

# COMPUTER SECURITY

## INTRODUCTION



# Text Book

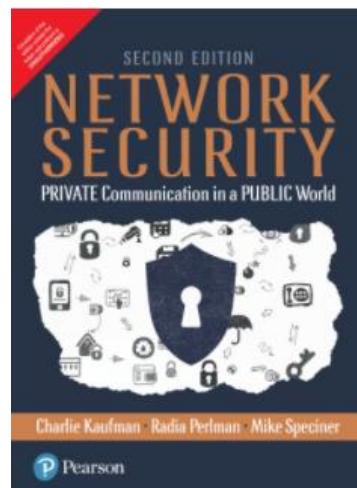
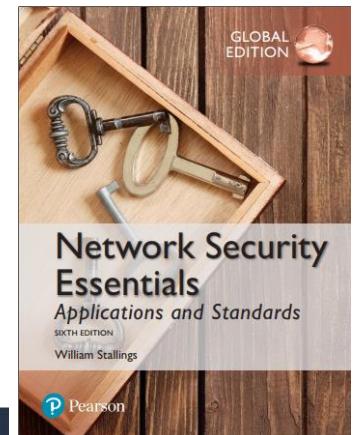
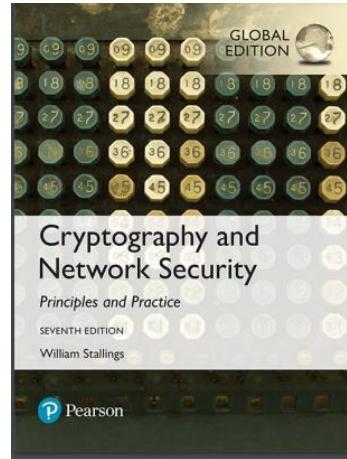
## Textbook

### Cryptography and Network Security: Principles and Practice, Global Edition

## Ref

Network Security: Private Communication in a Public World  
Charlie Kaufman, Radia Perlman,  
Mike Speciner, Michael Speciner,  
Prentice Hall

Network Security Essentials Applications and Standards- William Stalling-Applications and Standards



# Learning Objectives

- Describe the key security requirements of **confidentiality**, **integrity**, and **availability**
- Discuss the types of security threats and attacks
- Summarize the functional requirements for computer security
- Describe the X.800 security architecture for OSI
- Cryptography applications

# Introduction

- Humans have been in conflict since prehistoric times.
- In The Art of War 1, Sun Tzu says “**The entire art of war is based on deception**”.

# History – Pre-Renaissance

- Paper was first introduced to Europe by Crusaders around 1200 A.D.
- Information was stored using tokens, tablets, knotted strings, notched sticks or handwritten parchment.

# History – Pre-Renaissance

- Perhaps the best illustration of the importance of information security to ancient societies is the relatively **advanced state of cryptography and steganography**, technologies devoted to keeping information secret, when compared to the other technologies available for computing and communication.
- Cryptography is the science of writing in codes that are hard to decipher, and steganography is the art of hiding information to make it hard to detect.
- In Rome, Julius Caesar used a simple substitution cipher to hide information from his enemies
- During Abbasid caliphate [750 A.D], instruction to administrators explained how to use **substitution ciphers**. Muslim texts of that era explain how frequency counting can easily break substitution ciphers.

# History – Pre-Renaissance

- Even more effort was devoted to finding tools for steganography. Recorded ancient approaches for information hiding include
  - ❖ Texts written on shaved skulls that were later covered by letting hair grow (Greece),
  - ❖ Invisible inks (Rome),
  - ❖ Pin pricks placed above letters in a text to indicate letters in a secret message (Greece),
  - ❖ Messages hidden in images and hieroglyphics (Egypt), and
  - ❖ Messages written on a thin sheet, rolled into a wax ball, and hidden or swallowed (China).

# History – Renaissance to World War I

- The Renaissance was a period of renewed intellectual activity in Europe starting in the 14th century.
  - ❖ Napier's bones used for multiplication, division, and finding square roots [1617]
  - ❖ Blaise Pascal produced around 50 mechanical calculators starting in 1642.
  - ❖ In 1800s Herman Hollerith developed a punch card tabulation system
  - ❖ Charles Babbage produced the first programmable machines, the difference and analytical engines, in the mid-1800s
  - ❖ Poly-alphabetic solution ciphers

# History –World War II

- ❖ During WWII, German cryptographers secured their communications using the **Enigma encryption**
- ❖ The Japanese had a cipher device that was their equivalent to Enigma. It was code-named Purple.

# History –Cold War

- ❖ The new availability of computers allowed the science of cryptography to make major advances.
- ❖ In the United States, the National Security Agency (NSA) is in charge of cryptographic research for the United States government
- ❖ Cryptography also became available for civilian use.
- ❖ In the 1970s the U.S. National Bureau of Standards<sup>4</sup>, with NSA approval, agreed to the release of the Data Encryption Standard (DES) with a 56-bit key-space.
- ❖ The development of public key cryptography is a major step towards solving the problem of key distribution

# History –Cold War

- ❖ The new availability of computers allowed the science of cryptography to make major advances.
- ❖ In the United States, the National Security Agency (NSA) is in charge of cryptographic research for the United States government
- ❖ Cryptography also became available for civilian use.
- ❖ In the 1970s the U.S. National Bureau of Standards<sup>4</sup>, with NSA approval, agreed to the release of the Data Encryption Standard (DES) with a 56-bit key-space.
- ❖ The development of public key cryptography is a major step towards solving the problem of key distribution

# History – The Modern Era

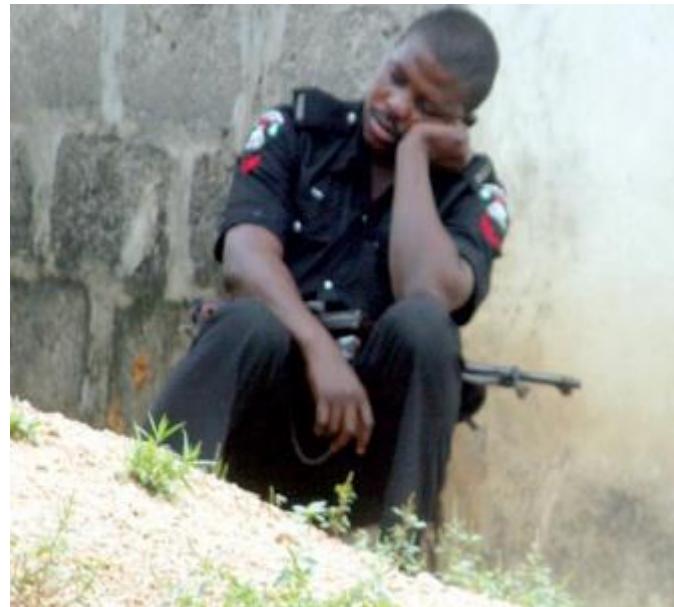
- Computers!
- Examples
  - Lucifer
  - Rijndael
  - RSA
  - ElGamal

# Organized Crime and Botnets

- ❖ Current cybercrime technologies and techniques include
  1. botnets – networks of compromised machines,
  2. spam – unsolicited e-mail advertisements, usually fraudulent,
  3. phishing – attempts to fraudulently collect sensitive personal information,
  4. pharming – redirecting web traffic to a fraudulent site,
  5. identity theft and identity fraud – fraudulent use of personal information,
  6. cross site scripting – inserting script commands into another's website,
  7. cross site request forgery – tricking a user's browser to making requests on another party's website, • underground forums, and
  8. money laundering.

# What is Security?

Security according to two boys of 10 years old-



# What is Security?

Security According to Junior High School ICT teacher-



# What is Security?

There is no clear cut definition.



# What is Security?

Security is a process, not an end state



# Computer Security Concepts

- Before the widespread use of data processing equipment, the security of **information** valuable to an organization was provided primarily **by physical and administrative means**
- With the introduction of the computer, **the need for automated tools** for **protecting** files and other information stored on the computer became evident
- Another major change that affected security is the introduction of **distributed systems** and the use of **networks and communications** facilities for carrying data between terminal user and computer and between computers

# Computer and Internet Security

- **Computer security**
  - The generic name for the **collection of tools** designed to protect data and to thwart hackers
- **Internet security**
  - Consists of measures to, **deter, prevent, detect and correct** security violations that involve the **transmission of information**

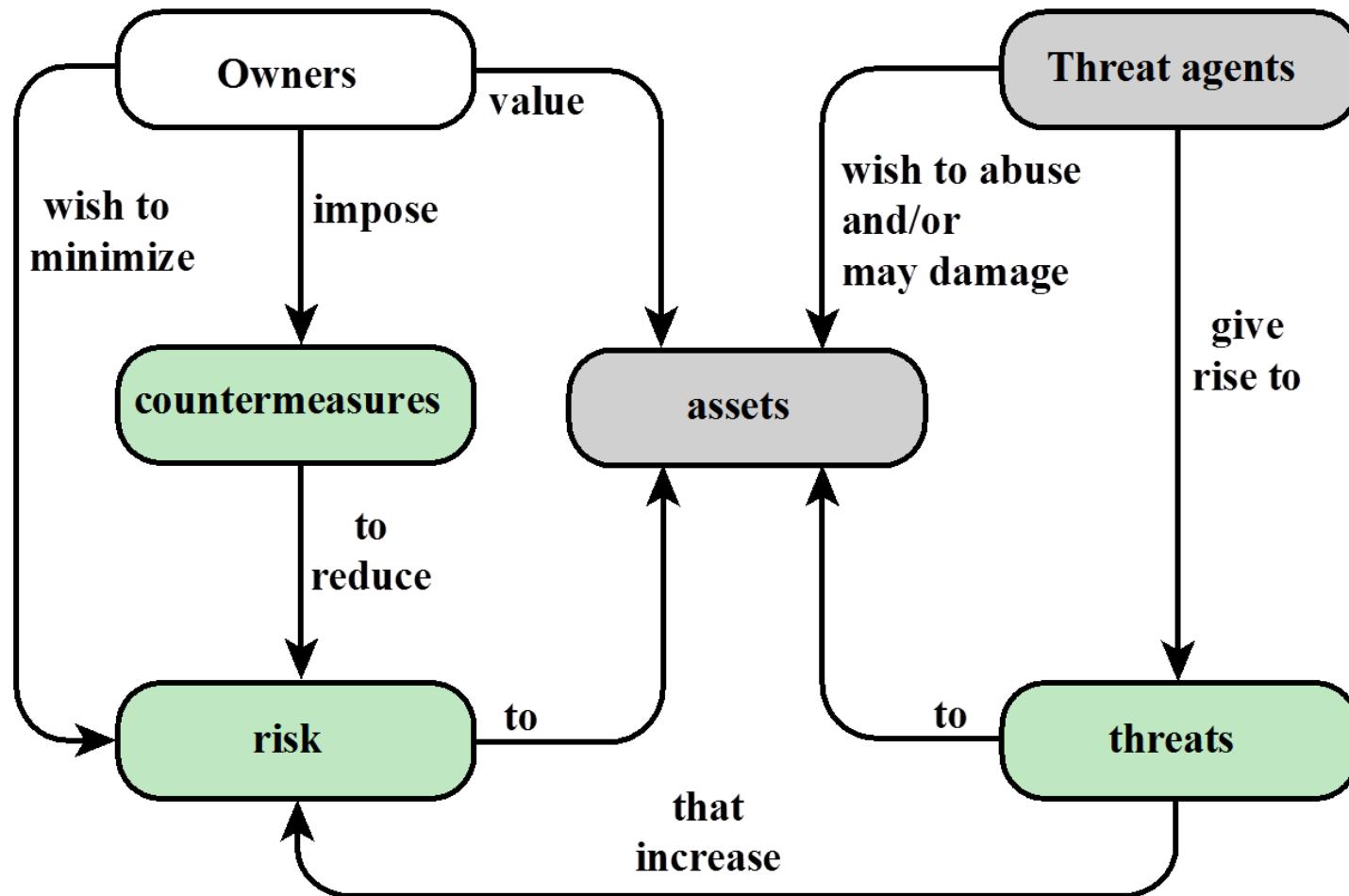
# What security is about in general?

- Security is about protection of assets
  - D. Gollmann, Computer Security, Wiley

## Need

- Prevention
  - take measures that prevent your assets from being damaged (or stolen)
- Detection
  - take measures so that you can detect when, how, and by whom an asset has been damaged
- Reaction
  - take measures so that you can recover your assets

# Relationships among the security Concepts



# Real world example

- Prevention
  - locks at doors, window bars, secure the walls around the property, hire a guard
- Detection
  - missing items, burglar alarms, closed circuit TV
- Reaction
  - attack on burglar (not recommended ☺), call the police, replace stolen items, make an insurance claim

# Information security in past & present

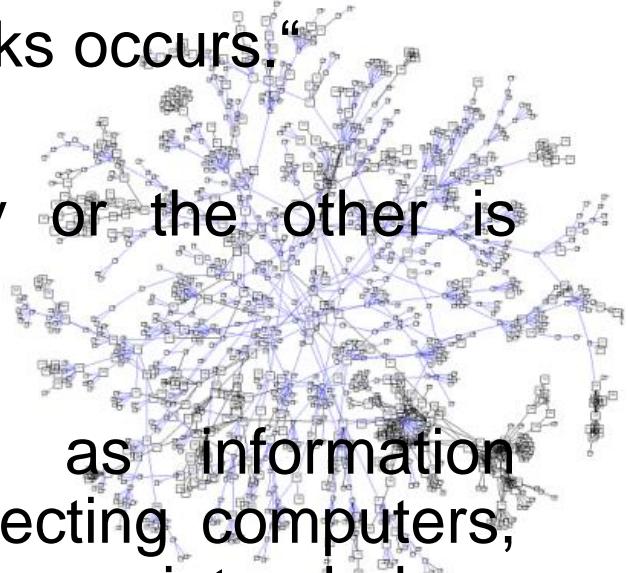
- Traditional Information Security
  - keep the cabinets locked
  - put them in a secure room
  - human guards
  - electronic surveillance systems
  - in general: physical and administrative mechanisms
- Modern World
  - Data are in computers
  - Computers are interconnected

**Data/ Computer and Network Security->  
Cyber Security**

# Definition

**Cyber space:** Cyberspace is "the environment in which communication over computer networks occurs."

And almost everybody in one way or the other is connected to it



**Cyber security:** also referred to as information technology security, focuses on protecting computers, networks, programs and data from unintended or unauthorized access, change or destruction.

<https://www.itu.int/en/ITU-D/Cybersecurity/Documents/Introduction%20to%20the%20Concept%20of%20IT%20Security.pdf>

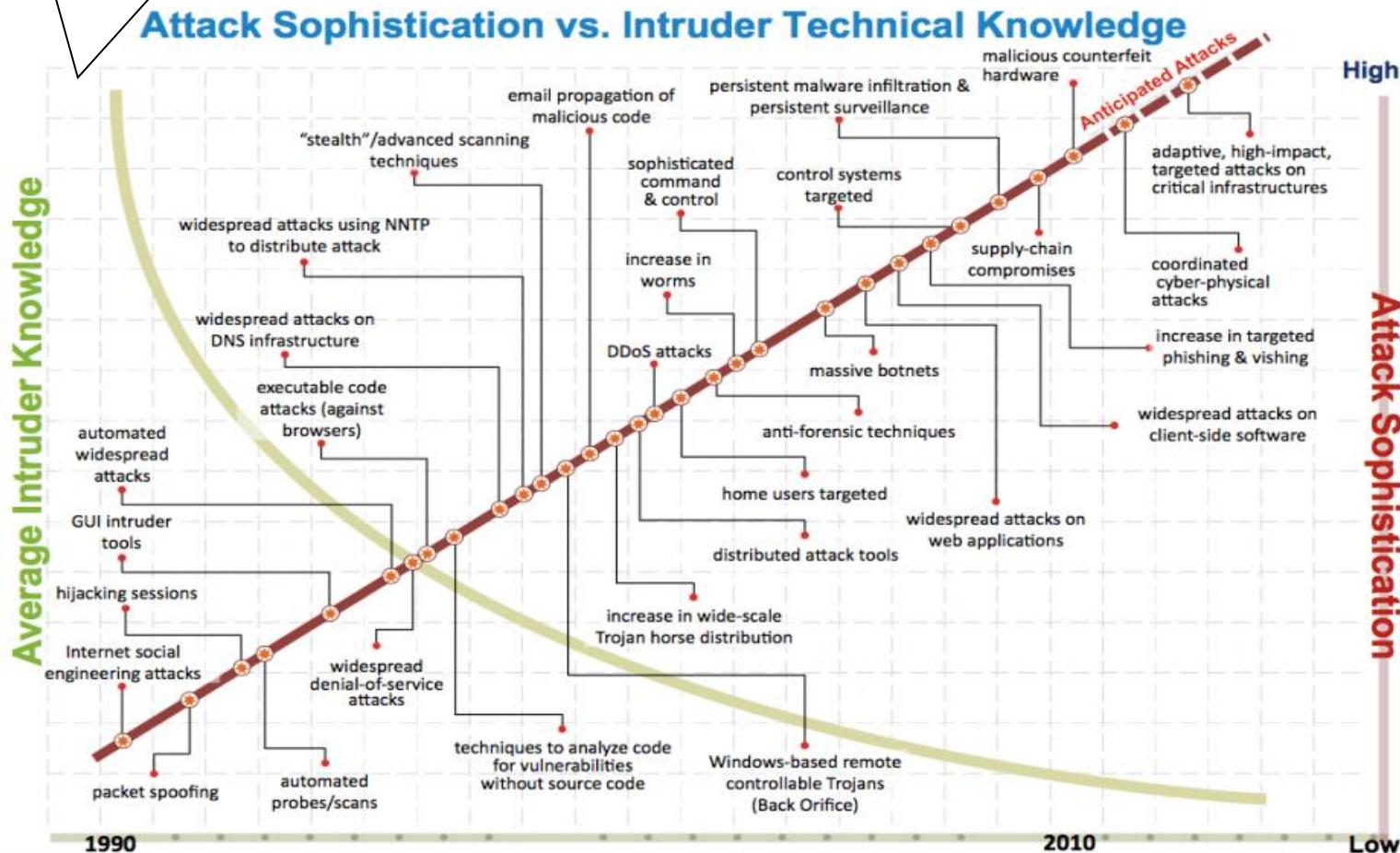
# Definition

Four domains of cybersecurity: a risk-based systems approach to cyber decisions

- <https://link.springer.com/article/10.1007/s10669-013-9484-z>

# Security Trends

Skill and knowledge required to mount an attack

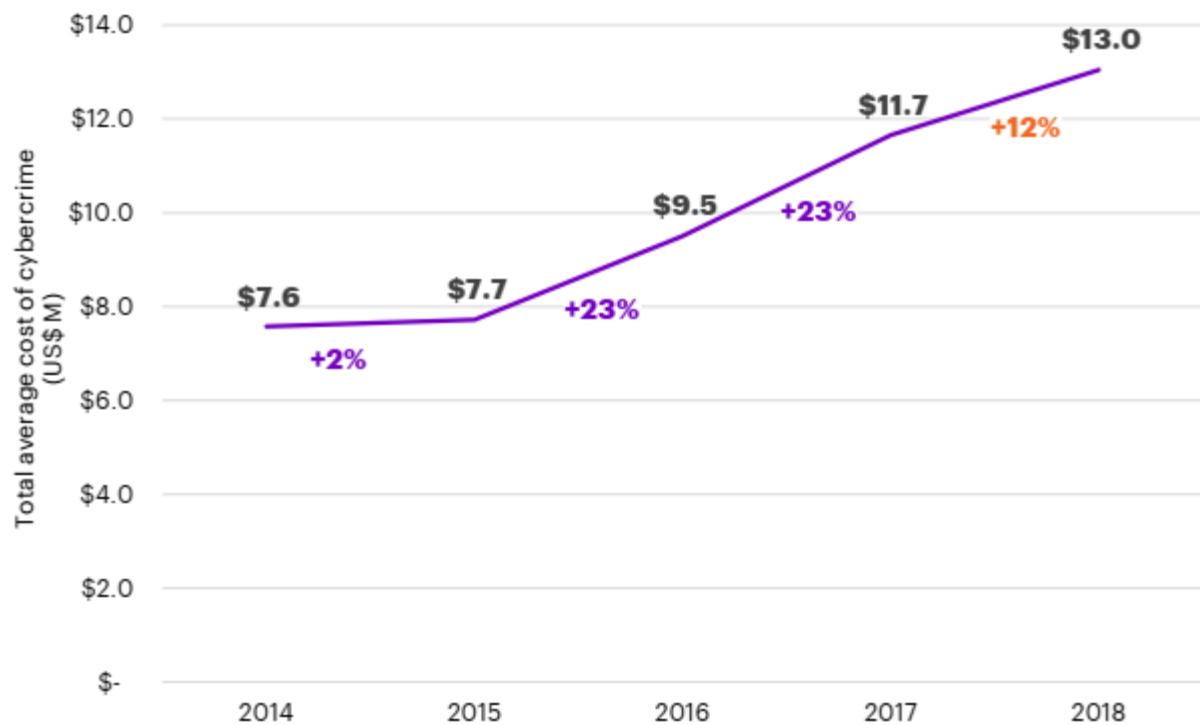


Software Engineering Institute

Carnegie Mellon

Trusted Computing in Embedded Systems  
Workshop November 2010  
© 2010 Carnegie Mellon University

# The global average cost of cyber crime/attacks

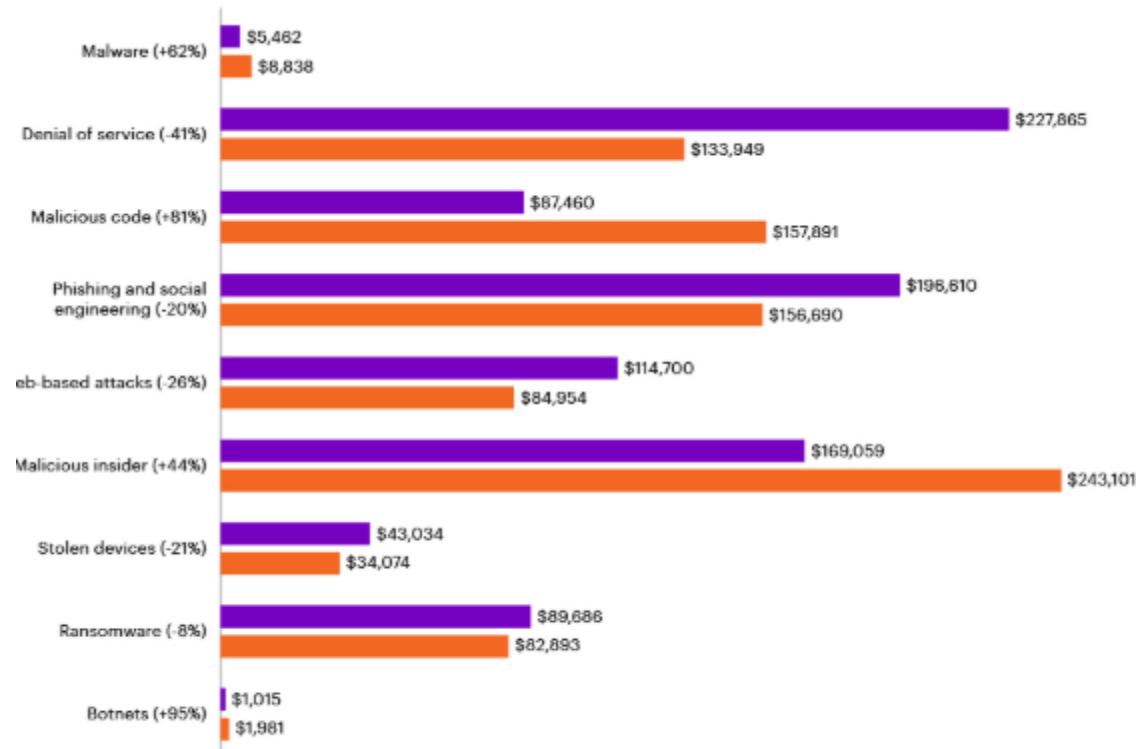


2018 Cost of  
Cyber Crime  
Study by  
Accenture\*

Steeper  
increasing trend  
in the recent  
years

<https://www.accenture.com/us-en/insights/security/cost-cybercrime-study>

# Types of cyber attacks experienced



2018 Cost of  
Cyber Crime  
Study by  
Accenture\*

*Average annual cost of cybercrime by type of attack for financial services firms*

■ 2017 ■ 2018

<https://www.accenture.com/us-en/insights/security/cost-cybercrime-study>

# Computer Security

- The NIST *Computer Security Handbook* defines the term computer security as:

**NIST: National Institute of Standards and Technology**

This definition introduces **three** key objectives that are at the heart of computer security.

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

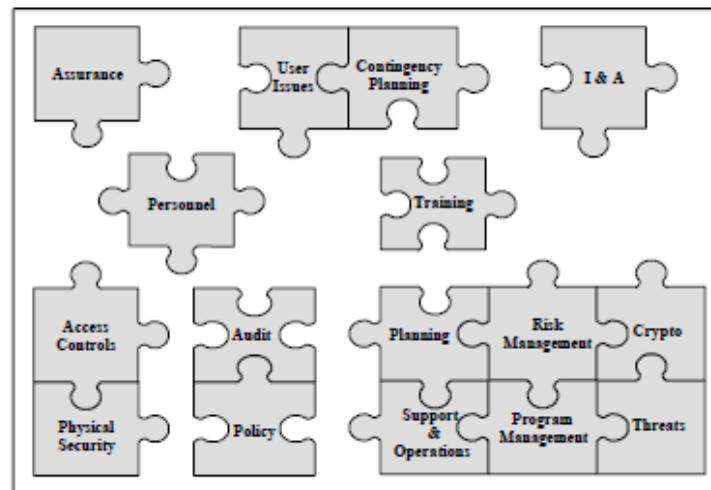
# NIST



National Institute of Standards and Technology  
Technology Administration  
U.S. Department of Commerce

## An Introduction to Computer Security: The NIST Handbook

Special Publication 800-12



# Computer Security Objectives

## Confidentiality

- **Data confidentiality**
  - Assures that private or confidential information is not made available or disclosed **to unauthorized individuals**
- **Privacy**
  - Assures that **individuals control** or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed

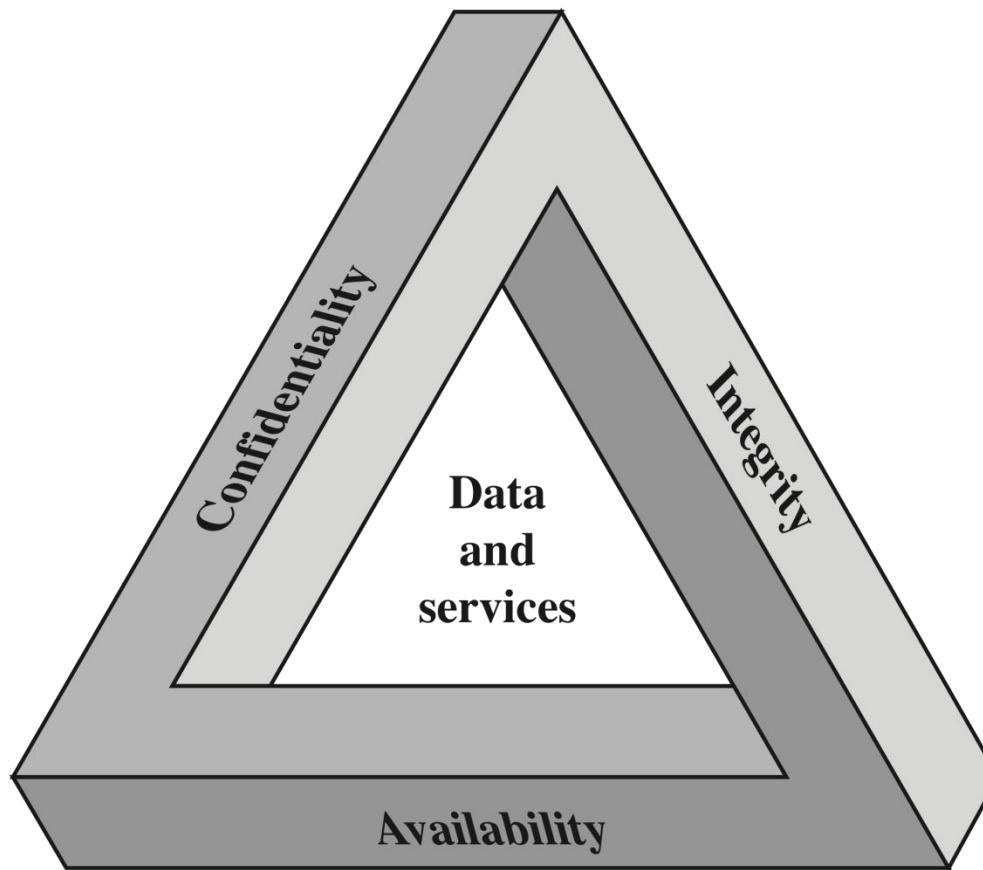
## Availability

- Assures that systems work promptly and **service is not denied** to authorized users

## Integrity

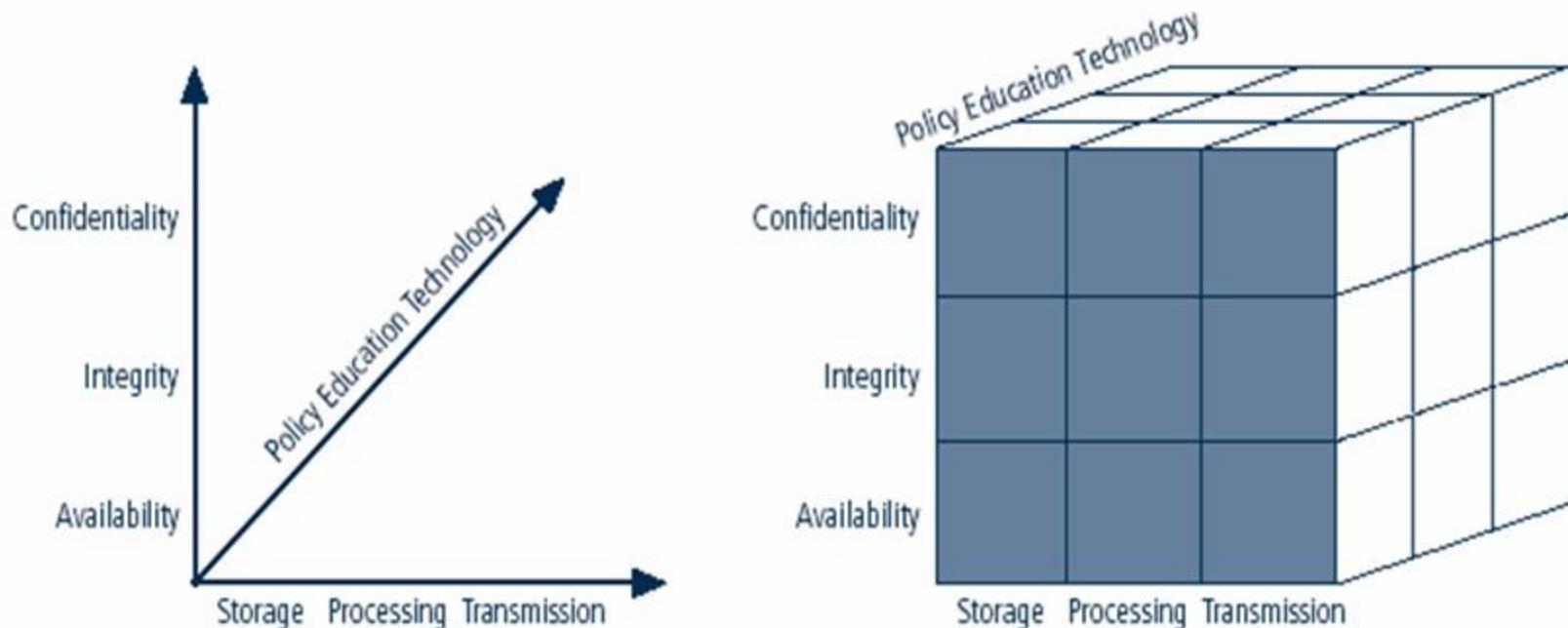
- **Data integrity**
  - Assures that information and programs are changed only in a specified and authorized manner
- **System integrity**
  - Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system

# CIA Triad



The Security Requirements Triad

# NSTISSC Security Model



NSTISSC Security Model

'National Security Telecommunications & Information systems security committee' document.

