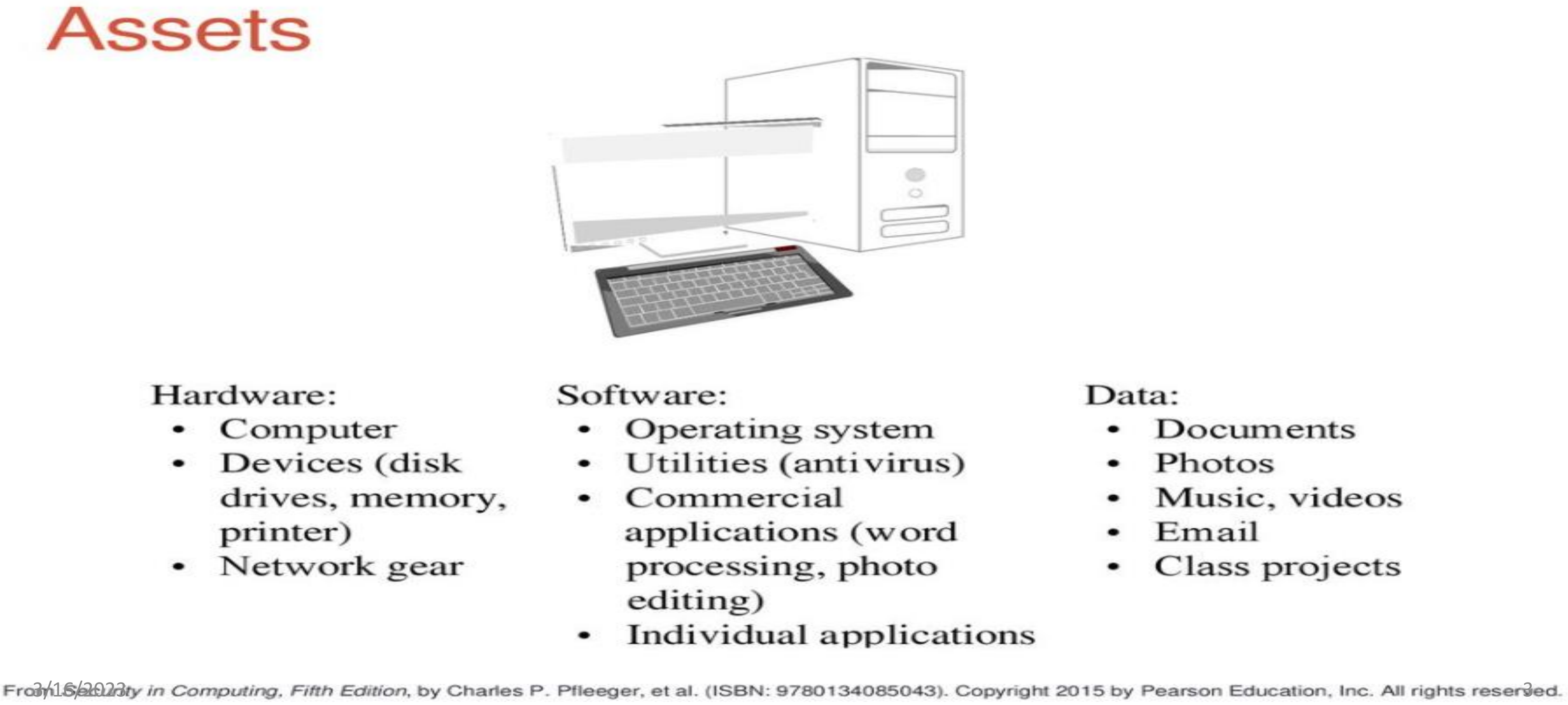
19CSE311 – Computer Security

Syllabus

- **Unit 1** Basics of Computer Security: Overview – Definition of terms – Security goals – Shortcomings – Attack and defense – Malicious code – Worms – Intruders – Error detection and correction Encryption and Cryptography: Ciphers and codes – Public key algorithms – Key distribution – Digital signatures.
- **Unit 2** Security Services: Authentication and Key Exchange Protocols - Access control matrix – User authentication – Directory authentication service – Diffie-Hellman key exchange – Kerberos.
- **Unit 3** System security and Security models: Disaster recovery - Protection policies. E-mail Security: Pretty good privacy - Database Security: Integrity constraints - Multi-phase commit protocols - Networks Security: Threats in networks - DS authentication -Web and Electronic Commerce: Secure socket layer - Client-side certificates - Trusted Systems : Memory protection.

The Items You Value - Assets



Assets

Hardware:

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

Software:

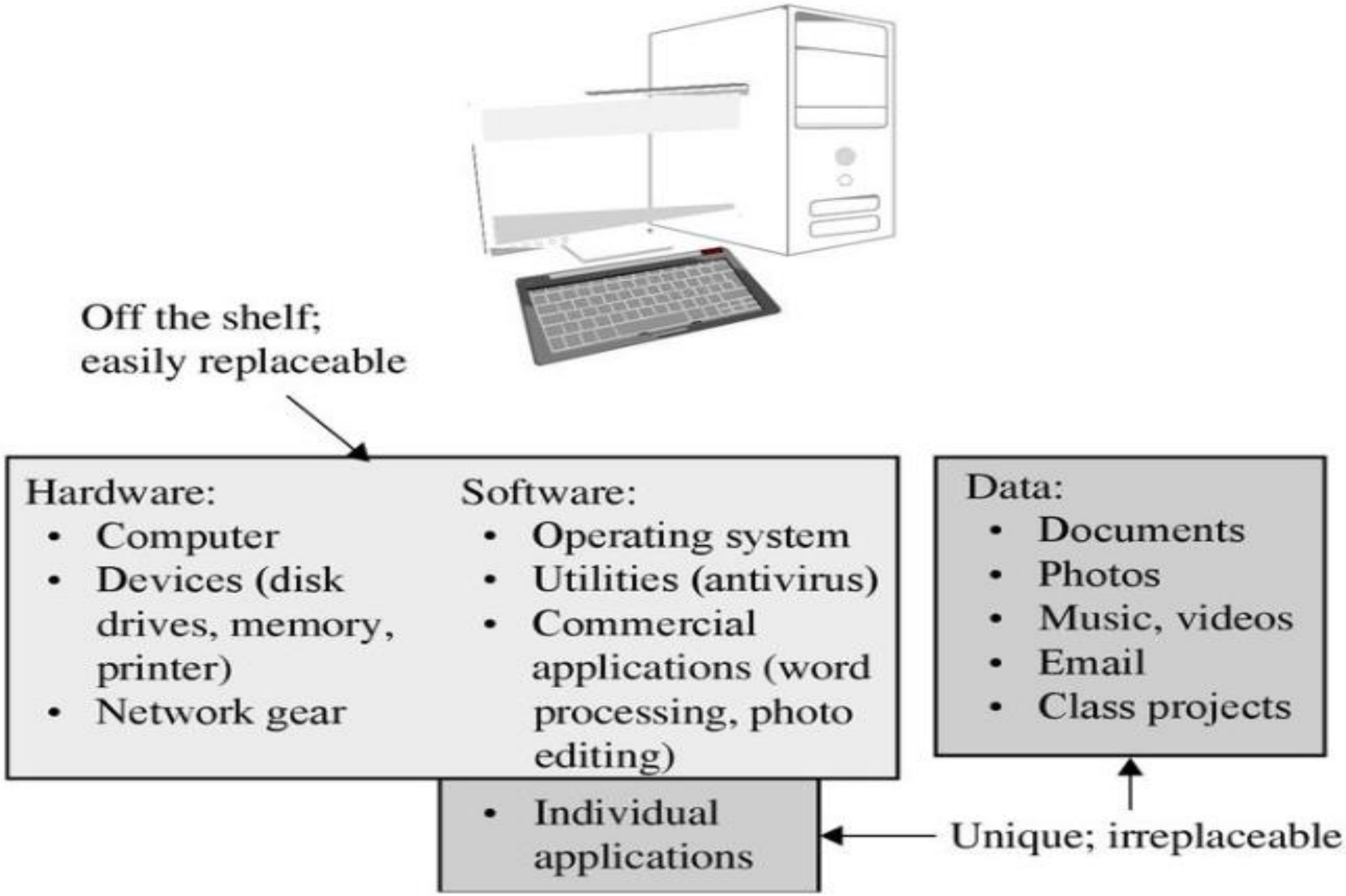
- Operating system
- Utilities (antivirus)
- Commercial applications (word processing, photo editing)
- Individual applications

Data:

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

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Value of Assets



Off the shelf; easily replaceable

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

Unique; irreplaceable

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Basic Terms

- **Vulnerability**
- **Threat**
- **Attack**
- **Countermeasure or control**

Vulnerabilities

- **Vulnerability** is a weakness in the security system

(i.e., in procedures, design, or implementation), that might be exploited to cause loss or harm.

Threats

- **Threat** to a computing system is a set of circumstances that has the *potential to cause loss damage or harm.*
- a potential violation of security

Attacks

- A human (criminal) who exploits a vulnerability perpetrates an **attack** on the system.

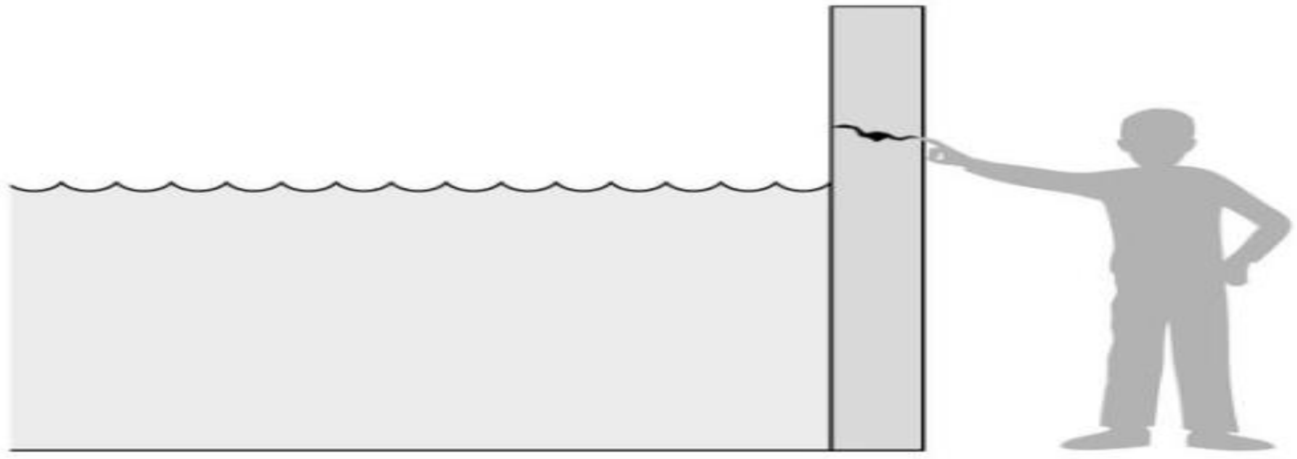
Controls

How do we address these problems?

- We use a **control** as a protective measure.
- That is, a control is an action, device, procedure, or technique that removes or reduces a vulnerability.

Relationship among threats, controls, and vulnerabilities:

- A threat is blocked by control of a vulnerability.
- To devise controls, we must know as much about threats as possible.
- The fact that the violation might occur means that the actions that might cause it should be guarder against.

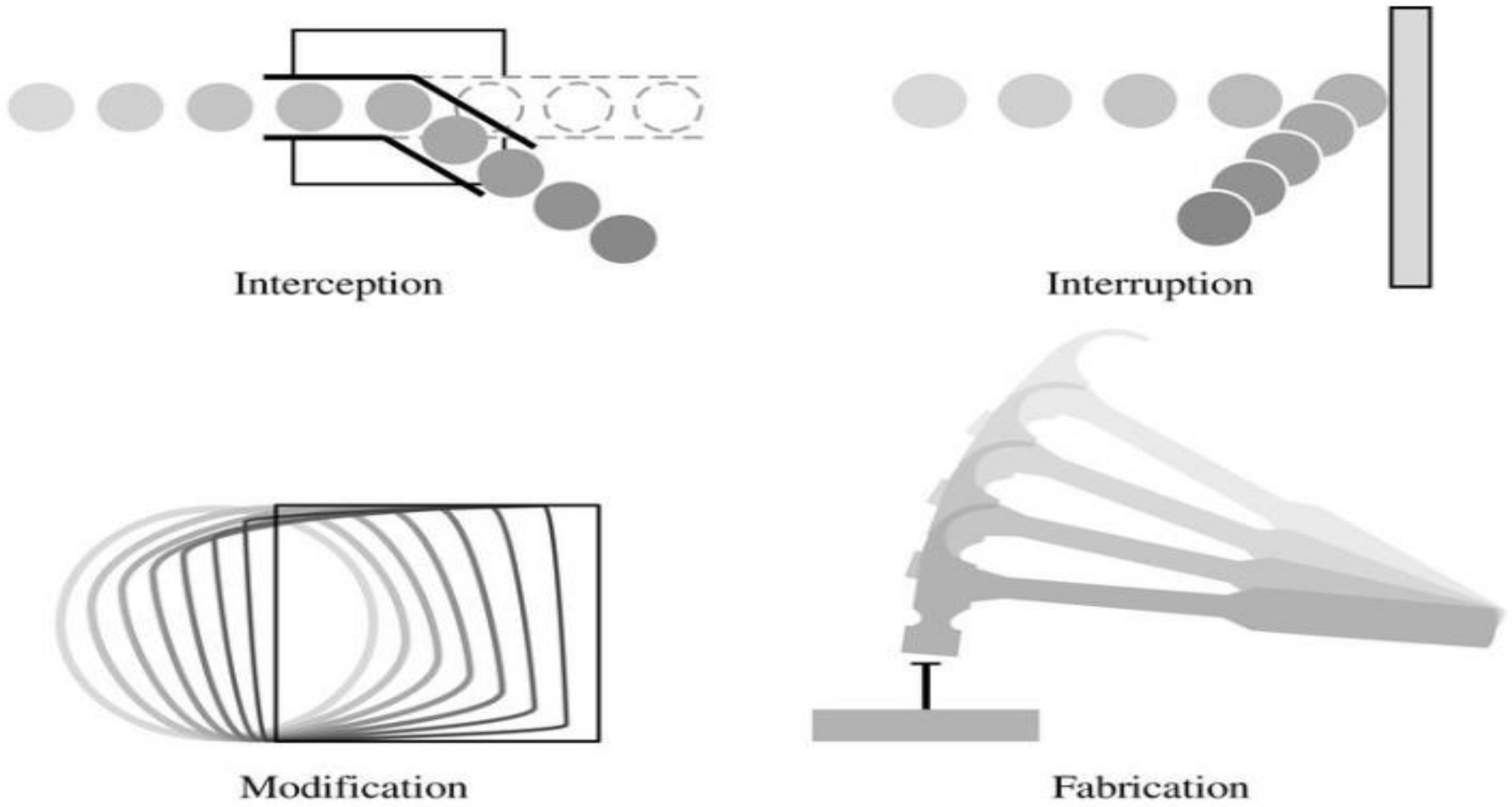


- The water is the threat, the crack the vulnerability, and the finger the control (for now).

C-I-A Triad Confidentiality Integrity Availability

- Sometimes two other desirable characteristics:
- Authentication: the process or action of proving or showing something to be true, genuine, or valid.
- Nonrepudiation is the assurance that someone cannot deny something. i.e. nonrepudiation refers to the ability to ensure that a party to a contract or a communication cannot deny the authenticity of their signature on a document or the sending of a message that they originated

Types of Harm



Interception

Interruption

Modification

Fabrication

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Threats

- In an **interception** means that some unauthorized party has gained access to an asset.
- In an **interruption**, an asset of the system becomes lost, unavailable, or unusable.
- If an unauthorized party not only accesses but **tampers** (forges) with an asset, the threat is a **modification**.
- Finally, an unauthorized party might create a **fabrication** of *counterfeit* objects on a computing system.

Confidentiality

- 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.
  - An unauthorized person learns the existence of a piece of data.

Integrity

- Integrity of an item can be preserved by following conditions:
  - Precise
  - Accurate
  - Unmodified
  - Modified only in acceptable ways
  - Modified only by authorized people
  - Modified only by authorized processes
  - Consistent
  - Internally consistent
  - Meaningful and usable

Integrity

- Integrity can also mean two or more of these properties .
- Welke and Mayfield recognize three particular aspects of integrity:
  - Authorized actions
  - Separation and protection of resources
  - Error detection and correction.

Availability

- 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 waiting mode, it has a bounded waiting time.
  - The service is completed in an acceptable period of time.

Availability

- We can construct an overall description of availability by combining these goals. Following are some criteria to define availability:
  - There is a timely response to our request.
  - Resources are allocated fairly so that some requesters are not favoured over others.
  - Concurrency is controlled; that is, simultaneous access, deadlock management, and exclusive access are supported as required.
  - The service or system follows fault tolerance.
  - The service or system can be used easily and in the way it was intended to be used.

