Cryptography and Network Security I

1st Lecture

Introduction to Cryptography and Information Security

By

Dr. Taqwa F. Hassan

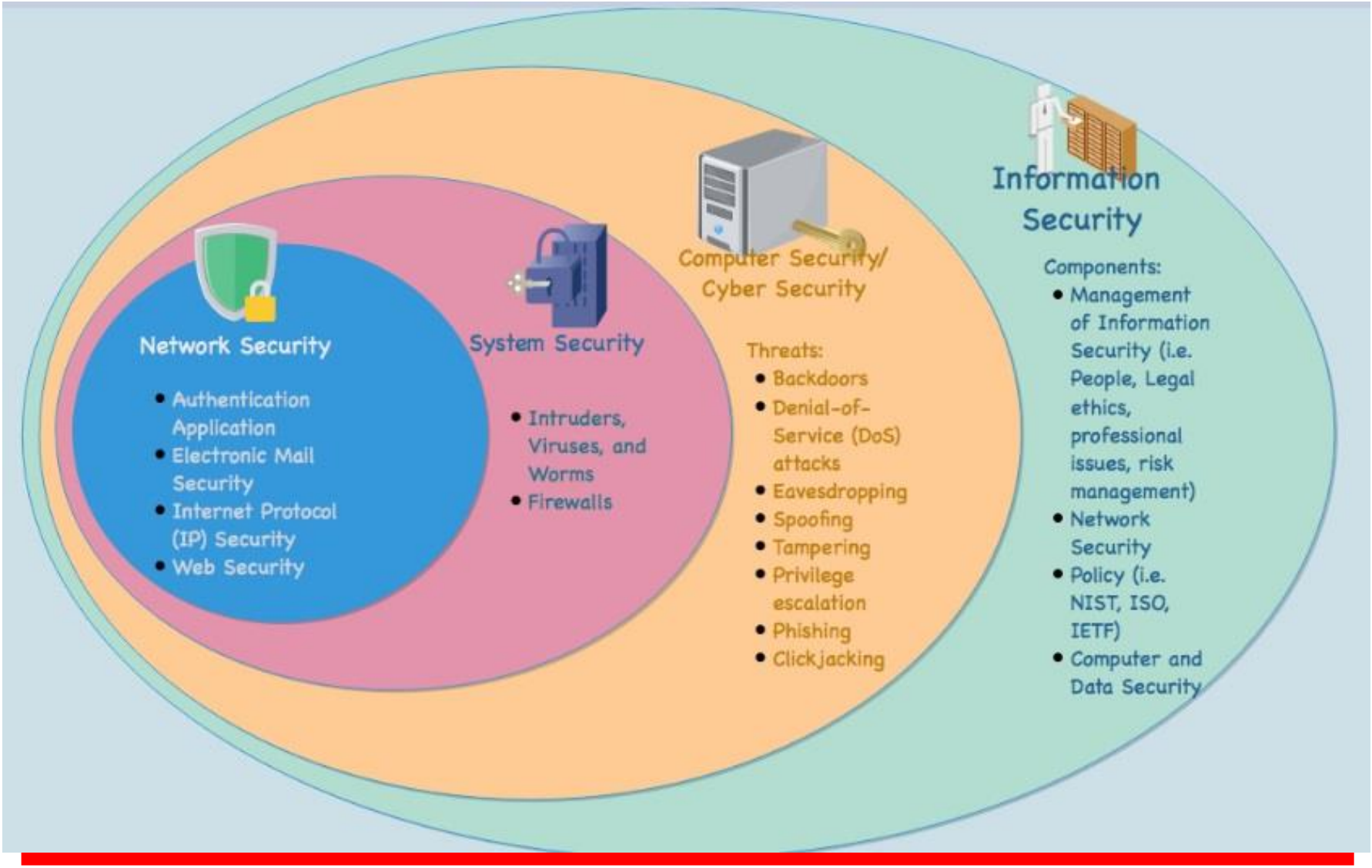
Edited by

Dr. Ali Albu-Rghaif

Department of Computer Engineering

College of Engineering - University of Diyala

Security


Network Security
• Authentication Application
• Electronic Mail Security
• Internet Protocol (IP) Security
• Web Security
System Security
• Intruders, Viruses, and Worms
• Firewalls
Computer Security/ Cyber Security
Threats:
• Backdoors
• Denial-of-Service (DoS) attacks
• Eavesdropping
• Spoofing
• Tampering
• Privilege escalation
• Phishing
• Clickjacking
Information Security
Components:
• Management of Information Security (i.e. People, Legal ethics, professional issues, risk management)
• Network Security
• Policy (i.e. NIST, ISO, IETF)
• Computer and Data Security

Information Security

Information security, or InfoSec, includes the tools and processes for preventing, detecting, and remediating attacks and threats to sensitive information, both digital and non-digital. InfoSec is also concerned with documenting the processes, threats, and systems that affect the security of information.

Information security, Cybersecurity & Network security

- **InfoSec** focuses on information, whether digitized or not.
- **Cybersecurity** focuses only on computer systems and their information and does not include non-digital resources.
- **Network security** is a subset of cybersecurity and focuses on protecting the network and its various components.

The information security vs. network security discussion hinges on whether the system is limited to a network or includes other information, including non-digital information.

The Goals/ Services of Information Security

InfoSec seeks to accomplish the following primary objectives (confidentiality, integrity, and availability).

- **Confidentiality:** An information security analyst aims to ensure the information that needs to be kept secret does not get into the wrong hands.

- **Integrity:** Integrity refers to the accuracy and completeness of data. Information security policies aim to make sure data is not just present but is whole and unaltered.

- **Availability:** In addition to being secure, correct, and complete, information has to be readily available to those who need it. Ransomware and other kinds of malware can block users from freely accessing the information they need.

The Goals of Information Security

Confidentiality (CIA) السرية	Integrity (CIA) النزاهة	Availability (CIA) التوفر
Encryption - التشفير إدارة عملية الدخول: • Identification - Username • Authentication - Password • Authorization - Permissions **Steganography - اخفاء البيانات** • الرسائل الخفية داخل المواقع • الرسائل الخفية داخل الملفات والصور	**التأكد من عدم العبث بالبيانات** التشفير إنشاء كود مشتق من خلال خوارزمية، إذا تم تغيير البيانات ، فسيتم تغيير الكود في المستقبل أيضًا Digital Signatures, Certificates, and Non-Repudiation إرسال توقيع رقمي فريد، بتوضيح من أرسل الرسالة ومن يسمح للمستلم بقراءتها PKI - Public Key Infrastructure تمكن التوقيعات والشهادات للعمل من خلال الحفاظ مفاتيح التشفير وإدارة الشهادات	**ضمان إمكانية استرداد البيانات عند الحاجة إليها** • أي فشل SPOF يتوقف النظام بأكمله عن العمل • Disk Redundancy تكرار القرص • Server Redundancy تكرار الخوادم • Load Balancing - توزيع الحمل • Site Redundancies - الزيادات في الموقع • Backups - النسخ الاحتياطية • Alternate Power - الطاقة البديلة • Cooling Systems - تبريد النظام • Patching - الترقيع

What is Computer Security

The Protection of the assets of a computer system

- **Hardware**
- **Software**
- **Data**

An asset is anything that needs to be protected, this could be:

- **Information** – Examples: medical records, social security numbers, banking data
- **Computer System** – Examples: defense systems, critical infrastructure
- **Service** – Examples: Websites, life/safety systems
- **Facilities** that house any of the 3 above

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

The Goals of Computer Security

Protect the computer assets from

- Human error, natural disaster, physical and electronic maliciousness.
- Confidentiality, Integrity, Availability

Some terms to be familiar with

- Vulnerability
- Threat
- Attack
- Countermeasure or Control

Vulnerability

A vulnerability is a flaw or weakness in the design or implementation of an asset that could be used by a threat or threat agent to undermine security, and could be:

- **Incorrect configurations of a system**
- **An open port on a networked computer**
- **Poor backup strategy**
- **Poor coding**

Threat/Threat Agent

A threat is any potential violation of security that could cause harm to the asset, and could be:

- **Someone wanting to do harm**
- **An insecure service**
- **Unacknowledged system, service, information**

A threat agent is anyone or anything that wants to do harm or harms an asset:

- **Hackers**
- **Hacktivists**
- **Not malicious entity – Example: someone that accidentally runs into a power pole and knocks out power to a facility**

Threat/Threat Agent

A potential violation of security

- **Someone wanting to do harm**
- **An insecure service**
- **Unacknowledged system, service, information**

A threat agent is anyone or anything that wants to do harm or harms an asset:

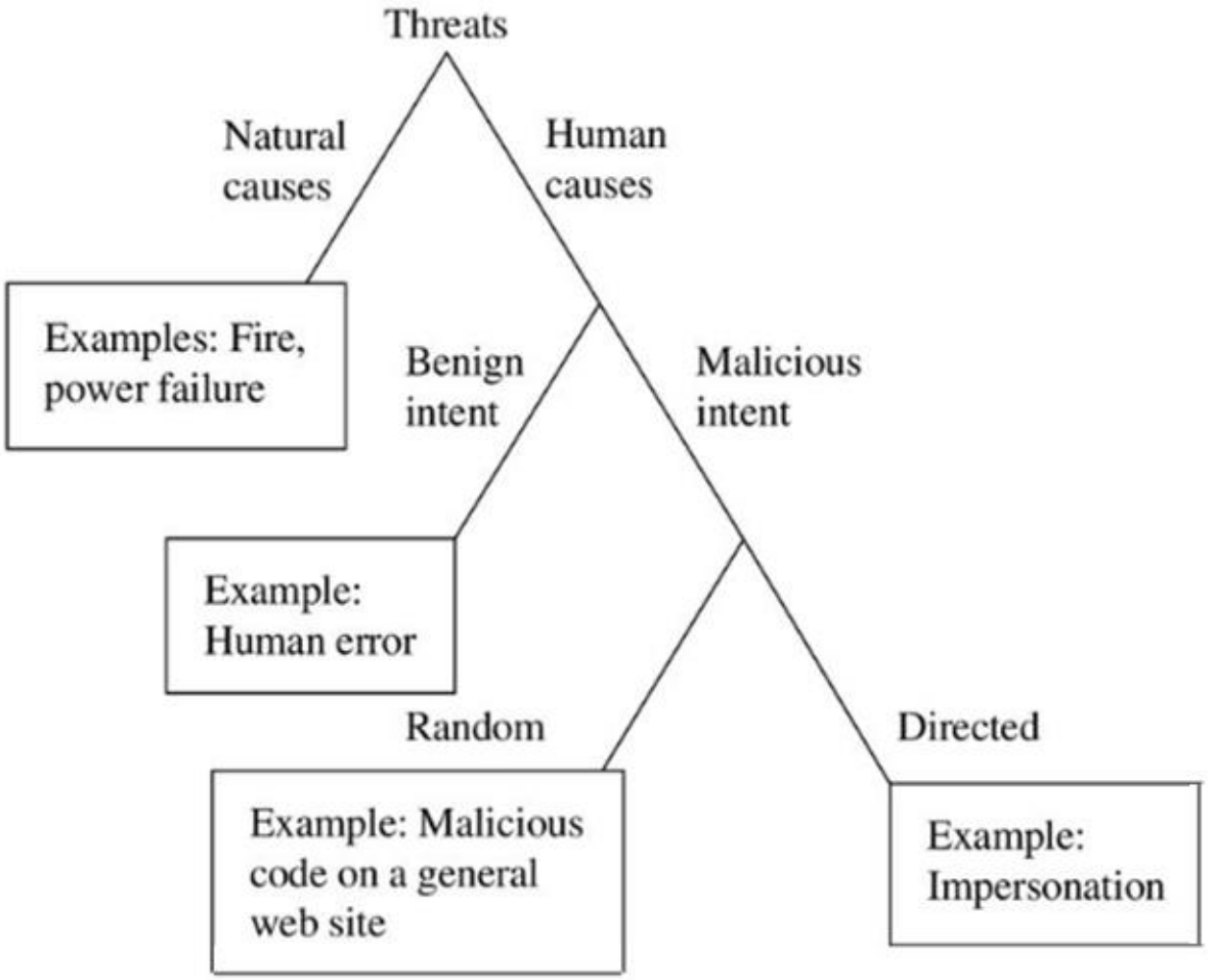
- **Hackers**
- **Hacktivists**
- **Not malicious entity – Example: someone that accidentally runs into a power pole and knocks out power to a facility**

Threat/Threat Agent

A potential violation of security

- **Physical threats:** weather, natural disaster, bombs, power, etc.
- **Human threats:** stealing, trickery, spying, sabotage, accidents.
- **Software threats:** viruses, Trojan horses, logic bombs.

Threat/Threat Agent



Threat/Threat Agent

One way to analyze harm is to consider the cause or source. We call a potential cause of harm a threat. Harm can be caused by either nonhuman events or humans.

- **Nonhuman threats** include natural disasters like fires or floods; loss of electrical power; failure of a component such as a communications cable, processor chip, or disk drive; or attack by a wild boar.

Human threats can be either benign (non malicious) or malicious.

- **Non malicious kinds of harm** include someone's accidentally spilling a soft drink on a laptop, unintentionally deleting text, inadvertently sending an email message to the wrong person, and carelessly typing "12" instead of "21" when entering a phone number or clicking "yes" instead of "no" to overwrite a file.

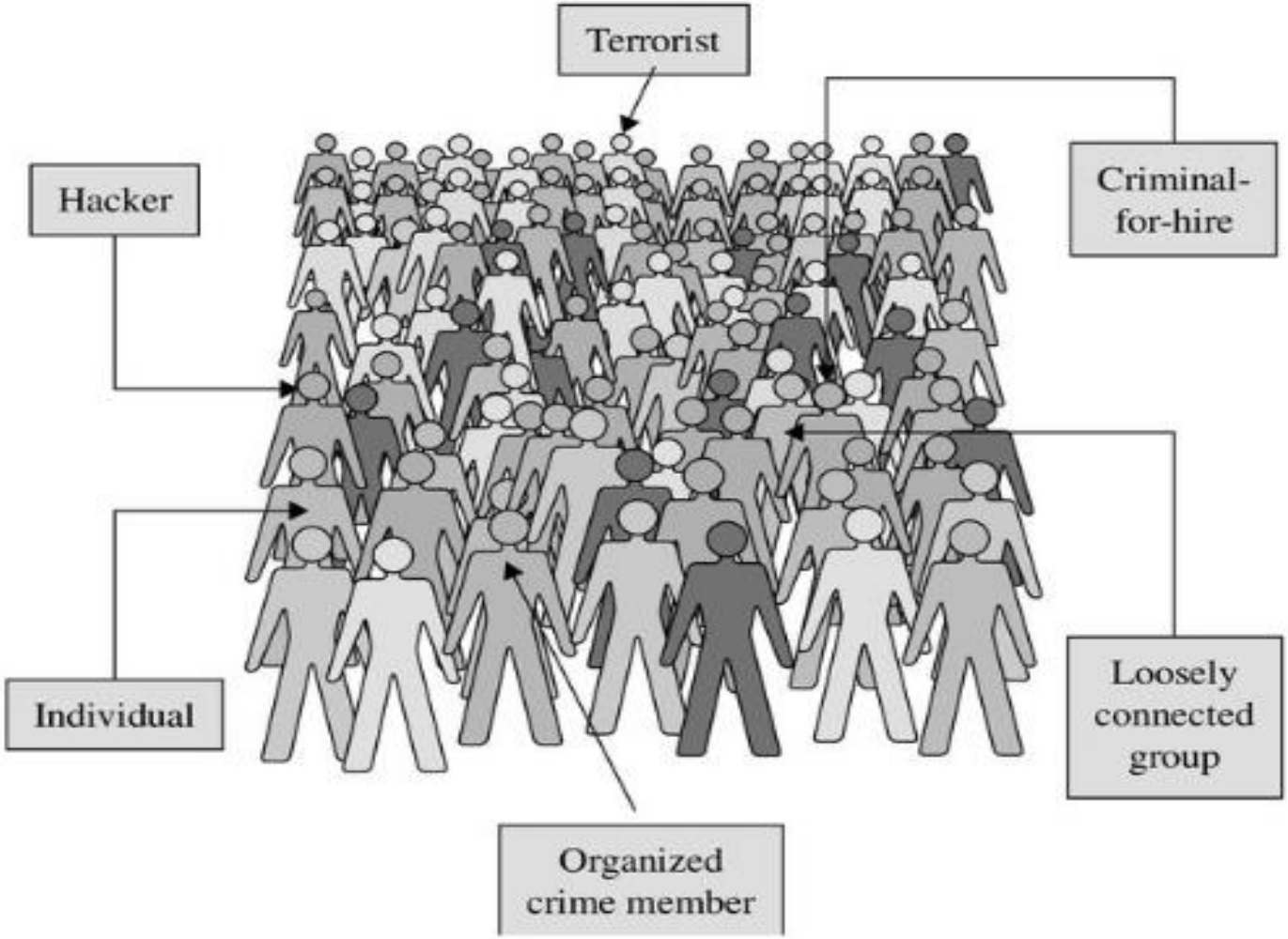
Attack

Most computer security activity relates to malicious, human-caused harm: A malicious person actually wants to cause harm, and so we often use the term attack for a malicious computer security event. Malicious attacks can be random or directed.

- **Random attack** the attacker wants to harm any computer or user; such an attack is analogous to accosting the next pedestrian who walks down the street. An example of a random attack is malicious code posted on a website that could be visited by anybody.

- **Directed attack** the attacker intends harm to specific computers, perhaps at one organization (think of attacks against a political organization) or belonging to a specific individual (think of trying to drain a specific person's bank account, for example, by impersonation). Another class of directed attack is against a particular product, such as any computer running a particular browser.

Types of Attacks



Types of Attacks

Individuals

Originally, computer attackers were individuals, acting with motives of fun, challenge, or revenge. Early attackers acted alone. Two of the most well known among them are Robert Morris Jr., the Cornell University graduate student who brought down the Internet.

.

Organized, Worldwide Groups

More recent attacks have involved groups of people. An attack against the government of the country of Estonia is believed to have been an uncoordinated outburst from a loose federation of attackers from around the world.

Types of Attacks

Organized Crime

Attackers' goals include fraud, extortion, money laundering, and drug trafficking, areas in which organized crime has a well-established presence. Evidence is growing that organized crime groups are engaging in computer crime. In fact, traditional criminals are recruiting hackers to join the lucrative world of cybercrime.

Terrorists

The link between computer security and terrorism is quite evident. We see terrorists using computers in four ways:

- Computer as target of attack.
- Computer as method of attack.
- Computer as enabler of attack.
- Computer as enhancer of attack

Hacker

Hacker Hat Types

The primary categories include black hat, white hat, gray hat, blue hat, red hat, and green hat hackers. Each type plays a distinct role in the cybersecurity ecosystem, with varying motivations, methods, and impacts. Understanding these different hacker types is essential for grasping the complexities of cybersecurity and the diverse approaches to hacking.

White hat hacker الهاكر الأخلاقي .

Black hat hacker (**Cracker**) الهاكر المفسد

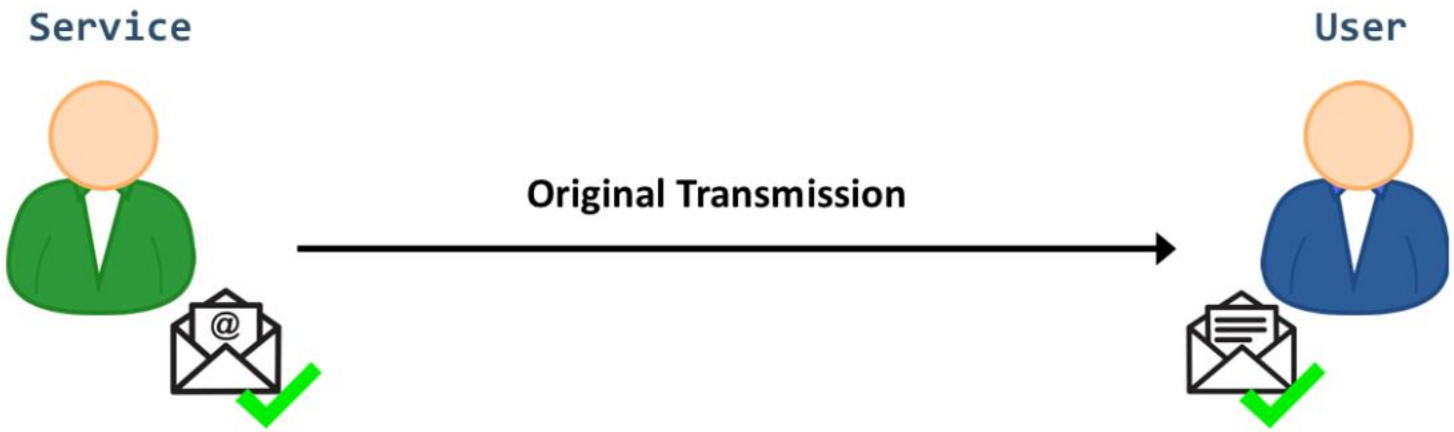
Grey hat hacker.المترنح بين الإصلاح والعبث ،

Hacker

Hacking Steps

Security Attacks Classification

The normal flow of data from the source (S) into the destination (D).

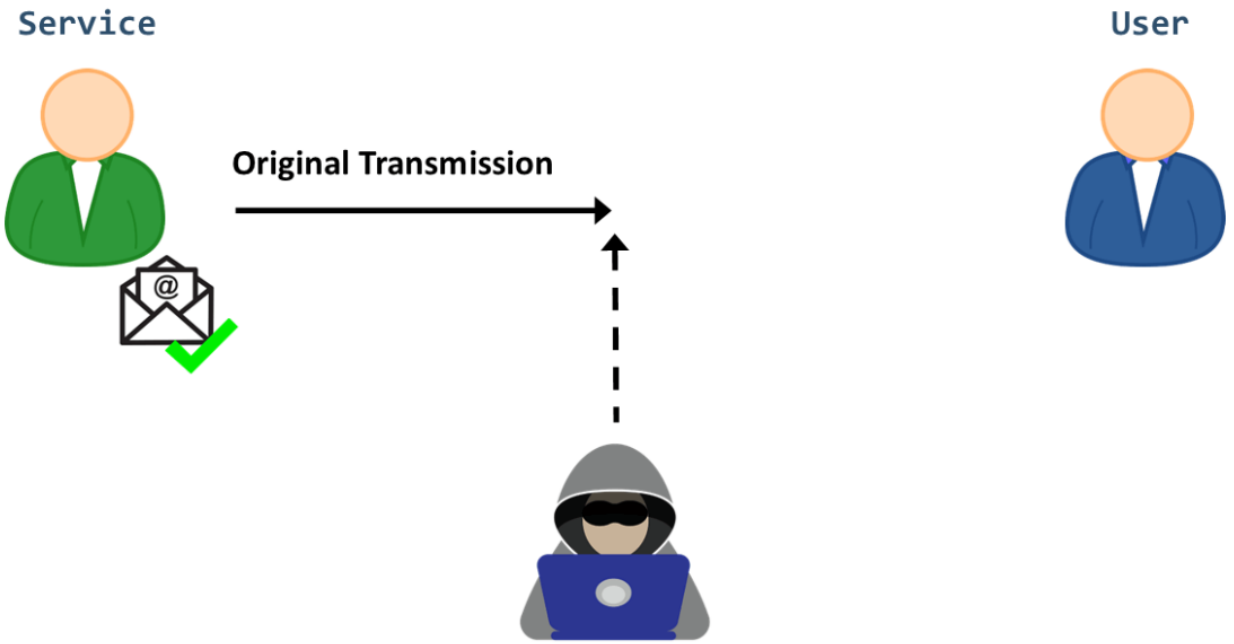


Normal information flow from source to destination

It is possible to identify the attacks on security services as following:

Security Attacks Classification

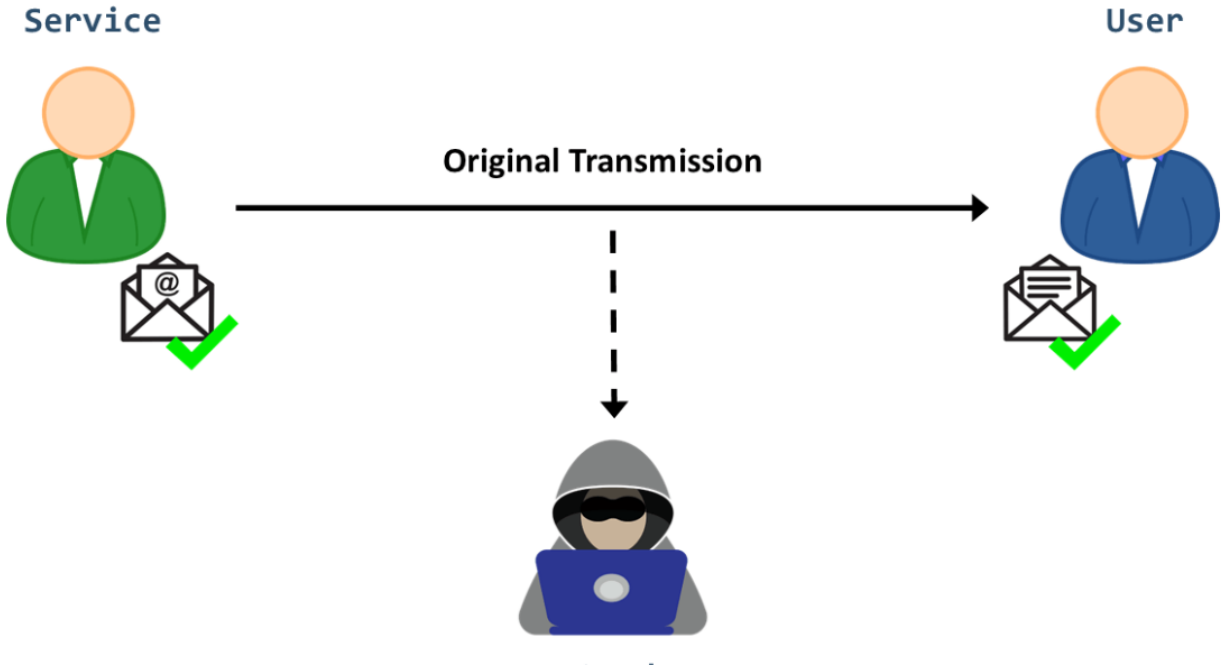
- **Interruption:** it is an attack on the “availability”, it happens when an asset of the system is destroyed or becomes unavailable or unusable.



Interruption of information flow

Security Attacks Classification

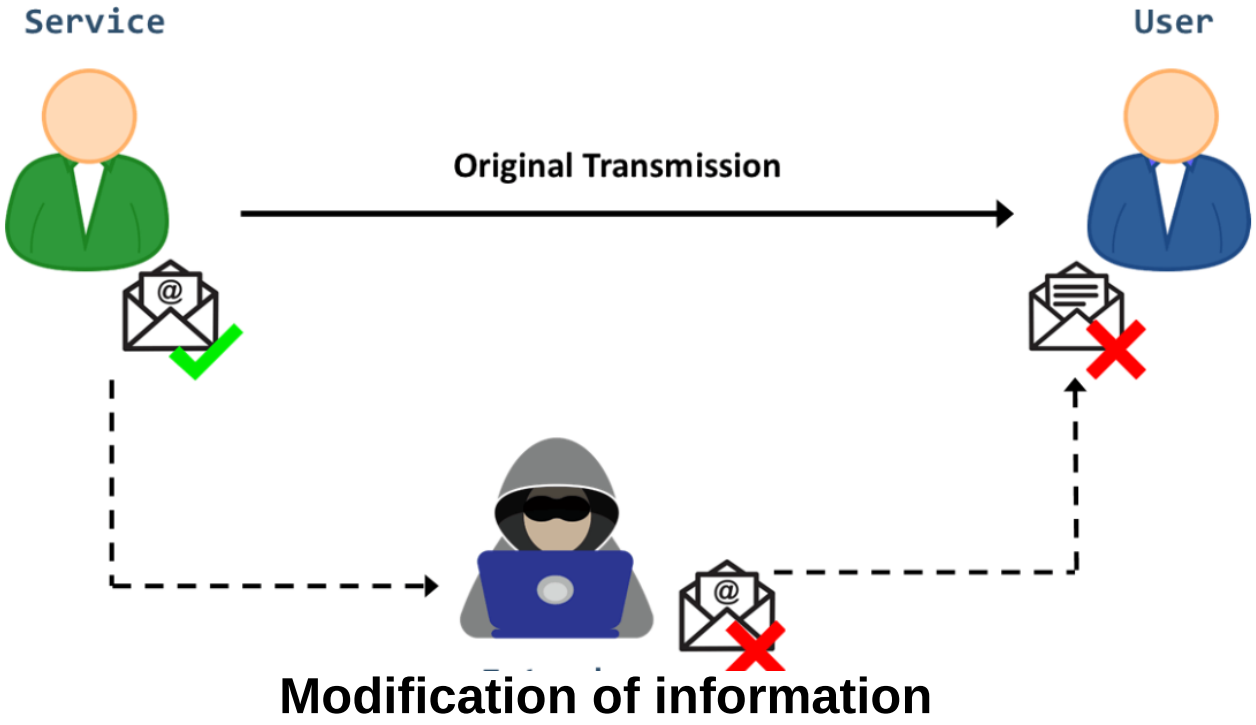
- **Interception:** it is an attack on “confidentiality”, it happens when an unauthorized party gain access into an asset.



Interception of information flow

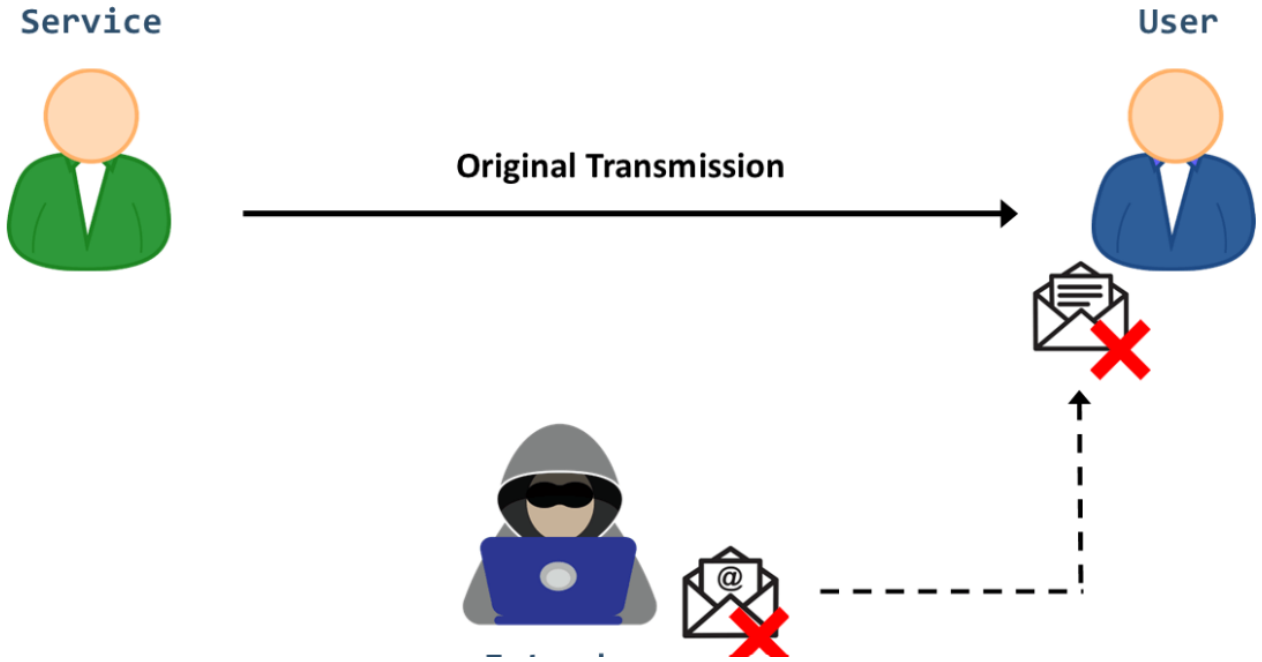
Security Attacks Classification

- **Modification:** it is an attack on “integrity”, it happens when an unauthorized party gain access and has the ability to temper with an asset.



Security Attacks Classification

- **Fabrication:** it is an attack on “authenticity”, it happens when an unauthorized party gain injects bogus data or creates a false trail in the system (changing password).

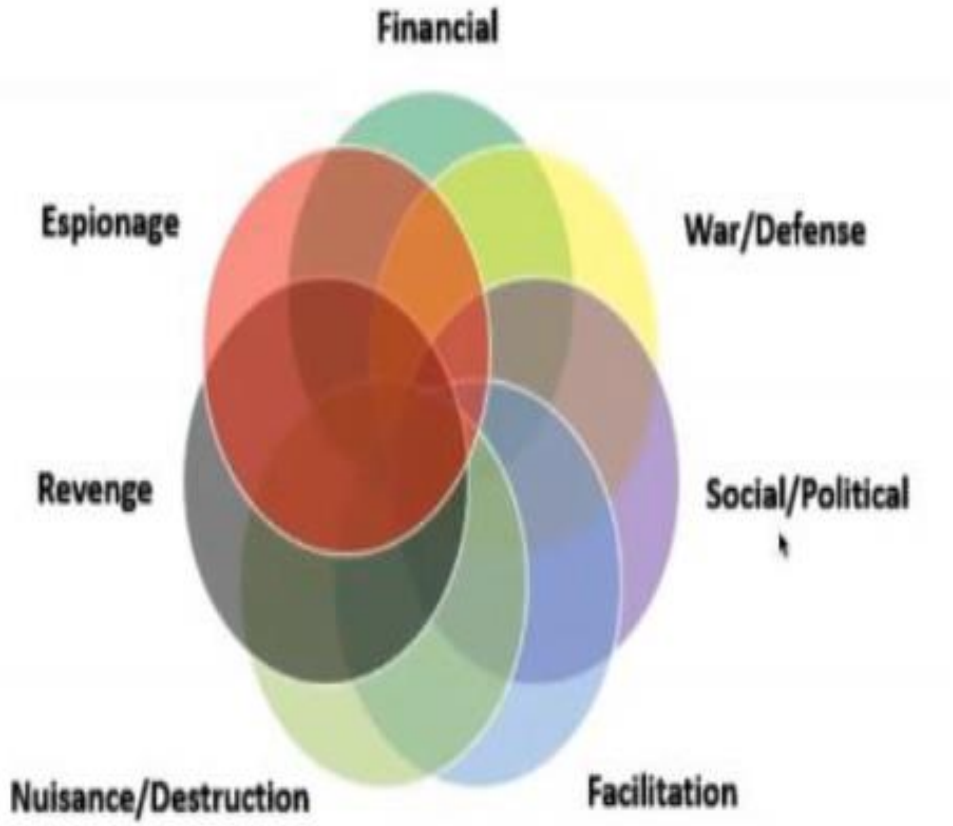


Fabrication of information

Security Attacks Classification

Attacks = Motivate + Method + Vulnerability

الهجوم = الهدف + الطريقة + الثغرة



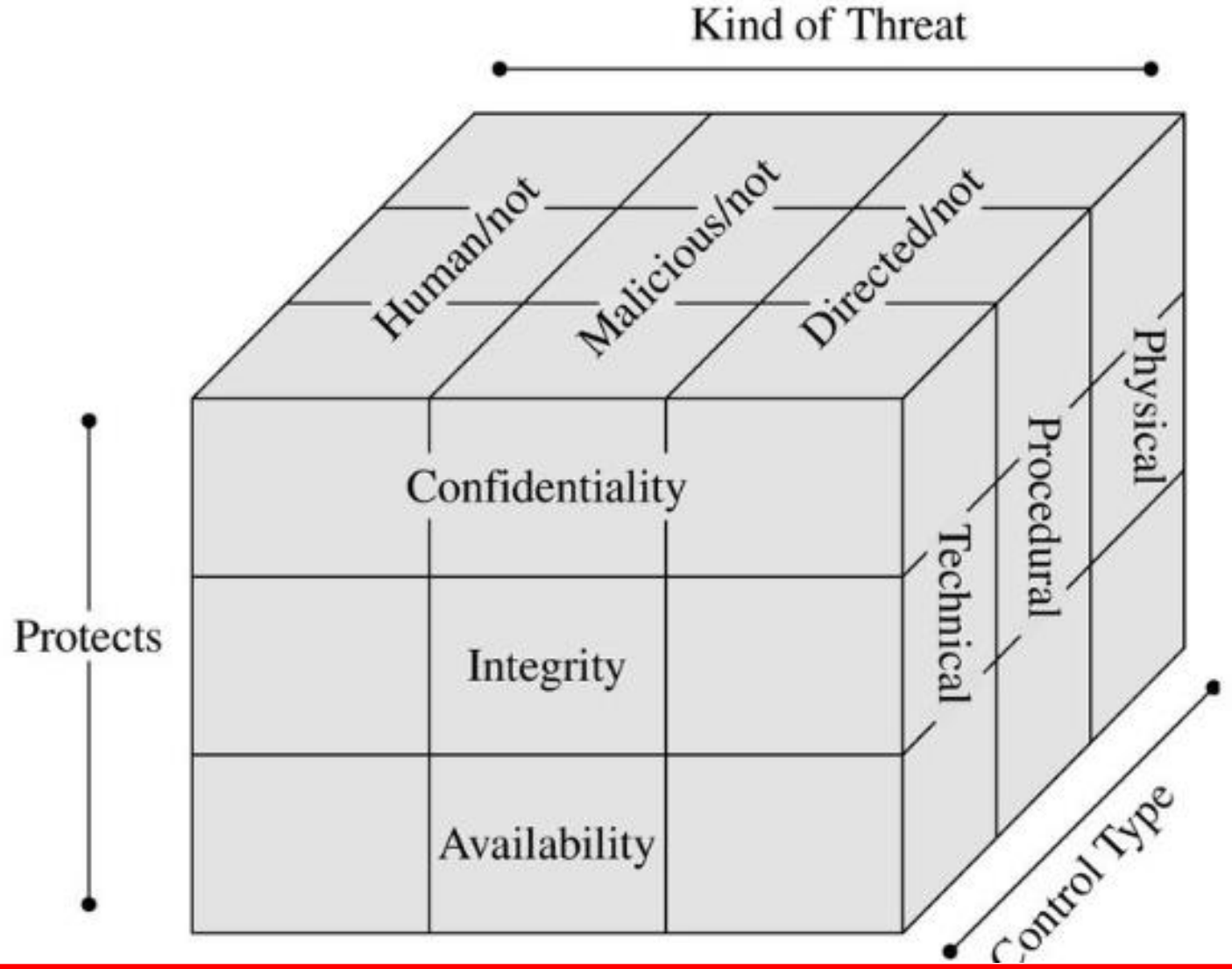
Control

A control or countermeasure is a means to counter threats. Harm occurs when a threat is realized against a vulnerability. To protect against harm, then, we can neutralize the threat, close the vulnerability, or both. The possibility for harm to occur is called risk. We can deal with harm in several ways:

- **prevent** it, by blocking the attack or closing the vulnerability
- **deter** it, by making the attack harder but not impossible
- **deflect** it, by making another target more attractive (or this one less so)
- **mitigate** it, by making its impact less severe
- **detect** it, either as it happens or some time after the fact
- **recover** from its effects

Security professionals balance the cost and effectiveness of controls with the likelihood and severity of harm.

Control


Kind of Threat
Human/not
Malicious/not
Directed/not
Physical
Procedural
Technical
Confidentiality
Integrity
Availability
Protects
Control Type

Types of Control

Physical controls stop or block an attack by using something tangible too, such as walls and fences, locks, (human) guards, sprinklers and other fire extinguishers

- Perimeter - المحيط
  - (الإحاطة سياج كبير مثل الجيش)
- Building - البناء
  - (أربعة جدران وباب كبير مغلق)
- Secure Work Areas - مناطق العمل الآمنة
  - (لا تدع الناس الذين ليس من المفترض أن يكونوا هناك)
- Server and Network Rooms - غرف الخوادم والشبكات
  - يسمح لموظفي التقنية فقط بالدخول واستخدم أقفالا
- Hardware - العتاد
  - أكثر اقفالاً

Physical controls

Doors - الابواب

- أقفال بالتشفير
- بطاقات التعريف الشخصية
- البصمة
- ID شارات الهوية الشخصية
- Tailgating - الملاحقة
- قوائم وسجلات الأحداث والدخول
- الأقفال أقفال كمبيوتر محمول والخزائن وقفل والخزائن
- سياسة المجموعات
- تحديد امتيازات الدخول Least Privilege

Physical controls

- تحديث الجهاز
- استخدام الجدار الناري
- تثبيت برامج ضد الفيروسات
- استخدام برامج ضد التجسس
- استخدام كلمات سر معقدة
- تجاهل البريد العشوائي

- إغلاق الكمبيوتر في حين عدم استخدامه
- عدم فتح أي مرفقات بدون التأكد من مصدرها
- تشفير الملفات المهمة
- استخدام الاتصال الآمن

Types of Control

Procedural or administrative controls use a command or agreement that

- requires or advises people how to act; for example,
- laws, regulations
- policies, procedures, guidelines
- copyrights, patents
- contracts, agreements

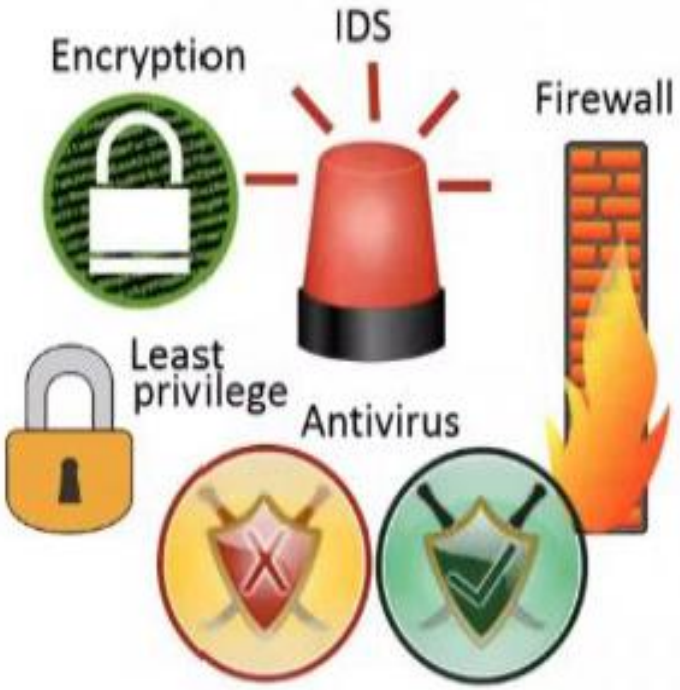
Types of Control

Technical controls counter threats with technology (hardware or software), including

- passwords
- program or operating system access controls
- network protocols
- firewalls, intrusion detection systems
- encryption
- network traffic flow regulators

(Note that the term "**logical controls**" is also used, but some people use it to mean administrative controls, whereas others use it to mean technical controls. To avoid confusion, we do not use that term.)

Technical controls



تثبيت التقنيات بواسطة مسؤول الحماية التلقائية وتقليل من الثغرات الأمنية

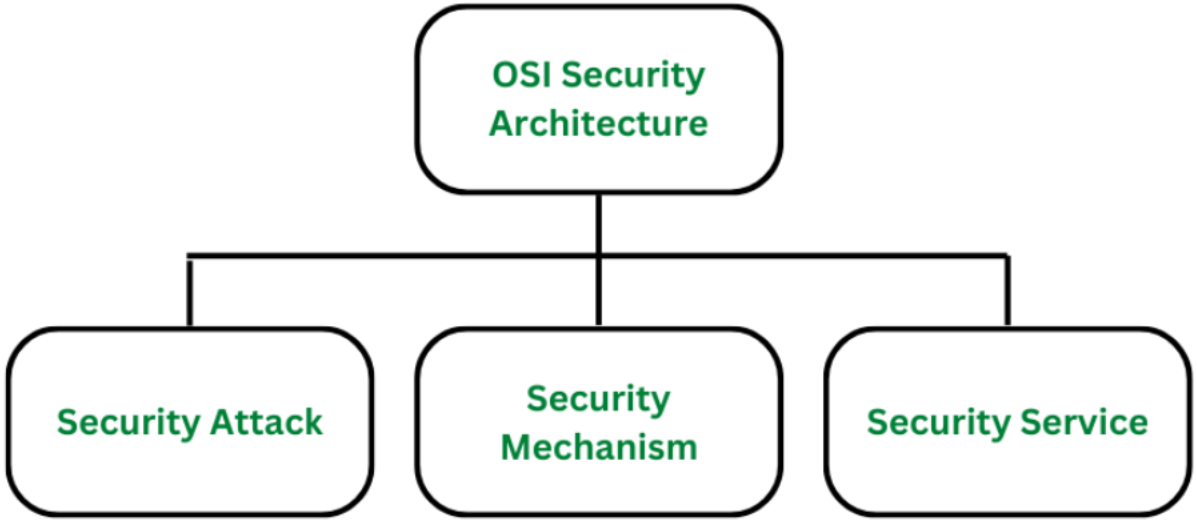
- **Encryption** - التشفير
- **Antivirus Software** - برامج حماية البرمجيات الخبيثة والفيروسات
- **IDSs- Intrusion Detection Software** - برامج كشف التسلل
  - رصد وتقرير عمليات الدخول الى الخوادم
- **Firewalls** - الجدار الناري
  - تقييد حركة مرور الإدخال / الإخراج إلى خادم أو مضيف
- **Least Privilege** - إدارة الإمتيازات
  - السماح فقط لكل مستخدم بالحد الأدنى من الامتيازات التي يحتاجها للحد من المخاطر في حالة حدوث خطأ ما
- **Motion detectors** - نظام كشف التحركات
  - أنظمة إخماد الحرائق وغيرها من الأجهزة

OSI Security Architecture

OSI: (Open Systems Interconnection) security refers to a set of protocols, standards, and techniques used to ensure the security of data and communications in a network environment based on the OSI model.

OSI Security Architecture focuses on these concepts:

- Security Attack
- Security mechanism
- Security Service



OSI Security Architecture

- Security Attack:

  Action that compromises the security.

- Security mechanism

  Detect, prevent, or recover from a security attack.

- Security Service

  Enhance the security, counter security attack, and provide the service.

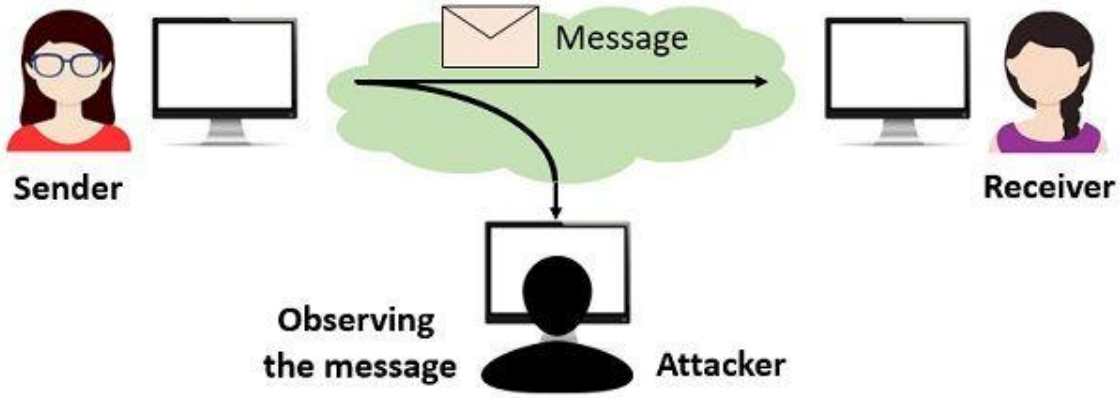
Security Attack

A security attack is an attempt by a person or entity to gain unauthorized access to disrupt or compromise the security of a system, network, or device. These are defined as the actions that put at risk an organization's safety. They are further classified into 2 sub-categories:

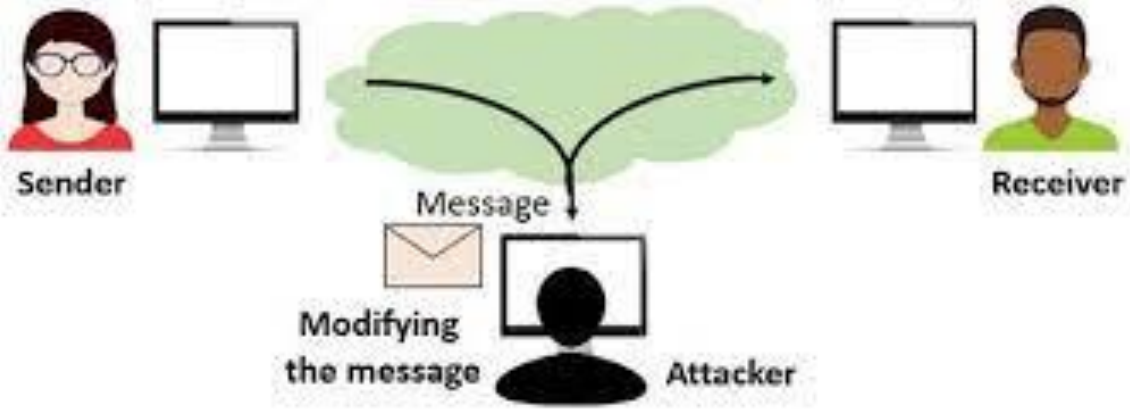
- Passive Attack
- Active Attack

Passive & Active Attacks

A useful means of classifying security attacks, is in terms of **passive** and **active** attacks.



Passive Attack



Active Attack

Passive & Active Attacks

Passive Attacks

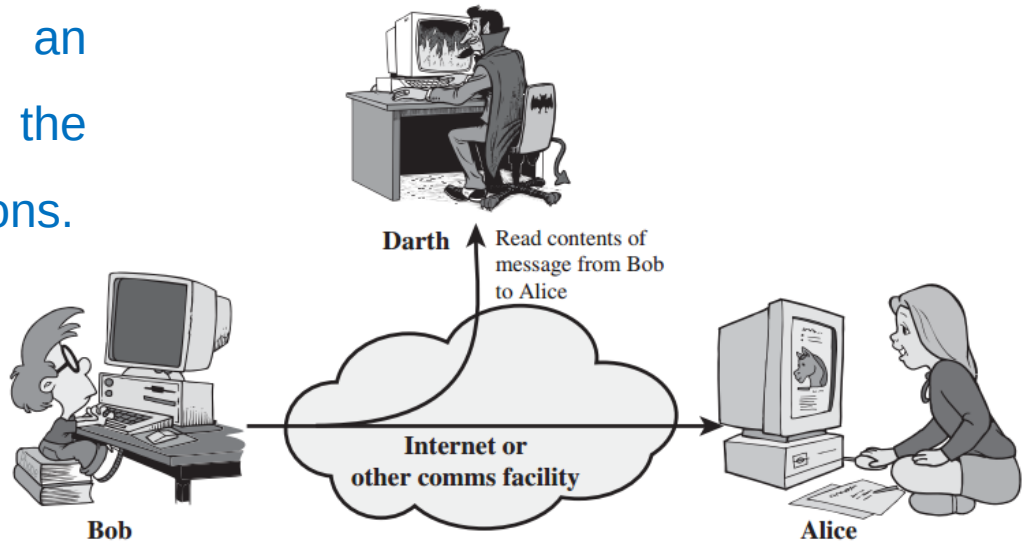
A Passive attack attempts to learn or make use of information from the system but does not affect system resources. Passive Attacks are in the nature of eavesdropping on or monitoring of transmission. The goal of the opponent is to obtain information is being transmitted. Types of Passive attacks are as following:

Passive & Active Attacks

A. Passive Attacks

- **The release of message contents:** is easily understood. A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information.