It is now called the National Training Standard for Information security professionals.



**Confidentiality** is probably the most common aspect of information security. We need to protect our confidential information. An organization needs to guard against those malicious actions that endanger the confidentiality of its information.

**Integrity** : Information needs to be changed constantly. Integrity means that changes need to be done only by authorized entities and through authorized mechanisms.

**Availability** : The information created and stored by an organization needs to be available to authorized entities. Information needs to be constantly changed, which means it must be accessible to authorized entities.

# Possible additional concepts:

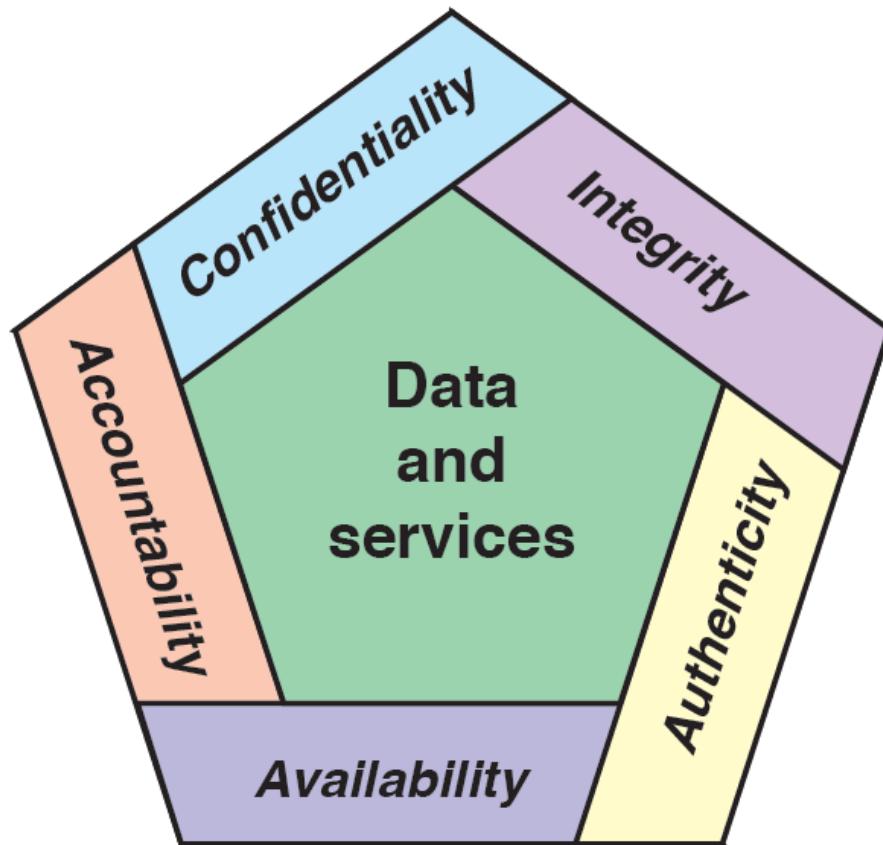
## Authenticity

- Verifying that users are who they say they are and that each input arriving at the system came from a **trusted** source

## Accountability

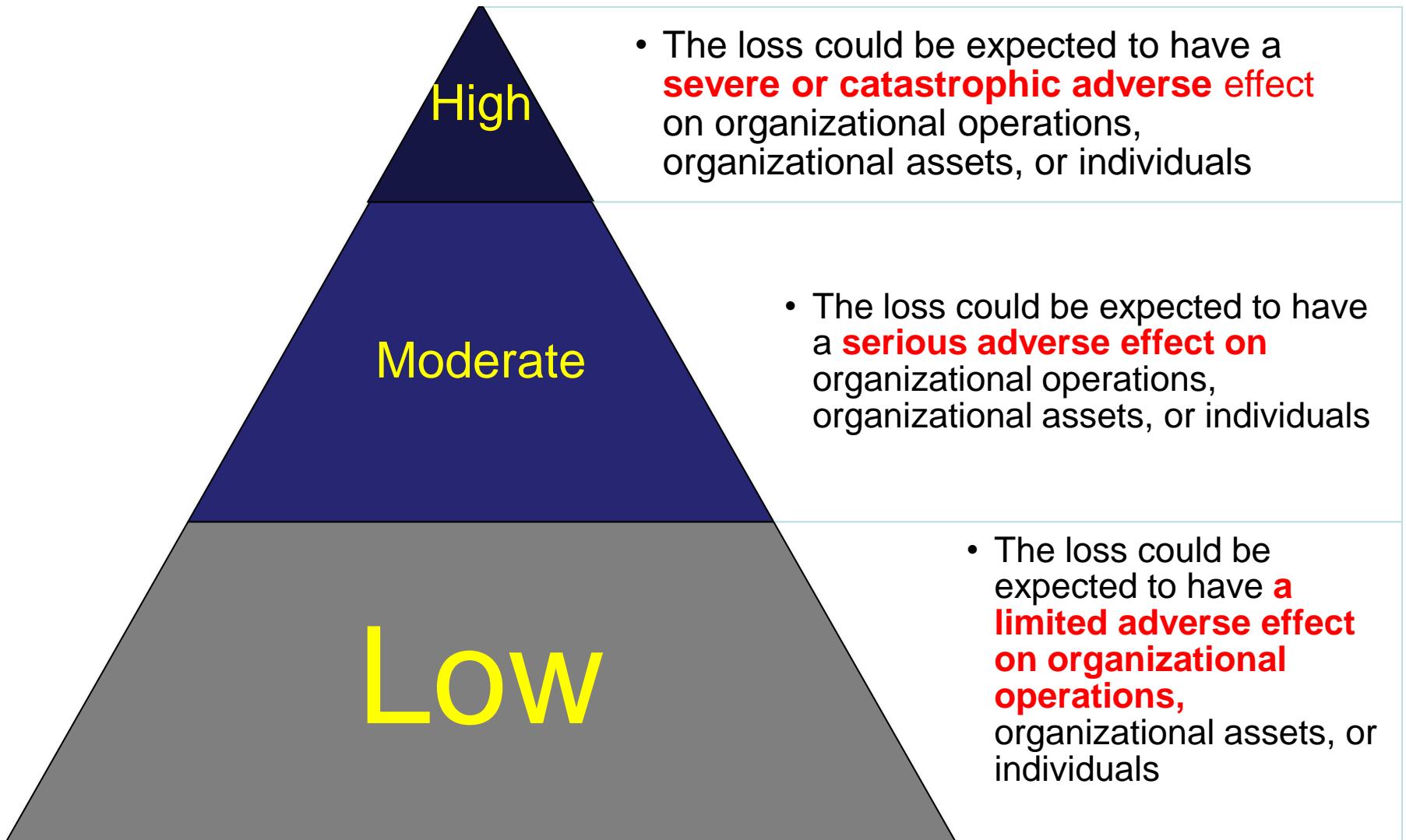
- The security goal that generates the requirement for actions of an entity to be **traced uniquely** to that entity

# Possible additional concepts:



# Breach of Security

## 3 Levels of Impact (These levels are defined in FIPS 199)



# Examples of Security Requirements

## Confidentiality

**Student grade information** is an asset whose confidentiality is considered to be highly important by students

Regulated by the Family Educational Rights and Privacy Act (FERPA)

## Integrity (consistency)

**Patient information** stored in a database – inaccurate information could result in serious harm or death to a patient and expose the hospital to massive liability

A **Web site that offers a forum to registered users** to discuss some specific topic would be assigned a moderate level of integrity

An example of a low-integrity requirement is an anonymous **online poll**

## Availability

The **more critical a component or service**, the higher the level of availability required

A moderate availability requirement is a **public Web site for a university**

An **online telephone directory lookup** application would be classified as a low-availability requirement

# Computer Security Challenges

- Security is **not simple**
- **Potential attacks** on the security features need **to be considered**
- Procedures used to provide particular services are often **counter-intuitive**
- It is necessary to decide **where to use** the various security mechanisms
- Requires **constant monitoring**
- Is too often an **afterthought**
- Security mechanisms typically involve **more than a particular algorithm or protocol**
- Security is essentially a **battle of wits** between a perpetrator and the designer
- **Little benefit** from security investment **is perceived** until a security failure occurs
- Strong security is often viewed as **an impediment** to efficient and user-friendly operation



# The OSI Security Architecture

- To assess effectively the **security needs** of an organization and to evaluate and choose various **security products** and policies, the manager responsible for computer and network security needs some systematic way of defining the requirements for **security** and characterizing the approaches to satisfying those requirements.
- ITU-T Recommendation X.800, **Security Architecture for OSI**, defines such a systematic approach.
- The OSI security architecture is useful to managers as a way of **organizing the task of providing security**.



## ITU-T

The **ITU Telecommunication Standardization Sector (ITU-T)** is one of the three sectors (divisions or units) of the International Telecommunication Union (ITU); it coordinates standards for telecommunications.

The ITU-T mission is to ensure the efficient and timely production of standards covering all fields of telecommunications on a worldwide basis, as well as defining tariff and accounting principles for international telecommunication services.

# OSI

ISO- The International Organization for Standardization (French: *Organisation internationale de normalisation*;) produced OSI (Open Systems Interconnection Reference Model, the OSI Reference Model, or even just the OSI Model)

# History of OSI

In the late 1970s, two projects began independently, with the same goal: to define a unifying standard for the architecture of networking systems.

One was administered by the International Organization for Standardization (ISO), while the other was undertaken by the International Telegraph and Telephone Consultative Committee, or CCITT (the abbreviation is from the French version of the name).

These two international standards bodies each developed a document that defined similar networking models. ISO 7498, ITU-T (formerly CCITT ) standard X.200 (1984)

# Security Services: X.800

X.800 defines a security service as a service that is provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or of data transfers.

## X.800 Recommendation:

1. provides a general description of **security services and related mechanisms**, which may be provided by the Reference Model; and
2. defines the positions within the Reference Model where the services and mechanisms may be provided.

This Recommendation extends the field of application of recommendation X.200, to cover secure communications between open systems.

# OSI Security Architecture

ITU-T Recommendation X.800, *Security Architecture for OSI* describes a systematic way of defining the requirements for security and characterizing the approaches to satisfying those requirements.

## Focus

- **Security attack**

**Any action** that **compromises** the security of information owned by an organization

- **Security mechanism**

**A process** (or a device incorporating such a process) that is designed **to detect, prevent, or recover** from a security attack

- **Security service**

**A processing** or **communication service** that **enhances the security** of the data processing systems and the **information transfers** of an organization

**Intended to counter security attacks**, and they make use of one or more security mechanisms to provide the service

# IETF and RFC

The **Internet Engineering Task Force (IETF)** (1986) develops and promotes voluntary Internet standards, in particular the standards that comprise the Internet protocol suite (TCP/IP).

A **Request for Comments (RFC)** is a publication of the Internet Engineering Task Force (IETF) and the Internet Society, the principal technical development and standards-setting bodies for the Internet.

The **Internet Society (ISoc)** is an international, non-profit organization founded in 1992 to provide leadership in Internet related standards, education, and policy.

# ATTACKS

The three goals of security, confidentiality, integrity, and availability can be threatened by security attacks.

# Threats and Attacks (RFC 4949)

## Internet Security Glossary, Version 2

This Glossary provides definitions, abbreviations, and explanations of terminology for information system security.

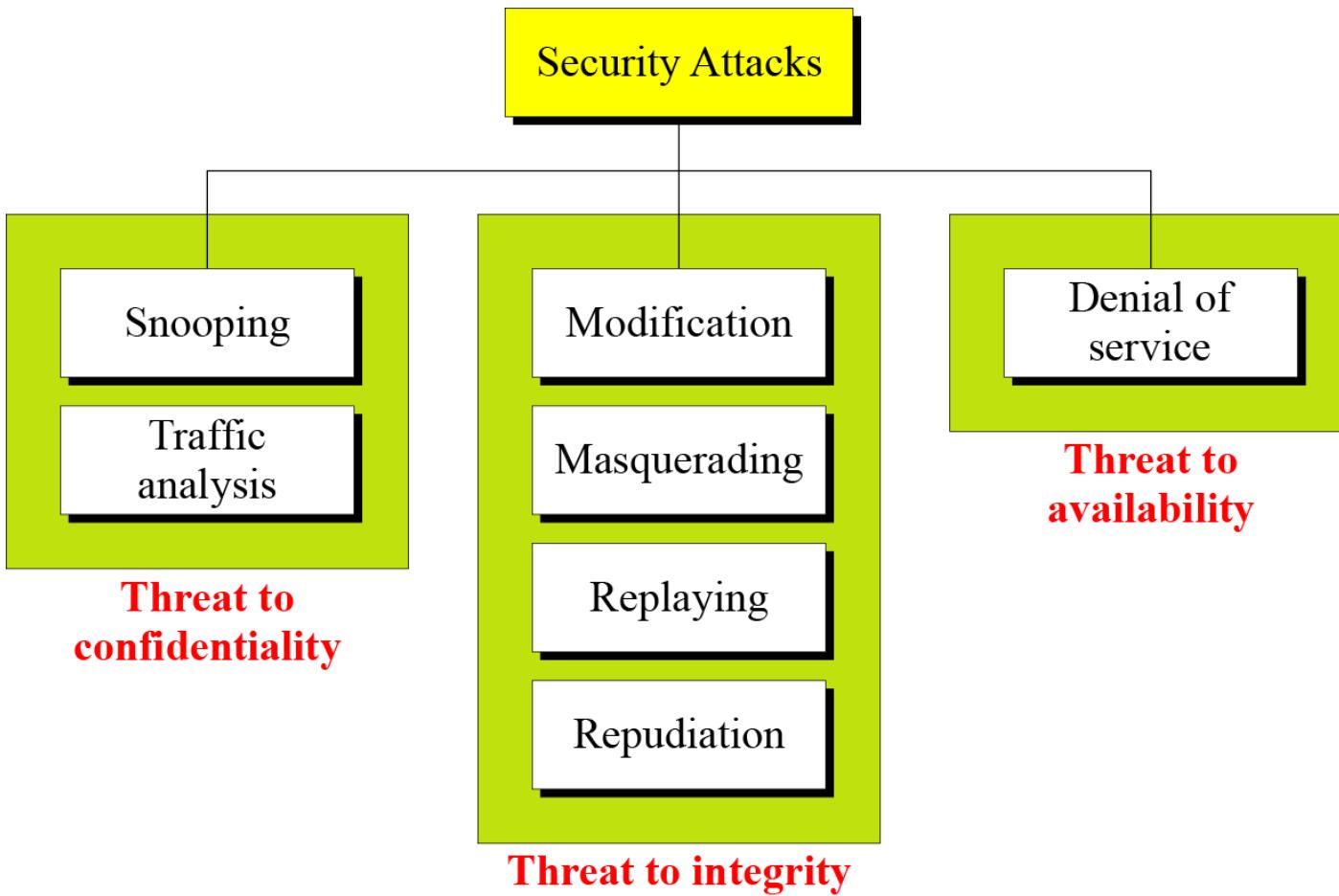


### Threat

A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. That is, a threat is a possible danger that might exploit a vulnerability.

### Attack

An assault on system security that derives from an intelligent threat; that is, an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system.



# Security Attacks

- A means of classifying security attacks, used both in X.800 and RFC 4949, is in terms of **passive attacks** and **active attacks**
- A **passive attack** attempts to learn or **make use of information** from the system but **does not affect system resources**
- An **active attack** attempts to **alter** system resources or **affect their operation**

# Passive Attacks

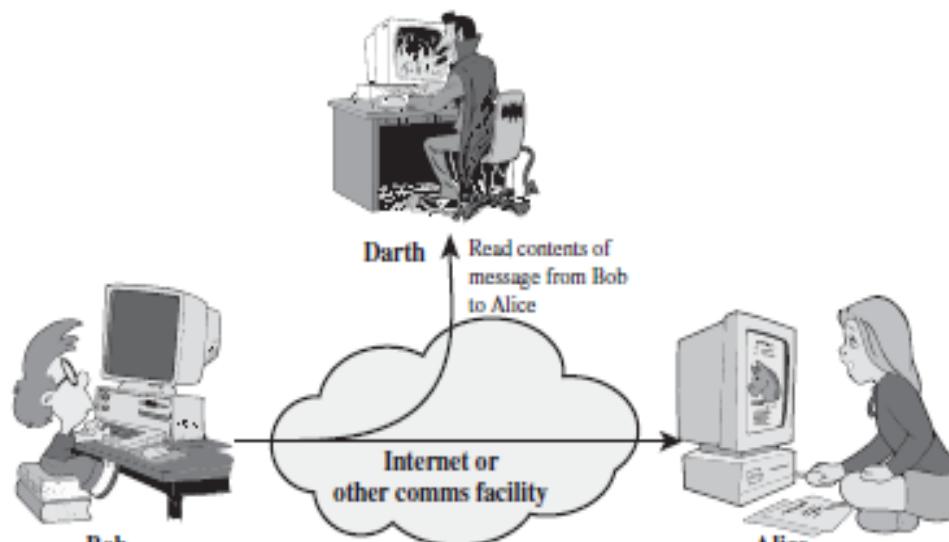
## (Two types)

- Are in the nature of **eavesdropping on, or monitoring of, transmissions**
- Goal of the opponent is to **obtain information** that is being transmitted



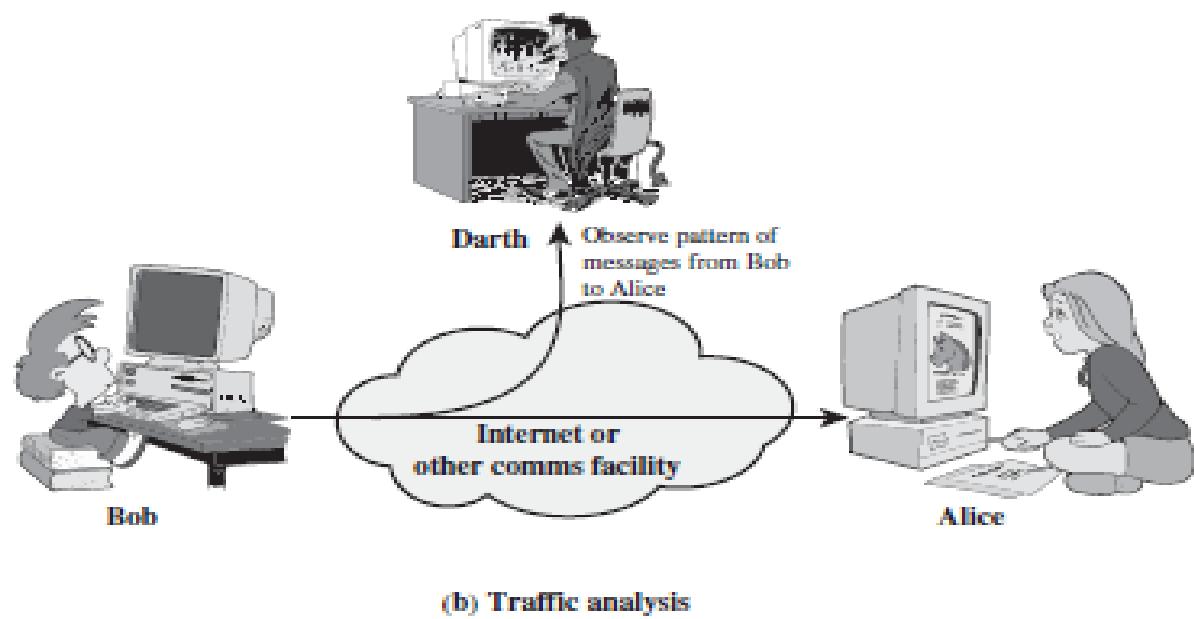
- **Two types** of passive attacks are:
  - The release of message contents
  - Traffic analysis

**Snooping** refers to unauthorized access to or interception of data.



(a) Release of message contents

Traffic analysis refers to obtaining some other type of information by monitoring online traffic.



# Active Attacks (4 types)

- **Involve some modification** of the data stream or the creation of a false stream
- **Difficult to prevent** because of the wide variety of potential physical, software, and network vulnerabilities
- **Goal** is to **detect attacks** and **to recover** from any disruption or delays caused by them



**Modification** means that the attacker intercepts the message and changes it.

**Masquerading** or spoofing happens when the attacker impersonates somebody else.

**Replaying** means the attacker obtains a copy of a message sent by a user and later tries to replay it.

**Repudiation** means that sender of the message might later deny that she has sent the message; the receiver of the message might later deny that he has received the message.

**Denial of service (DoS)** is a very common attack. It may slow down or totally interrupt the service of a system.

<i>Attacks</i>	<i>Passive/Active</i>	<i>Threatening</i>
Snooping Traffic analysis	Passive	Confidentiality
Modification Masquerading Replaying Repudiation	Active	Integrity
Denial of service	Active	Availability

# Active Attacks (4 types)

## Masquerade

- Takes place when one entity **pretends** to be a different entity
- Usually includes one of the other forms of active attack

## Modification of messages

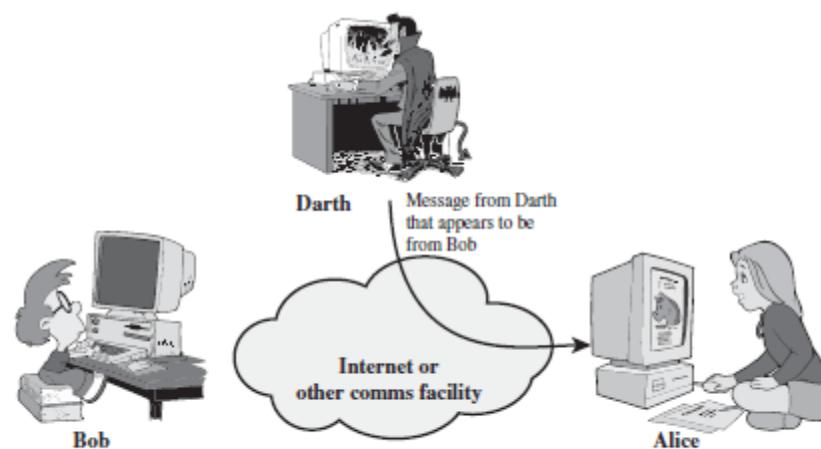
- Some portion of a legitimate message **is altered**, or messages are delayed or reordered to produce an **unauthorized effect**

## Replay

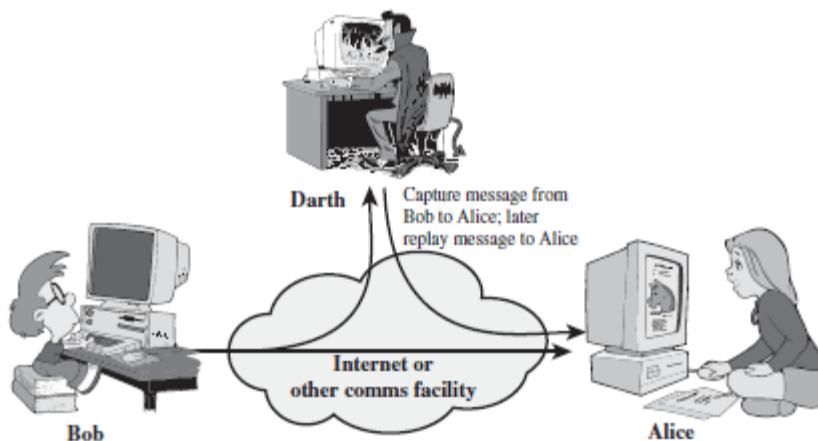
- Involves the passive **capture** of a data unit and its **subsequent retransmission** to produce an **unauthorized effect**

## Denial of service

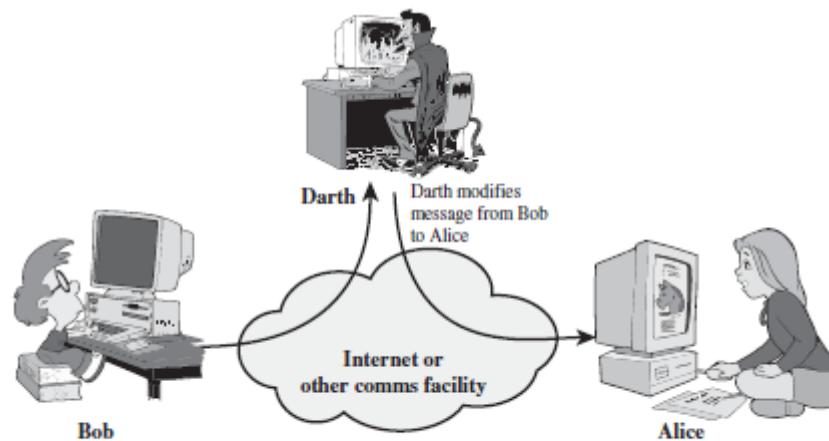
- Prevents or inhibits the normal use or management of communications facilities



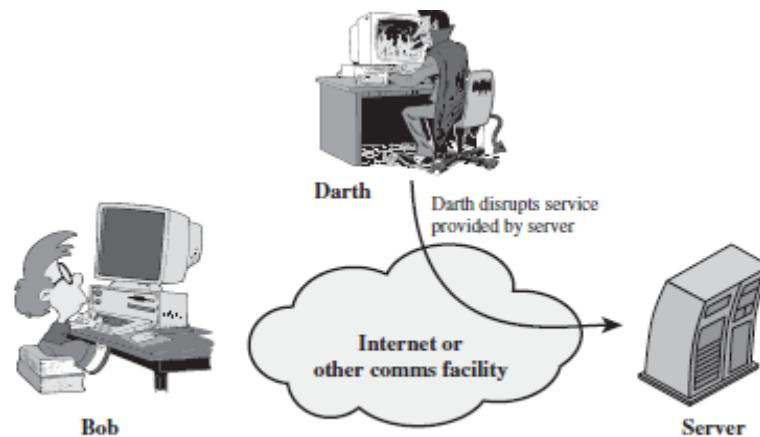
(a) Masquerade



(b) Replay



(c) Modification of messages



(d) Denial of service

# Security Services

- to prevent or detect attacks
- to enhance the security
- replicate functions of physical documents
  - e.g.
    - have signatures, dates
    - need protection from disclosure, tampering, or destruction
    - notarize
    - record

# Basic Security Services

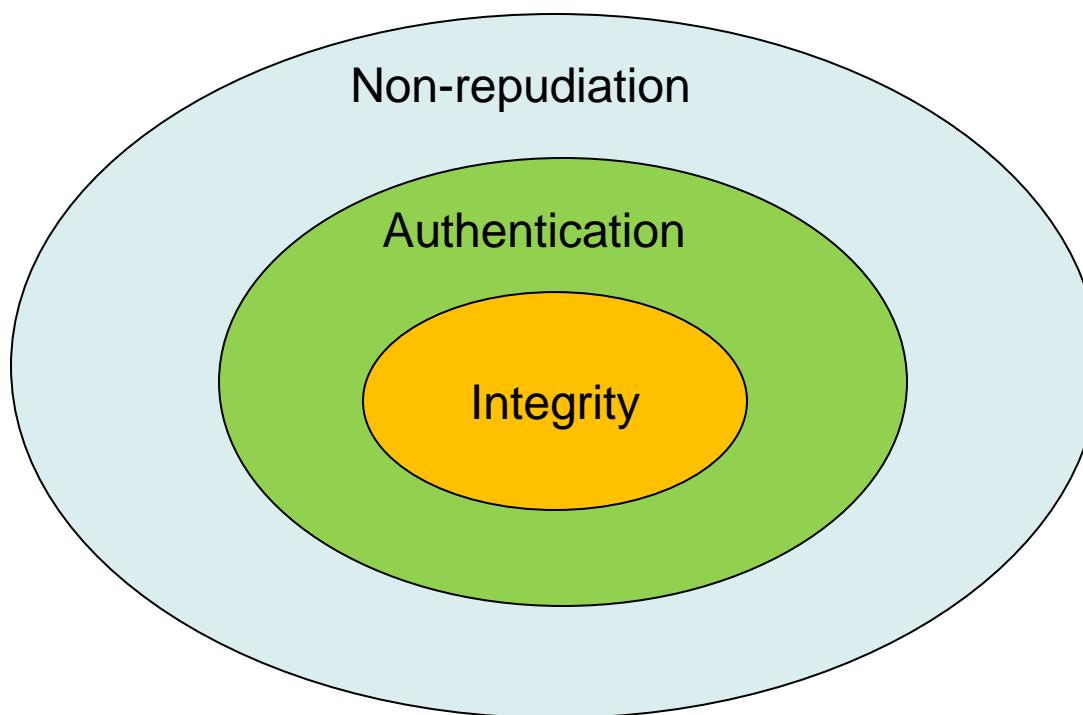
- **Authentication**
  - assurance that the communicating entity is the one it claims to be
  - peer entity authentication
    - mutual confidence in the identities of the parties involved in a connection
  - Data-origin authentication
    - assurance about the source of the received data
- **Access Control**
  - prevention of the unauthorized use of a resource
  - to achieve this, each entity trying to gain access must first be identified and authenticated, so that access rights can be tailored to the individual

# Basic Security Services

- Non-Repudiation
  - protection against denial by one of the parties in a communication
  - Origin non-repudiation
    - proof that the message was sent by the specified party
  - Destination non-repudiation
    - proof that the message was received by the specified party

# Relationships

- among integrity, data-origin authentication and non-repudiation



# Services and Mechanisms

**ITU-T** provides some security services and some mechanisms to implement those services. Security services and mechanisms are closely related because a mechanism or combination of mechanisms are used to provide a service.