Availability

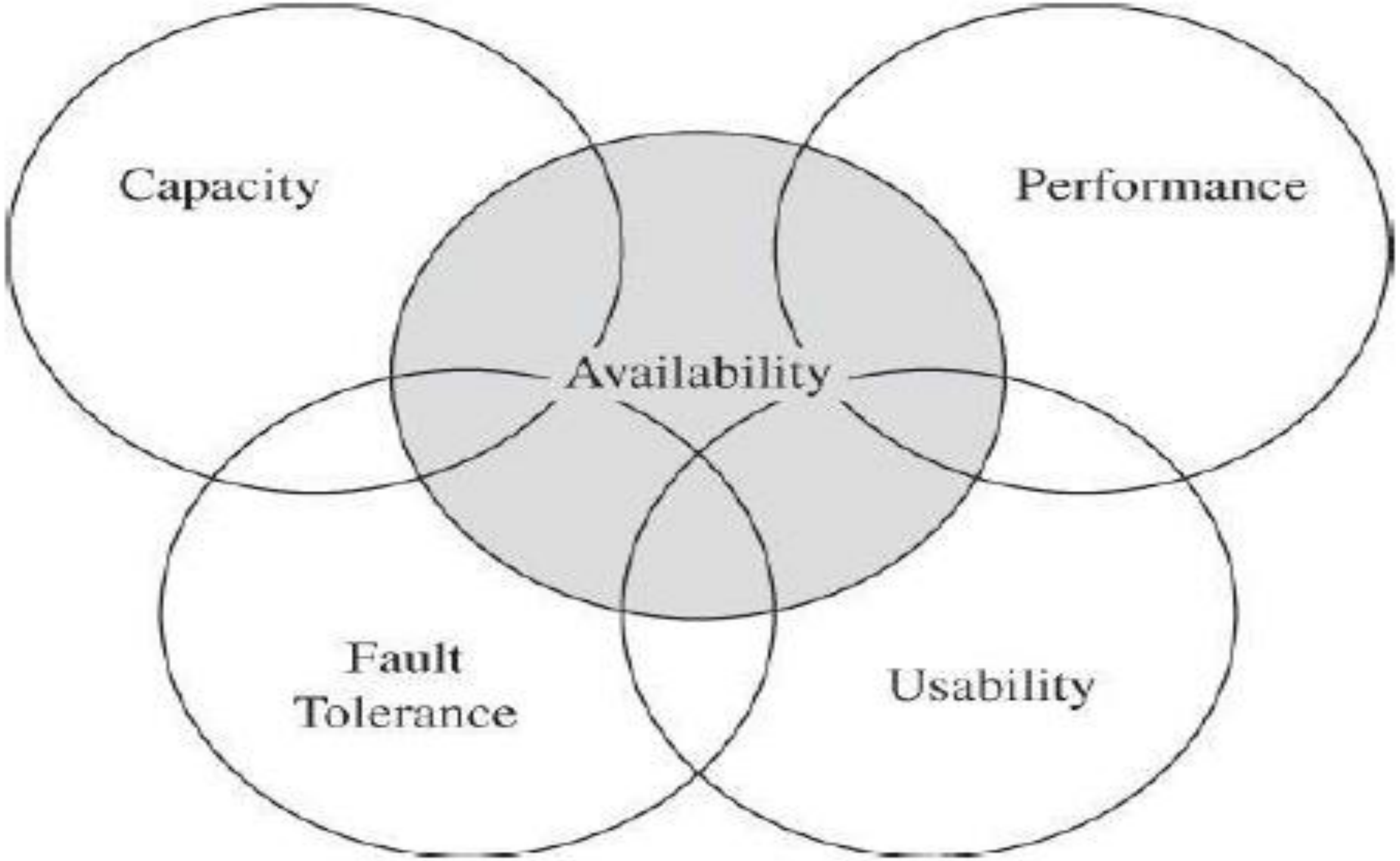
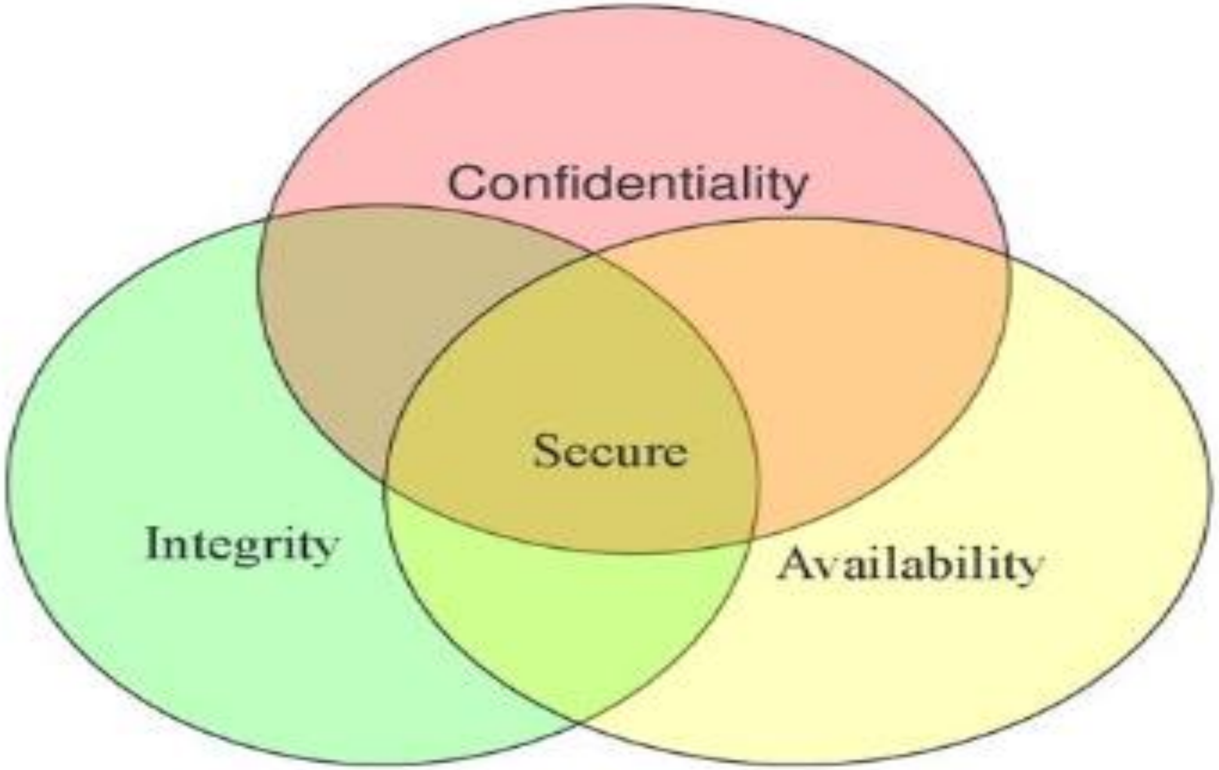


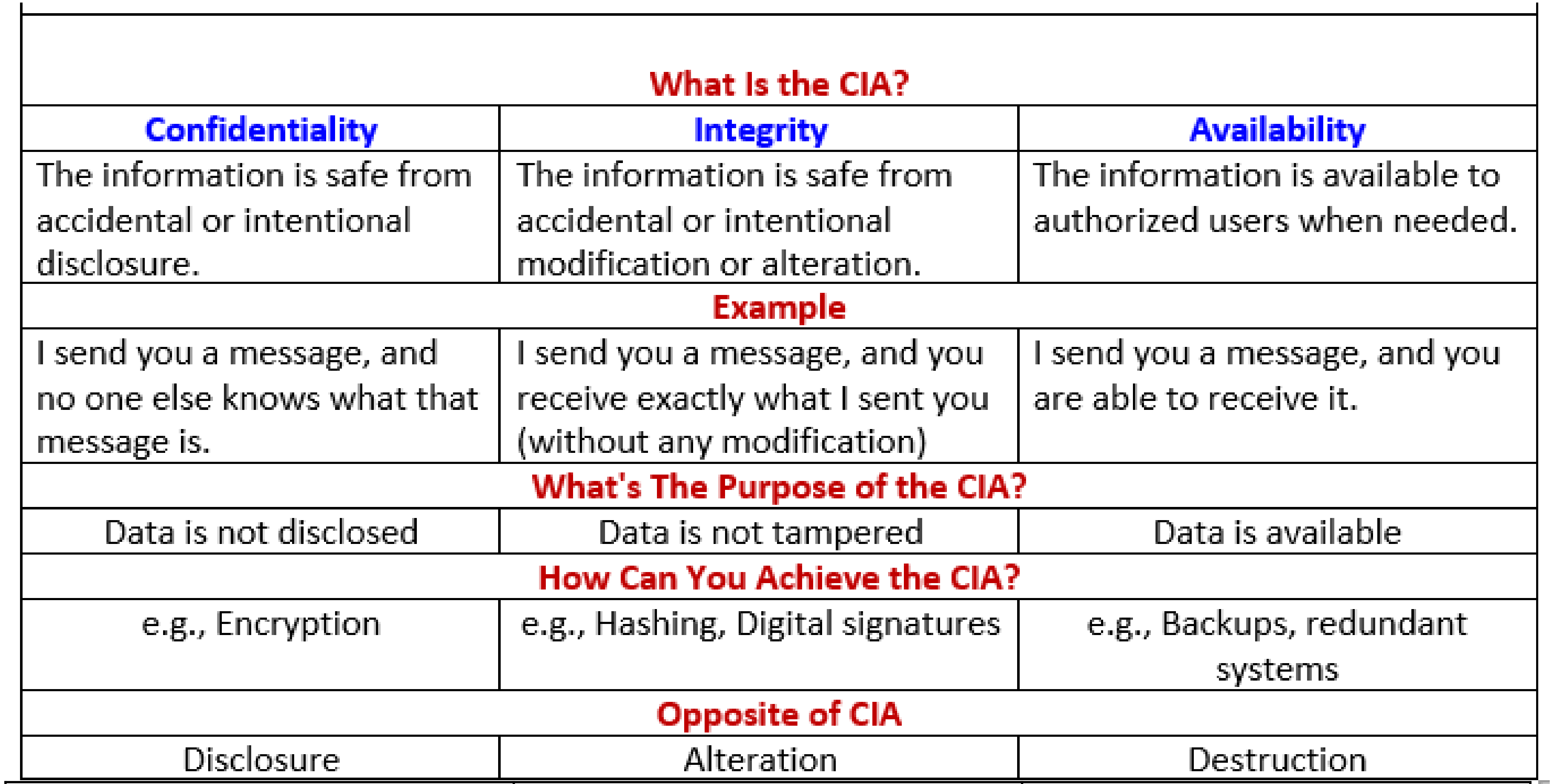
FIGURE 1-7 Availability and Related Aspects

Relationship between Confidentiality Integrity and Availability

- In fact, these three characteristics can be independent, can overlap, and can even be mutually exclusive.

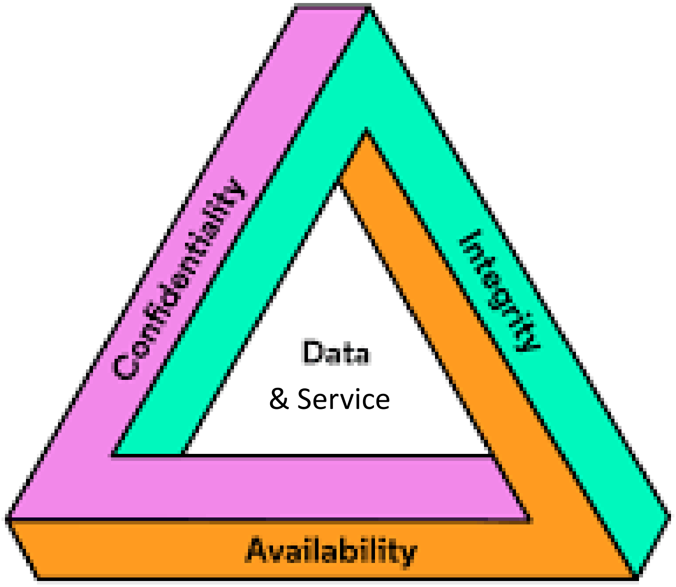


CIA Triad



What Is the CIA?		
Confidentiality	**Integrity**	**Availability**
The information is safe from accidental or intentional disclosure.	The information is safe from accidental or intentional modification or alteration.	The information is available to authorized users when needed.
Example		
I send you a message, and no one else knows what that message is.	I send you a message, and you receive exactly what I sent you (without any modification)	I send you a message, and you are able to receive it.
What's The Purpose of the CIA?		
Data is not disclosed	Data is not tampered	Data is available
How Can You Achieve the CIA?		
e.g., Encryption	e.g., Hashing, Digital signatures	e.g., Backups, redundant systems
Opposite of CIA		
Disclosure	Alteration	Destruction

Eg: Account Information Banking Server to Web or Mobile Banking, Data Traffic should be Encrypted.	Eg: Patient's information Sensors are connected and collects the data which sends to doctor.(IoT) If hacker hacks the server and modifies the data, then it will be a problem.	Eg: Authentication Service (google.com anytime to anywhere)



Additional to CIA:

- Authenticity
- Accountability

Security Goals

- When we talk about computer security, we mean that we are addressing three important aspects of any computer-related system: **confidentiality, integrity, & availability (CIA)**
- **Confidentiality** ensures that computer-related assets are accessed only by authorized parties.
- i.e. reading, viewing, printing, or even knowing their existence
- Secrecy or privacy
- **Integrity** means that assets can be modified only by authorized parties or only in authorized ways.
- i.e. writing, changing, deleting, creating.
- **Availability** means that assets are accessible to authorized parties at appropriate times.
- i.e. often, availability is known by its opposite, denial of service.

Goals of Security

Prevention :

- Prevent attackers from violating security policy

Detection :

- Detect attackers’ violation of security policy

Recovery :

- Stop attack, assess and repair damage
- Continue to function correctly even if attack succeeds

Security Attacks

If Attack is successfully launched
Loss of data, corruption of data, injection of viruses, defacing the server.
The attacker only knows for what he has launched an attack.

- Action that compromises the security of an individual or an organization.

Types:

- Passive Attacks

Attempts to learn or make use of information from the system.

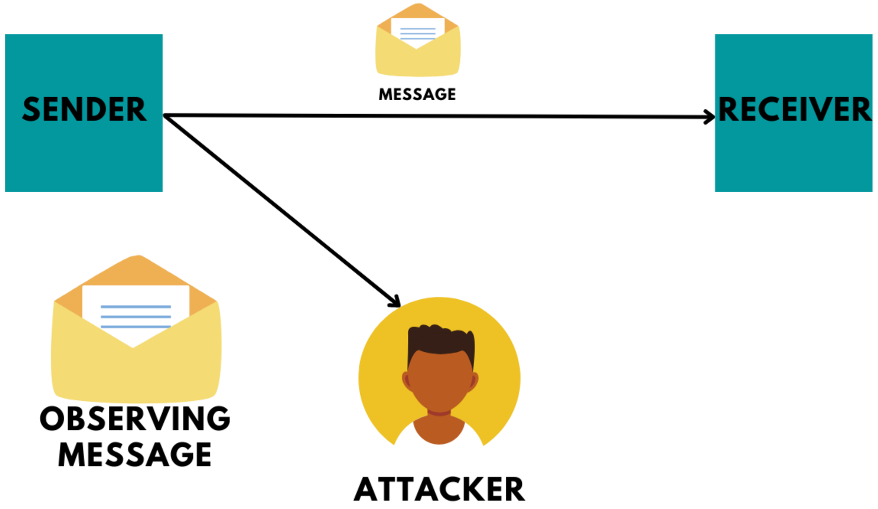
Does not affect system resources.

Eavesdropping or monitoring of transmissions.

Goal: Obtain information that is being transmitted.

Types: 1. Release of message contents

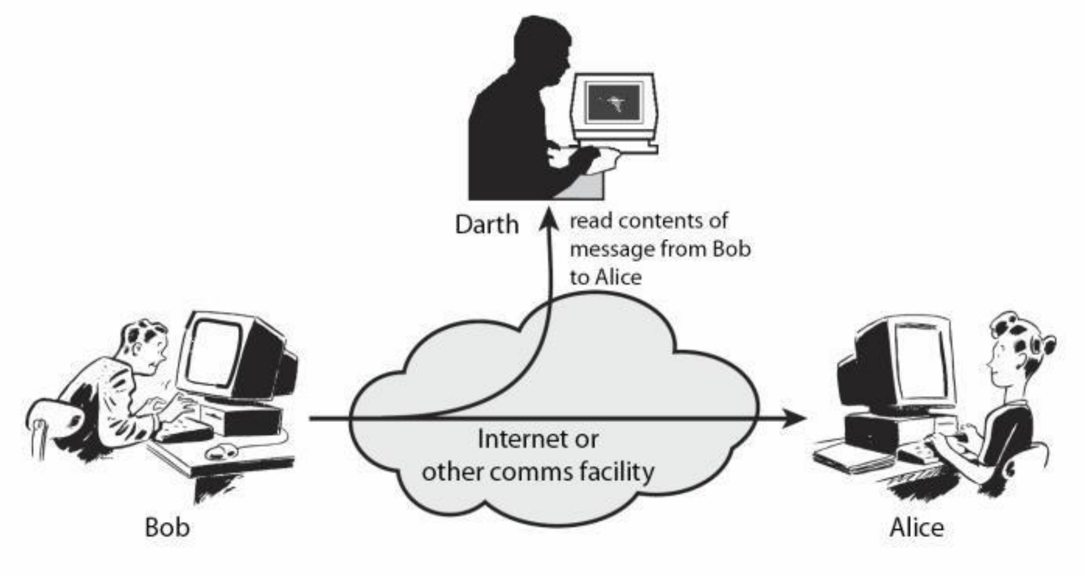
2. Traffic Analysis



Passive Attack cont..