We would like to prevent an opponent from learning the contents of these transmissions.



(a) Release of message contents

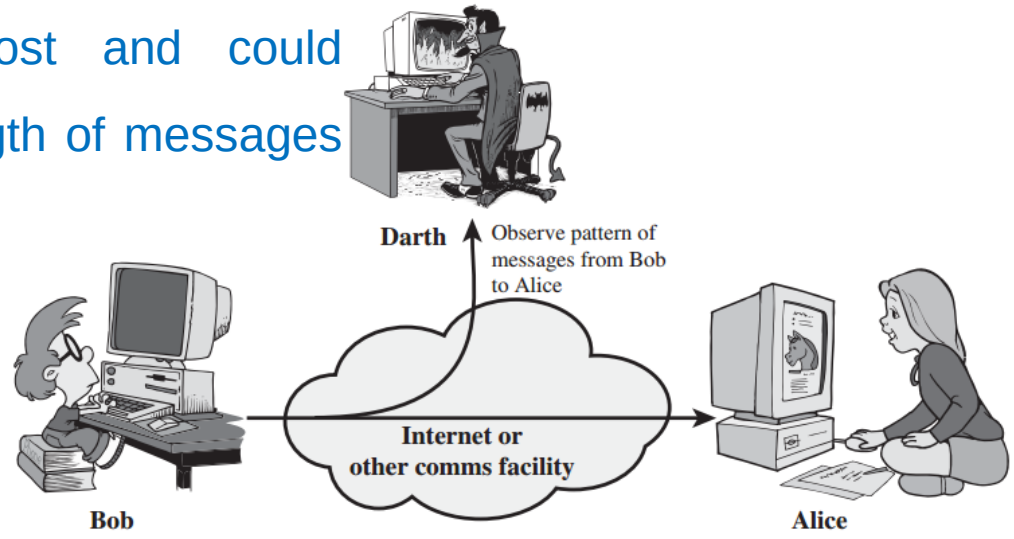
Passive & Active Attacks

A. Passive Attacks

- **Traffic analysis:** Suppose that we had a way of masking (encryption) of information, so that the attacker even if captured the message could not extract any information from the message.

The opponent could determine the location and identity of communicating host and could observe the frequency and length of messages being exchanged.

This information might be useful in guessing the nature of the communication that was taking place.



Passive & Active Attacks

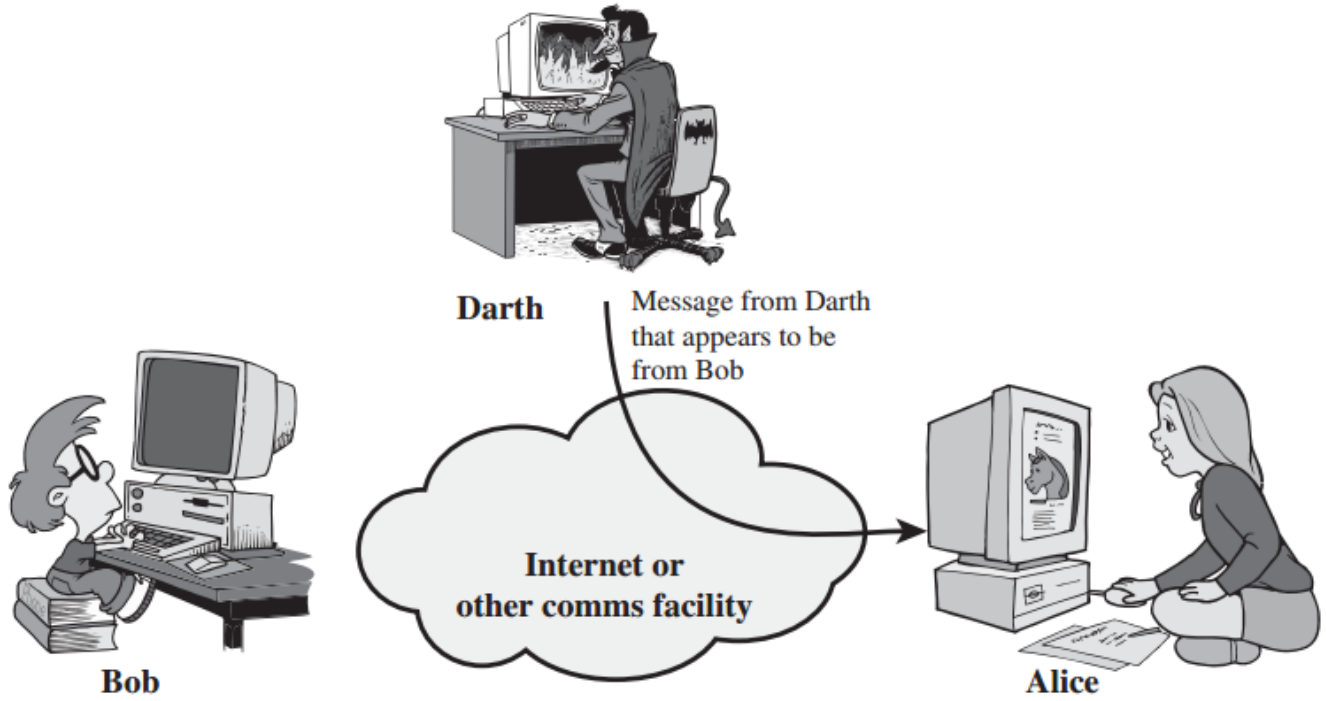
B. Active Attacks

An Active attack attempts to alter system resources or effect their operations. Active attack involves some modification of the data stream or creation of false statement.

Passive & Active Attacks

B. Active Attacks

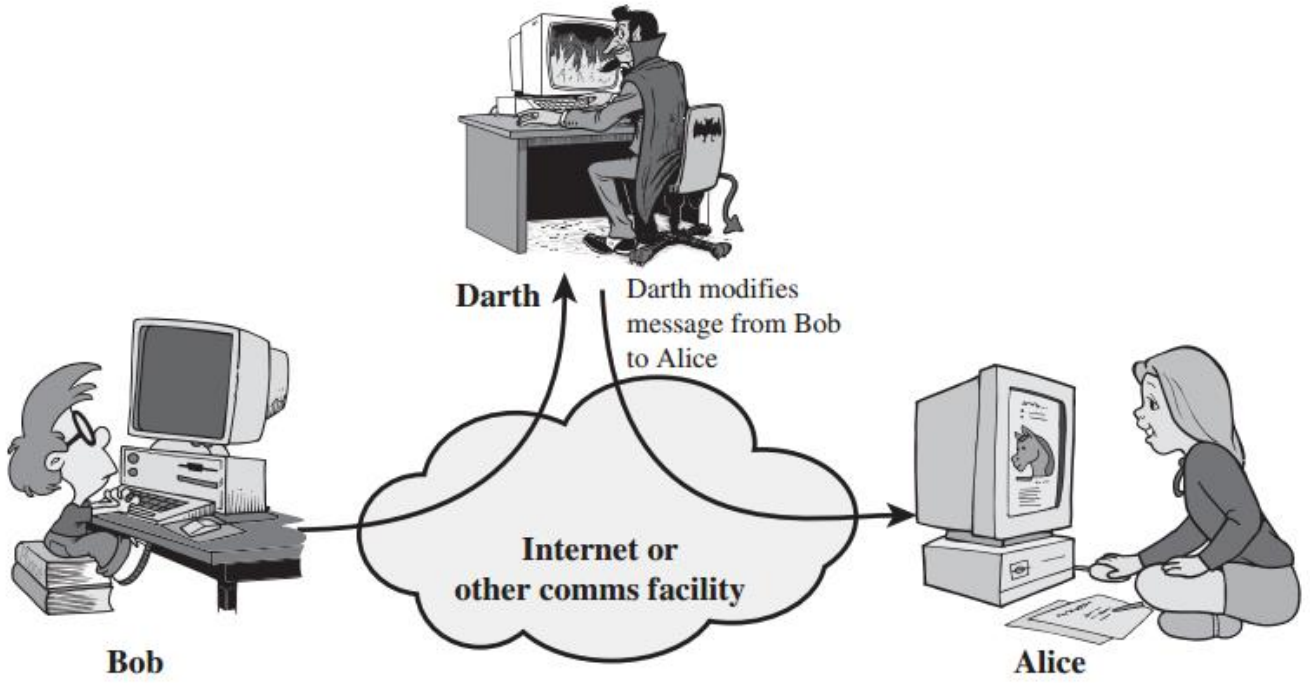
- Masquerade: takes place when one entity pretends to be different entity.



Passive & Active Attacks

B. Active Attacks

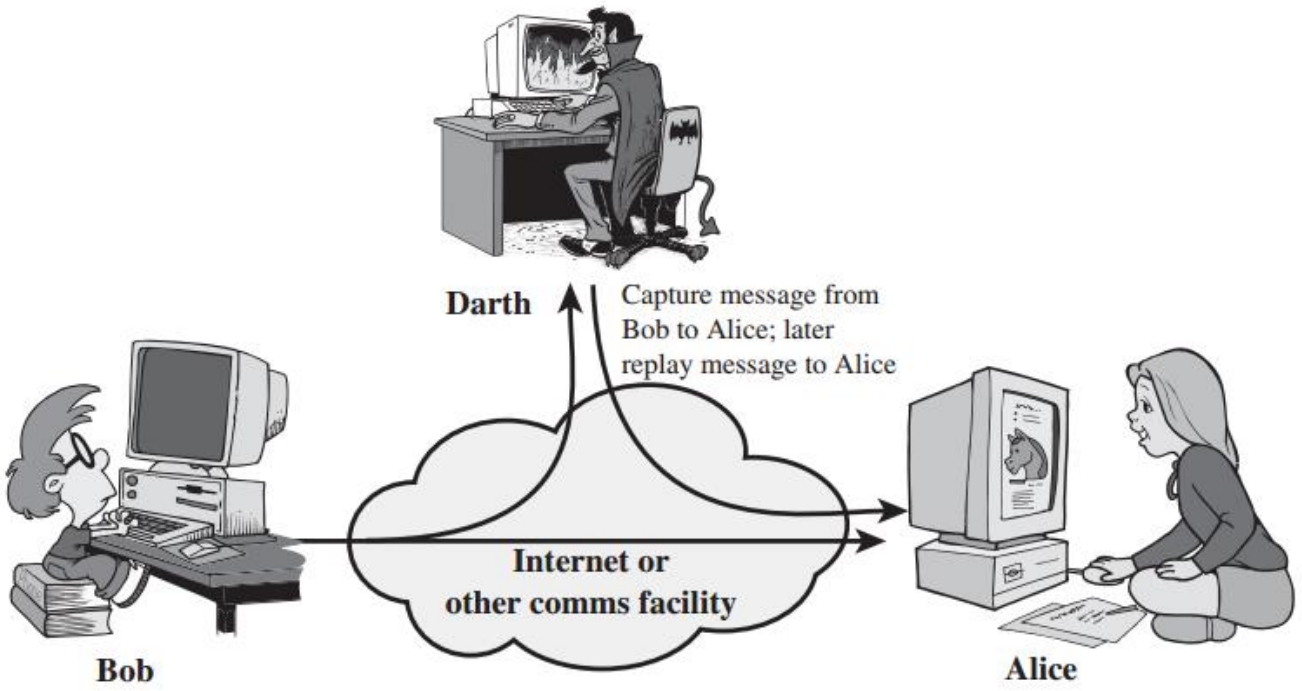
- Modification of messages: some portion of a message is altered or delayed or reordered to produce an unauthorized effect.



Passive & Active Attacks

B. Active Attacks

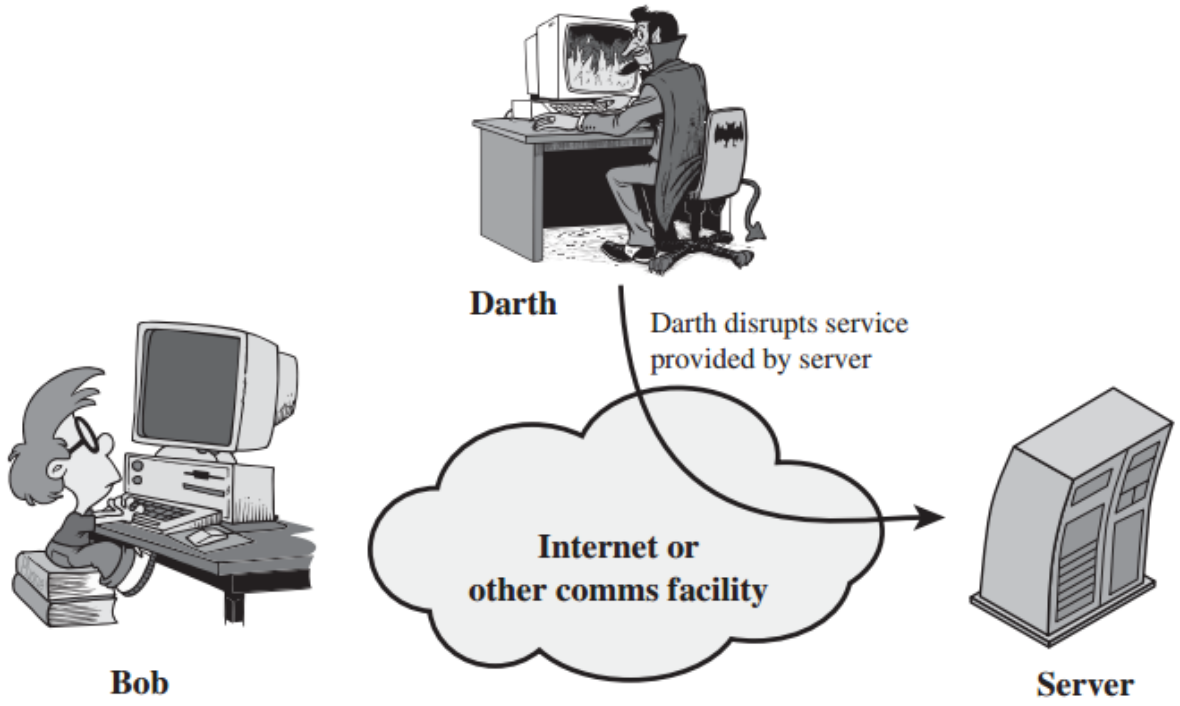
- Replay: involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.



Passive & Active Attacks

B. Active Attacks

- The denial of service: prevents or inhibits the normal use or management of communications facilities.



Passive Attacks vs Active Attacks

Active Attacks	Passive Attacks
Modification in information takes place.	Modification in the information does not take place.
Danger to Integrity as well as availability.	Danger to Confidentiality.
Attention is on prevention.	Attention is on detection.
system is always damaged, System resources can be changed.	No harm to the system, System resources are not changing
Can be easily detected	Very difficult to detect
The purpose of an active attack is to harm the ecosystem.	The purpose of a passive attack is to learn about the ecosystem.
The duration of an active attack is short.	The duration of a passive attack is long.
The prevention possibility of active attack is High	The prevention possibility of passive attack is low.
Complexity is High	Complexity is low.

Cryptography and Network Security I

2nd Lecture

Authentication, Access Control & Biometrics

By

Dr. Taqwa F. Hassan

Edited by

Dr. Ali Albu-Rghaif

Department of Computer Engineering

College of Engineering - University of Diyala

User Authentication

A user authentication policy is a process for verifying that the persons attempting to access services and applications is who they claim to be.

- This can be accomplished through a various of authentication methods, such as entering a password on your laptop or phone or a PIN number on the ATM.

An authentication process consists of two steps:

- Identification Step: Presenting an identifier to the security system. (Identifiers should be assigned carefully, because authenticated identities are the basis for other security services, such as access control service.)
- Verification Step: Presenting or generating authentication information that corroborates the binding between the entity and the identifier.

User Authentication

The four means of authenticating user identity are based on

- **Something You Know**: A password or PIN.
- **Something You Have**: A physical device like a smart card or a security token.
- **Something You Are**: Biometric data, such as fingerprints or facial recognition.
- **Something You Do**: Behavioral patterns, like typing rhythm or voice recognition.

Something You Know
Password Authentication

In a typical password authentication process, a user (or admin) provides a password that is checked against stored credentials to verify identity. The system typically follows these steps:

- User enters credentials (e.g., username and password).
- System hashes the entered password and compares it with the stored hashed password.
- Access is granted or denied based on whether the credentials match.

Something You Know
Password Vulnerabilities

1) Offline Dictionary Attack:

The attacker uses a list of common words or phrases to guess passwords.

2) Specific Account Attack:

If the attacker has a target in mind, they may focus on cracking that account's password.

3) Popular Password Attack:

The attacker uses a list of commonly used passwords to try and guess accounts.

4) Password Guessing against Single User:

The attacker tries to guess the password for a specific user.

Something You Know
Password Vulnerabilities

5) Workstation Hijacking:

If the attacker can gain physical access to a workstation, they may attempt to bypass security measures and access passwords.

6) Exploiting User Mistakes:

The attacker can trick users into revealing their passwords or making mistakes that lead to compromised accounts.

7) Exploiting Multiple Password Use:

If a user uses the same password for multiple accounts, compromising one account can lead to access to others.

8) Electronic Monitoring:

The attacker may use surveillance or other electronic means to gather information that can be used to compromise passwords.

Something You Know
Password Cracking

refers to the process of gaining unauthorized access to a system by guessing or exploiting vulnerabilities in the password authentication process.

- **Brute Force Attack: The attacker tries all possible password combinations until the correct one is found.**
- **Dictionary Attack: In this method, attackers use a predefined list of words or commonly used passwords.**
- **Rainbow Table Attack: Attackers use precomputed hash values and compare them to the hashed passwords in a system.**
- **Phishing: Rather than cracking the password algorithmically, attackers trick users into revealing their password through fake websites or deceptive communications.**
- **Credential Stuffing: Using passwords from previous data breaches, attackers attempt to use them across other platforms, betting on the possibility of password reuse.**

Something You Know
Password Selection Strategies

- **User education:** Users can be told the importance of using hard-to-guess passwords and can be provided with guidelines for selecting strong passwords.
- **Computer-generated passwords:** In this method, attackers use a predefined list of words or commonly used If the passwords are quite random in nature, users will not be able to remember them.
- **Reactive password checking:** The system periodically runs its own password cracker to find guessable passwords.
- **Proactive password checker**: User is allowed to select his or her own password. At the time of selection, the system checks to see if the password is allowable and, if not, rejects it.

Something You Have
Memory Cards

- Can store but do not process data
- The most common is the magnetic stripe card.
  - Can be reprogrammed by an inexpensive card reader
- Can include an internal electronic memory
- Can be used alone for physical access.
  - Hotel room
  - ATM
- Provides significantly greater security when combined with a password or PIN.
- Drawbacks of memory cards include:
  - Requires a special reader
    - Cost of the reader and keeping it secure
  - Loss of token
    - Prevents the owner from gaining system access
    - Adversary needs only to find the PIN
  - User dissatisfaction: for computer access may be inconvenient

Something You Have
Smart Tokens

- **Physical characteristics:**
  - Include an **embedded microprocessor**
  - A **smart token** that looks like a bank card
  - Can look like calculators, keys, small portable objects

- **Interface:**
  - Manual interfaces include a keypad and display for interaction
  - Electronic interfaces communicate with a compatible reader/writer
- **Authentication protocol:**
  - **Static**: The user authenticates to the token → token authenticates the user to the computer
  - **Dynamic password generator**: token generates a unique password (periodically, every 60 seconds) → Synchronization between token/server
  - **Challenge-response:** A random string of numbers → the smart token generates response based on the challenge. Ex. Public–key crypt.

Something You Have
Smart Cards

- **The most important category of smart cards**
  - Has the appearance of a credit card
  - Has an electronic interface
  - May use any of the smart token protocols
- **Contain:**
  - An entire **microprocessor** (Processor, Memory, I/O ports)
- **Typically include three types of memory:**
  - Read-only memory (**ROM**):
    - Stores data that does not change during the card's life
  - Electrically erasable programmable ROM (**EEPROM**)
    - Holds application data and programs
  - Random access memory (**RAM**)
    - Holds temporary data generated when applications are executed

Something You Are
Biometrics

The term biometrics is derived from the Greek words bio meaning life and metric meaning to measure.

Biometrics is the measurement and statistical analysis of people's unique physical and behavioral

- **e.g. fingerprints, face, palm veins, etc. to be used as an individualized code for recognition.**

Biometrics is a multidisciplinary field concerned with

- **Representing,**
- **Measuring**
- **Statistical analysis**

Something You Are
Why need Biometrics

The need and the complexity of recognition of humans has never been in this great in our history as it is now, so that, the need for fast and accurate new technology is increasing day by day, Which are used in:

- **Verification (control access systems),**
- **Identification (surveillance systems).**

The first use of biometrics was to link between identity documents and their holders through a face photos as in a passport.

Today the Authentication by biometric verification is becoming increasingly common and an indispensable tool to overcome the difficulties being faced in security systems.

Something You Are
Advantages and Benefits of Biometrics

- Hard to fake or steal, unlike passwords or Key.
- Fast Verify a large number of individuals in short time.
- Fast Identify the individual from a very huge number of people.
- Ease of use and convenience (Nothing to forget or lose it).
- Simple change along the user's life.(long life)
- Non-transferable from one to another.
- Templates take up less storage space.
- Continuous authentication in behavioral identifiers.

Something You Are
Disadvantages and Benefits of Biometrics

- It is costly to get up a biometric system and running it.
- If the system fails to capture all of the biometric data, it can lead to failure in identifying the person required.
- Databases holding biometric data can still be hacked.
- If a user gets injured or burn, then a biometric authentication system may fail to identify them.
- Errors such as false rejects can still happen.

Something You Are
Biometrics Vs Privacy

- Using biometrics can help to protect privacy by combating identity fraud, But it is also can reveal an element of personal privacy.
- The biometrics are not usually ‘secret’, and cannot be easily changed, destroyed or declared invalid in case it was stolen, this prevent using it again.

Something You Are
Types of Biometrics

- **Physiological identifiers**

  relate to the unique composition of the user, include:

  - **Face recognition.**
  - **Fingerprints.**
  - **Iris recognition.**
  - **DNA matching.**

- **Behavioral identifiers**

  It's the unique ways in which individuals act, including

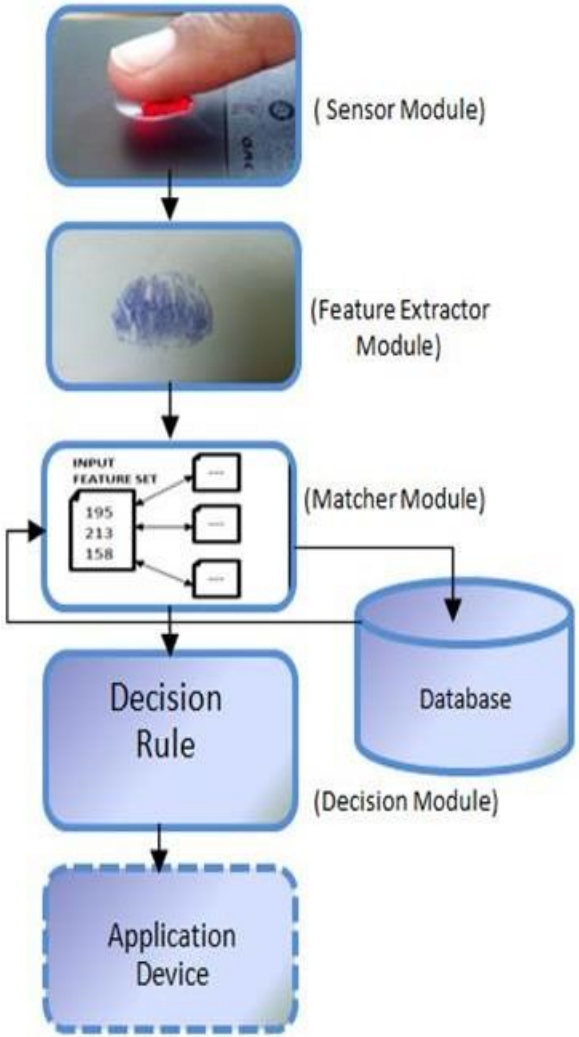
  - **Handwriting**
  - **keystroke**
  - **Voice recognition**
  - **Walking speed**
  - **Gestures.**

Something You Are
Components of Biometric System

The Biometric systems have four basic modules which are:

- **Sensor module,**
- **Feature extractor module,**
- **Matcher module,**
- **Decision module.**

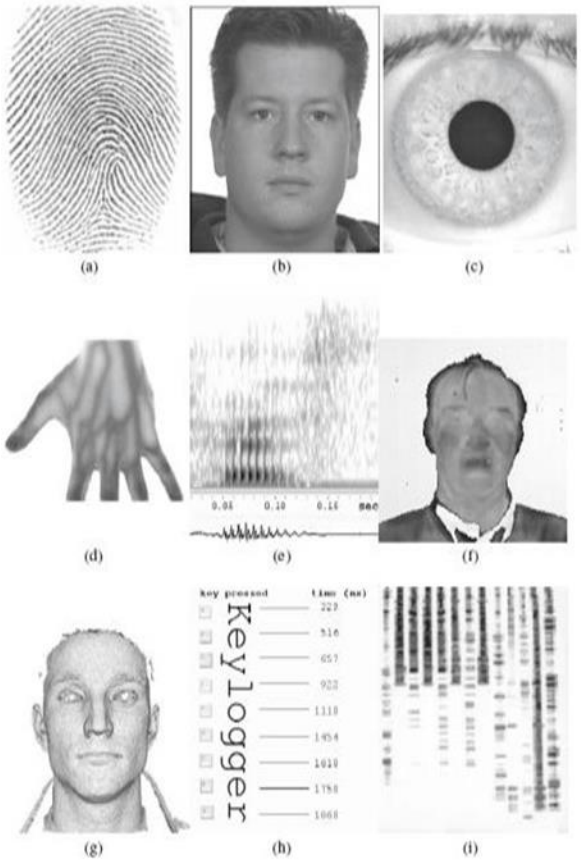
These four modules are necessary in any biometric system to acquire and process raw biometric data and convert it into some useful information. The block diagram of biometric system is shown in figure.

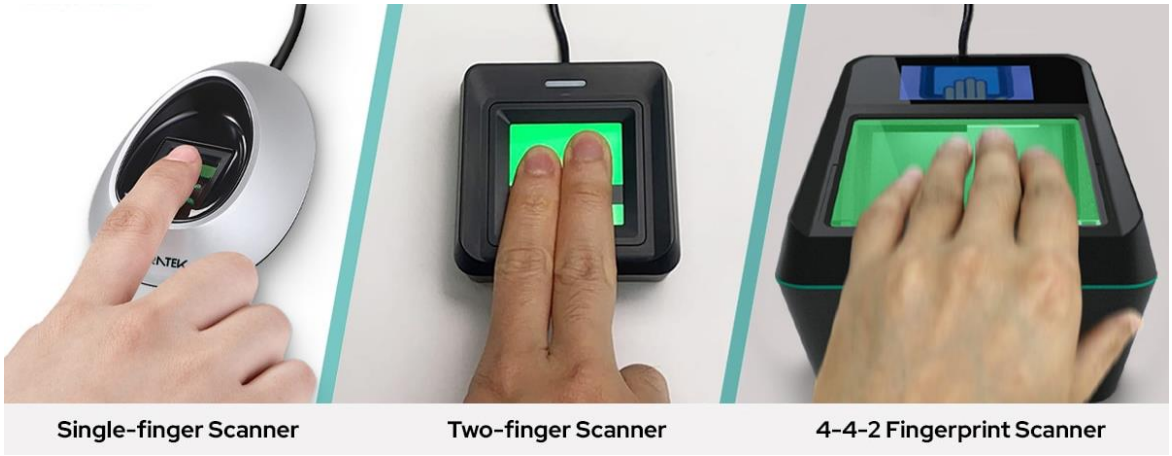


Something You Are
Components of Biometric System

1- Sensor Module

In this type of module raw biometric data is captured by the sensor and it scans the biometric trait to convert it into digital form. After converting it to digital form, this module transmits the data to feature extraction module.



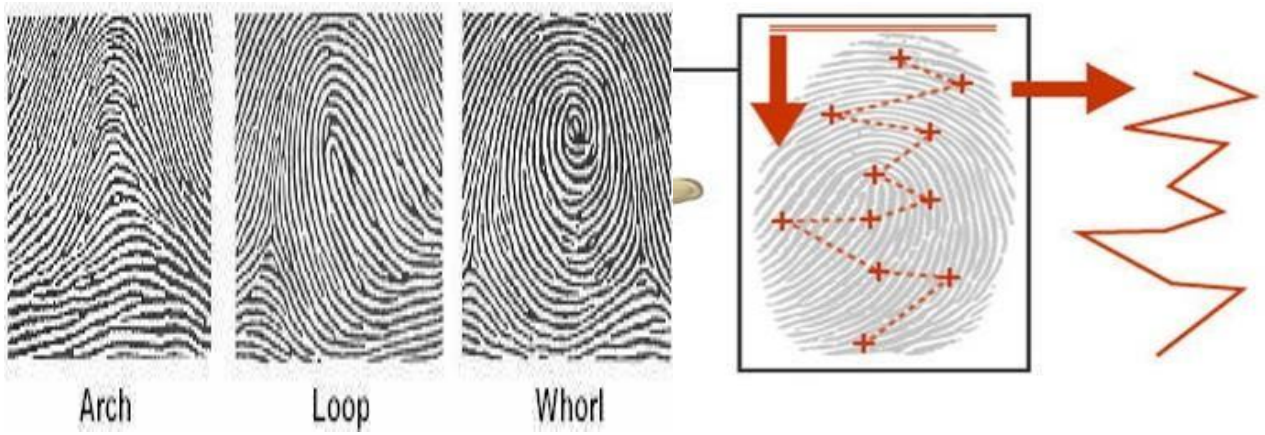


Single-finger Scanner | Two-finger Scanner | 4-4-2 Fingerprint Scanner

Something You Are
Components of Biometric System

2- Feature Extraction Module

It processes the raw data captured by sensor and generate a biometric template. It extracts the necessary features from the raw data which needs much attention because essential features must be extracted in an optimal way. It basically removes noise from the input sample and transmits the sample to the matcher module.

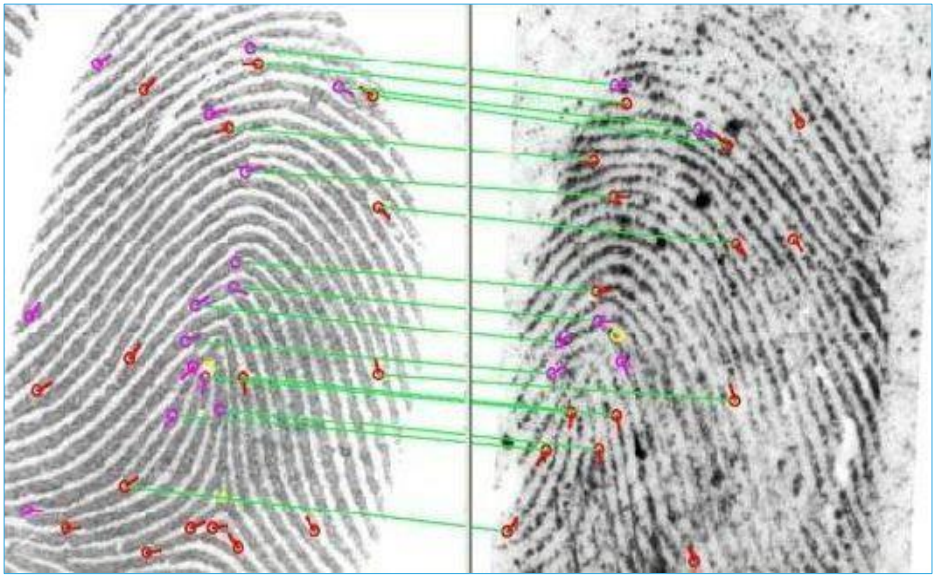


Something You Are
Components of Biometric System

3- Matcher Module

This module compares the input sample with the templates being stored in the database using matching algorithm and produces match score.

The Resulting Match Score is transmitted to the decision module.



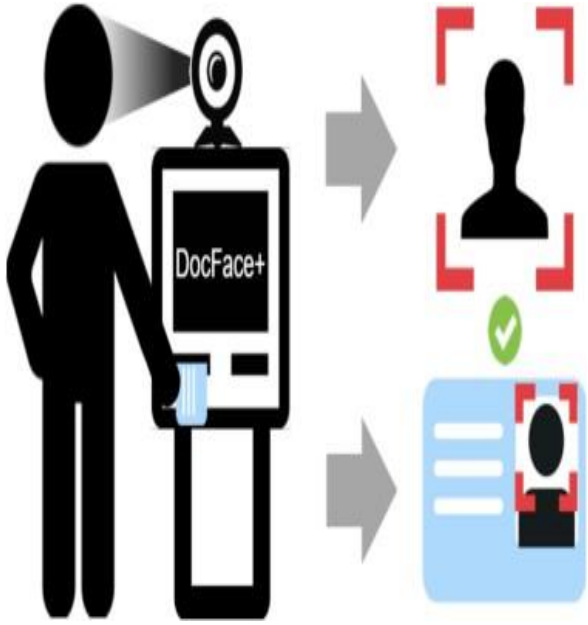
Something You Are
Components of Biometric System

4- Decision Module

After accepting the match score from matcher module, it compares the matching score against the predefined security threshold.

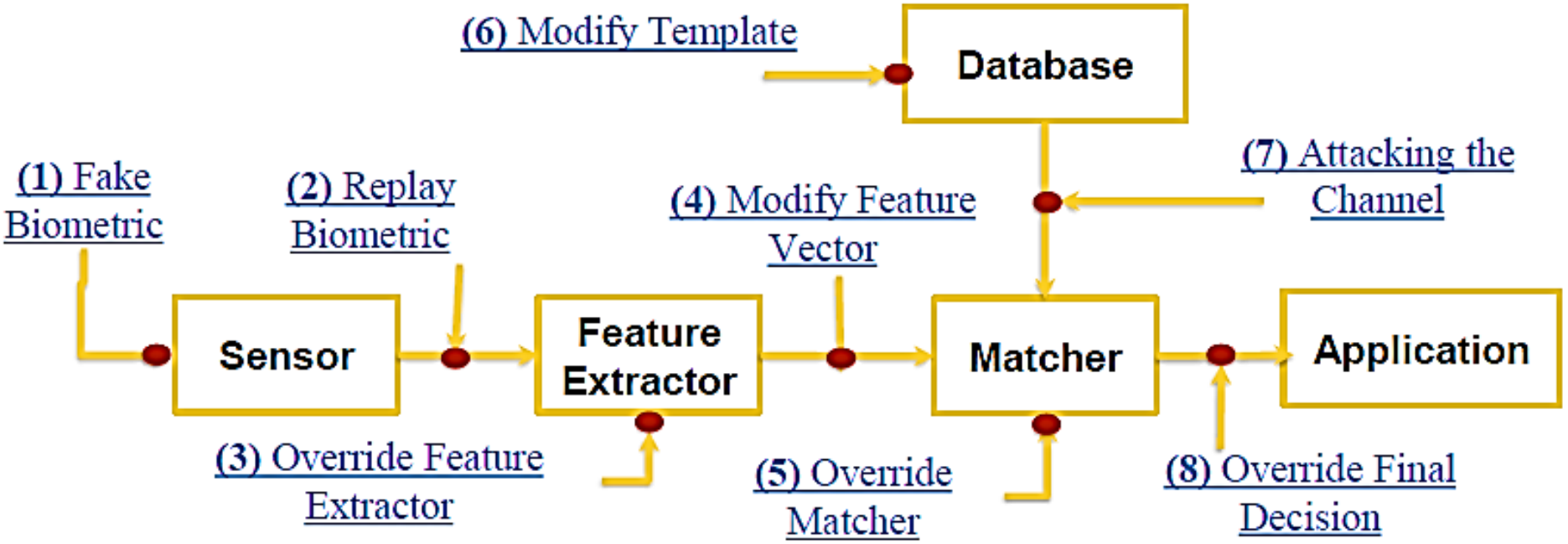
If match score is greater than predefined security threshold it will accept the individual

otherwise reject it.



Something You Are
Attack Points in a Biometric system

Biometric systems provide great advantages over traditional systems but they are vulnerable to attacks. There are eight attack points in biometric system which can be attacked as shown in the figure below.



Something You Are
Attack Points in a Biometric system

1) Fake Biometric

In this type of attack a fake biometric such as a fake finger or image of the face is presented at the sensor.

2) Replay Biometric

Biometric Signals In this mode of attack a recorded signal is replayed to the system bypassing to the sensor.

3) Override Feature Extractor

The feature extractor is forced to produce feature sets chosen by the attacker, instead of the actual values generated from the data obtained from the sensor.

4) Modify Feature Vector

The features extracted using the data obtained from the sensor is replaced with a different fraudulent feature set.

Something You Are
Attack Points in a Biometric system

5) Override Matcher

The matcher component is attacked to produce pre- selected match scores regardless of the input feature set.

6) Replay Modify

Modifying one or more templates in the database, which could result either in authorizing a fraud or denying service to the person, associated with the corrupted template.

7) Attacking the Channel

Data traveling from the stored template to the matcher is intercepted and modified in this form of attack.

8) Override Final Decision

Here the final match decision is overridden by the hacker disabling the entire authentication system.

Something You Do
Voice Recognition

Voice recognition is a form of biometrics, and voice authentication is the use of a user's speech to authenticate users. Like fingerprints and facial scans, voice and user speech can serve as a unique marker of a user's ID.

Typically, the user will repeat a phrase or collection of phrases to "train" the recognition software to recognize unique vocal qualities based on two qualities:

- **Physiological qualities:** These include things like tone and volume.
- **Behavioral qualities:** unique inflections in speech that come from accents, regional dialects.

Something You Do
Voice Recognition

Voice recognition is a form of biometrics, and voice authentication is the use of a user's speech to authenticate users. Like fingerprints and facial scans, voice and user speech can serve as a unique marker of a user's ID.

Typically, the user will repeat a phrase or collection of phrases to "train" the recognition software to recognize unique vocal qualities based on two qualities:

- **Physiological qualities:** These include things like tone and volume.
- **Behavioral qualities:** unique inflections in speech that come from accents, regional dialects.

Something You Do
Voice Recognition

voice authentication carries many of the same advantages of other biometrics, including:

- **Harder to fake than other forms of authentication**

  A password can be stolen, and a token can be copied or forged if security isn't kept tight. Biometric data is much harder to fake, all things being equal, and it is much harder to steal through practices like broad phishing attacks.

- **Supports streamlined user experience**

  Logging into a system shouldn't be difficult, regardless of whether or not it's one of your employees or one of your clients or customers. With biometrics, secure authentication methods can rely on the person just being there, rather than remembering a complex password.

Something You Do
Voice Recognition

- **Accessible and convenient on a variety of devices**

  Biometrics are becoming incredibly common on devices like laptops, tablets or smartphones. That makes it that much easier to integrate next-level security across a productive device ecosystem for distributed teams.

- **Contactless login**

  This is something unique to speech recognition (among a few other biometrics like facial scanning). With speech recognition, you don't have to touch anything–which, as we've learned from the pandemic, is a safe and responsible step to take.

Something You Do
Voice Authentication Features

- **Hackable**

  When your system stores vocal data, it does so the same way it does other data: in a server or database.

- **Non-replaceable**

  If biometrics are compromised, it cannot be replaced like a password because you can't simply change your voice.

- **Not 100% accurate**

  No authentication method is foolproof. This is just as true for speech recognition.

- **Not as applicable in all environments**

  Speech recognition requires a rather silent area. Audio artifacts outside of the user's voice can interfere with authentication, which can cause problems if you are using voice alone.

Cryptography and Network Security I

3rd Lecture

Classical Encryption Techniques

By

Dr. Taqwa F. Hassan

Edited by

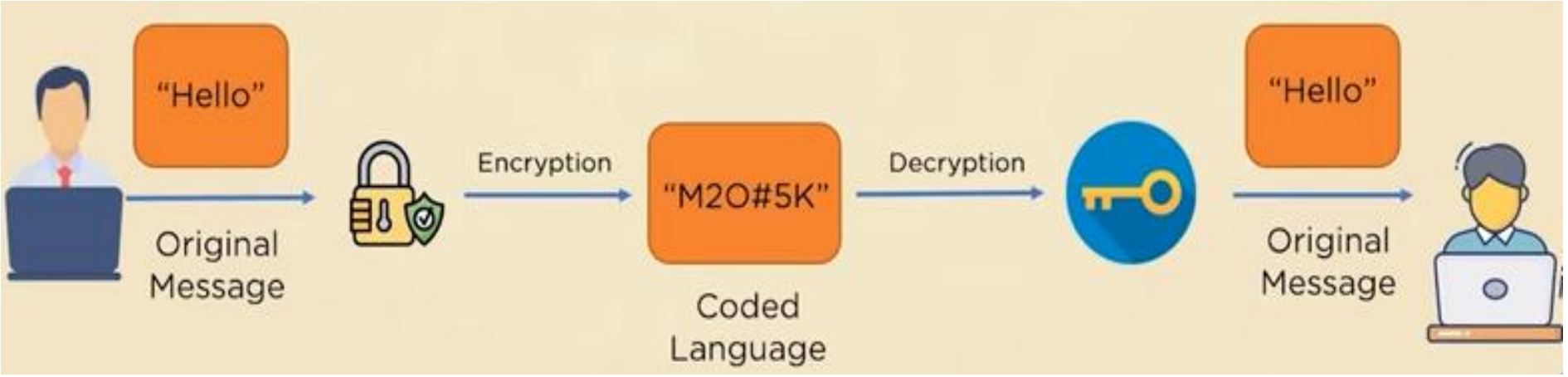
Dr. Ali Albu-Rghaif

Department of Computer Engineering

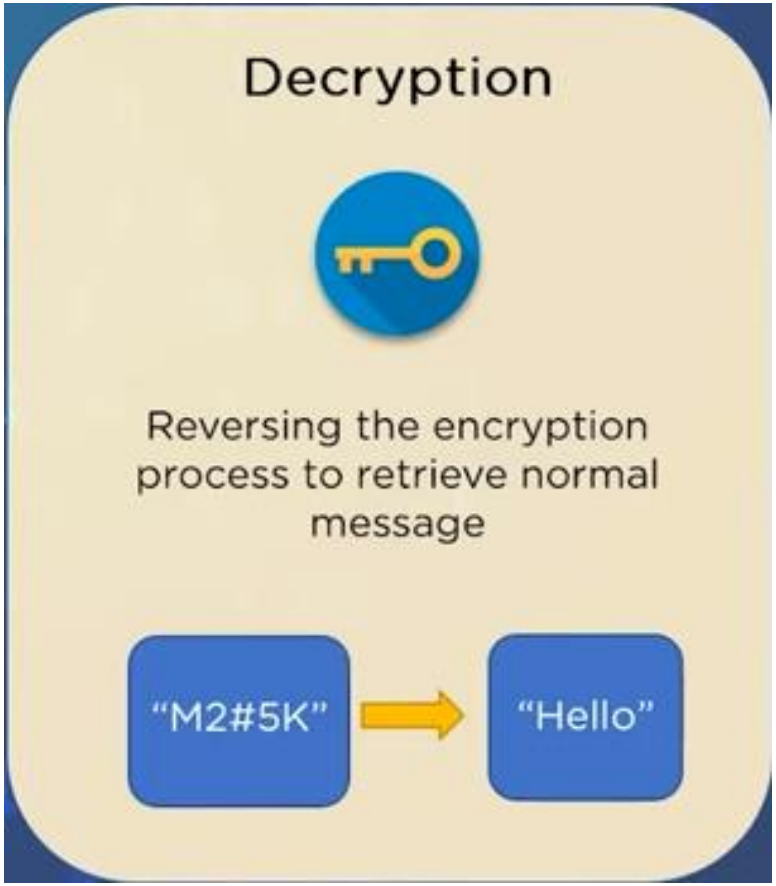
College of Engineering - University of Diyala

What is Cryptography

Cryptography is the practice and study of techniques used to secure communication and protect information from unauthorized access. It involves creating methods to transform readable data (plaintext) into an unreadable format (ciphertext) using encryption, so only authorized parties can decrypt and access the original data. Cryptography ensures the confidentiality, integrity, and authenticity of information and plays a crucial role in digital security.



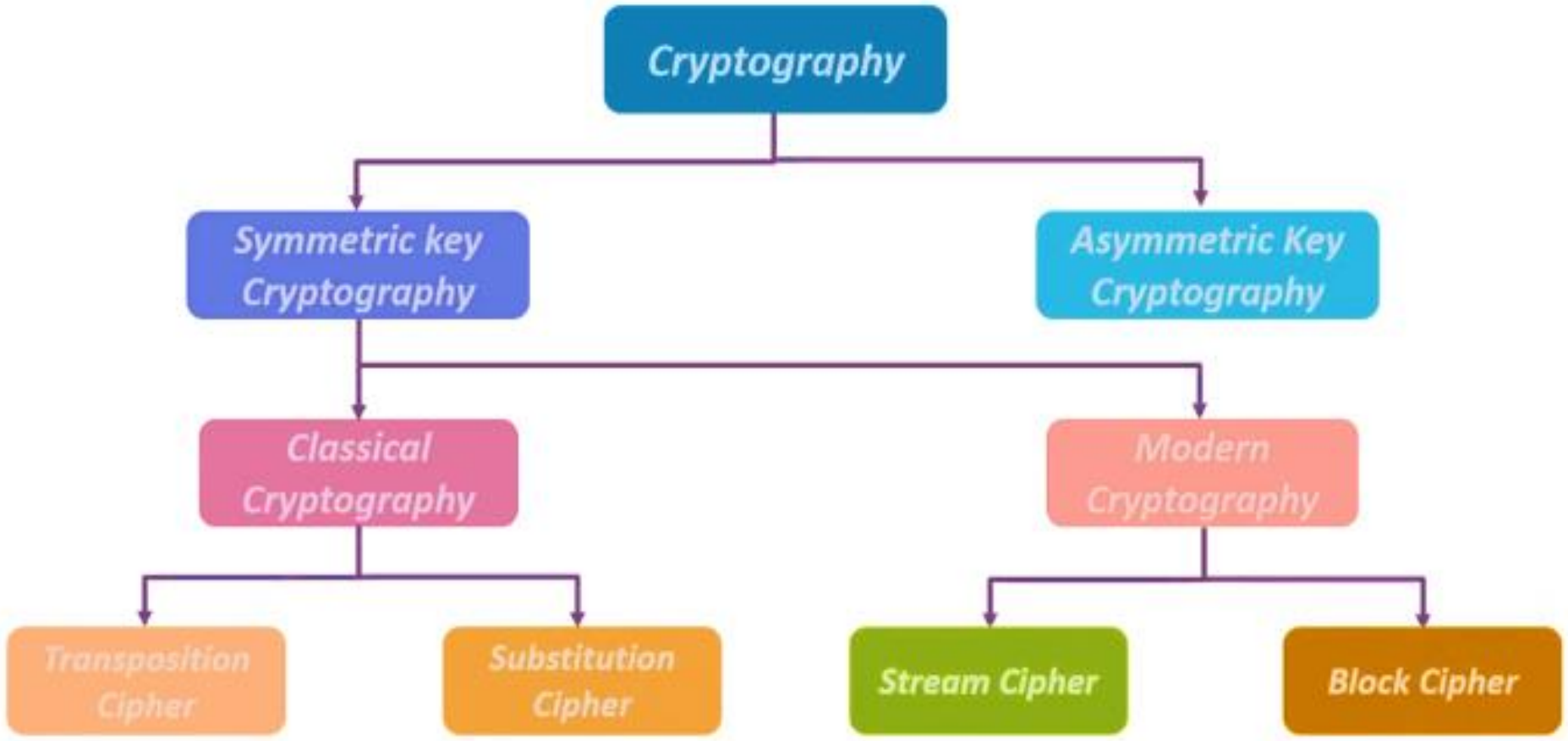
How Does Cryptography Work



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

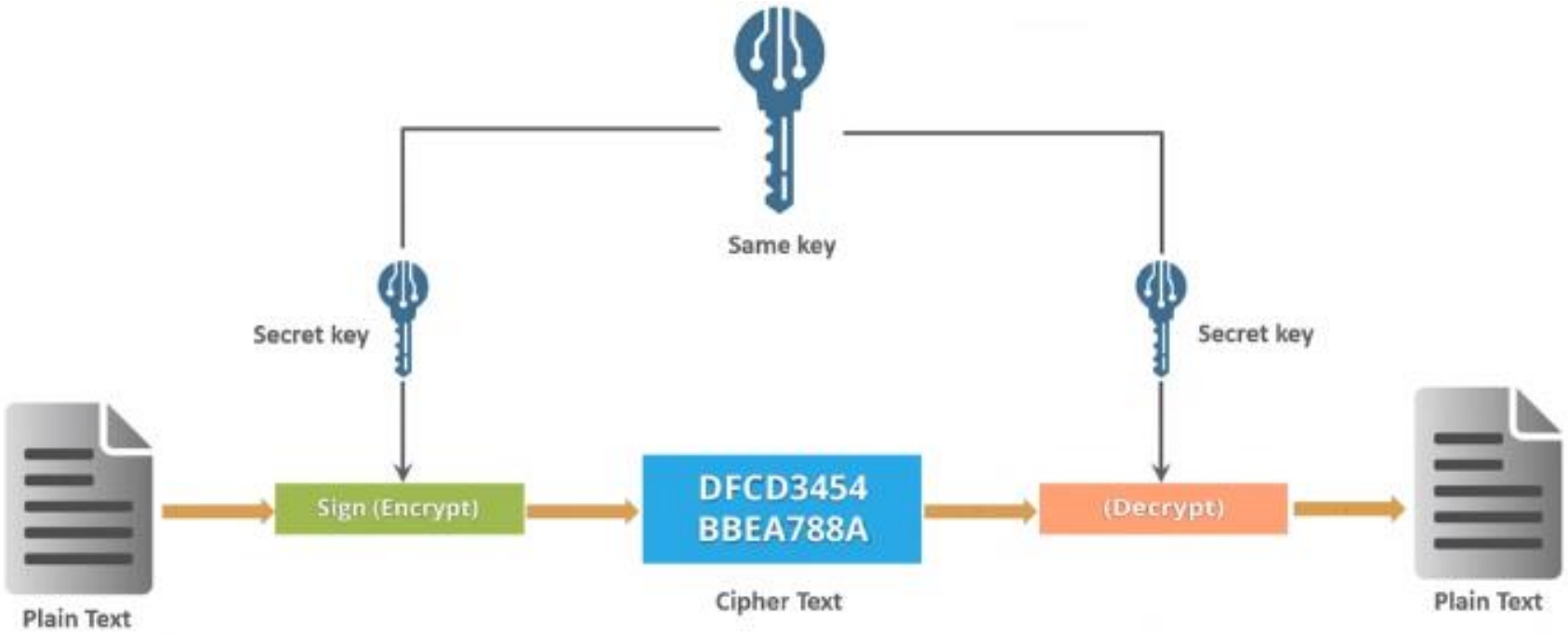
Classification of Cryptography



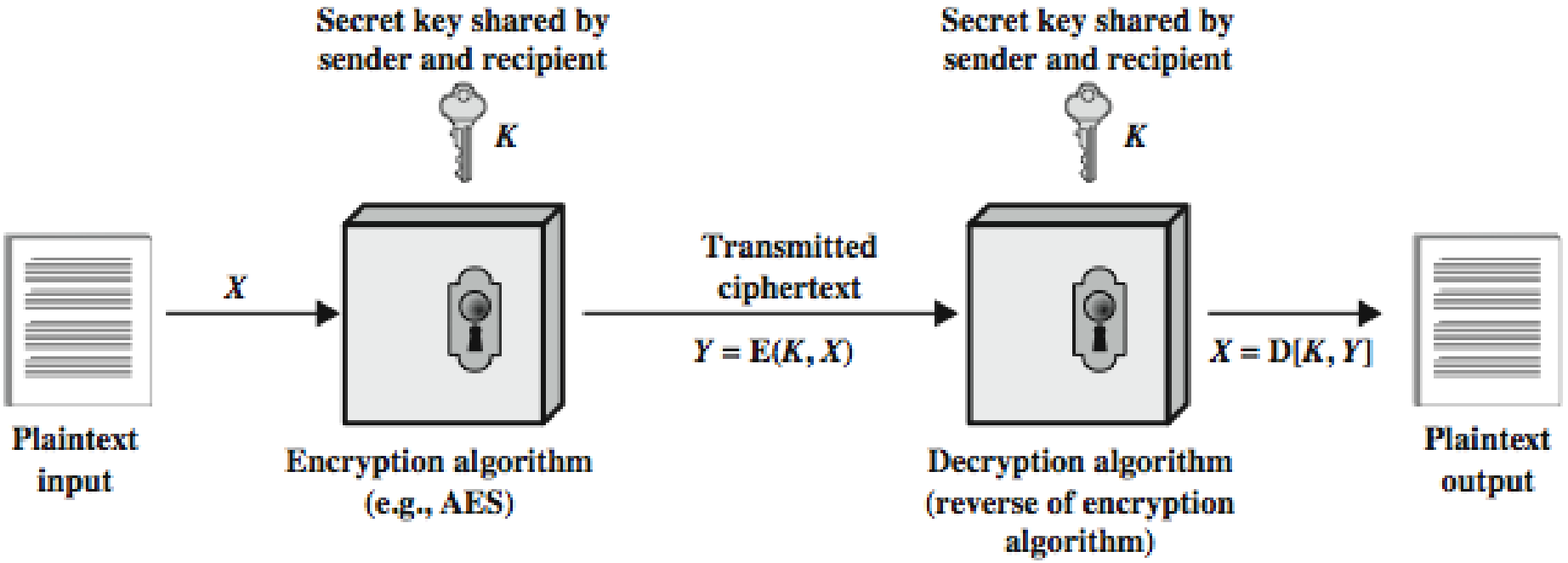
Symmetric Key Cryptography

An encryption system in which the sender and receiver of a message share a single, common key that is used to encrypt and decrypt the message.