# Security Services

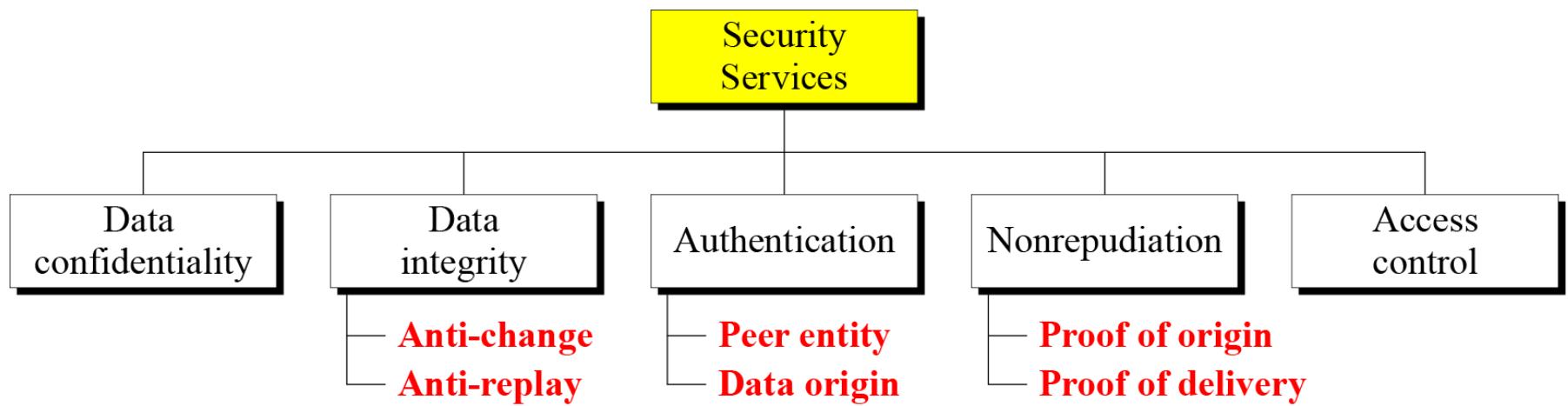
- Security service defined by X.800 as:  
A service provided by a protocol layer of communicating open systems and **that ensures adequate security** of the systems or of data transfers
- Defined by RFC 4949 as:  
A **processing or communication service** provided by a system to give **a specific kind of protection** to system resources

# X.800 Service Categories

X.800 divides these **services** into five categories and fourteen specific services

- Authentication
- Access control
- Data confidentiality
- Data integrity
- Nonrepudiation





# Security Services (X.800)

AUTHENTICATION	DATA INTEGRITY
The assurance that the communicating entity is the one that it claims to be.	The assurance that data received are exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay).
<b>Peer Entity Authentication</b> Used in association with a logical connection to provide confidence in the identity of the entities connected.	<b>Connection Integrity with Recovery</b> Provides for the integrity of all user data on a connection and detects any modification, insertion, deletion, or replay of any data within an entire data sequence, with recovery attempted.
<b>Data-Origin Authentication</b> In a connectionless transfer, provides assurance that the source of received data is as claimed.	<b>Connection Integrity without Recovery</b> As above, but provides only detection without recovery.
<b>ACCESS CONTROL</b>  The prevention of unauthorized use of a resource (i.e., this service controls who can have access to a resource, under what conditions access can occur, and what those accessing the resource are allowed to do).	<b>Selective-Field Connection Integrity</b> Provides for the integrity of selected fields within the user data of a data block transferred over a connection and takes the form of determination of whether the selected fields have been modified, inserted, deleted, or replayed.
<b>DATA CONFIDENTIALITY</b>  The protection of data from unauthorized disclosure.	<b>Connectionless Integrity</b> Provides for the integrity of a single connectionless data block and may take the form of detection of data modification. Additionally, a limited form of replay detection may be provided.
<b>Connection Confidentiality</b> The protection of all user data on a connection.	<b>Selective-Field Connectionless Integrity</b> Provides for the integrity of selected fields within a single connectionless data block; takes the form of determination of whether the selected fields have been modified.
<b>Connectionless Confidentiality</b> The protection of all user data in a single data block	<b>NONREPUDIATION</b>  Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication.
<b>Selective-Field Confidentiality</b> The confidentiality of selected fields within the user data on a connection or in a single data block.	<b>Nonrepudiation, Origin</b> Proof that the message was sent by the specified party.
<b>Traffic-Flow Confidentiality</b> The protection of the information that might be derived from observation of traffic flows.	<b>Nonrepudiation, Destination</b> Proof that the message was received by the specified party.

# Authentication

- Concerned with assuring that a

In the case of a **single message**, assures the recipient that the message is from the source that it claims to be from.

In the case of ongoing interaction, assures the **are authentic** and that the connection is not interfered with in such a way that a third party can masquerade as one of the two legitimate parties

**Two specific authentication services are defined in X.800:**

- Peer entity authentication
- Data origin authentication

Two entities are considered peers if they implement the same protocol in different systems (e.g., two TCP modules in two communicating systems).

# Access Control

- The ability **to limit and control the access** to host systems and applications via communications links
- To achieve this, each entity trying to gain access must first be **identified**, or **authenticated**, so that access rights can be tailored to the individual

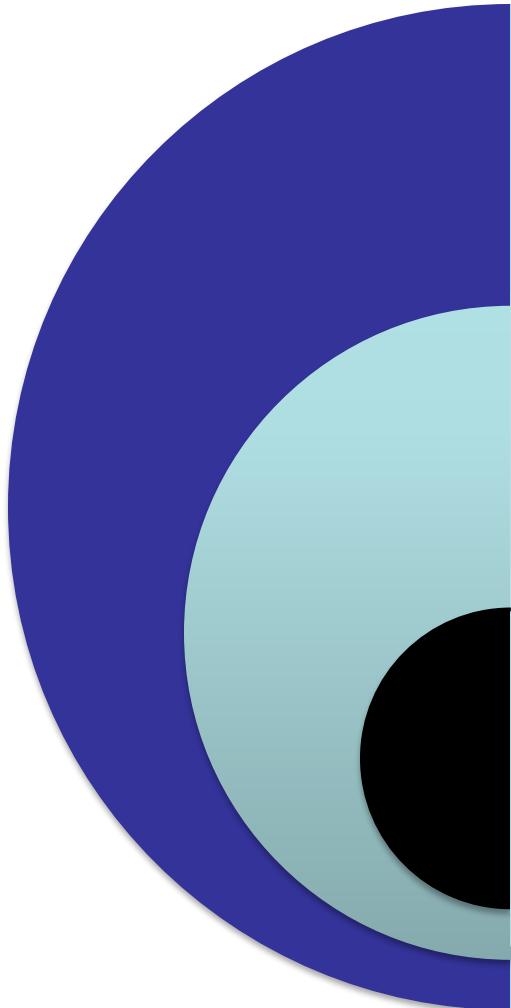


# Data Confidentiality

- The protection of transmitted data **from passive attacks**  
**Broadest service** protects **all user data** transmitted between two users over a period of time  
**Narrower forms of service** include the **protection of a single message** or even specific fields within a message
- The **protection of traffic flow** from analysis  
This requires that **an attacker not be able to observe the source and destination, frequency, length, or other characteristics of the traffic on a communications facility**



# Data Integrity



Can apply to a **stream of messages, a single message, or selected fields within a message**

**Connection-oriented integrity service** deals with a **stream of messages** and assures that messages are received as sent with **no duplication, insertion, modification, reordering, or replays**

**A connectionless integrity** service deals with **individual messages** without regard to any larger context and generally provides protection against **message modification only**

# Nonrepudiation

- Prevents either sender or receiver from **denying a transmitted message**
- When a message is sent, **the receiver can prove** that the alleged sender in fact sent the message
- When a message is received, **the sender can prove** that the alleged receiver in fact received the message



# Availability service

- Availability
  - The property of a system or a system resource being **accessible and usable upon demand by an authorized system entity**, according to performance specifications for the system
- Availability service
  - One that protects a system to **ensure its availability**
  - Addresses the security** concerns raised by **denial-of-service attacks**
  - Depends on proper management and control of system resources

# Security Mechanisms

## **Security mechanisms defined in X.800:**

- The mechanisms are divided into those that are implemented in a specific protocol layer, such as TCP or an application-layer protocol, and those that are not specific to any particular protocol layer or security service.

Table 1.3 Security Mechanisms (X.800)

<b>SPECIFIC SECURITY MECHANISMS</b>	<b>PERVASIVE SECURITY MECHANISMS</b>
<p>May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services.</p>	<p>Mechanisms that are not specific to any particular OSI security service or protocol layer.</p>
<p><b>Encipherment</b> The use of mathematical algorithms to transform data into a form that is not readily intelligible. The transformation and subsequent recovery of the data depend on an algorithm and zero or more encryption keys.</p>	<p><b>Trusted Functionality</b> That which is perceived to be correct with respect to some criteria (e.g., as established by a security policy).</p>
<p><b>Digital Signature</b> Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery (e.g., by the recipient).</p>	<p><b>Security Label</b> The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.</p>
<p><b>Access Control</b> A variety of mechanisms that enforce access rights to resources.</p>	<p><b>Event Detection</b> Detection of security-relevant events.</p>
<p><b>Data Integrity</b> A variety of mechanisms used to assure the integrity of a data unit or stream of data units.</p>	<p><b>Security Audit Trail</b> Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.</p>
<p><b>Authentication Exchange</b> A mechanism intended to ensure the identity of an entity by means of information exchange.</p>	<p><b>Security Recovery</b> Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions.</p>
<p><b>Traffic Padding</b> The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.</p>	
<p><b>Routing Control</b> Enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.</p>	
<p><b>Notarization</b> The use of a trusted third party to assure certain properties of a data exchange.</p>	

# Cryptographic Security Mechanisms

- **Encryption (a.k.a. Encipherment)**
  - use of mathematical algorithms to transform data into a form that is not readily intelligible
    - keys are involved

# Cryptographic Security Mechanisms

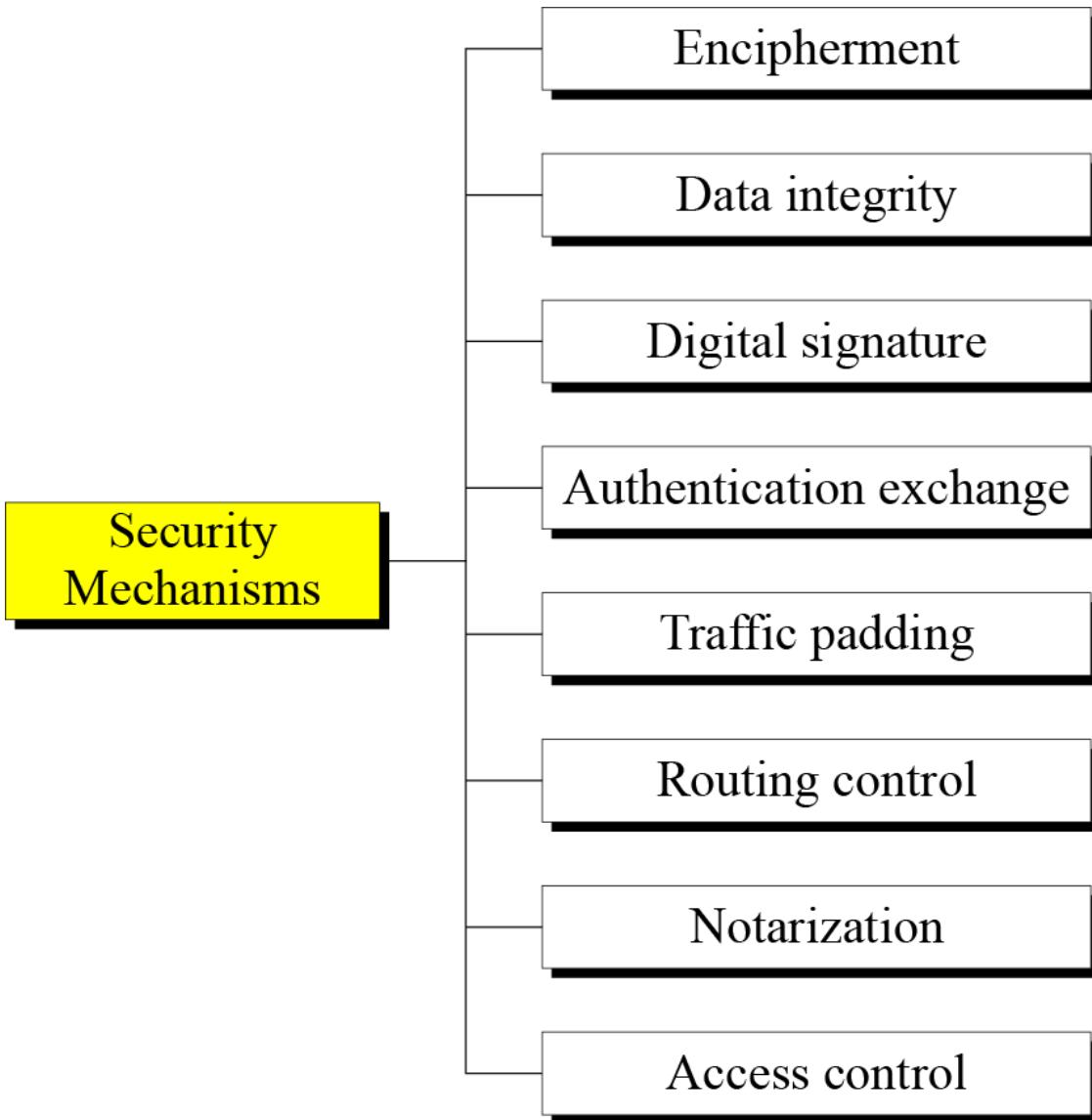
- **Message Digest**
  - similar to encryption, but one-way (recovery not possible)
  - generally no keys are used
- **Digital Signatures and Message Authentication Codes**
  - Data appended to, or a cryptographic transformation of, a data unit to prove the source and the integrity of the data
- **Authentication Exchange**
  - ensure the identity of an entity by exchanging some information

# Security Mechanisms

- Notarization
  - use of a trusted third party to assure certain properties of a data exchange
- Timestamping
  - inclusion of correct date and time within messages

# Security Mechanisms (X.800)

- Specific security mechanisms: incorporated into the appropriate protocol layer in order to provide some of the OSI security services
  - Encipherment
  - digital signatures
  - access controls
  - data integrity
  - authentication exchange
  - traffic padding
  - routing control
  - notarization



## SPECIFIC SECURITY MECHANISMS

May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services.

### Encipherment

The use of mathematical algorithms to transform data into a form that is not readily intelligible. The transformation and subsequent recovery of the data depend on an algorithm and zero or more encryption keys.

### Digital Signature

Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery (e.g., by the recipient).

### Access Control

A variety of mechanisms that enforce access rights to resources.

### Data Integrity

A variety of mechanisms used to assure the integrity of a data unit or stream of data units.

### Authentication Exchange

A mechanism intended to ensure the identity of an entity by means of information exchange.

### Traffic Padding

The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.

### Routing Control

Enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.

### Notarization

The use of a trusted third party to assure certain properties of a data exchange.

## PERVERSIVE SECURITY MECHANISMS

Mechanisms that are not specific to any particular OSI security service or protocol layer.

### Trusted Functionality

That which is perceived to be correct with respect to some criteria (e.g., as established by a security policy).

### Security Label

The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.

### Event Detection

Detection of security-relevant events.

### Security Audit Trail

Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.

### Security Recovery

Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions.

# Security Mechanisms (X.800)

# Relationship Between Security Services and Mechanisms

Service	Encipherment	Digital Signature	Access Control	Data Integrity	Mechanism
					Authentication Exchange
Peer Entity Authentication	Y	Y			Y
Data Origin Authentication	Y	Y			
Access Control			Y		
Confidentiality	Y				
Traffic Flow Confidentiality	Y				
Data Integrity	Y	Y		Y	
Nonrepudiation		Y		Y	
Availability				Y	Y

# Relationship Between Security Services and Mechanisms

<i>Security Service</i>	<i>Security Mechanism</i>
Data confidentiality	Encipherment and routing control
Data integrity	Encipherment, digital signature, data integrity
Authentication	Encipherment, digital signature, authentication exchanges
Nonrepudiation	Digital signature, data integrity, and notarization
Access control	Access control mechanism

# Model for Network Security

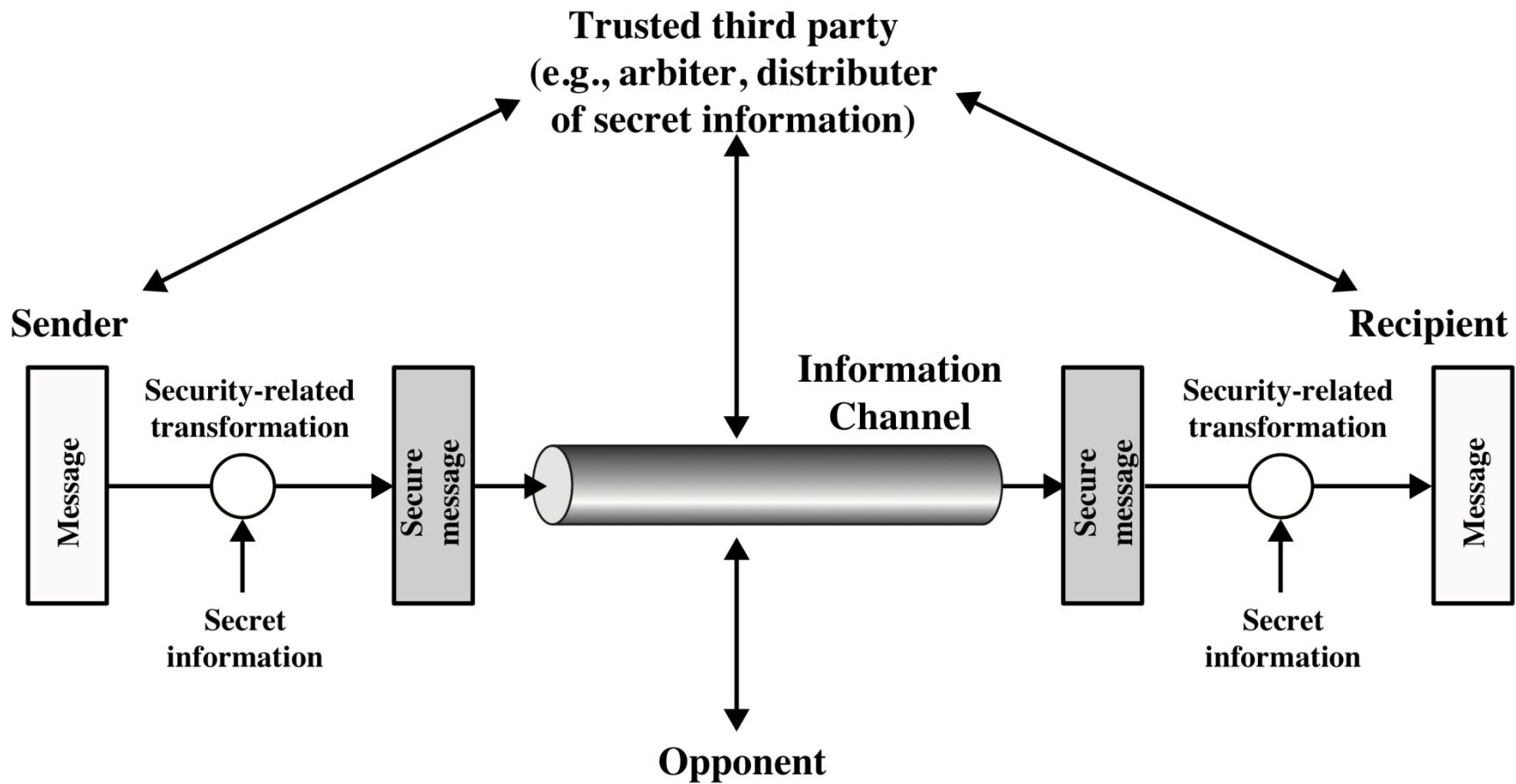
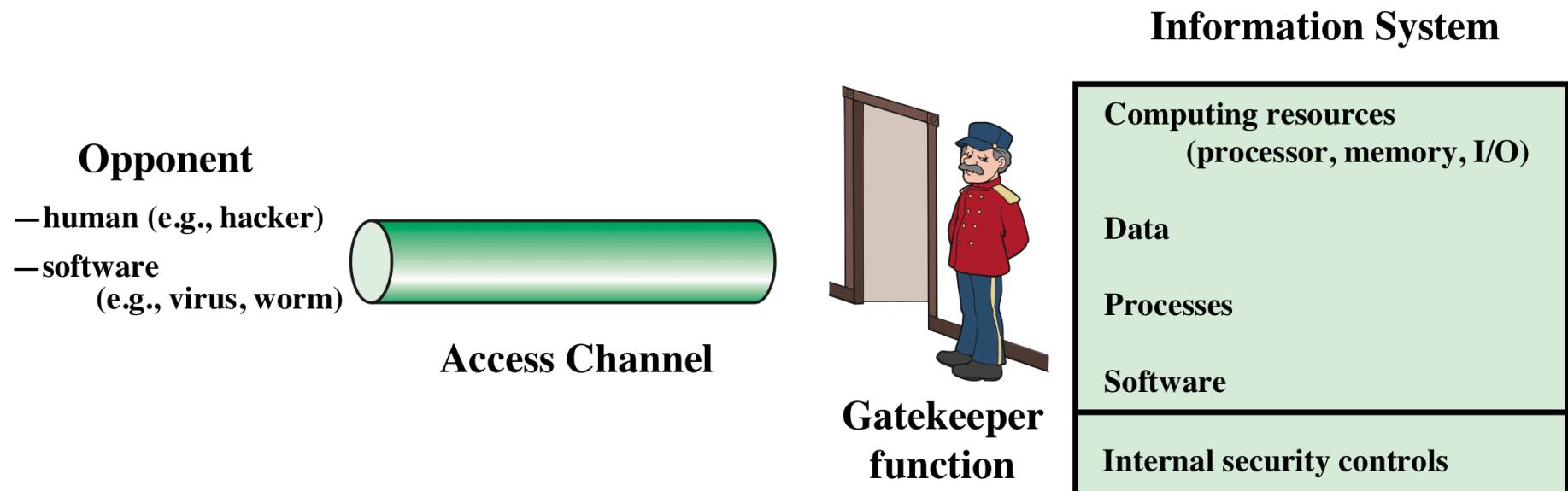


Figure 1.2 Model for Network Security

# A Model for Network Security

- Using this model requires us to:
  1. Design a suitable **algorithm** for the security transformation
  2. Generate the **secret information (keys)** used by the algorithm
  3. Develop methods to **distribute and share** the secret information
  4. Specify a **protocol** enabling the principals to use the transformation and secret information for a security service

# Network Access Security Model



**Figure 1.3 Network Access Security Model**

# Fundamental Security design Principles

- Economy of mechanism
- Fail-safe defaults (default is **lack of access**)
- Complete mediation (every access must be checked against the access control mechanism.)
- Open design
- Separation of privilege (practice in which multiple privilege attributes are required to achieve access to a restricted resource. )
- Least privilege
- Least common mechanism
- Psychological acceptability
- Isolation
- Encapsulation
- Modularity
- Layering
- Least astonishment

The National Centers of Academic Excellence in Information Assurance/Cyber Defense, which is jointly sponsored by the U.S. National Security Agency and the U.S. Department of Homeland Security, list the following as fundamental security design principles [NCAE13]:

# A Model for Network Access Security

- Using this model requires us to:
  1. Select appropriate **gatekeeper functions** to identify users
  2. Implement **security controls** to ensure only authorized users access designated information or resources

# More on Computer System Security

- Based on “Security Policies”
  - Set of rules that specify
    - How resources are managed to satisfy the security requirements
    - Which actions are permitted, which are not
  - Ultimate aim
    - Prevent security violations such as unauthorized access, data loss, service interruptions, etc.
  - Scope
    - Organizational or Individual
  - Implementation
    - Partially automated, but mostly humans are involved
  - Assurance and Evaluation
    - Assurance: degree of confidence to a system
    - Security products and systems must be evaluated using certain criteria in order to decide whether they assure security or not

# Attack Surfaces

- An attack surface consists of the reachable and exploitable vulnerabilities in a system
- Examples:
  - Open ports on outward facing Web and other servers, and code listening on those ports
  - Services available in a firewall
  - Code that processes incoming data, email, XML, office documents, etc.
  - Interfaces and Web forms
  - An employee with access to sensitive information vulnerable to a social engineering attack

# Attack Surface Categories

- **Network attack surface**
  - Refers to vulnerabilities over an enterprise network, wide-area network, or the Internet
    - E.g. DoS, intruders exploiting network protocol vulnerabilities
- **Software attack surface**
  - Refers to vulnerabilities in application, utility, or operating system code
- **Human attack surface**
  - Refers to vulnerabilities created by personnel or outsiders
  - E.g. social engineering, insider traitors

An attack surface analysis is useful for assessing the scale and severity of threats to a system.

# Attack Surface Categories

- As illustrated in Figure 1.3, the use of layering, or defense in depth, and attack surface reduction complement each other in mitigating security risk.

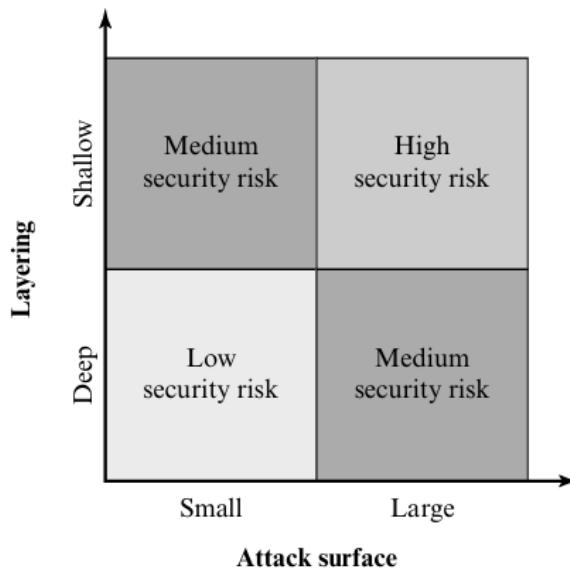


Figure 1.3 Defense in Depth and Attack Surface

# Attack Trees

- An attack tree is a branching, hierarchical data structure that represents a set of potential techniques for exploiting security vulnerabilities
- The security incident that is the goal of the attack is represented as the root node of the tree, and the ways that an attacker could reach that goal are iteratively and incrementally represented as branches and sub nodes of the tree.
- The motivation for the use of attack trees is to effectively exploit the information available on attack patterns.