1. Release of message contents

Reads contents of message from Bob to Alice.


Darth
read contents of message from Bob to Alice
Internet or other comms facility
Bob
Alice

2. Traffic Analysis

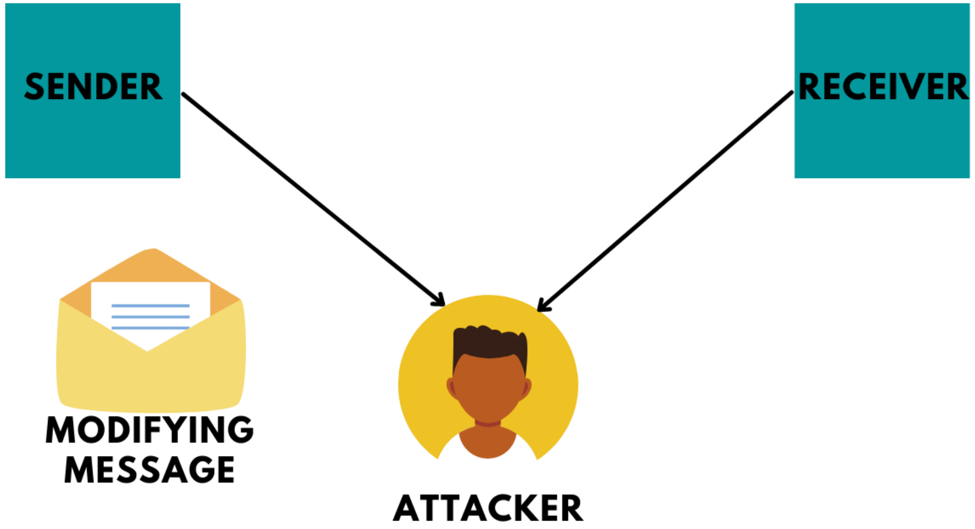
Observe pattern of messages from Bob to Alice.

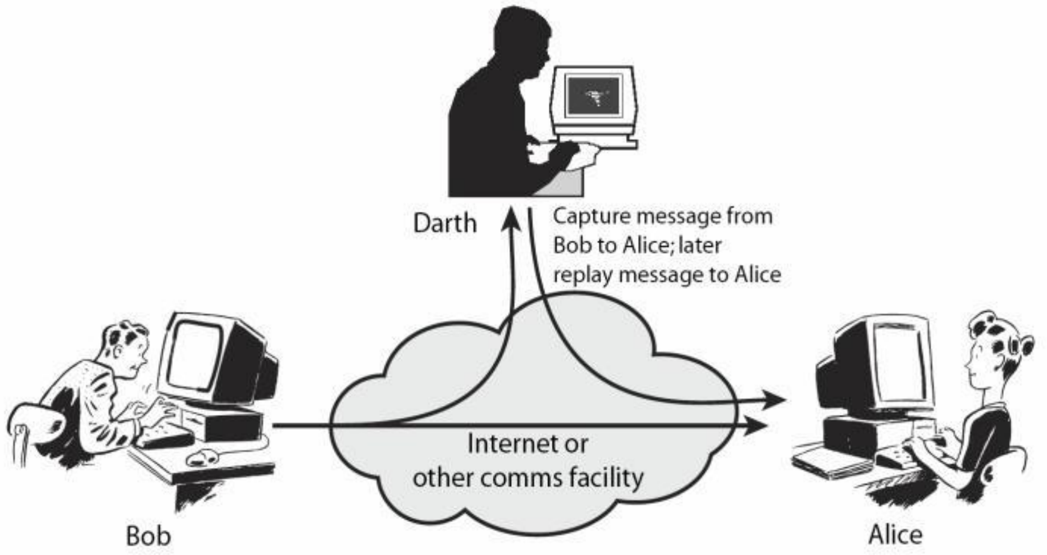
(length of message, location, identity of community host, frequency of message transfer).

Active Attack

- Active attack involve some modification of the data stream or the creation of a false stream.

- Subdivided into four categories:
  1. Masquerade
  2. Replay
  3. Modification of messages
  4. Denial of Service (DoS)




Darth
Capture message from Bob to Alice; later replay message to Alice
Internet or other comms facility
Bob
Alice

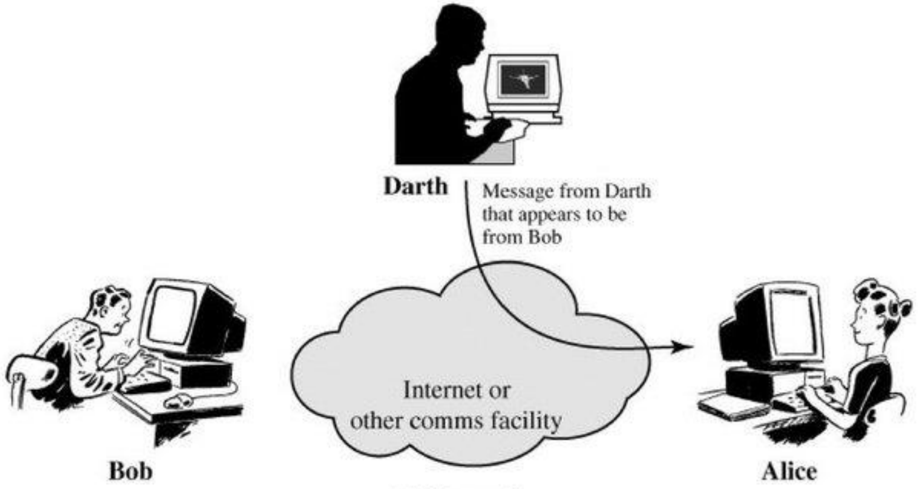
- Masquerade

  Message from Darth appears to be from Bob.

  1-Entity pretends to be a different entity.

  (If I steal some one username and password, and I login, then I am a masquerade.)

  An authorized entity with few privileges may demand for more privileges.

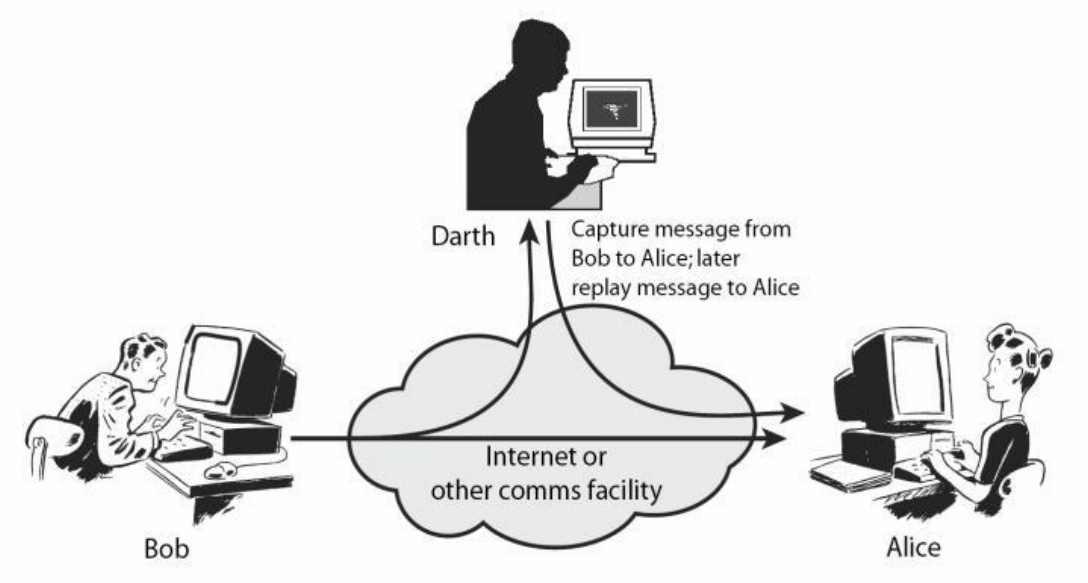

Darth
Message from Darth that appears to be from Bob
Internet or other comms facility
Bob
Alice

(a) Masquerade

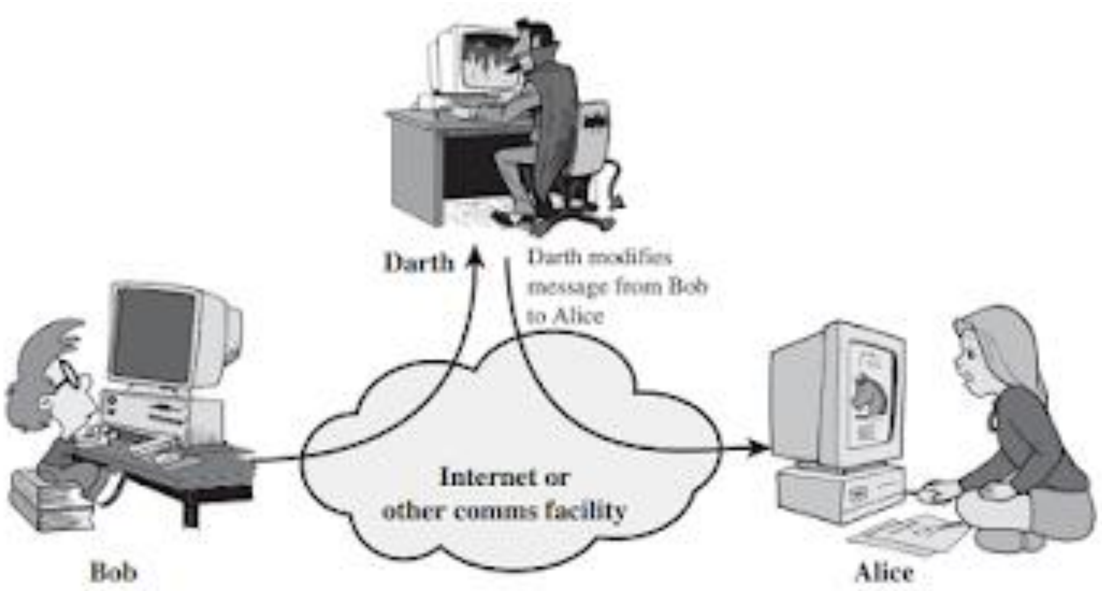
Active Attack – Masquerade

- Replay

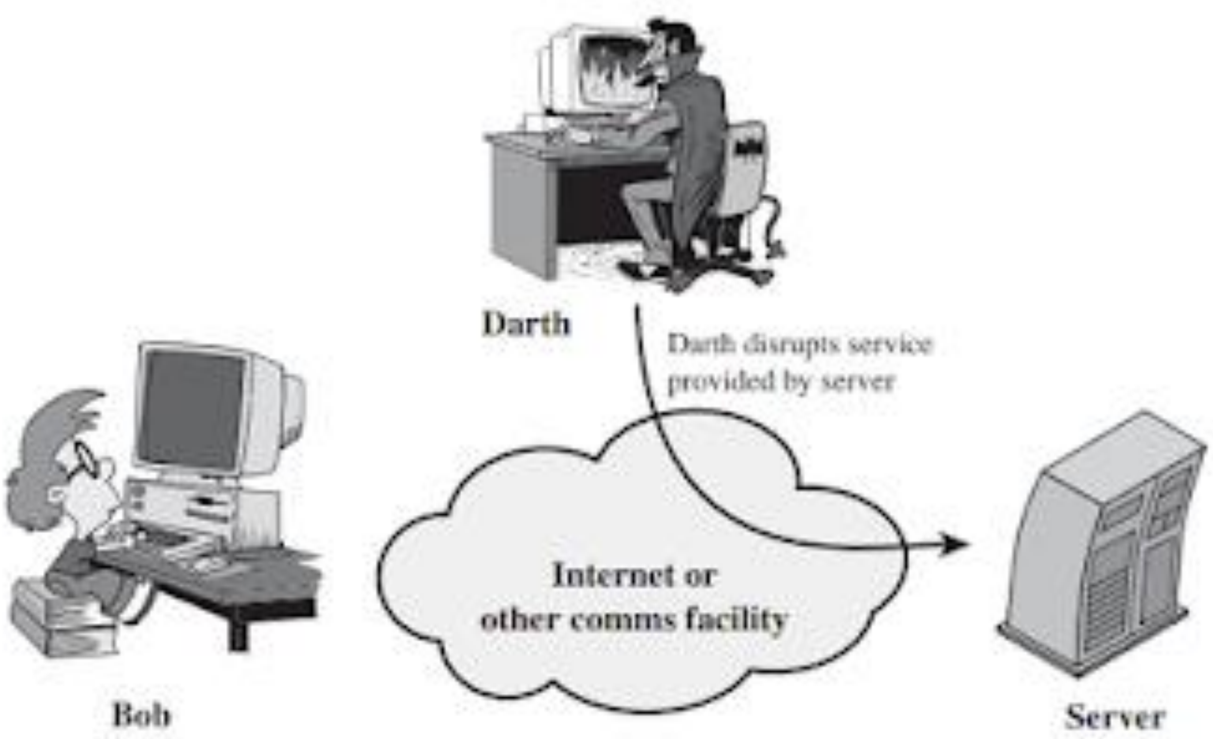
Capture message from Bob to Alice; later replay message to Alice.


Darth
Capture message from Bob to Alice; later replay message to Alice
Internet or other comms facility
Bob
Alice

- Modification of message


Darth
Darth modifies message from Bob to Alice
Internet or other comms facility
Bob
Alice

- Denial of Service (DoS)


Darth
Darth disrupts service provided by server
Bob
Internet or other comms facility
Server

Passive Attack

- Hard to Detect
- Neither sender nor receiver is aware of the attack.
- Encryption prevents the success of the passive attacks.
- More emphasis is on prevention than detection.

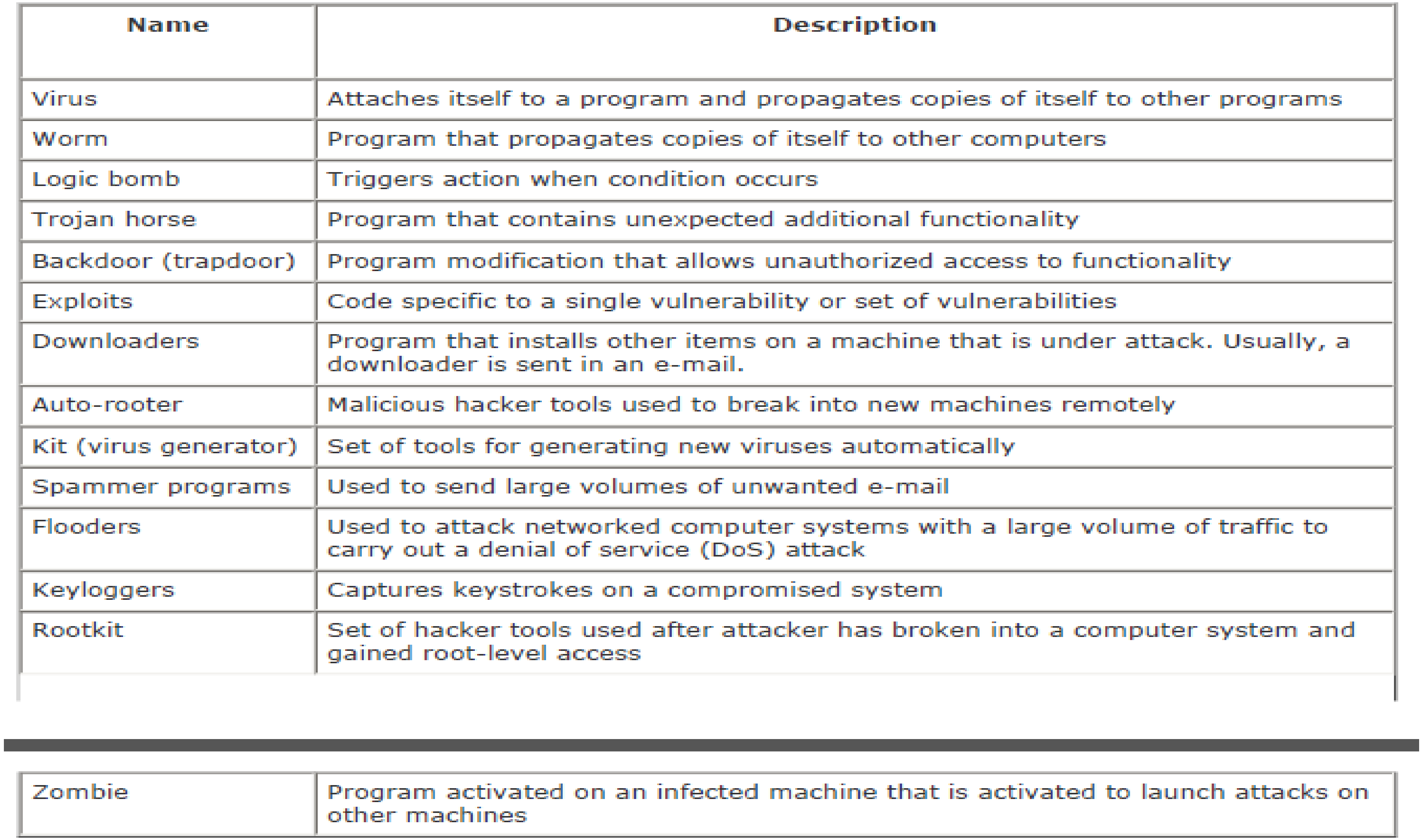
Active Attack

- Hard to Prevent
- Difficult to prevent-Physical, software and network vulnerabilities.
- Detect and recover from any disruption or delays.
- If the detection has a deterrent effect, it may also contribute to prevention.

