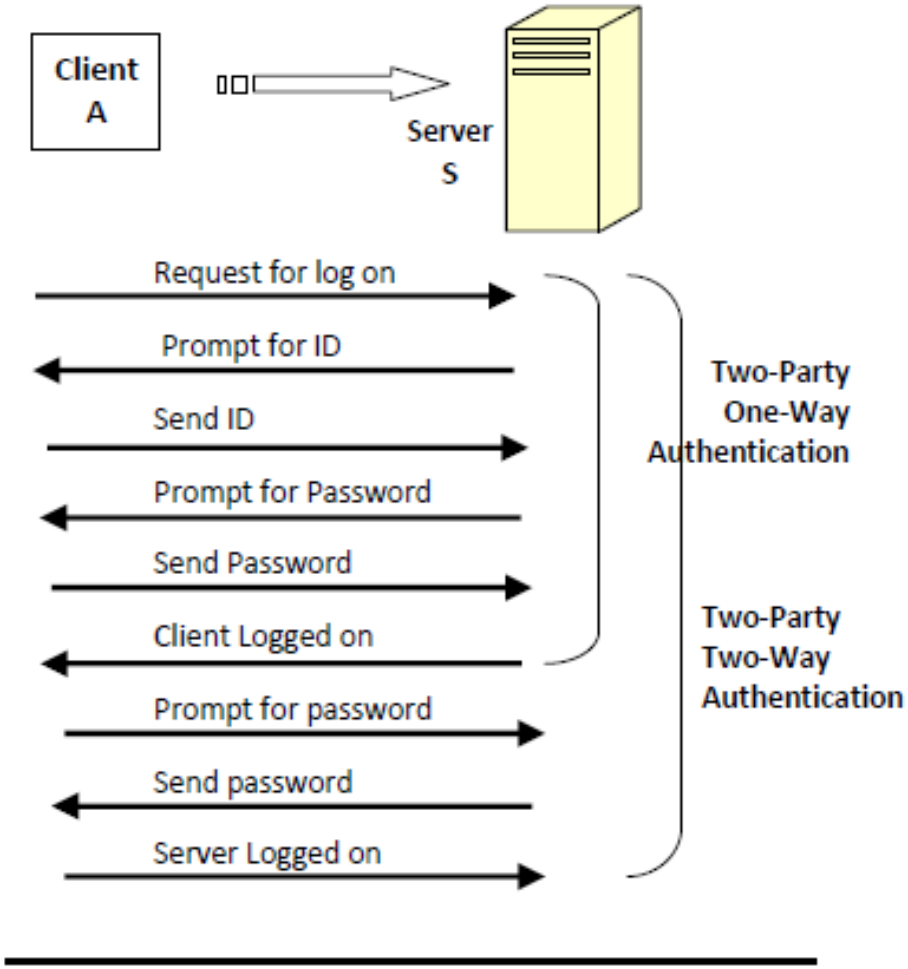


Figure 3: Two-Party One-Way Authentication and Two-Way Authentication.