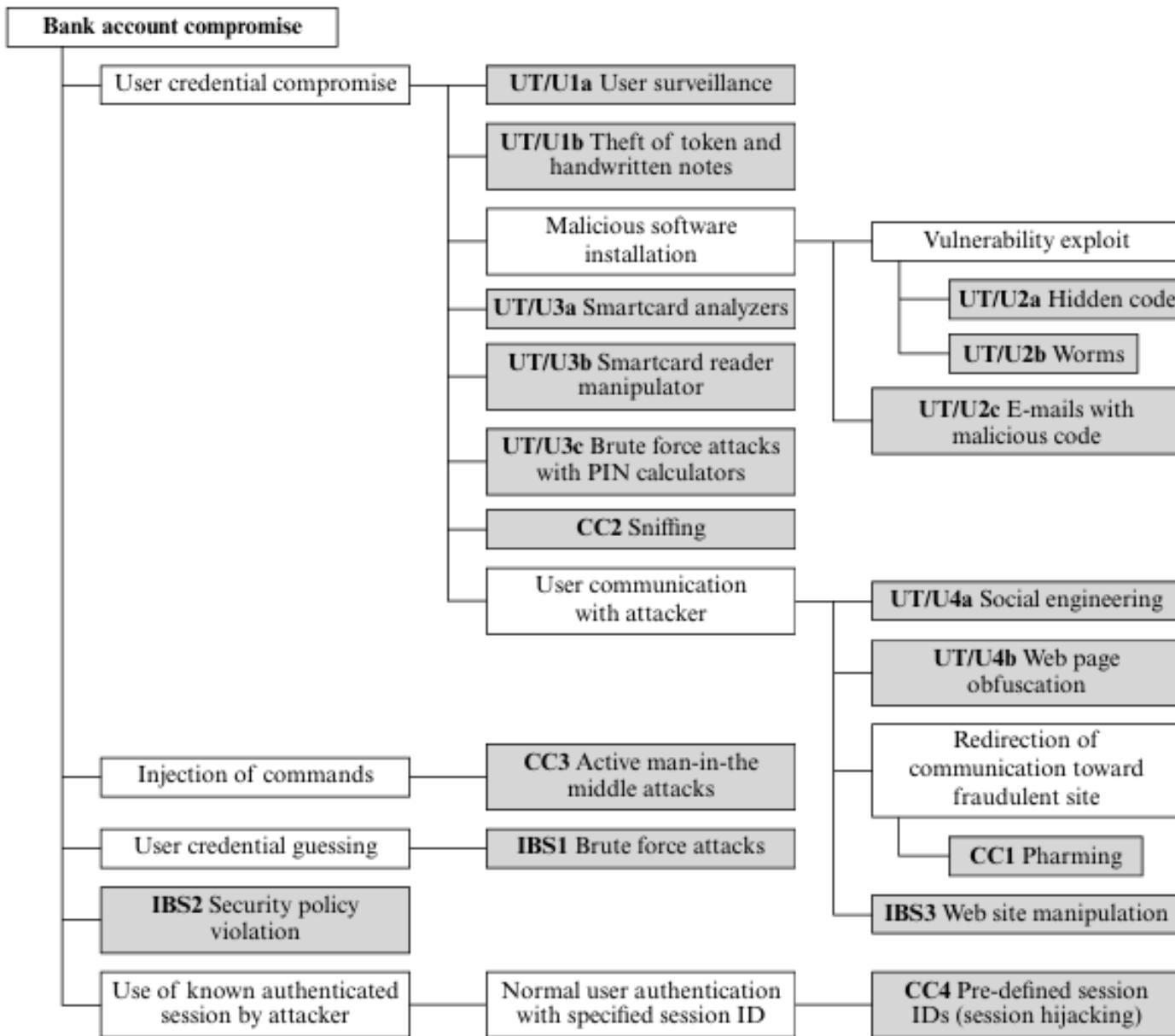


Figure 1.4 An Attack Tree for Internet Banking Authentication

# Unwanted Access

- Placement in a computer system of logic that exploits vulnerabilities in the system and that can affect application programs as well as utility programs



Programs can present two kinds of threats:

Information access threats

Service threats

Intercept or modify data on behalf of users who should not have access to that data

Exploit service flaws in computers to inhibit use by legitimate users

# Standards

- NIST
  - National Institute of Standards and Technology
  - U.S. federal agency that deals with measurement science, standards, and technology related to U.S. government use and to the promotion of U.S. private-sector innovation
  - NIST Federal Information Processing Standards (FIPS) and Special Publications (SP) have a worldwide impact
- ISOC
  - Internet Society
  - Professional membership society with worldwide organizational and individual membership
  - Provides leadership in addressing issues that confront the future of the Internet
  - Is the organization home for the groups responsible for Internet infrastructure standards, including the Internet Engineering Task Force (IETF) and the Internet Architecture Board (IAB)
  - Internet standards and related specifications are published as Requests for Comments (RFCs)

# Techniques

Mechanisms discussed already are only theoretical recipes to implement security. The actual implementation of security goals needs some techniques. Two techniques are prevalent today: **cryptography and steganography.**

# Cryptography

Cryptography, a word with Greek origins, means “secret writing.” However, we use the term to refer to the science and art of transforming messages to make them secure and immune to attacks.

# Steganography

The word steganography, with origin in Greek, means “covered writing,” in contrast with cryptography, which means “secret writing.”

Example: covering data with text

This book is mostly about cryptography, not steganography.

<input type="checkbox"/>							
0	1	0	0	0	0	1	

## Example: using dictionary

A	<b>friend</b>	<b>called</b>	a	<b>doctor.</b>
0	10010	0001	0	01001

## Example: covering data under color image

0101001 <u><b>1</b></u>	1011110 <u><b>0</b></u>	0101010 <u><b>1</b></u>
0101111 <u><b>0</b></u>	1011110 <u><b>0</b></u>	0110010 <u><b>1</b></u>
0111111 <u><b>0</b></u>	0100101 <u><b>0</b></u>	0001010 <u><b>1</b></u>

# Cryptographic Services

Cryptography supports the following services:

1. Confidentiality
2. Integrity
3. Authentication
4. Identity
5. Timeliness
6. Proof of ownership

Each has various different requirements in different circumstances, and each is supported by a wide variety of schemes.

# Applications

1. Communications (encryption or authentication)
2. File and data base security
3. Electronic funds transfer
4. Electronic Commerce
5. Digital cash
6. Contract signing
7. Electronic mail
8. Authentication: Passwords, PINs
9. Secure identification, Access control
10. Secure protocols
11. Proof of knowledge

# Applications (cont.)

- 12. Construction by collaborating parties (secret sharing)
- 13. Copyright protection
- 14. etc.

# Some Other Security Facts

- Not as simple as it might first appear to the novice
- Must consider all potential attacks when designing a system
- Generally yields complex and counterintuitive systems
- Battle of intelligent strategies between attacker and admin
- Requires regular monitoring
- Not considered as a beneficial investment until a security failure occurs
  - Actually security investments must be considered as insurance against attacks
- too often an afterthought
  - Not only from investment point of view, but also from design point of view

# Summary

- Computer security concepts
  - Definition
  - Examples
  - Challenges
- The OSI security architecture
- Security attacks
  - Passive attacks
  - Active attacks
- Security services
  - Authentication
  - Access control
  - Data confidentiality
  - Data integrity
  - Nonrepudiation
  - Availability service
- Security mechanisms
- Model for network security
- Standards

# Cryptographic Functions

- A communication game
- Protocol
- Magic function
- Cryptographic functions

# Cryptographic Functions

- Alice and Bob are two parties
- Want to go for dinner
- Alice for Chinese, Bob for Japanese
- How to resolve?

# Use of Unbiased Coin

- Alice tosses an unbiased coin with her hands covering, asks Bob of his choice: **HEADS** or **TAILS**
- If Bob's choice matches with the outcome of the toss, they go for Japanese food
- Consider the situation when they are far apart
- They can communicate with a telephone
- **What is the problem?**

# Problem of Trust

- Bob can not trust Alice, as Alice can tell a lie:
  - How do we solve the problem
- Solution to these kind of multi-party(plural number of players) problem are called technically “protocols”
- In order to resolve the problem, they engage “**a protocol**”
  - They use a magic function,  $f(x)$

# The Protocol

- Both of them agree on the function  $f(x)$
- An even number  $x$  represents HEAD
- An odd number  $x$  represents TAIL

# The Protocol

- Both of them agree on the function  $f(x)$
- An even number  $x$  represents HEAD
- An odd number  $x$  represents TAIL

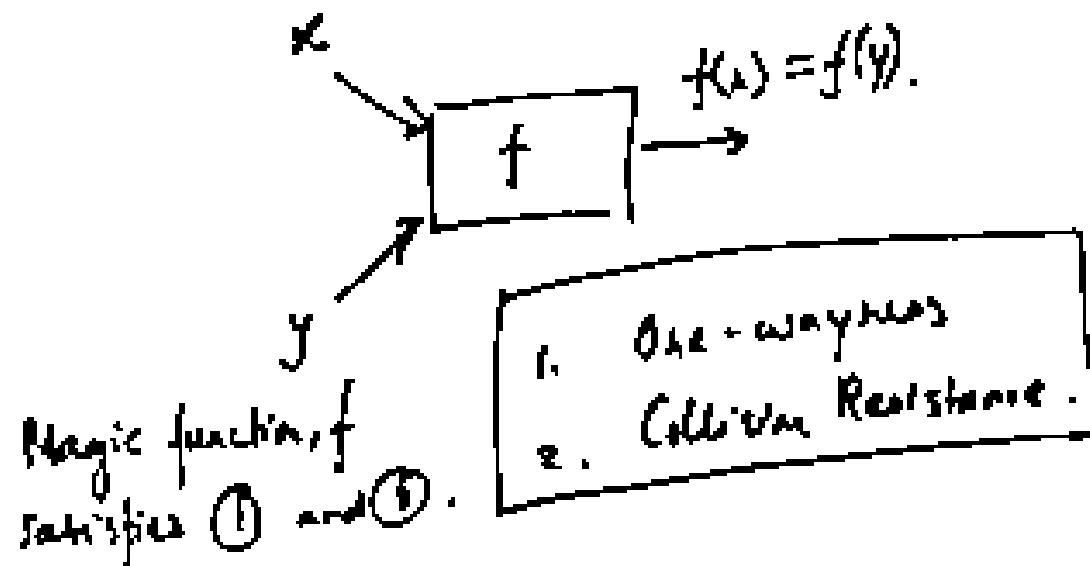
# Problem of Trust

Collision-resistance.

Given  $f(x)$ , output two values  $x \neq y$ ,

a.  $f(x) = f(y)$ .

Hard/Difficult problem.



# The Protocol

Alice & Bob.

they need to do the coin toss by using  
the function  $f$ .

Alice chooses randomly a value  $x$ .

$$\Pr[x \text{ is even}] = \Pr[x \text{ is odd}] = \frac{1}{2}$$

Alice  $\xrightarrow{x \text{ is even/odd}}$  Bob.

→ Head, then  
 $x$  is even.  
→ Tail, then  
 $x$  is odd.

Tail  $\Rightarrow x$  is odd.

Use function  
f to resolve

# Coin Flipping over Telephone

- Alice picks up a randomly large integer,  $x$  and computes  $f(x)$
- Bob tells Alice his guess of whether  $x$  is odd or even
- Alice then sends  $x$  to Bob
- Bob verifies by computing  $f(x)$

# Security Analysis

- Can Alice cheat?
- For that Alice need to create a  $y \neq x$ , st  $f(x) = f(y)$ . Hard to do.
- Can Bob guess better than a random guess?
- Bob listens to  $f(x)$  which speaks nothing of  $x$ . so his probability of guess is  $\frac{1}{2}$  (random guess)

# An Example

- Alice and Bob wish to resolve a dispute over telephone. We can encode the possibilities of the dispute by a binary value. For this they engage a protocol:
- Alice to Bob: Alice picks up randomly an  $x$ , which is a 200 bit number and computes the function  $f(x)$ . Alice sends  $f(x)$  to Bob.
- Bob to Alice: Bob tells Alice whether  $x$  was even or odd
- Alice to Bob: Alice then sends  $x$  to Bob, so Bob can verify whether his guess is correct.

# An Example

Alice  $\rightarrow$  Bob: Chooses  $x$  randomly.

$x$  is a 200 bit number.

(computes  $f(x)$ ). Sends it to Bob.

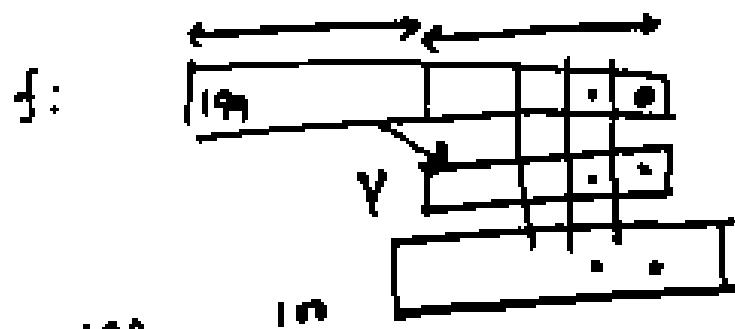
Bob  $\rightarrow$  Alice: Bob guesses whether  $x$  was even or odd.

Alice  $\rightarrow$  Bob: Sends  $x$  to Bob.

Bob verifies by computing  $f(x)$  and tallying with the previously received values.

# An Example

$$x \in \{0, 1\}^{2^{\infty}}$$



$$x = \overbrace{1111}^{100} \overbrace{1100 \dots}^{10} \dots$$

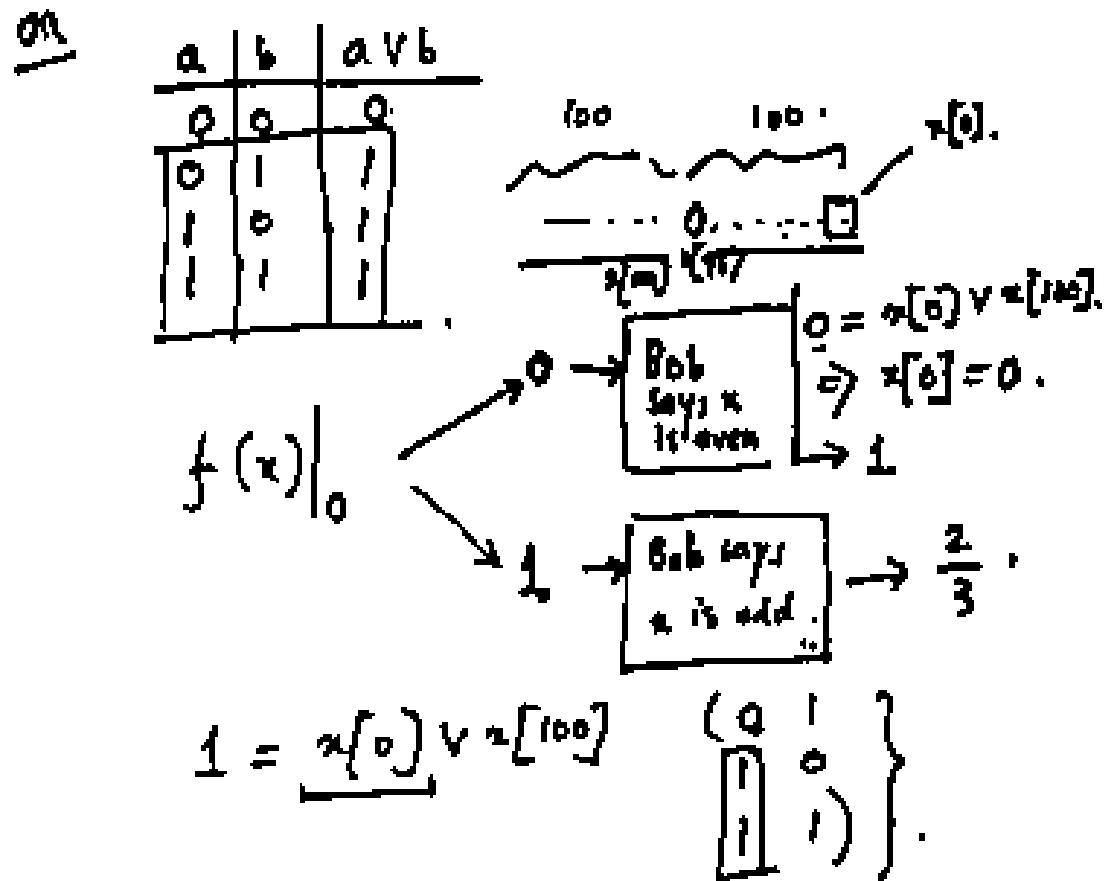
$$f(x) = \begin{matrix} 111 \dots & 1 \\ 100 \dots & 0 \\ \vdots & \vdots \\ 1 \dots & 111 \end{matrix}$$

$$\left\{ \begin{array}{l} p_r[\text{Bob succeeds}] = ? \\ p_r[\text{Alice cheats}] = ? \end{array} \right.$$

# Bob's Strategy

- Bob's Experiment:
  - Input  $f(x)$
  - Output parity of  $x$
- Algorithm
  - If  $[f(x)]_0 = 0$  the  $x$  is even
  - Else  $x$  is odd

# Bob's Strategy



# Bob's Prob of Success

- If  $X$  is chosen at random

$$\Pr[X \text{ is even}] = \Pr[X \text{ is odd}] = 1/2$$

$$\begin{aligned}\Pr[\text{Bob succeeds}] &= \Pr[X \text{ is even}] \Pr[\text{Bob succeeds} | X \text{ is even}] + \Pr[X \text{ is odd}] \Pr[\text{Bob succeeds} | X \text{ is odd}] \\ &= 1/2 \cdot 1/2 + 1/2 \cdot 1 = 3/4\end{aligned}$$

# Bob's Prob of Success

- Remember we compute alice's cheating probability irrespective of Bob's strategy

# Bob's Prob of Success

$$\begin{aligned} \Pr[\text{Bob succeeds}] &= \Pr[X \text{ is even}] \Pr[\text{Bob succeeds} | X \text{ is even}] \\ &\quad + \Pr[X \text{ is odd}] \Pr[\text{Bob succeeds} | X \text{ is odd}]. \\ &= \frac{1}{2} \left( \Pr[\text{Bob succeeds} | X \text{ is even}] \right. \\ &\quad \left. + \Pr[\text{Bob succeeds} | X \text{ is odd}] \right). \end{aligned}$$

$$\boxed{\begin{array}{ll} x[0] = 0. & f(\psi)|_0 = \alpha[100] \vee \alpha[0] \\ & = \alpha[100]. \\ x[1] = 1. & f(\psi)|_1 = \alpha[100] \vee 1 \\ & = 1. \end{array}}$$



# Classical Encryption Techniques



Dr. Md. Mahbubur Rahman



# Outline

---

1. To define the terms and the concepts of symmetric key ciphers
2. To emphasize the two categories of traditional ciphers: substitution and transposition ciphers
3. To describe the categories of cryptanalysis used to break the symmetric ciphers
4. To introduce the concepts of the stream ciphers and block ciphers
5. To discuss some very dominant ciphers used in the past, such as the Enigma machine



# Cryptographic Algorithms

---

- ▶ **Symmetric encryption:** Used to conceal the contents of blocks or streams of data of any size, including messages, files, encryption keys and passwords.
- ▶ **Asymmetric encryption:** Used to conceal small blocks of data, such as encryption keys and hash function values, which are used in digital signatures.
- ▶ **Data integrity algorithms:** Used to protect blocks of data, such as messages, from alteration.
- ▶ **Authentication protocols:** These are schemes based on the use of cryptographic algorithms designed to authenticate the identity of entities.

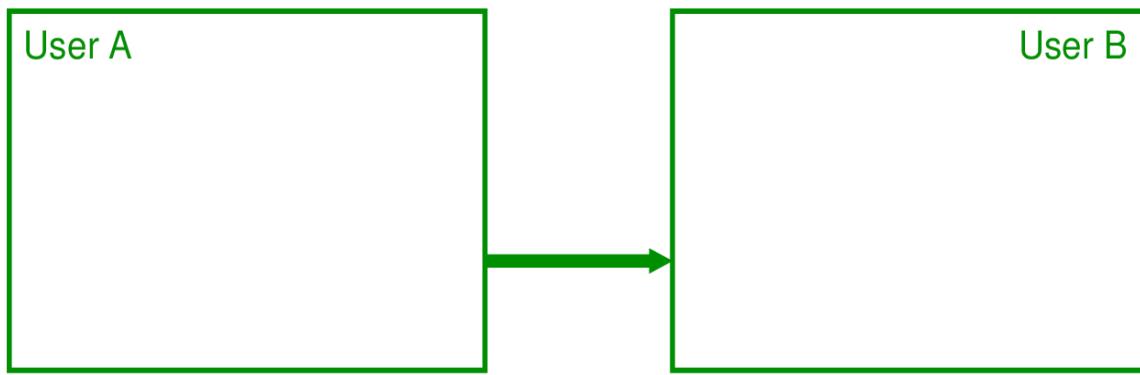


# Encryption for Confidentiality

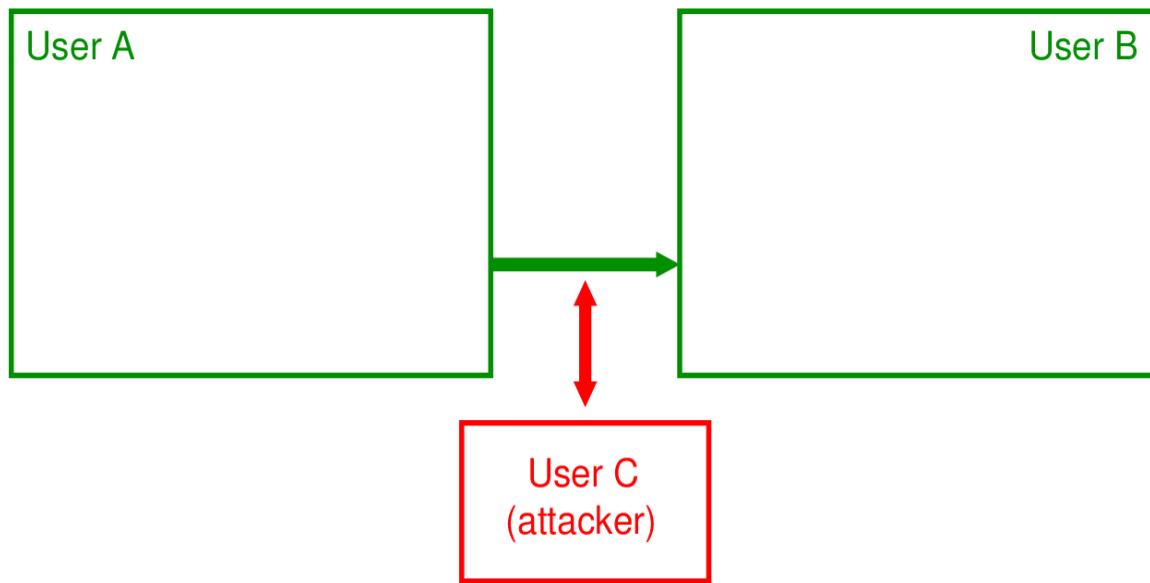
- ▶ Aim: assure **confidential** information not made available to unauthorized individuals (**data confidentiality**)
- ▶ How: **encrypt** the original data; anyone can see the encrypted data, but only **authorized individuals** can **decrypt** to see the original data
- ▶ Used for both **sending** data across network and **storing** data on a computer system



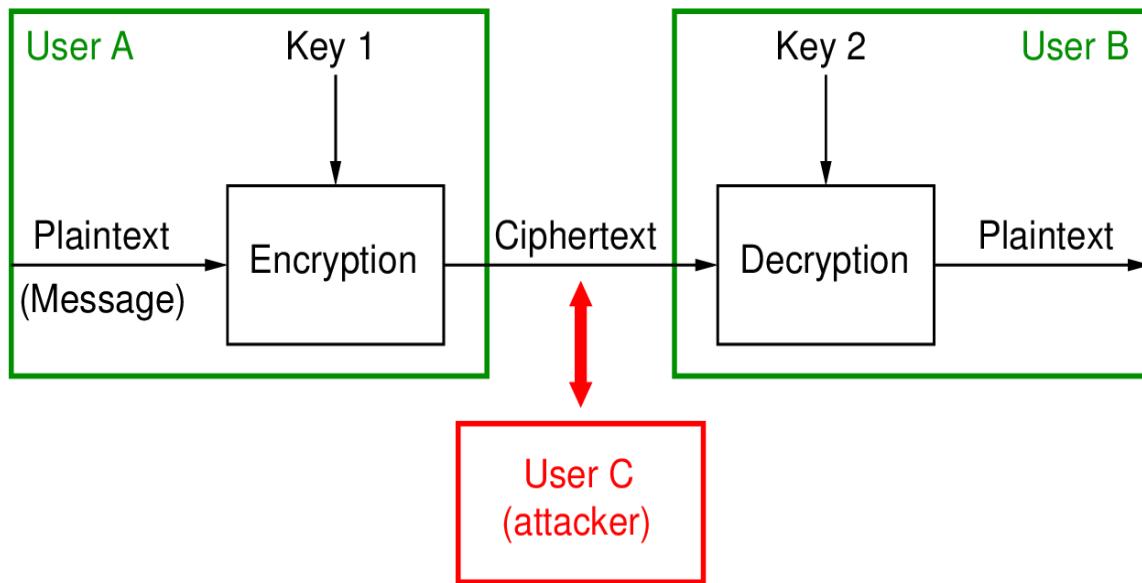
# Model of Encryption for Confidentiality



# Model of Encryption for Confidentiality



# Model of Encryption for Confidentiality



# Terminology

---

Plaintext	original message
Ciphertext	encrypted or coded message
Encryption	convert from plaintext to ciphertext (enciphering)
Decryption	restore the plaintext from ciphertext (deciphering)
Key	information used in cipher known only to sender/receiver
Cipher	a particular algorithm (cryptographic system)
Cryptography	study of algorithms used for encryption
Cryptanalysis	study of techniques for decryption without knowledge of plaintext
Cryptology	areas of cryptography and cryptanalysis



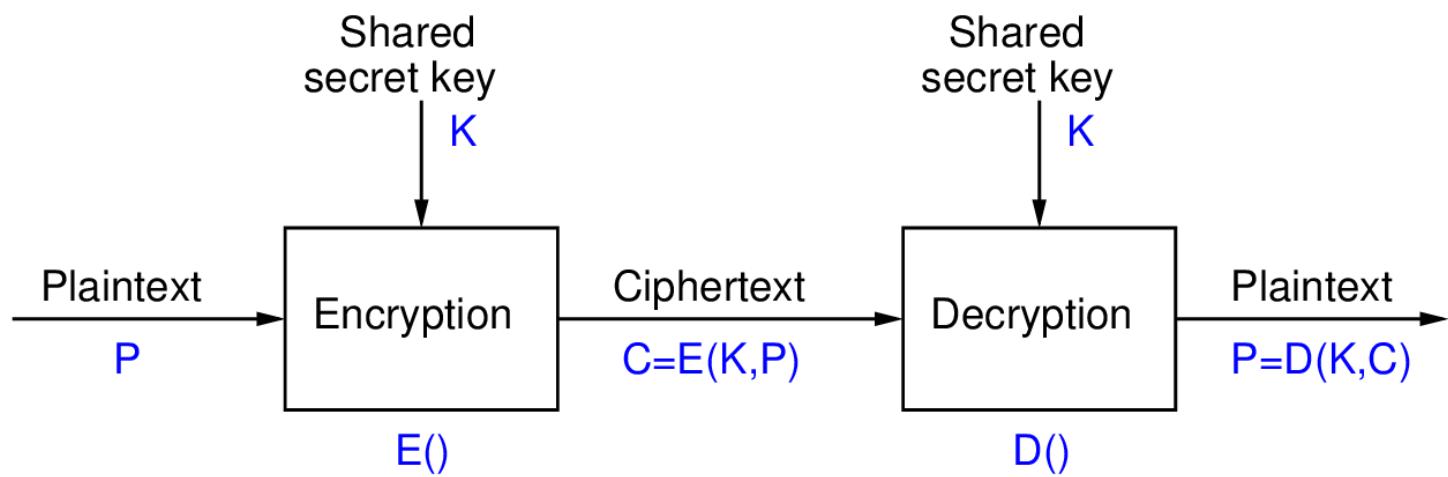
# Symmetric Encryption

---

- ❑ or conventional / private-key / single-key
- ❑ sender and recipient share a common key
- ❑ all classical encryption algorithms are private-key
- ❑ was only type prior to invention of public-key in 1970's
- ❑ and by far most widely used



# Symmetric Cipher Model



# Requirements and assumptions

---

- ▶ two requirements for secure use of symmetric encryption:
  - a strong encryption algorithm (**cannot decrypt and know key**)
  - a secret key known only to sender / receiver
- ▶ mathematically have:
$$C = E(K, P)$$
$$P = D(K, C)$$
- ▶ Assumptions
  - encryption algorithm is known
  - a secure channel to distribute key



## Kerckhoff's principle

---

- ▶ Although it may appear that a cipher would be more secure if we hide both the encryption/decryption algorithm and the secret key, this is not recommended.
- ▶ Based on Kerckhoff's principle, one should always assume that the adversary , Eve, knows the encryption/decryption algorithm. The resistance of the cipher to attack must be based only on the secrecy of key.
- ▶ In other words, guessing the key should be so difficult that there is no need to hide the encryption/decryption algorithm.