Malware

- Programs planted by an agent with malicious intent to cause unanticipated or undesired effects
- Virus
  - A program that can replicate itself and pass on malicious code to other non-malicious programs by modifying them
- Worm
  - A program that spreads copies of itself through a network
- Trojan horse
  - Code that, in addition to its stated effect, has a second, nonobvious, malicious effect

Terminology of Malicious Code

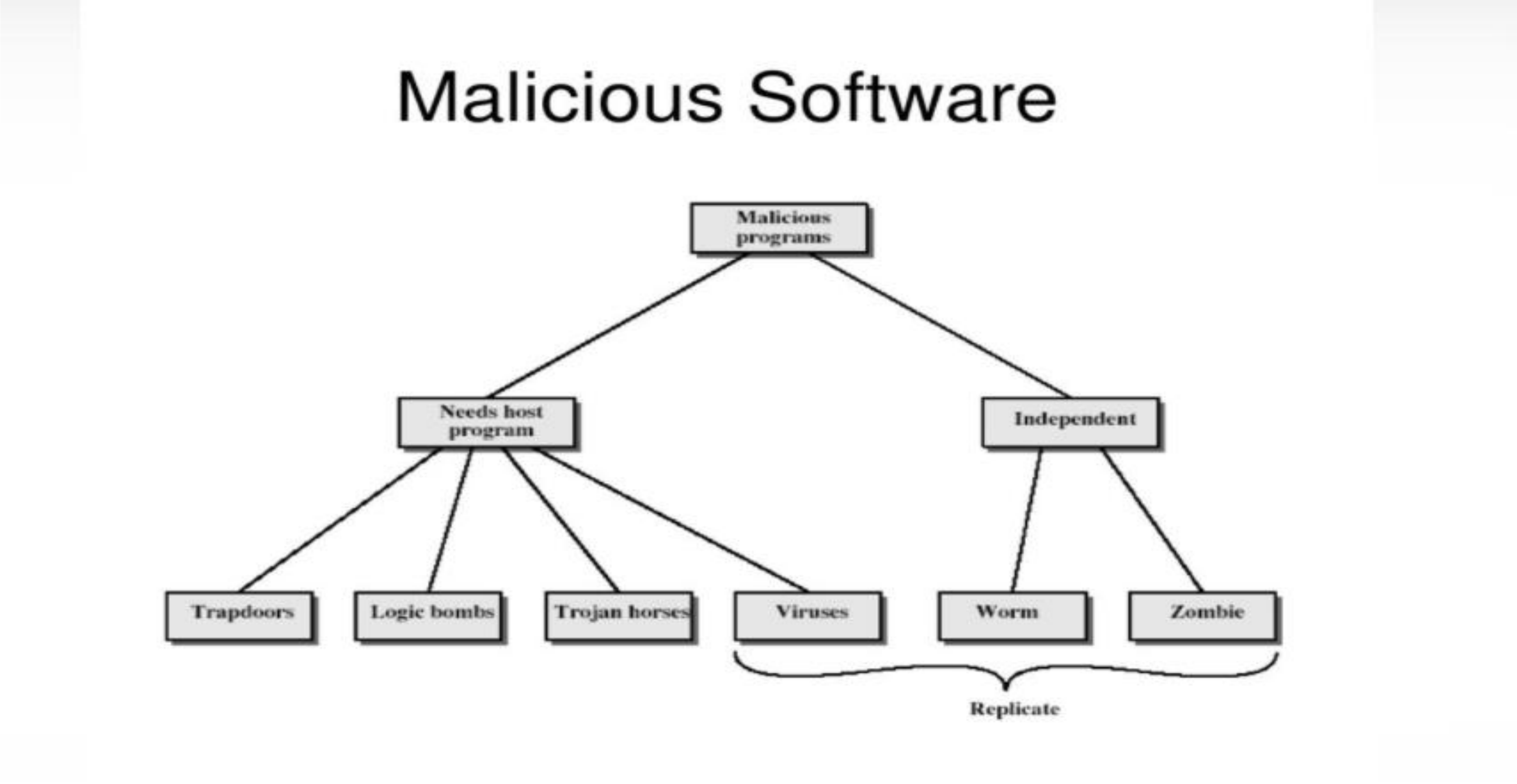


Name	Description
Virus	Attaches itself to a program and propagates copies of itself to other programs
Worm	Program that propagates copies of itself to other computers
Logic bomb	Triggers action when condition occurs
Trojan horse	Program that contains unexpected additional functionality
Backdoor (trapdoor)	Program modification that allows unauthorized access to functionality
Exploits	Code specific to a single vulnerability or set of vulnerabilities
Downloaders	Program that installs other items on a machine that is under attack. Usually, a downloader is sent in an e-mail.
Auto-rooter	Malicious hacker tools used to break into new machines remotely
Kit (virus generator)	Set of tools for generating new viruses automatically
Spammer programs	Used to send large volumes of unwanted e-mail
Flooders	Used to attack networked computer systems with a large volume of traffic to carry out a denial of service (DoS) attack
Keyloggers	Captures keystrokes on a compromised system
Rootkit	Set of hacker tools used after attacker has broken into a computer system and gained root-level access
Zombie	Program activated on an infected machine that is activated to launch attacks on other machines

Difference between Virus, worm and Trojan horse

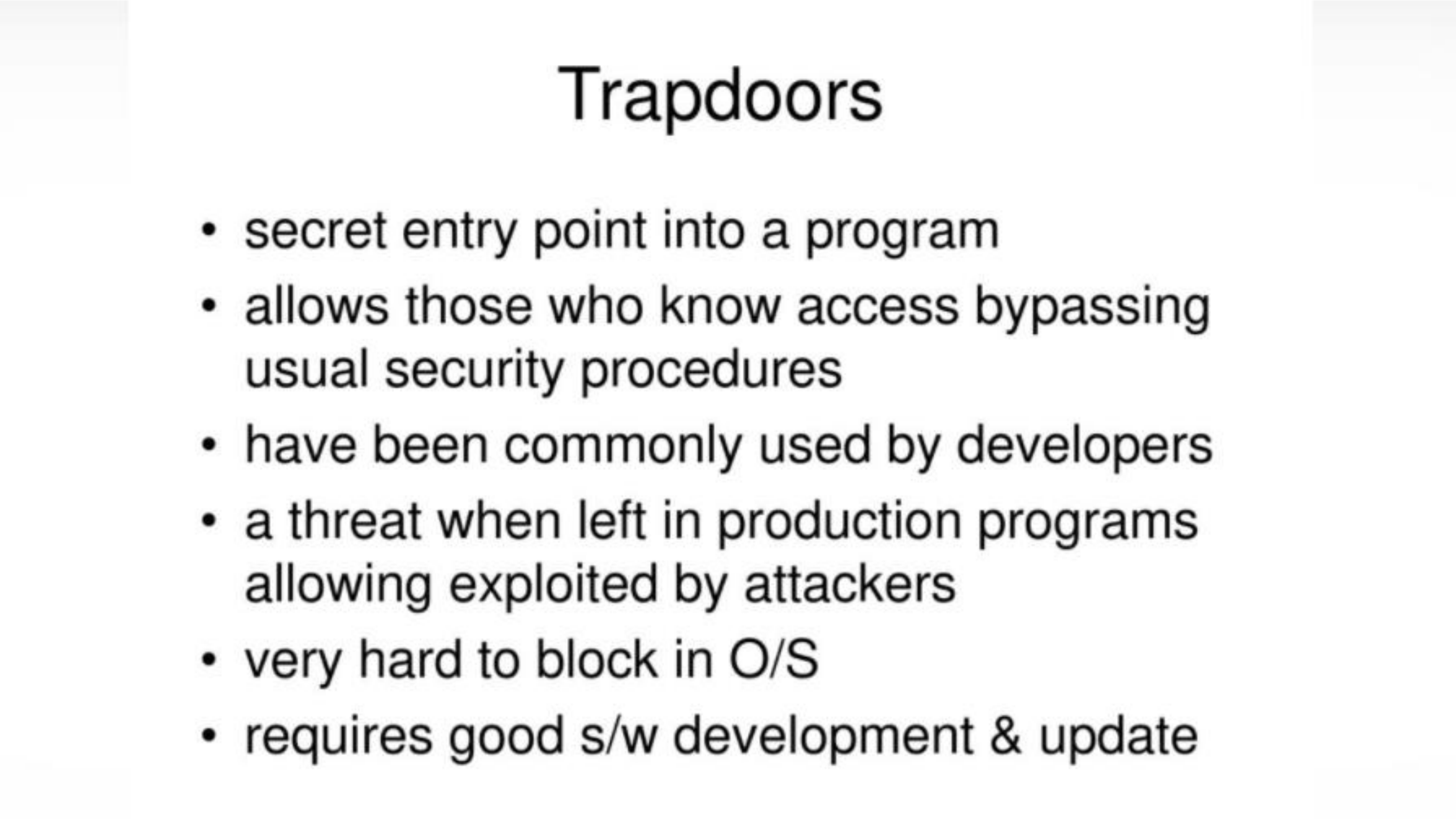
Virus	Worm	Trojan Horse
Virus is a software or computer program that connect itself to another software or computer program to harm computer system.	Worms replicate itself to cause slow down the computer system.	Trojan Horse rather than replicate capture some important information about a computer system or a computer network.
Virus replicates itself.	Worms are also replicates itself.	But Trojan horse does not replicate itself.
Virus can’t be controlled by remote.	Worms can be controlled by remote.	Like worms, Trojan horse can also be controlled by remote.
Spreading rate of viruses are moderate.	While spreading rate of worms are faster than virus and Trojan horse.	And spreading rate of Trojan horse is slow in comparison of both virus and worms.
The main objective of virus to modify the information.	The main objective of worms to eat the system resources.	The main objective of Trojan horse to steal the information.
Viruses are executed via executable files.	Worms are executed via weaknesses in system.	Trojan horse executes through a program and interprets as utility software.

Categories of Malware



Malicious Software

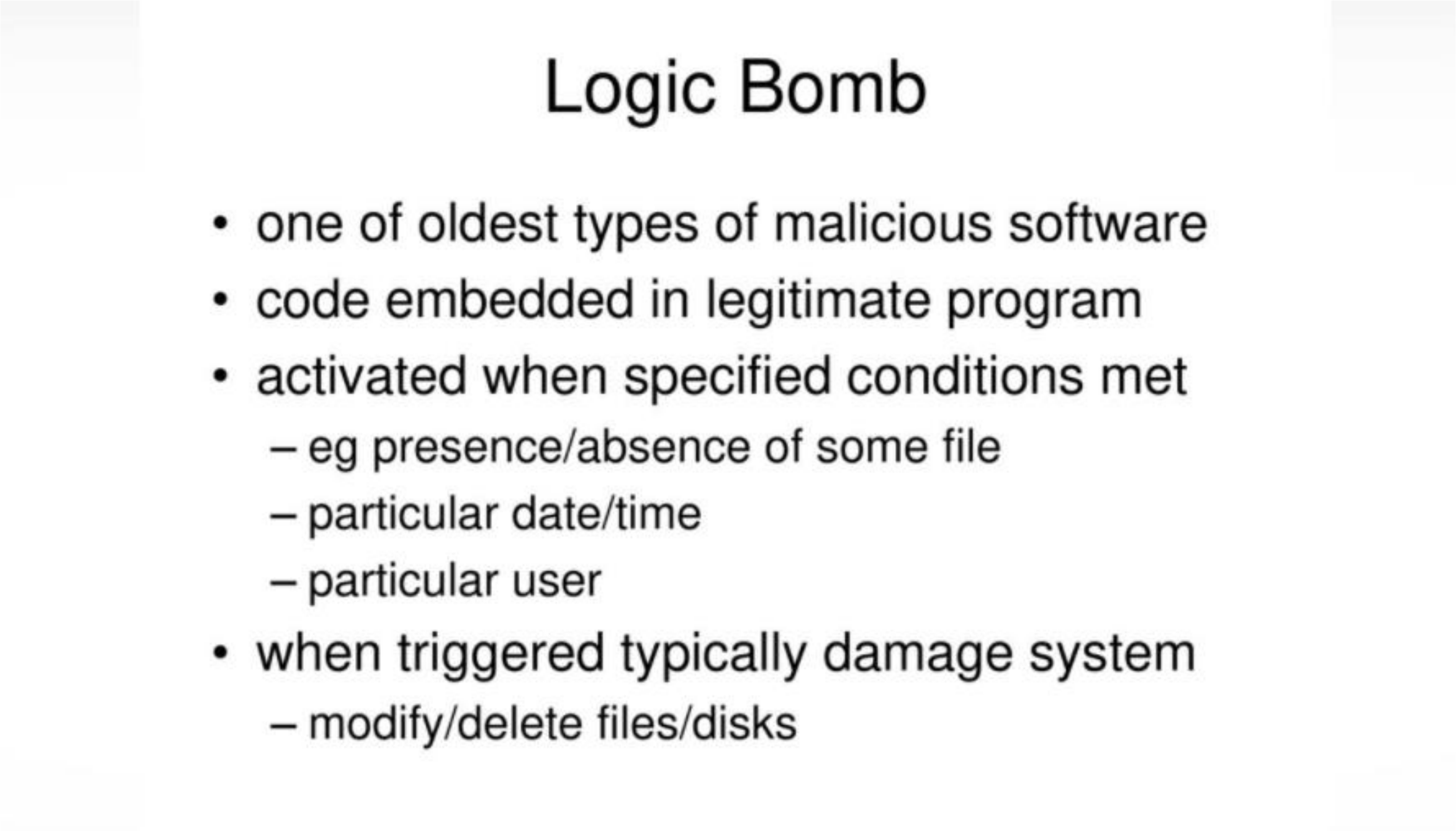
Back doors / Trap doors



Trapdoors

- secret entry point into a program
- allows those who know access bypassing usual security procedures
- have been commonly used by developers
- a threat when left in production programs allowing exploited by attackers
- very hard to block in O/S
- requires good s/w development & update

Logic Bombs



Logic Bomb

- one of oldest types of malicious software
- code embedded in legitimate program
- activated when specified conditions met
  - eg presence/absence of some file
  - particular date/time
  - particular user
- when triggered typically damage system
  - modify/delete files/disks

Trojan Horse

- program with hidden side-effects
- which is usually superficially attractive
  - eg game, s/w upgrade etc
- when run performs some additional tasks
  - allows attacker to indirectly gain access they do not have directly
- often used to propagate a virus/worm or install a backdoor
- or simply to destroy data

Zombie

- program which secretly takes over another networked computer
- then uses it to indirectly launch attacks
- often used to launch distributed denial of service (DDoS) attacks
- exploits known flaws in network systems

Viruses

- a piece of self-replicating code attached to some other code
  - cf biological virus
- both propagates itself & carries a payload
  - carries code to make copies of itself
  - as well as code to perform some covert task