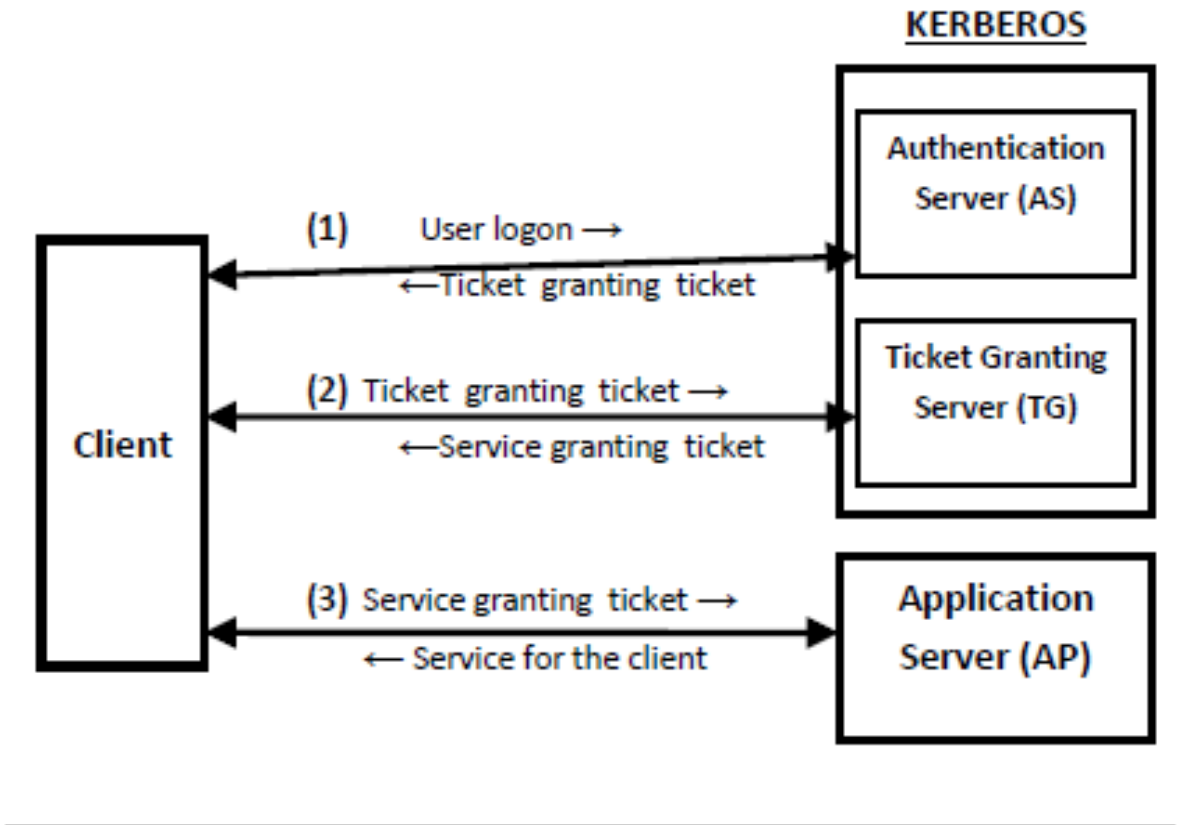


Figure 4: KERBEROS system

Network Security

IT424-Computer Security

Assoc. Prof. Noha Hikal

NETWORK SECURITY PROBLEMS

- Sharing: more users and computing systems have access
- Complexity: a network may combine two or more dissimilar operating systems with mechanisms for interhost connection
- Unknown perimeter: host may be node on multiple networks; unknown or uncontrolled group of possibly malicious users; new hosts may be added to net at any time; user is unaware of potential connections from users of other networks
- File may pass through many host machines to get to user
- Administrator of one network domain has no control over other network domains
- There may be many paths from one host to another
- Network users seldom have control over routing of messages

Additional NETWORK SECURITY PROBLEMS

- Privacy: concealing sensitive data more difficult with many unknown users on a network
- Integrity: many nodes and many users, so risk of data corruption is higher. Data corruption includes:
  - modification of messages
  - insertion of bogus messages
  - deletion of messages
  - replay of messages
  - reordering of messages
- Authentication: hard to assure identity of user on a remote system. Network may not be able to trust authenticity of the hosts themselves.
- User can't control routing through hostile or insecure nodes

NETWORK SECURITY SOLUTIONS

- Include traditional host solutions
- In addition to physical, administrative, legal...
- Encryption
- Authentication
- Access Controls

Reconnaissance

A clever attacker investigates and plans before acting. He begins preparation by finding out as much as possible about the target. Some techniques to collect information are:

- **Port Scan**
- **Social Engineering**
- **Intelligence**
- **OS and Application Fingerprinting**
- **Bulletin Boards and Chats**
- **Availability of Documentation**

The best defense against reconnaissance is silence. Give out as little information about your site as possible, whether by humans or machines.

Threats in Transit

- **Eavesdrop** implies overhearing without expending any extra effort. E.g. a system administrator is eavesdropping by monitoring all traffic passing through a node. The administrator might have a legitimate purpose, s.a. watching for inappropriate use of resources **or** passing files to an enemy from a military computer.

Threats in Transit (cont.)

- **Wiretap** means intercepting communications through some effort.
  - **Passive wiretapping** is just "listening," much like eavesdropping.
  - **Active wiretapping** means injecting something into the communication, s.a. changing the message.

*** Wiretapping works differently depending on the communication medium used.

Threats in Transit (cont.)

- **Protocol Flaws**
- **Impersonation**
- **Spoofing**
  - **masquerade:** one host pretends to be another.
  - **Session hijacking** is intercepting and carrying on a session begun by another entity.
  - **Man-in-the-Middle Attack** in which one entity intrudes between two others.