# Characterizing Cryptographic Systems

Operations used for encryption:

- Substitution replace one element in plaintext with another
- Transposition re-arrange elements
- Product systems multiple stages of substitutions and transpositions

Number of keys used:

- Symmetric sender/receiver use same key (single-key, secret-key, shared-key, conventional)
- Public-key sender/receiver use different keys (asymmetric)

Processing of plaintext:

- Block cipher process one block of elements at a time
- Stream cipher process input elements continuously



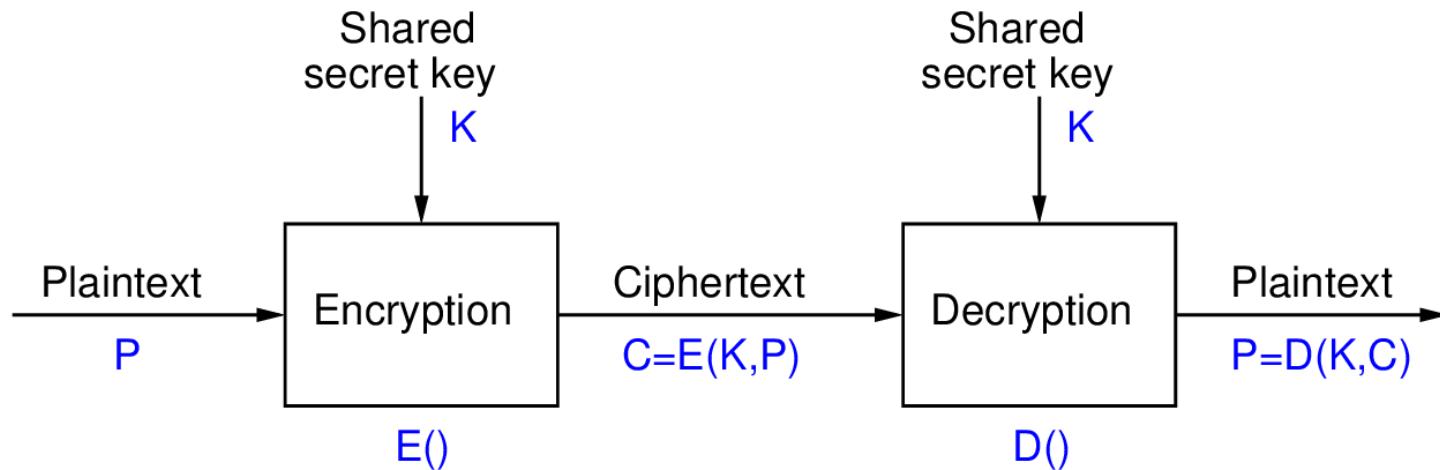
# Cryptography Classification

---

- By type of encryption operations used
  - Substitution: Meet Me  $\Rightarrow$  Offu Of
  - Transposition: Meet Me  $\Rightarrow$  Me etM
  - Product
- By number of keys used
  - Single-key or Secret Key
  - Two-key or Public Key
- By the way in which plaintext is processed
  - Block: ABCD EFGH IJKL
  - Stream: ABCDEFGHIJKL



# Symmetric Key Encryption for Confidentiality



## Requirements

- ▶ **Strong encryption algorithm:** given algorithm, ciphertext and known pairs of (plaintext, ciphertext), attacker should be unable to find plaintext or key
- ▶ **Shared secret keys:** sender and receiver both have shared a secret key; no-one else knows the key

# Attacks

## Goal of the Attacker

- Discover the plaintext (good)
- Discover the key (better)

## Assumed Attacker Knowledge

- Ciphertext (want to decrypt)
- Algorithm (nature of the algorithm) or general idea of the type of plaintext
- Other pairs of (plaintext, ciphertext) using **same key** (not the plaintext in question)

## Attack Methods

Brute-force attack Try every possible key on ciphertext

Cryptanalysis Exploit characteristics of algorithm to deduce plaintext or key

Assumption: attacker can recognize correct plaintext



# Attacks on Block Ciphers

## ► Brute Force Attack

---

**Approach:** try all keys in key space

**Metric:** number of operations (time)

k bit key requires  $2^k$  operations

Depends on key length and computer speed

## ► Cryptanalysis

**Approach:** Find weaknesses in algorithms

**Methods:** Linear cryptanalysis, differential cryptanalysis, meet-in-the-middle attack, side-channel attacks

**Metrics:** Number of operations

Amount of memory

Number of known plaintexts/ciphertexts

If either succeed all key usages are compromised



# Cryptanalysis and Brute-Force Attack

---

- ▶ **Cryptanalysis :** Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext-ciphertext pairs.
- ▶ **Brute-Force Attack :** The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained.



# Brute Force Attack

- ▶ always possible to simply try every key
- ▶ most basic attack, proportional to key size
- ▶ assume either know / recognise plaintext

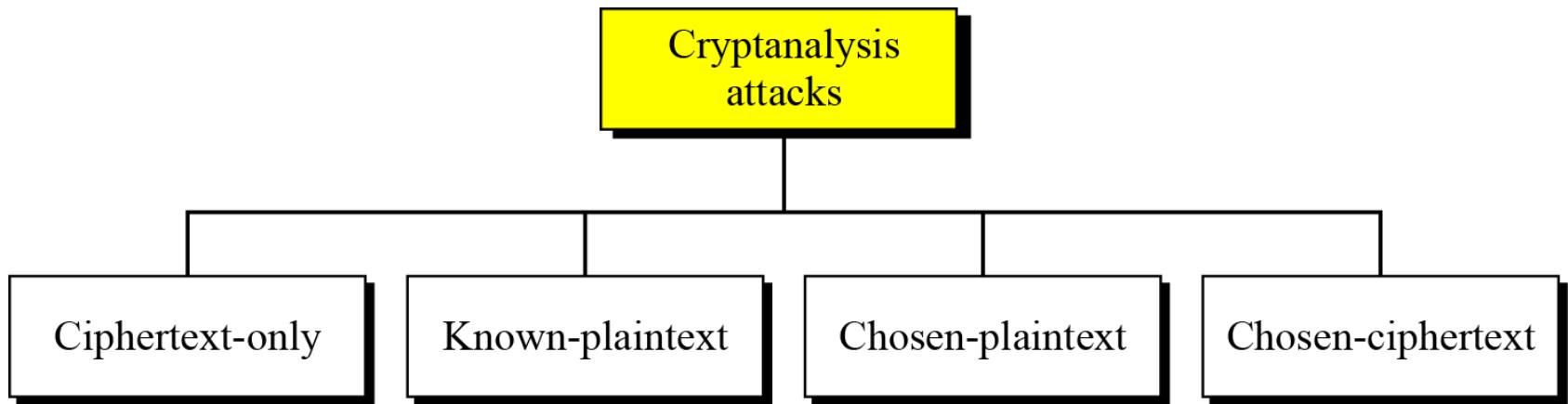
Key length	Key space	Worst case time at speed:		
		$10^9/\text{sec}$	$10^{12}/\text{sec}$	$10^{15}/\text{sec}$
32	$2^{32}$	4 sec	4 ms	4 us
56	$2^{56}$	833 days	20 hrs	72 sec
64	$2^{64}$	584 yrs	213 days	5 sec
128	$2^{128}$	$10^{22}$ yrs	$10^{19}$ yrs	$10^{16}$ yrs
192	$2^{192}$	$10^{41}$ yrs	$10^{38}$ yrs	$10^{35}$ yrs
256	$2^{256}$	$10^{60}$ yrs	$10^{57}$ yrs	$10^{54}$ yrs
26!	$2^{88}$	$10^{10}$ yrs	$10^7$ yrs	$10^4$ yrs

Age of Earth:  $4 \times 10^9$  years

Age of Universe:  $1.3 \times 10^{10}$  years

# Cryptanalysis

As cryptography is the science and art of creating secret codes, **cryptanalysis** is the science and art of breaking those codes.

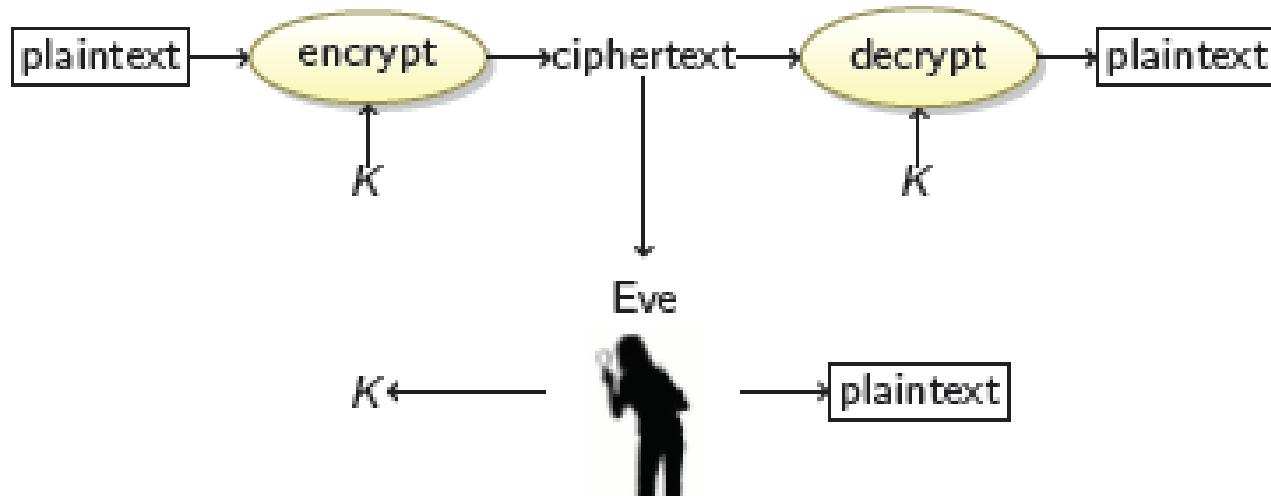


Cryptanalysis attacks



# Cryptanalysis (Cont.)

## Ciphertext-only attack

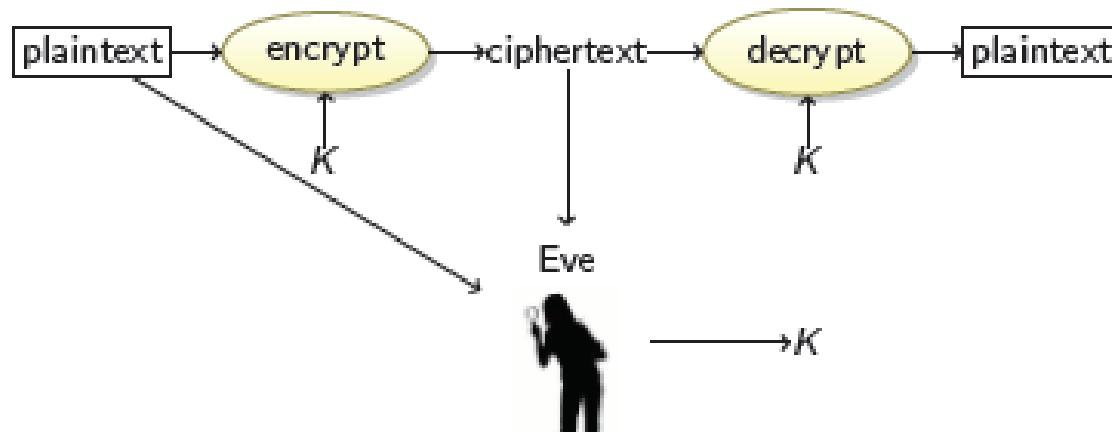


**We have:** the ciphertext of several messages that have been encrypted with the same key, K.

**We recover:** the plaintexts, or K.

# Cryptanalysis (Cont.)

## Known-Plaintext Attack

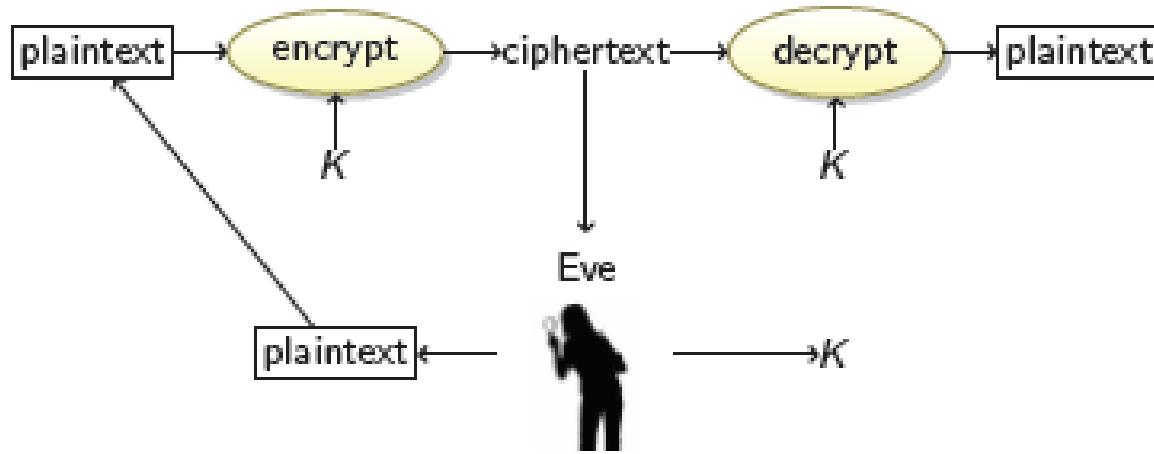


We have: the ciphertexts and corresponding plaintexts of several messages, all encrypted with the same key  $K$ .

We recover: the key  $K$ .

# Cryptanalysis (Cont.)

## Chosen-Plaintext Attack

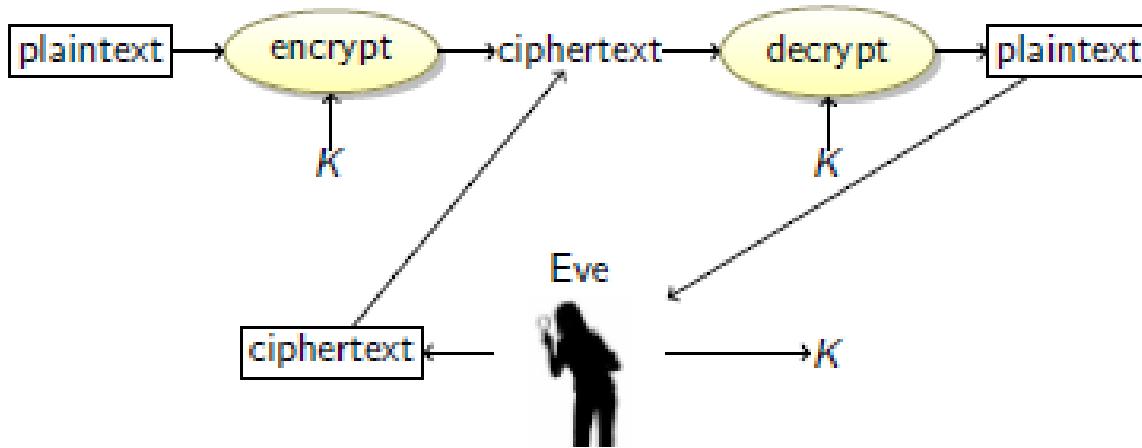


**We have:** the ciphertext of several messages that have been encrypted with the same key  $K$ , such that we get to choose the plaintexts.

**We recover:** the key  $K$ .

# Cryptanalysis (Cont.)

## Chosen-Ciphertext Attack



**We have:** the plaintext of several messages that have been encrypted with the same key  $K$ , such that we get to choose the ciphertexts.

**We recover:** the key  $K$ .

# Cryptanalysis: Summary

Table 2.1 Types of Attacks on Encrypted Messages

Type of Attack	Known to Cryptanalyst
Ciphertext Only	<ul style="list-style-type: none"><li>• Encryption algorithm</li><li>• Ciphertext</li></ul>
Known Plaintext	<ul style="list-style-type: none"><li>• Encryption algorithm</li><li>• Ciphertext</li><li>• One or more plaintext–ciphertext pairs formed with the secret key</li></ul>
Chosen Plaintext	<ul style="list-style-type: none"><li>• Encryption algorithm</li><li>• Ciphertext</li><li>• Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key</li></ul>
Chosen Ciphertext	<ul style="list-style-type: none"><li>• Encryption algorithm</li><li>• Ciphertext</li><li>• Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key</li></ul>
Chosen Text	<ul style="list-style-type: none"><li>• Encryption algorithm</li><li>• Ciphertext</li><li>• Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key</li><li>• Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key</li></ul>

## Exercise

---

What type of attack is Eve employing here:

- ① Eve tricks Alice into decrypting a bunch of ciphertexts that Alice encrypted last month.
- ② Eve picks Alice's encrypted cell phone conversations.
- ③ Eve has given a bunch of messages to Alice for her to sign using the RSA signature scheme, which Alice does without looking at the messages and without using a one-way hash function. In fact, these messages are ciphertexts that Eve constructed to help her figure out Alice's RSA private key.
- ④ Eve has bet Bob that she can figure out the AES secret key he shares with Alice if he will simply encrypt 20 messages for Eve using that key. Bob agrees. Eve gives him 20 messages, which he then encrypts and emails back to Eve.



# Cryptography

---

- ▶ Cryptographic systems are characterized along **three** independent dimensions:
  - ▶ 1. The type of operations used for transforming plaintext to ciphertext.—**substitution and transposition (or product)**
  - ▶ 2. The number of keys used. (**sym. and asym.**)
  - ▶ 3 The way in which the plaintext is processed. (**block and stream**)



# Measures of Security

## Unconditionally Secure

- ▶ **Ciphertext** does not contain **enough information** to derive plaintext or key
- ▶ **One-time pad** is only unconditionally secure cipher ( but not very practical )

## Computationally Secure

If either:

- Cost of breaking cipher **exceeds** value of **encrypted information**
- Time required to break cipher exceeds **useful lifetime of encrypted information**
- ▶ Hard to estimate value/lifetime of some information
- ▶ Hard to estimate how much effort needed to break cipher



# Motivation for cryptanalysts

---

- ▶ All forms of cryptanalysis for symmetric encryption schemes are designed to **exploit** the fact that **traces of structure or pattern in the plaintext** may **survive encryption** and be discernible in the ciphertext.



# Substitution ciphers

A substitution cipher replaces one symbol with another. Substitution ciphers can be categorized as either monoalphabetic ciphers or polyalphabetic ciphers.

A substitution cipher replaces one symbol with another.

## Topics:

Monoalphabetic Ciphers  
Polyalphabetic Ciphers



# Monoalphabetic Ciphers

In monoalphabetic substitution, the relationship between a symbol in the plaintext to a symbol in the ciphertext is always one-to-one.



# Encoding

- In these simple ciphers we typically
  - ➊ convert all letters to upper case;
  - ➋ remove spaces;
  - ➌ remove punctuation;
  - ➍ break into blocks of the same size (typically 5 letters);
  - ➎ add some unusual letter (like Z) to the last block, if necessary.
- Example:

It wAs A DAk and sTormY NighT ...

turns into

ITWAS ADARK ANDST ORMYN IGH TZ

- Knowing word boundaries can help with cryptanalysis.

# Continued

## Example

The following shows a plaintext and its corresponding ciphertext. The cipher is probably monoalphabetic because both I's (els) are encrypted as O's.

**Plaintext:** hello

**Ciphertext:** KHOOR

## Example

The following shows a plaintext and its corresponding ciphertext. The cipher is not monoalphabetic because each I (el) is encrypted by a different character.

**Plaintext:** hello

**Ciphertext:** KHOLR



# Additive Cipher

The **simplest** monoalphabetic cipher is the additive cipher.

This cipher is sometimes called a **shift cipher** and sometimes a **Caesar cipher**, but the term additive cipher better reveals its mathematical nature.

Plaintext and ciphertext in  $Z_{26}$

Plaintext →	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
Ciphertext →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25



# Additive Cipher

- ▶ Replace each letter by the letter **three positions** along in alphabet

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

## Generalized Caesar Cipher

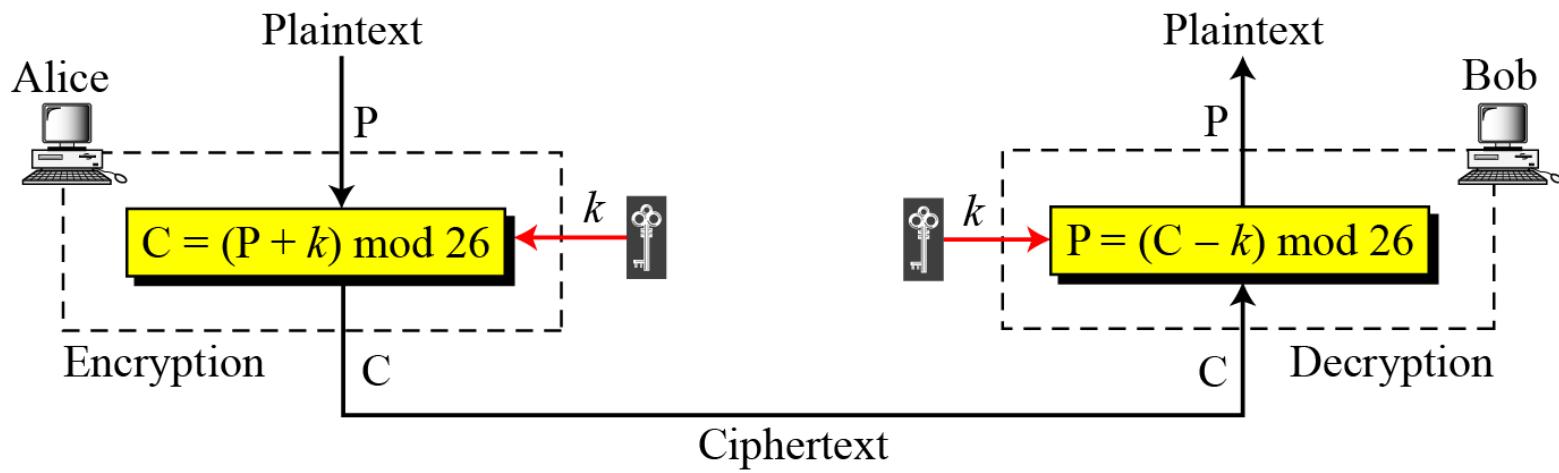
- ▶ Allow shift by  $k$  positions
- ▶ Assume each letter assigned number ( $a = 0, b = 1, \dots$ )

$$C = E(k,p) = (p + k) \bmod 26$$

$$p = D(k,C) = (C - k) \bmod 26$$



# Additive Cipher



When the cipher is additive, the plaintext, ciphertext, and key are integers in  $\mathbb{Z}_{26}$ .

# Additive Cipher

Use the additive cipher with key = 15 to encrypt the message “hello”.

## Solution

We apply the encryption algorithm to the plaintext, character by character:

Plaintext: h → 07	Encryption: $(07 + 15) \text{ mod } 26$	Ciphertext: 22 → W
Plaintext: e → 04	Encryption: $(04 + 15) \text{ mod } 26$	Ciphertext: 19 → T
Plaintext: l → 11	Encryption: $(11 + 15) \text{ mod } 26$	Ciphertext: 00 → A
Plaintext: l → 11	Encryption: $(11 + 15) \text{ mod } 26$	Ciphertext: 00 → A
Plaintext: o → 14	Encryption: $(14 + 15) \text{ mod } 26$	Ciphertext: 03 → D



# Continued

## Shift Cipher and Caesar Cipher

Hitorically, additive ciphers are called shift ciphers. Julius Caesar used an additive cipher to communicate with his officers. For this reason, additive ciphers are sometimes referred to as the Caesar cipher. Caesar used a key of 3 for his communications.

Additive ciphers are sometimes referred to as shift ciphers or Caesar cipher.



# Continued

Eve has intercepted the ciphertext “UVACLYFZLJBYL”. Show how she can use a brute-force attack to break the cipher.

## Solution

Eve tries keys from 1 to 7. With a key of 7, the plaintext is “not very secure”, which makes sense.

**Ciphertext:** UVACLYFZLJBYL

<b>K = 1</b>	→	<b>Plaintext:</b> tuzbkxeykiaxk
<b>K = 2</b>	→	<b>Plaintext:</b> styajwdxjhzwj
<b>K = 3</b>	→	<b>Plaintext:</b> rsxzivcwigyvi
<b>K = 4</b>	→	<b>Plaintext:</b> qrwyhubvhfxuh
<b>K = 5</b>	→	<b>Plaintext:</b> pqvxgtaugewtg
<b>K = 6</b>	→	<b>Plaintext:</b> opuwfsztfdvsf
<b>K = 7</b>	→	<b>Plaintext:</b> notverysecure



# Breaking the Caesar Cipher

---

- ▶ Brute force attack
  - The encryption and decryption algorithms are known.
    - Try all 25 keys, e.g.  $k = 1, k = 2, \dots$
    - Plaintext should be recognised
- ▶ Recognising plaintext in brute force attacks
  - Need to know “structure” of plaintext
  - Language? Compression?
- ▶ How to improve against brute force?
  - Hide the encryption/decryption algorithm: **Not practical**
  - Compress, use different language: **Limited options**
  - Increase the number of keys



## Substitution in other forms: Monoalphabetic Ciphers

---

With only 25 possible keys, the Caesar cipher is far from secure. A dramatic increase in the key space can be achieved by allowing an arbitrary substitution (**Random substitution**).

*Permutation:* A **permutation** of a finite set of elements  $S$  is an ordered sequence of all the elements of  $S$ , with each element appearing exactly once.

For example, if  $S = \{a, b, c\}$ , there are six permutations of  $S$ :  
abc, acb, bac, bca, cab, cba  
For  $n$  elements,  $n!$  permutations.