The most popular symmetric key system is the Data Encryption Standard (DES) & Advanced Encryption Standard (AES).



Symmetric Cipher Model



Symmetric Encryption Features

- **Single Key Usage:** A single cryptographic key is used for encryption and decryption.
- **Efficiency:** Symmetric encryption algorithms are generally faster and less computationally intensive than asymmetric encryption algorithms.
- **Security Dependence:** The security of symmetric encryption depends entirely on keeping the key secret.
- **Common Algorithms:** Popular symmetric encryption algorithms include: AES, DES & 3DEs

Symmetric Encryption Requirements

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

$$Y = E(K, X) = E_K(X)$$
$$X = D(K, Y) = D_K(Y)$$

  - X: Plaintext
  - Y: Ciphertext
  - E: Encryption algorithm
  - D: Decryption algorithm

Cryptography

Cryptographic systems are characterized along 3 independent dimensions:

1. **The type of operations used for transforming plaintext to ciphertext**

- **All encryption algorithms are based on two general principles:**
  - **Substitution: in which each element in the plaintext is mapped into another element.**
  - **Transposition: in which elements in the plaintext are rearranged.**
- **The fundamental requirement is that no information be lost (i.e., that all operations are reversible).**
- **Most systems, referred to as product systems, involve multiple stages of substitutions and transpositions**

Cryptography

Cryptographic systems are characterized along 3 independent dimensions:

2. The number of keys used

- **If both sender and receiver use the same key, the system is referred to as symmetric, single-key, secret-key, or conventional encryption.**
- **If the sender and receiver use different keys, the system is referred to as asymmetric, two-key, or public-key encryption.**

Cryptography

Cryptographic systems are characterized along 3 independent dimensions:

3. The method used to process plaintext – block cipher, stream cipher

- **Block cipher – Input is divided into blocks. For a block of elements at a particular time instance, the output generated is also a block of elements.**
- **Stream cipher – Elements of input are processed continuously, one element at a time, and one element at a time is produced as output.**

Cryptanalysis & Brute-Force Attack

- **Cryptanalysis** is the science of studying and breaking cryptographic systems to understand the hidden messages without prior knowledge of the secret key.
- **Brute-force attack** is one of the simplest forms of cryptanalysis but is often highly resource-intensive. It involves systematically trying all possible keys or passwords until the correct one is found.

Cryptanalysis

Key Concepts in Cryptanalysis

- **Ciphertext-Only Attack (COA):** The attacker only has access to the ciphertext and attempts to deduce the plaintext or key.
- **Known-Plaintext Attack (KPA):** The attacker has both the plaintext and the corresponding ciphertext, which helps understand the encryption process.
- **Chosen-Plaintext Attack (CPA):** The attacker can choose to encrypt arbitrary plaintexts and can study the corresponding ciphertexts.
- **Chosen-Ciphertext Attack (CCA):** The attacker can decrypt chosen ciphertexts to gain insights into the encryption system.

Brute-Force Attack

Key Characteristics of Brute-Force Attack

- **Exhaustive Search:** It tests all possible combinations of keys or passwords until the correct one is identified.
- **Time Complexity:** The time required to break the encryption grows exponentially with the length of the key. For example, a 128-bit key has 21282 128 possible combinations, making it highly time-consuming.
- **No Cryptographic Weakness:** Unlike other cryptanalytic methods, brute-force attacks don’t rely on flaws in the algorithm but instead on the computational limits of the attacker.

Brute-Force Attack

Preventing Brute-Force Attacks

- **Strong Passwords:** Using longer and more complex passwords makes brute-force attacks more difficult.
- **Key Length:** Increasing the length of the cryptographic key can significantly enhance security. For example, moving from a 56-bit key (DES) to a 128-bit or 256-bit key (AES) makes brute-force attacks practically impossible with current technology.
- **Rate Limiting:** Systems can implement measures to limit the number of incorrect guesses an attacker can make in a certain time period.
- **Salting:** Adding random data (a salt) to passwords before hashing them makes brute-force attacks less effective against password databases.

Cryptanalysis vs Brute-Force Attack

- **Approach:**
  - Cryptanalysis involves analyzing weaknesses in cryptographic algorithms, while brute-force is a trial-and-error method that doesn't rely on any weakness in the encryption scheme.
- **Efficiency:**
  - Cryptanalysis can sometimes break encryption faster than brute-force by exploiting algorithmic flaws. Brute-force attacks are typically much slower, especially for modern encryption algorithms.
- **Computational Resources:**
  - Brute-force attacks are highly dependent on the attacker's computational power, while cryptanalysis might require more specialized knowledge of mathematics and algorithms.

Classical Encryption Techniques

classical encryption techniques are substitution and transposition.

- **Substitution: means replacing an element of the plaintext with an element of ciphertext.**
- **Transposition: means rearranging the order o appearance of the elements of the plaintext. Transposition is also referred to as permutation.**

Substitution Cipher

- Monoalphabetic Substitution Cipher
  - Caesar cipher
- Polyalphabetic Substitution Cipher
  - Vigenère Cipher
- Multiple Letter Substitution Cipher
  - Playfair Cipher
- Polygram Substitution Cipher
  - Hill Cipher

Caesar Cipher

The Caesar cipher is a simple substitution cipher, which replaces each plaintext letter by a different letter of the alphabet.
The Caesar cipher is based on transposition and involves shifting each letter of the plaintext message by a certain number of letters.

Algorithm

Steps to encrypt and decrypt using Caesar cipher method:

Write down the English alphabet and numbered it from 0 to 25

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Apply mathematical formulas to transform the message into the ciphertext in encryption or ciphertext into the message for decryption

Encryption: $C = (M + K) mod\ 26$

Decryption: $M = (C - K) mod\ 26$

Caesar Cipher

The Caesar cipher is a simple substitution cipher, which replaces each plaintext letter by a different letter of the alphabet.
The Caesar cipher is based on transposition and involves shifting each letter of the plaintext message by a certain number of letters.

Algorithm

Steps to encrypt and decrypt using Caesar cipher method:

Write down the English alphabet and numbered it from 0 to 25

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Apply mathematical formulas to transform the message into the ciphertext in encryption or ciphertext into the message for decryption

Encryption: C = (P + K)mod 26

Decryption: P = (C − K)mod 26

Caesar Cipher

Example:

Encrypt the following message “Happy Day” using Caesar cipher method and key=5.

Solution:

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Encryption: C = (P + K)mod 26

$C(H)$ = (7 + 5)mod 26 = 12 mod 26 = 12 = M

$C(A)$ = (0 + 5)mod 26 = 5 mod 26 = 5 = F

$C(P)$ = (15+ 5)mod 26 = 20 mod 26 = 20 = U

$C(P)$ = (15+ 5)mod 26 = 20 mod 26 = 20 = U

$C(Y)$ = (24 + 5)mod 26 = 29 mod 26 = 3 = D

$C(D)$ = (3 + 5)mod 26 = 8 mod 26 = 8 = I

$C(A)$ = (0 + 5)mod 26 = 5 mod 26 = 5 = F

$C(Y)$ = (24 + 5)mod 26 = 29 mod 26 = 3 = D

C=MFUUDIFD

Caesar Cipher

Example:

Decrypt the following Ciphertext “MFUUDIFD” using Caesar cipher method and key=5.

Solution:

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Decryption: P = $(C - K) mod\ 26$

$M(M)$ = $(12 - 5) mod\ 26$ = $7\ mod\ 26 = 7$ = H

$M(F)$ = $(5 - 5) mod\ 26$ = $0\ mod\ 26 = 0$ = A

$M(U)$ = $(20 - 5) mod\ 26$ = $15\ mod\ 26 = 15$ = P

$M(U)$ = $(20 - 5) mod\ 26$ = $15\ mod\ 26 = 15$ = P

$M(D)$ = $(3 - 5) mod\ 26$ = $-2\ mod\ 26 = 26 - 2 = 24$ = Y

$M(I)$ = $(8 - 5) mod\ 26$ = $3\ mod\ 26 = 3$ = D

$M(F)$ = $(5 - 5) mod\ 26$ = $0\ mod\ 26 = 0$ = A

$M(D)$ = $(3 - 5) mod\ 26$ = $-2\ mod\ 26 = 26 - 2 = 24$ = Y

Vigenère Cipher

Vigenere Cipher is a method of encrypting alphabetic text. It uses a simple form of polyalphabetic substitution.

Algorithm:

The Vigenère cipher is a kind of polyalphabetic substitution cipher. It is about replacing plaintext letters with other letters. The parties have to agree on the common shared keyword (which may be also a sentence), which is used during the encryption algorithm.

During encrypting and decrypting, one should use a table which contains all alphabet letters in the correct order in the first row and then, in subsequent rows, letters shifted to the left by one subsequent position.

Vigenère Cipher

Algorithm

- write the plaintext out
- write the keyword repeated above it
- use each key letter as a Caesar cipher key
- encrypt the corresponding plaintext letter

Example: using keyword deceptive

- key: d e c e p t i v e d e c e p t i v e d e c e p t i v e
- plaintext: w e a r e d i s c o v e r e d s a v e y o u r s e l f
- ciphertext: z i c v t w q n g r z g v t w a v z h c q y g l m g j

Vigenère Cipher

Question : Encrypt " we are discovered save yourself " using Vigenère Cipher with keyword : ' deceptive'

Solution:

$$C_i = (P_i + K_{i \bmod m}) \bmod 26$$

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

PT	w	e	a	r	e	d	i	s	c	o	v	e	r	e	d	s	a	v	e	y	o	u	r	s	e	l	f
Key	d	e	c	e	p	t	i	v	e	d	e	c	e	p	t	i	v	e	d	e	c	e	p	t	i	v	e

PT	22	4	0	17	4	3	8	18	2	14	21	4	17	4	3	18	0	21	4	24	14	20	17	18	4	11	5
Key	3	4	2	4	15	19	8	21	4	3	4	2	4	15	19	8	21	4	3	4	2	4	15	19	8	21	4
CT	25	8	2	21	19	22	16	13	6	17	25	6	21	19	22	0	21	25	7	2	16	24	6	11	12	6	9
	z	i	c	v	t	w	q	n	g	r	z	g	v	t	w	a	v	z	h	c	q	y	g	l	m	g	j

Vigenère Cipher

Question : Decrypt " zicvtwqngrzgvtwavzhcqyglmgj " using Vigenère Cipher with keyword : ' deceptive'

Solution:

$$P_i = (C_i - K_{i \bmod m}) \bmod 26$$

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

CT	z	i	c	v	t	w	q	n	g	r	z	g	v	t	w	a	v	z	h	c	q	y	g	l	m	g	j
Key	d	e	c	e	p	t	i	v	e	d	e	c	e	p	t	i	v	e	d	e	c	e	p	t	i	v	e

CT	25	8	2	21	19	22	16	13	6	17	25	6	21	19	22	0	21	25	7	2	16	24	6	11	12	6	9
Key	3	4	2	4	15	19	8	21	4	3	4	2	4	15	19	8	21	4	3	4	2	4	15	19	8	21	4
PT	22	4	0	17	4	3	8	18	2	14	21	4	17	4	3	18	0	21	4	24	14	20	17	18	4	11	5
	w	e	a	r	e	d	i	s	c	o	v	e	r	e	d	s	a	v	e	y	o	u	r	s	e	l	f

Playfair Cipher

The Playfair cipher is a kind of polygraphic substitution cipher. A plaintext i divided into groups of characters and then one of the predefined characters is assigned to each group. Playfair's algorithm operates on groups of size of two letters.

Algorithm:

Before encryption, a table based on a secret keyword must be prepared. The table has dimensions of 5 by 5 cells and contains 26 letters, letters (I/J) should be put in the same cell. The given keyword is going to use to fill the table, but it is necessary to remove the repeated letters from the keyword first. Then, all the remaining letters should be entered into the table, without changing their original order found in the keyword, the table must be filled row by row from left to right and from top to bottom. The rest cells of the table should be filled with the rest alphabet letters in the ordinary alphabetical order.

Playfair Cipher

Encrypting and Decrypting Algorithm:

- If a pair is a repeated letter, insert a filler like 'X' If 'X' is a double letter, then insert another infrequent letter, say Q.
- If both letters fall in the same row,
  - (**Encrypting**) replace each with a letter to the right (wrapping back to start from the end)
  - (**Decrypting**) replace each with a letter to the left (wrapping back to start from the end)
- If both letters fall in the same column,
  - (**Encrypting**) replace each with the letter below it (again wrapping to top from bottom)
  - (**Decrypting**) replace each with the letter above it (again wrapping to top from bottom)
- Otherwise, each letter is replaced by the letter in the same row and in the column of the other letter of the pair

Playfair Cipher

EX) Encrypt the following message "**THE SCHEME REALLY WORKS**" using the Playfair cipher method, keywords "**CHARLES**".

- Break the plaintext in a two character diagram:
  - Plaintext is divided into 2-letter diagram
  - Use X to separate double letter
  - Use X to pad the last single letter

TH ES CH EM ER EA LL YW OR KS

TH ES CH EM ER EA L**X** LY WO RK S

TH ES CH EM ER EA LX** LY WO RK S**X**

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

Playfair Cipher

TH ES CH EM ER EA LX LY WO RK SX

- **TH -> PR**
- **ES -> SB**
- **CH -> HA**
- **EM -> DG**
- **ER -> DC**
- **EA -> BC**
- **LX -> AZ**
- **LY -> RZ**
- **WO -> VP**
- **RK -> AM**
- **SX -> BW**

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

Thus the message:

THE SCHEME REALLY WORKS

Becomes

PR SB HA DG DC BC AX RZ VP AM BW

Playfair Cipher

EX) Decrypt the following message "**PRSBHADGDCBCAXRZVPAMBW**" using the Playfair cipher method, keywords "**CHARLES**".

- **PR ->TH**
- **SB ->ES**
- **HA ->CH**
- **DG ->EM**
- **DC ->ER**
- **BC ->EA**
- **AZ ->LX**
- **RZ ->LY**
- **VP ->WO**
- **AM ->RK**
- **BW ->SX**

PR SB HA DG DC BC AX RZ VP AM BW

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

The message:

TH ES CH EM ER EA LX LY WO RK SX

THE SCHEME REALLY WORKS

Playfair Cipher

EX) Encrypt the following message "**Shi Sherry loves Heath Ledger**" using the Playfair cipher method, keywords "**Sherry**".

SH IS HE RR YL OV ES HE AT HL ED GE R

SH IS HE RX RY LO VE SH EA TH LE DG ER

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

Plaintext: SH IS HE RX RY LO VE SH EA TH LE DG ER

Ciphertext: HE GH ER DR YS IQ WH HE SC OY KR AL RY

Playfair Cipher

EX) Decrypt the following "**HEGHERDRYSIQWHHESCOYKRALRY**" using the Playfair cipher method, keywords "**Sherry**".

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

Ciphertext: **HE GH ER DR YS IQ WH HE SC OY KR AL RY**

Plaintext: **SH IS HE RX RY LO VE SH EA TH LE DG ER**

The Hill Cipher

Hill cipher is a polygraphic substitution cipher based on linear algebra. Each letter is represented by a number modulo 26. Often the simple scheme A = 0, B = 1, …, Z = 25 is used, but this is not an essential feature of the cipher. To encrypt a message, each block of n letters (considered as an n-component vector) is multiplied by an invertible n × n matrix, against modulus 26. To decrypt the message, each block is multiplied by the inverse of the matrix used for encryption.

The matrix used for encryption is the cipher key, and it should be chosen randomly from the set of invertible n × n matrices (modulo 26).

The Hill Cipher

Algorithm

Encryption

C = (k * p) mod 26

Decryption

P = (k^{-1} * C) mod 26

- P is the plaintext
- C is the cipher text
- k is the key matrix or encryption matrix
- k -1 inverse key matrix or decryption matrix

The Hill Cipher

Encryption

We have to encrypt the message ‘ACT’ (n=3).The key is ‘GYBNQKURP’ which can be written as the nxn matrix:

The message ‘ACT’ is written as vector:

$$\begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix}$$

Key

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix}$$

The enciphered vector is given as:

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix} = \begin{bmatrix} 67 \\ 222 \\ 319 \end{bmatrix} \equiv \begin{bmatrix} 15 \\ 14 \\ 7 \end{bmatrix} \pmod{26}$$

which corresponds to ciphertext of ‘**POH**’

The Hill Cipher

Decryption

To decrypt the message, we turn the ciphertext back into a vector, and then simply multiply by the inverse matrix of the key matrix.

The inverse of the matrix used in the previous example

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix}^{-1} \equiv \begin{bmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{bmatrix} \pmod{26}$$

For the previous Ciphertext ‘POH’:

$$\begin{bmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{bmatrix} \begin{bmatrix} 15 \\ 14 \\ 7 \end{bmatrix} \equiv \begin{bmatrix} 260 \\ 574 \\ 539 \end{bmatrix} \equiv \begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix} \pmod{26}$$

which gives us back ‘ACT’

The Hill Cipher

Encrypt the message 'EXAM' → (2 x 2) matrix & The key is '**HILL**'

1) **Convert the letters of the key "HILL" into their corresponding numerical values (A = 0, B = 1, ..., Z = 25):**

H = 7 I = 8 L = 11 L = 11

So, the key matrix is:

Key Matrix $\begin{bmatrix} H & I \\ L & L \end{bmatrix}$ → K = $\begin{bmatrix} 7 & 8 \\ 11 & 11 \end{bmatrix}$

2) **Convert the Plaintext ("EXAM") into Numbers**

E = 4 X = 23 A = 0 M = 12

So, the plaintext vector will be:

Plaintext Vectors $\begin{bmatrix} E & A \\ X & M \end{bmatrix}$ → **P =** $\begin{pmatrix} 4 \\ 23 \end{pmatrix}, \begin{pmatrix} 0 \\ 12 \end{pmatrix}$

The Hill Cipher

3) Perform Matrix Multiplication

$$\text{Key Matrix} = \begin{pmatrix} H & I \\ L & L \end{pmatrix} = \begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix} \qquad \text{Plaintext} \begin{pmatrix} E \\ X \end{pmatrix} \begin{pmatrix} A \\ M \end{pmatrix} = \begin{pmatrix} 4 \\ 23 \end{pmatrix} \begin{pmatrix} 0 \\ 12 \end{pmatrix}$$

$$C = PK \bmod 26$$

$$\begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix} \begin{pmatrix} 4 \\ 23 \end{pmatrix}$$

$$7 \times 4 + 8 \times 23 = 212$$

$$11 \times 4 + 11 \times 23 = 297$$

$$\begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix} \begin{pmatrix} 4 \\ 23 \end{pmatrix} = \begin{pmatrix} 212 \\ 297 \end{pmatrix}$$

$$\begin{pmatrix} 212 \\ 297 \end{pmatrix} = \begin{pmatrix} 4 \\ 11 \end{pmatrix} \bmod 26 = \begin{pmatrix} E \\ L \end{pmatrix}$$

$$\begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix} \begin{pmatrix} 0 \\ 12 \end{pmatrix}$$

$$7 \times 0 + 8 \times 12 = 96$$

$$11 \times 0 + 11 \times 12 = 132$$

$$\begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix} \begin{pmatrix} 0 \\ 12 \end{pmatrix} = \begin{pmatrix} 96 \\ 132 \end{pmatrix}$$

$$\begin{pmatrix} 96 \\ 132 \end{pmatrix} = \begin{pmatrix} 18 \\ 2 \end{pmatrix} \bmod 26 = \begin{pmatrix} S \\ C \end{pmatrix}$$

Ciphertext = “ELSC”

The Hill Cipher

Decrypt the message '**ELSC**' & The key is '**HILL**'

1) Find the Inverse of the Key Matrix mod 26

The inverse of a 2x2 matrix is calculated using:

$$\text{Inverse Matrix} = \frac{1}{ad - bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \mod 26$$

Where: a = 7 b = 8 c = 11 d = 11

Step 1: Calculate the Determinant

The determinant of the matrix is:

Determinant = ad − bc

= (7) (11) − (8) (11)

= 77 − 88 = −11

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

$$\begin{pmatrix} \text{H} & \text{I} \\ \text{L} & \text{L} \end{pmatrix} = \begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix}$$

Now, we need the determinant mod 26:

−11 mod 26 = 15

The Hill Cipher

Decrypt the message '**ELSC**' & The key is '**HILL**'

Step 2: Find the Multiplicative Inverse of the Determinant mod 26

We need to find the inverse of **15 mod 26**, which is the number x:

$$15x \equiv 1 \bmod 26$$

Using the extended Euclidean algorithm, we find that the inverse of **15 mod 26 is 7**

(15 x 7) mod 26 = 105 mod 26 = 1

Step 3: Apply the Inverse to the Matrix

Now, multiply each element of the matrix

$$adj\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \Rightarrow adj\begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix} = \begin{pmatrix} 11 & -8 \\ -11 & 7 \end{pmatrix}$$

Applying mod 26: $\begin{pmatrix} 11 & 18 \\ 15 & 7 \end{pmatrix}$

The Hill Cipher

Decrypt the message '**ELSC**' & The key is '**HILL**'

2) Convert Ciphertext ("ELSC") to Numbers

Where: **E = 4, L = 11, S = 18, C = 2**

$$\begin{pmatrix}4\\11\end{pmatrix}, \begin{pmatrix}18\\2\end{pmatrix}$$

3) Perform Matrix Multiplication for Decryption

$$P = CK^{-1} \bmod 26$$

$$\begin{pmatrix}25 & 22\\1 & 23\end{pmatrix}\begin{pmatrix}4\\11\end{pmatrix}$$

$$25 \times 4 + 22 \times 11 = 342$$

$$1 \times 4 + 23 \times 11 = 257$$

$$\begin{pmatrix}25 & 22\\1 & 23\end{pmatrix}\begin{pmatrix}4\\11\end{pmatrix} = \begin{pmatrix}342\\257\end{pmatrix}$$

$$\begin{pmatrix}342\\257\end{pmatrix} = \begin{pmatrix}4\\23\end{pmatrix} \bmod 26 = \begin{pmatrix}E\\X\end{pmatrix}$$

$$\begin{pmatrix}25 & 22\\1 & 23\end{pmatrix}\begin{pmatrix}18\\2\end{pmatrix}$$

$$25 \times 18 + 22 \times 2 = 494$$

$$1 \times 18 + 23 \times 2 = 64$$

$$\begin{pmatrix}7 & 8\\11 & 11\end{pmatrix}\begin{pmatrix}0\\12\end{pmatrix} = \begin{pmatrix}494\\64\end{pmatrix}$$

$$\begin{pmatrix}494\\64\end{pmatrix} = \begin{pmatrix}0\\12\end{pmatrix} \bmod 26 = \begin{pmatrix}A\\M\end{pmatrix}$$

Transposition Ciphers
Real Fence Transposition

Plaintext: THIS IS A QUICK WAY TO MIX

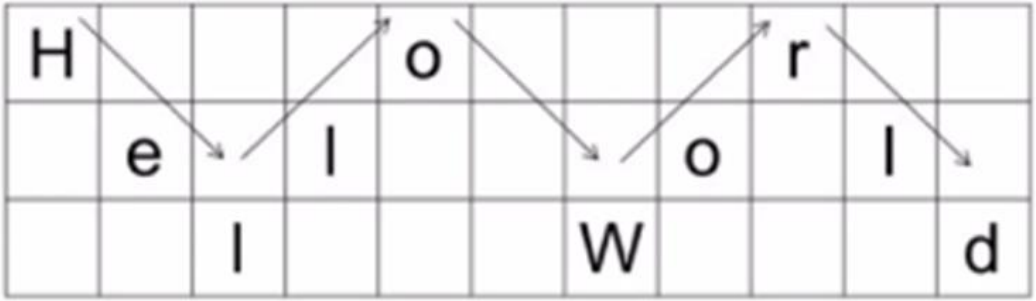
Depth: 3

T				I				U				W				O		
	H		S		S		Q		I		K		A		T		M		X
		I				A				C				Y				I	

Ciphertext: TIUWOHSSQIKATMXIACYI

Plaintext: Hello World

Depth: 3



Ciphertext: HORELOLLWD

Transposition Ciphers
Real Fence Transposition

Ciphertext: MEMATEAKETETHPR

Create a table

Number of column = number of letter in Ciphertext

Number of row = depth

Put X in Zig Zag fashion

X		X		X		X		X		X		X		X
	X		X		X		X		X		X		X	

Transposition Ciphers
Real Fence Transposition

Ciphertext: MEMATEAKETETHPR

Now, write down Ciphertext, row by row where X is written

M		E		M		A		T		E		A		K
	E		T		E		T		H		P		R	

Then, write down letter in Zig-Zag fashion

Plaintext: MEETMEATTHEPARK

Transposition Ciphers
Real Fence Transposition

Cypher text: I E O R L V M C U T Y O Y N

KEY : 3

Columns = Length of plain text = 14

Row = Key = 3

I				E				O				R	
	L		V		M		C		U		T		Y
		O				Y				N			

Plain Text: I LOVE MY COUNTRY

Transposition Ciphers
Columnar Transposition

Algorithm:

The secret key is usually a word (or just a sequence of letters). It has to be converted into a sequence of numbers. The numbers are defined by an alphabetical order of the letters in the keyword. The letter which is first in the alphabet will be the number **1**, the second letter in the alphabetical order will be **2**, and so on.

If there are multiple identical letters in the keyword, each next occurrence of the same letter should be converted into a number that is equal to the number for the previous occurrence increased by one.

Transposition Ciphers
Columnar Transposition

For example, the keyword: SWINDON

by rearranging the letters alphabetically the keyword becomes:

DINNOSW from that the letter **D** would be number **1** and letter (**I**) number **2** and letters **N** into the numbers **3** and **4** and so on.

would produce the following sequence of numbers: **6723154**

To encrypt a message, all the letters should be entered into the matrix, row by row, from left to right. The size of the matrix depends on the length of the message. The only known dimension is width, which is determined by the length of the secret keyword (which is the same as the length of the corresponding sequence of numbers), and known to both sides of the communication.

Transposition Ciphers
Columnar Transposition

Example:

Encrypt the following message "A Nice Midsummer Night's Dream" using Column transposition method and Key **"6723154".**

Solution:

Number of columns = number of the given key digits=7

6	7	2	3	1	5	4
A	N	I	C	E	M	I
D	S	U	M	M	E	R
N	I	G	H	T	S	D
R	E	A	M	X	X	X

Insert the message row by row into the key matrix and the empty remaining cells filled with **X**

1	2	3	4	5	6	7
E	I	C	I	M	A	N
M	U	M	R	E	D	S
T	G	H	D	S	N	I
X	A	M	X	X	R	E

Rearrange the matrix by columns into the correct order

Read the cipher text column by column

C= EMTX IUGA CMHM IRDX MESX ADNR NSIE

Transposition Ciphers
Columnar Transposition

Example:

Decrypt the following cipher text "EMTXIUGACMHMIRDXMESXADNRNSIE" using Column transposition method and Key **"6723154".**

Solution: Number of columns = number of the given key digits = 7

$$\text{Number of rows} = \frac{\textit{Number of Ciphertext letters}}{\text{number of the given key digits}} = \frac{28}{7} = 4$$

1	2	3	4	5	6	7
E	I	C	I	M	A	N
M	U	M	R	E	D	S
T	G	H	D	S	N	I
X	A	M	X	X	R	E

Insert the cipher text column by column into the correct order matrix

Notice no empty spaces should be found

6	7	2	3	1	5	4
A	N	I	C	E	M	I
D	S	U	M	M	E	R
N	I	G	H	T	S	D
R	E	A	M	X	X	X

Rearrange the matrix by columnsinto the Key order

Read the Message row by row:

ANICEMIDSUMMERNIGHTSDREAMXXX → A NICE MIDSUMMER NIGHTS DREAM XXX

Steganography

Steganography is the practice of hiding information within another medium so that its existence is concealed. Unlike cryptography, which protects the contents of a message by transforming it into an unreadable format, steganography hides the very fact that a message is being sent. The goal is to make the message undetectable to a casual observer, embedding it in a way that doesn't attract.

Types of Steganography

- Image Steganography:
  - Hiding a message within an image by modifying the pixel values in a way that is imperceptible to the human eye.
- Audio Steganography:
  - Hiding a message in an audio file by modifying its sound waves. Similar to image steganography.
- Video Steganography:
  - Hiding information within video files. Since videos are composed of many frames (which are images), steganography techniques can be applied to each frame, making it difficult to detect.
- Text Steganography:
  - Hiding messages within text files by manipulating the structure, such as inserting extra spaces, using different font types, or changing the formatting in subtle ways that a reader wouldn't notice.
- Network Steganography:
  - Hiding information within network protocols or data traffic. For example, modifying packet headers or using covert channels to embed hidden data within regular network communications.

Applications of Steganography

- Covert Communication:
  - Steganography can be used to send messages secretly without raising suspicion, often in espionage or secure communications.
- Watermarking:
  - Used to embed copyright information or digital signatures within multimedia files (images, videos, audio) to prove ownership and protect intellectual property.
- Digital Rights Management (DRM):
  - Protecting media content by embedding hidden data that identifies the rightful owner or usage restrictions.
- Authentication:
  - Steganography can help authenticate documents or files by embedding hidden codes that validate their integrity.

Advantages & Disadvantages of Steganography

Advantages

- Undetectable: Since the hidden message is embedded within a seemingly harmless file, a casual observer or attacker may not realize that a message exists.
- Combination with Cryptography: Steganography can be combined with cryptography for enhanced security, where the message is encrypted before being embedded.

Disadvantages

- Limited Payload: The amount of data that can be hidden is limited by the size and quality of the cover object. For example, a high-resolution image can store more hidden data than a low-resolution image.
- Vulnerability to Steganalysis: Advanced techniques (steganalysis) can detect the presence of hidden messages by analyzing statistical anomalies in the stego object.

Steganography vs. Cryptography

Cryptography

Makes the contents of a message unreadable to unauthorized parties but doesn't hide the fact that a message exists.

Steganography

Hides the fact that a message exists, but if discovered, the hidden message could be read unless it's combined with cryptography.

Cryptography and Network Security I

4th Lecture

Modern Encryption Techniques

By

Dr. Taqwa F. Hassan

Edited by

Dr. Ali Albu-Rghaif

Department of Computer Engineering

College of Engineering - University of Diyala

Block Cipher

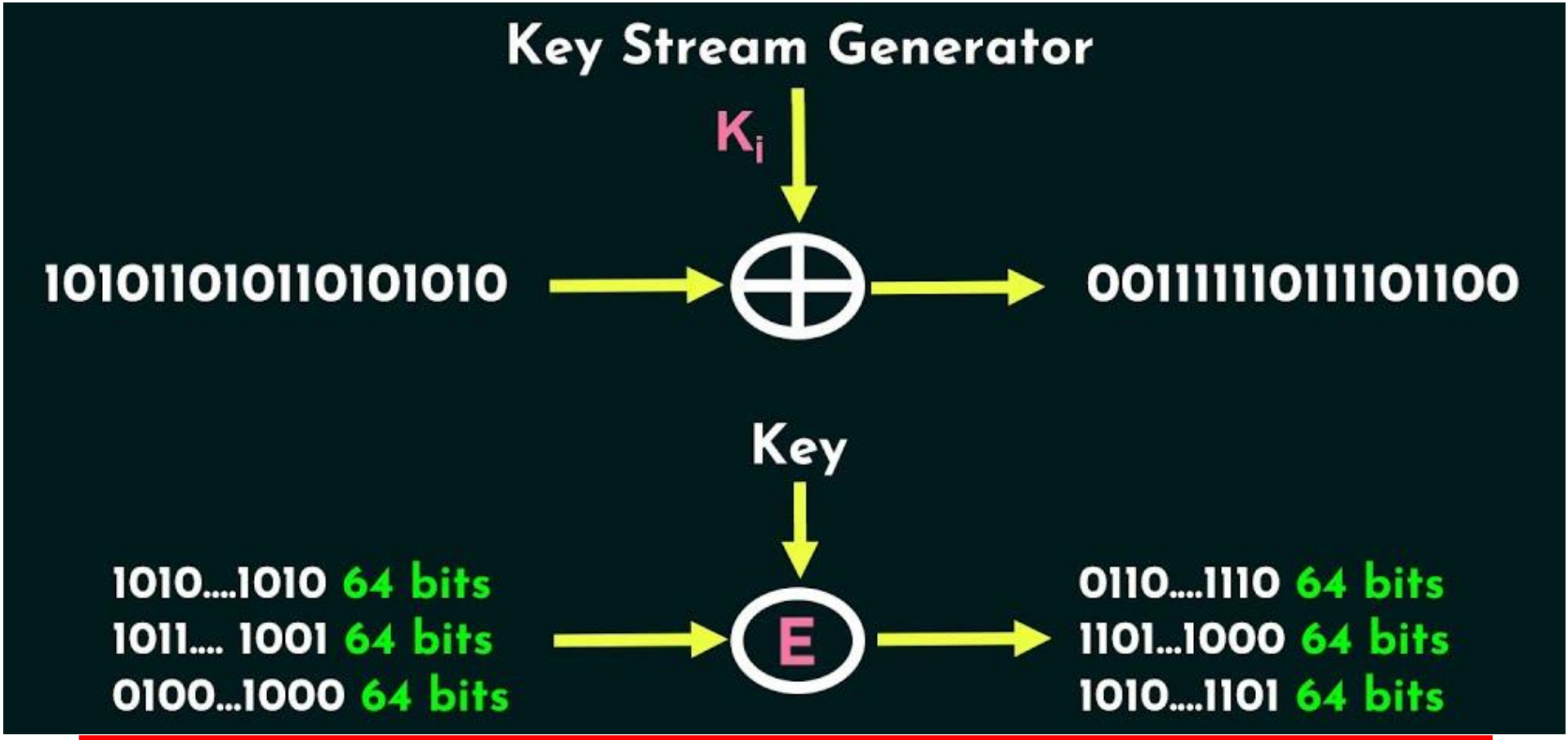
Confusion and Diffusion

there are two primitive operations with which strong encryption algorithms can be built:

1) **Confusion** is an encryption operation where the relationship between key and ciphertext is obscured. Today, a common element for achieving confusion is substitution, which is found in both DES and AES.
2) **Diffusion** is an encryption operation where the influence of one plaintext symbol is spread over many ciphertext symbols with the goal of hiding the statistical properties of the plaintext. A simple diffusion element is the bit permutation, which is used frequently within DES. AES uses the more advanced Mix-column operation.

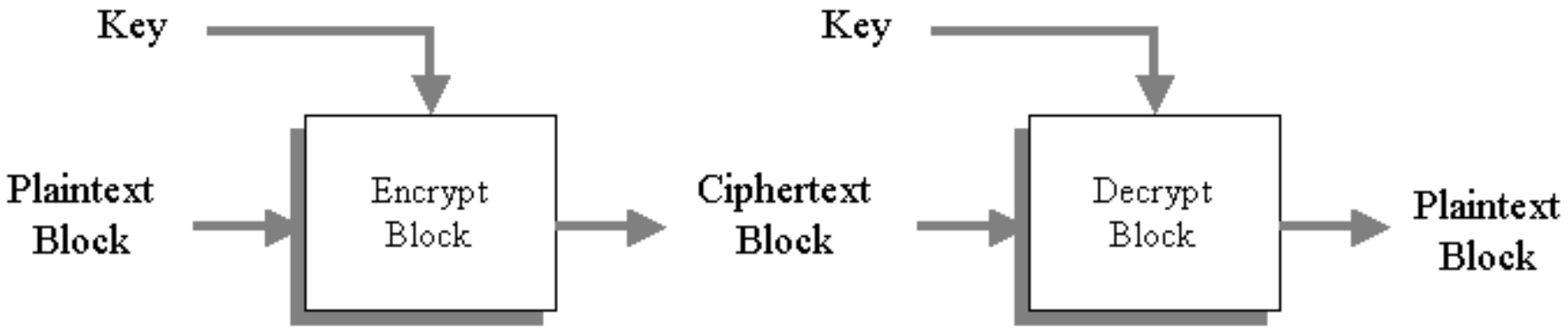
Block Cipher vs Stream Cipher

Stream ciphers convert one symbol of plaintext directly into a symbol of ciphertext. Block ciphers encrypt a group of plaintext symbols as one block.



Block Cipher

- A **block cipher** is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.
- Typically, a block size of **64 or 128 bits** is used. As with a stream cipher, the two users share a symmetric encryption key.



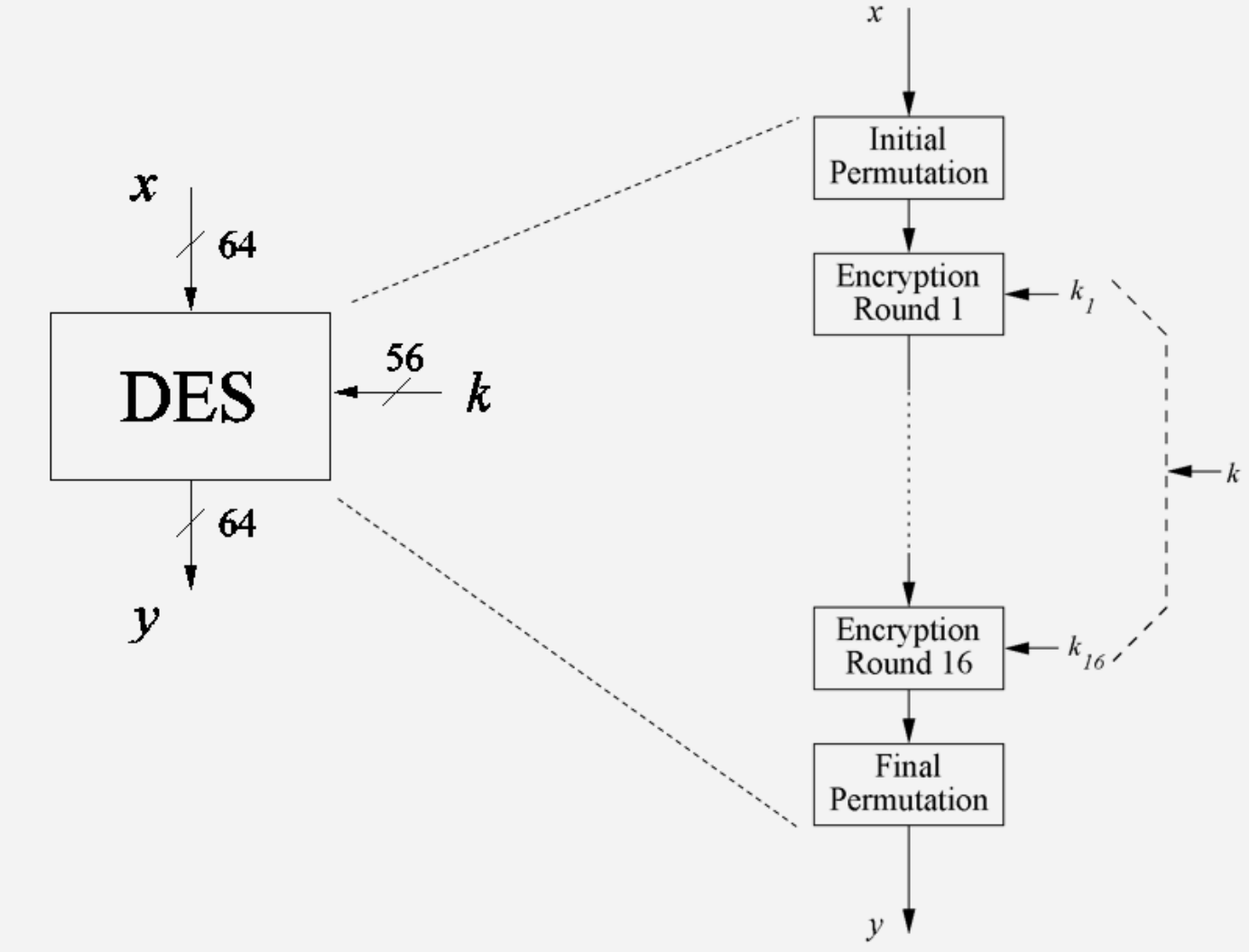
Block Ciphers Features

- **Block size:** in general larger block sizes mean greater security.
- **Key size:** a larger key size means greater security (larger key space).
- **Number of rounds:** multiple rounds offer increasing security.
- **Encryption modes:** define how messages larger than the block size are encrypted, very important for the security of the encrypted message.