Message Confidentiality Threats

- Eavesdropping and impersonation attacks can lead to a confidentiality or integrity failure.
- Other vulnerabilities that can affect confidentiality are misdelivery, exposure, and traffic flow analysis.

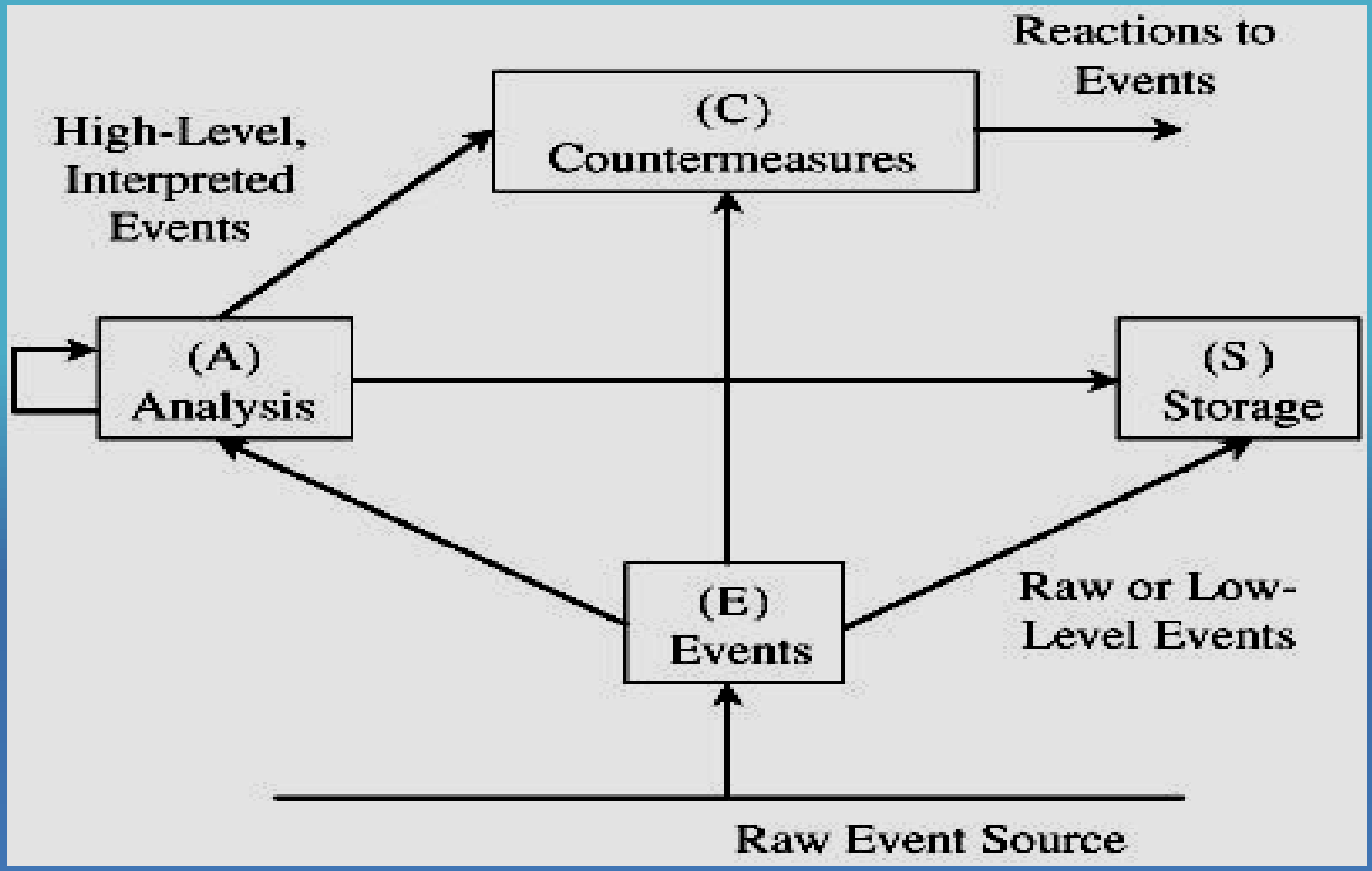
Message Integrity Threats

- Falsification of Messages
- Noise
- Format Failures
- Malformed Packets
- Protocol Failures and Implementation Flaws

Intrusion detection

- Is the process of detecting and identifying unauthorized or unusual activity on the system, Such a scheme requires specification of what constitute an undesirable activity and a means of automatically detecting such activity as it occurs.