Virus Operation

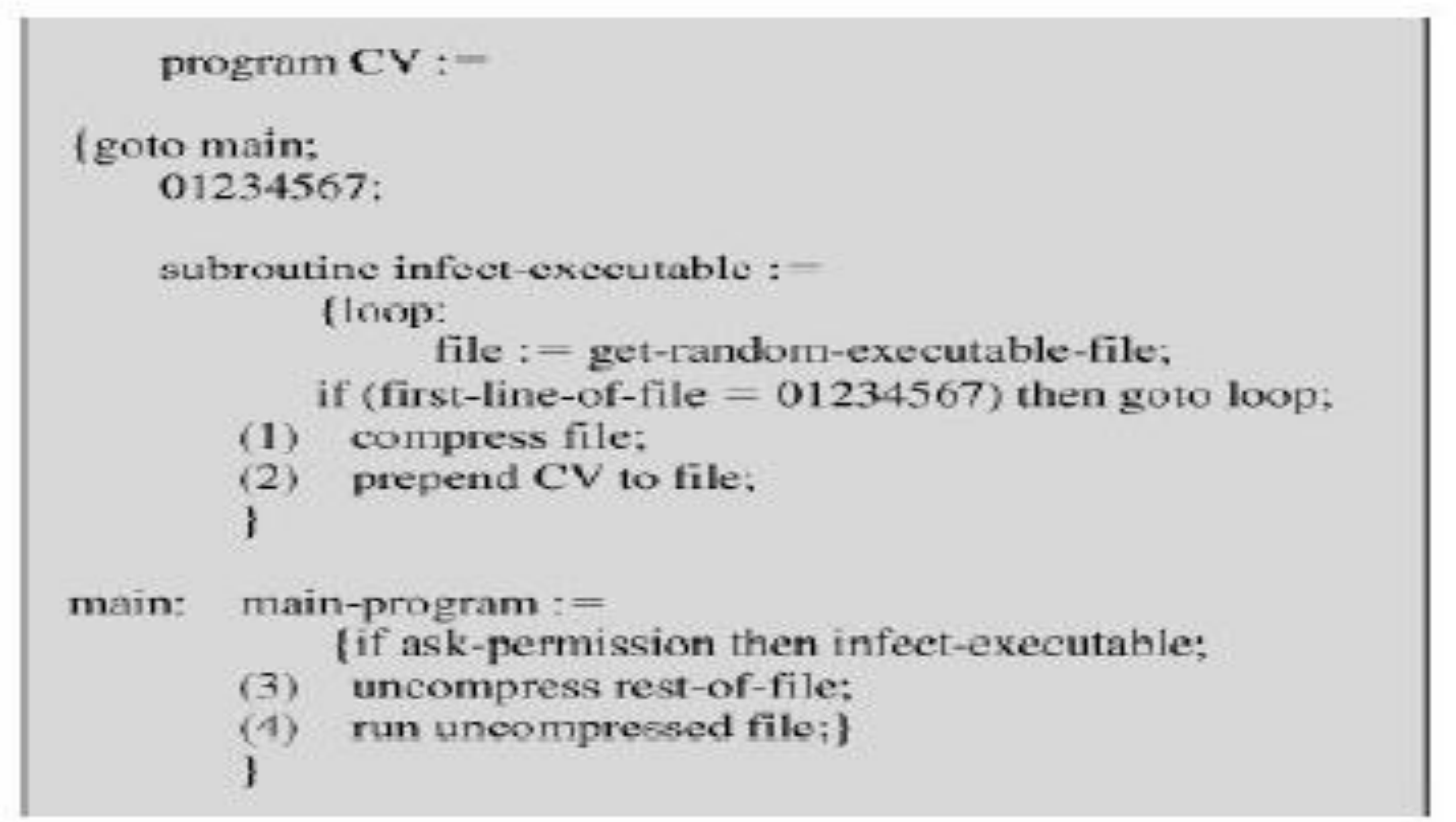
- virus phases:
  - dormant – waiting on trigger event
  - propagation – replicating to programs/disks
  - triggering – by event to execute payload
  - execution – of payload
- details usually machine/OS specific
  - exploiting features/weaknesses

Virus Structure

```
program V :=
    {goto main;
    1234567;
    subroutine infect-executable :=         {loop:
                    file := get-random-executable-file;
                    if (first-line-of-file = 1234567) then goto loop
                    else prepend V to file; }
    subroutine do-damage :=         {whatever damage is to be done}
    subroutine trigger-pulled :=    {return true if some condition holds}
    main: main-program :=           {infect-executable;
                                    if trigger-pulled then do-damage;
                                    goto next;}

    next:
}
```

Logic for a Compression Virus



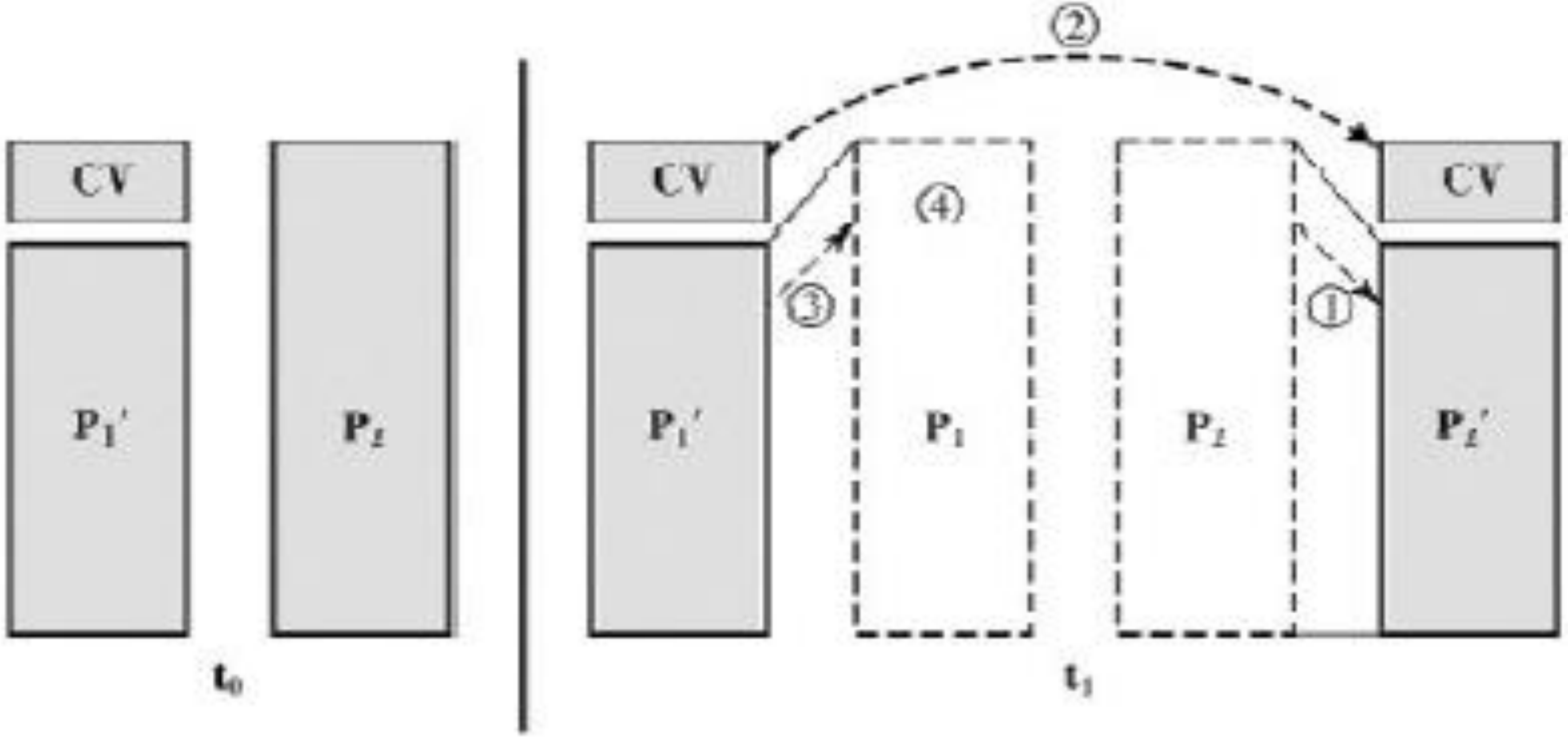
```
     program CV :=

{goto main;
     01234567;

     subroutine infect-executable :=
               {loop:
                       file := get-random-executable-file;
               if (first-line-of-file = 01234567) then goto loop;
          (1)   compress file;
          (2)   prepend CV to file;
          }

main:     main-program :=
               {if ask-permission then infect-executable;
          (3)   uncompress rest-of-file;
          (4)   run uncompressed file;}
          }
```

Compression Virus



Types of Viruses

- can classify on basis of how they attack
- parasitic virus
- memory-resident virus
- boot sector virus
- stealth
- polymorphic virus
- macro virus

Worms

- replicating but not infecting program
- typically spreads over a network
  - cf Morris Internet Worm in 1988
  - led to creation of CERTs
- using users distributed privileges or by exploiting system vulnerabilities
- widely used by hackers to create **zombie PC's**, subsequently used for further attacks, esp DoS
- major issue is lack of security of permanently connected systems, esp PC's

Network worm

- Use network connections to spread from system to system.
- Once active within a system, it can behave as a computer virus or it could implant Trojan horse programs or perform any no. of disruptive or destructive actions.
- To replicate itself, a network worm uses some sort of network vehicle.
- Examples include the following:

I. Electronic mail Facility: A worm mails a copy of itself to other systems.

II. Remote execution capability: A worm executes a copy of itself on another systems.

III. Remote login capability: A worm logs onto a remote system as a user and then uses commands to copy itself from one system to the other.

❖ *A Network worm exhibits the same characteristics as a computer virus:*

Worm Operation

- worm phases like those of viruses:
  - dormant
  - propagation
    - search for other systems to infect
    - establish connection to target remote system
    - replicate self onto remote system
  - triggering
  - execution

State of Worm Technology

- Multiplatform
- Multi-exploit
- Ultrafast Spreading
- Polymorphic
- Metamorphic
- Transport vehicles
- Zero-day Exploit

Virus Countermeasures

- viral attacks exploit lack of integrity control on systems
- to defend need to add such controls
- typically by one or more of:
  - **prevention** - block virus infection mechanism
  - **detection** - of viruses in infected system
  - **reaction** - restoring system to clean state

Anti-Virus Software

- **first-generation**
  - scanner uses virus signature to identify virus
  - or change in length of programs
- **second-generation**
  - uses heuristic rules to spot viral infection
  - or uses program checksums to spot changes
- **third-generation**
  - memory-resident programs identify virus by actions
- **fourth-generation**
  - packages with a variety of antivirus techniques
  - eg scanning & activity traps, access-controls

Advanced Anti-Virus Techniques

- generic decryption
  - use CPU simulator to check program signature & behavior before actually running it
- digital immune system (IBM)
  - general purpose emulation & virus detection
  - any virus entering org is captured, analyzed, detection/shielding created for it, removed

Generic Decryption (GD)

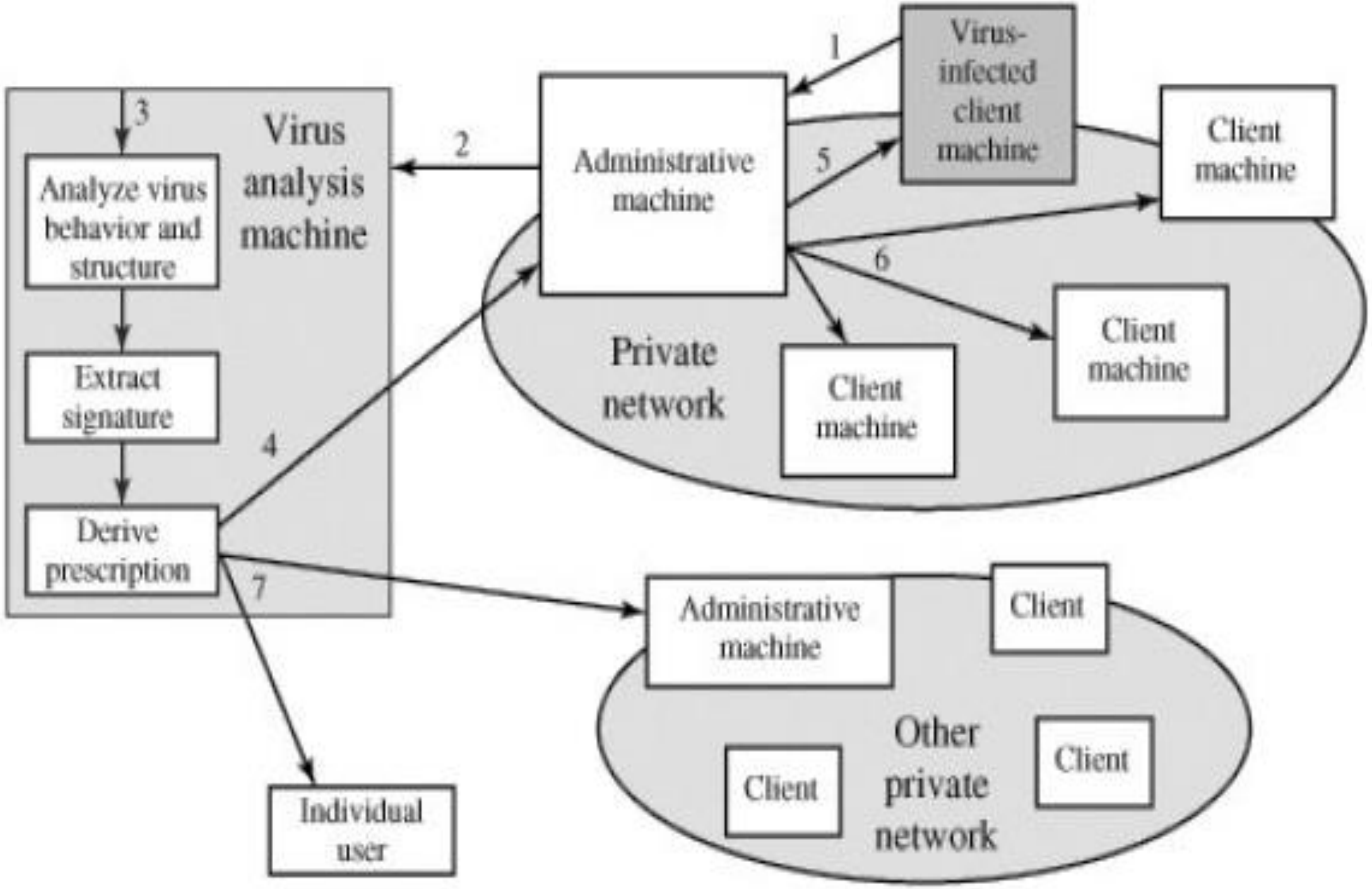
- GD technology enables the antivirus program to easily detect even the most complex polymorphic viruses, while maintaining fast scanning speeds.
- In order to detect such a structure, executable files are run through a GD scanner, which contains the following elements:

❖**CPU emulator:** ***A software based virtual computer.***

❖**Virus signature scanner:** ***A module that scans the target code looking for known virus signatures.***

❖**Emulation control module:** ***Controls the execution of a target code.***

Digital Immune system



Intruders

- An Intruder is a person who attempts to gain unauthorized access to a system, to damage that system, or to disturb data.
- significant issue for networked systems is hostile(dislike) or unwanted access, trespass(enter without permission) being unauthorized login or by software such as a virus, worm, or Trojan horse.
- either via network or local or remote user

- **Types of intruders:**
  - Masquerader- An individual who is not authorized to use the computer (outsider)

  - Misfeasor- authentic user doing unauthorized actions. A legitimate user who accesses unauthorized data, programs, or resources (insider)

  - Clandestine(confidential)user- : An individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection (either)

Intrusion Techniques

- *The basic* aim is to gain access and/or increase privileges on some system.
- basic attack methodology is taken from
  - target acquisition and information gathering
  - initial access
  - privilege escalation (rise)
  - covering tracks
- key goal often is to *acquire a user (preferably administrator) password*
- **so the attacker can login and exercise all the** access rights of the account owner.

Intrusion Detection

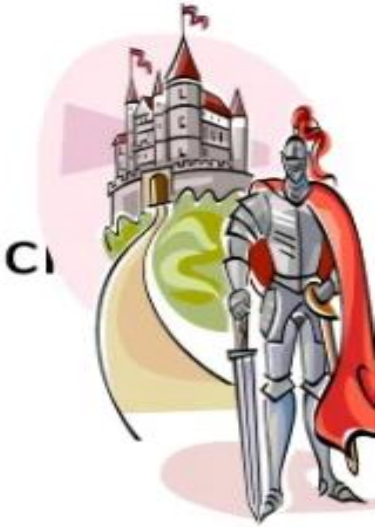
- Intrusions are the activities that violate the security policy of system
- Intrusion Detection is the process used to identify intrusions(ie).,
  - block if detected quickly (block access & minimize damage)
  - act as deterrent (can collect info on intruders to improve future security)
- Intrusion detection is based on the assumption that the behavior of the intruder differs from that of a legitimate user in ways that can be quantified

Approaches to Intrusion Detection

- Statistical anomaly detection- collect data relating to the behavior of legitimate users, then use statistical tests to determine with a high level of confidence whether new behavior is legitimate user behavior or not.

  -Threshold(entry)-define thresholds, independent of user, for the frequency of occurrence of events.

  -Profile based-develop profile of activity of eac user and use to detect changes in the behavior



- Rule-based detection-attempt to define a set of rules used to decide if given behavior is an intruder.
- There are two types,
  - Anomaly(abnormal)-rules detect deviation from previous usage patterns
  - Penetration(entry -action) identification-expert system approach that searches for suspicious behavior(doubt)

Audit Records

- A fundamental tool for intrusion detection is the **audit record**.
- Some record of ongoing activity by users must be maintained as input to an intrusion detection system.
- There are two types of records:

    1.Native audit records

    2.Detection-specific audit records

native audit records

- multiuser operating systems include accounting software that collects information on user activity
- advantage is that no additional collection software is needed
- disadvantage is that records may not contain the needed information or in a convenient form

detection-specific audit record

- collection facility that generates records containing only information required by the IDS
- advantage is that it could be made vendor independent and ported to a variety of systems
- disadvantage is the extra overhead of having, in effect, two accounting packages running on a machine

Audit Record Analysis

- An analysis of audit records over a period of time can be used to determine the activity profile of the average user.