Data Encryption Standard (DES) Algorithm

- DES is a Symmetric cipher: uses the same key for encryption and decryption
- Block size = 64 bits
- Key size = 56 bits (in reality, 64 bits, but 8 are used as parity-check bits for error control)
- Different subkey (48 bit) in each round derived from the main key
- Uses 16 rounds which all perform the identical operation

DES Algorithm

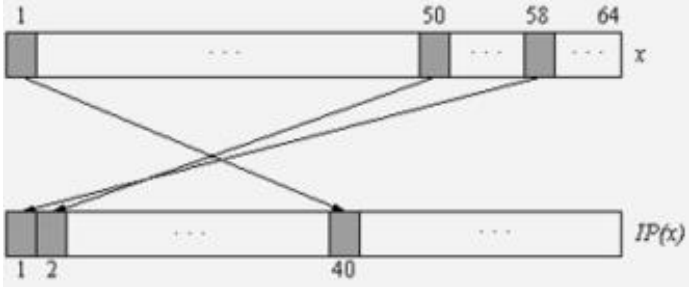


Internal Structure of DES

1. Initial Permutation (IP)

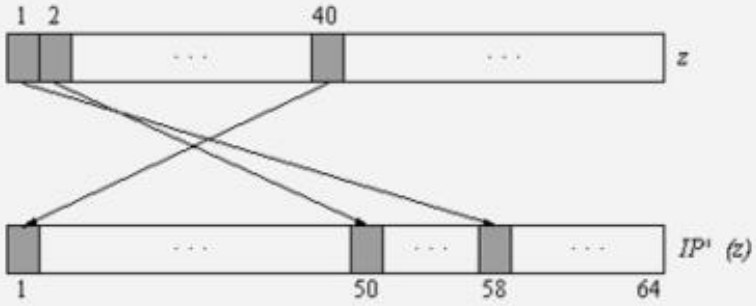
Initial Permutation

IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7



Final Permutation

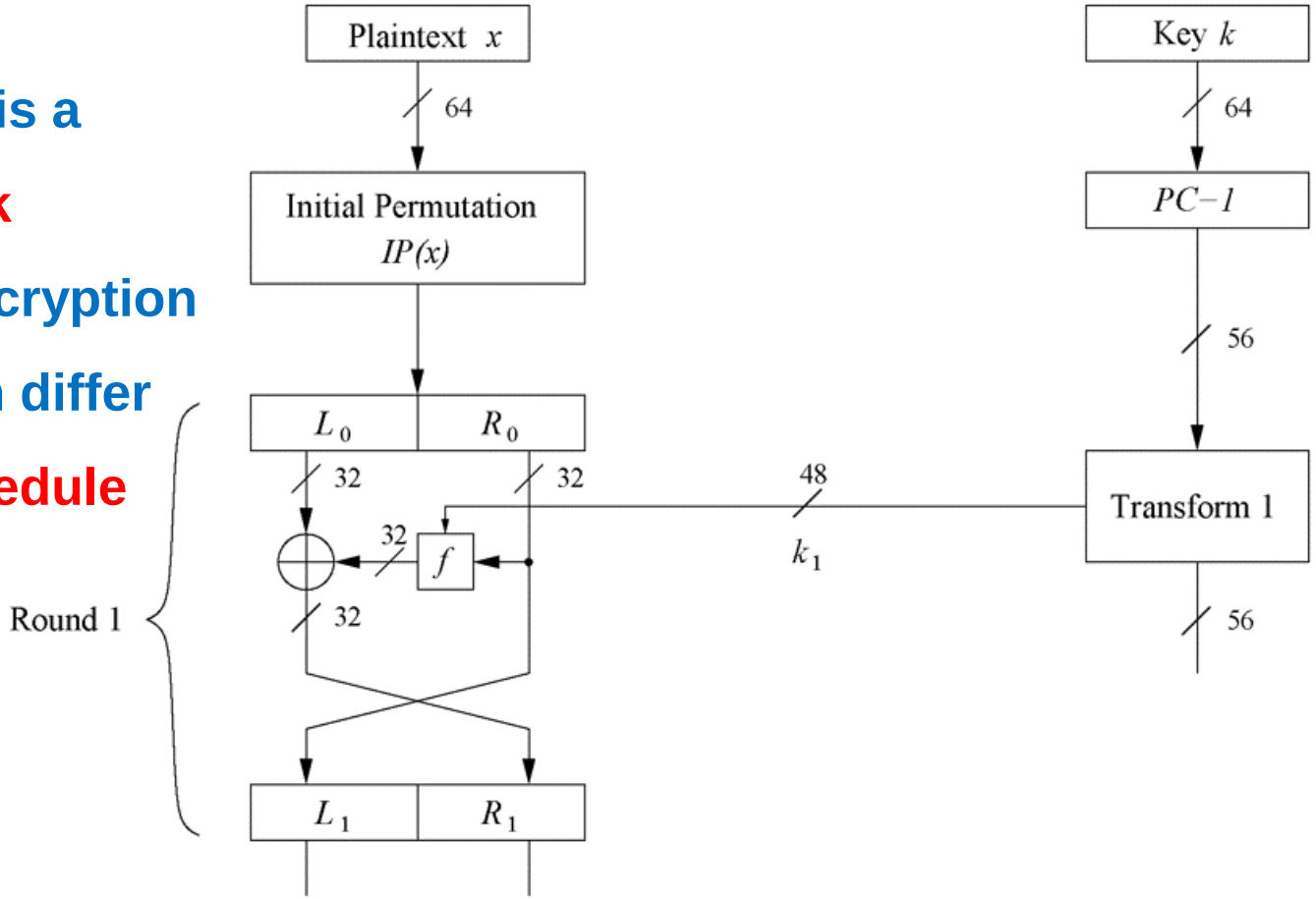
IP^{-1}							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25



Internal Structure of DES

2. DES Encryption Round - Feistel Networks

- DES structure is a **Feistel network**
- Advantage: encryption and decryption differ only in **keyschedule**



Internal Structure of DES

2. DES Encryption Round - Feistel Networks

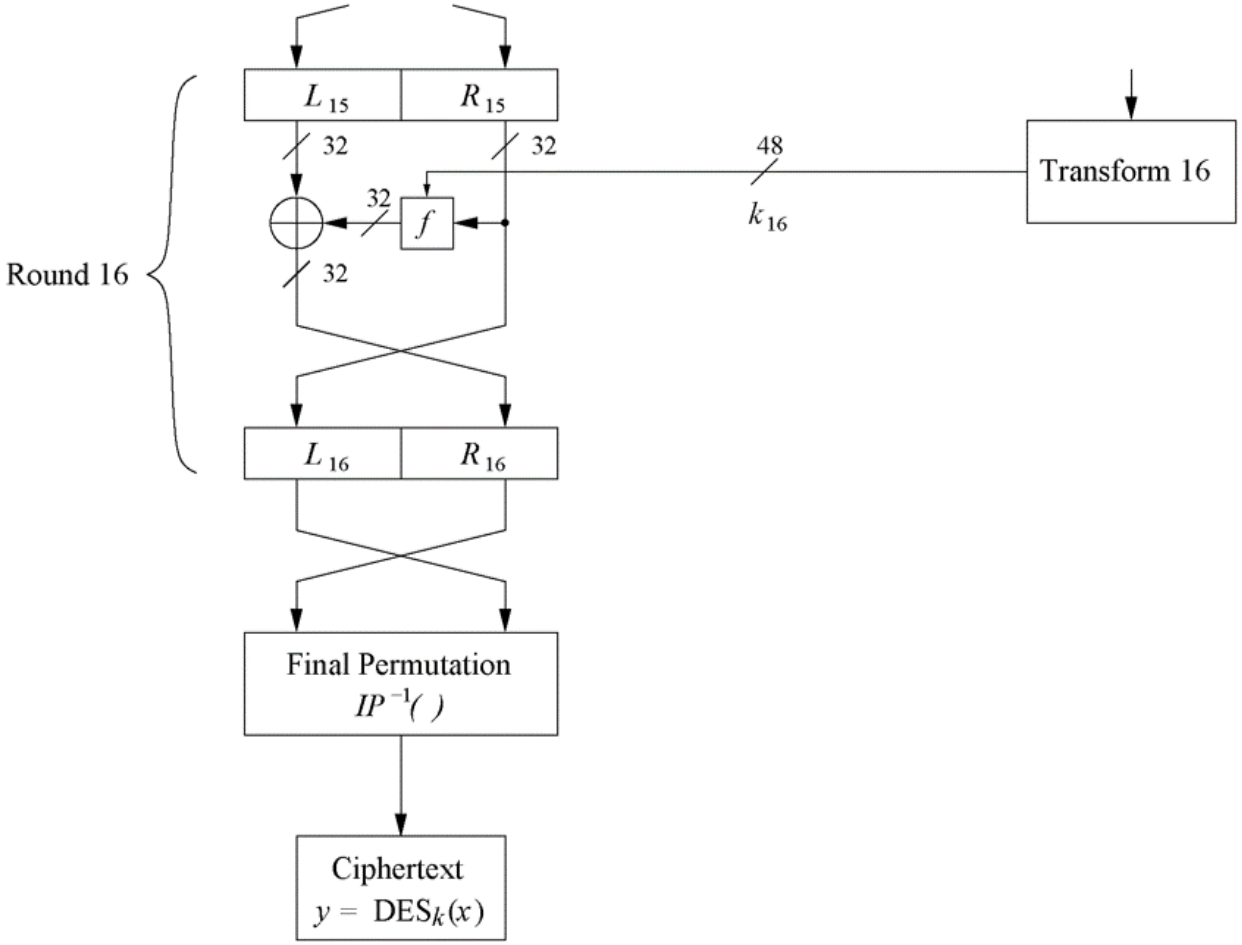
- The encryption block for round 1 takes an input of 64 bit data permuted in the IP
- Plaintext is split into 32-bit halves Li and Ri
- Ri is fed into the function f, the output of which is then XORed with Li
- Left and right half halves are swapped at the end of one encryption round
- Each encryption round can be expressed as :

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$$

Internal Structure of DES

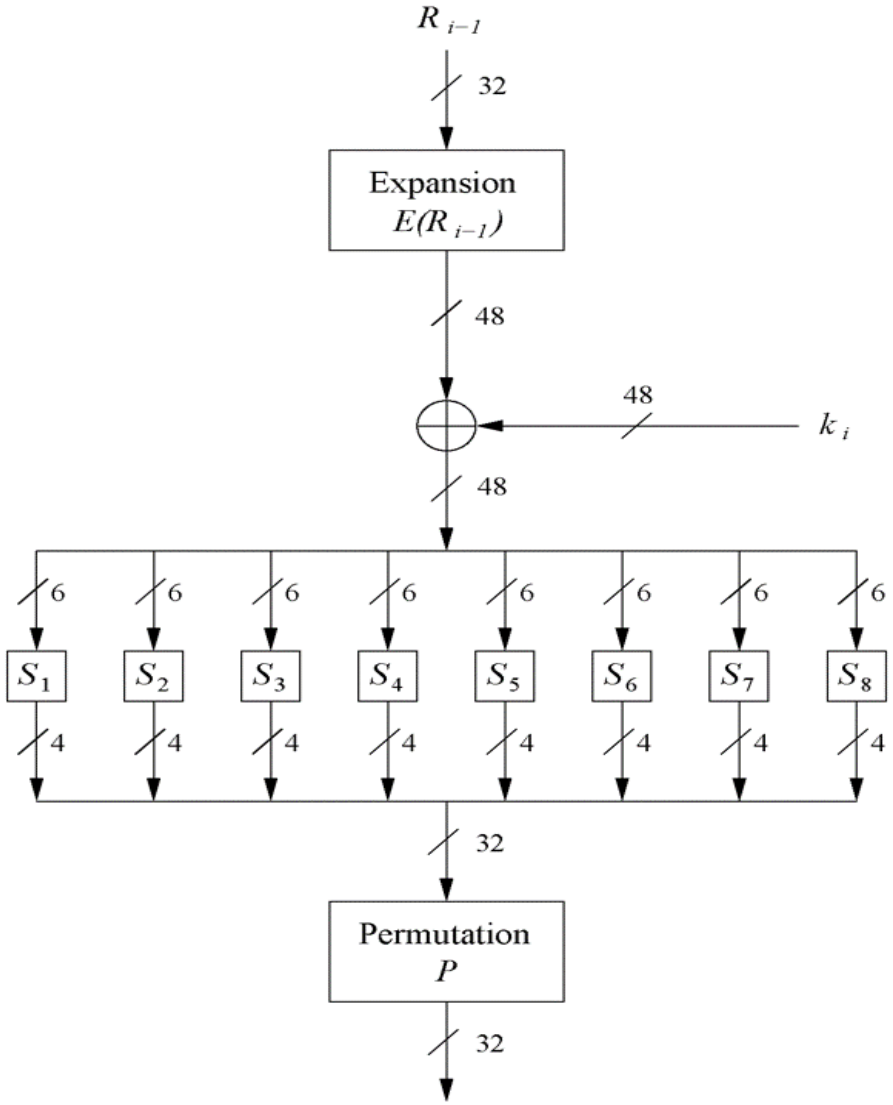
2. DES Encryption Round - Feistel Networks



Internal Structure of DES

3. The f-function

- The main operation of DES
- Inputs to the f function are Ri−1 and round key ki
- It has 4 main steps in:
  - Expansion block E
  - XOR with round key
  - S-box substitution (eight of them)
  - Permutation



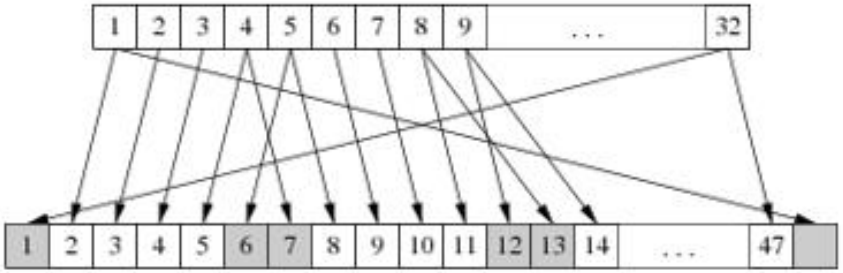
Internal Structure of DES

3. The f-function

3.1 The Expansion function E

The main purpose of the expansion function is to increase diffusion in the input Ri−1 bits.

E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1



Internal Structure of DES

3. The f-function

3.2 XOR with round key

- Bitwise XOR of the round key ki and the output of the expansion function E
- We take a 48-bit expanded message bit and XOR with 48-bit key input and the output data is also 48-bit

3.3 The DES S-Box substitution

- Eight substitution tables which form the core security of DES
- Take 6 bits of input and give 4-bit output
- Non-linear and resistant to differential cryptanalysis

Internal Structure of DES

3. The f-function

3.4 The Permutation P

- This is the last step in the f-function.
- It is also a bitwise permutation, which introduces diffusion.
- Output bits of one S-Box affect several S-Boxes in the next round.
- Diffusion by E, S-Boxes, and P guarantees that after Round 5 every bit is a function of each key bit and each plaintext bit.

P							
16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

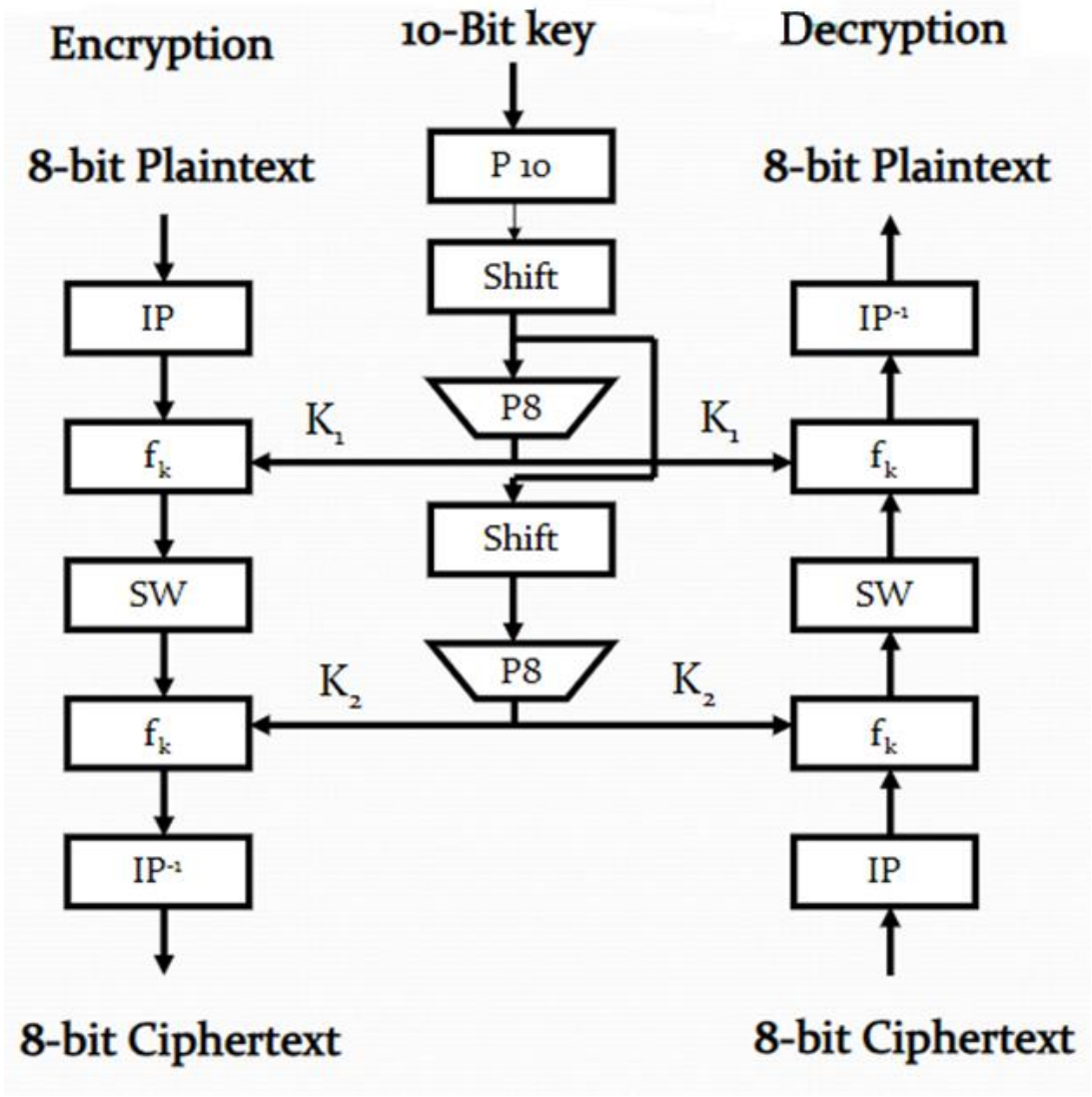
Simplified DES (SDES) Algorithm

SDES is a symmetric-key block cipher, meaning the same key is used for encryption and decryption. SDES operates on 8-bit blocks with a 10-bit key, rather than the 64-bit block and 56-bit key used in the full DES.

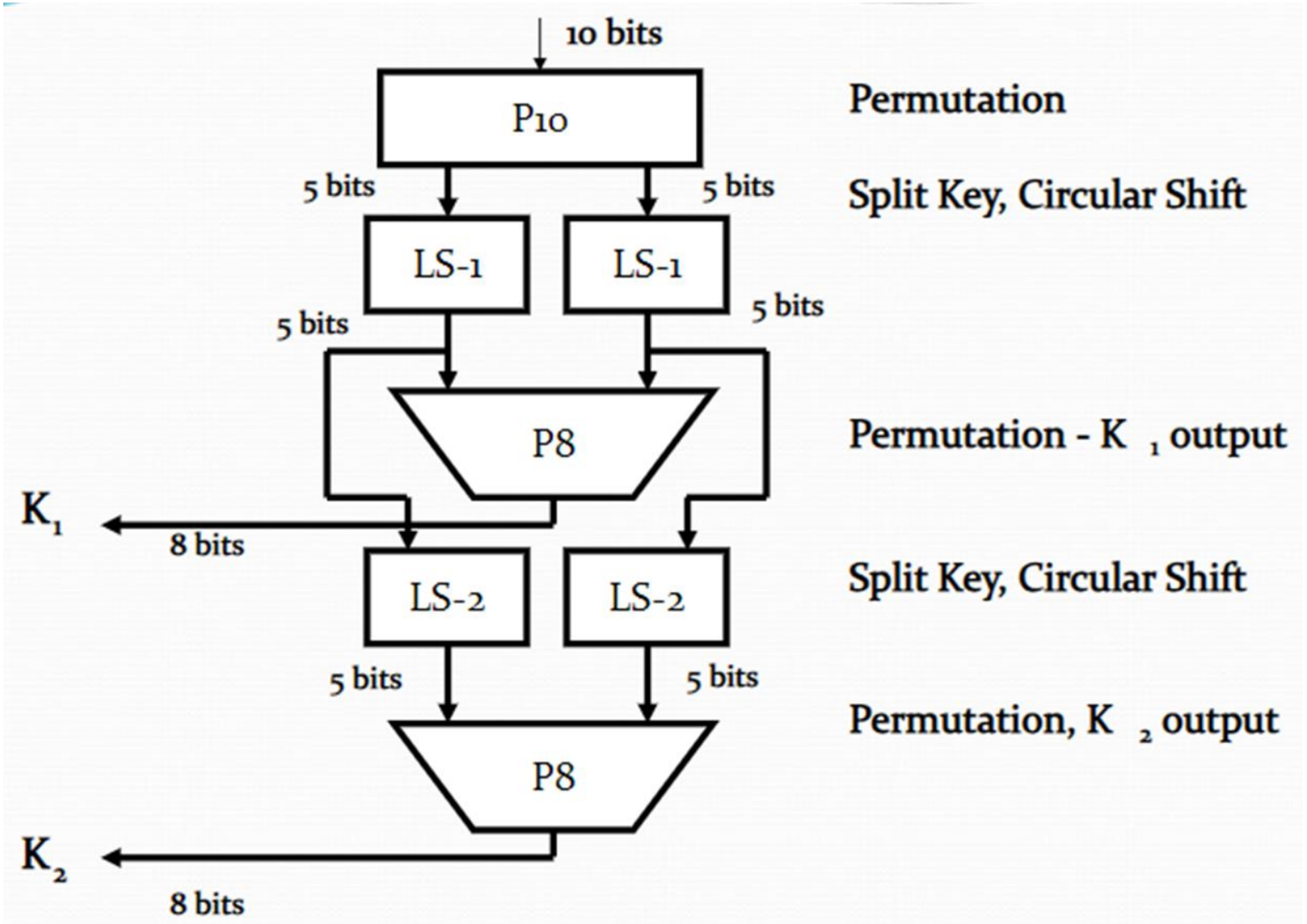
Key Components

- Input Block (Plaintext): 8 bits
- Key: 10 bits (used to generate subkeys)
- Rounds: 2
- Output Block (Ciphertext): 8 bits

SDES Algorithm



1- SDES Key Generation



1- SDES Key Generation

Let the plaintext be the string 0010 1000.

Let the 10-bit key be 1 1 0 0 0 1 1 1 1 0.

The keys k1 and k2 are derived using the functions P10, Shift, and P8

P10 is defined as follows:

P10									
3	5	2	7	4	10	1	9	8	6

P8 is defined to be as follows:

P8							
6	3	7	4	8	5	10	9

1- SDES Key Generation

The first key k1 is therefore equal to:

Bit #	1	2	3	4	5	6	7	8	9	10
K	1	1	0	0	0	1	1	1	1	0
P10	3	5	2	7	4	10	1	9	8	6
P10(K)	0	0	1	1	0	0	1	1	1	1
Shift(P10(K))	0	1	1	0	0	1	1	1	1	0
P8	6	3	7	4	8	5	10	9		
P8(Shift(P10(K)))	1	1	1	0	1	0	0	1		

The second key k2 is derived in a similar manner:

K	1	1	0	0	0	1	1	1	1	0
P10	3	5	2	7	4	10	1	9	8	6
P10(K)	0	0	1	1	0	0	1	1	1	1
Shift^3(P10(K))	1	0	0	0	1	1	1	0	1	1
P8	6	3	7	4	8	5	10	9		
P8(Shift^2(P10(K)))	1	0	1	0	0	1	1	1		

2- Initial and Final Permutation

The plaintext undergoes an initial permutation when it enters the encryption function, IP. It undergoes a reverse final permutation at the end IP^{-1}

- The function IP is defined as follows:

IP							
2	6	3	1	4	8	5	7

- The function IP^{-1} is defined as follows:

IP^{-1}							
4	1	3	5	7	2	8	6

- Applied to the input, we have the following after the initial permutation:

Bit #	1	2	3	4	5	6	7	8
P	0	0	1	0	1	0	0	0
IP	0	0	1	0	0	0	1	0

3- Functions f_K, SW, K

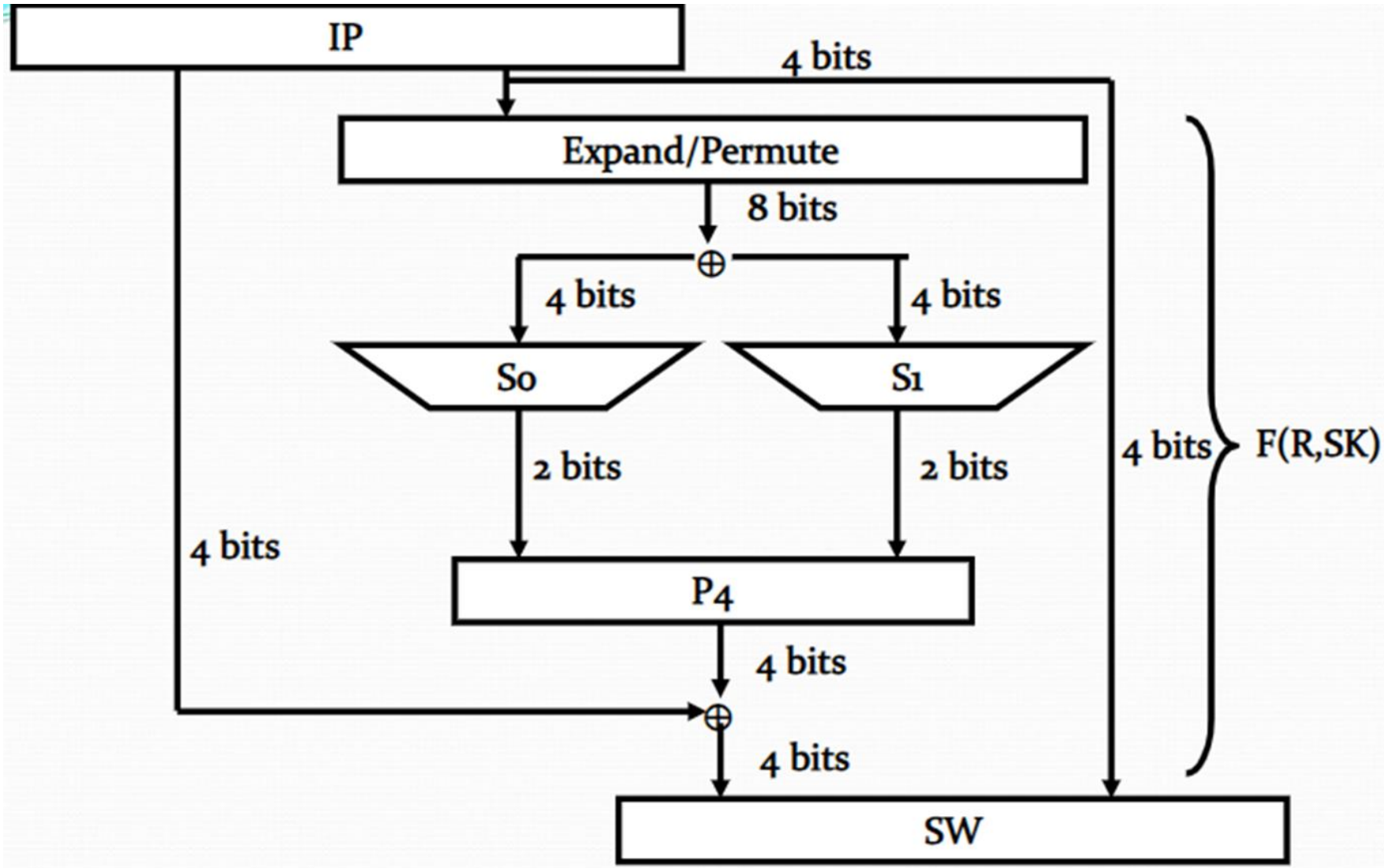
- The function f_k is defined as follows.
  - Let P = (L, R), then $f_K(L, R) = (L \oplus F(R, SK), R)$.
- The function SW just switches the two halves of the plaintext, so SW(L, R) → (R, L)
- The function F(p, k) takes a four-bit string p and eight-bit key k and produces a four-bit output. It performs the following steps.
  1) First, it runs an expansion permutation E/P:

E/P							
4	1	2	3	2	3	4	1

  2) Then it XORs the key with the result of the E/P function
  3) Then it substitutes the two halves based on the S-Boxes.
  4) Finally, the output from the S-Boxes undergoes the P4 permutation:

P4			
2	4	3	1

3- Functions f_K, SW, K



3- Functions f_K, SW, K

Applying the functions, we must perform the following steps:

IP^{-1} ◦ fK2 ◦ SW ◦ fK1 ◦ IP.

- We have already calculated IP(P) = {0010 0010}
- fK1(L, R) = f{1110 1001}(0010 0010)
  = (0010 ⊕ F(0010, {1110 1001}), 0010)
- F(0010, {1110 1001}) = P4 ◦ SBoxes ◦ {1110 1001} ⊕ (E/P(0010))

Bit #	1	2	3	4	5	6	7	8
R	0	0	1	0				
E/P(R)	0	0	0	1	0	1	0	0
k1	1	1	1	0	1	0	0	1
E/P(R) ⊕ k1	1	1	1	1	1	1	0	1
SBoxes(E/P(R) ⊕ k1)	1	0	0	0				
P4(Sboxes(E/P(R) ⊕ k1))	0	0	0	1				

3- Functions f_K, SW, K

S_0

1	0	3	2
3	2	1	0
0	2	1	3
3	1	3	2

S_1

0	1	2	3
2	0	1	3
3	0	1	0
2	1	0	3

3- Functions f_K, SW, K

- The result from F is therefore 0001
- Calculating we then have fk1 (L ⊕ F, R)

  (L, R) = (0010 ⊕ 0001, 0010) = (0011, 0010)
- So far, then L = 0011 and R = 0010. SW just swaps them so R = 0011 and L = 0010
- We now do the calculation of fk2 (L, R) = f{1010 0111}(0010 0011)

  = (0010⊕F(0011, {1010 0111}, 0011))

Bit #	1	2	3	4	5	6	7	8
R	0	0	1	1				
E/P(R)	1	0	0	1	0	1	1	0
k2	1	0	1	0	0	1	1	1
E/P(R) ⊕k2	0	0	1	1	0	0	0	1
SBoxes(E/P(R) ⊕ k2)	1	0	1	0				
P4(Sboxes(E/P(R) ⊕ k2))	0	0	1	1				

3- Functions f_K, SW, K

- So now we have the outcome of F as 0011
- Calculating we then have fk2(L, R) (L ⊕ F, R)

  = (0010 ⊕ 0011, 0011) = (0001, 0011)
- Last, we perform the IP^{-1} permutation:

Bit #	1	2	3	4	5	6	7	8
R,L	0	0	0	1	0	0	1	1
IP^{-1} (R,L)	1	0	0	0	1	0	1	0

- So the final result of the encryption is 1000 1010

SDES Algorithm

Example: Encrypt the following plaintext (1 1 0 1 0 1 1 1), and the key is (1 0 1 0 0 0 0 0 1 0)

Step-by-Step Key Generation

1. Initial Key (K) = 1 0 1 0 0 0 0 0 1 0.
2. Permuted Choice 10 (P10) = [3 5 2 7 4 10 1 9 8 6]

   K' = 1 0 0 0 0 0 1 1 0 0
3. Split K' into two halves, 10000 and 01100, and left shift each half by 1 bit:
   - Left half: 10000 -> 00001
   - Right half: 01100 -> 11000
4. Permuted Choice 8 (P8) = [6 3 7 4 8 5 10 9]

   Combine the shifted halves (0 0 0 0 1 1 1 0 0 0) and apply P8:

   The result is the first subkey, K1 = 1 0 1 0 0 1 0 0

SDES Algorithm

5. Left Shift 2 (LS-2): Shift both halves of (0 0 0 0 1 1 1 0 0 0) by 2 bits to get:

- **Left half: 00001 -> 00100**
- **Right half: 11000 -> 00011**

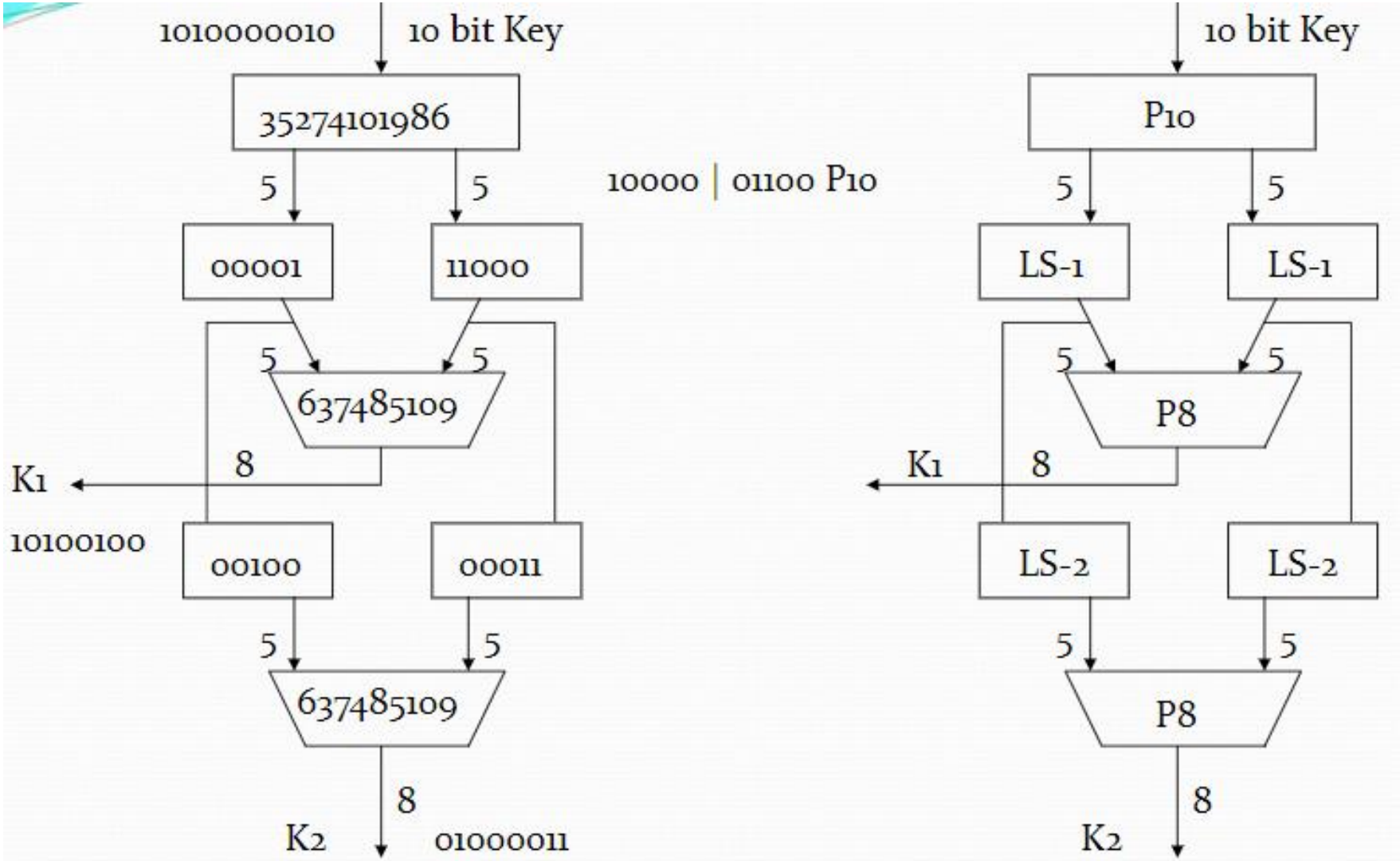
6. Combine the shifted halves (0 0 1 0 0 0 0 0 1 1)

Apply P8 [6 3 7 4 8 5 10 9] Again: K2 = 0 1 0 0 0 0 1 1.

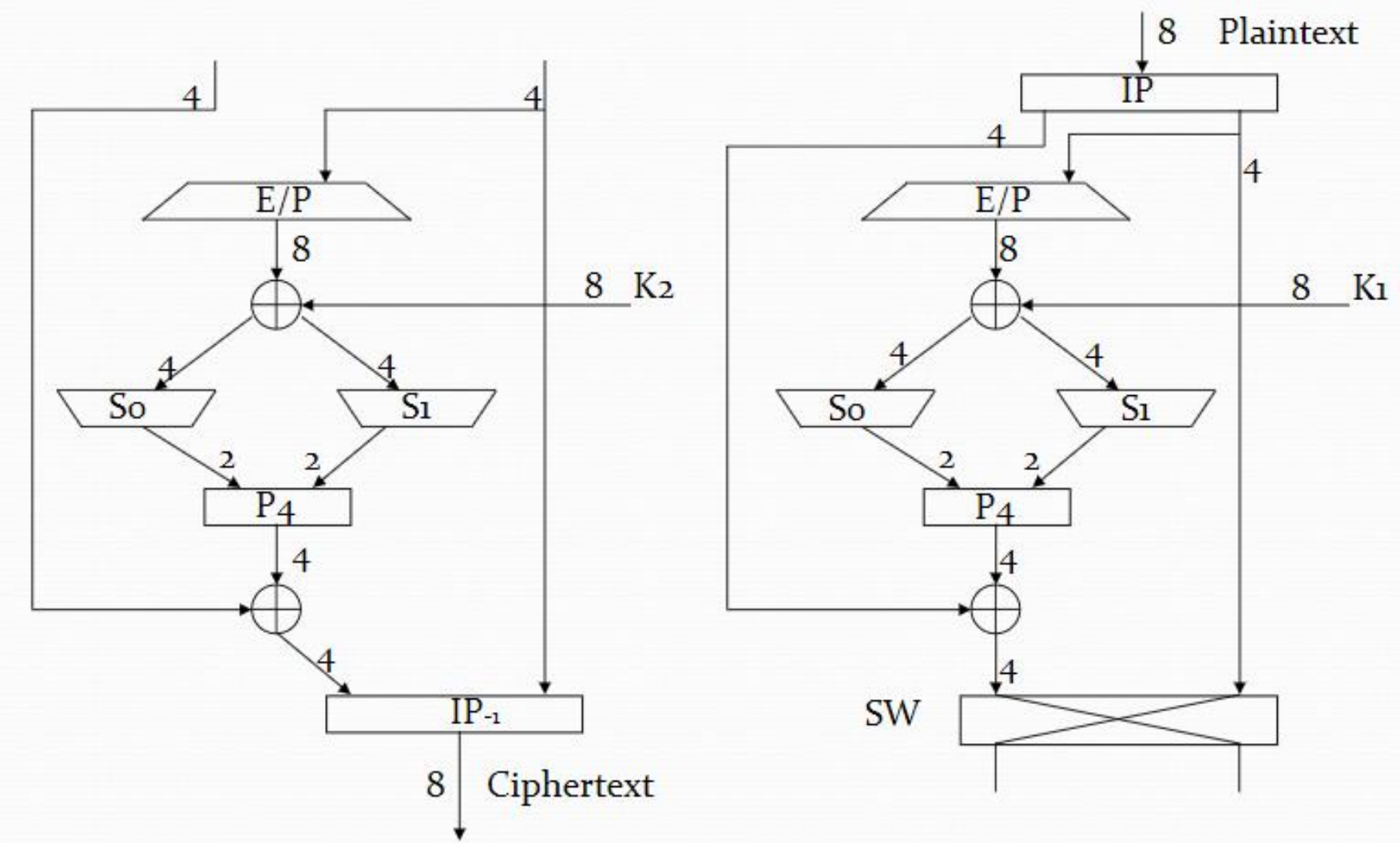
Summary of Key Generation

- **K1: 1 0 1 0 0 1 0 0**
- **K2: 0 1 0 0 0 0 1 1**

SDES Algorithm



SDES Algorithm



SDES Algorithm

Encryption Steps

1. **Initial Permutation (IP): Rearrange the plaintext bits using the IP permutation: IP = [2 6 3 1 4 8 5 7]**
   - **Plaintext 1 1 0 1 0 1 1 1**
   - **IP produces 1 1 0 1 1 1 0 1**
2. **Round 1 with K1:**
   - **Divide the permuted block into left (1101) and right (1101) halves.**
   - **Expansion (E/P) on Right Half: Expand the 4-bit right half using E/P = [4 1 2 3 2 3 4 1]**
     For 1101, E/P produces 1 1 1 0 1 0 1 1.

SDES Algorithm

Encryption Steps

- XOR with K1: E/P = 1 1 1 0 1 0 1 1
  with K1 = 1 0 1 0 0 1 0 0
  giving 0 1 0 0 1 1 1 1
- S-Box Substitution: $[S_0S_0S_0S_0]$,$[S_1S_1S_1S_1]$

S-Box 0 (S_0):

1	0	3	2
3	2	1	0
0	2	1	3
3	1	3	2

S−Box 1 (S_1):

0	1	2	3
2	0	1	3
3	0	1	0
2	1	0	3

- For 0100 inputs S_0 produce (3): 11
- For 1111 inputs S_1 produce (3): 11
- P4 Permutation: Rearrange 1111 using P4 = [2 4 3 1]
  Result: 1 1 1 1

SDES Algorithm

Encryption Steps

- The result from Round 1 (F) is therefore 1111
- IP produces 1 1 0 1 1 1 0 1
- Calculating fk1

  (L, R) = (L⊕ F, R) =

  (L, R) = (1 1 0 1 ⊕ 1 1 1 1, 1 1 0 1) = (0010, 1101)
- Then L = 0010 and R = 1101
- SW them so R = 0010 and L = 1101
- fk2 (L, R) = f{K2}(L R)

  = (1101 ⊕F(0010, {01000011}, 0010))
- E/P(R) = (00010100)
- E/P(R) ⊕ K2 = (00010100) ⊕ (01000011)

  = 01010111

E/P							
4	1	2	3	2	3	4	1

SDES Algorithm

Encryption Steps

- S-Box Substitution: 0101 0111 $[S_0S_0S_0S_0],[S_1S_1S_1S_1]$

S_0_Raw: 01 , S_0_Col: 10 S_1_Raw: 01 , S_1_Col: 11

S-Box 0 (S_0): $\begin{vmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{vmatrix}$ S−Box 1 (S_1): $\begin{vmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{vmatrix}$

- For 0101 inputs S_0 produce (1) 01
- For 0111 inputs S_1 produce (3) 11
- P4 Permutation: Rearrange 0111 using P4 = [2 4 3 1]

Result: 1 1 1 0

SDES Algorithm

Encryption Steps

- Result F: 1 1 1 0
- Calculating we then have fk2(L, R)
  - The last R = 0010 and L = 1101

    = (1101 $\oplus$ 1 1 1 0, 0010) = (0011 0010)
- Last, we perform the IP^{-1} permutation:

IP^{-1}							
4	1	3	5	7	2	8	6

- So the final result of the encryption is

  1010 1000

Advanced Encryption Standard (AES) Algorithm

Advanced Encryption Standard (AES) is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001. AES is widely used today as it is much stronger than DES and triple DES despite being harder to implement.

Points to Remember

- AES is a Block Cipher.
- The key size can be 128/192/256 bits.
- Encrypts data in blocks of 128 bits each.

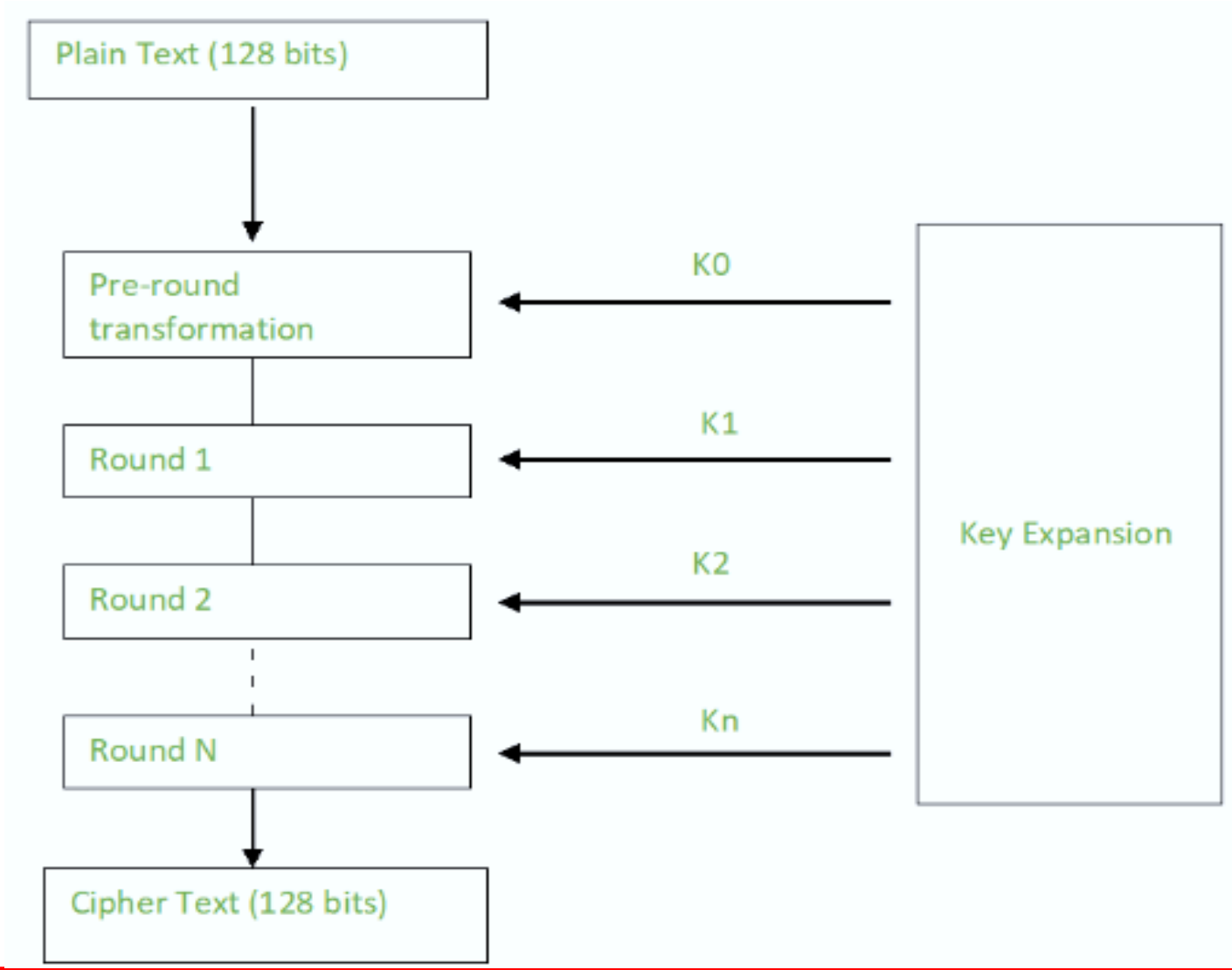
The number of rounds depends on the key length as follows:

128-bit key – 10 rounds

192-bit key – 12 rounds

256-bit key – 14 rounds

AES Algorithm



AES Algorithm

Encryption

Each round comprises of 4 steps :

- SubBytes
- ShiftRows
- MixColumns
- Add Round Key

Decryption

The stages of each round of decryption are as follows :

- Add round key
- Inverse MixColumns
- ShiftRows
- Inverse SubByte

Rivest Cipher 4 (RC4) Algorithm

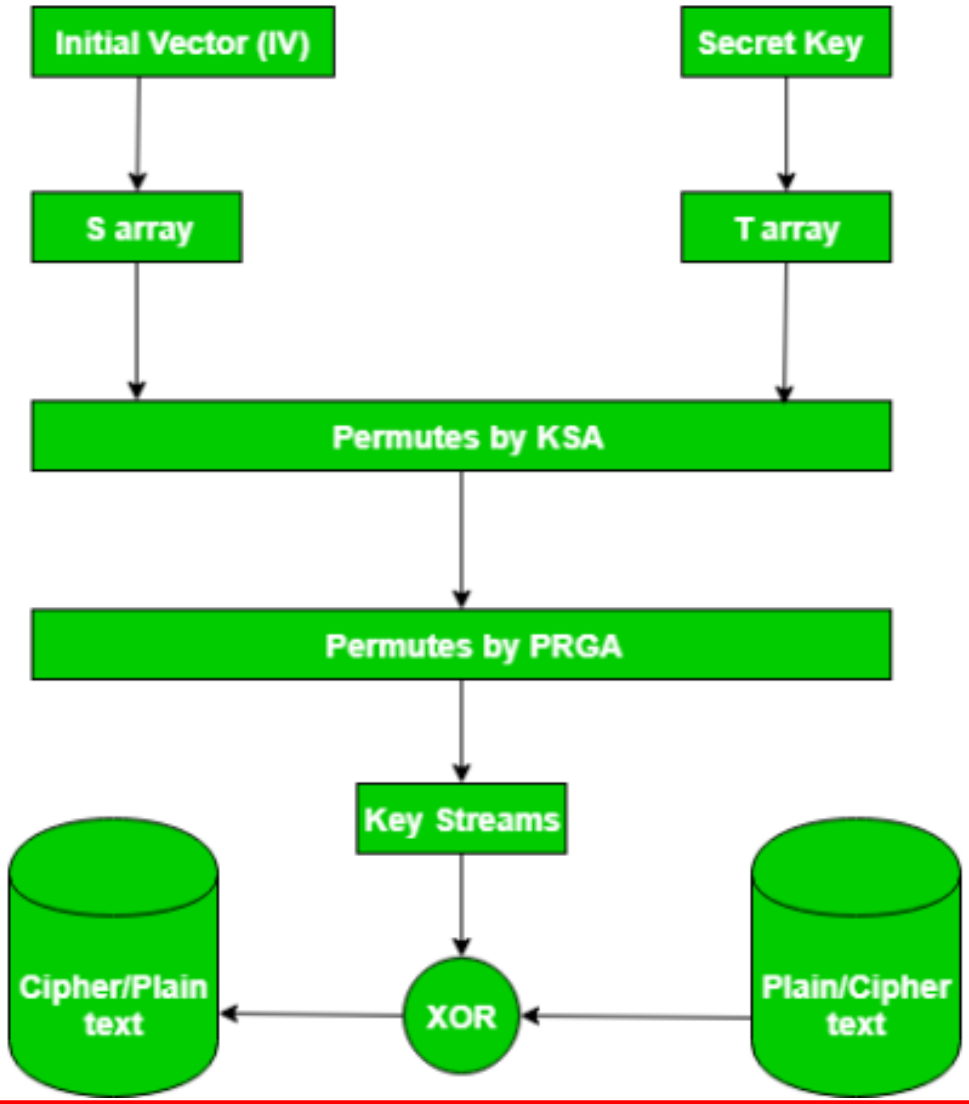
RC4 is a stream cipher and variable-length key algorithm. This algorithm encrypts one byte at a time (or larger units at a time). A key input is a pseudorandom bit generator that produces a stream 8-bit number that is unpredictable without knowledge of the input key, The output of the generator is called key-stream, and is combined one byte at a time with the plaintext stream cipher using X-OR operation.

RC4 Algorithm

Steps of RC4 Algorithm

- **Key Scheduling (KSA):** Initialize the S array with values from 0 to 255. Shuffle the S array based on the key bytes using the encryption key.
- **Pseudo-Random Generation (PRGA):** Continuously modify the S array to produce a stream of pseudo-random bytes. XOR each byte of the keystream with the corresponding byte of plaintext to produce ciphertext.

RC4 Algorithm



RC4 Algorithm

Features of the RC4 encryption algorithm:

- **Symmetric key algorithm:** RC4 is a symmetric key encryption algorithm
- **Stream cipher algorithm:** RC4 is a stream cipher algorithm
- **Variable key size:** RC4 supports variable key sizes, from 40 bits to 2048 bits.
- **Fast and efficient:** RC4 is a fast and efficient encryption algorithm that is suitable for low-power devices and applications.
- **Widely used:** RC4 has been widely used in various applications, including wireless networks, secure sockets layer (SSL), virtual private networks (VPN).
- **Vulnerabilities:** RC4 has several vulnerabilities, including a bias in the first few bytes of the keystream,, RC4 is no longer recommended for use in new applications.

RC4 Algorithm

Advantages:

- Fast and efficient
- Simple to implement
- Variable key size
- Widely used

Disadvantages:

- Vulnerabilities
- Security weaknesses
- Limited key length
- Not recommended for new applications

Cryptography and Network Security I

5th Lecture

Asymmetric Cryptography Techniques

By

Dr. Taqwa F. Hassan

Edited by

Dr. Ali Albu-Rghaif

Department of Computer Engineering

College of Engineering - University of Diyala

Rivest Shamir Adleman Algorithm (RSA)

The RSA crypto scheme is currently the most widely used public key asymmetric cryptographic scheme.

There are many applications for RSA, but in practice it is most often used for:

- encryption of small pieces of data, especially for key transport.
- digital signatures, (digital certificates on the Internet)

Rivest Shamir Adleman Algorithm (RSA)

It should be noted that RSA is slow due to large number of computations involved in performing RSA (or any other public-key algorithm).

The main use of the encryption feature is to securely exchange a key for a symmetric cipher (key transport).

RSA strength is in the integer factorization problem: Multiplying two large primes is computationally easy but factoring the resulting product is very hard.

RSA Encryption and Decryption

RSA Key Generation

Steps to find public key k_{pub}= (n, e) and private key k_{pr}= (d)

Step 1: Choose two large prime numbers **p** and **q**

Step 2: Compute $n = p.\ q$.

Step 3: Compute $\Phi(n) = (p - 1)(q - 1)$.

Step 4: Select the public exponent $e \in \{1,2,\ldots, \Phi(n) - 1\}$ such that

$GCD(e,\Phi(n)) = 1$.

Step 5: Compute the private key d such that $d = e^{(\Phi(n)-1)} \ mod\ \Phi(n)$.

RSA Encryption and Decryption

RSA Encryption

Given the public key $k_{pub} = (n, e)$ and the plaintext x, RSA encryption function is:

$$y = E_{k_{pub}}(x) = x^e \bmod n$$

RSA Decryption

Given the private key k_{pr} **= (d)** and the ciphertext y, RSA decryption function is:

$$x = D_{k_{pr}}(y) = y^d \bmod n$$

RSA Encryption and Decryption

Example:

Encrypt then decrypt the following message “x=4” using RSA algorithm and k_{pub} = (33,3).

Solution:

Encryption in the sender side:

$\because k_{pub} = (n, e) \therefore n = 33 \; and \; e = 3$

$y = E_{k_{pub}}(x) = x^e \; mod \; n \Rightarrow E_{k_{pub}}(4) = 4^3 \; mod \; 33 = 31$.

RSA Encryption and Decryption

To decrypt the ciphertext in the receiver side, it is necessary to generates RSA key first then decrypt the ciphertext using the generated RSA key.

Step 1: Choose two large prime numbers p and q

Let p=3 and q=11 (two prime numbers)

Step 2: Compute $n = p.q$.

$n = 3 * 11 = 33$

Step 3: Compute $\Phi(n) = (p - 1)(q - 1)$.

$\Phi(n) = (3 - 1)(11 - 1) = 20$

Step 4: Select the public exponent $e \in \{ 1,2,\ldots, \Phi(n) - 1\}$ such that

$GCD(e,\Phi(n)) = 1$.

Let $e = 3 \in \{ 1,2,\ldots,19\}$ and $GCD(3,20) = 1$.

RSA Encryption and Decryption

Step 5: Compute the private key d such that $d = e^{(\Phi(n)-1)} \; mod \; \Phi(n)$.

$$d = e^{(20-1)} \; mod \; 20 = 7 = k_{pr}$$

The decryption formula for RSA algorithm is:

$$x = D_{k_{pr}}(y) = \; y^d \; mod \; n$$

Plaintext = $D_{k_{pr}}(31) = \; 31^7 \; mod \; 33$

= 4

Practical RSA parameters are much, much larger. The RSA modulus n should be at least 1024 bits long, which results in a bit length for p and q of 512 bits.

RSA Encryption and Decryption

Example:

Encrypt then decrypt the following message “x=9” using RSA algorithm, where the uses prime numbers 7 and 11

Solution:

Encryption in the sender side:

n = p x q = 7 x 11 = 77

Compute $\Phi(n) = (p - 1)(q - 1)$.

$\Phi(n) = (7 - 1)(11 - 1) = 60$

Select the public exponent $e \in \{1,2,\ldots, \Phi(n) - 1\}$ such that $GCD(e,\Phi(n)) = 1$.

Let $e = 7 \in \{1,2,\ldots,59\}$ and $GCD(7,60) = 1$.

$\because k_{pub} = (n, e) \therefore n = 77$ and $e = 7$

$y = E_{k_{pub}}(x) = x^e \ mod \ n \ \Rightarrow E_{k_{pub}}(9) = 9^7 \ mod \ 77 = 37$.

RSA Encryption and Decryption

Compute the private key d such that $d = e^{(\Phi(n)-1)} \ mod \ \Phi(n)$.