- An **intrusion detection system (IDS)** is a device, typically another separate computer, that monitors activity to identify malicious or suspicious events.
- An IDS receives raw inputs from sensors. It saves those inputs, analyzes them, and takes some controlling action.



Common components of an intrusion detection framework

IDSs perform a variety of functions:

- monitoring users and system activity
- auditing system configuration for vulnerabilities and misconfigurations
- assessing the integrity of critical system and data files
- recognizing known attack patterns in system activity
- identifying abnormal activity through statistical analysis
- managing audit trails and highlighting user violation of policy or normal activity
- correcting system configuration errors
- installing and operating traps to record information about intruders

Types of IDSs

- **Signature-based intrusion detection systems** perform simple pattern-matching and report situations that match a pattern corresponding to a known attack type.

- **Heuristic intrusion detection systems, " anomaly based"** build a model of acceptable behavior and flag exceptions to that model; for the future, the administrator can mark a flagged behavior as acceptable or not so that the heuristic IDS will now treat that previously unclassified behavior as how it was classified.

Heuristic IDS

Denning's IDS model

Format	Example
Subject	Jhon (action initiator)
Action	File write
Object	Employee record file
Exception condition	No
Resources usage	10
Time stamp	0800 02112010

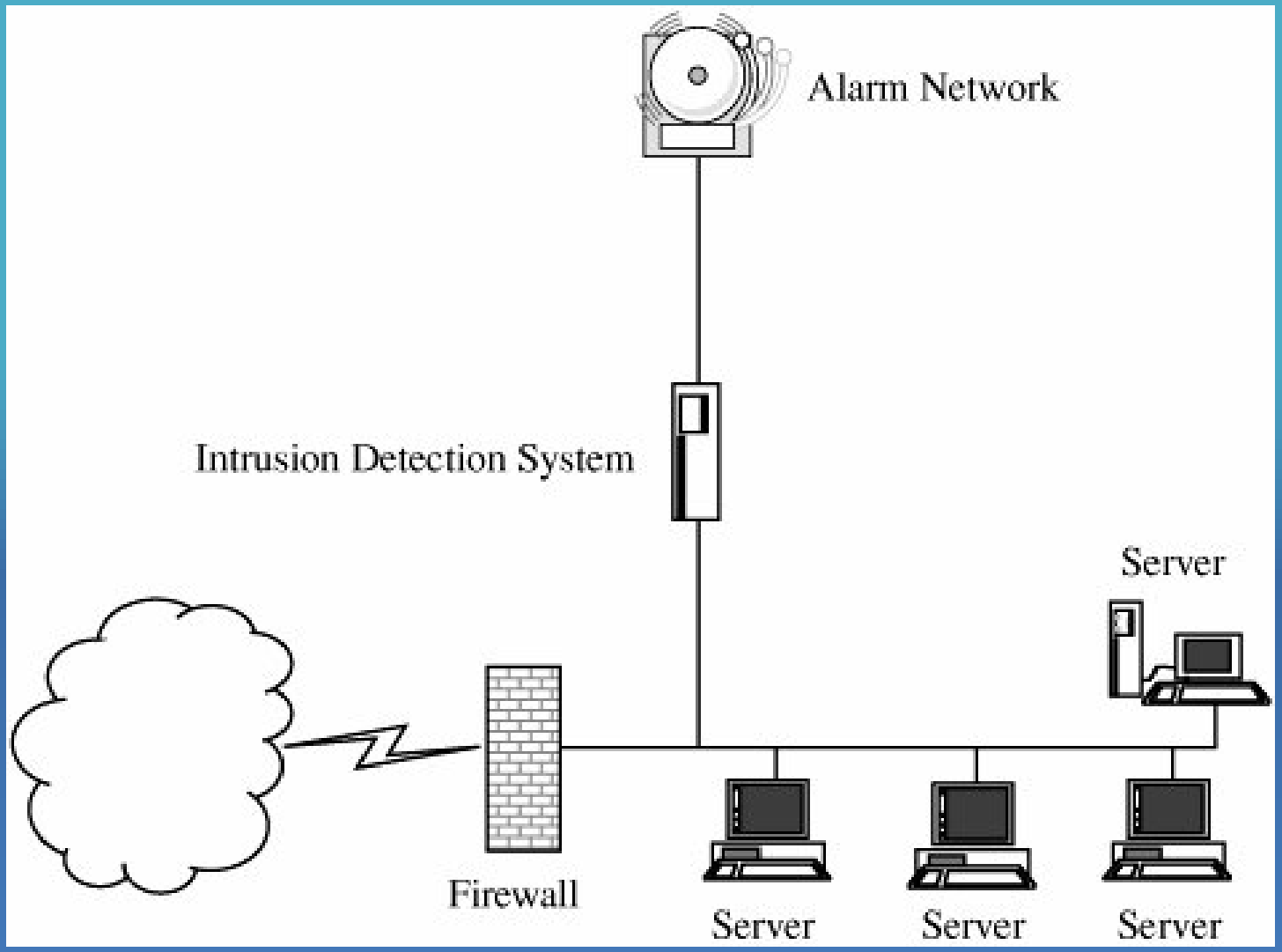
- ***Profiles*** characterize the normal behavior of the subjects on an objects so the IDS can detect the abnormal.
- ***Three candidates profiles:***