---

## For Caesar cipher:

plain: 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

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

## For mono-alphabetic substitution:

If, instead, the “cipher” line can be any permutation of the 26 alphabetic characters,

then there are  $26!$  or greater than  $4 * 10^{26}$  possible keys



## Example

We can use the key in Figure in previous slide to encrypt the message

this message is easy to encrypt but hard to find the key

The ciphertext is

ICFVQRVVNEFVRNVSIYRGAHSLIOJICNHTIYBFGTICRXRS

# Attacks on Mono-alphabetic Ciphers

---

- ▶ Exploit the **regularities** of the language
  - Frequency of letters, digrams, trigrams
  - Expected words**
- ▶ Fundamental problem with mono-alphabetic ciphers
  - Ciphertext **reflects the frequency** data of original plaintext
  - Solution 1: encrypt multiple letters of plaintext
  - Solution 2: use multiple cipher alphabets



# Language Redundancy and Cryptanalysis

---

- Human languages are **redundant**  
e.g., "th lrd s m shphrd shll nt wnt"
- Letters are not equally commonly used
- In English E is by far the most common letter
  - followed by T,R,N,I,O,A,S
- Other letters like Z,J,K,Q,X are fairly rare
- Have tables of single, double & triple letter frequencies for various languages



# Continued

## Frequency of characters in English

Letter	Frequency	Letter	Frequency	Letter	Frequency	Letter	Frequency
E	12.7	H	6.1	W	2.3	K	0.08
T	9.1	R	6.0	F	2.2	J	0.02
A	8.2	D	4.3	G	2.0	Q	0.01
O	7.5	L	4.0	Y	2.0	X	0.01
I	7.0	C	2.8	P	1.9	Z	0.01
N	6.7	U	2.8	B	1.5		
S	6.3	M	2.4	V	1.0		

## Frequency of diagrams and trigrams

Digram	TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS, OR, TI, IS, ET, IT, AR, TE, SE, HI, OF
Trigram	THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, FOR, DTH



# Relative Frequency of Letters in English Text

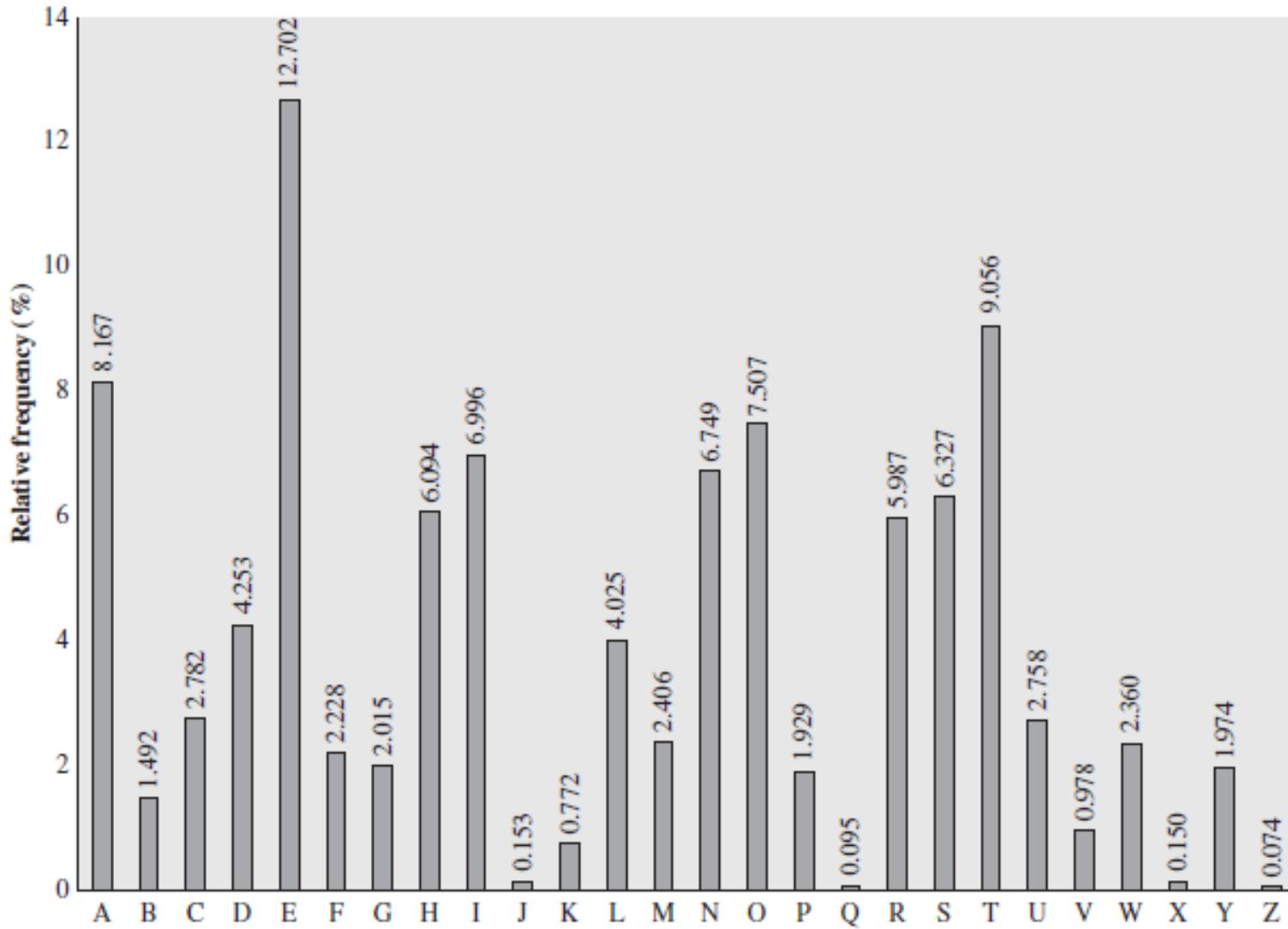


Figure 2.5 Relative Frequency of Letters in English Text

# Breaking Monoalphabetic Substitution Cipher

- ▶ **Letter Frequency Analysis results:**

Letter frequencies in English text

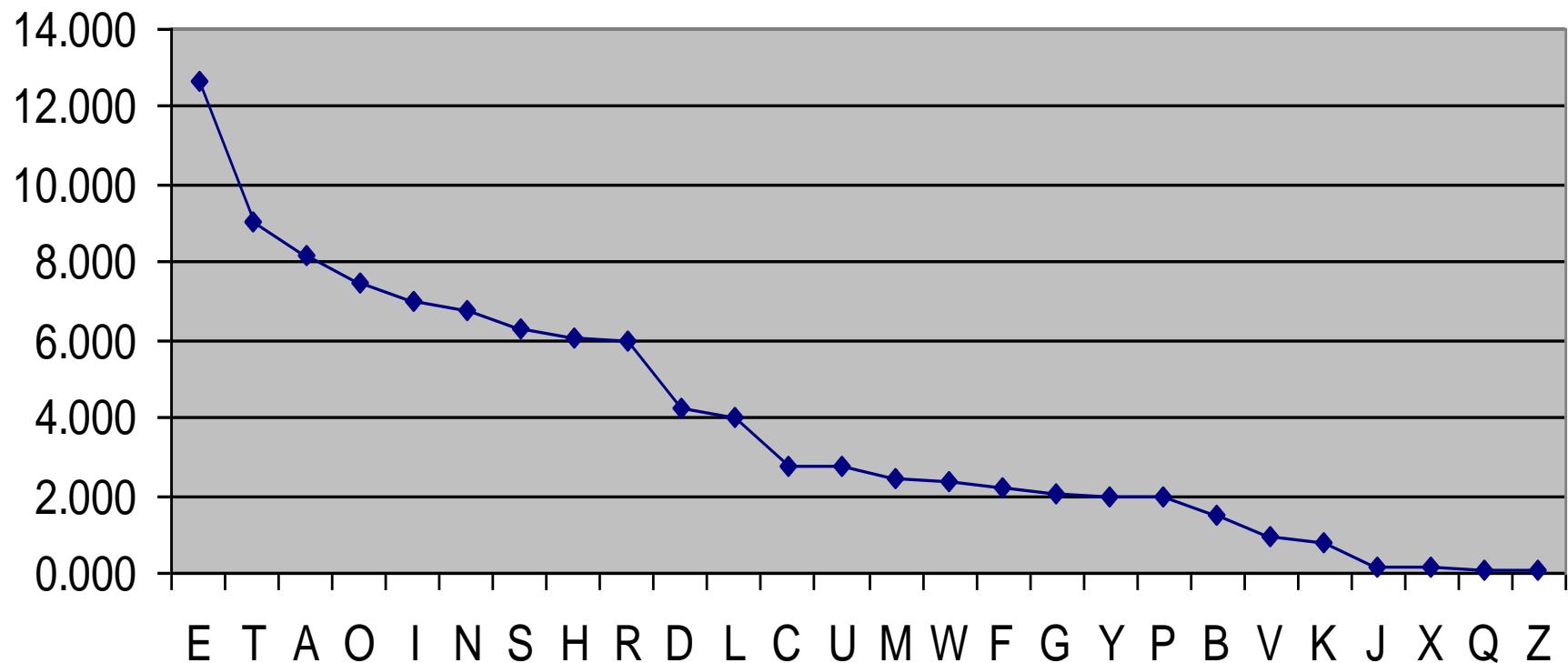


Letter frequencies in ciphertext



# English Letter Frequencies

## Sorted Relative Frequencies



## Example cryptanalysis

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ  
VUEPHZHMDZSHZOWSFPAAPPDTSPVQUZWYMXUZUHSX  
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

frequency

P 13.33	H 5.83	F 3.33	B 1.67	C 0.00
Z 11.67	D 5.00	W 3.33	G 1.67	K 0.00
S 8.33	E 5.00	Q 2.50	Y 1.67	L 0.00
U 8.33	V 4.17	T 2.50	I 0.83	N 0.00
O 7.50	X 4.17	A 1.67	J 0.83	R 0.00
M 6.67				

As a first step, the relative frequency of the letters can be determined and compared to a standard frequency distribution for English



So far, then, we have

UZQSOVUOHOXMOPVGPOZPEVSGZWSZOPFPESXUDBMETXAIZ

 t a            e e te a that e e a            a

VUEPHZHMDZSHZOWSFAPPDTSPQUZWYMXUZUHSX

 e t    ta t ha e ee a e th    t a

E PYEPOP DZSZUF POMBZWPFUPZHMDJUDTMOHMQ

 e e e tat e the t

# Example Cryptanalysis

- given ciphertext

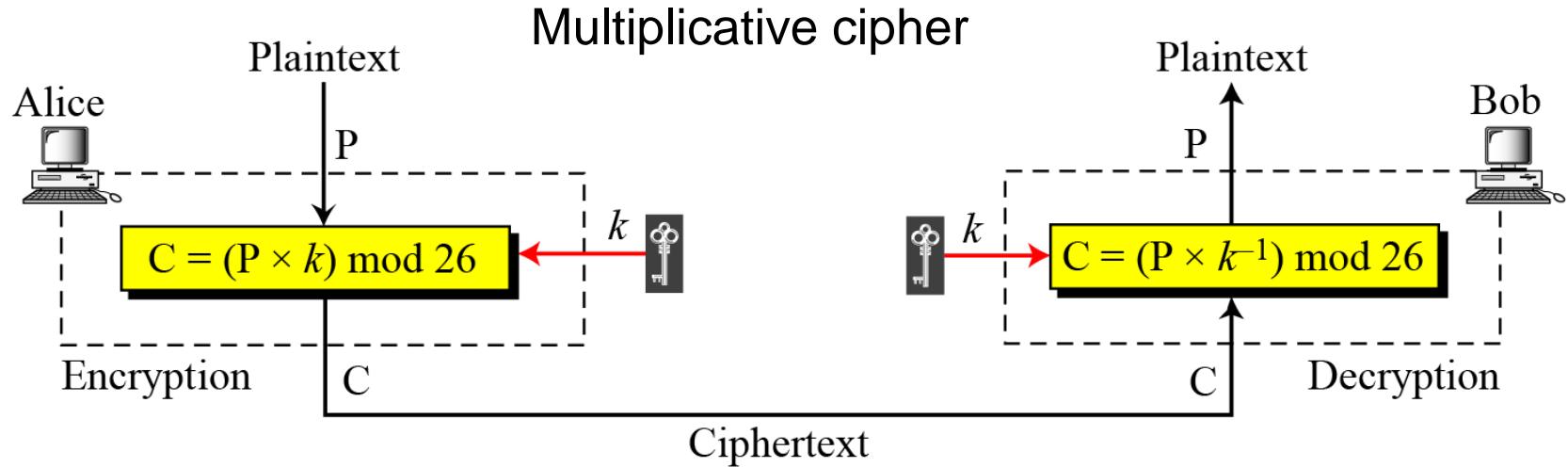
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ  
VUEPHZHMDZSHZOWSFAPPDTSVPQUZWYMXUZUHSX  
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

- guess P & Z are e and t
- guess ZW is th and hence ZWP is “the”
- proceeding with trial and error finally get:

it was disclosed yesterday that several informal but direct contacts have been made with political representatives of the viet cong in moscow



# Multiplicative Ciphers



In a multiplicative cipher, the plaintext and ciphertext are integers in  $Z_{26}$ ; the key is an integer in  $Z_{26}^*$ .



# Continued.

## Example

What is the key domain for any multiplicative cipher?

## Solution

The key needs to be in  $Z_{26}^*$ . This set has only 12 members: 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25.

## Example

We use a multiplicative cipher to encrypt the message “hello” with a key of 7. The ciphertext is “XCZZU”.

Plaintext: h → 07  
Plaintext: e → 04  
Plaintext: l → 11  
Plaintext: l → 11  
Plaintext: o → 14

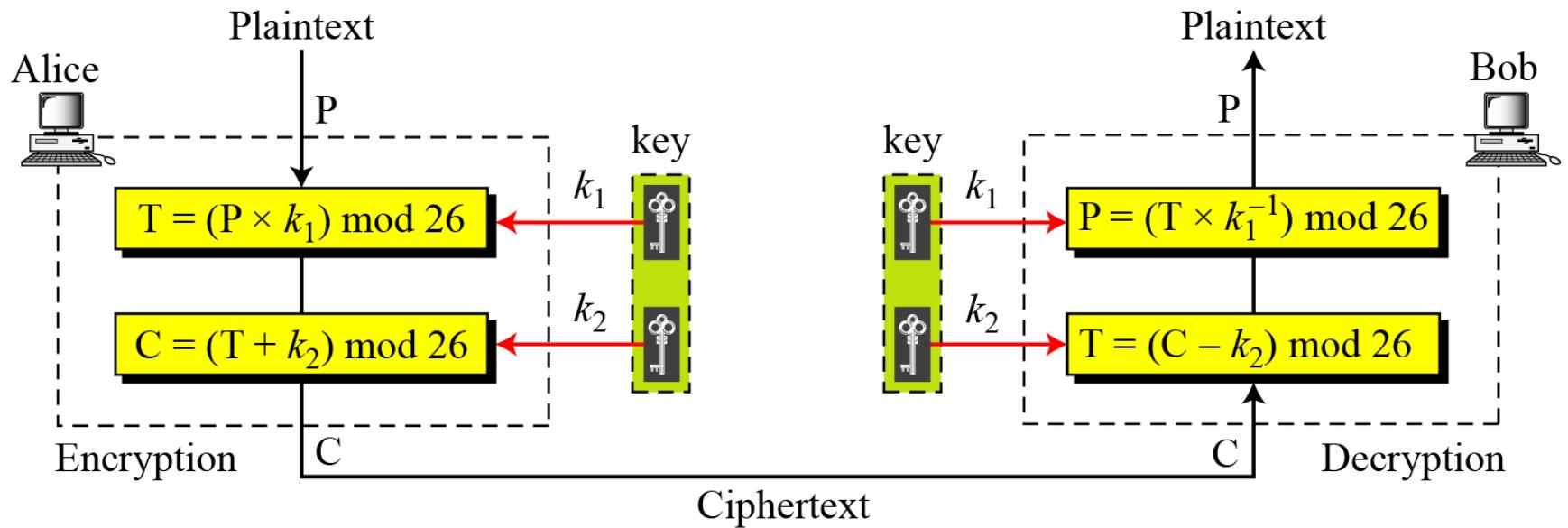
Encryption:  $(07 \times 07) \bmod 26$   
Encryption:  $(04 \times 07) \bmod 26$   
Encryption:  $(11 \times 07) \bmod 26$   
Encryption:  $(11 \times 07) \bmod 26$   
Encryption:  $(14 \times 07) \bmod 26$

ciphertext: 23 → X  
ciphertext: 02 → C  
ciphertext: 25 → Z  
ciphertext: 25 → Z  
ciphertext: 20 → U



# Affine Ciphers

Combining additive and multiplicative cipher



$$C = (P \times k_1 + k_2) \text{ mod } 26$$

$$P = ((C - k_2) \times k_1^{-1}) \text{ mod } 26$$

where  $k_1^{-1}$  is the multiplicative inverse of  $k_1$  and  $-k_2$  is the additive inverse of  $k_2$

# Affine Ciphers

## Example

The affine cipher uses a pair of keys in which the first key is from  $Z_{26}^*$  and the second is from  $Z_{26}$ . The size of the key domain is  $26 \times 12 = 312$ .

## Example

Use an affine cipher to encrypt the message “hello” with the key pair (7, 2).

P: h → 07

Encryption:  $(07 \times 7 + 2) \bmod 26$

C: 25 → Z

P: e → 04

Encryption:  $(04 \times 7 + 2) \bmod 26$

C: 04 → E

P: l → 11

Encryption:  $(11 \times 7 + 2) \bmod 26$

C: 01 → B

P: l → 11

Encryption:  $(11 \times 7 + 2) \bmod 26$

C: 01 → B

P: o → 14

Encryption:  $(14 \times 7 + 2) \bmod 26$

C: 22 → W



# Affine Ciphers

## Example

Use the affine cipher to decrypt the message “ZEBBW” with the key pair (7, 2) in modulus 26.

## Solution

C: Z → 25	Decryption: $((25 - 2) \times 7^{-1}) \bmod 26$	P: 07 → h
C: E → 04	Decryption: $((04 - 2) \times 7^{-1}) \bmod 26$	P: 04 → e
C: B → 01	Decryption: $((01 - 2) \times 7^{-1}) \bmod 26$	P: 11 → l
C: B → 01	Decryption: $((01 - 2) \times 7^{-1}) \bmod 26$	P: 11 → l
C: W → 22	Decryption: $((22 - 2) \times 7^{-1}) \bmod 26$	P: 14 → o

## Example

The additive cipher is a special case of an affine cipher in which  $k_1 = 1$ . The multiplicative cipher is a special case of affine cipher in which  $k_2 = 0$ .



# Polygraphic Substitution Ciphers

- In a **polygram** cipher blocks of characters in the plaintext are mapped to blocks of characters in the ciphertext:

$\text{ARF} \rightarrow \text{RTW}, \text{ING} \rightarrow \text{PWQ}, \dots$

- We represent the cipher with a **Substitution Box (S-Box)**:

	A	B	C	D	E	F
A	BA	CA	DC	DD	DE	FB
B	EA	AB	EC	BD	BE	AF
C	AA	BB	AC	ED	CE	BF
D	EB	DB	BC	CD	DF	FC
E	DA	CB	CC	AD	AE	FF
F	FA	CF	EE	FD	EF	FE

$\text{AA} \rightarrow \text{BA}$

- Examples:  $\text{AB} \rightarrow \text{CA}$

$\text{EF} \rightarrow \text{FF}$

## **Substitution: Other forms (Cont)**

---

- Use two-letter combinations: Playfair Cipher
- Use multiple letter combinations: Hill Cipher

Use multiple ciphers. Use a key to select which alphabet (code) is used for each letter of the message



# Poly-alphabetic Ciphers

---

- ▶ Monoalphabetic ciphers are easy to break because they reflect the frequency data of the original alphabet. A countermeasure is to provide multiple substitutes known as homophones, for a single letter.
- ▶ For example, the letter e could be assigned a number of different cipher symbols, such as 16, 74, 35, and 21, with each homophone assigned to a letter in rotation or randomly.
- ▶ Use different mono-alphabetic substitutions as proceed through plaintext
  - Set of mono-alphabetic ciphers
  - Key determines which mono-alphabetic cipher to use for each plaintext letter
- ▶ Examples: Vigenere cipher, Vernam cipher, One time pad



# Polyalphabetic Ciphers

In polyalphabetic substitution, each occurrence of a character may have a different substitute. The relationship between a character in the plaintext to a character in the ciphertext is one-to-many.

## Autokey Cipher

$$P = P_1 P_2 P_3 \dots$$

$$C = C_1 C_2 C_3 \dots$$

$$k = (k_1, P_1, P_2, \dots)$$

Encryption:  $C_i = (P_i + k_i) \bmod 26$

Decryption:  $P_i = (C_i - k_i) \bmod 26$



# Autokey Cipher

Assume that Alice and Bob agreed to use an autokey cipher with initial key value  $k_1 = 12$ . Now Alice wants to send Bob the message “Attack is today”. Enciphering is done character by character.

Plaintext:	a	t	t	a	c	k	i	s	t	o	d	a	y
P's Values:	00	19	19	00	02	10	08	18	19	14	03	00	24
Key stream:	12	00	19	19	00	02	10	08	18	19	14	03	00
C's Values:	12	19	12	19	02	12	18	00	11	7	17	03	24
Ciphertext:	M	T	M	T	C	M	S	A	L	H	R	D	Y



# Autokey Cipher

- ▶ Vigenère proposed what is referred to as an **autokey system**, in which a keyword is concatenated with the plaintext itself to provide a running key. For our example,

<b>key:</b>	<i>deceptivewearediscoveredsav</i>
<b>plaintext:</b>	<i>wearediscoveredsaveyourself</i>
<b>ciphertext:</b>	<b>ZICVTWQNGKZEIIIGASKSTSLVVWLA</b>

Even this scheme is vulnerable to cryptanalysis. Because the key and the plaintext share the same frequency distribution of letters, a statistical technique can be applied.



# Playfair Cipher

---

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



# Playfair Cipher

## ► Initialization

1. Create 5x5 matrix and write keyword (row by row)
2. Fill out remainder with alphabet, not repeating any letters
3. Special: Treat I and J as same letter

## ► Encryption

1. Operate on pair of letters (digram) at a time
2. Special: if digram with same letters, separate by special letter (e.g. **X**)
3. Plaintext in **same row**: replace with letters **to right**
4. Plaintext in **same column**: replace with letters **below**
5. Else, replace by letter in same row as it and same column as other plaintext letter

# Playfair Cipher

- Rules to encrypt the digraph  $\alpha\beta$ :

- If  $\alpha = \beta$ , add an **X**, encrypt the new pair.
- If one letter is left, add an **X**, encrypt the new pair.
- If  $\alpha, \beta$  are in the same row:

*	*	*	*	*
*	*	*	*	*
$\alpha$	<b>X</b>	*	$\beta$	<b>Y</b>
*	*	*	*	*
*	*	*	*	*

$$\Rightarrow \alpha\beta \rightarrow XY$$

If necessary, wrap around.

- If  $\alpha\beta$  occur in the same column:

*	*	*	*	*
*	*	$\alpha$	*	*
*	*	<b>X</b>	*	*
*	*	$\beta$	*	*
*	*	<b>Y</b>	*	*

$$\Rightarrow \alpha\beta \rightarrow XY$$

# Playfair Cipher

- And the final rule:

- ⑤ If the letters are not on the same row or column:

X	*	*	$\alpha$	*
*	*	*	*	*
*	*	*	*	*
$\beta$	*	Y	*	*
*	*	*	*	*

$\Rightarrow \alpha\beta \rightarrow XY$

Order matters: X is on the same row as  $\alpha$ .

- To decrypt:

- ① Use the inverse of the last three rules.
- ② Drop any Xs that don't make sense.



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

balloon	<b>balxloxon</b>
ar	RM
mu	CM
hs	BP
ea	IM

In this case, the keyword is *monarchy*.

Plaintext is encrypted two letters at a time



# An example of a secret key in the Playfair cipher

Secret Key =

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

## Example

Let us encrypt the plaintext “hello” using the key in Figure

he → EC

Plaintext: hello

lx → QZ

Ciphertext: ECQZBX



# An example of a secret key in the Playfair cipher

- Example plaintext:

IT WA SA DA RK AN DS TO RM YN IG HT

- IT→MP

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

- WA→XI

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



# An example of a secret key in the Playfair cipher

- SA→XG

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

- DA→IM

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



## Exercise

---

- ① Construct a Playfair table using the key phrase **BLINKENLIGHTS**.
- ② Encode the message **Run, RAbbit, Run!**
- ③ Encrypt the plaintext message from 2.
- ④ Decrypt the ciphertext message from 3.



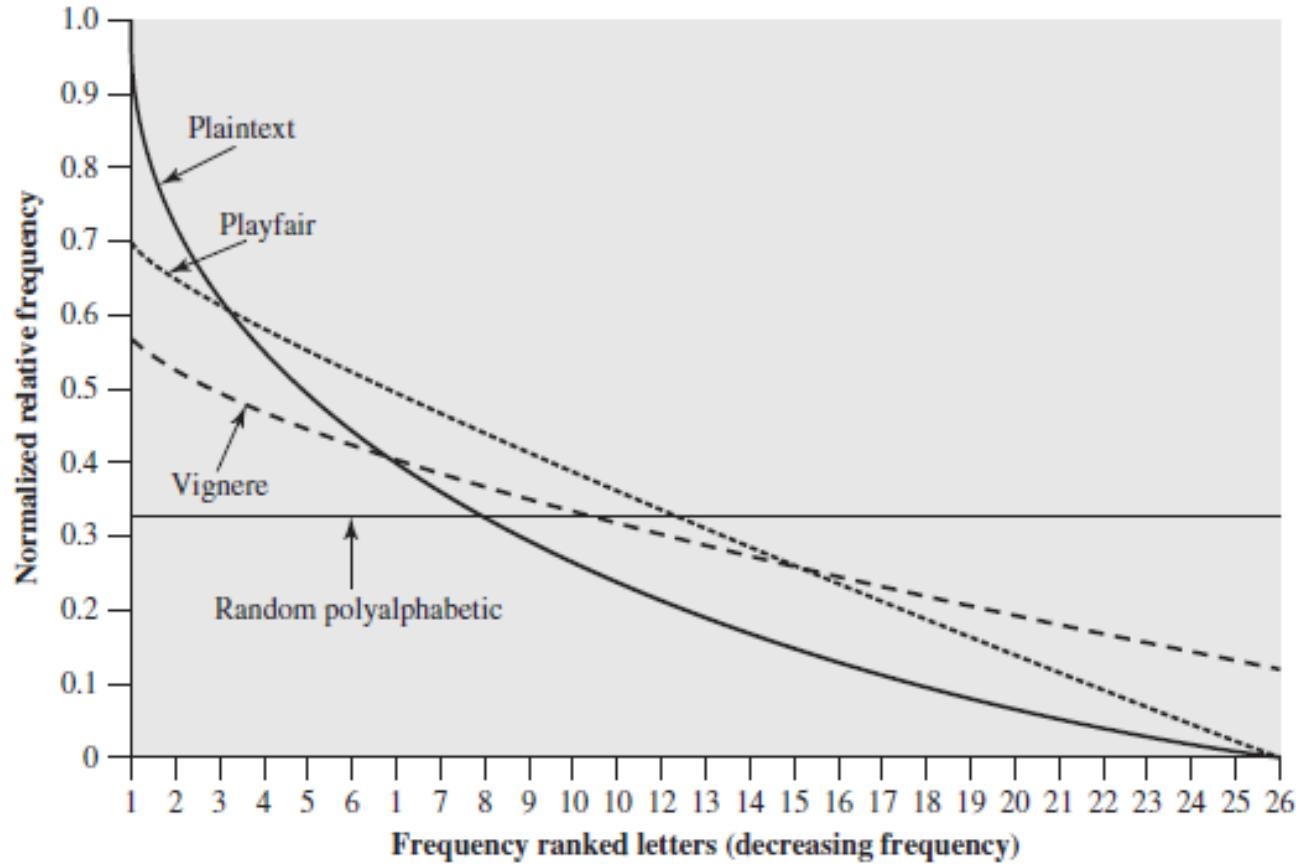
# Measuring Effectiveness of the Playfair and other ciphers

---

- ▶ The following plot is developed: (i) The number of occurrences of each letter in the text is counted and divided by the number of occurrences of the most frequently used letter. e is the most frequently used letter.
- ▶ To normalize the plot, the number of occurrences of each letter in the ciphertext was again divided by the number of occurrences of e in the plaintext.
- ▶ If the frequency distribution information were totally concealed in the encryption process, the ciphertext plot of frequencies would be flat, and cryptanalysis using ciphertext only would be effectively impossible.



# Relative Frequency of Occurrence of Letters



# Security of Playfair Cipher

---

- ▶ Security much improved over monoalphabetic since have  $26 \times 26 = 676$  digrams
- ▶ Would need a 676 entry frequency table to analyse (versus 26 for a monoalphabetic) and correspondingly more ciphertext was widely used for many years
- ▶ E.g. by US & British military in WW1
- ▶ It can be broken, given a few hundred letters since still has much of plaintext structure



# Playfair Cipher - Is it Breakable?

---

- ▶ Better than mono-alphabetic: relative frequency of digrams much less than of individual letters
- ▶ But relatively easy (digrams, trigrams, expected words)



# Vigenere Cipher

---

- ▶ Set of 26 general Caesar ciphers  
[26 Caesar ciphers with shifts of 0 through 25.](#)
- ▶ Letter in key determines the Caesar cipher to use
  - Key must be as long as plaintext: repeat a keyword
  - Key: pqr
  - Plaintext: attack is today
- ▶ Example:

Plain: a t t a c k i s t o d a y

Key: p q r p q r p q r p q r p

Cipher:

Multiple ciphertext letters for each plaintext letter



$$\begin{matrix} P = p_0, p_1, p_2, \dots, p_{n-1} \\ \vdots \end{matrix}$$

$$K = k_0, k_1, k_2, \dots, k_{m-1}, \text{ where typically } m < n$$

$$C = C_0, C_1, C_2, \dots, C_{n-1}$$

$$\begin{aligned} C &= C_0, C_1, C_2, \dots, C_{n-1} = E(K, P) = E[(k_0, k_1, k_2, \dots, k_{m-1}), (p_0, p_1, p_2, \dots, p_{n-1})] \\ &= (p_0 + k_0) \bmod 26, (p_1 + k_1) \bmod 26, \dots, (p_{m-1} + k_{m-1}) \bmod 26, \\ &\quad (p_m + k_0) \bmod 26, (p_{m+1} + k_1) \bmod 26, \dots, (p_{2m-1} + k_{m-1}) \bmod 26, \dots \end{aligned}$$

For the next *m* letters of the plaintext, the key letters are repeated.



$$C_i = (p_i + k_{i \bmod m}) \bmod 26$$

$$p_i = (C_i - k_{i \bmod m}) \bmod 26$$

key:

deceptivedeceptivedeceptive

plaintext:

wearediscoveredsaveyourself

ciphertext:

ZICVTWQNGRZGVTWAVZHCQYGLMGJ

Expressed numerically, we have the following result.

key	3	4	2	4	15	19	8	21	4	3	4	2	4	15
plaintext	22	4	0	17	4	3	8	18	2	14	21	4	17	4
ciphertext	25	8	2	21	19	22	16	13	6	17	25	6	21	19

key	19	8	21	4	3	4	2	4	15	19	8	21	4
plaintext	3	18	0	21	4	24	14	20	17	18	4	11	5
ciphertext	22	0	21	25	7	2	16	24	6	11	12	6	9

# Vigenere Cipher

---

$$P = P_1 P_2 P_3 \dots$$

$$C = C_1 C_2 C_3 \dots$$

$$K = [(k_1, k_2, \dots, k_m), (k_1, k_2, \dots, k_m), \dots]$$

$$\text{Encryption: } C_i = P_i + k_i$$

$$\text{Decryption: } P_i = C_i - k_i$$



# Vigenere Cipher

## Example

We can encrypt the message “She is listening” using the 6-character keyword “PASCAL”.