$d = 7^{(60-1)} \ mod \ 60 = 43 = k_{pr}$

The decryption formula for RSA algorithm is:

$x = D_{k_{pr}}(y) = \ y^d \ mod \ n$

Plaintext = $D_{k_{pr}}(43) = \ 37^{43} \ mod \ 77$

= 9

In this example, Plain text = 9 and the ciphertext = 37

Diffie–Hellman (DH)

The Diffie–Hellman (DH) Algorithm is a key-exchange protocol that enables two parties communicating over public channel to establish a mutual secret without it being transmitted over the Internet. DH enables the two to use a public key to encrypt and decrypt their conversation or data using symmetric cryptography.

Diffie–Hellman (DH)

Diffie-Hellman is generally explained by two sample parties, Alice and Bob, initiating a dialogue. Each has a piece of information they want to share, while preserving its secrecy. To do that they agree on a public piece of benign information that will be mixed with their privileged information as it travels over an insecure channel. Their secrets are mixed with the public information, or public key, and as the secrets are exchanged the information they want to share is commingled with the common secret. As they decipher the other's message, they can extract the public information and with knowledge of their own secret, deduce the new information that was carried along. While seemingly uncomplicated in this method's description, when long number strings are used for private and public keys, decryption by an outside party trying to eavesdrop is mathematically infeasible even with considerable resources.

Diffie Hellman Key Exchange

As the name suggests,

- This algorithm is used to exchange the secret key between the sender and the receiver.
- This algorithm facilitates the exchange of secret key without actually transmitting it.

Diffie Hellman Key Exchange

Alice	Bob
Public Keys available = P, G	Public Keys available = P, G
Private Key Selected = a	Private Key Selected = b
Key generated = $x = G^a mod P$	Key generated = $y = G^b mod P$
Exchange of generated keys takes place	
Key received = y	key received = x
Generated Secret Key = $k_a = y^a mod P$	Generated Secret Key = $k_b = x^b mod P$

Diffie Hellman Key Exchange Algorithm

Let-

- Private key of the sender = X_s
- Public key of the sender = Y_s
- Private key of the receiver = X_r
- Public key of the receiver = Y_r

Using Diffie Hellman Algorithm, the key is exchanged in the following steps

Diffie Hellman Key Exchange Algorithm

Step-01:

- One of the parties chooses two numbers **'G'** and **'P'** and exchange with the other party.
- 'G' is the primitive root of the prime number **'P'**.
- After this exchange, both parties know the value of **'G'** and **'P'**.

Diffie Hellman Key Exchange Algorithm

Step-02:

- Both the parties already know their own private key.
- Both the parties calculate the value of their public key and exchange with each other.

Sender calculate its public key as-

$$Y_s = G^{X}_{s} \bmod P$$

Receiver calculate its public key as-

$$Y_r = G^{X}_{r} \bmod P$$

Diffie Hellman Key Exchange Algorithm

Step-03:

- Both the parties receive public key of each other.
- Now, both the parties calculate the value of secret key.

Sender calculates secret key as-

$$\textbf{Secret key} = (Y_r)^{X_s} \bmod P$$

Receiver calculates secret key as-

$$\textbf{Secret key} = (Y_s)^{X_r} \bmod P$$

Diffie Hellman Key Exchange Examples

Example:

Suppose that two parties A and B wish to set up a common secret key (D-H key) between themselves using the Diffie Hellman key exchange technique. They agree on 7 as the modulus and 3 as the primitive root. Party A chooses 2 and party B chooses 5 as their respective secrets.

Solution:

Given-

P = 7

G = 3

Private key of A = 2

Private key of B = 5

Diffie Hellman Key Exchange Examples

Solution:

Step-01:

Both the parties calculate the value of their public key and exchange with each other.

Public key of A

$= 3^{\text{private key of A}} \bmod 7$

$= 3^2 \bmod 7$

$= 2$

Public key of B

$= 3^{\text{private key of B}} \bmod 7$

$= 3^5 \bmod 7$

$= 5$

Diffie Hellman Key Exchange Examples

Solution:

Step-02:

Both the parties calculate the value of secret key at their respective side.

Secret key obtained by A

$= 5^{\text{private key of A}} \bmod 7$

$= 5^2 \bmod 7$

$= 4$

Secret key obtained by B

$= 2^{\text{private key of B}} \bmod 7$

$= 2^5 \bmod 7$

$= 4$

Finally, both the parties obtain the same value of secret key.

Diffie Hellman Key Exchange Examples

Example:

In a Diffie-Hellman Key Exchange, Alice and Bob have chosen prime value P = 17 and primitive root G = 5. If Alice's secret key is 4 and Bob's secret key is 6, what is the secret key they exchanged?

Solution:

Given-

P = 17

G = 5

Private key of Alice = 4

Private key of Bob = 6

Diffie Hellman Key Exchange Examples

Solution:

Step-01:

Both the parties calculate the value of their public key and exchange with each other.

Public key of Alice

$= 5^{\text{private key of Alice}} \bmod 17$

$= 5^4 \bmod 17$

$= 13$

Public key of Bob

$= 5^{\text{private key of Bob}} \bmod 17$

$= 5^6 \bmod 17$

$= 2$

Diffie Hellman Key Exchange Examples

Solution:

Step-02:

Both the parties calculate the value of secret key at their respective side.

Secret key obtained by Alice

$= 2^{\text{private key of Alice}} \bmod 17$

$= 2^4 \bmod 17$

$= 16$

Secret key obtained by Bob

$= 13^{\text{private key of Bob}} \bmod 17$

$= 13^6 \bmod 17$

$= 16$

Finally, both the parties obtain the same value of secret key.

Diffie Hellman Key Exchange Examples

Example :

Step 1: Alice and Bob get public numbers P = 23, G = 9

Step 2: Alice selected a private key a = 4 and
Bob selected a private key b = 3

Step 3: Alice and Bob compute public values
Alice: x =(9^4 mod 23) = (6561 mod 23) = 6
Bob: y = (9^3 mod 23) = (729 mod 23) = 16

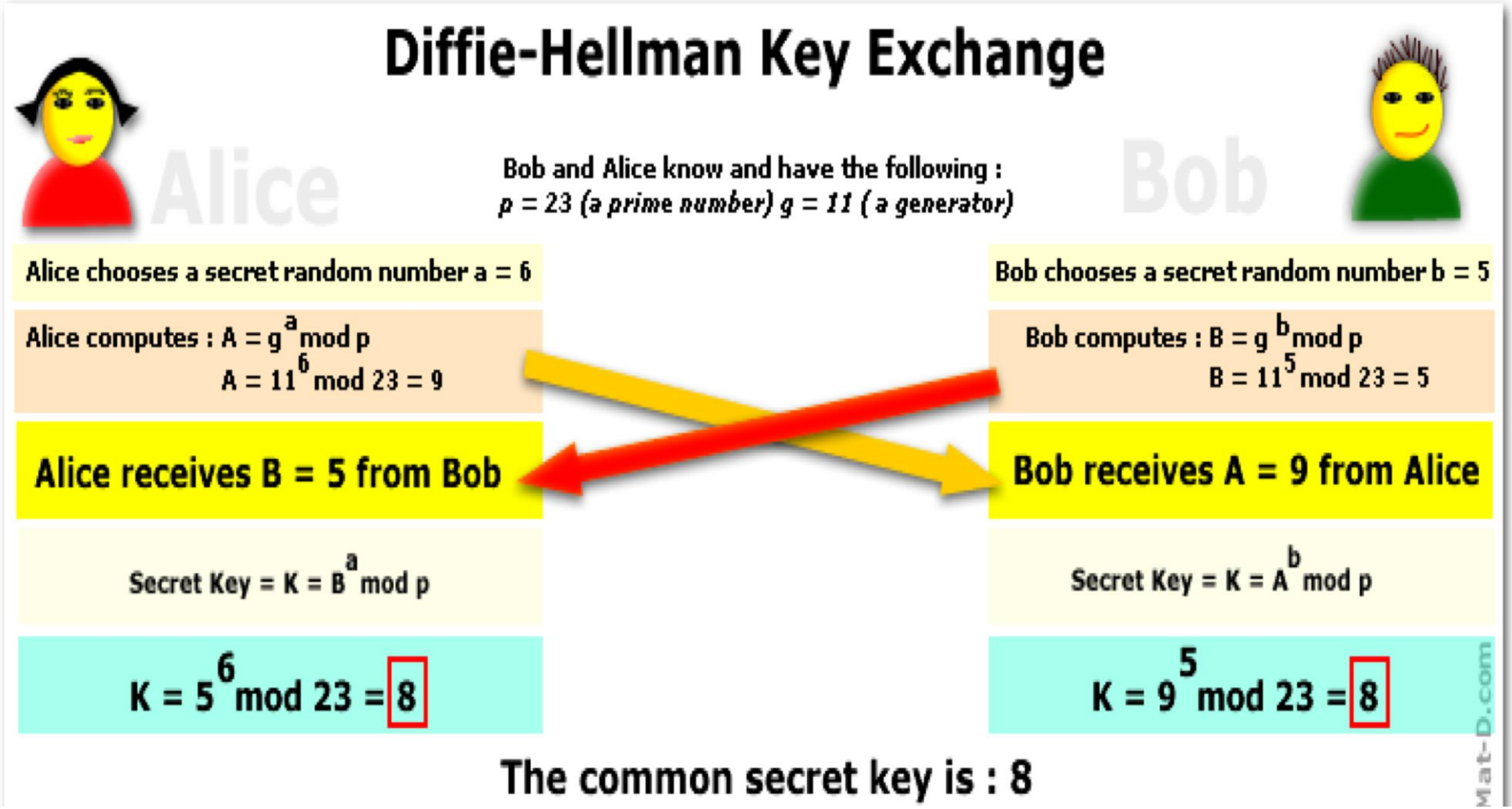
Step 4: Alice and Bob exchange public numbers

Step 5: Alice receives public key y =16 and
Bob receives public key x = 6

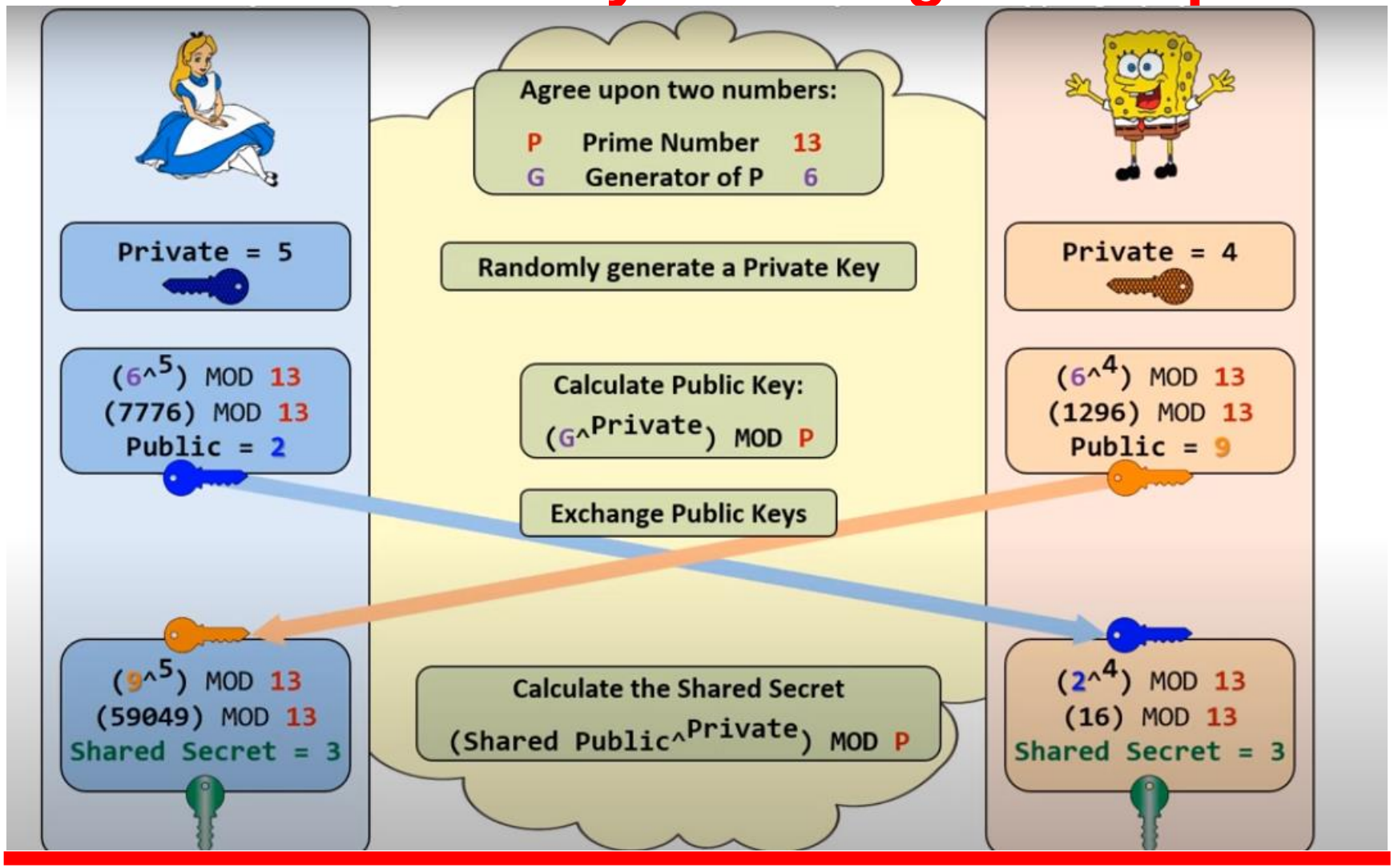
Step 6: Alice and Bob compute Secret keys
Alice: ka = y^a mod p = (16^4) mod 23 = 65536 mod 23 = 9
Bob: kb = x^b mod p = (6^3) mod 23 = 216 mod 23 = 9

Step 7: 9 is the shared secret.

Diffie Hellman Key Exchange Examples



Diffie Hellman Key Exchange Examples



Cryptography and Network Security I

6th Lecture

Shift Register

By

Dr. Taqwa F. Hassan

Edited by

Dr. Ali Albu-Rghaif

Department of Computer Engineering

College of Engineering - University of Diyala

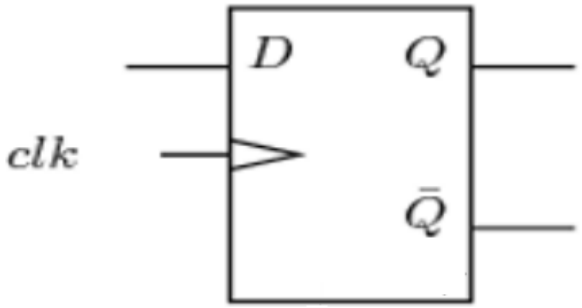
Shift Register

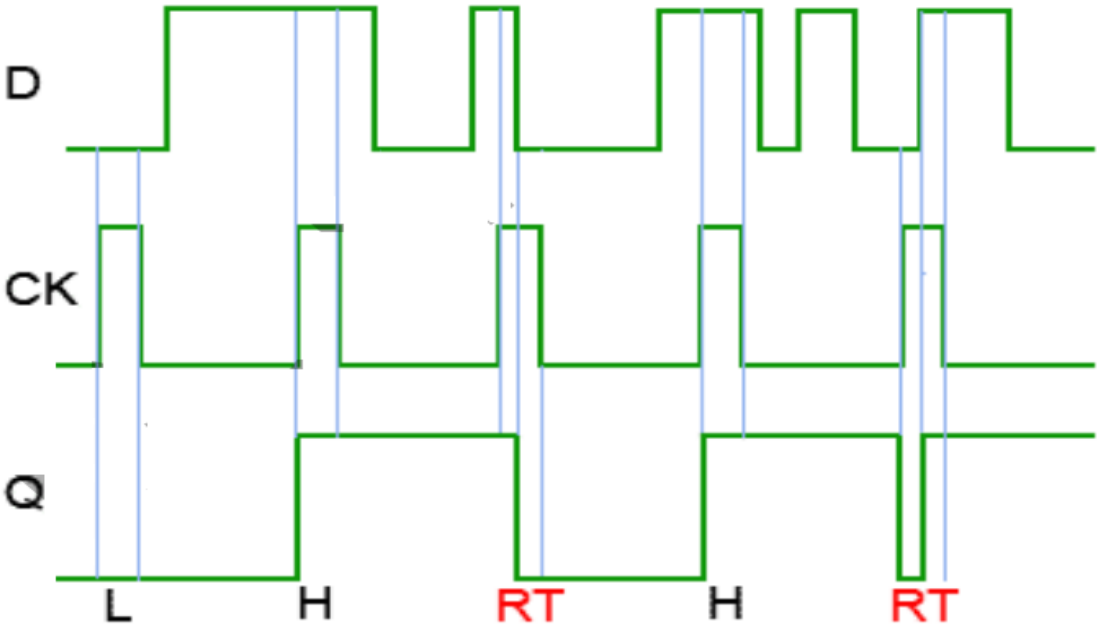
Flip Flops Overview

A flip flop is an electronic circuit with two stable states that can be used to store binary data. The stored data can be changed by applying varying inputs. In this course design D flip-flop is going to be used.

Data flip flop (D flip-flop) tracks the input, making transitions with match those of the input D. This flip-flop stores the value that is on the data line. It can be thought of as a basic memory cell.

Clock	D	Q	Q'
↓ » 0	X	Q	Q'
↑ » 1	0	0	1
↑ » 1	1	1	0

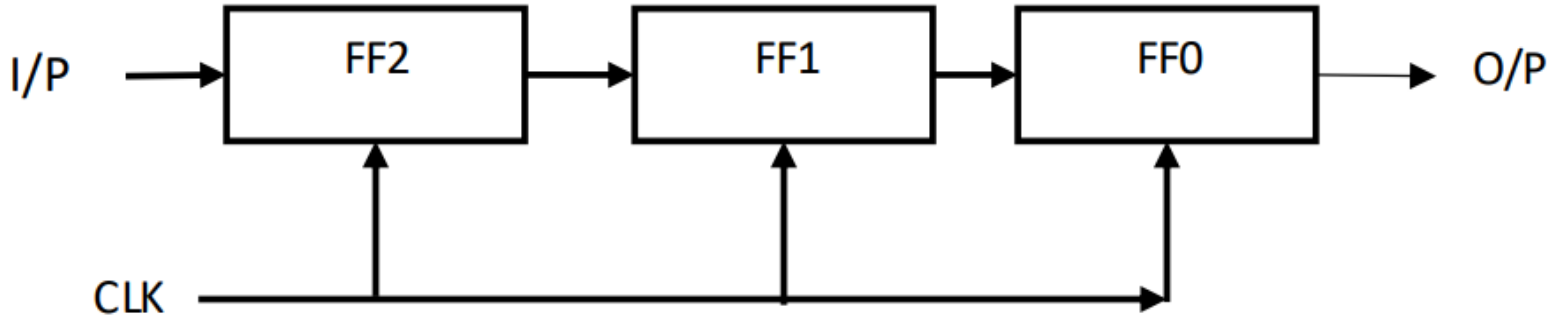




Shift Register

Example: Design shift register using 3D flip-flops and initial state (100)

Solution: 3D flip-flops are going to connect sequentially with no feedback.



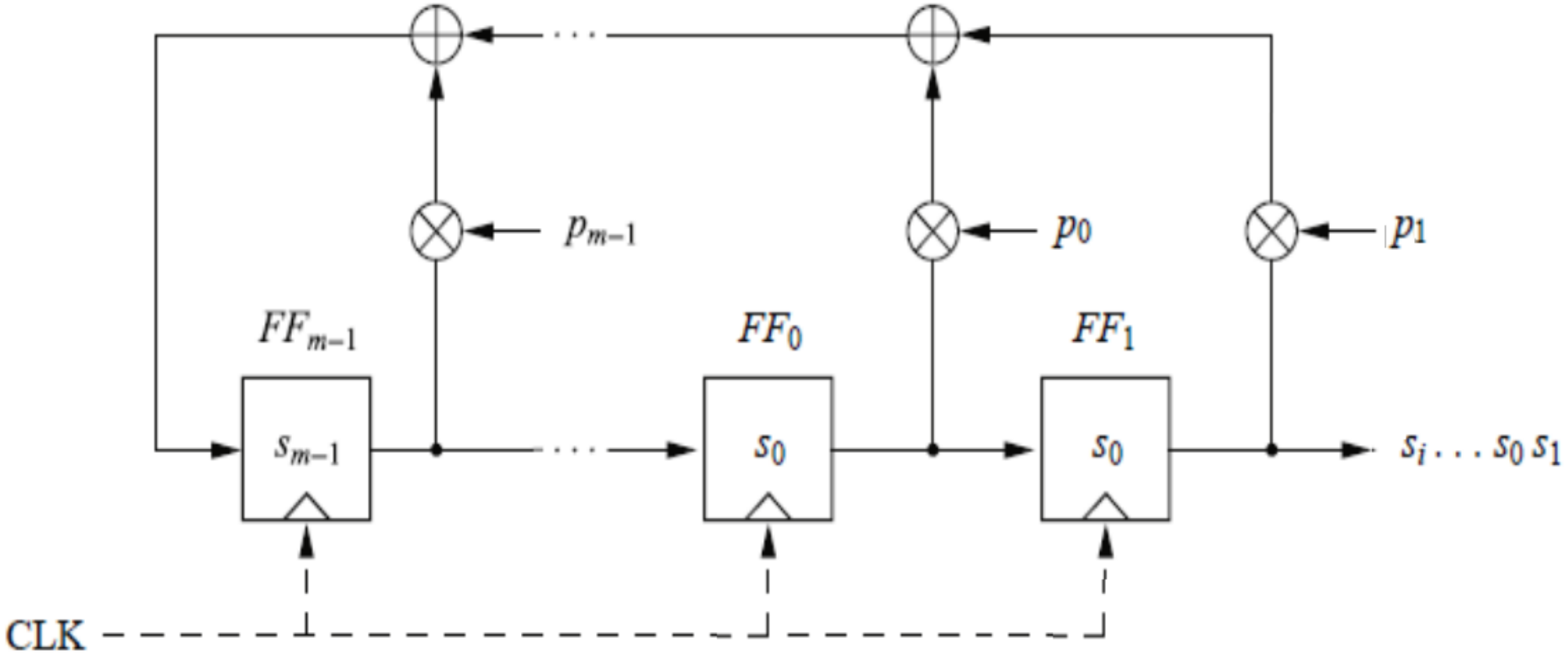
I/P	FF2	FF1	FF0	O/P
NON	1	0	0	
NON	X	1	0	0
NON	X	X	1	0
NON	X	X	X	1

It is impossible to take all zeroes as an initial state.

After 3 clocks pulses the output would be end because there is no input, no feedback to provide an input.

Linear Feedback Shift Registers (LFSR)

An LFSR consists of clocked storage elements (flip-flops) and a feedback path, as shown in the figure bellow:



Linear Feedback Shift Registers (LFSR)

The number of storage elements gives the degree of the LFSR. In other words, an LFSR with m flip-flops is said to be of degree m. The feedback network computes the input for the last flip-flop as XOR-sum of certain flip-flops in the shift register.

There are specific terms that are necessary to be found in order to design the required LFSR, which are:

LFSR degree (m): is the number of flip-flops that construct it.

LFSR maximum sequence length **= $2^m - 1$** of nonzero states

Notes:

- **All zero state must be excluded. If an LFSR assumes this state, it will get "stuck" in it, i.e., it will never be able to leave it again.**
- **LFSR actual sequence length is not necessary is the same length as the maximum length.**

Linear Feedback Shift Registers (LFSR)

Example:

For LFSR with m=4, find the degree, the maximum sequence length and the number of D flip-flops that construct it.

Solution:

LFSR degree = Number of D flip-flops = m = 4

maximum sequence length = $\mathbf{2^m - 1}$

$= \mathbf{2^4 - 1}$

$= 15$

Linear Feedback Shift Registers (LFSR)

Using Polynomials in LFSR Representation

In real life polynomials are usually used to represents LFSR, due to the large numbers of registers in them.

An LFSR with a feedback coefficient vector $(p_{m-1}, \ldots, p_1, p_0)$ is represented by the polynomial : $P(x) = x^m + p_{m-1}x^{m-1} + \ldots + p_1 x + p_0$

It is possible to deduce the LFSR information from the given polynomial using the following formula: $P(x) = x^m + p_{m-1}x^{m-1} + \ldots + p_1 x + 1$

- **m:** is the number of flip-flops (fixed)
- **x exponent (i.e. (m-1), (m-2) …):** are the flip-flops with feedback to the input.
- **1:** is a constant because FF0 always has feedback.

Linear Feedback Shift Registers (LFSR)

Example:

Design the following LFSR $\boldsymbol{P(x) = x^4 + x^1 + 1}$ using entail state (1000).

Solution:

$\boldsymbol{P(x) = x^4 + x^1 + 1 = x^4 + 0x^3 + 0x^2 + 1x^1 + 1x^0}$

Number of FF=m=4.

$p_3 = 0, p_2 = 0 \; and \; p_1 = 1$

FF3 has NO Feedback, FF2 has NO Feedback, FF1 has Feedback and FF0 always has Feedback

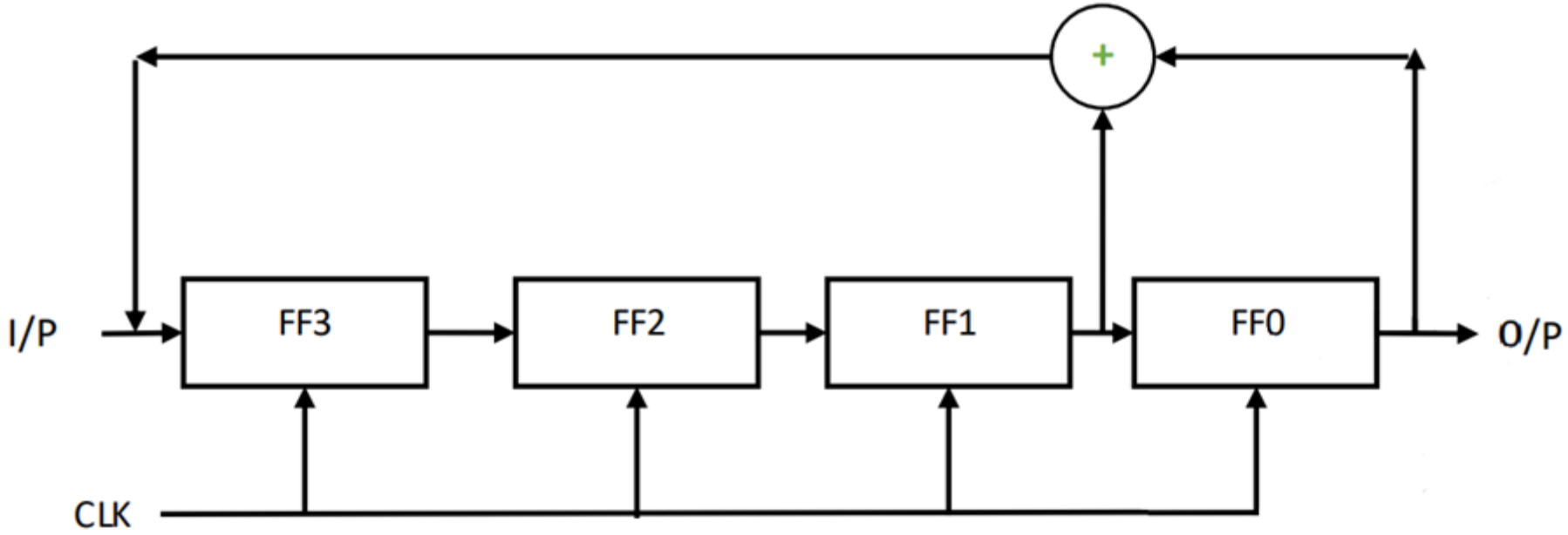
maximum sequence length = $\boldsymbol{2^m - 1 = 2^4 - 1}$ = 15

actual sequence length= 15.

Linear Feedback Shift Registers (LFSR)

Example:

Design the following LFSR $P(x) = x^4 + x^1 + 1$ using entail state (1000).



Linear Feedback Shift Registers (LFSR)

I / P	FF3	FF2	FF1 ⊕ FF0		O / P
NON	1	0	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
1	1	0	0	1	1
1	1	1	0	0	0
0	0	1	1	0	0
1	1	0	1	1	1
0	0	1	0	1	1
1	1	0	1	0	0
1	1	1	0	1	1
1	1	1	1	0	0
1	1	1	1	1	1
0	0	1	1	1	1
0	0	0	1	1	1
0	0	0	0	1	1
STOP	1	0	0	0	STOP
	Repeated Results				

Linear Feedback Shift Registers (LFSR)

It is possible to express the LFSR polynomials using brackets expression, such as:

$x^4 + x^1 + 1$ = (4,1,0)

$x^5 + x^2 + 1$ = (5,2,0)

$x^5 + x^3 + x^1 + 1$ = (5,3,1,0)

The numbers in the brackets represent the exponent of x in the LFSR polynomial.

Linear Feedback Shift Registers (LFSR)

Example:

Design LFSR using the following information, (5,3,2,0) and initial state (10000).

Solution:

Number of FF=m=5.

$\boldsymbol{P(x) = x^5 + x^3 + x^2 + 1}$

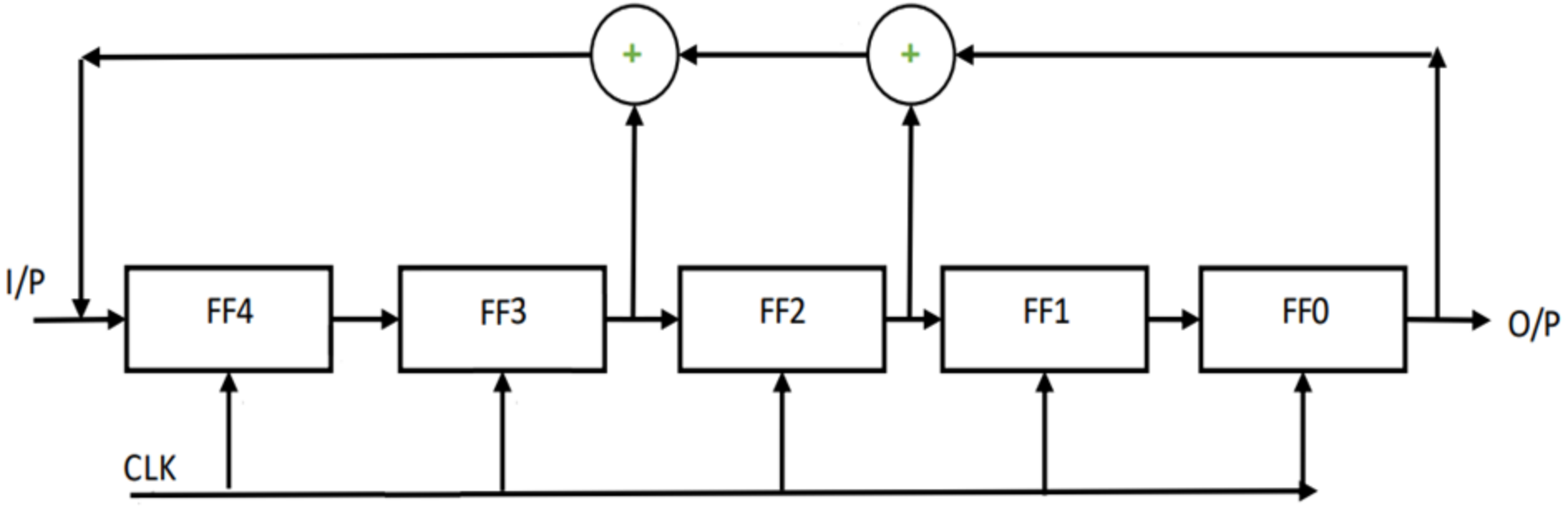
maximum sequence length = $\boldsymbol{2^m - 1 = 2^5 - 1}$ = 31

actual sequence length= 12.

Linear Feedback Shift Registers (LFSR)

Example:

Design LFSR using the following information, (5,3,2,0) and initial state (10000).



Linear Feedback Shift Registers (LFSR)

I / P	FF4	RES = FF3 ⊕ FF2		FF1	RES ⊕ FF0	O / P
NON	1	0	0	0	0	0
0	0	1	0	0	0	0
1	1	0	1	0	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	1
1	1	1	1	1	0	0
0	0	1	1	1	1	1
1	1	0	1	1	1	1
0	0	1	0	1	1	1
0	0	0	1	0	1	1
0	0	0	0	1	0	0
0	0	0	0	0	1	1
1	1	0	0	0	0	0

Cryptography and Network Security I

7th Lecture

Hash functions

By

Dr. Taqwa F. Hassan

Edited by

Dr. Ali Albu-Rghaif

Department of Computer Engineering

College of Engineering - University of Diyala

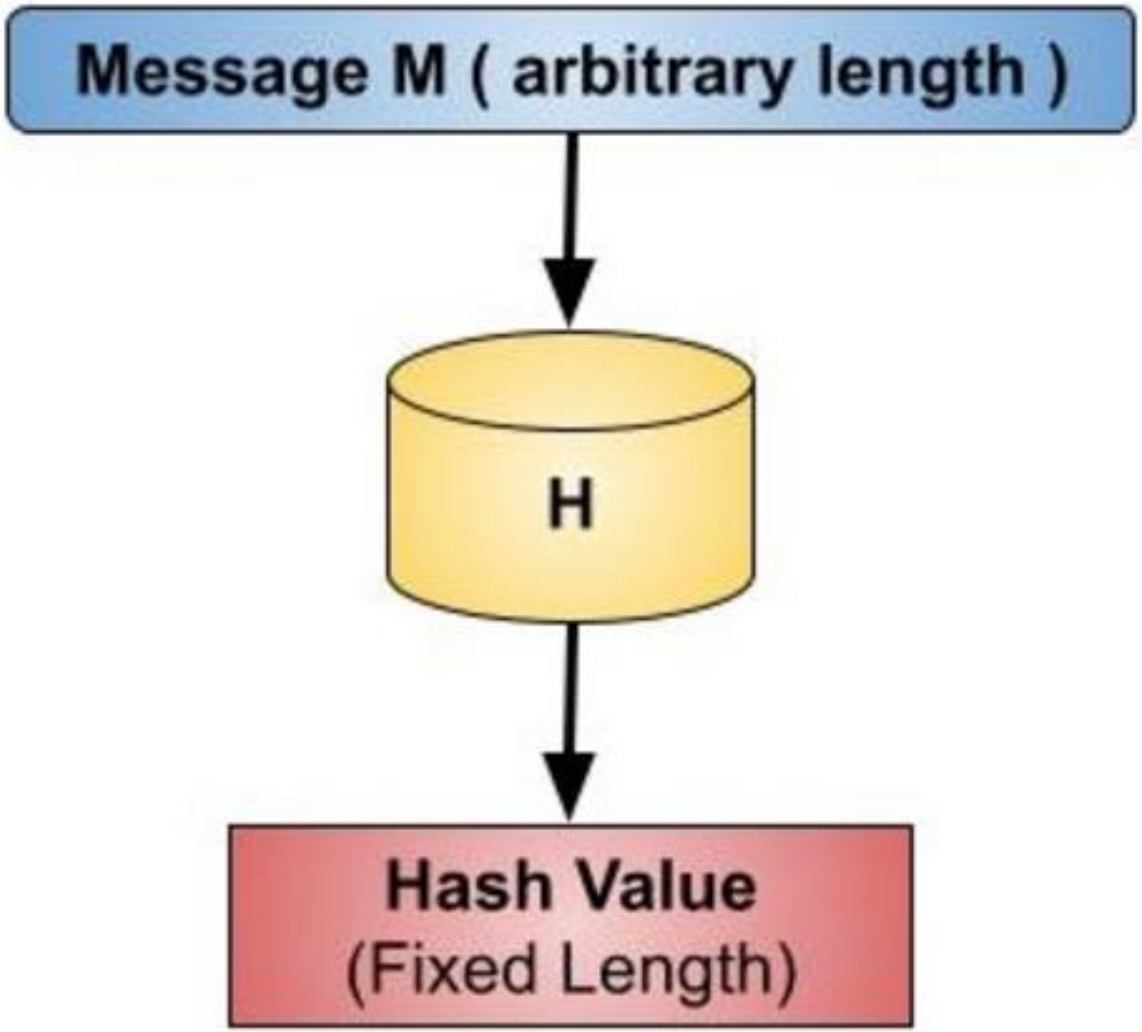
Hash functions

A hash function in cryptography is like a mathematical function that takes various inputs, like messages or data, and transforms them into fixed-length strings of characters. This means the input to the hash function is of any length but output is always of fixed length. This is like compressing a large balloon into a compact ball.

The importance of this process lies in its generation of a unique "fingerprint" for each input. Any minor alteration in the input results in a substantially different fingerprint, a quality known as "collision resistance."

Hash functions play a crucial role in various security applications, including password storage (hash values instead of passwords), digital signatures, and data integrity checks. Hash values, or message digests, are values that a hash function returns.

Hash functions



Key Points of Hash Functions

- Hash functions are mathematical operations that "map" or change a given collection of data into a fixed-length bit string that is referred to as the "hash value."
- Hash functions have a variety of complexity and difficulty levels and are used in cryptography.
- Cryptocurrency, password security, and communication security all use hash functions.

Operation of Cryptographic Hash Functions

In computing systems, hash functions are frequently used data structures for tasks like information authentication and message integrity checks. They are not easily decipherable, but because they can be solved in polynomial time, they are regarded as cryptographically "weak".

Typical hash functions have been improved with security characteristics by cryptographic hash functions, which make it more challenging to decipher message contents or recipient and sender information.

Operation of Cryptographic Hash Functions

Cryptographic hash functions display the following three characteristics:

- The hash function is called "collision-free." As a result, no two input hashes should be equal to the same output hash.
- They are hidden. A hash function's output should make it difficult to figure out the input value from it.
- They should be friendly to puzzles. The selection of an input that generates a predetermined result needs to be difficult. As such, the input needs to be taken from as wide as possible.

Properties of hash functions

The hash function should have the following properties:

1) Pre-Image Resistance

- Reversing a hash function should be computationally difficult.
- If a hash function h generates a hash value z, it should be difficult to identify an input value x that hashes to z.
- This feature defends against an attacker attempting to locate the input with just the hash value.

2) Second Pre-Image Resistance

- This property says that given an input and its hash, it should be difficult to find another input with the same hash.
- In other words, it should be challenging to find another input value y such that h(y) equals h(x) if a hash function h for an input x returns the hash value h(x).
- This feature of the hash function protects against an attacker who wants to replace a new value for the original input value and hash but only holds the input value and its hash.

Properties of hash functions

3) Collision Resistance

- This feature says that it should be difficult to identify two different inputs of any length that produce the same hash. This characteristic is also known as a collision-free hash function.
- In other words, for a hash function h, it is difficult to identify two distinct inputs x and y such that h(x)=h(y).
- A hash function cannot be free of collisions because it is a compression function with a set hash length. The collision-free condition simply indicates that these collisions should be difficult to locate.
- This characteristic makes it very hard for an attacker to identify two input values that have the same hash.
- Furthermore, a hash function is second pre-image resistant if it is collision-resistant.

Properties of hash functions

4) Efficiency of Operation

- Computation of h(x) for any hash function h given input x can be an easy process.
- Hash functions are computationally considerably faster than symmetric encryption.

5) Fixed Output Size

- Hashing generates an output of a specific length, regardless of the input size, and helps to make an output of the same size from different input sizes**.**

6) Deterministic

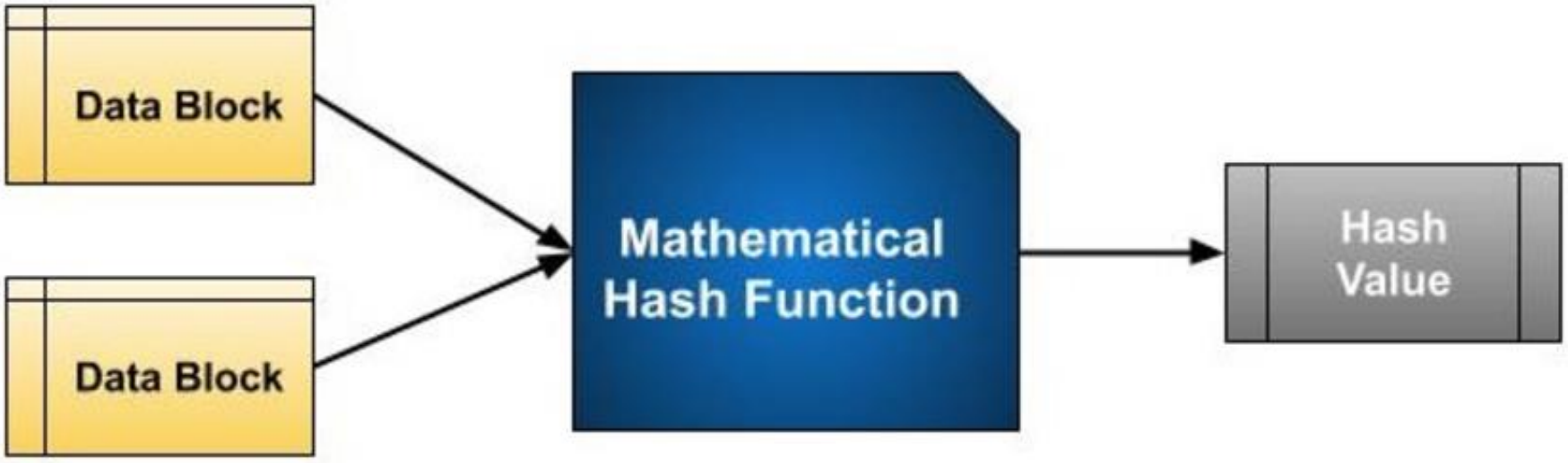
- For a given input, the hash function consistently produces the same output, like a recipe that always yields the same dish when followed precisely.

7) Fast Computation

- Hashing operations occur rapidly, even for large amounts of data sets.

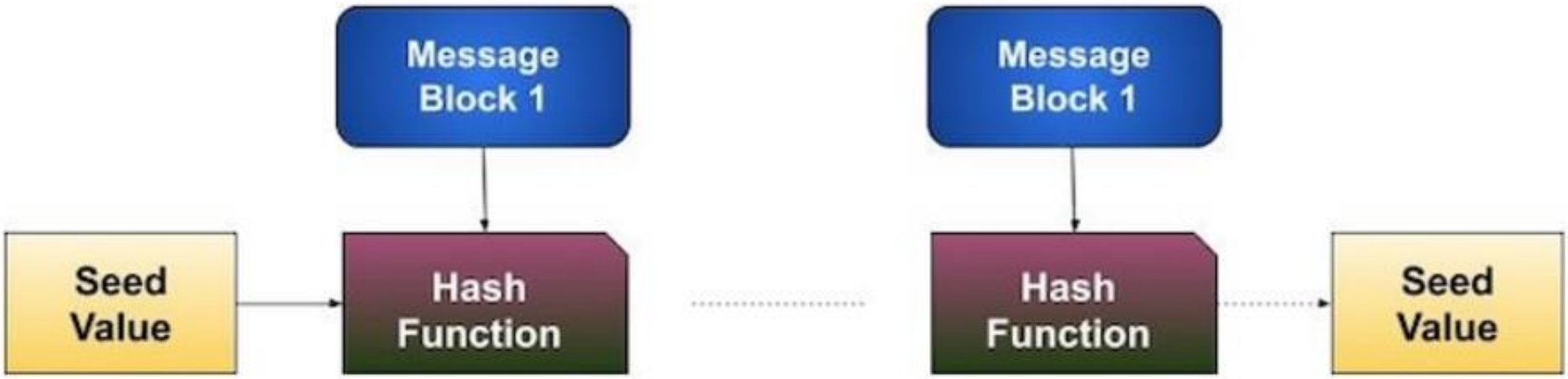
Design of Hashing Algorithms

Hashing essentially involves a mathematical function that takes two data blocks of fixed size and converts them into a hash code. The function is a key part of the hashing algorithm. The length of these data blocks differ according to the algorithm used. Usually, they range from 128 bits to 512 bits.



Design of Hashing Algorithms

Hashing algorithms use a sequence of rounds, similar to a block cipher, to process a message. In each round, a fixed-size input is used, which usually combines the current message block and the result from the previous round. This process continues for multiple rounds until the entire message is hashed.



Popular Hash Functions

1) Message Digest (MD)

- The hash functions MD2, MD4, MD5, and MD6 are members of the MD family. It is a 128-bit hash function.
- In 2004, collisions were found in MD5. It was claimed that an analytical attack using a computer cluster was successful in under one hour. Since MD5 was compromised by this collision attack, using it is no longer recommended.

2) Secure Hash Function (SHA)

- The four SHA algorithms which make up the SHA family are SHA-0, SHA-1, SHA-2, and SHA-3. Despite coming from the same family, the structure of it differs.
- In 2005, a technique was discovered for SHA-1 collision detection that can be used in a realistic time frame. So it is doubtful of SHA-1's long-term usability.
- SHA-224, SHA-256, SHA-384, and SHA-512 are the other four SHA variants in the SHA-2 family. The SHA-2 hash function has not yet been the target of any effective attacks

Popular Hash Functions

3) CityHash

- CityHash is another non-cryptographic hash function that is designed for fast hashing of large amounts of data. It is optimized for modern processors and offers good performance on both 32-bit and 64-bit architectures.

4) BLAKE2

- BLAKE2 is a fast and secure hash function that improves upon SHA-3. It is widely used in applications like cryptocurrency mining that need fast hashing. There are two types of BLAKE2:
  - **BLAKE2b –** Best for 64-bit computers, it produces hash values up to 512 bits long**.**
  - **BLAKE2s –** Best for smaller computers (8-32 bits), it produces hash values up to 256 bits long.

Standard Length

Input Message	Hash Function	Output (Hash Value)
CFI	MD5 (128-bit, 16-byte) 32 characters	3A10 0B15 B943 0B17 11F2 E38F 0593 9A9A
CFI	SHA-1 (160-bit, 20-byte) 40 characters	569D C9F0 7B48 7F58 9241 AD4C 5C28 7DA0 A448 8D08
CFI	SHA-256 (256-bit, 32-byte) 64 characters	F3ED 0867 48FF 3641 3091 0BB6 6293 7080 2958 B5A2 52AF F364 1FC5 07FD E80D 9929

Standard Length

Input Message	Hash Function	Output (Hash Value)
CFI	SHA-1	569D C9F0 7B48 7F58 9241 AD4C 5C28 7DA0 A448 8D08
Corporate FI	SHA-1	82C0 5EDC 608F AA08 8EE0 BDD8 8E22 3B38 CA38 82CC
CF Input	SHA-1	2013 85FC EEE4 F73D 07F0 4F2A A4CB BOE9 12BF BBB8
CFI 1	SHA-1	C501 23CE 8BB2 A42D 5BB4 4DA7 3FC2 3B3D 62F5 14A5

Applications of Hash Functions

Password Storage

Hash functions protect password storage. Instead of storing passwords in clear, mostly all login processes store the hash values of passwords in the file.

The Password file is a table of pairs in the format (user id, h(P)).

Even if an attacker has access to the password, all they can see is the hashes of the passwords. Because the hash function contains the pre-image resistance feature, he cannot use it to log in or get the password from it.

Data Integrity

A hash function ensures that the data has not been altered. If even one character in the input changes, the hash value will change drastically. This is crucial for verifying the integrity of files during transmission (e.g., downloading software or files).

Applications of Hash Functions

Password Storage

Hash functions protect password storage. Instead of storing passwords in clear, mostly all login processes store the hash values of passwords in the file.

The Password file is a table of pairs in the format (user id, h(P)).

Even if an attacker has access to the password, all they can see is the hashes of the passwords. Because the hash function contains the pre-image resistance feature, he cannot use it to log in or get the password from it.

Data Integrity

A hash function ensures that the data has not been altered. If even one character in the input changes, the hash value will change drastically. This is crucial for verifying the integrity of files during transmission (e.g., downloading software or files).

Applications of Hash Functions

Digital Signatures

In the context of digital signatures, hash functions are used to create a fixed-size representation of a message. The message is hashed first, and then the hash is signed. This allows for efficient verification of the signature.

Unique Identifiers

In systems like databases or blockchain, hash functions are used to create unique identifiers for data entries or transactions.