1-logon and session activity

2-command or program usage

3-file access activity

Anomaly record is created when abnormal behavior appears, it consists of "*event, time, profile*"



Stealth Mode IDS Connected to Two Networks.

Intrusion detection devices can be :

-A network-based IDS is a stand-alone device attached to the network to monitor traffic throughout that network.

-A host-based IDS runs on a single workstation or client or host, to protect that one host.

Early intrusion detection systems [DEN87b, LUN90a, FOX90, LIE89])

Other IDS Types

- The ***tripwire program*** *[KIM98] is the most well* known software (or static data) comparison program.
- System vulnerability scanners, such as ***ISS Scanner or Nessus****, can be run against a network.* They check for known vulnerabilities and report flaws found.
- A ***honey pot*** is a faux environment intended to trick an attacker. It can be considered an IDS, in the sense that the honeypot may record an intruder's actions and even attempt to trace who the attacker is from actions, packet data, or connections.

Firewalls

- A firewall is a device that filters all traffic between a protected or "inside" network and a less trustworthy or "outside" network.
- It needs a security policy.
- Working principles:
  - Default permit -Default deny

Designing a firewall must satisfy:

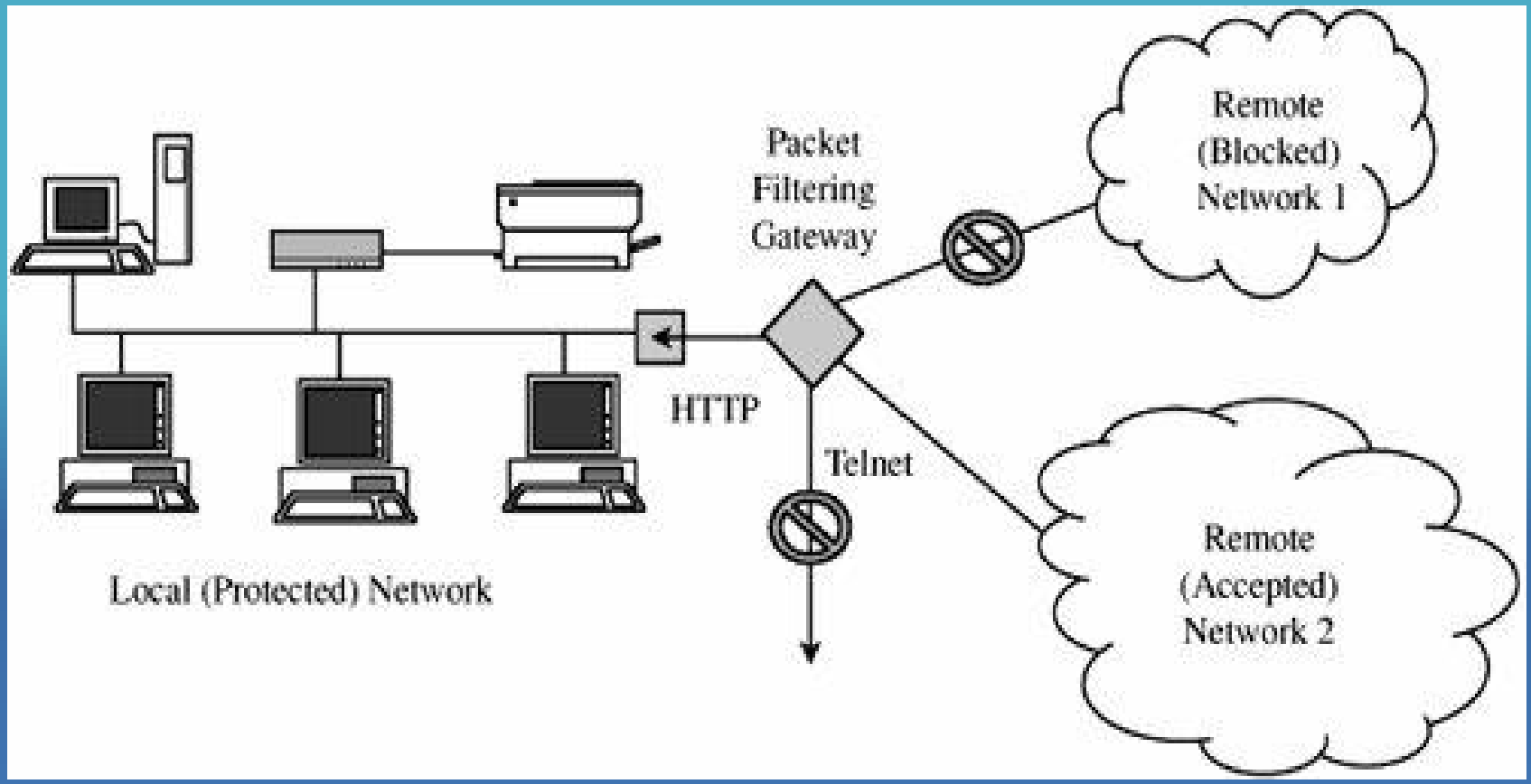
- Always invoked
- Well isolated (separate computer)
- Tamperproof
- Small and simple enough for rigorous analysis