- Then current audit records are used as input to detect intrusion, by analyzing incoming audit records to determine deviation from average behavior.

- Example:

Counter-times of logins, command executed during a single user session, number of password failures,

Gauge-the number of logical connections assigned to a user application

Interval timer-the length of time between successive logins to an account,

Resource Utilization-number of pages printed during a user session

- Using the examples,

various tests can be performed to determine whether current activities fit within acceptable limits.

- • Mean and standard deviation.
- • Multivariate
- • Markov process
- • Time series
- • Operational

Base-Rate Fallacy

practically an intrusion detection system needs to detect a substantial percentage of intrusions with few false alarms

- if few intrusions detected -> the system provides a false sense of security

- If the system frequently triggers an alert when there is no intrusion (a false alarm), then either system managers will begin to ignore the alarms.

Distributed Intrusion Detection

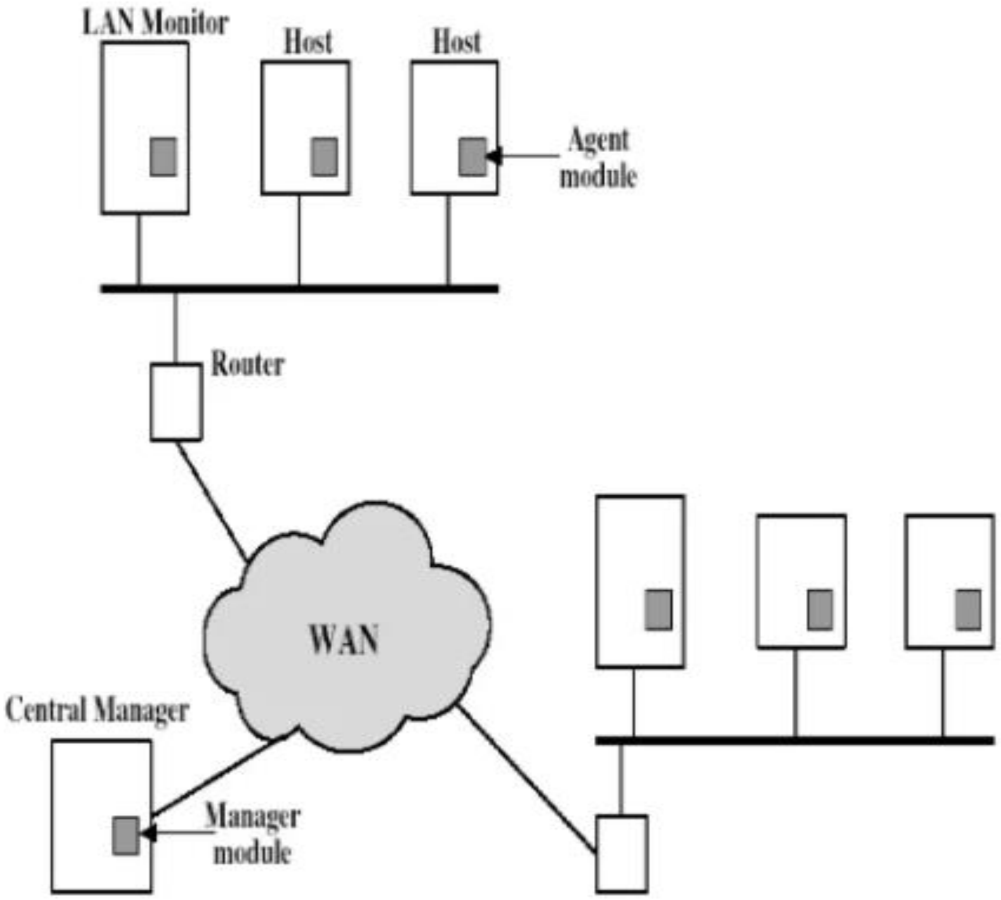
- traditional focus is on single systems
- but typically have networked systems
- more effective defense can be achieved by coordination and cooperation among intrusion detection systems across the network.

Deisgn Issuses

– integrity & confidentiality of networked data

– centralized or decentralized architecture

Distributed Intrusion Detection - Architecture



There are 3 components:

1. Host agent module

   -audit collection module operating as a background process on a monitored system

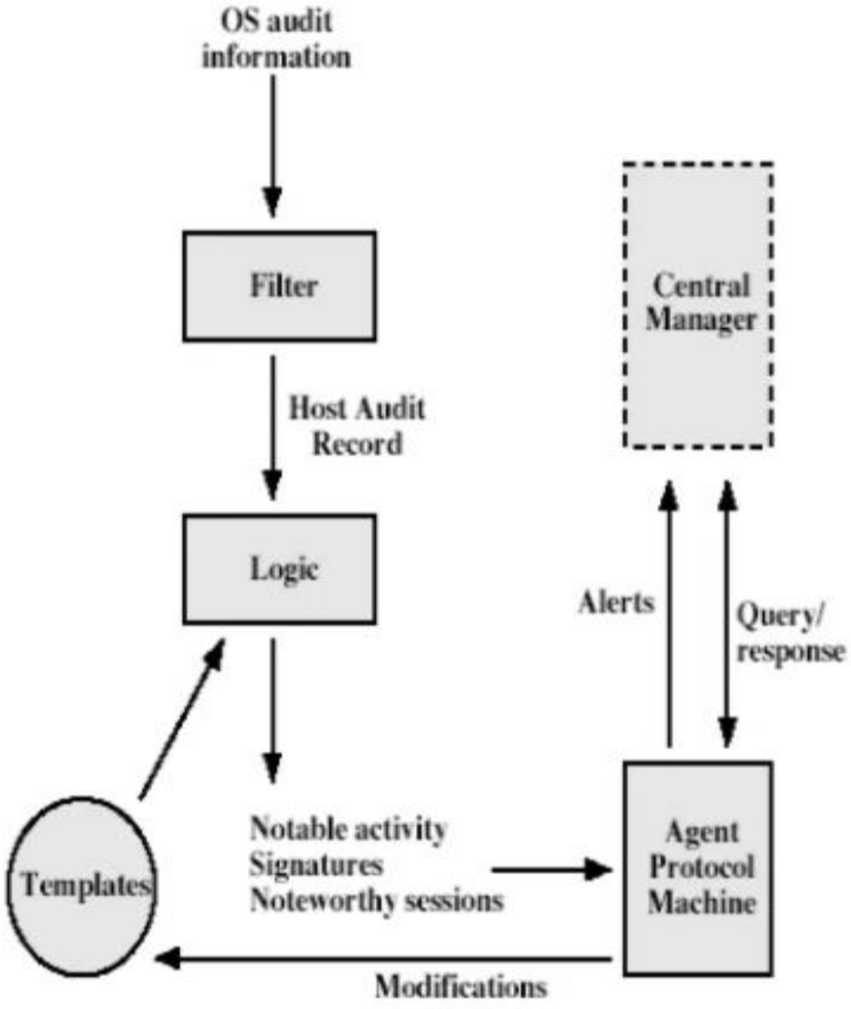
2. LAN monitor agent module

   -like a host agent module except it analyzes LAN traffic

3. Central manager module

   -Receives reports from LAN monitor and host agents and processes and correlates these reports to detect intrusion

Distributed Intrusion Detection – Agent Implementation



- The agent captures each native O/S audit record, & applies a filter that retains only records of security interest.
- These records are then reformatted into a standardized format (HAR).
- Then a template-driven logic module analyzes the records for suspicious activity. When suspicious activity is detected, an alert is sent to the central manager.
- The central manager includes an expert system that can draw inferences from received data.
- The manager may also query individual systems for copies of HARs to correlate with those from other agents.

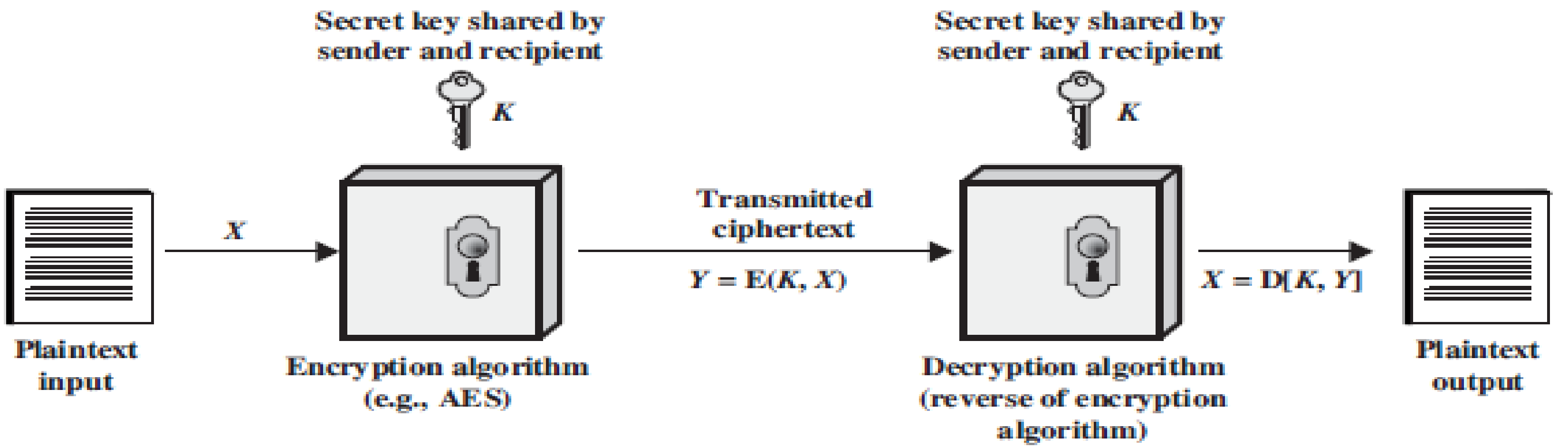
Honeypots

- a **honeypot** is a computer security mechanism set to detect, deflect, or, in some manner, counteract attempts at unauthorized use of information systems.
- Honeypots are decoy systems to lure attackers away from accessing critical systems
  - to collect information about the attacker’s activity
  - to encourage attacker to stay on system so administrator can respond
- These pots are filled with fabricated information designed to valuable.
- instrumented to collect detailed information on attackers activities
- single or multiple networked systems

Classical Encryption Techniques

- **Basic Terminology**
- **Plaintext** - the original message
- **Cipher text** - the coded message
- **Cipher** - algorithm for transforming plaintext to cipher text
- **Key** - info used in cipher known only to sender/receiver
- **Encipher (encrypt)** - converting plaintext to cipher text
- **Decipher (decrypt)** - recovering cipher text from plaintext
- **Cryptography** - study of encryption principles/methods
- **Cryptanalysis (code breaking)** - the study of principles/ methods of deciphering cipher text *without* knowing key
- **Cryptology** - the field of both cryptography and cryptanalysis

Schematic Representation of a Cryptosystem


Secret key shared by sender and recipient
K
Secret key shared by sender and recipient
K
X
Transmitted ciphertext
$Y = E(K, X)$
$X = D[K, Y]$
Plaintext input
Encryption algorithm (e.g., AES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

- Cryptography can be characterized 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 and Brute-force attack

- Two general attacks on conventional encryption
  - **Cryptanalytic attacks** rely on the nature of the algorithm plus some knowledge of the general characteristics of the plaintext or even some sample plaintext–ciphertext pairs.

  - This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used.

  - **Brute-force attack**: The attacker tries every possible key on a piece of cipher text until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success.

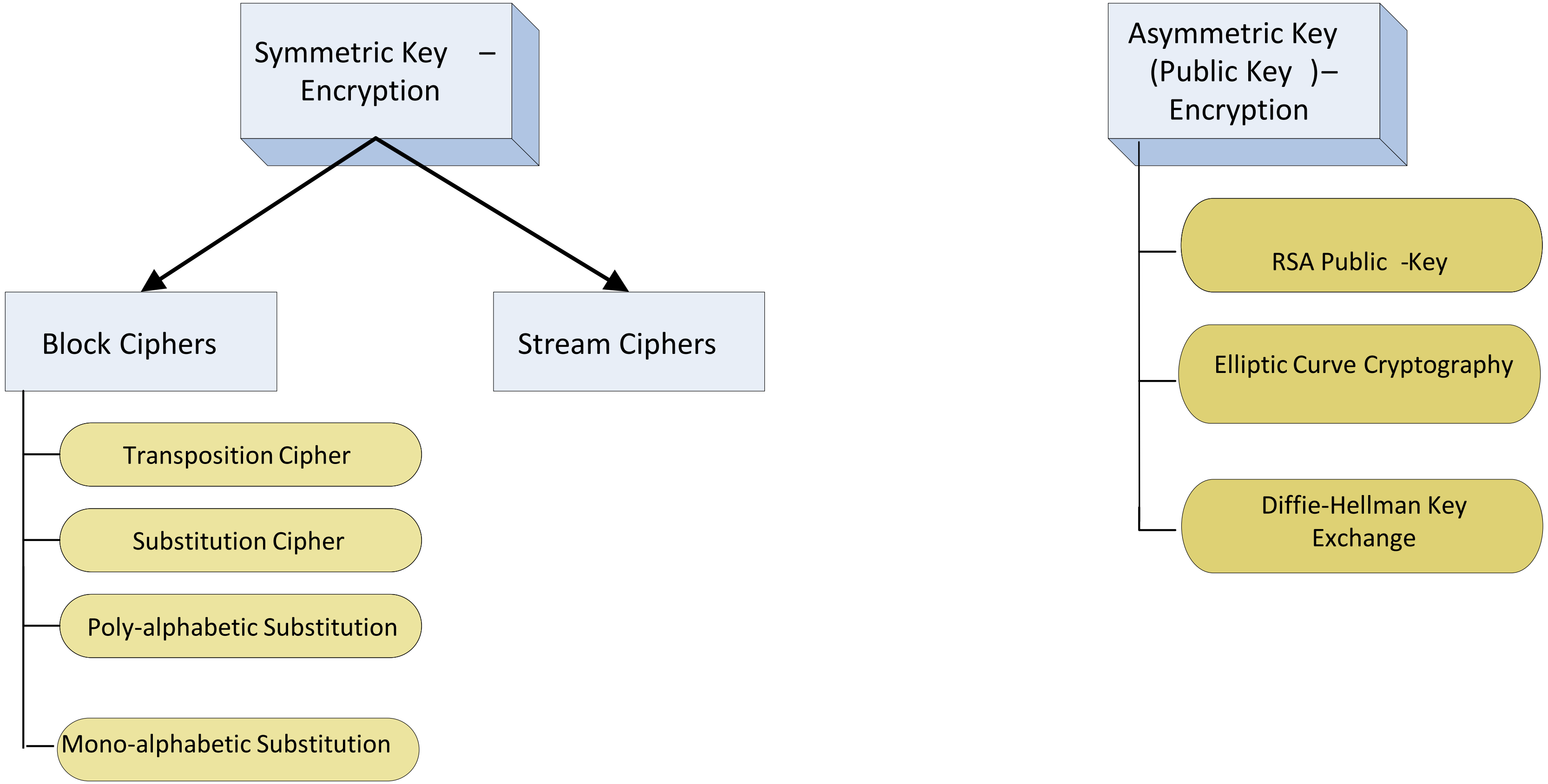
Types of Cryptanalytic Attacks

- **Cipher text only**
  - Only knows the algorithm, cipher text, and statistics
  - can identify plaintext
- **Known plaintext**
  - Knows algorithm, One or more plaintext–cipher text pairs formed with the secret key, cipher text
- **Chosen plaintext**
  - Knows algorithm, Plaintext message chosen by cryptanalyst, together with its corresponding cipher text generated with the secret key, cipher text
- **Chosen cipher text**
  - Knows algorithm; cipher text; Cipher text chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
- **Chosen text**
  - Knows algorithm; cipher text; select either plaintext or cipher text to encrypt or decrypt

Security of Cryptography

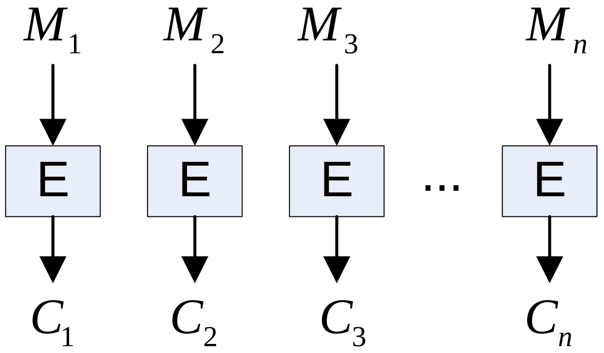
- Unconditional security
  - no matter how much computer power is available, the cipher cannot be broken since the ciphertext provides insufficient information to uniquely determine the corresponding plaintext
- Computational security
  - given limited computing resources (e.g. time needed for calculations is greater than age of universe), the cipher cannot be broken
  - No encryption algorithm is unconditionally secure, hence the users of the algorithm shall check for the following criteria to be satisfied
    - Cost of breaking the cipher exceeds the value of the encrypted information
    - Time required to break the cipher exceeds the useful lifetime of the information