Examples of Hash Functions

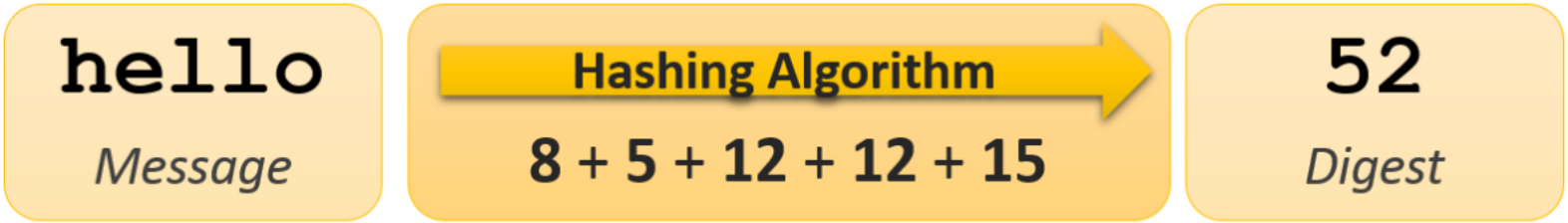
Hashing the Word "hello“ based SHA-256

2cf24dba5fb0a30e26e83b2ac5b9e29e1b169e1b4ba2db9c8f06fd9b75c6c3ed

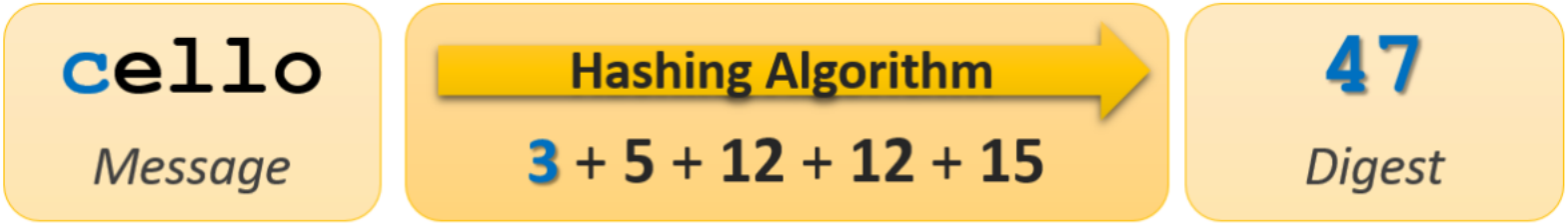
Hashing the Word "hEllo“ based SHA-256

b94d27b9934d3e08a52e52d7da7dabfad8f5a3c9db7d00f74f4a08624dbbb05b

Hashing the Word "hello“ based MD



Hashing the Word “cello“ based MD



Cryptography and Network Security I

8th Lecture

Digital Signature

By

Dr. Taqwa F. Hassan

Edited by

Dr. Ali Albu-Rghaif

Department of Computer Engineering

College of Engineering - University of Diyala

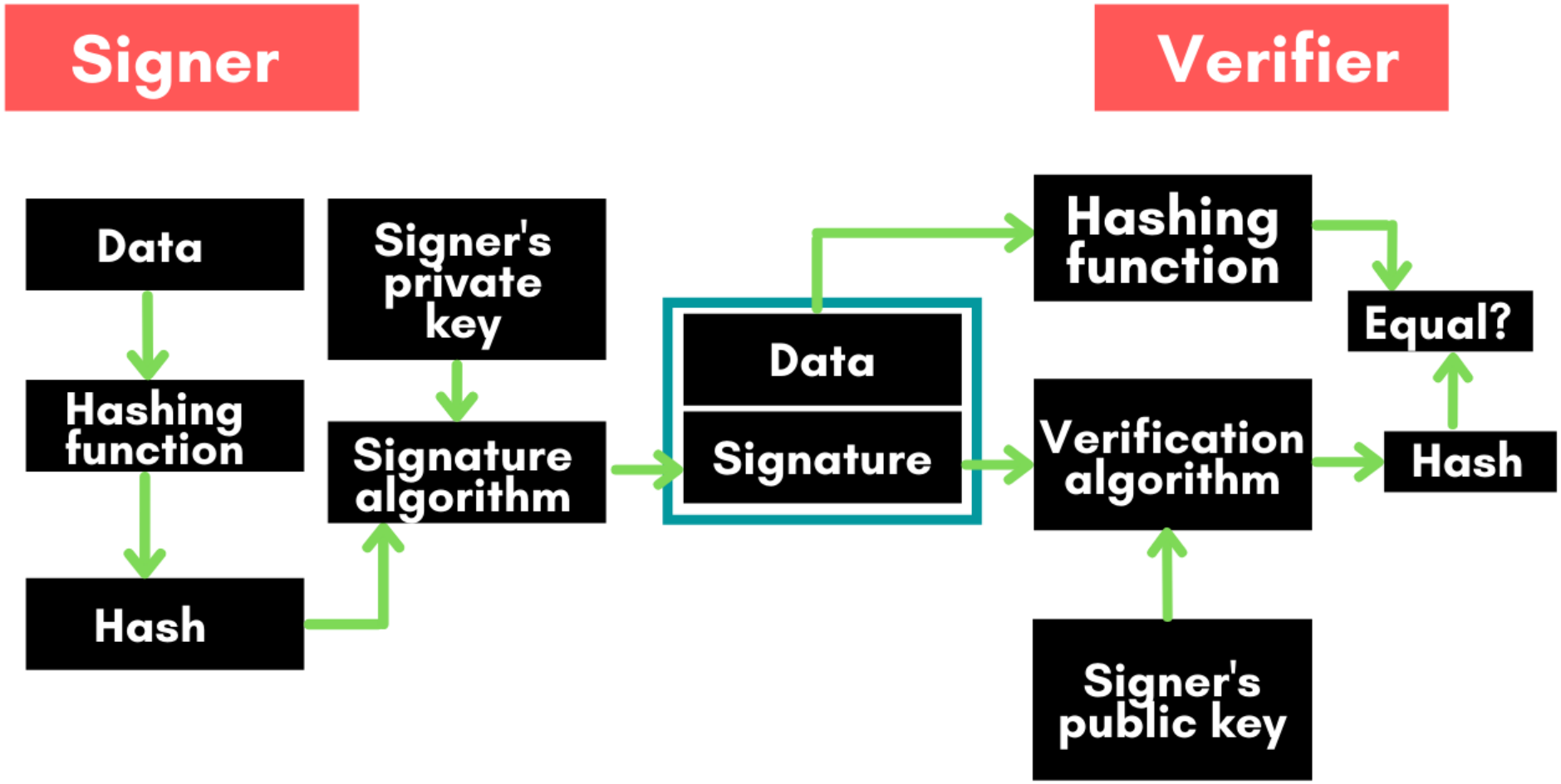
Digital Signature

Digital signatures are the public-key primitives of message authentication. In the physical world, it is common to use handwritten signatures on handwritten or typed messages. They are used to bind the signatory to the message.

Similarly, a digital signature is a technique that binds a person/entity to the digital data. This binding can be independently verified by the receiver as well as any third party.

The digital signature is a cryptographic value that is calculated from the data and a secret key known only by the signer.

Model of Digital Signature



Model of Digital Signature

- Each person adopting this scheme has a public-private key pair.
- Generally, the key pairs used for encryption/decryption and signing/verifying are different. The private key used for signing is referred to as the signature key and the public key as the verification key.
- The signer feeds data to the hash function and generates a hash of data.
- Hash value and signature key are then fed to the signature algorithm which produces the digital signature on a given hash. Signature is appended to the data and then both are sent to the verifier.
- The verifier feeds the digital signature and the verification key into the verification algorithm. The verification algorithm gives some value as output.
- The verifier also runs the same hash function on received data to generate a hash value.
- For verification, this hash value and the output of the verification algorithm are compared. Based on the comparison result, the verifier decides whether the digital signature is valid.
- Since the digital signature is created by the ‘private’ key of the signer and no one else can have this key; the signer cannot repudiate signing the data in the future.

Encryption with Digital Signature

In many digital communications, it is desirable to exchange encrypted messages than plaintext to achieve confidentiality. In a public key encryption scheme, a public (encryption) key of the sender is available in the open domain, and hence anyone can spoof his identity and send an encrypted message to the receiver.

This makes it essential for users employing PKC for encryption to seek digital signatures along with encrypted data to be assured of message authentication and non-repudiation.

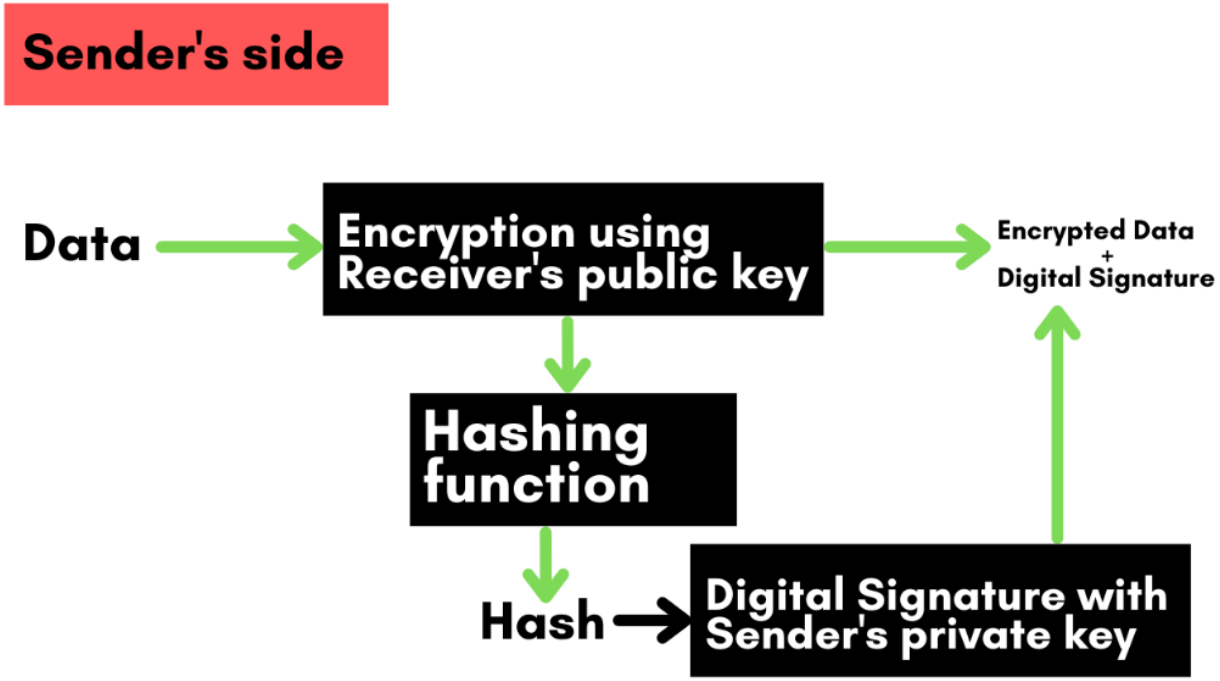
This can archived by combining digital signatures with an encryption scheme. Let us briefly discuss how to achieve this requirement. There are two possibilities, **sign-then-encrypt** and **encrypt-then-sign**.

Encryption with Digital Signature

However, the cryptosystem based on sign-then-encrypt can be exploited by the receiver to spoof the identity of the sender and send that data to a third party. Hence, this method is not preferred. The process of encrypt-then-sign is more reliable and widely adopted.

The receiver after receiving the encrypted data and signature on it,

- **first verifies the signature using the sender's public key.**
- **After ensuring the validity of the signature, he then retrieves the data through decryption using his private key.**



Importance of Digital Signature

- **Message Authentication**
  - When the verifier validates the digital signature using the public key of a sender, he is assured that the signature has been created only by a sender who possesses the corresponding secret private key and no one else.
- **Data Integrity**
  - In case an attacker has access to the data and modifies it, the digital signature verification at the receiver end fails. The hash of modified data and the output provided by the verification algorithm will not match. Hence, a receiver can safely deny the message assuming that data integrity has been breached.
- **Non-Repudiation**
  - Since it is assumed that only the signer knows the signature key, he can only create a unique signature on a given data. Thus the receiver can present data and the digital signature to a third party as evidence if any dispute arises in the future.

Digital Signature Applications

- **Email**
  - To sign and verify the authenticity of email messages.
- **Software Distribution**
  - To ensure the integrity and authenticity of software updates or downloads.
- **Contracts and Legal Documents**
  - Many businesses and governments use digital signatures for legally binding contracts and other official documentation.
- **Blockchain and Cryptocurrencies**
  - In blockchain networks like Bitcoin, digital signatures are used to authenticate transactions.
- **Government Services**
  - Many governments require digital signatures for tax filings, application submissions, etc.

Limitations of Digital Signatures

- **Key Management**
  - Managing public and private keys is essential for the security of digital signatures. If a private key is compromised, an attacker could forge signatures.
- **Trust Models**
  - The security of a digital signature depends on the trust in the public key infrastructure (PKI) and the certification authorities (CAs) that validate the identity of users. If a CA is compromised or issues a fraudulent certificate, the system can be broken.
- **Legal Recognition**
  - In many jurisdictions, digital signatures must meet specific legal and regulatory standards to be considered valid for legal contracts. The exact requirements can vary.

Digital Signatures vs. Electronic Signatures

- It’s important to note that **Digital Signatures** are not the same as **Electronic Signatures**. An electronic signature can be as simple as typing your name at the end of an email or clicking "**I agree**" on an online form. These are not necessarily secure and can be easily replicated.

- In contrast, a **Digital Signature** is a cryptographically secure mechanism that provides a higher level of assurance about the identity of the signer and the integrity of the document.

Elgamal Digital Signatures

- **Key Generation**
  - Select a large prime number p
  - Choose a primitive root g modulo p
  - Select a private key x such that $1 \leq x \leq p-2$
  - Compute the public key y as: $y = g^x \bmod p$
  - Public key: (p, g, y)
  - Private key: x

- **Message**
  - Let M be the message to sign.

Elgamal Digital Signatures

- **Signing the Message**
  - Choose a random integer k such that $1 \leq k \leq p-2$ and $\gcd(k, p-1) = 1$
  - Compute **r** as: $r = g^k \bmod p$
  - Compute the signature component s using the equation:

$$s = (H(M) - x \,.\, r) \,.\, k^{-1} \bmod (p-1)$$

    Where $H(M)$ is the hash of the message and k^{-1} is the modular inverse of $k \bmod (p-1)$
  - The signature is the pair (r,s)
- **Verifying the Signature**
  - Compute $v1$ as: $v1 = y^r . r^s \bmod p$
  - Compute $v2$ as: $v2 = g^{H(M)} \bmod p$
  - The signature is valid if and only if: $v1 = v2$

Elgamal Digital Signatures

Example

- **Key Generation**
  - p = 467, g = 2, x = 123
  - Compute $y = g^x \bmod p = 2^{123} \bmod 467 = 194$
  - Public key: **(p,g,y) ➔ (467,2,194)**
  - Private key: **x =123**
- **Message**
  - Message M="$Hello$“, Assume $H(M)$=123.
- **Signing the Message**
  - Choose a random integer **k = 57** $(1 \leq k \leq p-2)$
  - Compute $r = g^k \bmod p = 2^{57} \bmod 467 = 318$
  - Compute $k^{-1} \bmod (p-1) = 57^{-1} \bmod (467-1) = 183$
  - Compute $s = (H(M) - x \,.\, r) \,.\, k^{-1} \bmod (p-1) = (123 - 123 \,.\, 318) \,.\, 183 = 365$
  - The signature **(r,s) ➔ (318, 365)**

Elgamal Digital Signatures

Example

- **Verifying the Signature**
  - Compute $v1$ as: $\boldsymbol{v1 = y^r . r^s \bmod p = 194^{318} . 318^{365} \bmod 467 = 232}$
  - Compute $v2$ as: $\boldsymbol{v2 = g^{H(M)} \bmod p = 2^{123} \bmod 467 = 232}$
  - Since $\boldsymbol{v1 = v2}$, the signature is valid.

RSA Digital Signature

Example

- **Key Generation**
  - **Choose two large prime numbers, p and q**

    p = 61, q = 53
  - Compute $n = p \times q$ (the modulus)

    $n = p \times q$ = 61 × 53 = 3233
  - Compute $\phi(n) = (p-1) \times (q-1)$

    $\phi(n) = (p-1) \times (q-1)$ = 60 × 52 = 3120.
  - Select the public exponent $e \in \{1,2,\ldots, \Phi(n)-1\}$ such that GCD(e,$\Phi(n)$) = 1.

    Let e = 17
  - Compute the private key d such that $d = e^{(\Phi(n)-1)} \ mod \ \Phi(n)$

    $d = 17^{3119}$ mod 3120 = 2753
  - **Public Key: (e,n) = (17,3233)**
  - **Private Key: (d,n) = (2753,3233)**

RSA Digital Signature

Example

- **Signing the Message**
  - **Take the message M**

    M=65 (a numeric representation of the message, or a hash of the message)
  - Compute the signature S using the private key:

    $$S = M^d \bmod n = 65^{2753} \bmod 3233$$

    S = 2790
- **Verifying the Signature**
  - Use the signature S, the public key (e,n), and the original message M
  - Compute:

    $$M' = S^e \bmod n.$$

    $$M' = 2790^{17} \bmod 3233.$$

    $$M' = 65$$
  - Check if $M'=M$
  - **Since M′ = 65, the signature is valid.**

Cryptography and Network Security II

4th year

Lecture No. 1

Transport-Layer Security

DR. Lecturer . Taqwa.F.Hassan

Computer Department - College of Engineering
University of Diyala

2023-2024

The **web browser** is an application program that displays a www document. It usually uses other internet services to access the document. **Web server** is a program or a computer that can provide services to other programs called clients. The main difference between a Web browser and a Web server is that Web browser requests for the document and services, and act as an interface between a client and a server which displays the web content. On the other hand, Webserver accepts, approve, and response to the request made by a web browser for a web document.

A **Web browser** can be considered as a utility which client uses to access webservices and documents from the server. There are various types of browsers are in use, for example, the default browser for windows platform is internet explorer , similarly for apple device default browser is Safari. Although, there are other browsers also like Google Chrome, Mozilla Firefox, opera and UC.

Web Server is a piece of software running on a computer whose primary job is to distribute web pages to users whenever they demand it and provides an area in which to store and organize the pages of the website. The machine that executes the web server software can be a remote machine placed at the other side of your network or even on the other end of the globe, or it be your very own personal computer at home. We also introduced the idea that the user's browser was the client in this relationship.

Web Security Threats

Table 1 provides a summary of the types of security threats faced when using the Web. One way to group these threats is in terms of passive and active attacks.

Passive attacks include eavesdropping on network traffic between browser and server and gaining access to information on a Web site that is supposed to be restricted. Active attacks include impersonating another user, altering messages in transit between client and server, and altering information on a Web site. Another way to classify Web security threats is in terms of the location of the threat: Web server, Web browser, and network traffic between browser and server.

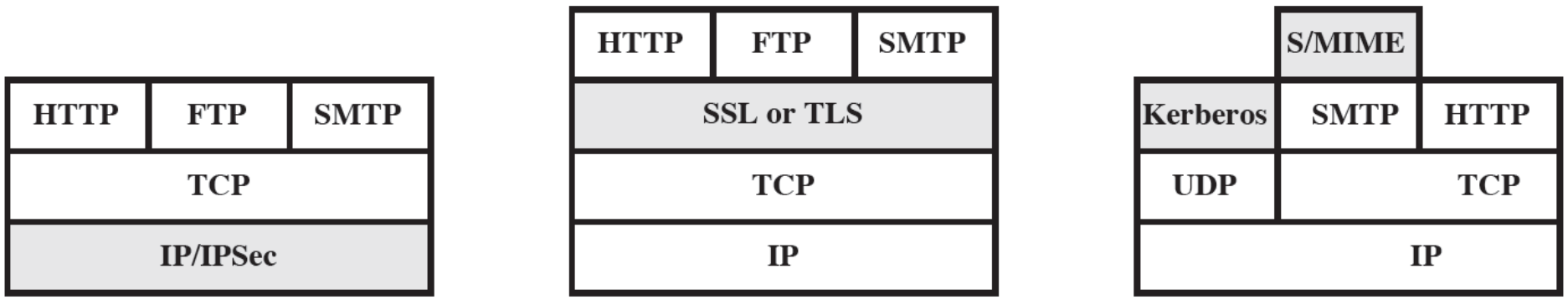
Table 1 A Comparison of Threats on the Web

	Threats	Consequences	Countermeasures
Integrity	• Modification of user data • Trojan horse browser • Modification of memory • Modification of message traffic in transit	• Loss of information • Compromise of machine • Vulnerability to all other threats	Cryptographic checksums
Confidentiality	• Eavesdropping on the net • Theft of info from server • Theft of data from client • Info about network configuration • Info about which client talks to server	• Loss of information • Loss of privacy	Encryption, Web proxies
Denial of Service	• Killing of user threads • Flooding machine with bogus requests • Filling up disk or memory • Isolating machine by DNS attacks	• Disruptive • Annoying • Prevent user from getting work done	Difficult to prevent
Authentication	• Impersonation of legitimate users • Data forgery	• Misrepresentation of user • Belief that false information is valid	Cryptographic techniques

Web Traffic Security Approaches

A number of approaches to providing Web security are possible. The various approaches that have been considered are similar in the services they provide and, to some extent, in the mechanisms that they use, but they differ with respect to their scope of applicability and their relative location within the TCP/IP protocol stack.

Where to provide security?



(a) Network Level (b) Transport Level (c) Application Level

Figure 1 Relative Location of Security Facilities in the TCP/IP Protocol Stack

Transport layer security

One of the most widely used security services is **Transport Layer Security (TLS)**; the current version is Version 1.2, defined in RFC 5246. TLS is an Internet standard that evolved from a commercial protocol known as **Secure Sockets Layer (SSL)**. TLS is a general purpose service implemented as a set of protocols that rely on TCP. At this level, there are two implementation choices. For full generality, TLS could be provided as part of the underlying protocol suite and therefore be transparent to applications. Alternatively, TLS can be embedded in specific packages. For example, most browsers come equipped with TLS, and most Web servers have implemented the protocol.

SSL History

- Netscape developed The Secure Sockets Layer Protocol (SSL) in 1994, as a response to the growing concern over security on the Internet.

- SSL was originally developed for securing web browser and server communications.
- SSL v3.0 was specified in an Internet Draft (1996)

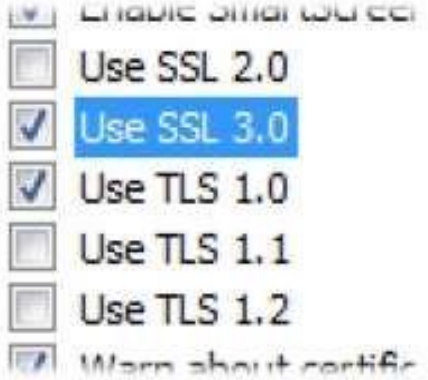
SSL (Secure Socket Layer)

- SSL is a Secure Sockets Layer
- SSL is the standard security technology for establishing an encrypted link between a web server and a browser.
- This link ensures that all data passed between the web server and browsers remain private and integral
- There are several versions of the SSL protocol defined. The latest version, the Transport Layer Security Protocol (TLS), is based on SSL 3.0

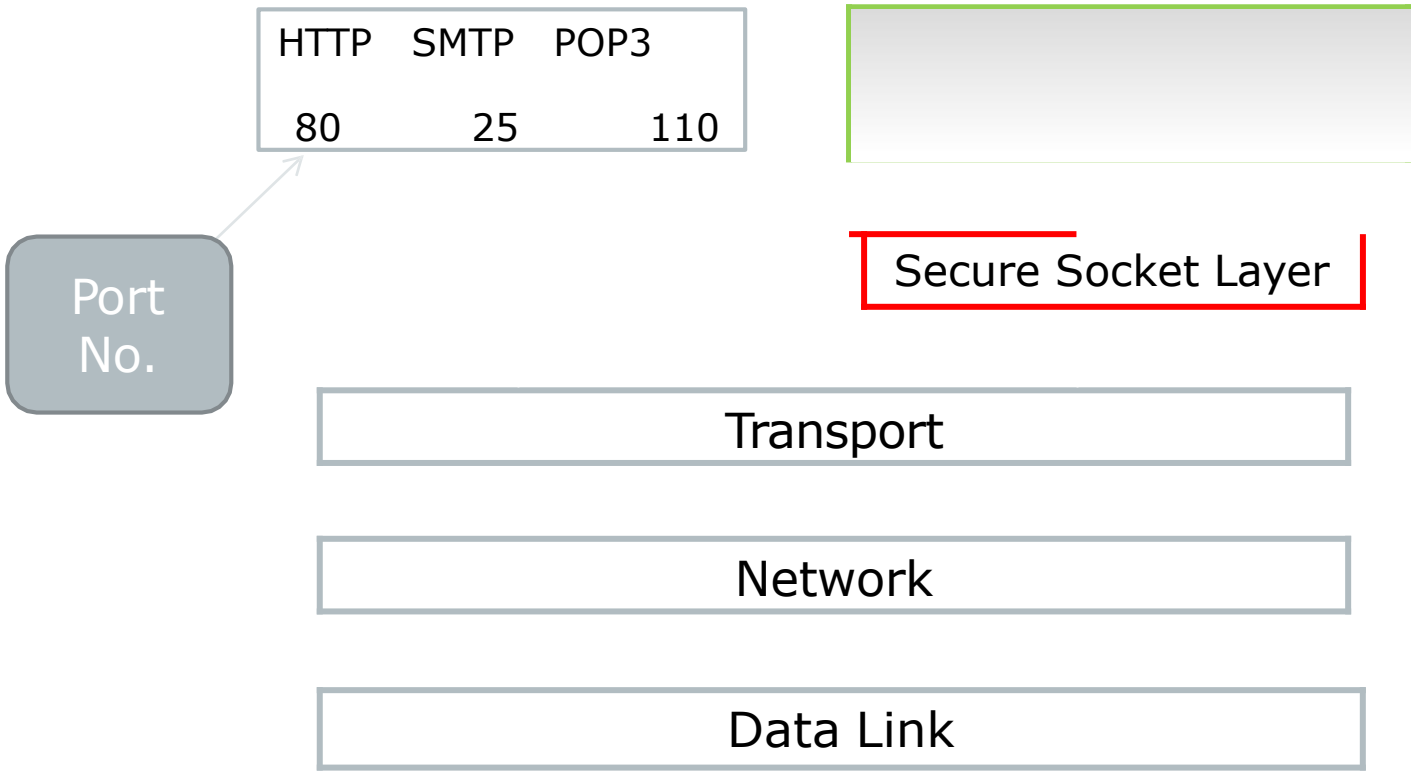
SSL Version 1.0

SSL Version 2.0

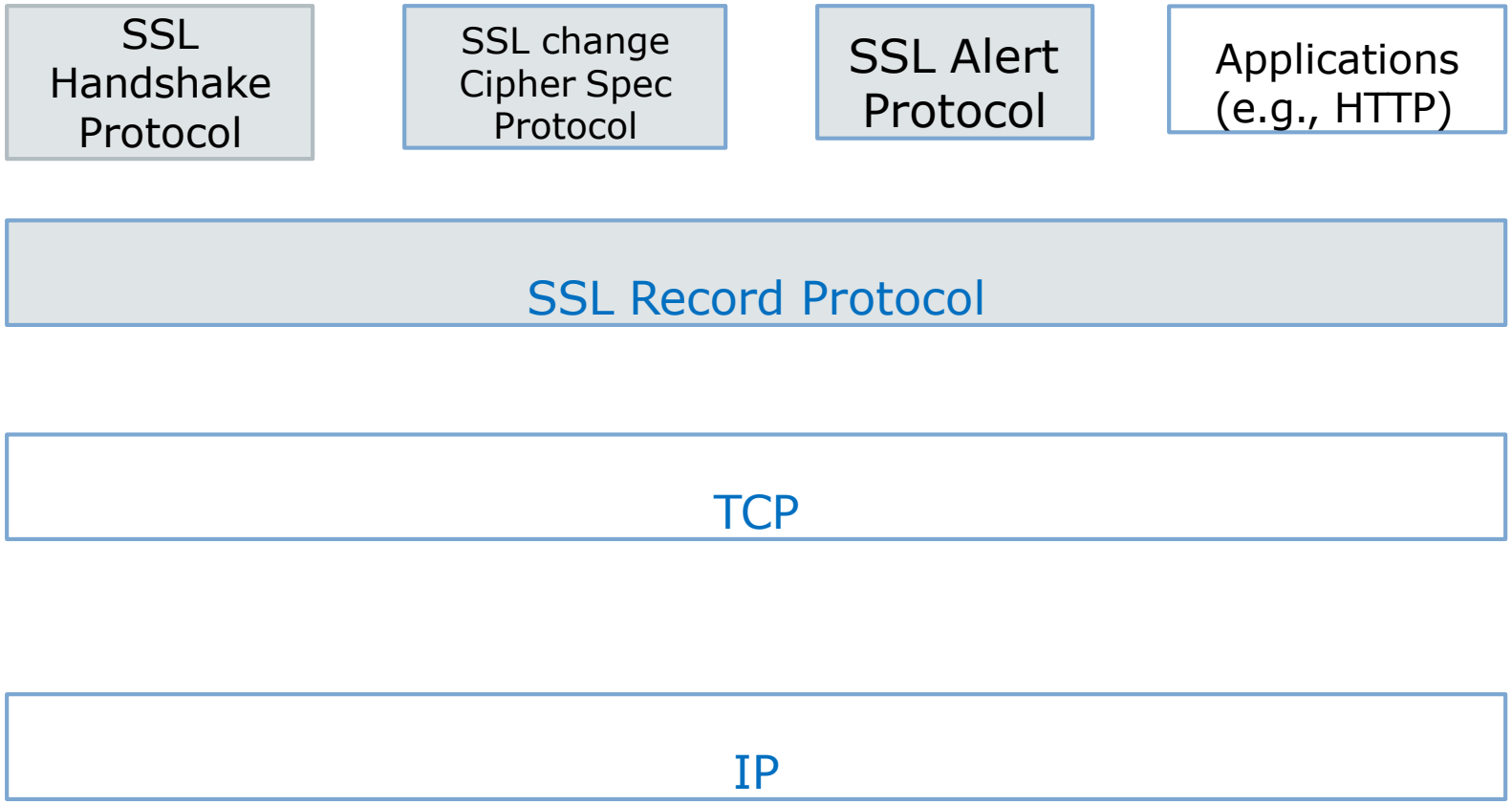
SSL Version 3.0



Where SSL fits?

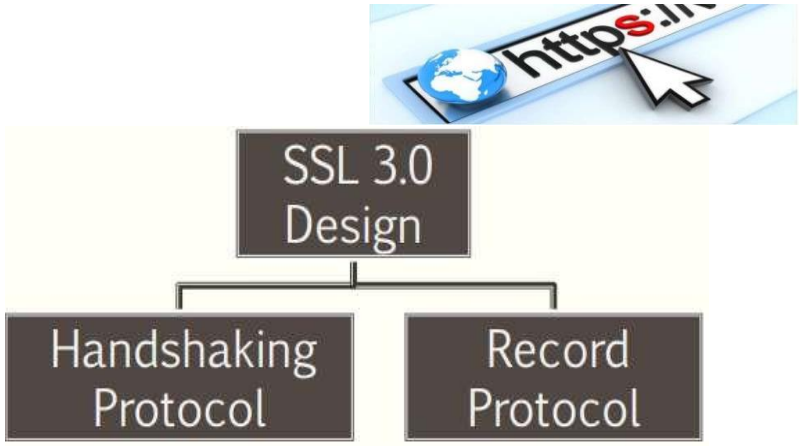


SSL architecture



SSL

- It is the most widely known as the protocol that, coupled with HTTP, secures the Web and uses the “https” URI scheme



SSL Goals

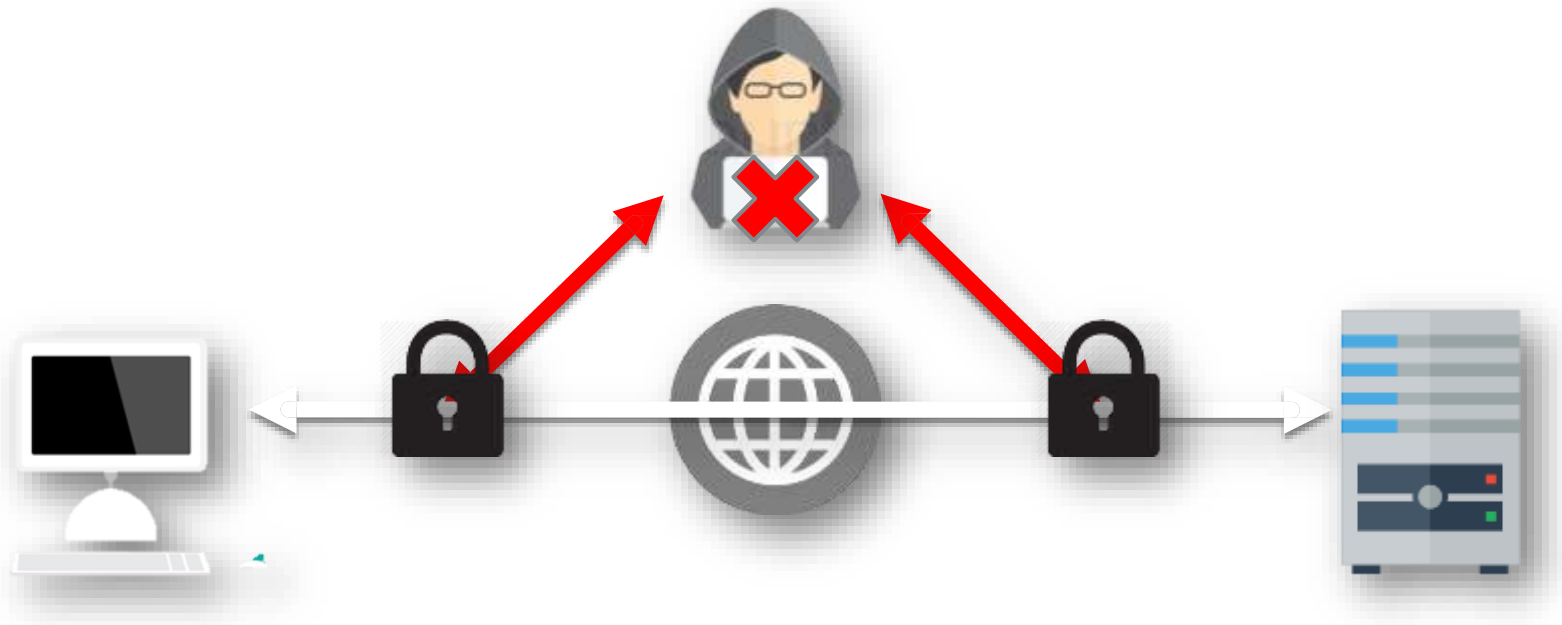
- **Confidentiality**
- **Integrity Protection**
- **Authentication**

Transport Layer Security (TLS)

- TLS is the successor to the Secure Sockets Layer (SSL).
- Transport Layer Security (**TLS**) is a protocol that ensures privacy between communicating applications and their users on the Internet.
- Is a widely deployed protocol for securing client-server communications over the internet.
- TLS is designed to prevent eavesdropping, tampering, and message forgery

Why do we need it?

- TLS ensures that no third party may eavesdrop or tamper with any message.

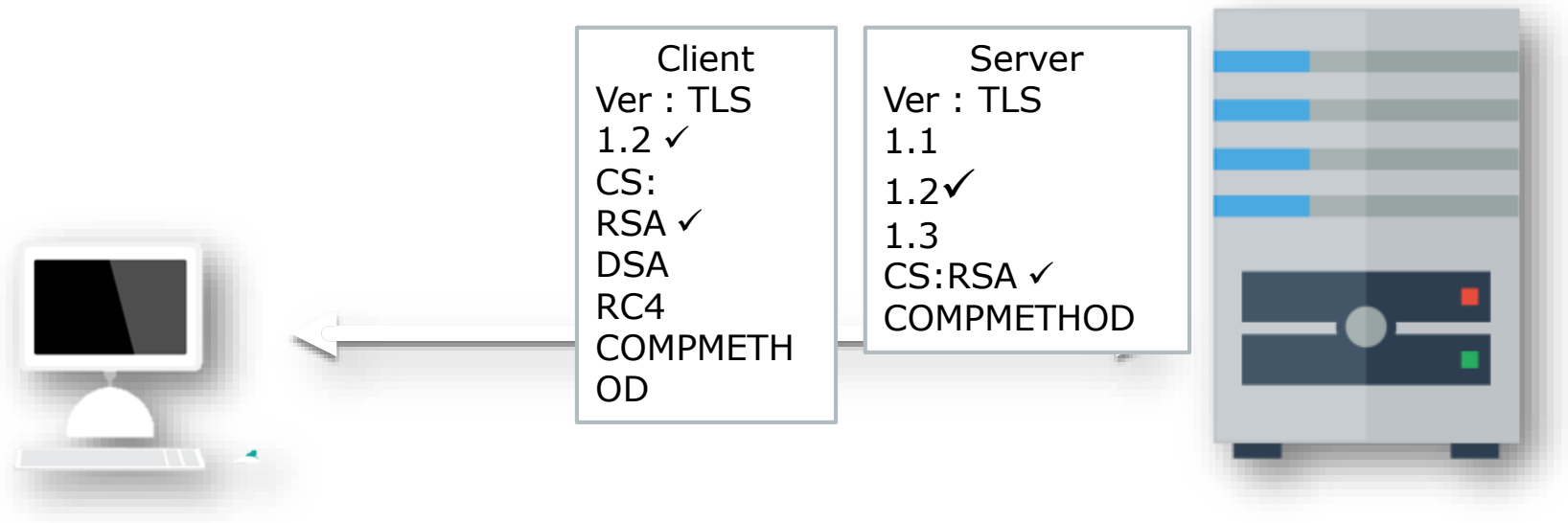


Working of Transport Layer Security

- The Client connect to server (using TCP). The client can be anything.
- The Client sends a number of specifications :
  - Version of SSL/TLS
  - Which cipher suites, compression method it wants to use.

Working of Transport Layer Security

- The server checks what the highest SSL/TLS version is that is supported by them both, picks a cipher suite from one of the client's options (if it supports one), and optionally picks a compression method.

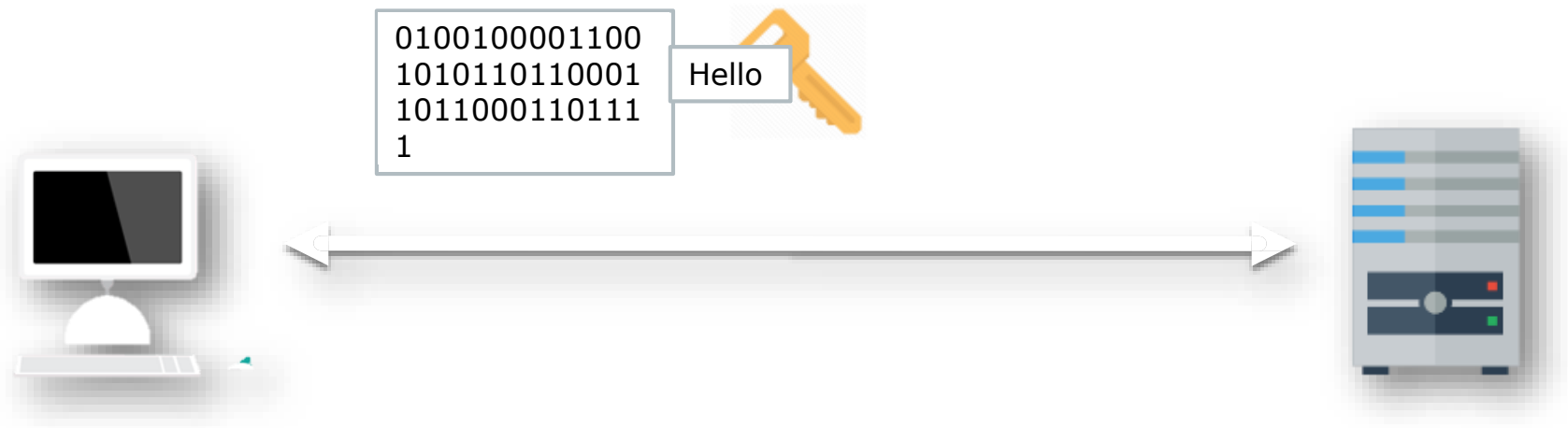


Working of Transport Layer Security

- After this the basic setup is done, the server sends its certificate.
- This certificate must be trusted by either the client itself or a party that the client trusts.

Working of Transport Layer Security

- Both the server and the client can now compute the key for the symmetric encryption.



Working of Transport Layer Security

- The handshake is now finished, and the two hosts can communicate securely.

Working of Transport Layer Security

- To close the connection, a close notify 'alert' is used. If an attacker tries to terminate the connection by finishing the TCP connection (injecting a FIN packet), both sides will know the connection was improperly terminated.
  The connection cannot be compromised by this though, merely interrupted

Benefits of TLS\SSL

- Encryption
- Interoperability
- Algorithm flexibility
- Ease of deployment
- Ease of use

Attacks on TLS/SSL

- Attacks on the handshake protocol
- Attacks on the record and application data protocols
- Attacks on the PKI
- Other attacks

POWERCERT VIDEOS

A lot of websites are now using HTTPS by default, regardless if sensitive data is going to be exchanged or not.

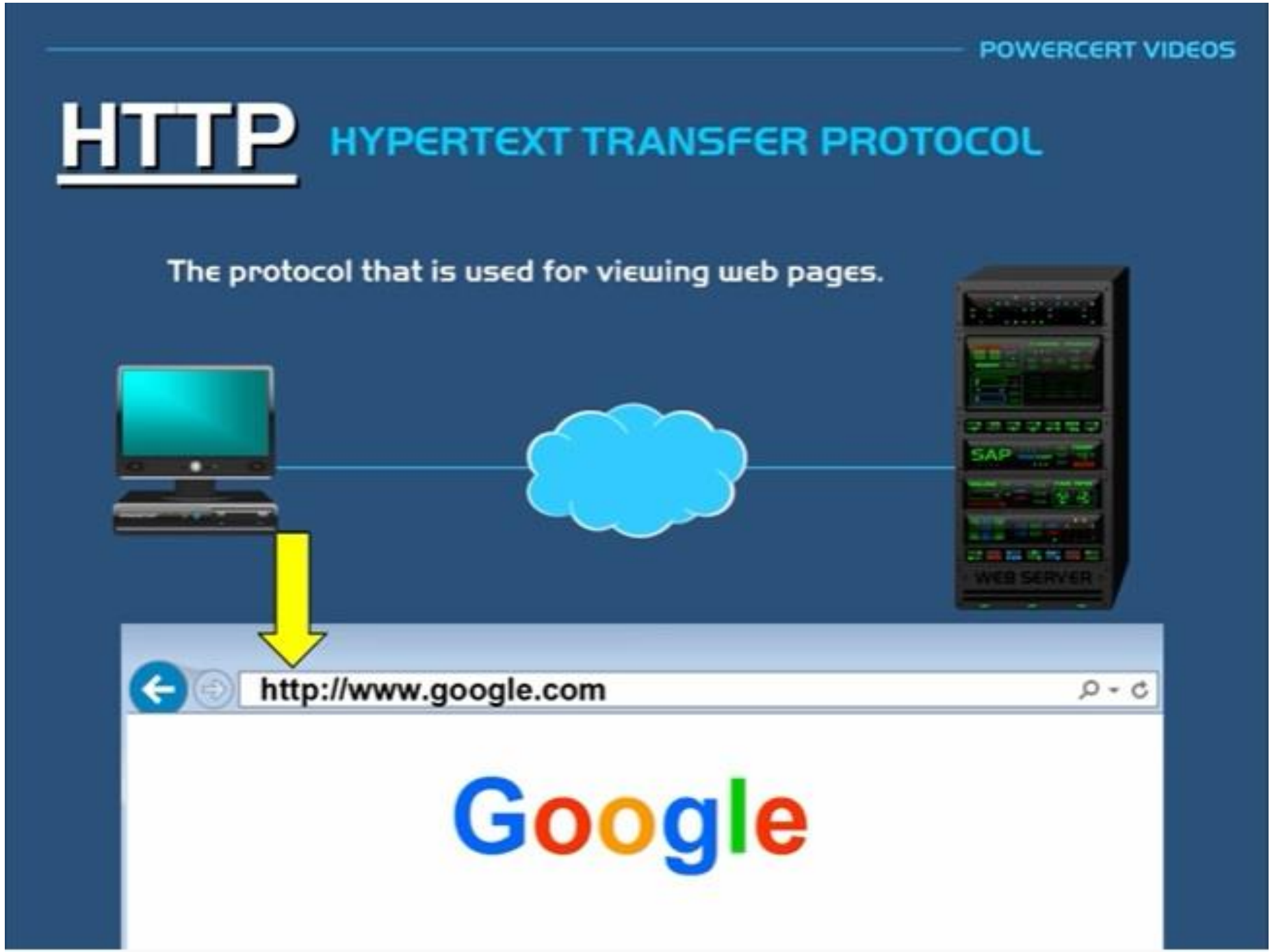
Google is flagging websites as "not secure" if they are not SSL protected.

Google is penalizing websites that are not SSL protected.

HTTP (Hypertext Transfer Protocol) is the base of the data communication for the web this is how the internet works when it comes to delivering the web pages. It is TCP/IP based protocol and things like text, audio, videos, images can be transmitted through it.

HTTP works on **request** and **response** cycle where the client requests a web page. Suppose, if you browse to google.com, you are requesting a web page from the server, and the server will deliver you response.

The main issue of HTTP is that it is **not encrypted** and **plain text** is used, meaning that it is unsecured at transferring data among the computer and server. It is popular to exploit the man-in-the-middle attacks, if you run a HTTP connection anyone can put himself in the middle and start using names, emails, passwords in the plain text.


POWERCERT VIDEOS
HTTP
HYPERTEXT TRANSFER PROTOCOL
The protocol that is used for viewing web pages.
SAP
WEB SERVER
http://www.google.com
Google

HTTPS (HTTP over SSL)

Refers to the combination of **HTTP** and **SSL** to implement secure communication between a Web browser and a Web server. The HTTPS capability is built into all modern Web browsers. Its use depends on the Web server supporting HTTPS communication. For example, some search engines do not support HTTPS.

The principal difference seen by a user of a Web browser is that URL (uniform resource locator) addresses begin with https:// rather than http://. A normal HTTP connection uses port 80. If HTTPS is specified, port 443 is used, which invokes SSL.

When HTTPS is used, the following elements of the communication are encrypted:

- URL of the requested document
- Contents of the document
- Contents of browser forms (filled in by browser user)
- Cookies sent from browser to server and from server to browser
- Contents of HTTP header

HTTPS is documented in RFC 2818, *HTTP Over TLS*. There is no fundamental change in using HTTP over either SSL or TLS, and both implementations are referred to as HTTPS.

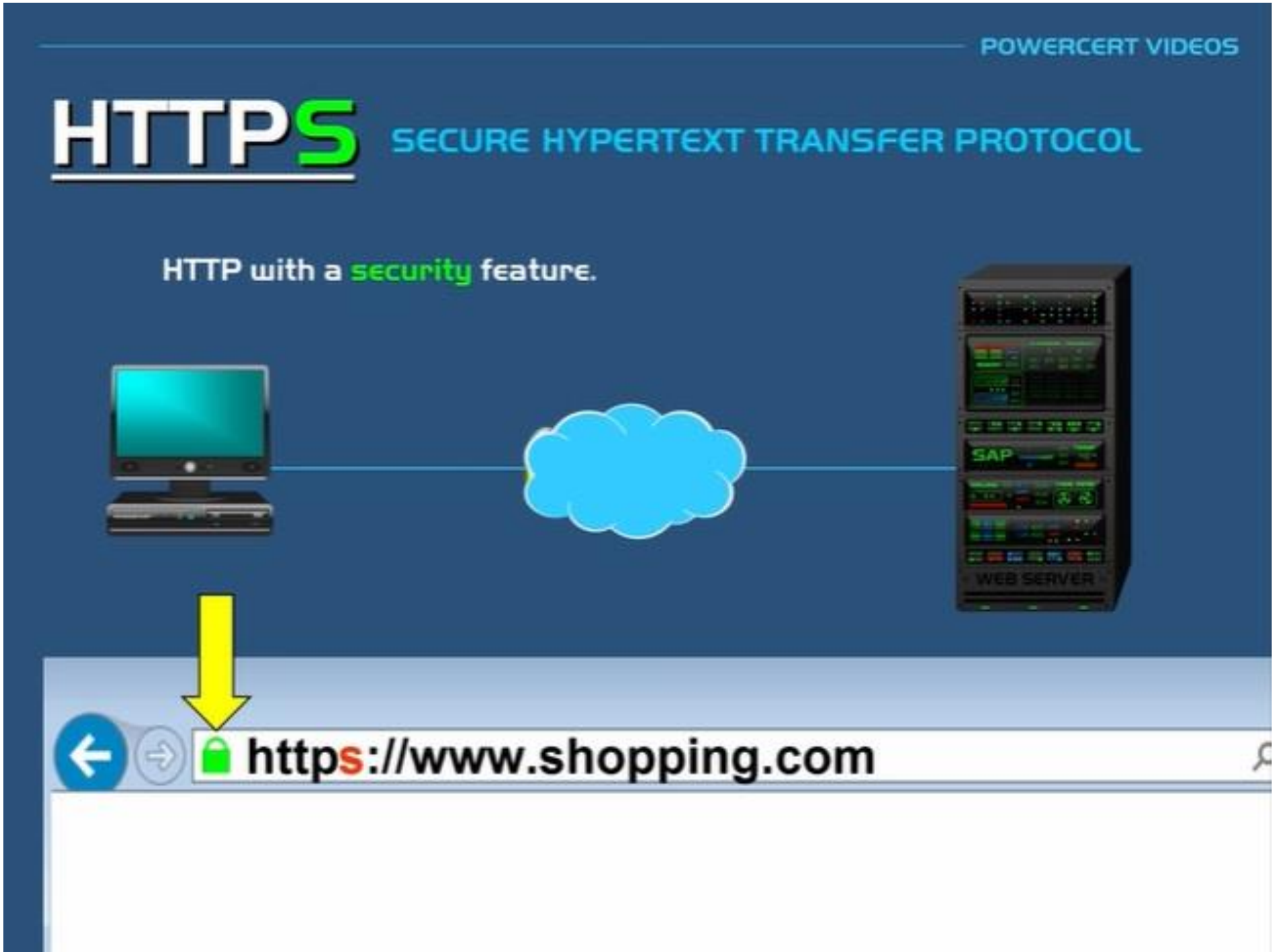
HTTPS

SECURE HYPERTEXT TRANSFER PROTOCOL

HTTP with a security feature.

Encrypts the data that is being retrieved by HTTP.

Uses encryption algorithms to scramble the data that's being transferred.


POWERCERT VIDEOS
HTTPS
SECURE HYPERTEXT TRANSFER PROTOCOL
HTTP with a security feature.
SAP
WEB SERVER
https://www.shopping.com

TELNET and SSH (Secure Shell)

Are the general purpose client server application program and uses remote terminal service which allows a user at one site to interact with a remote time-sharing system at another site as if the user's keyboard and a display connected directly to the remote machine. The main difference between Telnet and SSH is that the Telnet is conventional protocol whereas SSH is the replacement for Telnet protocol and also SSH have enhanced features.

TELNET is a client-server program that permits the user to retrieve any application program on a remote computer. The function of a telnet is to provide the services to the user on the remote computer and transferring the result to the local computer.