Types of firewalls:

- Packet filtering gateways or screening routers
- Stateful inspection firewalls
- Application proxies
- Guards
- Personal firewalls

1- Packet Filtering Gateway "screening router "

- The simplest, and in some situations, the most effective type of firewall.
- Controls access to packets on the basis of packet address (source or destination) or specific transport protocol type (HTTP)
- It doesn't see inside the packet, the packet is accepted or rejected based on its IP address.
- **Disadvantage:** filtering rules need to be very detailed



Packet Filter Blocking Addresses and Protocols.

2-Stateful Inspection Firewall

- Works on packets one at a time, accepting or rejecting each packet and moving on to the next.
- The attackers commonly break the attack into multiple packets each have very short lengthes that the firewall can’t be able to detect the signature of the attack.

3-Application proxy
"bastion host"

- It intrudes in the middle of a protocol exchange, seeming like a destination in communication with the sender that is outside the firewall, and seeming like the sender in communication with the real destination on the inside. The proxy in the middle has the opportunity to screen the transfers
- the proxy interprets the protocol stream to an application, to control actions through the firewall on the basis of things visible *within the protocol, not just on external header data.*

When a proxy gateway can be used?

- A company wants to set up an online price list so that outsiders can see the products and prices offered. It wants to be sure that

(a) no outsider can change the prices or product list

(b) outsiders can access only the price list, not any of the more sensitive files stored inside.

- A school wants to allow its students to retrieve any information from World Wide Web resources on the Internet. To help provide efficient service, the school wants to know what sites have been visited and what files from those sites have been fetched. particularly popular files will be cached locally.

4-Guard

- It acts like proxy firewall, it receives protocol data units, interprets them, and passes through the same or different protocol data units that achieve either the same result or a modified result.
  - The guard decides what services to perform on the user's behalf in accordance with its available knowledge
- We call the proxy a guard. Since the security policy implemented by the guard is somewhat more complex than the action of a proxy.
- The guard's code is also more complex and therefore more exposed to error..

When a guard firewall can be used?

- A university wants to allow its students to use e-mail up to a limit of so many messages or so many characters of e-mail in the last so many days. Although this result could be achieved by modifying e-mail handlers, it is more easily done by monitoring the common point through which all e-mail flows, the mail transfer protocol.

- A school wants its students to be able to access the World Wide Web but, because of the slow speed of its connection to the web, it will allow only so many characters per downloaded image (that is, allowing text mode and simple graphics, but disallowing complex graphics, animation, music, or the like).