Cryptographic Methods


Symmetric Key – Encryption
Block Ciphers
Stream Ciphers
Transposition Cipher
Substitution Cipher
Poly-alphabetic Substitution
Mono-alphabetic Substitution
Asymmetric Key (Public Key)– Encryption
RSA Public -Key
Elliptic Curve Cryptography
Diffie-Hellman Key Exchange

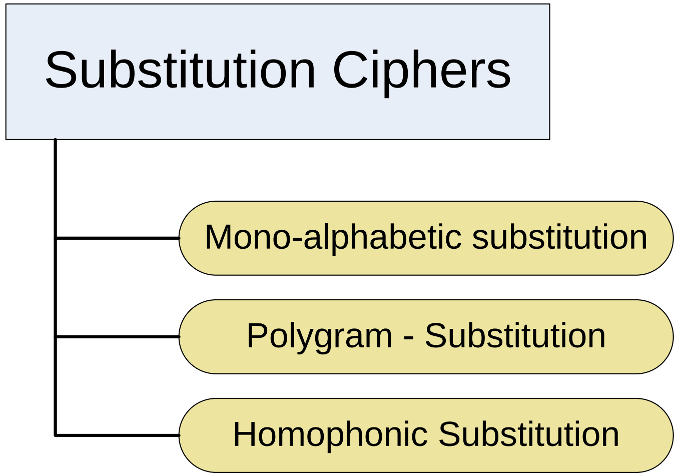
Block Ciphers

- transforms a fixed-length block of *plaintext* data into a block of *cipher text* data of the same length
- transformation takes place under the action of a user-provided secret key
- decryption is performed by applying the reverse transformation to the cipher text block using the same secret key
- the fixed length is called the block size, and for many block ciphers, the block size is 64 bits
- most modern ciphers are of this form e.g. DES, 3-DES, IDEA, AES, and Blowfish



Substitution Ciphers

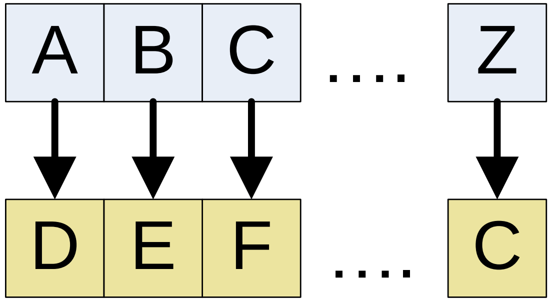
- units of plaintext are substituted with cipher text according to a regular system
- "units" may be single letters (the most common), pairs of letters, triplets of letters, mixtures of the above, and so forth
- receiver deciphers the text by performing an inverse substitution



Substitution Ciphers

- Ciphertext and plaintext character sets are the same.
- Let $m = m_1 m_2 m_3.....$ be the plaintext message and the alphabet $\mathrm{A} = \{A, B, ..., Z\}$
- The substitution employs a permutation *e* over A
- **Example**: Shift cipher (Caesar – Cipher)
  - Alphabetic shift through k characters for some fixed k
  - More precisely, if $|\mathrm{A}| = s$ and plaintext m_i is associated with the integer value p, 0<=p<=s-1,
  - then $c_i = e(m_i) = (p + k) mod\ s$
  - And the decryption mapping is defined by

    $d(c_i) = (c - k) mod\ s$
  - Caesar used k = 3



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 cipher text bit patterns
- earliest known substitution cipher is **Caesar Cipher**
- Developed by Julius Caesar
- first attested use in military affairs
- replaces each letter by 3rd letter on
- example:

```
meet me after the toga party
PHHW PH DIWHU WKH WRJD SDUWB
```

Caesar Cipher

- can define transformation as:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

- mathematically give each letter a number

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12
n	o	p	q	r	s	t	u	v	w	x	y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

- then have Caesar cipher as:

$$C = \mathrm{E}(p) = (p + k) \bmod (26)$$

$$p = \mathrm{D}(\mathrm{C}) = (\mathrm{C} - k) \bmod (26)$$

Cryptanalysis of Caesar Cipher

- only have 26 possible ciphers
    - A maps to A,B,..Z
- could simply try each in turn
- a **brute force search**
- given cipher text, just try all shifts of letters
- do need to recognize when have plaintext
- eg. break cipher text "GCUA VQ DTGCM"

Mono alphabetic Cipher

- rather than just shifting the alphabet
- could shuffle (jumble) the letters arbitrarily
- each plaintext letter maps to a different random cipher text letter
- hence key is 26 letters long

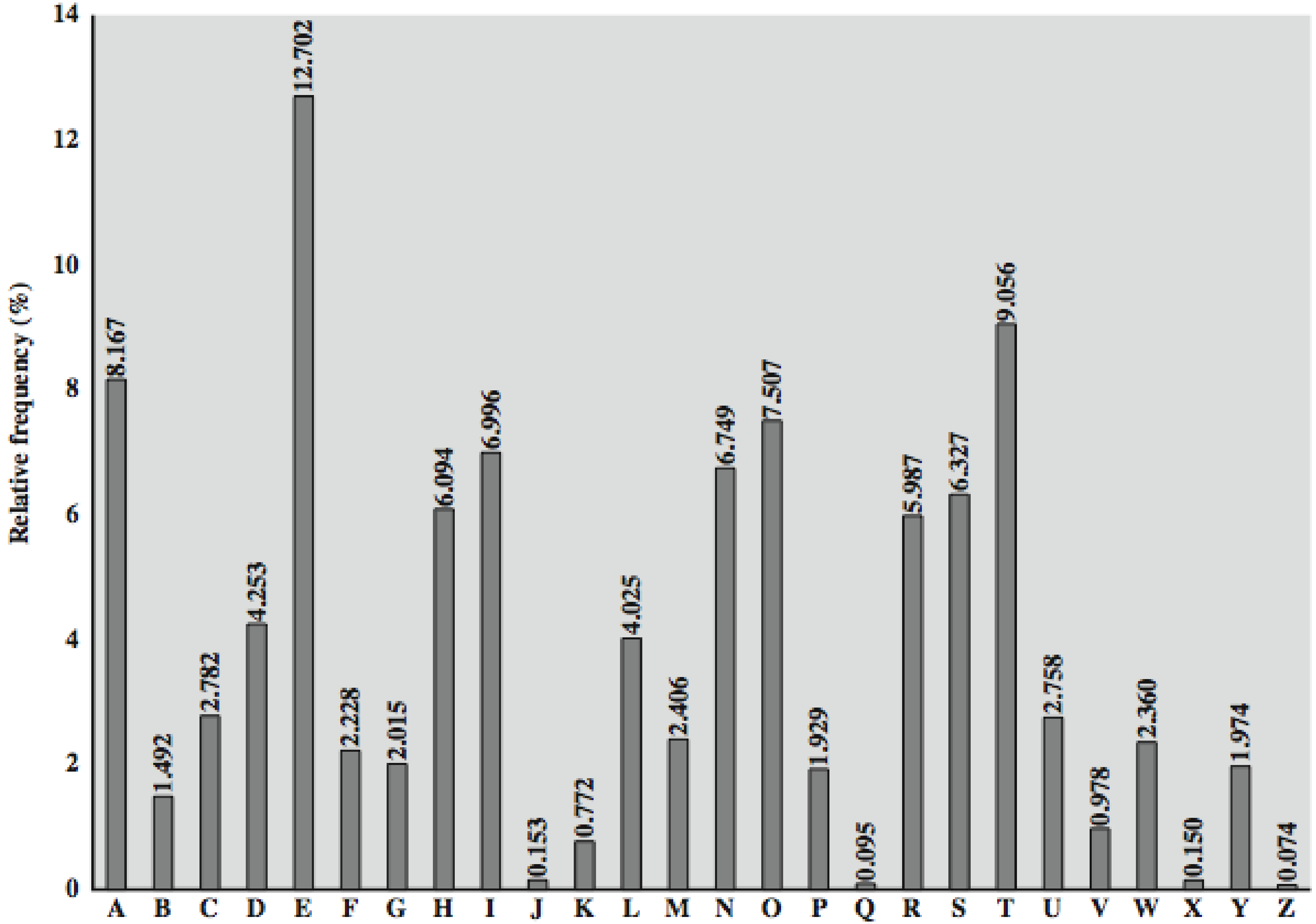
```
Plain:  abcdefghijklmnopqrstuvwxyz
Cipher: DKVQFIBJWPESCXHTMYAUOLRGZN
Plaintext:  ifwewishtoreplaceletters
Ciphertext: WIRFRWAJUHYFTSDVFSFUUFYA
```

- Security of Monoalphabetic Cipher
  - now have a total of 26! = 4 x 10^{26} keys
  - with so many keys, might think is secure
  - but would be **!!!WRONG!!!**
  - problem is language characteristics

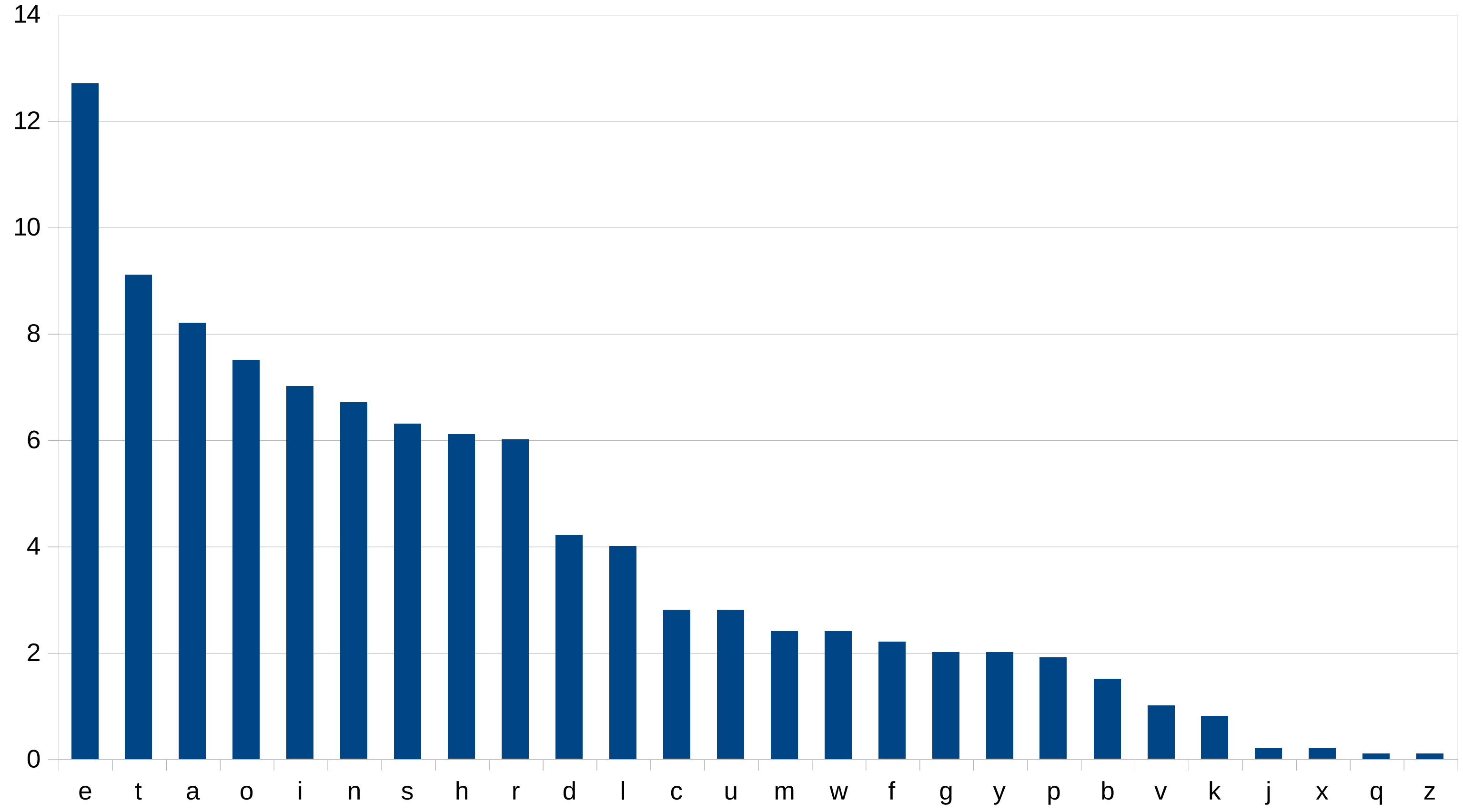
Language Redundancy and Cryptanalysis

- human languages are **redundant**
- eg "th lrd s m shphrd shll nt wnt"
- letters are not equally commonly used
- in English **E** is by far the most common letter
- then *T,A,O,I,N,S,H*
- other letters are fairly rare
- cf. *Z,J,K,Q,X*
- have tables of single, double & triple letter frequencies

English Letter Frequencies



Sorted Frequencies of Letters in English



Use in Cryptanalysis

- key concept - monoalphabetic substitution ciphers do not change relative letter frequencies
- discovered by Arabian scientists in 9th century
- calculate letter frequencies for ciphertext
- compare counts/plots against known values
- if caesar cipher look for common peaks/troughs
  - peaks at: A-E-I triple, N-O pair, R-S-T triple
  - troughs at: J-K, U-V-W-X-Y-Z
- for monoalphabetic must identify each letter
  - tables of common double/triple letters help
  - (digrams and trigrams)
- amount of ciphertext is important – statistics!

Example Cryptanalysis

- given ciphertext:
  - UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
  - VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
  - EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
- count relative letter frequencies (see text)

Example Cryptanalysis

- given ciphertext:
  - UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
  - VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
  - EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
- guess P & Z are e and t
- guess ZW is th and hence ZWP is “the”
- proceeding with trial and error finally get:
  - it was disclosed yesterday that several informal but
  - direct contacts have been made with political
  - representatives of the viet cong in moscow

Substitution Ciphers – Polygram

- Involves groups of characters being substituted by other groups of characters

- Sequences of two plaintext characters (digrams) may be replaced by other digrams

- Same may done with trigrams, generally n-grams

- key space is extremely large: in full digram substitution over an alphabet of 26 characters, there are $\left(26^2\right)!$ possible keys

- first practical historical use in 1854 by Sir Charles Wheatstone

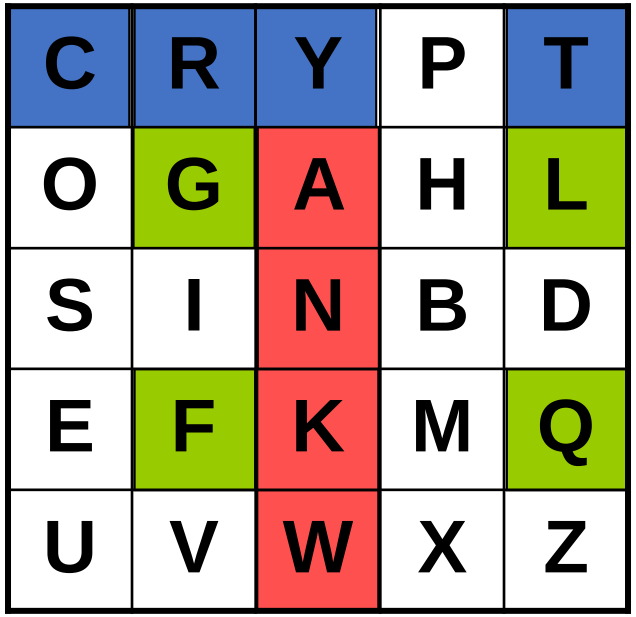
  → ***Playfair - Cipher***

Playfair - Cipher

- usual form with 5 by 5 table and a key word or phrase
- Fill in the spaces in the table with the letters of the key; duplicate letters are dropped; I = J; fill the remaining spaces
- 4 rules to encrypt each pair of letters:
  - *if letters of a pair are both the same, add an X after the first letter, encrypt new pairs*
  - *If letters appear on the same row, replace them with the letters to their immediate right (with wrapping)*
  - *If letters appear on the same column, replace them with the letters immediately below (also with wrapping)*
  - *If letters are not on the same row or column, replace them with the letters on the same row respectively but at the other pair of corners of the rectangle by the original pair*

Playfair – Cipher (2)

- **Example:** Key-Word CRYPTOGRAPHY CLASS SPRING



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

Rule 2: letters on same line

RT → YC

Rule 3: letters on same column

AK → NW

Rule 4: letters form rectangle

GQ → LF

Security of the Playfair Cipher

- Security much improved over mono alphabetic
- Since have 26 x 26 = 676 digrams
- would need a 676 entry frequency table to analyse (verses 26 for a mono alphabetic) and correspondingly more cipher text
- was widely used for many years (eg. US & British military in WW1)
- It **can** be broken, given a few hundred letters
- Since still has much of plaintext structure

Cryptanalysis of Playfair

- **Identification:**
  - Cipher message contains an even number of letters
  - Only 25 letters present