Telnet

Run programs

Create folders

Delete files

Create files

Transfer files

Browse directories

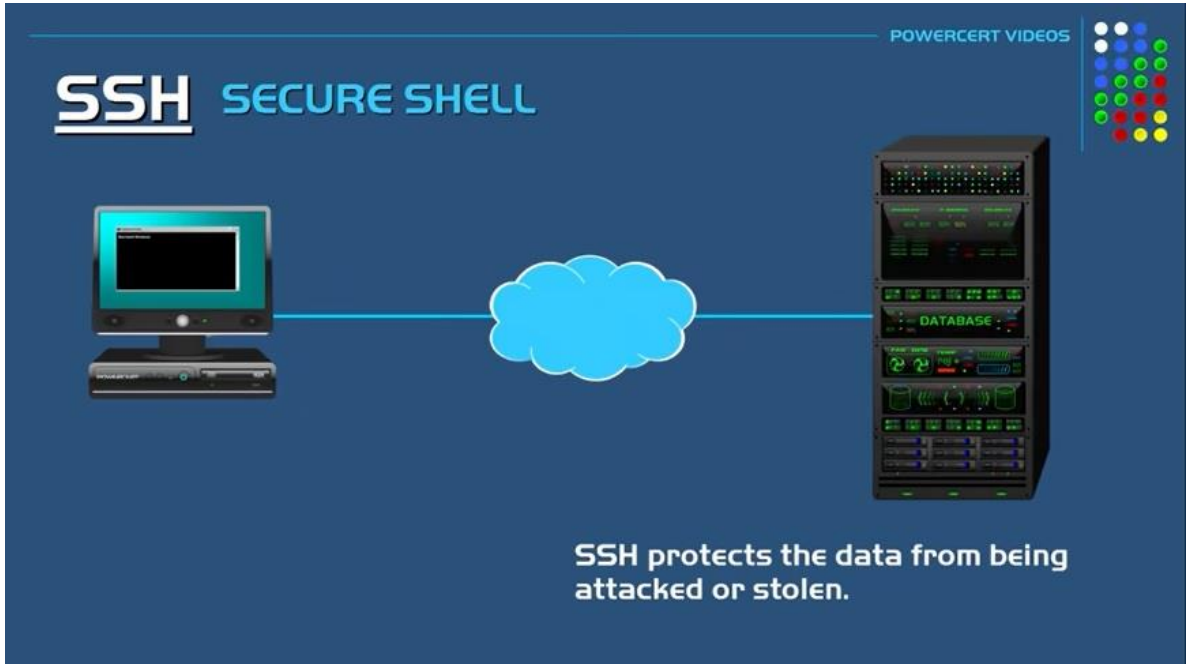
Start/stop services

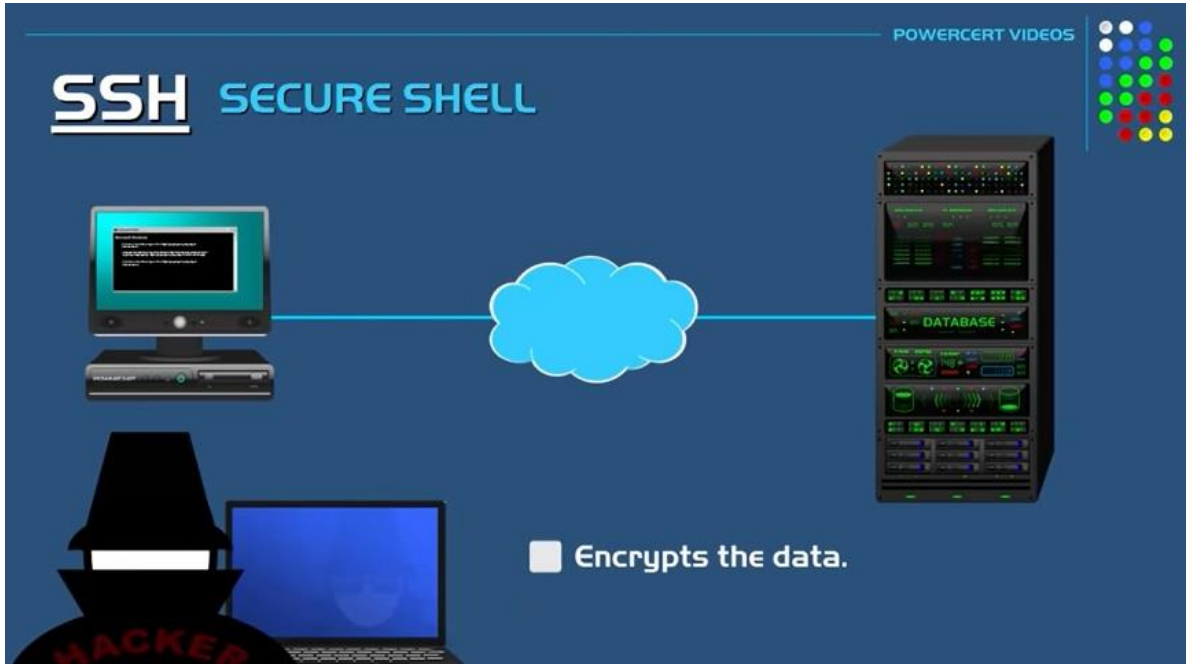
Secure Shell (SSH)

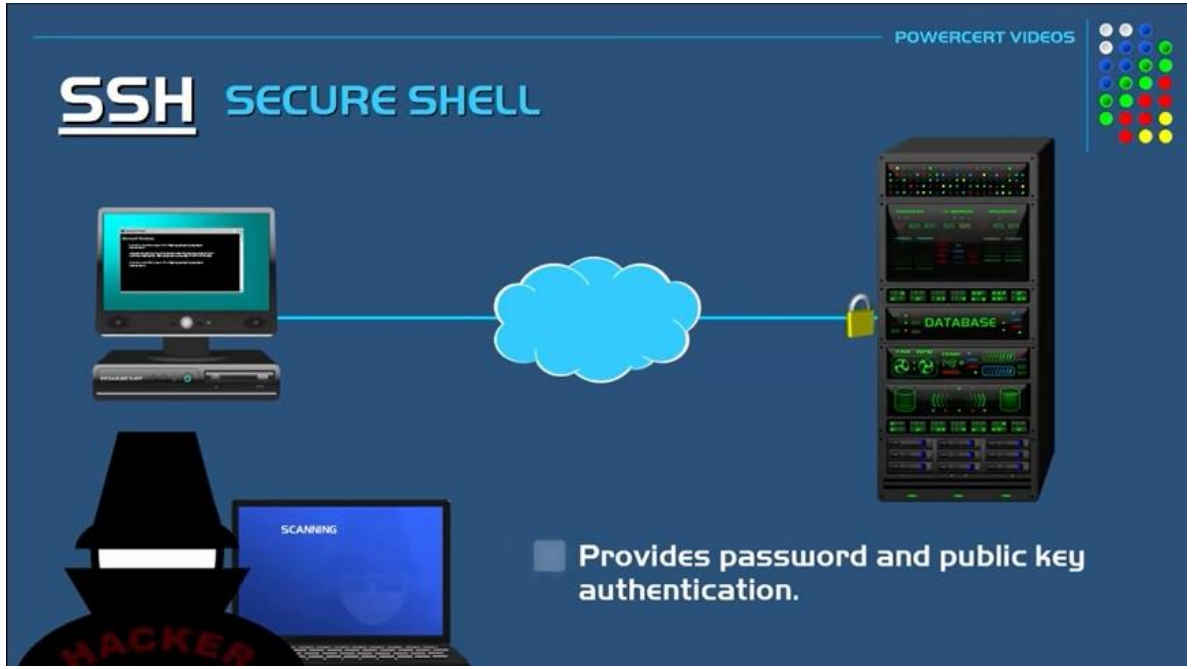
Is a protocol for secure network communications designed to be relatively simple and inexpensive to implement. SSH also provides a more general client/server capability and can be used for such network functions as file transfer and email. SSH client and server applications are widely available for most operating systems.

SSH is organized as three protocols that typically run on top of TCP

- **Transport Layer Protocol:** Provides server authentication, data confidentiality, and data integrity with forward secrecy (i.e., if a key is compromised during one session, the knowledge does not affect the security of earlier sessions). The transport layer may optionally provide compression.
- **User Authentication Protocol:** Authenticates the user to the server.
- **Connection Protocol:** Multiplexes multiple logical communications channels over a single, underlying SSH connection


POWERCERT VIDEOS
SSH SECURE SHELL
DATABASE
SSH protects the data from being attacked or stolen.


POWERCERT VIDEOS
SSH SECURE SHELL
DATABASE
HACKER
Encrypts the data.


POWERCERT VIDEOS
SSH SECURE SHELL
DATABASE
SCANNING
HACKER
Provides password and public key authentication.

Cryptography and Network Security II
4th year

Lecture No. 2

Electronic Mail Security

DR. Lecturer . Taqwa .F. Hassan

Computer Department - College of Engineering
University of Diyala

2023-2024

Roadmap

- Overview of Emails
- Email Services and Security
- PGP (Pretty Good Privacy)
- S/MIME

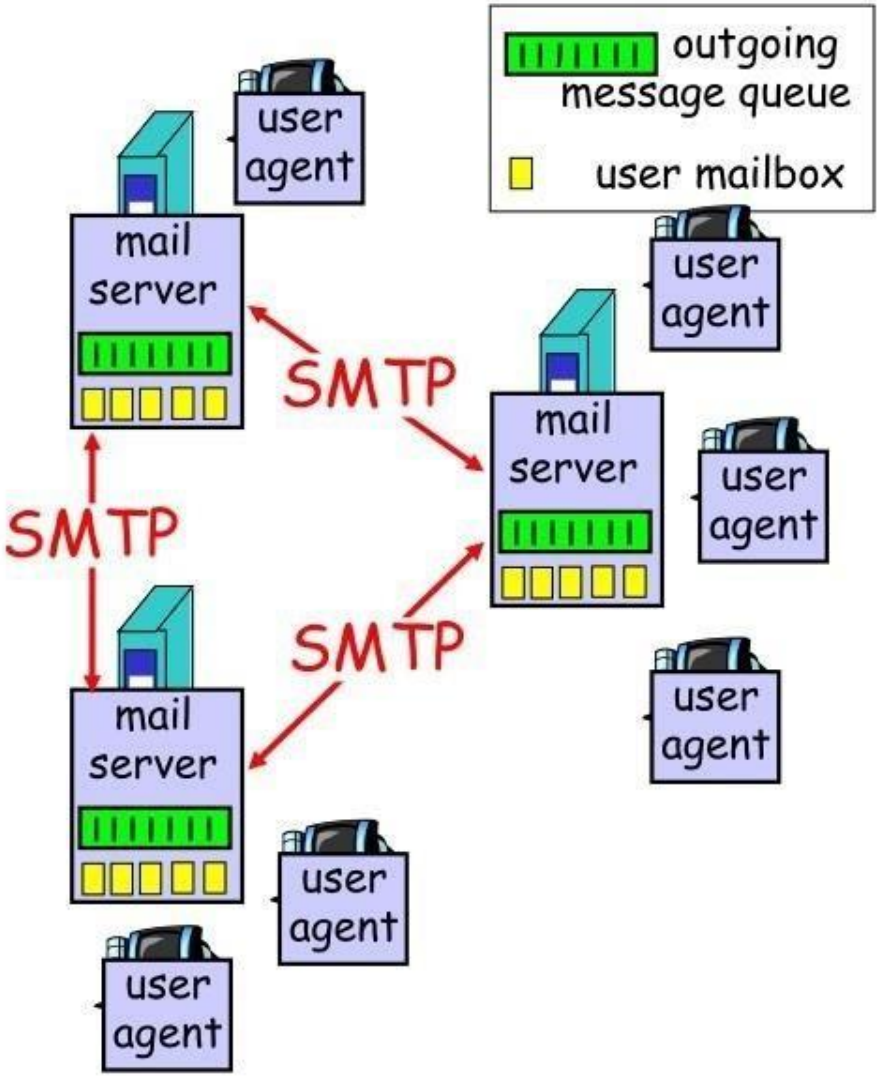
Overview of Electronic Mail

Three major components:

- user agents (UAs)
- mail servers
- simple mail transfer protocol: SMTP
  - Mail Transfer Agents (MTAs)

User Agent

- Known as "mail reader"
- composing, editing, reading mail messages
- e.g., Eudora, MS Outlook, Outlook Express, Netscape Messenger
- outgoing, incoming messages stored on server



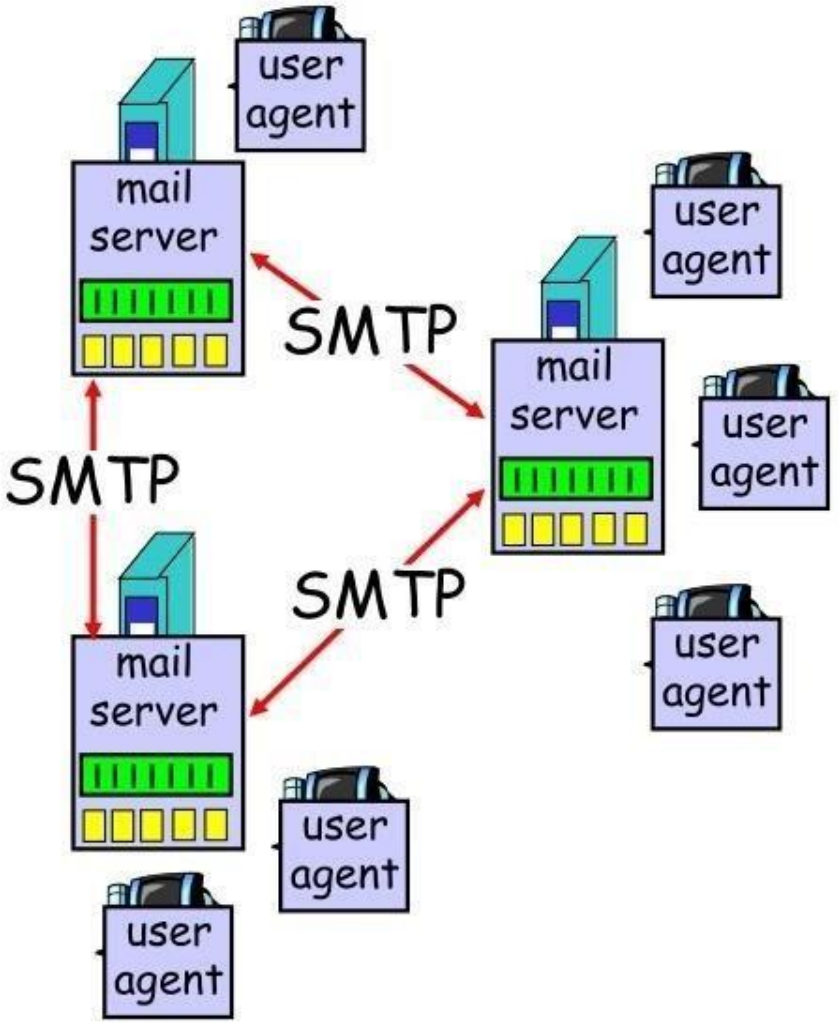
Electronic Mail (cont.)

Mail Servers

- mailbox contains incoming messages for user
- message queue of outgoing (to be sent) mail messages

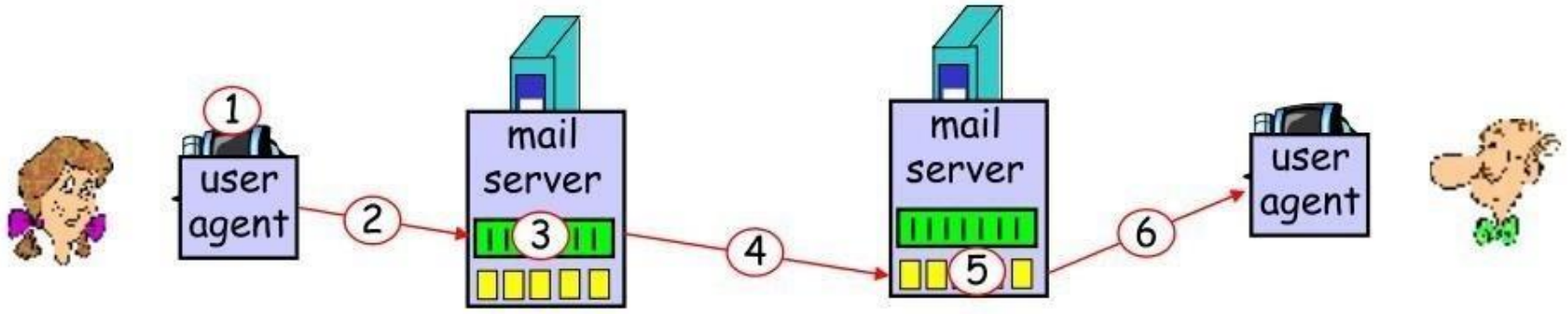
SMTP protocol

- Deliver emails from user agent to user's mail server
- Deliver emails between mail servers

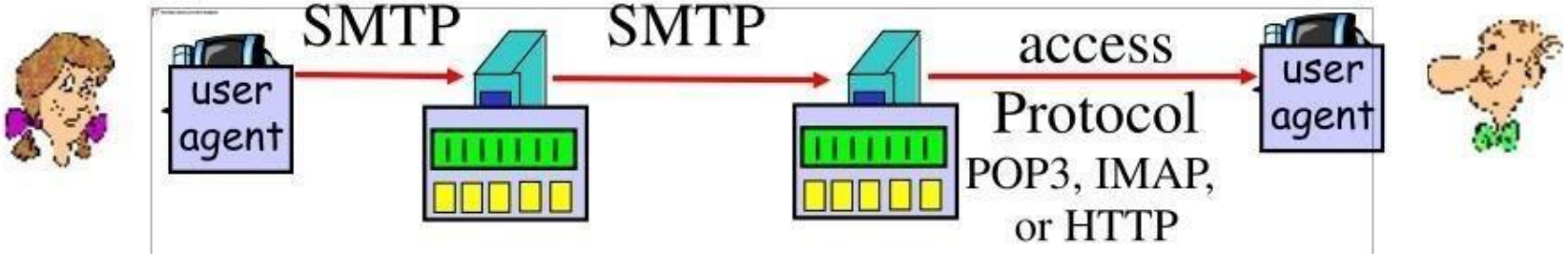


SMTP Protocol

- Uses TCP to reliably transfer email message from client to server, port 25
- Direct transfer: sending server to receiving server via many Mail Transfer Agents (MTAs)



Mail Access Protocols



- SMTP: delivery/storage to receiver's server
- Mail access protocol: retrieval from server
  - POP: Post Office Protocol [RFC 1939]
    - authorization (agent <--> server) and download
  - IMAP: Internet Mail Access Protocol [RFC 1730]
    - more features (more complex)
    - manipulation of stored messages on server
  - HTTP (web-based email): Hotmail , Yahoo! Mail, etc.

Roadmap

- Overview of Email
- Email Services and Security
- PGP (Pretty Good Privacy)
- S/MIME

Email Security

- email is one of the most widely used and regarded network services
- currently message contents are not secure
  - may be inspected either in transit
  - or by suitably privileged users on destination system

Email Security Enhancements

- confidentiality
  - protection from disclosure
- authentication
  - of sender of message
- message integrity
  - protection from modification
- non-repudiation of origin
  - protection from denial by sender

Roadmap

- Overview of Email
- Email Services and Security
- PGP
- S/MIME

What is PGP?

- PGP - Pretty Good Privacy
- general purpose application to protect (encrypt and/or sign) files
- can be used to protect e-mail messages
- can be used by corporations as well as individuals
- based on strong cryptographic algorithms (IDEA, RSA, SHA-1)
- available free of charge at http://www.pgpi.org
- first version developed by Phil Zimmermann
- PGP is now on an Internet standards track (RFC 3156)

PGP services

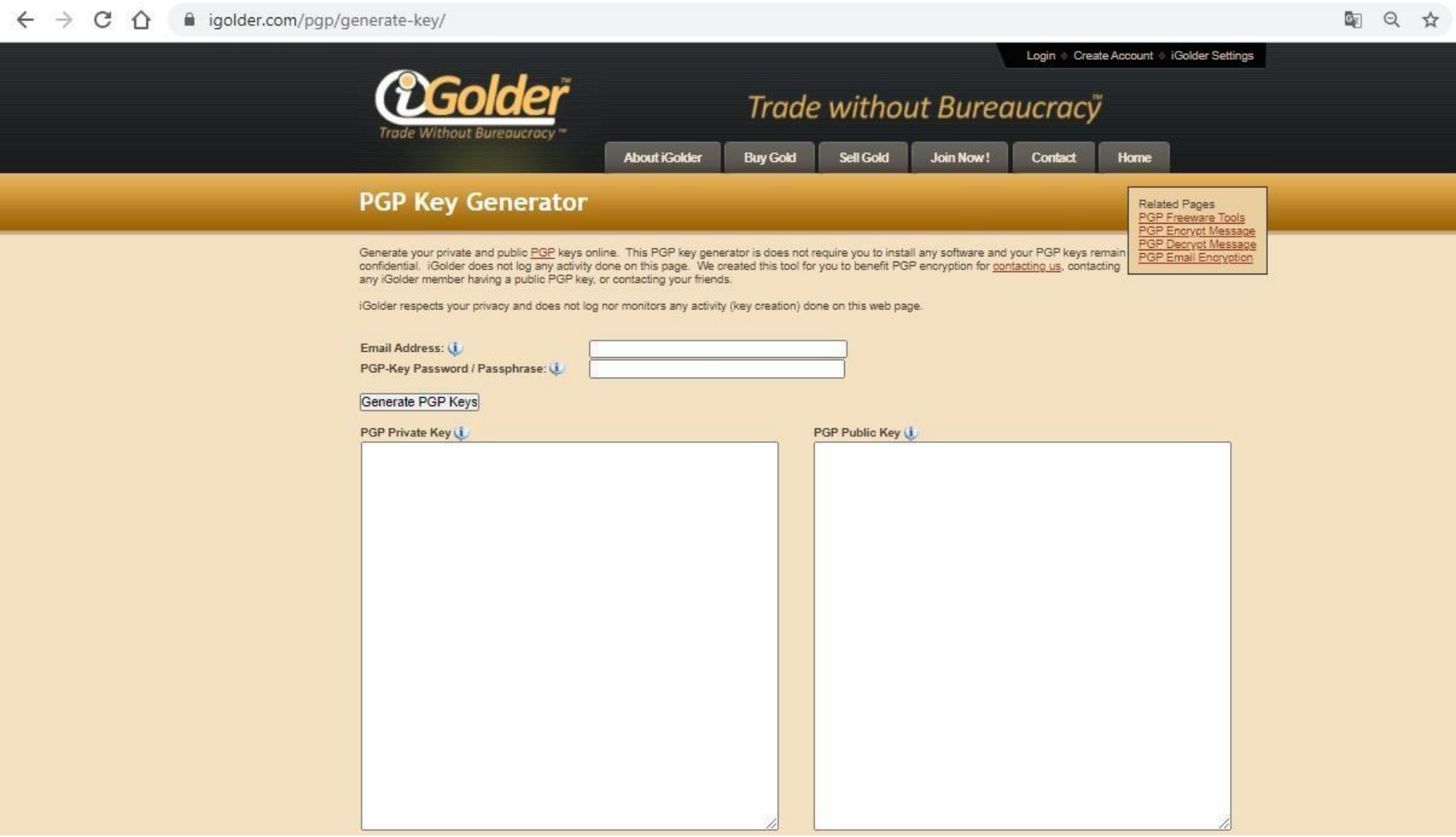
- messages
  - authentication
  - confidentiality
  - compression
  - e-mail compatibility
  - segmentation and reassembly
- key management
  - generation, distribution, and revocation of public/private keys
  - generation and transport of session keys and IVs

Summary of PGP Services

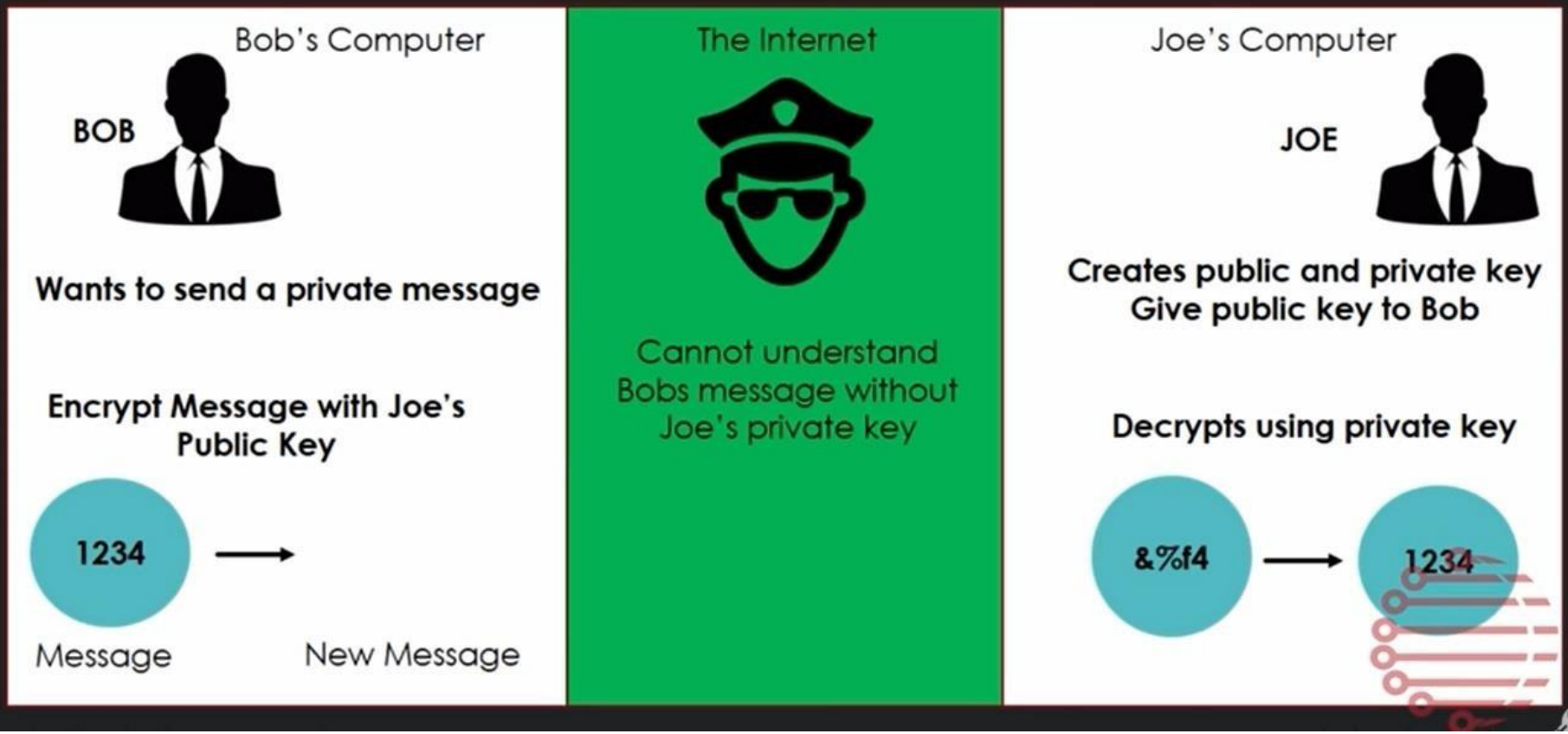
Function	Algorithms Used	Description
Digital signature	DSS/SHA or RSA/SHA	A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key and included with the message.
Message encryption	CAST or IDEA or Three-key Triple DES with Diffie-Hellman or RSA	A message is encrypted using CAST-128 or IDEA or 3DES with a one-time session key generated by the sender. The session key is encrypted using Diffie-Hellman or RSA with the recipient's public key and included with the message.
Compression	ZIP	A message may be compressed, for storage or transmission, using ZIP.
Email compatibility	Radix 64 conversion	To provide transparency for email applications, an encrypted message may be converted to an ASCII string using radix 64 conversion.
Segmentation	—	To accommodate maximum message size limitations, PGP performs segmentation and reassembly.

PGP Key Rings

- each PGP user has a pair of keyrings:
  - public-key ring contains all the public-keys of other PGP users known to this user, indexed by key ID
  - private-key ring contains the public/private key pair(s) for this user, indexed by key ID & encrypted keyed from a hashed passphrase
- security of private keys thus depends on the pass-phrase security


igolder.com/pgp/generate-key/
Login ♦ Create Account ♦ iGolder Settings
iGolder
Trade Without Bureaucracy™
Trade without Bureaucracy™
About iGolder
Buy Gold
Sell Gold
Join Now!
Contact
Home
PGP Key Generator
Related Pages
PGP Freeware Tools
PGP Encrypt Message
PGP Decrypt Message
PGP Email Encryption
Generate your private and public PGP keys online. This PGP key generator is does not require you to install any software and your PGP keys remain confidential. iGolder does not log any activity done on this page. We created this tool for you to benefit PGP encryption for contacting us, contacting any iGolder member having a public PGP key, or contacting your friends.
iGolder respects your privacy and does not log nor monitors any activity (key creation) done on this web page.
Email Address:
PGP-Key Password / Passphrase:
Generate PGP Keys
PGP Private Key
PGP Public Key

Understanding PGP



Roadmap

- Overview of Email
- Email Services and Security
- PGP (Pretty Good Privacy)
- S/MIME

S/MIME

- Secure Multi-purpose Internet Mail Extension
- security enhancement to MIME email
  - original Internet RFC822 email was text only
  - MIME provided support for varying content types and multi-part messages
  - with encoding of binary data to textual form
  - S/MIME added security enhancements
- have S/MIME support in many mail agents
  - eg MS Outlook, Mozilla, Mac Mail etc

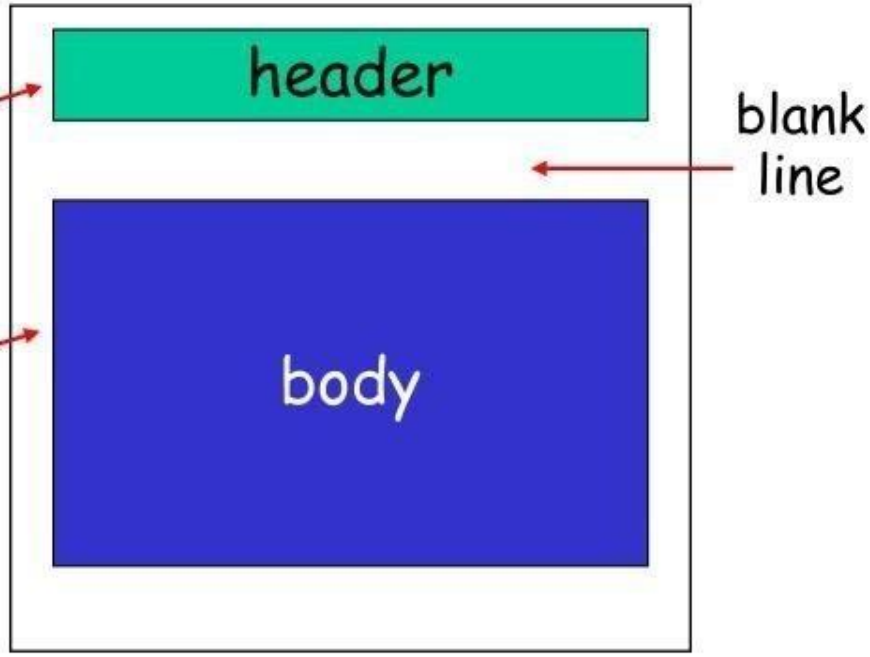
S/MIME services

- **enveloped data** (application/pkcs7-mime; smime-type = enveloped-data)
  - standard digital envelop
- **signed data** (application/pkcs7-mime; smime-type = signed-data)
  - standard digital signature ("hash and sign")
  - content + signature is encoded using base64 encoding
- **clear-signed data** (multipart/signed)
  - standard digital signature
  - only the signature is encoded using base64
  - recipient without S/MIME capability can read the message but cannot verify the signature
- **signed and enveloped data**
  - signed and encrypted entities may be nested in any order

Mail Message Format

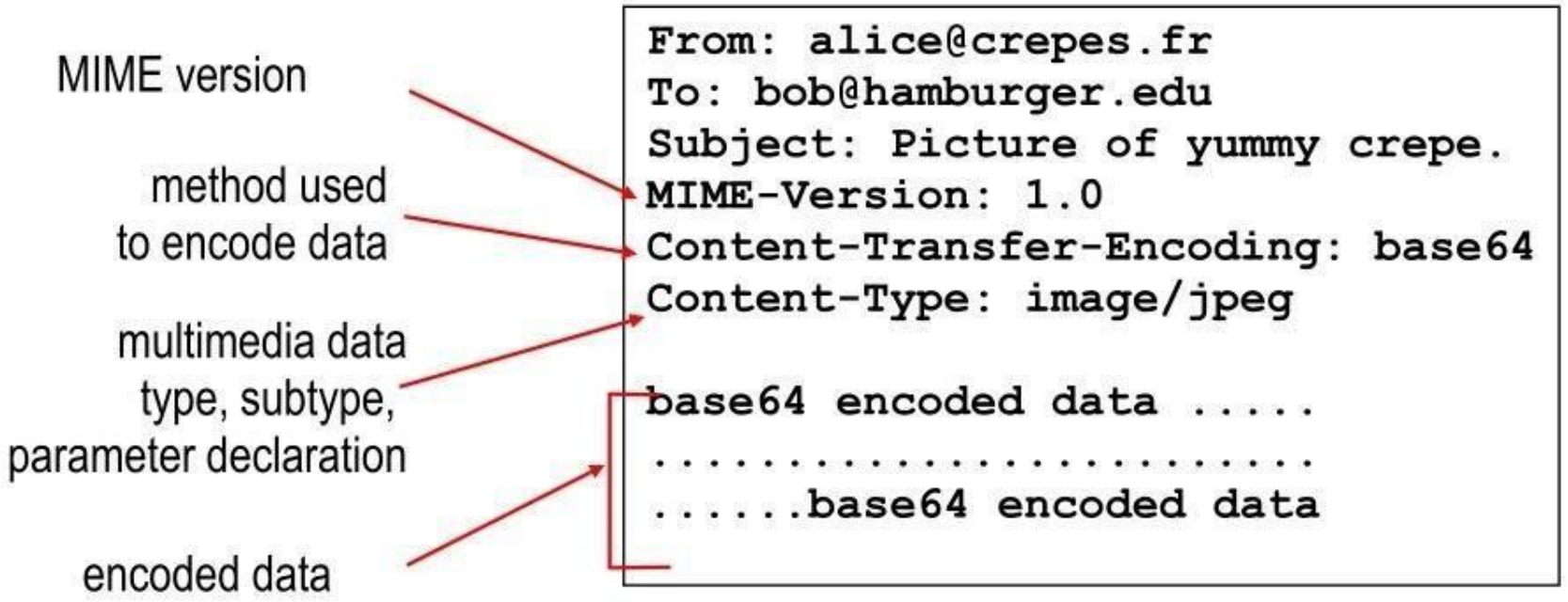
RFC 822: standard for text message format:

- header lines, e.g.,
  - To:
  - From:
  - Subject:
- body
  - the "message", 7-bit ASCII characters only



MIME

- Enable sending multimedia messages or attachments with non-ASCII format
- Additional lines in msg header declare MIME content type



The Received Message

```
Received: from crepes.fr by hamburger.edu; 12 Oct 98
15:27:39 GMT
From: alice@crepe.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

base64 encoded data . . . . .
. . . . .. . . . .. . . . .
  . . . .  Base64 encoded data
```

MIME Content Types

Type	Subtype	Description
Text	Plain	Unformatted text; may be ASCII or ISO 8859.
	Enriched	Provides greater format flexibility.
Multipart	Mixed	The different parts are independent but are to be transmitted together. They should be presented to the receiver in the order that they appear in the mail message.
	Parallel	Differs from Mixed only in that no order is defined for delivering the parts to the receiver.
	Alternative	The different parts are alternative versions of the same information. They are ordered in increasing faithfulness to the original, and the recipient's mail system should display the "best" version to the user.
	Digest	Similar to Mixed, but the default type/subtype of each part is message/rfc822.
Message	rfc822	The body is itself an encapsulated message that conforms to RFC 822.
	Partial	Used to allow fragmentation of large mail items, in a way that is transparent to the recipient.
	External-body	Contains a pointer to an object that exists elsewhere.
Image	jpeg	The image is in JPEG format, JFIF encoding.
	gif	The image is in GIF format.
Video	mpeg	MPEG format.
Audio	Basic	Single-channel 8-bit ISDN mu-law encoding at a sample rate of 8 kHz.
Application	PostScript	Adobe Postscript
	octet-stream	General binary data consisting of 8-bit bytes.

text/plain

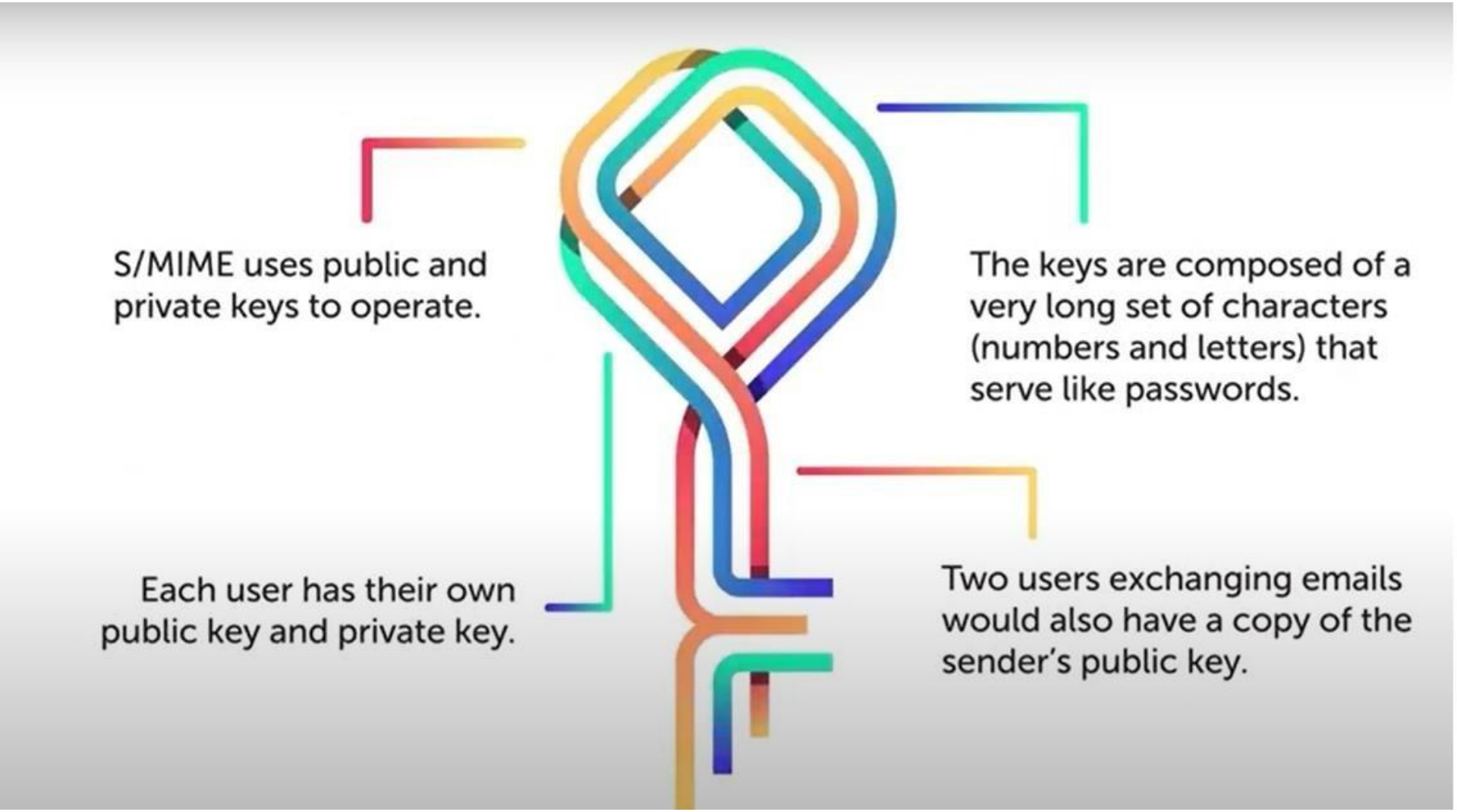
image/jpeg

MIME Transfer Encodings

7bit	The data are all represented by short lines of ASCII characters.
8bit	The lines are short, but there may be non-ASCII characters (octets with the high-order bit set).
binary	Not only may non-ASCII characters be present but the lines are not necessarily short enough for SMTP transport.
quoted-printable	Encodes the data in such a way that if the data being encoded are mostly ASCII text, the encoded form of the data remains largely recognizable by humans.
base64	Encodes data by mapping 6-bit blocks of input to 8-bit blocks of output, all of which are printable ASCII characters.
x-token	A named nonstandard encoding.

S/MIME Cryptographic Algorithms

- digital signatures: DSS & RSA
- hash functions: SHA-1 & MD5
- session key encryption: ElGamal & RSA
- message encryption: AES, Triple-DES, RC2/40 and others
- MAC: HMAC with SHA-1
- have process to decide which algs to use


S/MIME uses public and private keys to operate.
The keys are composed of a very long set of characters (numbers and letters) that serve like passwords.
Each user has their own public key and private key.
Two users exchanging emails would also have a copy of the sender's public key.

Key Differences Between PGP and S/MIME

1.PGP is designed to process plain text emails while S/MIME allows the emails containing the multimedia files too.

2.S/MIME is appropriate for industry use. As against, PGP serves a good purpose for personal and office use.

3. S/MIME is expensive as compared to PGP.

4.In terms of efficiency, the S/MIME is better than the PGP because of its centralized key management.

Cryptography and Network Security II

4th year

Lecture No. 3+4

Wireless Networks , IP Security ,

System security

DR. Lecturer . Taqwa.F.Hassan

Computer Department - College of Engineering
University of Diyala

2023-2024

Wireless Security

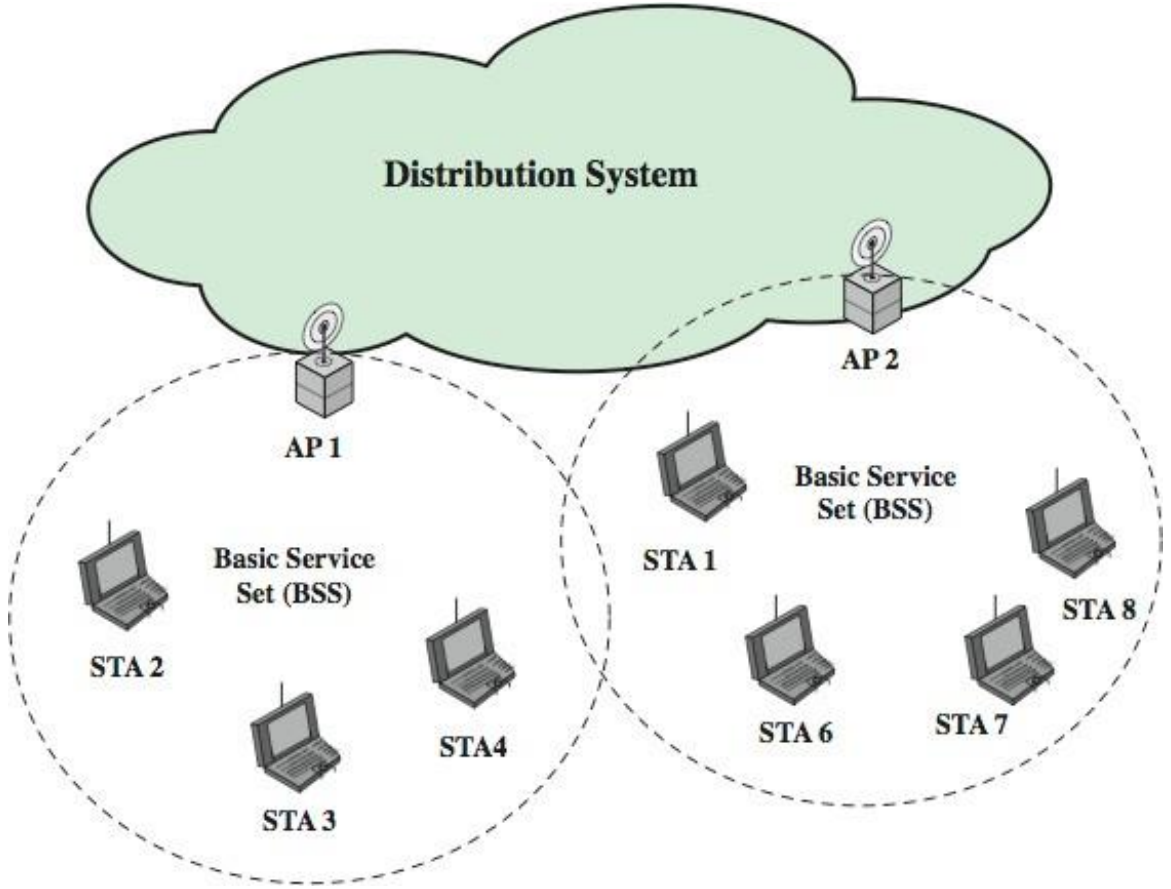
- Very convenient to have wireless connections
- Nightmare for security – the range for 802.11 is often a few hundred meters, so that one can spy on a company by leaving a laptop on in the parking lot
- Many problems arise from the vendors trying to make their products as friendly as possible – when you plug the device it starts working right away, usually with no security by default
- Take a look here at several ways wireless nets handle security
  - 802.11i
  - Bluetooth
  - Wap 2.0

802.11 security – key points

- IEEE 802.11 is a standard for wireless LANs
  - interoperable standard-compliant implementations are called Wi-Fi
- IEEE 802.11i specifies security standards for IEEE 802.11 LANs
  - interoperable implementations called Wi-Fi Protected Areas (WPA)
- Wireless Application Protocol (WAP) – standard to provide mobile use of wireless phones and other wireless devices access to telephony and information services, including the internet
- WAP security primarily provided by Wireless Transport Layer Security (WTLS) – provides security between the mobile device and the WAP gateway to the Internet

Network components and architectural model

- BSS
  - Client stations do not talk to each other directly
  - everything relayed through the AP
  - connections between a station and a BSS are dynamic
- Independent BSS (IBSS)
  - client stations talk directly to each other
  - typically an ad-hoc network



802.11 security

- 802.11 standard prescribed a data link-level security protocol called **WEP (Wired Equivalent Privacy)** designed to make the security of a wireless LAN as good as that of a wired LAN
- When 802.11 security is enabled, each station has a secret key shared with the AP – not specified how keys are distributed
- WEP encryption uses a stream cipher based on RC4 (generate a key stream that is XORed with the data), the IV used to augment the RC4 key is sent in plain
  - **WEP has been broken already in July 2001 (Borisov et al.)**
- **Solution**
  - **Replace WEP with WPA (Wi-fi Protected Access) or WPA2**
  - **Final proposal in 802.11i: Robust Security Network (RSN)**
    - **The Wi-Fi Alliance certifies vendors in compliance with the full 802.11i specification under the WPA 2 program**

Bluetooth Security

- Considerably shorter range than 802.11 – cannot be attacked from the parking lot but feasible to attack it from next office
  - An attacker next door can read the signals from one's keyboard or the data sent to the Bluetooth printer in the next office
- Bluetooth has 3 security modes ranging from nothing to full data encryption and integrity control – many users have security turned off

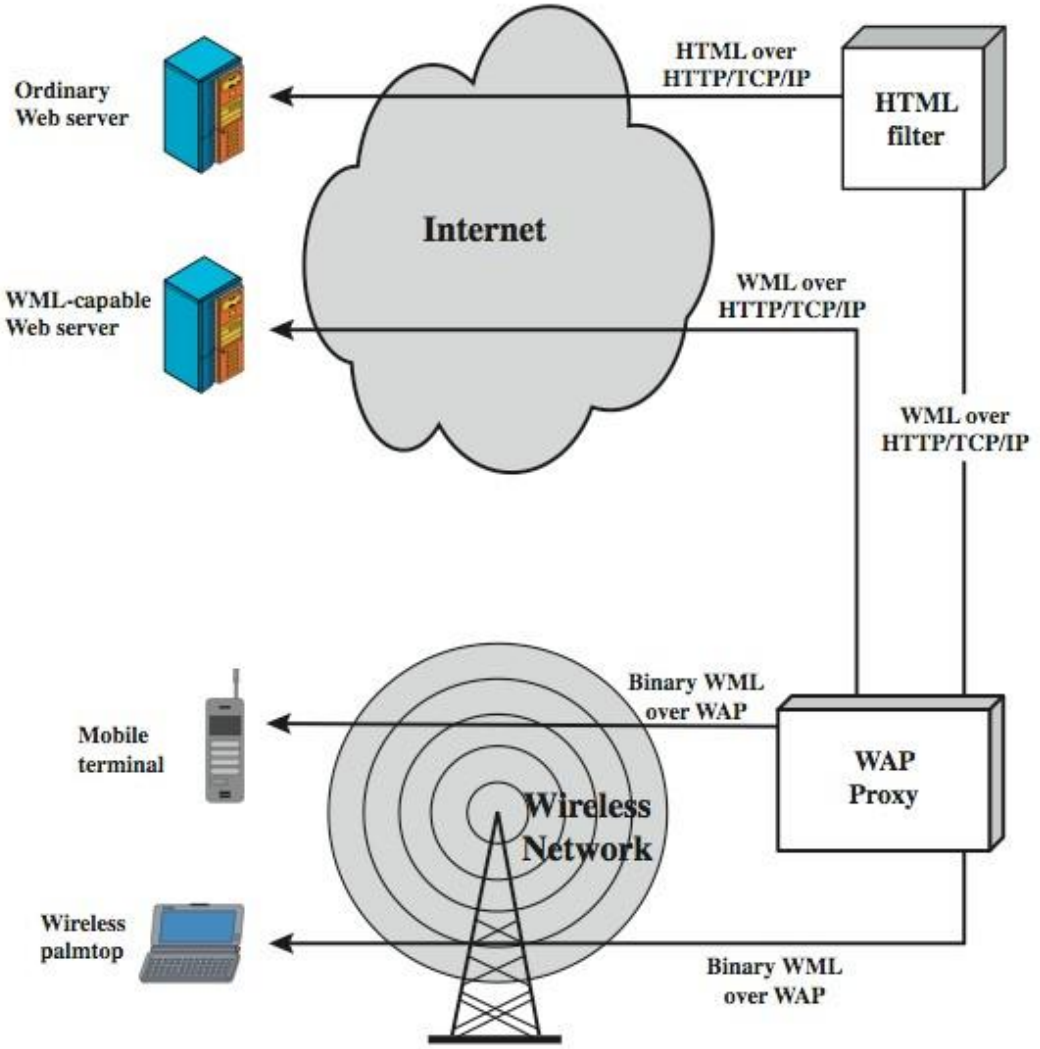
- Many reports about various attacks on Bluetooth
- Concerns about many devices left on and visible by default

WAP (Wireless Application Protocol) 2.0 security

- Introduced for handheld-devices connectivity (mobile phones)
- Uses standard protocols in all layers
- It is IP-based and it supports IPsec in the network layer
- TCP connections protected by TLS in the transport layer
- Uses HTTP client authentication
- Probably better than 802.11 and Bluetooth because it only relies on well-known security standards

WAP Infrastructure

- WAP architecture is designed to cope with the two principal limitations of wireless Web access:
  - the limitations of the mobile node (small screen size, limited input capability)
  - the low data rates of wireless digital networks.



Security for security-ignorant applications

- Applications providing security have been developed for email (S/MIME, PGP), client/server (Kerberos), Web access (SSL), etc.
- *Question*: Can security be provided also for the security-ignorant applications?
- Also, it is desirable to implement security in such a way that users are not required to pass special training and are not required to do security tasks on a daily basis
- *Solution*: Implement security at IP level – in this way, all communications are secured and/or authenticated without changing the existing applications and protecting the security-unaware users

IPsec

- Issued in 1998, the design for a new network layer was called IPsec (IP security) – RFC 2401, 2402, 2406, and others
  - IPsec can encrypt and/or authenticate all traffic at IP level – remote logon, client/server, email, file transfer, Web access, etc.
  - Major services of IP sec: secrecy, data integrity, and protection from replay attacks – all of these are based on symmetric-key cryptography.
  - Flexible design: one cryptographic protocol can be easily replaced with another, a null algorithm is also specified for those users who really do not want (cannot afford) security.