5-Personal firewalls

- **A personal firewall is an application program that runs on a workstation to block unwanted traffic, usually from the network.**

- **Suitable for home users, individual workers, and small businesses use cable modems or DSL connections with unlimited.**

- The personal firewall is configured to enforce some policy. For example:

  - The user may decide that certain sites are highly trustworthy, but most other sites are not.

  - The user defines a policy permitting download of code, unrestricted data sharing,…

- Personal firewalls can also generate logs of accesses, which can be useful to examine in case

  something harmful does slip through the firewall.

- Commercial implementations of personal firewalls include Norton Personal Firewall from Symantec, McAfee Personal Firewall, and Zone Alarm from Zone Labs (now owned by CheckPoint).

Packet Filtering	Stateful Inspection	Application Proxy	Guard	Personal Firewall
Simplest	More complex	Even more complex	Most complex	Similar to packet filtering firewall
Sees only addresses and service protocol type	Can see either addresses or data	Sees full data portion of packet	Sees full text of communication	Can see full data portion of packet
Auditing difficult	Auditing possible	Can audit activity	Can audit activity	Canand usually doesaudit activity
Screens based on connection rules	Screens based on information across packetsin either header or data field	Screens based on behavior of proxies	Screens based on interpretation of message content	Typically, screens based on information in a single packet, using header or data
Complex addressing rules can make configuration tricky	Usually preconfigured to detect certain attack signatures	Simple proxies can substitute for complex addressing rules	Complex guard functionality can limit assurance	Usually starts in "deny all inbound" mode, to which user adds trusted addresses as they appear

Different firewall configurations

Figure 7-38. Firewall with Screening Router.

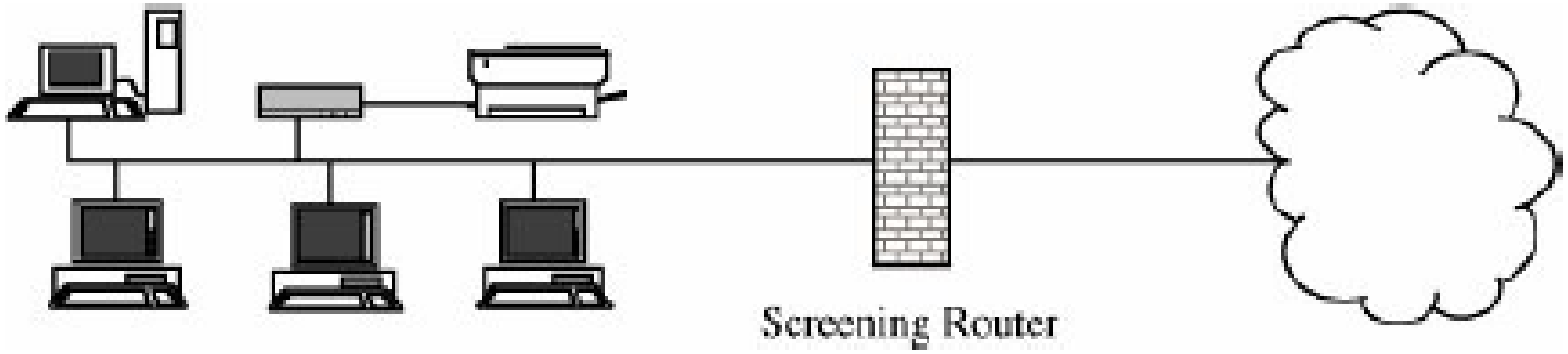


Figure 7-39. Firewall on Separate LAN.