- **Attack:**
  - Ultimate goal: recover the 5 x 5 matrix → general structure is known
  - Known Plaintext: probable words lead to partial decryption
  - Digram frequency count:

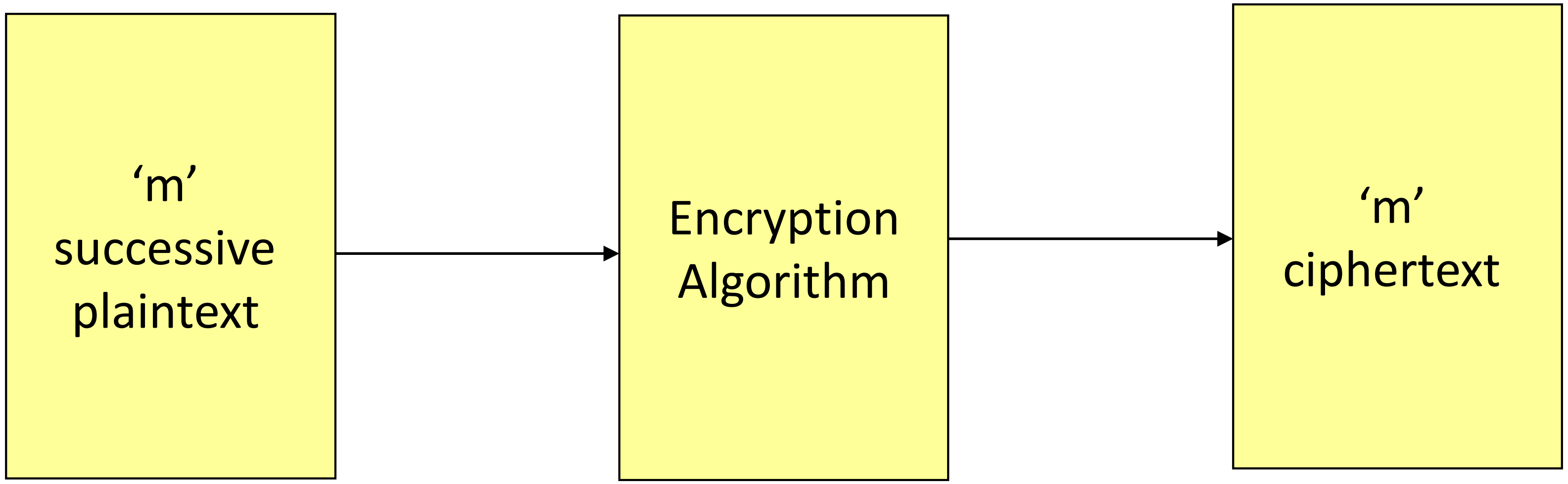
TH	3.15 %
HE	2.51 %
AN	1.72 %
IN	1.69 %

ER	1.54 %
RE	1.48 %
ES	1.45 %
ON	1.45 %

EA	1.31 %
TI	1.28 %
AT	1.24 %
ST	1.21 %

Hill Cipher

- Another interesting multiletter cipher, developed by the mathematician Lester Hill in 1929.



‘m’ successive plaintext

Encryption Algorithm

‘m’ ciphertext

- The substitution is determined by ‘m’ linear equations in which each character is assigned a numerical value (a=0, b=1,…z=25). For m=3, the system can be described as follows:

$c1 = (k_{11}p_1 + k_{12}p_2 + k_{13}p_3) \bmod 26$

$c2 = (k_{21}p_1 + k_{22}p_2 + k_{23}p_3) \bmod 26$

$c3 = (k_{31}p_1 + k_{32}p_2 + k_{33}p_3) \bmod 26$

- Expressed in term of column vectors and matrices:

$$\begin{pmatrix} c1 \\ C2 \\ C3 \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \bmod 26$$

(or)

$$C = \mathbf{KP} \bmod 26$$

- Where C & P are column vectors of length 3, representing the plaintext and ciphertext, and K is a 3 x 3 matrix – encryption key

- Consider the plaintext “pay mor emo ney”

- K = $\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$

- The first 3 letters of the plaintext “PAY” are represented by the vector

$$\begin{pmatrix} 15 \\ 0 \\ 24 \end{pmatrix}$$

- Then C $\begin{pmatrix} 15 \\ 0 \\ 24 \end{pmatrix} = \begin{pmatrix} 375 \\ 819 \\ 486 \end{pmatrix}$ mod 26 = $\begin{pmatrix} 11 \\ 13 \\ 18 \end{pmatrix}$ = LNS

- Decryption requires using the inverse of the matrix K.
- Whether inverse of a matrix exist for all?
- In this case, the inverse is

$$K^{-1} = \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix}$$

- This is demonstrated as follows:

$$\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} = \begin{pmatrix} 443 & 442 & 442 \\ 858 & 495 & 780 \\ 494 & 52 & 365 \end{pmatrix} \text{ mod } 26 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- In general terms, the Hill Cipher can be written as follows:

  $C = E(K,P) = KP \bmod 26$ ⇔ Encryption Algorithm

  $P = D(K,P) = K^{-1}C \bmod 26 = K^{-1}KP = P$ ⇔ Decryption Algorithm

Polyalphabetic Ciphers

- Another approach to improving security is to use multiple cipher alphabets
- called **polyalphabetic substitution ciphers**
- makes cryptanalysis harder with more alphabets to guess and flatter frequency distribution
- use a key to select which alphabet is used for each letter of the message
- use each alphabet in turn
- repeat from start after end of key is reached

Vigenère Cipher

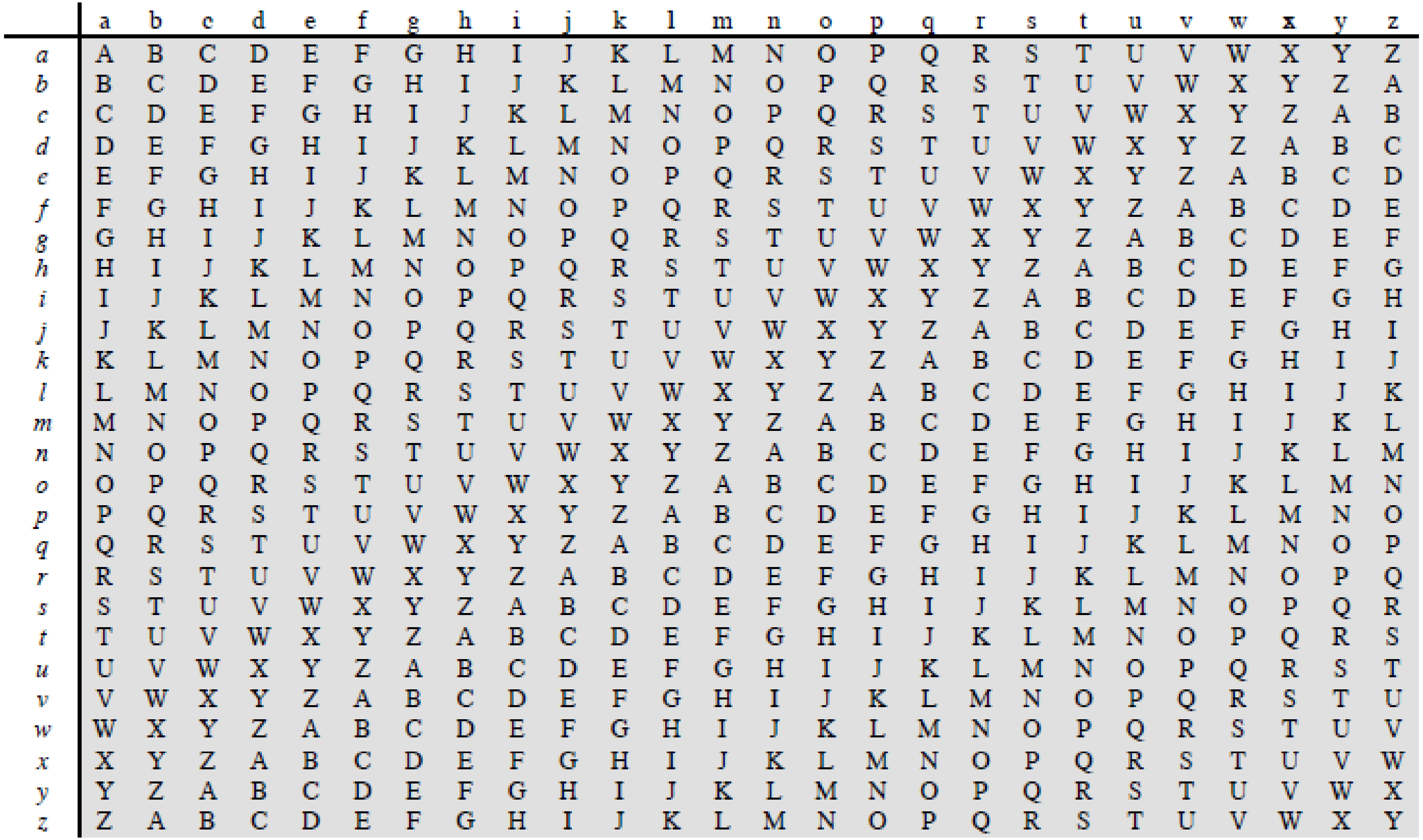
- simplest poly alphabetic substitution cipher is the **Vigenère Cipher**
- effectively multiple caesar ciphers
- key is multiple letters long K = k1 k2 ... kd
- ith letter specifies ith alphabet to use
- use each alphabet in turn
- repeat from start after d letters in message
- decryption simply works in reverse
- In the simplest case, the different alphabets are used sequentially and repeated
- Position of plaintext character determines which mapping is applied → the same plaintext character is encrypted to different ciphertext characters

Example

- write the plaintext out
- write the keyword repeated above it
- use each key letter as a caesar cipher key
- encrypt the corresponding plaintext letter
- eg using keyword *deceptive*

```
key:       deceptivedeceptivedeceptive
plaintext: wearediscoveredsaveyourself
ciphertext:ZICVTWQNGRZGVTWAVZHCQYGLMGJ
```

Table 2.3 The Modern Vigenère Tableau



	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
b	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
c	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
k	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Security of Vigenère Ciphers

- have multiple ciphertext letters for each plaintext letter
- hence letter frequencies are obscured, but not totally lost
- a considerable frequency information remains
- First suppose that the opponent believes that the cipher text was encrypted using either monoalphabetic substitution or a Vignere cipher.
- If it is monoalphabetic substitution, then the statistical properties of the cipher text should be the same as that of the plain text.

- On the other hand, if a Vignere cipher is suspected, then

  - determine the length of the keyword – how the length is found?

  - the important insight is,

  if 2 identical sequences of plain text letters occur at a distance that is an integer multiple of the keyword length, they will generate identical cipher text sequences.

```
key:       deceptivedeceptivedeceptive
plaintext: wearediscoveredsaveyourself
```

- What an analyst will do now?
- Detect the repeated sequences at a displacement of 9 and make the assumption that the keyword is either 3 or 9 letters in length.
- If the message is long enough , there will be a number of such repeated cipher text sequences.
- By looking for common factors in the displacements of the various sequences, the analyst should be able to make a good guess of the keyword length.
- With the keyword DECEPTIVE, the letters in positions 1, 10, 19, and so on are all encrypted with the same mono-alphabetic cipher.
- Thus known frequency characteristics of the plain text language to attack is used.
- How to find a solution?

- Periodic nature of the keyword can be eliminated by using a non repeating keyword that is as long as the message itself.
- Vignere proposed an **autokey system ,** in which a keyword is concatenated with the plain text itself to provide a running key.

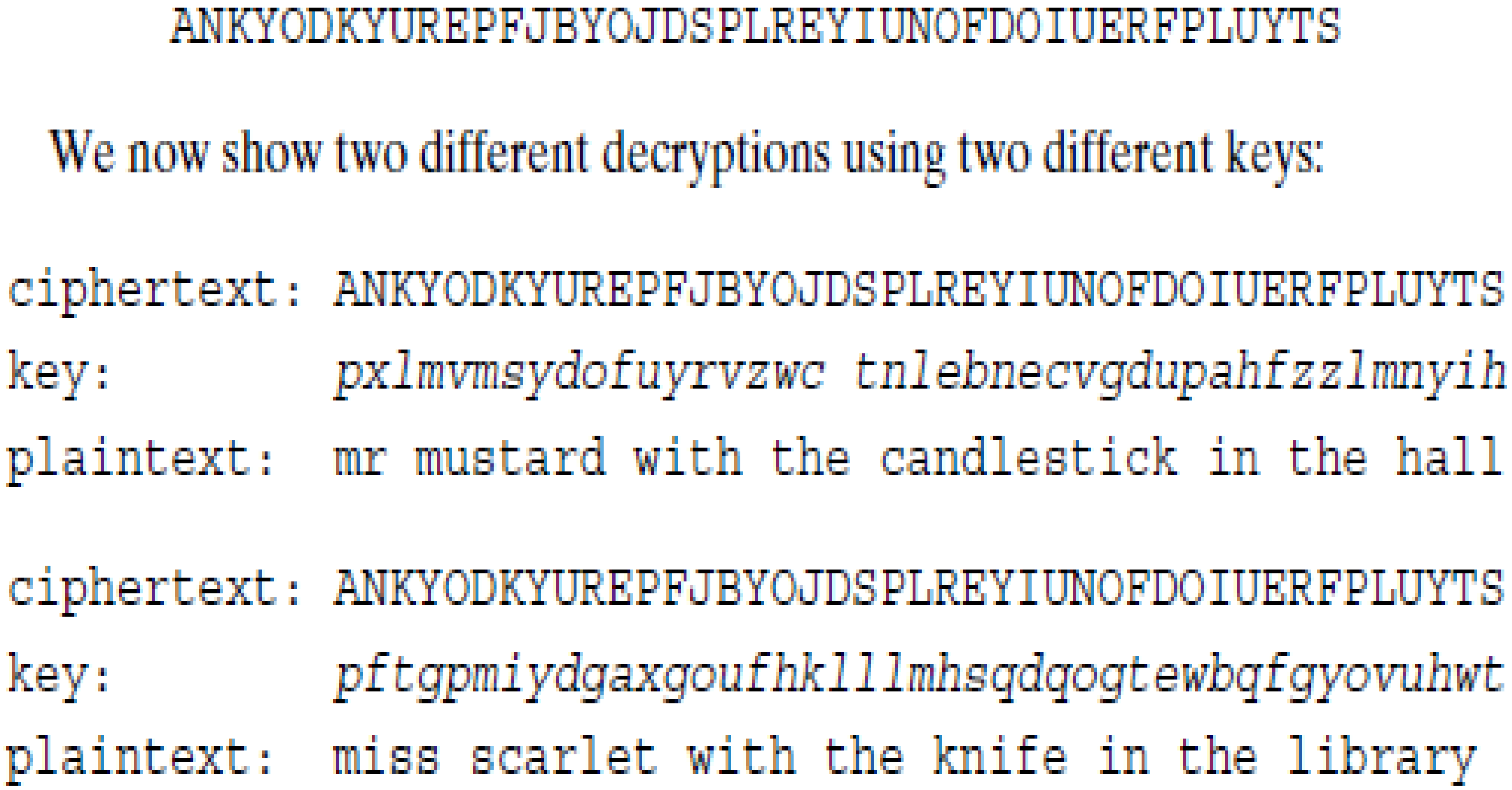
Key:	d**e**ceptivewear**e**discoveredsav
Plain text:	w**e**arediscover**e**dsaveyourself
Ciphertext:	Z**I**CVTWQNGKZEI**I**GASXSTSLVVWLA

- Even this scheme is vulnerable to cryptanalysis. Because the key and the plaintext share the same frequency distribution of letters. For eg: e encrypted by e, etc.
- These regularities can be exploited to achieve successful cryptanalysis.

One Time Pad

- Proposed by Joseph Mauborgne
- Suggested using random key that is as long as the message. So that the key need not be repeated.
- Key is used to Encrypt and decrypt a single message and then discarded.
- Each new message requires a new key of same length as the new message.
- Such a scheme, known as a **one-time pad**.

One Time Pad



ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS

We now show two different decryptions using two different keys:

```
ciphertext: ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS
key:        pxlmvmsydofuyrvzwc tnlebnecvgdupahfzzlmnyih
plaintext:  mr mustard with the candlestick in the hall

ciphertext: ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS
key:        pftgpmiydgaxgoufhklllmhsqdqogtewbqfgyovuhwt
plaintext:  miss scarlet with the knife in the library
```

Security of One Time pad

- Entirely due to the randomness of the key.
- If the stream of characters that constitute the key is truly random, then the stream of characters that constitute the Ciphertext will be truly random.
- Thus, there are no patterns or regularities that a cryptanalyst can use to attack the cipher text.

Limitations of one time Pad

- In Practice, 2 fundamental difficulties are.

1) Making large quantities of random keys.
2) Even more daunting is the problem of key distribution and protection.

Hence, it is of limited utility and is useful primarily for low bandwidth channels requiring very high security.

Onetime pad Cryptosystem - “Perfect secrecy”