Applications of IPsec

- **Secure branch office connectivity over the Internet:** a company can build a secure virtual private network over the Internet or over a public WAN – this enables a business to rely heavily on Internet rather than on private networks
- **Secure remote access over the Internet:** an end user having a system equipped with IPsec protocols can connect to the company's servers using the local Internet service provider – reduces costs for traveling employees
- **Enhances e-commerce security** – use of IPsec beneath the usual security protocols should enhance the security and the public trust in this industry

IPsec scenario

•The company maintains various LANs with nonsecure IP traffic inside the LAN

•For traffic offsite, secure IP traffic is used

•The protocols are used in the networking devices, e.g., in the firewalls – these devices will typically encrypt/decrypt all traffic – these operations are transparent to the other applications

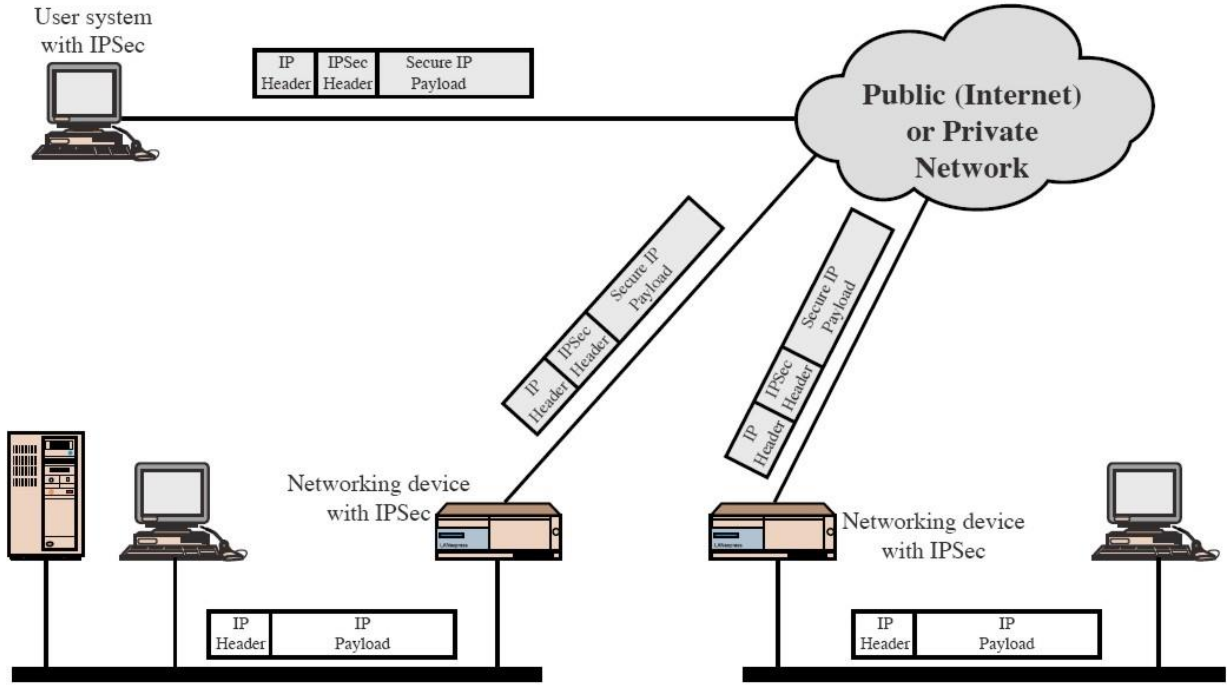


Figure 16.1 An IP Security Scenario

Briefly about the Internet Protocol (IP)

- IPv4 has been the keystone of the TC/IP protocol for many years
  - Header shown to the right – length 20 bytes
  - In particular, IPv4 only allows for addresses on 32 bits
  - Feb 2011: last batch of address blocks assigned
- 1996 – specifications for a next- generation IP: IPv6
  - Header shown bellow – length 40 bytes
  - Addresses on 128 bits
  - Longer header but fewer fields – faster processing in the routers
  - Migration to IPv6 expected to take many years
  - Oct 2011: 3% of domain names, 12% of the networks on the internet have IPv6 support
- IP sec optional in IPv4, mandatory in IPv6

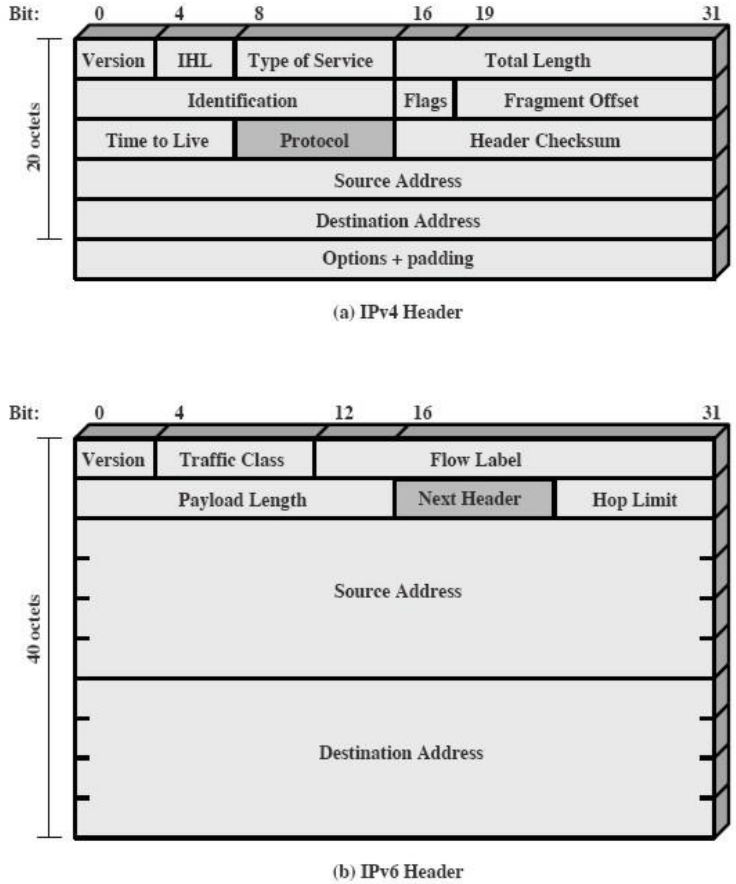


Figure 16.14 IP Headers

IP Security Overview

- Benefits of IPSec
  - Transparent to applications - below transport layer (TCP, UDP)
  - Provide security for individual users
- IPSec can assure that:
  - A router or neighbor advertisement comes from an authorized router
  - A redirect message comes from the router to which the initial packet was sent
  - A routing update is not forged

Usage of IP Security:

IPsec can be used in many things, those are below:

1.It helps to encrypt the application layer data.

2.It also provides the security for the routers so that it can transmit the data through the public internet.

3.It gives the authentication without encryption, and this data originates from the known sender.

4.It protects the network data which set up the circuit by using IPsec tunneling and data get sent between the two endpoints after that it gets connected with the Virtual Private Network (VPN).

System security

Intruder detection

Password management

Viruses

Intruders

- More of a system (computer) security problem, rather than a *network* security problem but does have influence on it
- Hostile or unwanted trespass by users or software can pose threats to the whole network
- Three general classes of intruders
  - Masquerader – individual who is not authorized to use the computer, exploits a legitimate user's account (attacker is likely to be an outsider)
  - Misfeasor – legitimate user who accesses data, programs, or resources for which such access is not authorized (attacker generally an insider)
  - Clandestine user – individual who seizes supervisory control of the systems and uses it to evade auditing and access control or to suppress audit collection (can be insider or outsider)

Intruder techniques – password security

- In most cases, intruders need to acquire protected information, most often passwords
- Password security is a big problem on any system: hackers may take over someone's user account and use it in a major attack inside that system or against a totally different system – liability may involve the careless user
- The system protects the user passwords in two ways
  - Keeps them "encrypted" on the disk – in fact, one keeps on the disk hashes of the password: in this way, nobody can attack the system by trying to "decrypt" the password file
  - The file with the user personal information should be public so that when the user initiates login, the login process can check his data
  - The password file should be "hidden": /etc/passwd, /etc/shadow in Linux
- To break the password file, the attacker essentially has to "guess" the password of a user, hash it, and then compare it with the entry in the password file

Password selection strategies

- **Users are generally stubborn and insist using bad passwords** – the reason for this is that they actually have to remember them, also they may be bad judges of what is a good password
  - Capitalizing the first/last letter of your friend's name does not make a good password
- **Computer-generated passwords are better**
  - Disadvantage: difficult to remember
  - Programs exist to generate random passwords that can be pronounced, but still difficult to remember
- **Systems have sometimes password checking programs**
  - Check from time to time for bad passwords – ask those users to change them immediately
  - At the time of password selection, check if the password is easy to guess – if so, ask the user to provide another one – most users will not complain and they will realize that there are actually many good passwords they can give
    - E.g., ask all passwords to have length 8
    - Always include an uppercase, lowercase, digit, and a punctuation mark
    - Compile a large collection of bad passwords

Malware as attack tool

	Hardware	Software	Network
Common attacks	• Hardware Trojan • Illegal clones • Side channel attacks (i.e. snooping hardware signals)	• Software programming bugs (e.g. memory management, user input validation, race conditions, user access privileges, etc.) • Software design bugs • Deployment errors	• Networking protocol attacks • Network monitoring and sniffing
Examples of countermeasures	• Tamper-Resistant Hardware (e.g. TPM) • Trusted Computing Base (TCB) • Hardware watermarking • Hardware obfuscation	• Secure coding practice (e.g. type checking, runtime error, program transformation, etc.) • Code obfuscation • Secure design and development • Formal methods	• Firewall • Intrusion prevention and detection • Virtual Private Network (VPN) • Encryption

Common attacks and examples of countermeasures in existing system.

Intrusion detection

- **Statistical methods**: how likely is it for this user to connect at this hour, using this computer, and trying to access this type of resources?
  - Need to find a balance – many false positive alarms will annoy legitimate users who try to work at unusual times and will induce system administrator to ignore alarms, many false negatives allow attackers to penetrate the system
- **Rule-based penetration identification** – based on expert system technology: collect suite of known penetration scenarios and evaluate the likelihood of such a scenario taking place for each alarm. Rules include:

1. Users must not write other users' files
2. Users who login after hours often access same files as they used earlier
3. Users do not generally open disk devices directly, rely on higher-level system utilities
4. Users should not be logged in more than once to the same system
5. Users do not make copies of system programs
6. Users should not read files in other users' personal directories

Malicious software

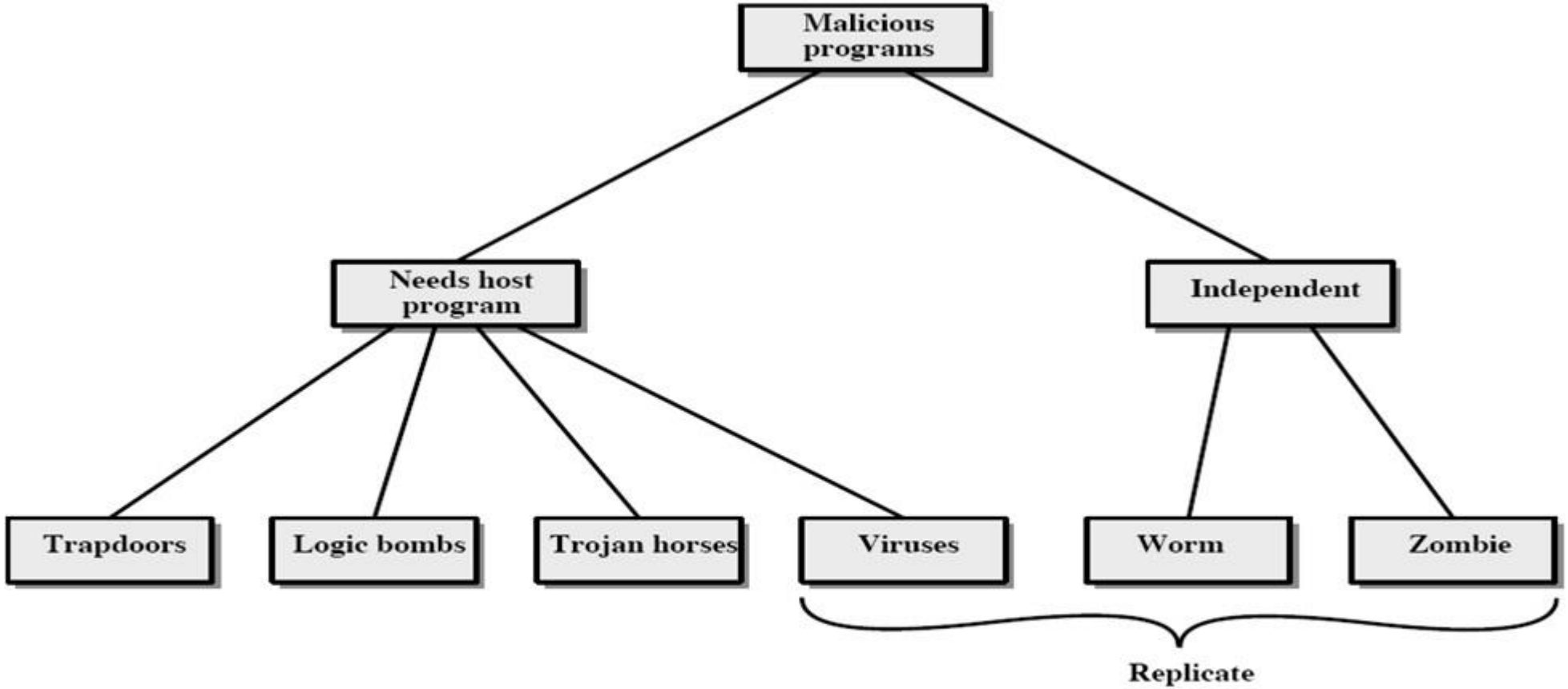


Figure .1 Taxonomy of Malicious Programs

Viruses

- A program that can infect other programs by modifying them, including a copy of the virus program
  - Similar to biological viruses
- Can do anything that other programs can
  - Attaches itself to another program and executes when the program executes
- The stages of a virus life are
  - Dormant phase – virus is idle, will be awaken by a certain trigger – a date, the presence of another program or file, the capacity of the disk exceeding some limit, etc. Not all viruses have this phase
  - Propagation phase – virus places an identical copy of itself into other programs or into certain system areas on the disk – each infected program will carry a clone of the virus which itself will enter a propagation phase
  - Triggering phase – virus is activated to perform its function, can be triggered by a number of events, including a count of its clones
  - Execution phase: function is performed

Types of viruses

- Stealth viruses
  Compression virus – compresses part of its code so that it does not modify the size of the infected file
- Polymorphic viruses – create virus clones that are functionally identical but different in code so that a scan will not show identical clones of the virus
- Macro viruses – make up nowadays two thirds of all viruses
  - Platform independent
  - Infect documents not executable code
  - Easily spread, e.g., by email
- E-mail viruses – send itself to all people on the local address book and then do some local damage

Worms

- Propagates itself from system to system (much like an e-mail virus)
  - A worm will actively seek out more machines to infect without any expected trigger from the human
  - Once active in the system it behaves like a virus
  - To spread, worms use e-mail, remote execution capability, remote login capability
  - Unlike a virus, it need not attach to an existing program
- Same phase as a virus: dormant, propagation, triggering, execution

Virus countermeasures

- Antivirus approaches
  - Detection, identification, removal
  - Use heuristics instead of looking for a signature
  - Activity traps
  - Access control capability
- Advanced Antivirus techniques
  - Digital immune system (prototype of IBM): when a new virus enter, the immune system captures it, analyzes it, adds detection and shielding for it, removes it, and passes information about it to other registered digital systems
  - Behavior blocking software – block potentially malicious software before it does the damage: attempts to open, view, delete, modify files, attempts to format disks, modifications to the logic of executable files, modifications of critical system settings, initiation of network communication, etc.

Emerging threats

1. Social media
2. Cloud computing
3. Smart phones
4. Critical infrastructure البنية التحتية الحرجة.
   - *Terrorism* الإرهاب
   - *Sabotage* تخريب
   - *Information warfare* حرب المعلومات
   - *Natural disaster* كارثة طبيعية
5. Othe remerging areas of concern

- Embedded systems
- sensors

Common characteristics	Common attack patterns
• Millions and billions of active users • Became part of our daily life • No geographical boundaries • Accessed 24/7 from anywhere at anytime • Services are available via Internet connection using Web Browsers • Services offered by many different devices such as mobiles and tablets	• Increased Attack through Web Browser • Increased attacks through social engineering websites • Increasing attacks coming from non-PC-based devices (e.g. mobiles, tablets, VoIP) • Increasing number of more organized attacks through botnet • Increasing number of attacks through the attackers with internal knowledge (i.e. insider threats)

Fig. Emerging Technologies: Their common characteristics and common attack patterns.

Cryptography and Network Security II

4th year

Lecture No. 5

Emerging Topics in Information Security

DR. Lecturer . Taqwa.F.Hassan

Computer Department - College of Engineering
University of Diyala

2023-2024

Internet of Things Security (IoT)

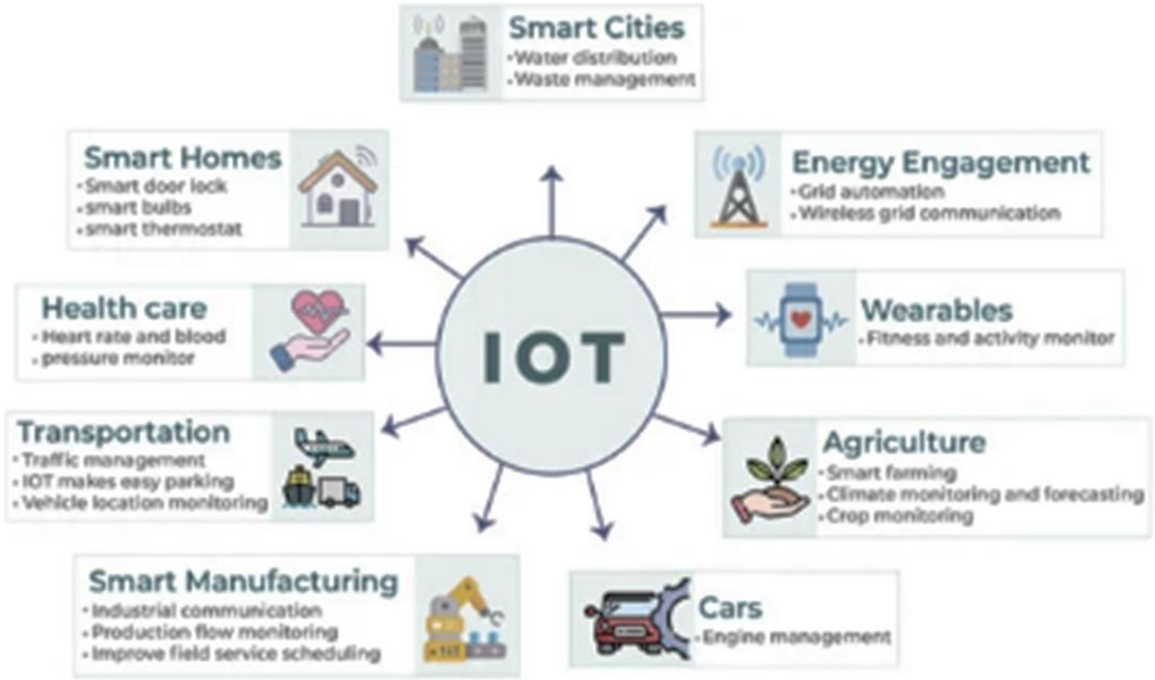
Cloud Computing Security

Cyber Warfare

We raise some issues emerging in the field of computer security. By emerging we mean that these areas are starting to be recognized outside the security community, although we do not mean there are solutions or even approaches to the security problems. Instead we raise these as interesting things to watch over time. In this lecture we discuss the so-called Internet of Things (the trend toward embedding Internet connected computing technology in new technology), Internet of things security, cloud computing security , and what many call cyber warfare (use of computers in political conflicts).

The Internet of Things (IoT) connects different IoT smart objects around people to make their life easier by connecting them with the internet, which leads IoT environments vulnerable to many attacks.

Internet of Things: A world of interconnected smart devices not ordinarily thought of as computers.



What are IoT Devices?

The devices that are **capable of connecting to a network** via the internet are known as Internet of Things devices. The IoT devices do not include the traditional computers such as the laptops and computers.

There was a time where it was only possible to connect standard computers to the internet. But, today, any digital appliance capable of generating huge amounts of data and connected to the internet falls under IoT devices.

IoT devices are usually **embedded in other devices**. When you put in some Internet of things technology in a device, it becomes an internet of things device. For example, a **smart car** is also a car but it is embedded with IoT technology.

Complex functionality and **computational technology** embeds IoT devices. The user need not deal with the complexity at the interface. The main purpose of IoT devices is to improve the functioning of the particular devices. The user can look into the working of the device at the interface easily. IoT devices are continuously connected to the internet.

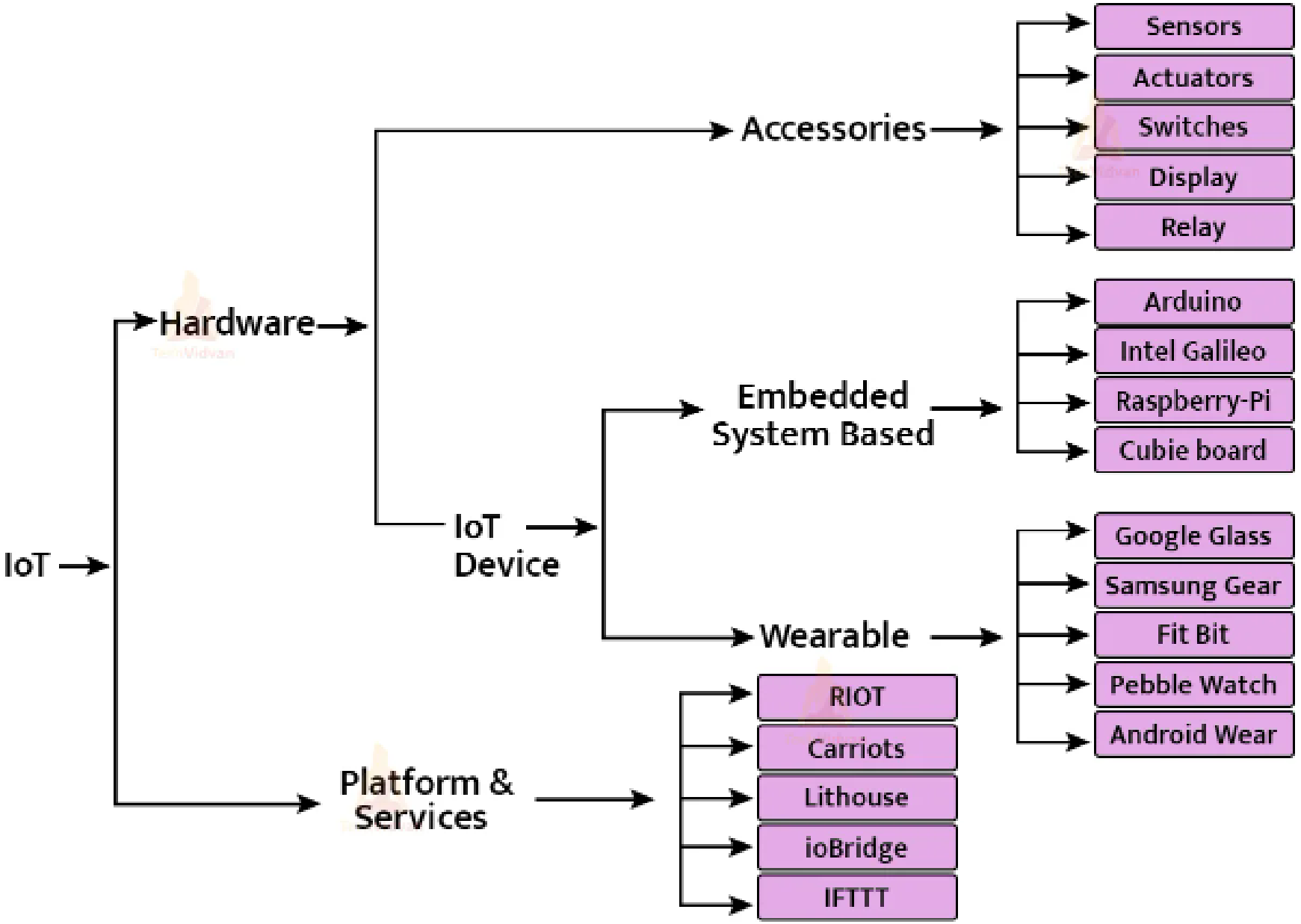
Examples of IoT devices could include all devices capable of connecting to the internet such as home appliances, digital cameras, TVs, cellphones and so on. IoT devices are highly efficient and reliable. The main purpose of IoT devices is to **generate quicker results**.

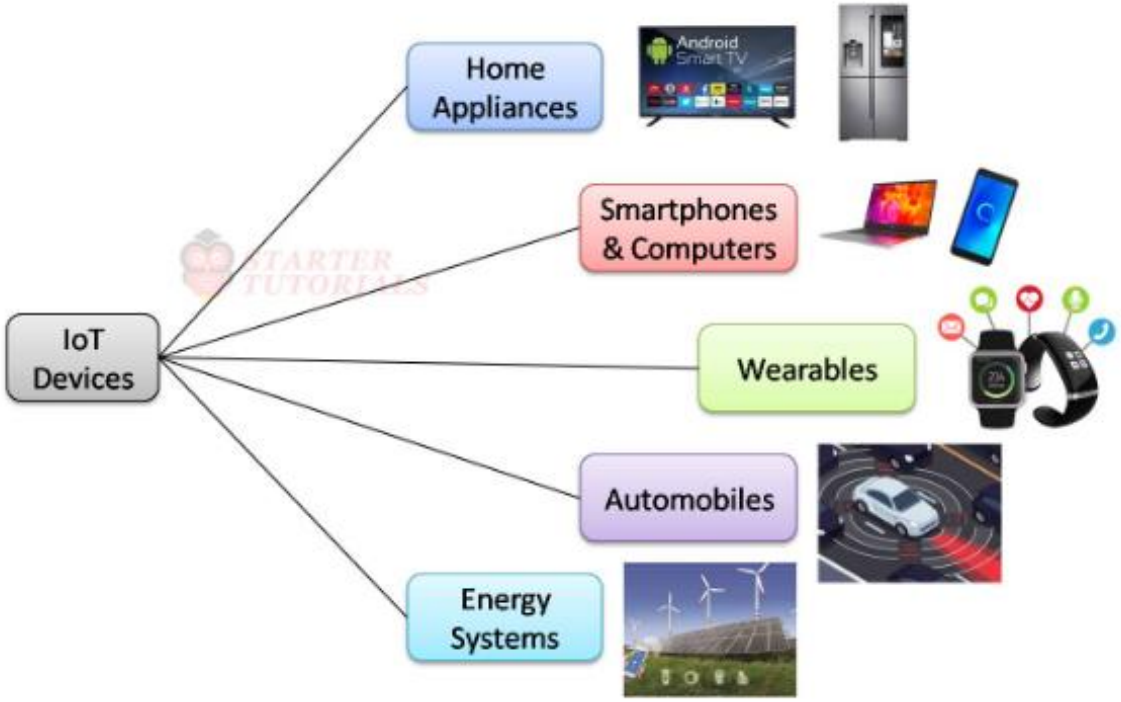
Properties of IoT devices

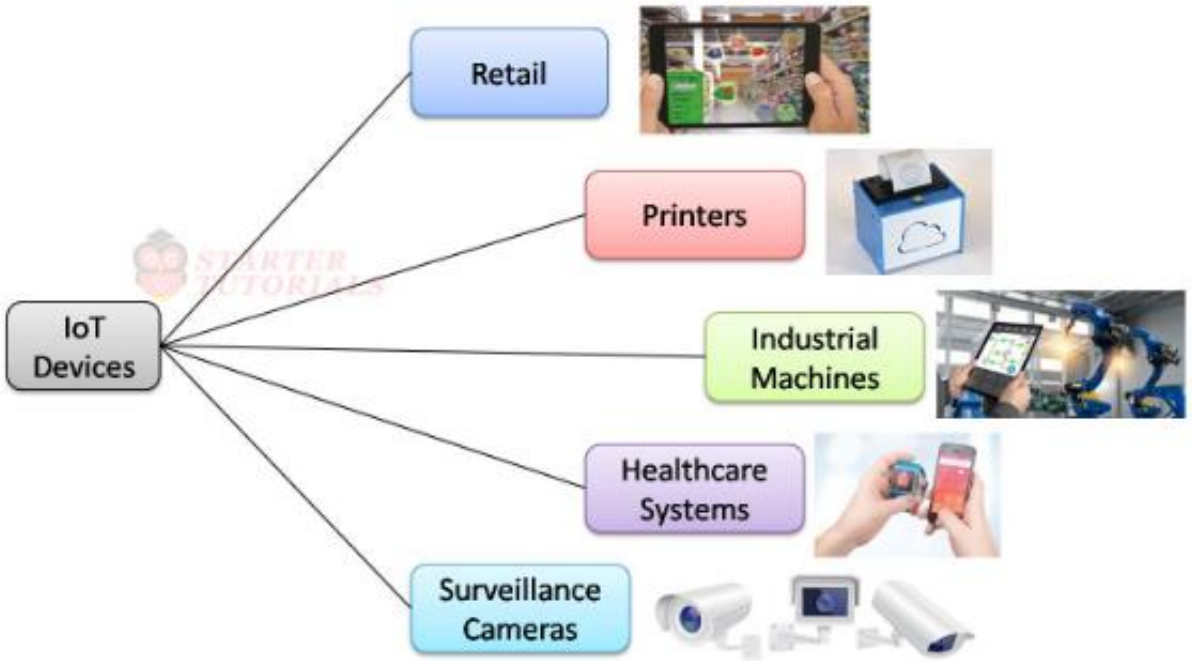
It is not necessary that every IoT device perform the same set of tasks. However, all IoT devices must have some common functionality to be qualified as IoT devices. They are

1. **Sense:** An IoT device must be capable of sensing its physical surroundings. It must be able to retrieve the data from external surroundings.
2. **Sense and receive data:** The data and information that an IoT device collects, it transfers to other devices via a network or connection.
3. **Analyze:** Collecting data is meaningless if one cannot operate on it. Therefore, an IoT device must be able to analyse the data that it collects.
4. **Controlled:** There must be a mechanism that ensures that an IoT device is under the direct control of the end user. Otherwise it may lead to system failure or even hacking.

IoT Devices and Technologies




IoT Devices
Home Appliances
Android Smart TV
Smartphones & Computers
Wearables
Automobiles
Energy Systems


IoT Devices
Retail
Printers
Industrial Machines
Healthcare Systems
Surveillance Cameras

Examples of smart IOT devices :

***smart appliances*.** Your refrigerator can sense when you are running low on milk and add that to your electronic shopping list. Your dishwasher chooses a time to run when electrical demand is low, for example, in the middle of the night, to shift use away from times of peak demand.

***smart home*.** Your home security system reports to you when it senses an intrusion or anomaly. Your heating system coordinates with your calendar to reduce your thermostat when your calendar says you will be away.

***smart health*.** Your exercise monitor interacts with your treadmill to make your workouts more strenuous as your physical condition improves. Your glucose monitor sends reports to your doctor.

smart transportation. Cars, trains, buses, and airplanes operate without human drivers, sensing adverse traffic conditions and rerouting public transportation (while simultaneously sending reports to waiting passengers advising them of revised arrival times and alternative pickup points).

***smart entertainment*.** Your video recorder predicts and records programs you will (or are likely to) want to watch.

***smart computer*.** Your computer manages local and Internet-based data storage to optimize retrieval time and use of local resources. Your computer uses spare execution cycles to contribute to solving computation-intensive problems throughout the world.

Each of these applications seems to be a noble activity that at least some users would embrace. But with your security hat on, you might detect a negative side to each, for example:

- ***Loss of privacy.***
- ***Loss of control.***
- ***Potential for subversion.***
- ***Mistaken identification.***
- ***Uncontrolled access.***

What is IoT security?

The segment of the internet of things that focuses on protecting the IoT system, servers, networks and physical devices is known as IoT security. It involves using methods, tools and strategies that benefit the users and protect the IoT ecosystems. The growing concern of Iot exposure to large numbers of malware attacks gave rise to IoT security. Some of the methods IT developers use to secure IoT systems are PKI, end-to-end encryption and API security.

Why IoT security?

Through IoT security, it is possible to prevent all types of attacks and vulnerabilities in any IoT system. Developers use technologies, methods and devices to prevent security breaches. Cryptography technologies fight communication attacks whereas security services battle lifecycle attacks. Developers make use of isolation measure to prevent software attacks and tamper mitigation methods to avoid physical device attacks.

Security issues in the IoT system
(Threats and risks)

1. Vulnerabilities
2. Malware
3. Cyberattack escalation
4. Information theft and unknown exposure
5. Device mismanagement and misconfiguration

How to protect your IoT devices?

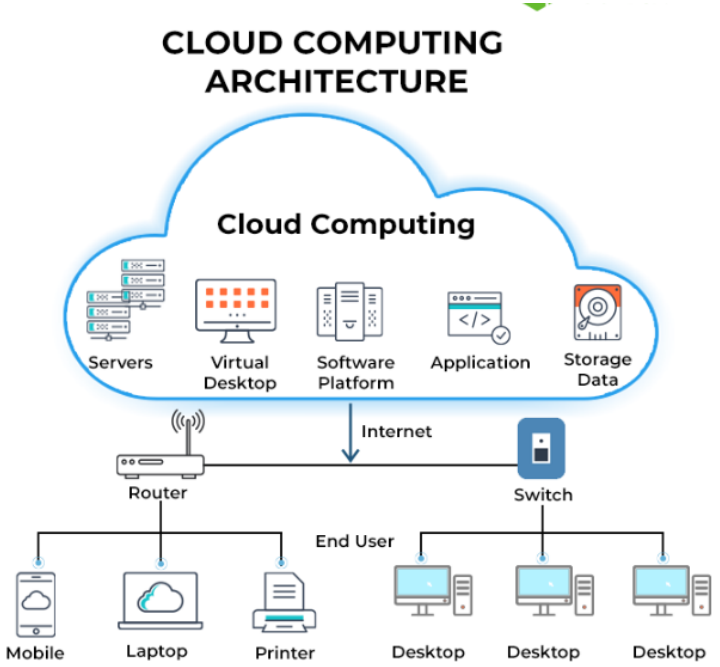
There are some ways that can guarantee security. But no device is completely secure. We can only try to obtain maximum security.

Given below are few of the many ways to ensure your IoT device is safe.

1. Make sure that you are fully aware about the nationality of each of the IoT devices you own.
2. Make sure the devices are up to date and discard any old devices.
3. Always use strong and unique passwords.
4. Before buying an IoT device, make sure to read about the manufacturer's reputation and security.
5. Most of the attacks can be avoided by introducing protection in the beginning stages of production.
6. In multiple networking systems, PKI ensures security between client-server devices.
7. Securing the API is crucial in any IoT system since API is the backbone of an IoT solution.
8. Invest in a good antivirus software. Antivirus software takes care of all the complicated processes and makes sure your system is out of danger.

What is cloud computing?

Cloud computing is the delivery of computing resources — including storage, processing power, databases, networking, analytics, artificial intelligence, and software applications — over the internet (the cloud). By outsourcing these resources, companies can access the computational assets they need, when they need them, without needing to purchase and maintain a physical, on-premise IT infrastructure. This provides flexible resources, faster innovation, and economies of scale. For many companies, a cloud migration is directly related to data and IT modernization.

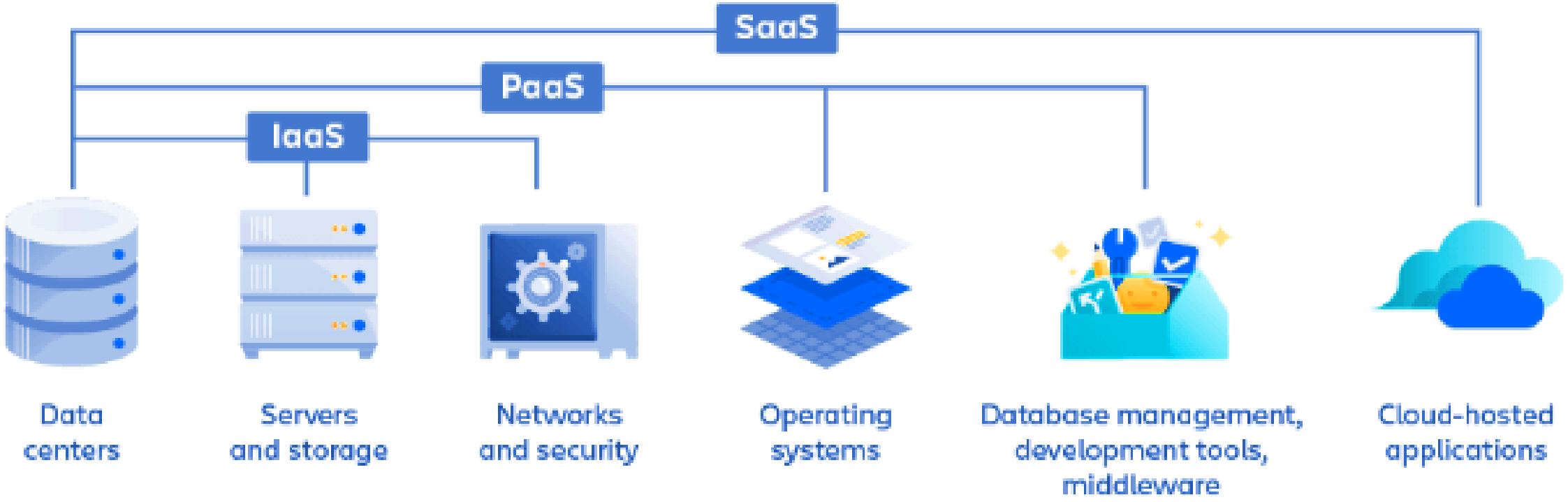


Characteristics of cloud computing

Before cloud computing, organizations purchased and maintained an on-premise IT infrastructure. Though cost-savings drove much of the initial shift to the cloud, many organizations find that public, private, or a hybrid cloud infrastructure offers a host of benefits.
The following is a list of characteristics that define cloud computing.

- **On-demand self service**
- **Broad network access**
- **Resource pooling**
- **Rapid elasticity**
- **Measured service**

Cloud computing services



The dynamic properties of cloud computing sets the foundation for novel higher-level services. These services can help not only complement, but often provide necessary services for agile and DevOps teams.

1. **Infrastructure as a service (IaaS)** is a foundational cloud service layer that allows organizations to rent IT infrastructure -- servers, storage, networks, operating systems -- from a cloud provider. IaaS lets users reserve and provision the resources they need out of raw physical server warehouses. In addition, IaaS lets users reserve pre-configured machines for specialized tasks like load balancers, databases, email servers, distributed queues.

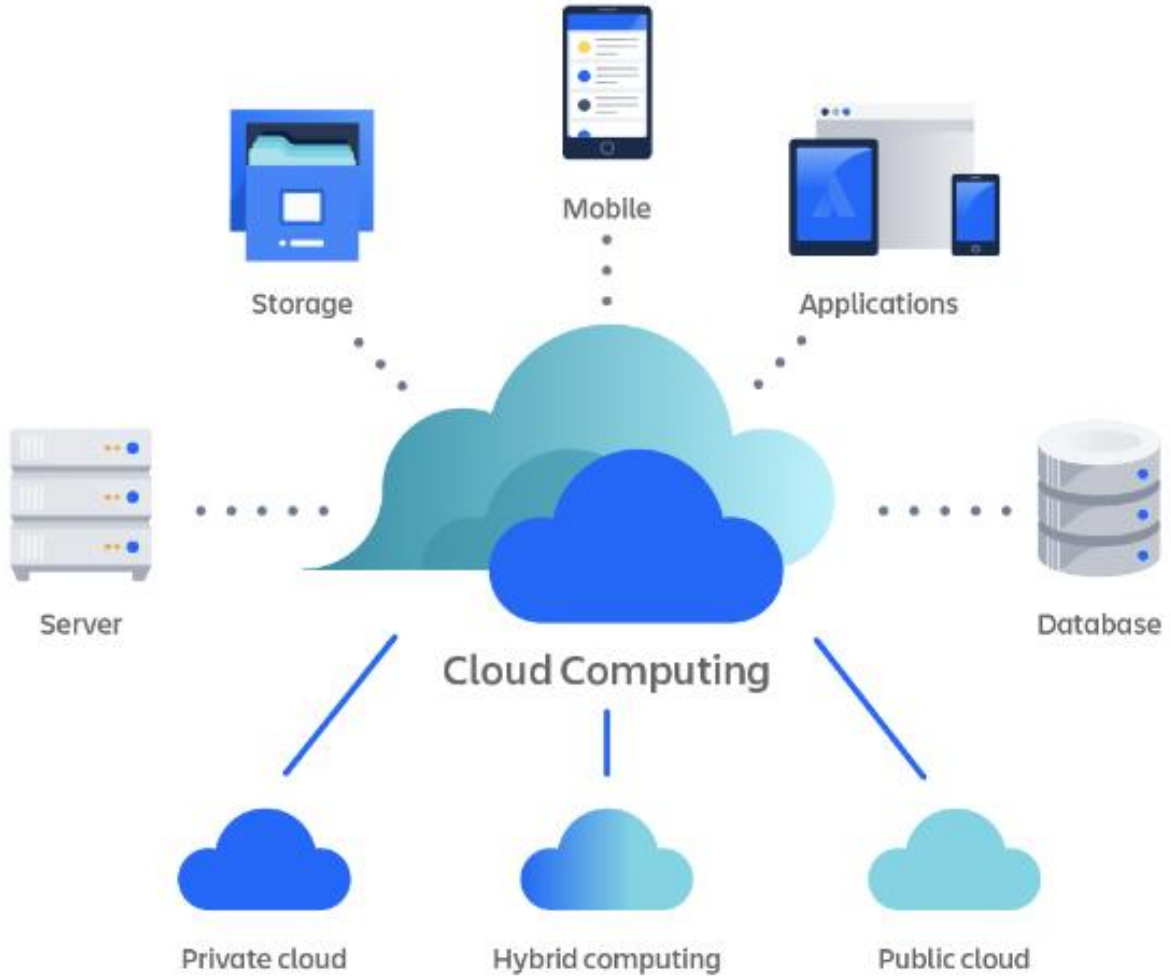
2. Platform as a service (PaaS) is a cloud infrastructure built on IaaS that provides resources to build user-level tools and applications. It provides the underlying infrastructure including compute, network, and storage resources, as well as development tools, database management systems, and middleware.

3. Software as a service (SaaS) delivers software applications over the internet, on-demand and typically by subscription. The cloud providers host and manage the application, addressing software upgrades and security patching as needed. Examples of SaaS are webmail applications, analytics tools, monitoring tools, chat applications, and more.

4. Function as a service (FaaS) is a cloud computing service that offers a platform where customers can develop, run, and manage applications. This alleviates the need for developers to build and maintain the infrastructure needed to develop and launch an app. Cloud providers offer cloud resources, execute a block of code, return the result, and then destroy the resources that were used.

Types of cloud deployments

There are three primary types of cloud deployments. Each has unique benefits and organizations often benefit from using more than one.



Public cloud

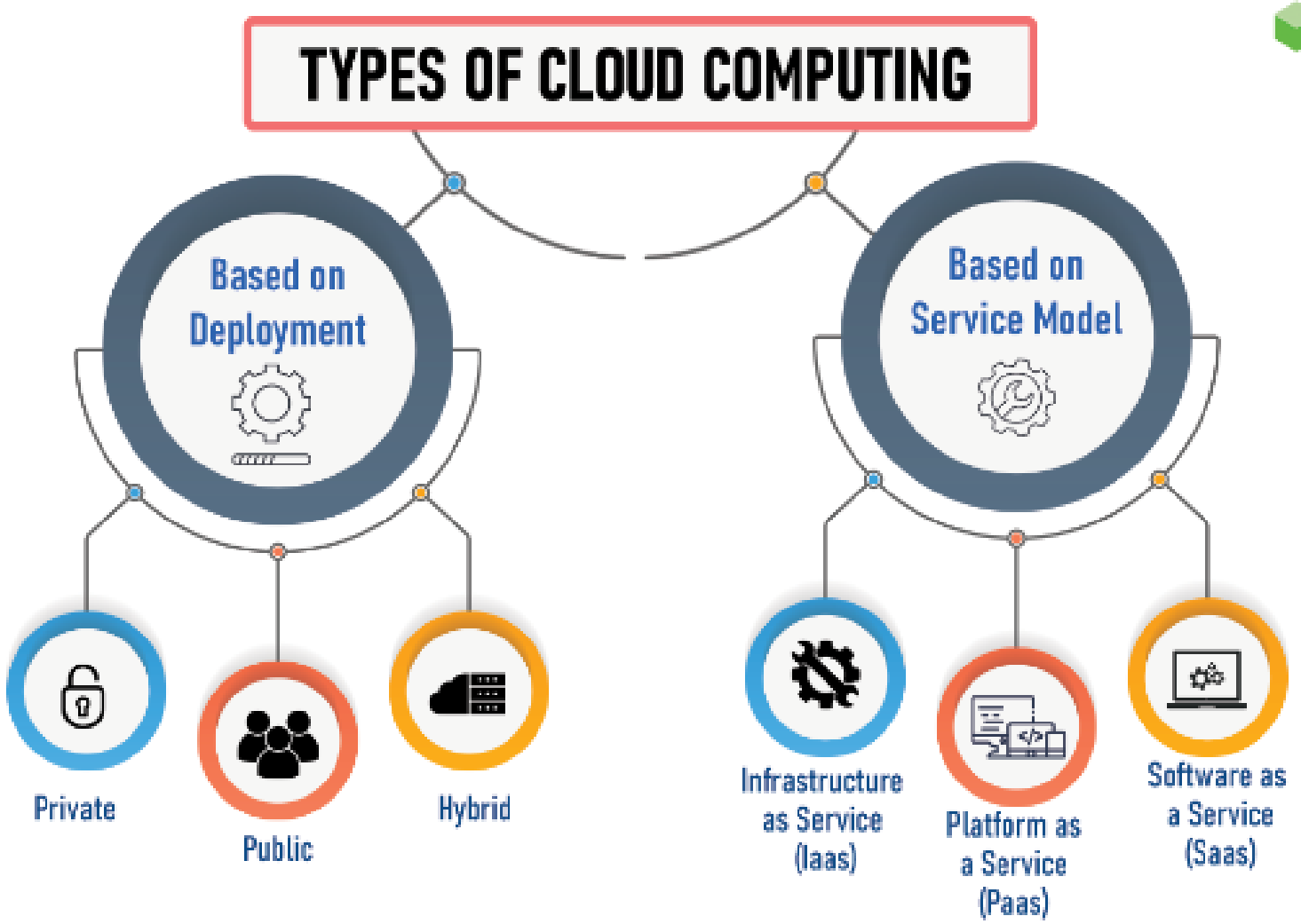
Public clouds deliver computing resources -- servers, storage, applications, etc. -- over the internet from a cloud service provider, such as AWS and Microsoft Azure. Cloud providers own and operate all hardware, software, and other supporting infrastructure.

Private cloud

A private cloud is computing resources dedicated exclusively to an organization. It can be physically located at an organization's on-site data center, or hosted by a cloud provider. A private cloud delivers a higher level of security and privacy than public clouds by offering dedicated resources to companies.

Hybrid cloud

Hybrid clouds are a combination of private and public clouds (for example, IBM Hybrid Cloud, powered by Red Hat), connected together with technology that enables data and application to work together. Sensitive services and applications can be kept in the secure private cloud while publicly-accessible web servers and customer-facing endpoints can live in the public cloud.



Types of Cloud Computing

Benefits of cloud computing

The unique properties of cloud infrastructures provide several novel technical and business benefits. The following are the key cloud computing benefits for agile teams.

1. **Reduced cost**
2. **Increased scalability**
3. **Better performance**
4. **Improved execution speed**
5. **Increased security**
6. **Continuous integration and delivery**
7. **Comprehensive monitoring and incident management**

Security Threats of Cloud Computing

1. **Abuse and Nefarious Use of Cloud Computing**
2. **Insecure application programming interfaces**
3. **Malicious insiders**
4. **Shared technology vulnerabilities**
5. **Data loss/leakage**
6. **Account ,Service & Traffic hijacking**
7. **Unknown risk profile**

SECURITY IN CLOUD COMPUTING

1. **Infrastructure Security:**
2. **Data Security and Storage**
3. **Identity and access**
4. **Security management**
5. **Privacy**
6. **Audit and Compliance**
7. **Security-as-a Service**

EXISTING SOLUTIONS FOR SECURITY THREATS

1. Mirage image management system.

2. Client based privacy manager.

3. Transparent cloud protection system.

4. Secure and efficient access to outsourced data.

Cyber Warfare

Is the use of cyber attacks against an enemy state, causing comparable harm to actual warfare and/or disrupting vital computer systems. Some intended outcomes could be espionage, sabotage, propaganda, manipulation or economic warfare.
An alternative view is that it is a suitable label for cyber attacks which cause physical damage to people and objects in the real world.

A global domain within the information environment consisting of the interdependent network of information technology infrastructures, including the Internet, telecommunications networks, computer systems, and embedded processors and controllers.

Cyber warfare is larger than cyber mischief, cybercrime, cyber espionage, cyber terrorism, or cyber attack. “Warfare” is a term typically reserved for active conflict between nation states.

Cyberwar

War conducted in and from computers and the networks connecting them, waged by states or their proxies against other states. Cyberwar is usually waged against government and military networks in order to disrupt, destroy, or deny their use. Cyberwar should not be confused with the terrorist use of cyberspace or with cyberespionage or cybercrime. Even though similar tactics are used in all four types of activities, it is a misinterpretation to define them all as cyberwar. Some states that have engaged in cyberwar may also have engaged in disruptive activities such as cyberespionage, but such activities in themselves do not constitute cyberwar.

Cybercrime, also known as **computer crime**, Any use of a computer as an instrument to further illegal ends, such as committing fraud, trafficking in child pornography and intellectual property, stealing identities, or violating privacy. Cybercrime, especially through the Internet, has grown in importance as the computer has become central to commerce, entertainment, and government. The international nature of cybercrimes has led to international cyber laws.

Attacks can be made against the **syntactic layer** by using cyber weapons that destroy, interfere with, corrupt, monitor, or otherwise damage the software operating the computer systems. Such weapons include malware, malicious software such as viruses, trojans, spyware, and worms that can introduce corrupted code into existing software, causing a computer to perform actions or processes unintended by its operator. Other cyber weapons include distributed denial-of-service, or DDoS, attacks, in which attackers, using malware, hijack alarge number of computers to create so-called botnets, groups of “zombie”computers that then attack other targeted computers, preventing their proper function.

Semantic cyberattacks, also known as social engineering, manipulate human users' perceptions and interpretations of computer-generated data in order to obtain valuable information (such as passwords, financial details, and classified government information) from the users through fraudulent means. Social-engineering techniques include phishing—in which attackers send seemingly innocuous e-mails to targeted users, inviting them to divulge protected information for apparently legitimate purposes—and baiting, in which malware-infected software is left in a public place in the hope that a target user will find and install it, thus compromising the entire computer system.