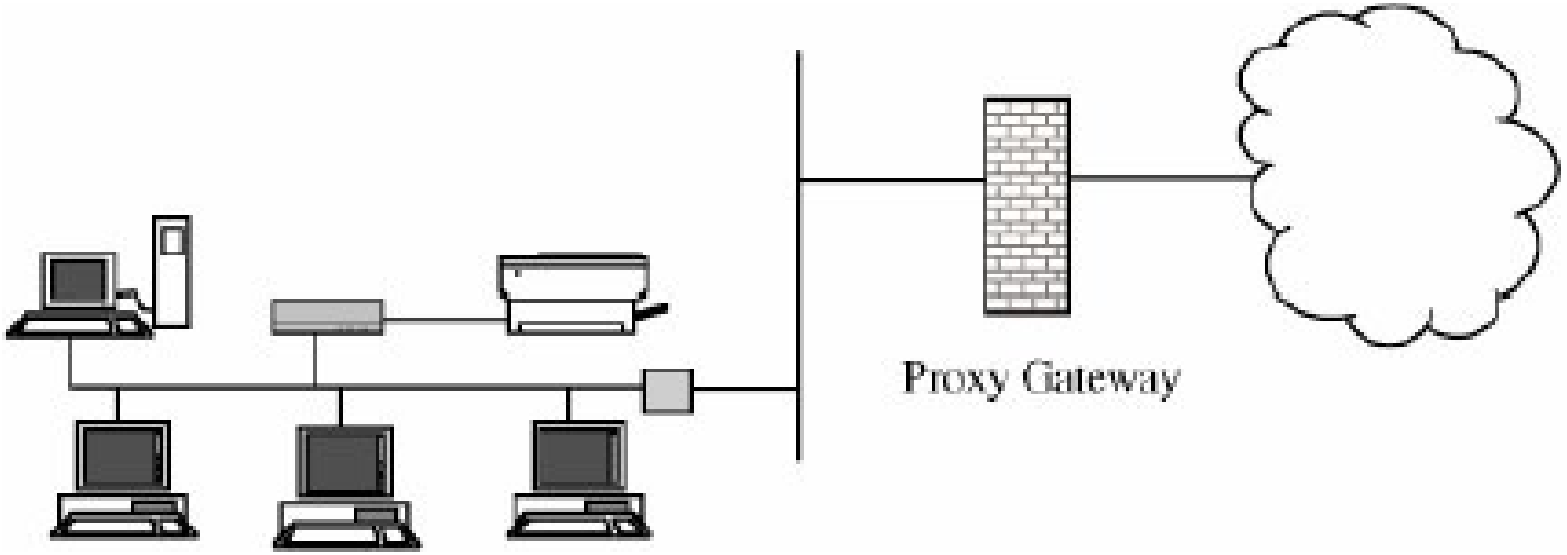
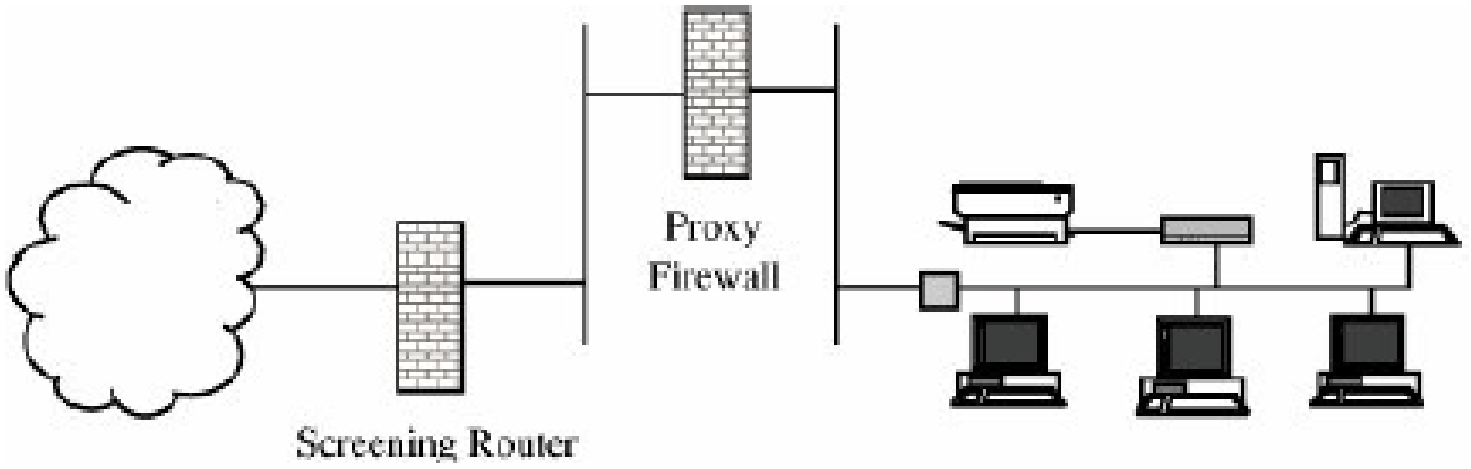


Figure 7-40. Firewall with Proxy and Screening Router.



What Firewalls Can and Cannot Block?

- firewalls are not complete solutions to all computer security problems. A firewall protects only the perimeter of its environment against attacks from outsiders who want to execute code or access data on the machines in the protected environment.