Let us see how we can encrypt the message “She is listening” using the 6-character keyword “PASCAL”. The initial key stream is (15, 0, 18, 2, 0, 11). The key stream is the repetition of this initial key stream (as many times as needed).

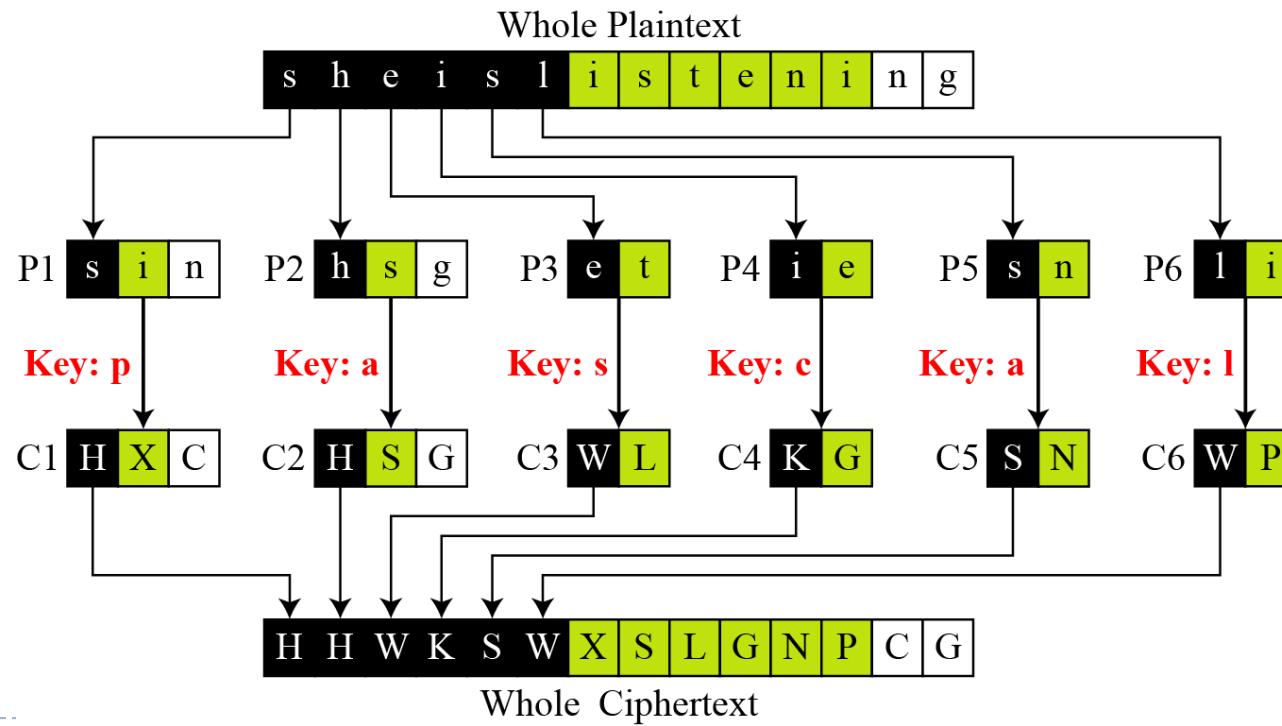
Plaintext:	s	h	e	i	s	l	i	s	t	e	n	i	n	g
P's values:	18	07	04	08	18	11	08	18	19	04	13	08	13	06
Key stream:	<b>15</b>	<b>00</b>	<b>18</b>	<b>02</b>	<b>00</b>	<b>11</b>	<b>15</b>	<b>00</b>	<b>18</b>	<b>02</b>	<b>00</b>	<b>11</b>	<b>15</b>	<b>00</b>
C's values:	<b>07</b>	<b>07</b>	<b>22</b>	<b>10</b>	<b>18</b>	<b>22</b>	<b>23</b>	<b>18</b>	<b>11</b>	<b>6</b>	<b>13</b>	<b>19</b>	<b>02</b>	<b>06</b>
Ciphertext:	<b>H</b>	<b>H</b>	<b>W</b>	<b>K</b>	<b>S</b>	<b>W</b>	<b>X</b>	<b>S</b>	<b>L</b>	<b>G</b>	<b>N</b>	<b>T</b>	<b>C</b>	<b>G</b>



# Vigenere Cipher

Vigenere cipher can be seen as combinations of m additive ciphers.

Figure A Vigenere cipher as a combination of m additive ciphers



# Vigenere Cipher

Using Example , we can say that the additive cipher is a special case of Vigenere cipher in which  $m = 1$ .

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

Table  
A Vigenere Tableau



# Vigenere Cipher

(Cryptanalysis)

## Example

Let us assume we have intercepted the following ciphertext:

LIOMWGFEGGDVWGHHCQUCRHRWAGWIOWQLKGZETKKMEVLWPCZVGTH-  
VTSGXQOVGCSVETQLTJSUMVVVEUVLXEWSLGZMVVWLGYHCUSWXQH-  
KVGSHEEVFLCFDGVSUMPHKIRZDMPHHBVWWJWIXGFWLTSHGJOUEEHH-  
VUCFVGOWICQLTJSUXGLW

The Kasiski test for repetition of three-character segments yields the results shown in Table .

<i>String</i>	<i>First Index</i>	<i>Second Index</i>	<i>Difference</i>
JSU	68	168	100
SUM	69	117	48
VWV	72	132	60
MPH	119	127	8

# Vigenere Cipher

(Crypanalysis)

## Example Cont.

Let us assume we have intercepted the following ciphertext:

LIOMWGEGGDVWGHHCQUCRHRWAGWIOWQLKGZETKKMEVLWPCZVGTH-  
VTSGXQOVGCSVETQLTJSUMVWVEUVLXEWSLGFMVVWLGYHCUSWXQH-  
KVGSHEEVFLCFDGVSUMPHKIRZDMPHHBVWVWJWIXGFWLTSHGJOUEEHH-  
VUCFVGOWICQLTJSUXGLW

The **Kasiski test** for repetition of three-character segments in ciphertext yields the results shown in Table

<i>String</i>	<i>First Index</i>	<i>Second Index</i>	<i>Difference</i>
JSU	68	168	100
SUM	69	117	48
VWV	72	132	60
MPH	119	127	8



## Example Cont.

The greatest common divisor of differences is 4, which means that the key length is multiple of 4. First try  $m = 4$ .

C1 : LWGWCRAOKTEPGTQCTJVUEGVGUQGECVPRPVJGTJEUGCJG

P1 : jueuapymircneroarhtsthihytrahcieixsthcarrehe

C2 : IGGGQHGWGKVCTSOSQS WVWFVYSHSVFSHZHWWFSOHCOQSL

P2 : ussscts is who feaeceihcetes oecat npn ther hctecex

C3 : OFDHURWQZKLZHGVVLUVLSZWHWKHF DUKDHVIWHUHF WL UW

P3 : lcaerotnwhi wed ssir si irh keteh retl ti ideat rair t

C4 : MEVHCWILEMWVVXGETMEXMLCXVELGMIMBWXLGEVVITX

P4 : i ard ysehaisrrt capia fpwt eth carha esf terect pt

In this case, the plaintext makes sense.

---

Julius Caesar used a cryptosystem in his wars, which is now referred to as Caesar cipher.

It is an additive cipher with the key set to three. Each character in the plaintext is shifted three characters to create ciphertext.

---

# Vigenere Cipher - Is it Breakable?

---

- ▶ Yes
  - ▶ Monoalphabetic or Vigenere cipher? Letter frequency analysis
  - ▶ Determine length of keyword
  - ▶ For keyword length  $m$ , Vigenere is  $m$  mono-alphabetic substitutions
  - ▶ Break the mono-alphabetic ciphers separately
- Weakness is repeating, structured keyword



# Hill Cipher

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

Plaintext are divided into equal size blocks. Each character in a block contributes to the encryption of the other characters in the block. (**Block Cipher**)

$$K = \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1m} \\ k_{21} & k_{22} & \dots & k_{2m} \\ \vdots & \vdots & & \vdots \\ k_{m1} & k_{m2} & \dots & k_{mm} \end{bmatrix}$$

$$C_1 = P_1 k_{11} + P_2 k_{21} + \dots + P_m k_{m1}$$

$$C_2 = P_1 k_{12} + P_2 k_{22} + \dots + P_m k_{m2}$$

...

$$C_m = P_1 k_{1m} + P_2 k_{2m} + \dots + P_m k_{mm}$$

The **key matrix** in the Hill cipher needs to have a multiplicative inverse.

## Exam Hilll1

For example, the plaintext “code is ready” can make a  $3 \times 4$  matrix when adding extra bogus character “z” to the last block and removing the spaces. The ciphertext is “OHKNIHGKLSS”.

Figure Example

$$\begin{bmatrix} 14 & 07 & 10 & 13 \\ 08 & 07 & 06 & 11 \\ 11 & 08 & 18 & 18 \end{bmatrix}^C = \begin{bmatrix} 02 & 14 & 03 & 04 \\ 08 & 18 & 17 & 04 \\ 00 & 03 & 24 & 25 \end{bmatrix}^P = \begin{bmatrix} 09 & 07 & 11 & 13 \\ 04 & 07 & 05 & 06 \\ 02 & 21 & 14 & 09 \\ 03 & 23 & 21 & 08 \end{bmatrix}^K$$

a. Encryption

$$\begin{bmatrix} 02 & 14 & 03 & 04 \\ 08 & 18 & 17 & 04 \\ 00 & 03 & 24 & 25 \end{bmatrix}^P = \begin{bmatrix} 14 & 07 & 10 & 13 \\ 08 & 07 & 06 & 11 \\ 11 & 08 & 18 & 18 \end{bmatrix}^C = \begin{bmatrix} 02 & 15 & 22 & 03 \\ 15 & 00 & 19 & 03 \\ 09 & 09 & 03 & 11 \\ 17 & 00 & 04 & 07 \end{bmatrix}^{K^{-1}}$$

b. Decryption

## Example Hill2

Assume that Eve knows that  $m = 3$ . She has intercepted three plaintext/ciphertext pair blocks (not necessarily from the same message) as shown in Figure .

Figure

$$\begin{bmatrix} 05 & 07 & 10 \end{bmatrix} \longleftrightarrow \begin{bmatrix} 03 & 06 & 00 \end{bmatrix}$$

$$\begin{bmatrix} 13 & 17 & 07 \end{bmatrix} \longleftrightarrow \begin{bmatrix} 14 & 16 & 09 \end{bmatrix}$$

$$\begin{bmatrix} 00 & 05 & 04 \end{bmatrix} \longleftrightarrow \begin{bmatrix} 03 & 17 & 11 \end{bmatrix}$$

P

C



## Example Hill2 cont.

She makes matrices P and C from these pairs. Because P is invertible, she inverts the P matrix and multiplies it by C to get the K matrix as shown in Figure.

Figure Example

$$\begin{bmatrix} 02 & 03 & 07 \\ 05 & 07 & 09 \\ 01 & 02 & 11 \end{bmatrix} = \begin{bmatrix} 21 & 14 & 01 \\ 00 & 08 & 25 \\ 13 & 03 & 08 \end{bmatrix} \begin{bmatrix} 03 & 06 & 00 \\ 14 & 16 & 09 \\ 03 & 17 & 11 \end{bmatrix}$$

**K**                    **P<sup>-1</sup>**                    **C**

Now she has the key and can break any ciphertext encrypted with that key.



# Hill Cipher

## ► Concepts from Linear Algebra

We define the inverse  $\mathbf{M}^{-1}$  of a square matrix  $\mathbf{M}$  by the equation  $\mathbf{M}(\mathbf{M}^{-1}) = \mathbf{M}^{-1}\mathbf{M} = \mathbf{I}$ , where  $\mathbf{I}$  is the identity matrix.  $\mathbf{I}$  is a square matrix that is all zeros except for ones along the main diagonal from upper left to lower right.

(we are concerned with matrix arithmetic modulo 26).

$$\mathbf{A} = \begin{pmatrix} 5 & 8 \\ 17 & 3 \end{pmatrix} \quad \mathbf{A}^{-1} \bmod 26 = \begin{pmatrix} 9 & 2 \\ 1 & 15 \end{pmatrix}$$

$$\begin{aligned} \mathbf{A}\mathbf{A}^{-1} &= \begin{pmatrix} (5 \times 9) + (8 \times 1) & (5 \times 2) + (8 \times 15) \\ (17 \times 9) + (3 \times 1) & (17 \times 2) + (3 \times 15) \end{pmatrix} \\ &= \begin{pmatrix} 53 & 130 \\ 156 & 79 \end{pmatrix} \bmod 26 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{aligned}$$

# Matrix Operations

- ▶ Matrix addition/subtraction
  - ▶ Matrices must be of same size.
- ▶ Matrix multiplication

$$\begin{array}{c} \text{m} \times \text{n} \\ \left[ \begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{array} \right] \end{array} \begin{array}{c} \text{n} \times \text{p} \\ \left[ \begin{array}{cccc} b_{11} & b_{12} & \dots & b_{1p} \\ b_{21} & b_{22} & \dots & b_{2p} \\ \dots & \dots & \dots & \dots \\ b_{q1} & b_{q2} & \dots & b_{qp} \end{array} \right] \end{array} = \begin{array}{c} \text{m} \times \text{p} \\ \left[ \begin{array}{cccc} c_{11} & c_{12} & \dots & c_{1p} \\ c_{21} & c_{22} & \dots & c_{2p} \\ \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \dots & c_{mp} \end{array} \right] \end{array}$$

**Condition:** n = q       $AB \neq BA$        $c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$



# Identity Matrix

$$AI = IA = A, \text{ where } I = \begin{bmatrix} 1 & 0 & . & 0 \\ 0 & 1 & . & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & . & 1 \end{bmatrix}$$



# Matrix Transpose

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdot & a_{1n} \\ a_{21} & a_{22} & \cdot & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \cdot & a_{mn} \end{bmatrix}, A^T = \begin{bmatrix} a_{11} & a_{21} & \cdot & a_{m1} \\ a_{12} & a_{22} & \cdot & a_{m2} \\ \dots & \dots & \dots & \dots \\ a_{1n} & a_{2n} & \cdot & a_{mn} \end{bmatrix}$$

Property:  $(AB)^T = B^T A^T$



# Symmetric Matrices

---

$$A = A^T \quad (a_{ij} = a_{ji})$$

Example:

$$\begin{bmatrix} 4 & 5 & -3 \\ 5 & 7 & 2 \\ -3 & 2 & 10 \end{bmatrix}$$



# Determinants

**2 × 2**

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \det(A) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{21}a_{12}$$

**3 × 3**

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{21} \begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} + a_{31} \begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix}$$

**n × n**

$$\det(A) = \sum_{j=1}^m (-1)^{j+k} a_{jk} \det(A_{jk}), \text{ for any } k: 1 \leq k \leq m$$

**Properties:**

$$\det(AB) = \det(A)\det(B)$$

$$\det(A + B) \neq \det(A) + \det(B)$$



# Matrix Inverse

---

- ▶ The inverse  $A^{-1}$  of a matrix  $A$  has the property:

$$AA^{-1} = A^{-1}A = I$$

- ▶  $A^{-1}$  exists only if

$$\det(A) \neq 0$$

- ▶ Terminology

- ▶ **Singular matrix:**  $A^{-1}$  does not exist
- ▶ **Ill-conditioned matrix:**  $A$  is “close” to being singular



# Matrix Inverse (cont'd)

---

- ▶ Properties of the inverse:

$$\det(A^{-1}) = \frac{1}{\det(A)}$$

$$(AB)^{-1} = B^{-1} A^{-1}$$

$$(A^T)^{-1} = (A^{-1})^T$$



# Inverse of a Matrix

## Determinant

For any square matrix ( $m \times m$ ), the determinant equals the sum of all the products that can be formed by taking exactly one element from each row and exactly one element from each column, with certain of the product terms preceded by a minus sign

$$\begin{pmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{pmatrix}$$

the determinant is  $k_{11}k_{22} - k_{12}k_{21}$ . For a  $3 \times 3$  matrix, the value of the determinant is  $k_{11}k_{22}k_{33} + k_{21}k_{32}k_{13} + k_{31}k_{12}k_{23} - k_{31}k_{22}k_{13} - k_{21}k_{12}k_{33} - k_{11}k_{32}k_{23}$ . If a square matrix  $\mathbf{A}$  has a nonzero determinant, then the inverse of the matrix is computed as  $[\mathbf{A}^{-1}]_{ij} = (\det \mathbf{A})^{-1}(-1)^{i+j}(D_{ji})$ , where  $(D_{ji})$  is the subdeterminant formed by deleting the  $j$ th row and the  $i$ th column of  $\mathbf{A}$ ,  $\det(\mathbf{A})$  is the determinant of  $\mathbf{A}$ , and  $(\det \mathbf{A})^{-1}$  is the multiplicative inverse of  $(\det \mathbf{A}) \bmod 26$ .



$$\mathbf{A} = \begin{pmatrix} 5 & 8 \\ 17 & 3 \end{pmatrix} \quad \mathbf{A}^{-1} \bmod 26 = \begin{pmatrix} 9 & 2 \\ 1 & 15 \end{pmatrix}$$

$$\begin{aligned}\mathbf{A}\mathbf{A}^{-1} &= \begin{pmatrix} (5 \times 9) + (8 \times 1) & (5 \times 2) + (8 \times 15) \\ (17 \times 9) + (3 \times 1) & (17 \times 2) + (3 \times 15) \end{pmatrix} \\ &= \begin{pmatrix} 53 & 130 \\ 156 & 79 \end{pmatrix} \bmod 26 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\end{aligned}$$

$$\text{Det}(\mathbf{A}) = \det \begin{pmatrix} 5 & 8 \\ 17 & 3 \end{pmatrix} = (5 \times 3) - (8 \times 17) = -121 \bmod 26 = 9$$

$$\text{Det}(\mathbf{A})^{-1} = 9^{-1} \bmod 26 = 3 \text{ (not } 1/9\text{)} \quad \text{As, } 9 \times 3 = 27 \bmod 26 = 1$$

**Note:**  $[\mathbf{A}^{-1}]_{ij} = (\det \mathbf{A})^{-1} (-1)^{i+j} (\mathbf{D}_{ji})$



$$\text{Det}(A) = \det \begin{pmatrix} 5 & 8 \\ 17 & 3 \end{pmatrix} = (5 \times 3) - (8 \times 17) = -121 \bmod 26 = 9$$

$$\text{Det}(A)^{-1} = 9^{-1} \bmod 26 = 3 \quad \text{As, } 9 \times 3 = 27 \bmod 26 = 1$$

$$A = \begin{pmatrix} 5 & 8 \\ 17 & 3 \end{pmatrix}$$

$$A^{-1} \bmod 26 = 3 \begin{pmatrix} 3 & -8 \\ -17 & 5 \end{pmatrix} = 3 \begin{pmatrix} 3 & 18 \\ 9 & 5 \end{pmatrix} = \begin{pmatrix} 9 & 54 \\ 27 & 15 \end{pmatrix} = \begin{pmatrix} 9 & 2 \\ 1 & 15 \end{pmatrix}$$

$$[A^{-1}]_{ij} = (\det A)^{-1} (-1)^{i+j} (D_{ji})$$



$$\det \begin{pmatrix} 5 & 8 \\ 17 & 3 \end{pmatrix} = (5 \times 3) - (8 \times 17) = -121 \bmod 26 = 9$$

We can show that  $9^{-1} \bmod 26 = 3$ , because  $9 \times 3 = 27 \bmod 26 = 1$  Therefore, we compute the inverse of  $\mathbf{A}$  as

$$\mathbf{A} = \begin{pmatrix} 5 & 8 \\ 17 & 3 \end{pmatrix}$$
$$\mathbf{A}^{-1} \bmod 26 = 3 \begin{pmatrix} 3 & -8 \\ -17 & 5 \end{pmatrix} = 3 \begin{pmatrix} 3 & 18 \\ 9 & 5 \end{pmatrix} = \begin{pmatrix} 9 & 54 \\ 27 & 15 \end{pmatrix} = \begin{pmatrix} 9 & 2 \\ 1 & 15 \end{pmatrix}$$

## The Hill Algorithm

This encryption algorithm takes  $m$  successive plaintext letters and substitutes for them  $m$  ciphertext letters. The substitution is determined by  $m$  linear equations in which each character is assigned a numerical value ( $a = 0, b = 1, c, \dots, z = 25$ ). For  $m = 3$ , the system can be described as

$$c_1 = (k_{11}p_1 + k_{21}p_2 + k_{31}p_3) \bmod 26$$

$$c_2 = (k_{12}p_1 + k_{22}p_2 + k_{32}p_3) \bmod 26$$

$$c_3 = (k_{13}p_1 + k_{23}p_2 + k_{33}p_3) \bmod 26$$

This can be expressed in terms of row vectors and matrices:<sup>6</sup>

$$(c_1 \ c_2 \ c_3) = (p_1 \ p_2 \ p_3) \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \bmod 26$$

or

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



## In general

---

$$\mathbf{C} = E(\mathbf{K}, \mathbf{P}) = \mathbf{PK} \bmod 26$$

$$\mathbf{P} = D(\mathbf{K}, \mathbf{C}) = \mathbf{CK}^{-1} \bmod 26 = \mathbf{PKK}^{-1} = \mathbf{P}$$

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

“paymoremoney”

The first three letters of the plaintext are represented by the vector  $(15 \ 0 \ 24)$ . Then  $(15 \ 0 \ 24)\mathbf{K} = (303 \ 303 \ 531) \bmod 26 = (17 \ 17 \ 11) = RRL$ . Continuing in this fashion, the ciphertext for the entire plaintext is RRLMWBKASPDH.

---



# One-Time Pad

---

- One of the goals of cryptography is **perfect secrecy**.
- A study by Shannon has shown that perfect secrecy can be achieved if each plaintext symbol is encrypted with a key **randomly chosen** from a key domain.
- This idea is used in a cipher called **one-time pad**, invented by **Vernam**.



# One Time Pad

---

- ▶ Similar to Vigenere, but use **random key as long as plaintext**
- ▶ Only known scheme that is unbreakable (**unconditional security or perfect security**)
  - Ciphertext **has no statistical relationship with plaintext**
  - Given **two potential plaintext messages**, attacker cannot identify the correct message

A cipher system has perfect secrecy if the ciphertext gives the cryptanalyst no information about the key. The one time pad achieves perfect secrecy.



# Continued

---

- ▶ Mauborgne suggested using a random key that is as long as the message, so that the key need not be repeated.
- ▶ In addition, the key is to be used to encrypt and decrypt a single message, and then is discarded.
- ▶ Each new message requires a new key of the same length as the new message.
- ▶ Such a scheme, known as a **one-time pad**, **is unbreakable**.
  
- ▶ Two practical limitations:
  1. Difficult to provide **large number of random keys**
  2. **Distributing** unique long random keys is difficult
- ▶ Limited practical use



## Ciphertext

**ciphertext:** ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUFPLUYTS

We now show two different decryptions using two different keys:

**ciphertext:** ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUFPLUYTS

**key:** pxlmvmsydoфuyrvzwc tnleбnecvgdупahfzzlmnyih

**plaintext:** mr mustard with the candlestick in the hall

**ciphertext:** ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUFPLUYTS

**key:** pftgpmiydgaxgoufhklllhmhsqdqogtewbqfgyovuhwt

**plaintext:** miss scarlet with the knife in the library

So, for using random key, the cryptanalyst will be confused.



# Vernam Cipher

- The ultimate defense against such a cryptanalysis is to choose a keyword that is as long as the plaintext and has no statistical relationship to it.

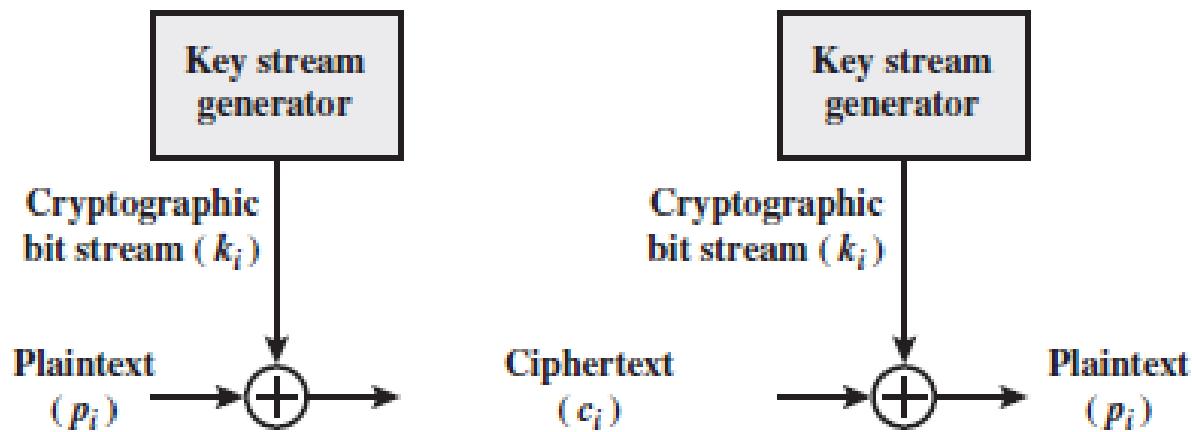


Figure 2.7 Vernam Cipher

His system works on binary data (bits) rather than letters. The system can be expressed succinctly as follows

$$c_i = p_i \oplus k_i$$

where

$p_i$  =  $i$ th binary digit of plaintext

$k_i$  =  $i$ th binary digit of key

$c_i$  =  $i$ th binary digit of ciphertext

$\oplus$  = exclusive-or (XOR) operation

$$p_i = c_i \oplus k_i$$

Vernam proposed the use of a running loop of tape that eventually repeated the key, so that in fact the system worked with a very long but repeating keyword.

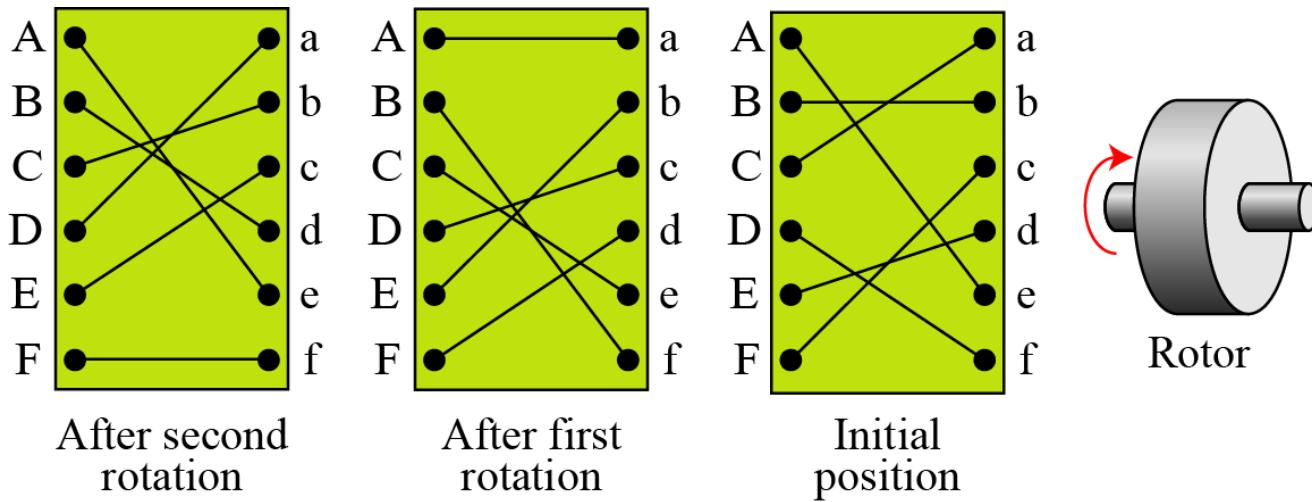


# Rotor Cipher

Although one-time pad is not practical, one step toward more secured encipherment is rotor cipher.

It uses the idea behind monalphabetic substitution, but changes the mapping between plaintext and ciphertext characters for each plaintext character.

Figure A rotor cipher

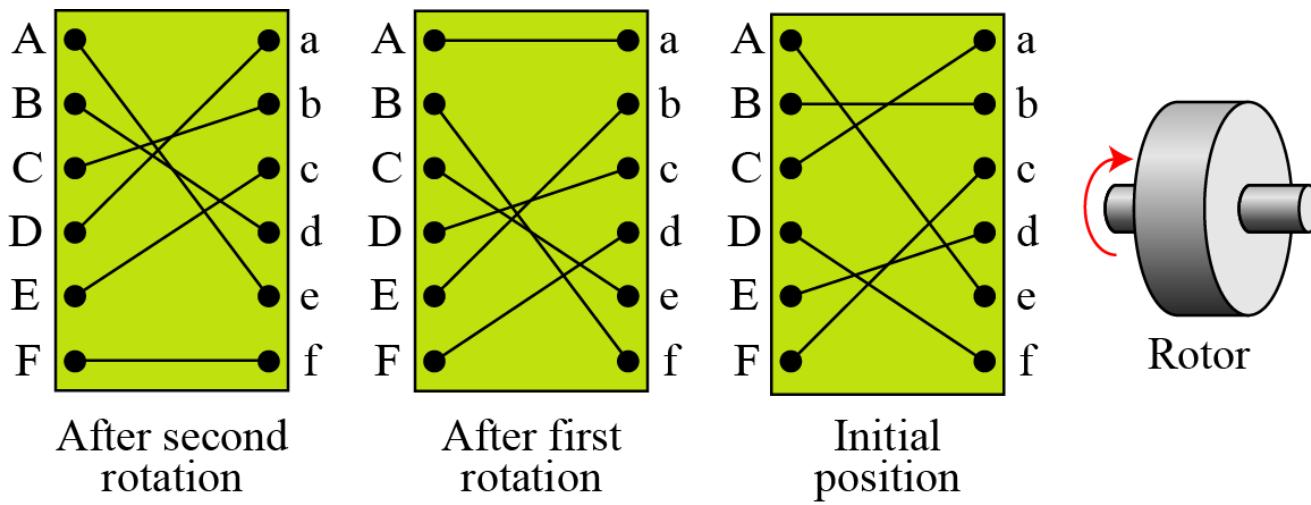


# Rotor Cipher

It uses only 6 letters, but actual rotor uses 26 letters.

Initial setting is the secret key.

'bee' encrypts as "BAA" if rotor is stationary, but become "BCA" as rotates.



# Enigma Machine

Originally designed by Sherbius, modified by German army and extensively used in WWII.

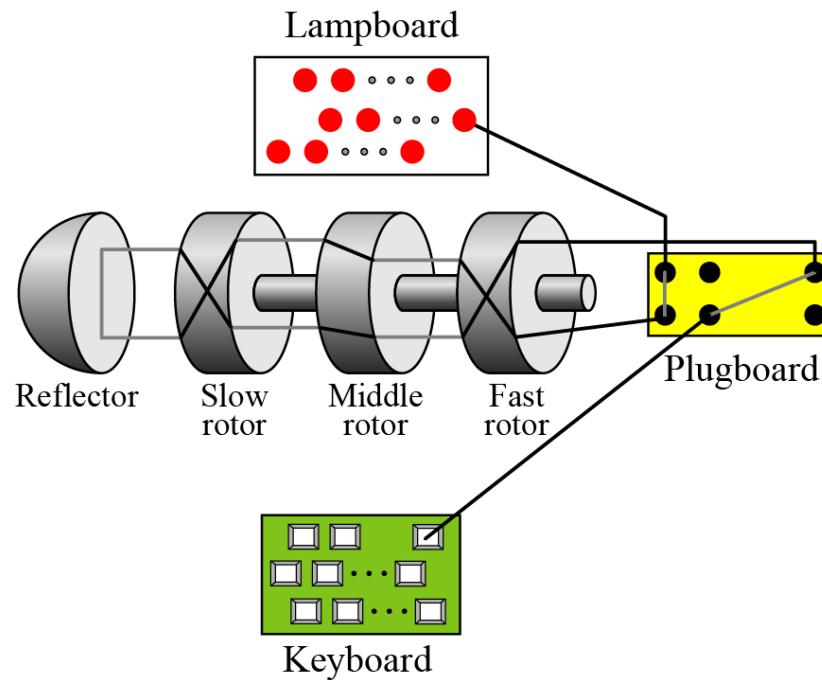


Figure A schematic of the Enigma machine

## Rotor machines

- ▶ The machine consists of a set of independently rotating cylinders through which electrical pulses can flow.
- ▶ Each cylinder has 26 input pins and 26 output pins, with internal wiring that connects each input pin to a unique output pin.
- ▶ After each input key is depressed, **the cylinder rotates one position, so that the internal connections are shifted accordingly.**
- ▶ After 26 letters of plaintext, the cylinder would be back to the initial position.



- ▶ For every complete rotation of the inner cylinder, the middle cylinder rotates one pin position. Finally, for every complete rotation of the middle cylinder, the outer cylinder rotates one pin position.
- ▶ Thus, a given setting of a 5-rotor machine is equivalent to a Vigenère cipher with a key length of 11,881,376.

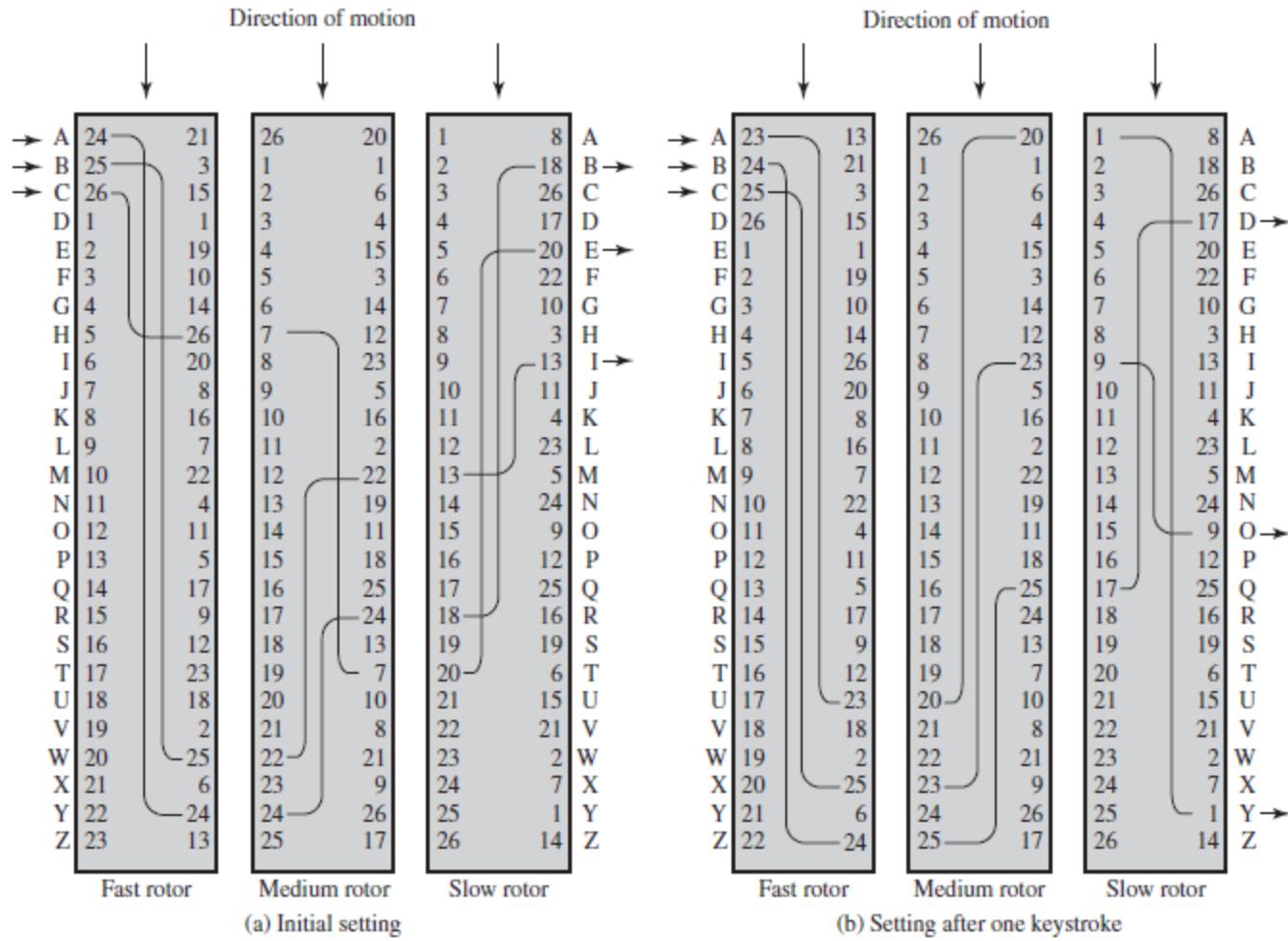


Figure 2.8 Three-Rotor Machine with Wiring Represented by Numbered Contacts

# Poly-alphabetic Ciphers Summary

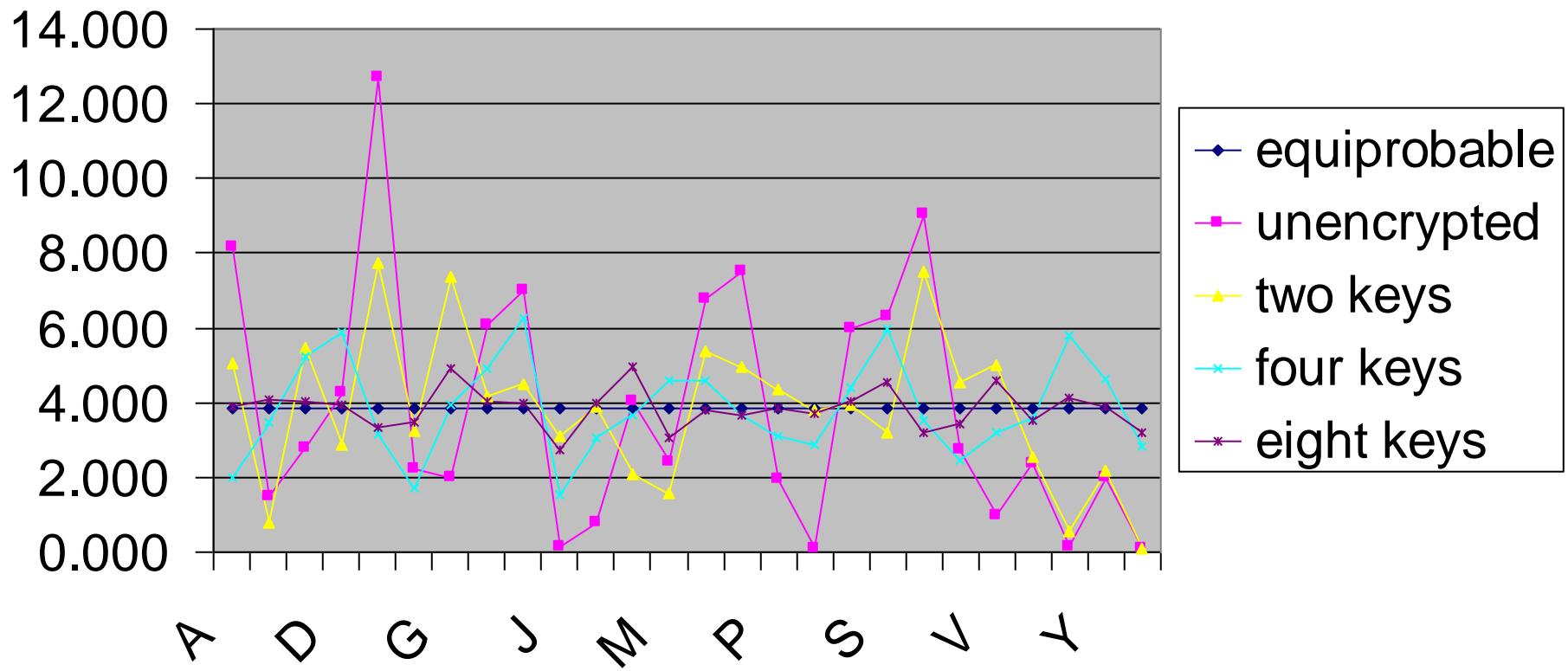
---

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



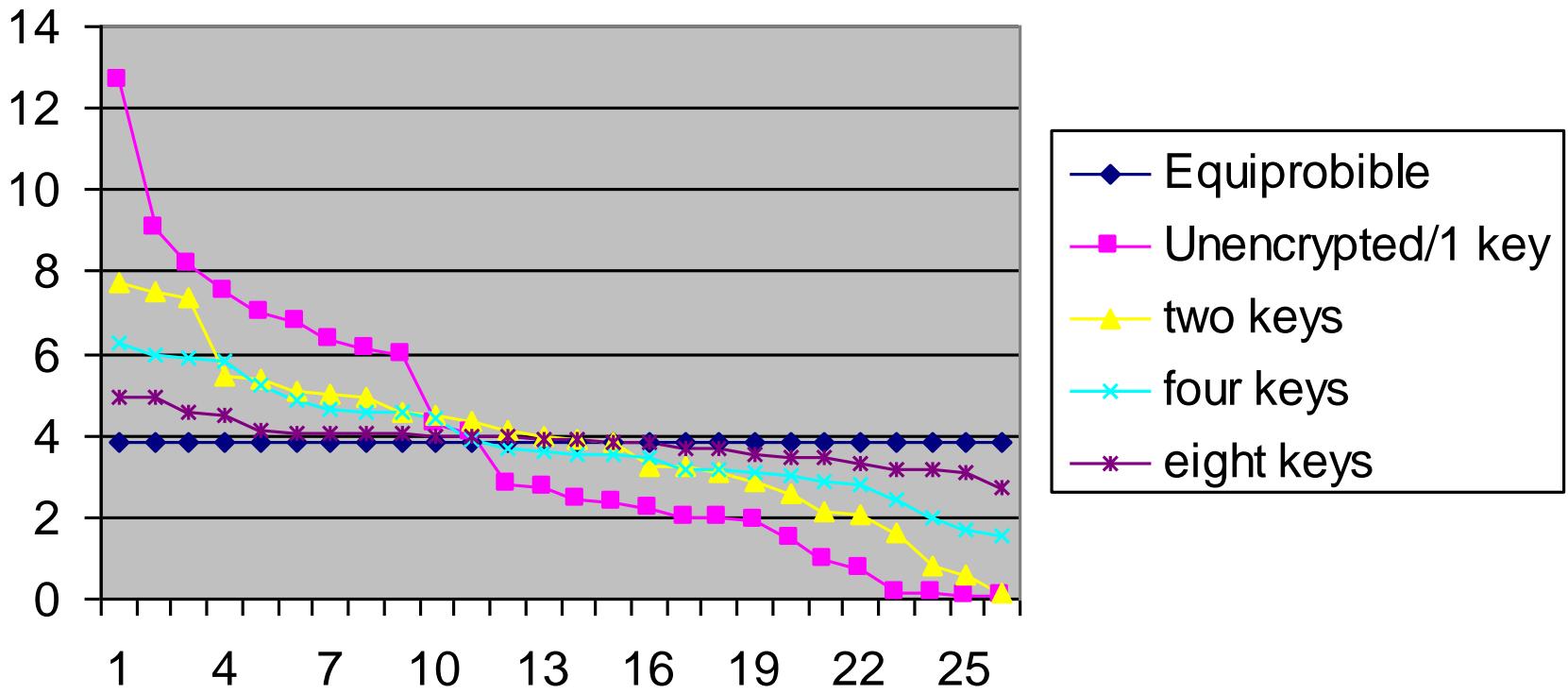
# Frequencies After Polyalphabetic Encryption

## Letter Relative Frequency



# Frequencies After Polyalphabetic Encryption

## Sorted relative frequencies



# Transposition Ciphers

A transposition cipher does not substitute one symbol for another, instead it changes the location of the symbols.

A transposition cipher reorders symbols.

## Topics:

Keyless Transposition Ciphers

Keyed Transposition Ciphers

Combining Two Approaches

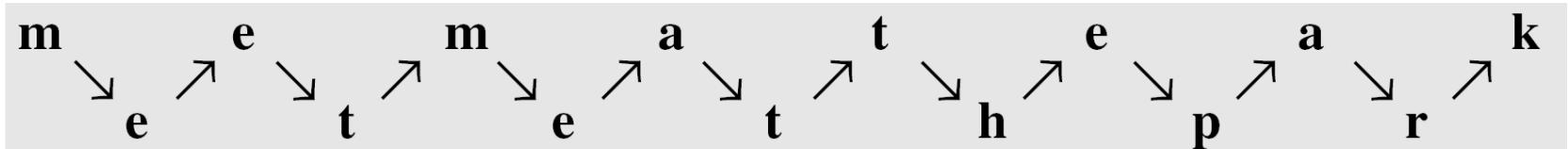


# Keyless Transposition Ciphers

Simple transposition ciphers, which were used in the past, are keyless.

## Example

A good example of a keyless cipher using the first method is the **rail fence cipher**. The ciphertext is created reading the pattern row by row. For example, to send the message “Meet me at the park” to Bob, Alice writes



She then creates the ciphertext “**MEMATEAKETETHPR**”.



## Example

Alice and Bob can agree on the number of columns and use the second method. Alice writes the same plaintext, row by row, in a table of four columns.

m	e	e	t
m	e	a	t
t	h	e	p
a	r	k	

She then creates the ciphertext “MMTAEEHREAEKTP”.



## Example

The cipher in Example is actually a transposition cipher. The following shows the permutation of each character in the plaintext into the ciphertext based on the positions.

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
01	05	09	13	02	06	10	13	03	07	11	15	04	08	12

The second character in the plaintext has moved to the fifth position in the ciphertext; the third character has moved to the ninth position; and so on. Although the characters are permuted, there is a pattern in the permutation: (01, 05, 09, 13), (02, 06, 10, 13), (03, 07, 11, 15), and (08, 12). In each section, the difference between the two adjacent numbers is 4.

# Keyed Transposition Ciphers

---

The keyless ciphers permute the characters by using writing plaintext in one way and reading it in another way. The permutation is done on the whole plaintext to create the whole ciphertext.

Another method is to divide the plaintext into groups of predetermined size, called blocks, and then use a key to permute the characters in each block separately.



## Example

Alice needs to send the message “Enemy attacks tonight” to Bob..

e n e m y      a t t a c k s      o n i g h t z

The key used for encryption and decryption is a permutation key, which shows how the character are permuted.

Encryption ↓

3	1	4	5	2
1	2	3	4	5

↑ Decryption

The permutation yields

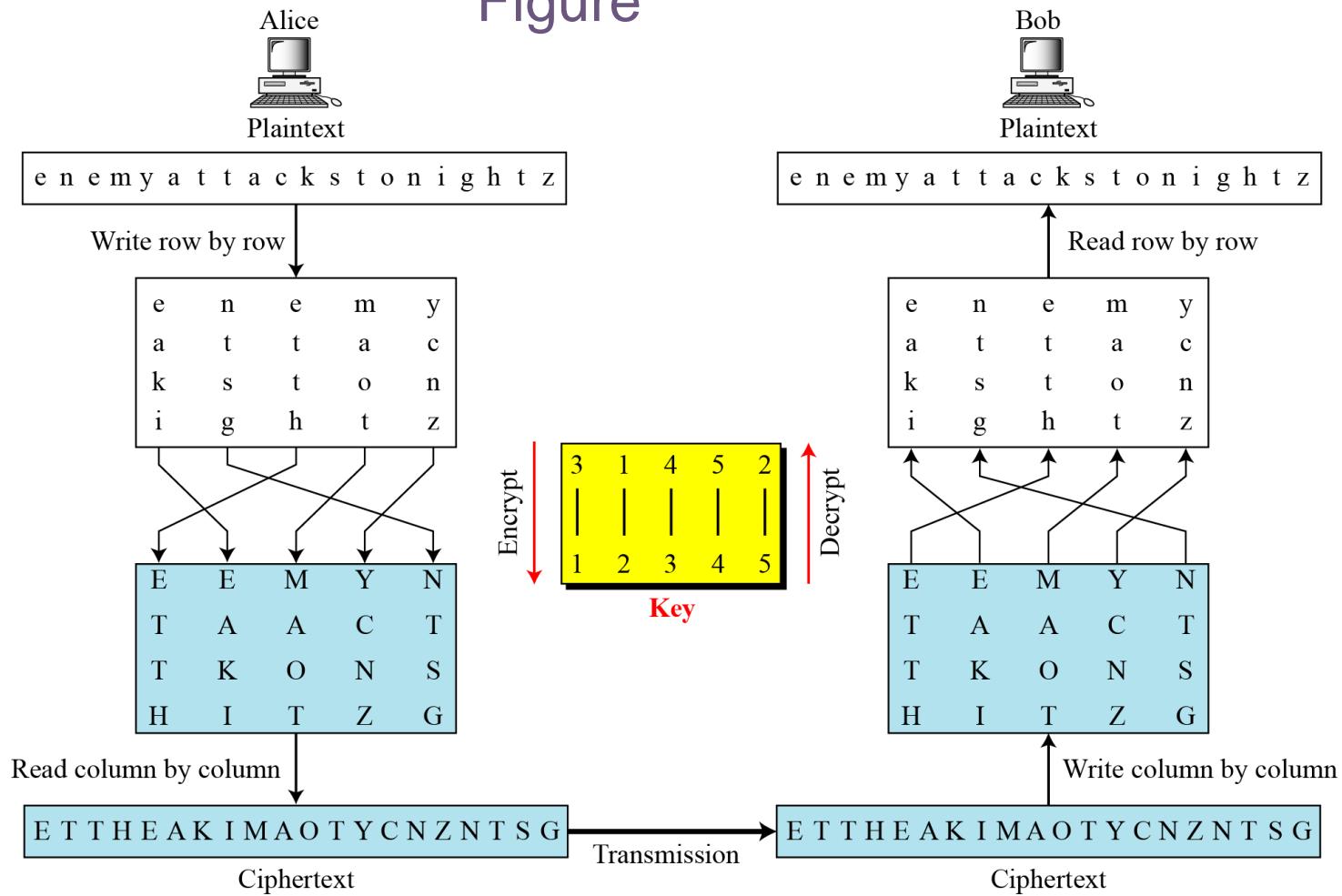
E E M Y N      T A A C T      T K O N S      H I T Z G



# Combining Two Approaches

## Example

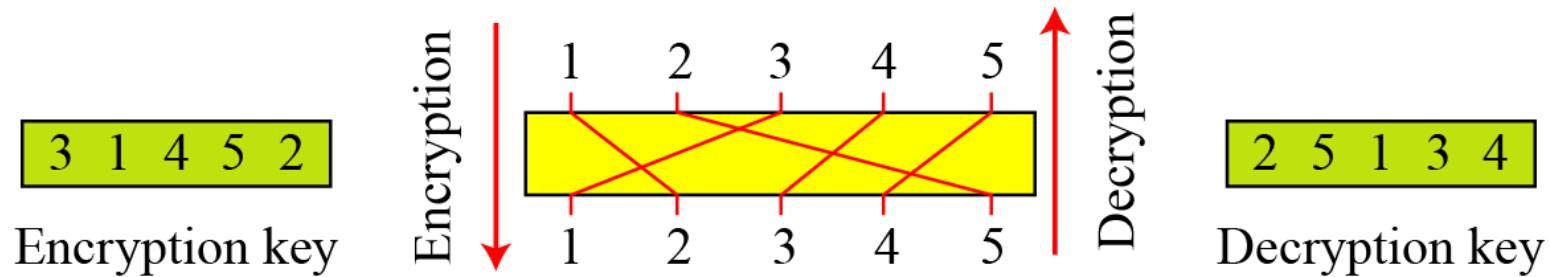
Figure



# Keys

In Example , a single key was used in two directions for the column exchange: downward for encryption, upward for decryption. It is customary to create two keys.

Figure Encryption/decryption keys in transpositional ciphers



# Continued

Figure Key inversion in a transposition cipher

Encryption key

2 6 3 1 4 7 5

2 6 3 1 4 7 5  
Add index

1 2 3 4 5 6 7

1 2 3 4 5 6 7  
Swap

2 6 3 1 4 7 5

4 1 3 5 7 2 6  
Sort

Decryption key

4 1 3 5 7 2 6

a. Manual process

Given: EncKey [index]

index  $\leftarrow 1$

while (index  $\leq$  Column)

{

DecKey[EncKey[index]]  $\leftarrow$  index

index  $\leftarrow$  index + 1

}

Return : DecKey [index]

b. Algorithm

# Continued

We can use matrices to show the encryption/decryption process for a transposition cipher.

## Example

Figure Representation of the key as a matrix in the transposition cipher

$$\begin{bmatrix} 04 & 13 & 04 & 12 & 24 \\ 00 & 19 & 19 & 00 & 02 \\ 10 & 18 & 19 & 14 & 13 \\ 08 & 06 & 07 & 19 & 25 \end{bmatrix} \times \begin{bmatrix} 3 & 1 & 4 & 5 & 2 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 04 & 04 & 12 & 24 & 13 \\ 19 & 00 & 00 & 02 & 19 \\ 19 & 10 & 14 & 13 & 18 \\ 07 & 08 & 19 & 25 & 06 \end{bmatrix}$$

Plaintext                                  Encryption key                                  Ciphertext



# Continued

Figure shows the encryption process. Multiplying the  $4 \times 5$  plaintext matrix by the  $5 \times 5$  encryption key gives the  $4 \times 5$  ciphertext matrix.

Figure Representation of the key as a matrix in the transposition cipher

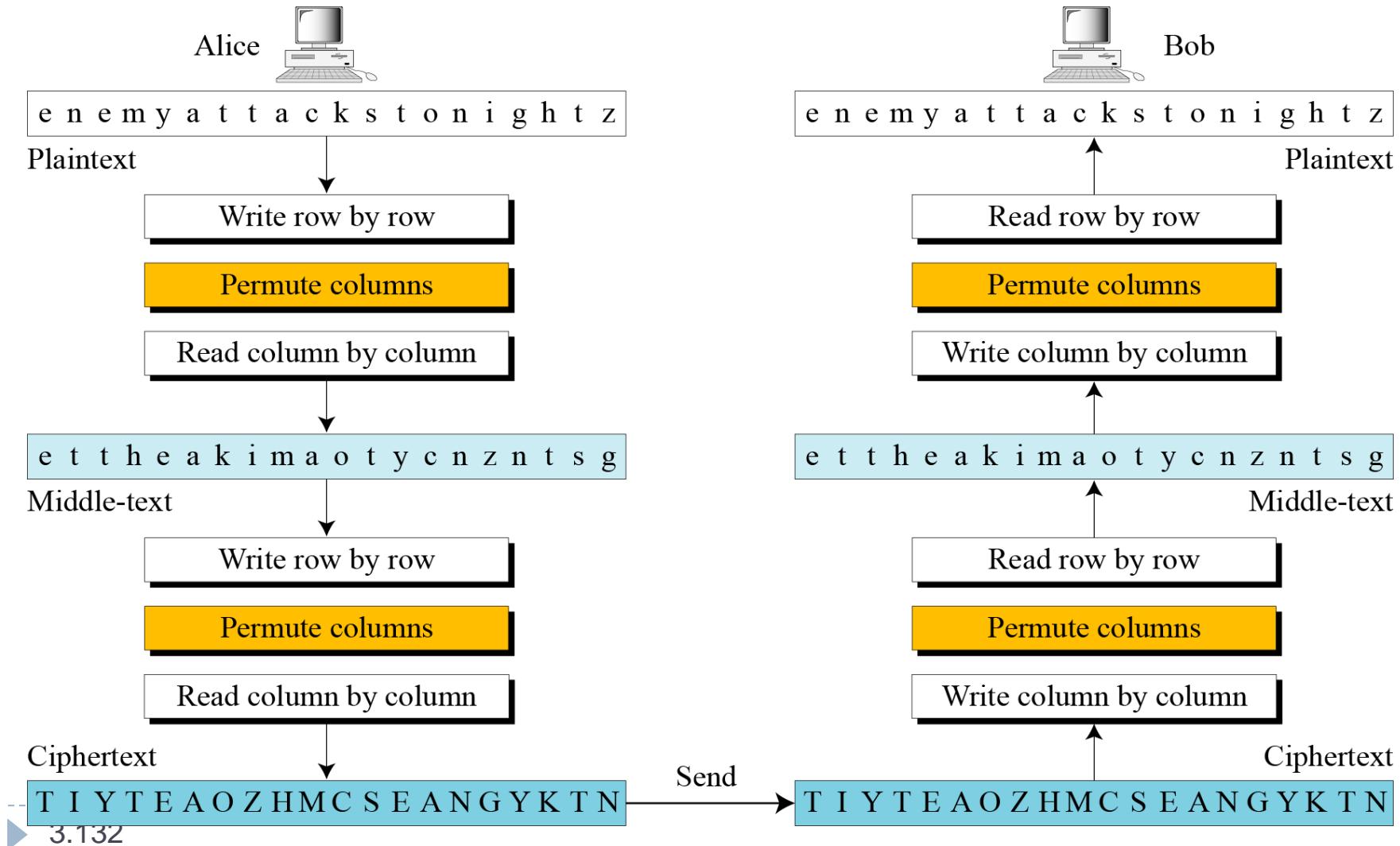
$$\begin{bmatrix} 04 & 13 & 04 & 12 & 24 \\ 00 & 19 & 19 & 00 & 02 \\ 10 & 18 & 19 & 14 & 13 \\ 08 & 06 & 07 & 19 & 25 \end{bmatrix}_{\text{Plaintext}} \times \begin{bmatrix} 3 & 1 & 4 & 5 & 2 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}_{\text{Encryption key}} = \begin{bmatrix} 04 & 04 & 12 & 24 & 13 \\ 19 & 00 & 00 & 02 & 19 \\ 19 & 10 & 14 & 13 & 18 \\ 07 & 08 & 19 & 25 & 06 \end{bmatrix}_{\text{Ciphertext}}$$



# Continued

## Double Transposition Ciphers

Figure Double transposition cipher



# Stream and Block Ciphers

---

The literature divides the symmetric ciphers into two broad categories: stream ciphers and block ciphers. Although the definitions are normally applied to modern ciphers, this categorization also applies to traditional ciphers.

## Topics :

Stream Ciphers

Block Ciphers

Combination



# Stream Ciphers

Call the plaintext stream P, the ciphertext stream C, and the key stream K.

$$P = P_1 P_2 P_3, \dots$$

$$C = C_1 C_2 C_3, \dots$$

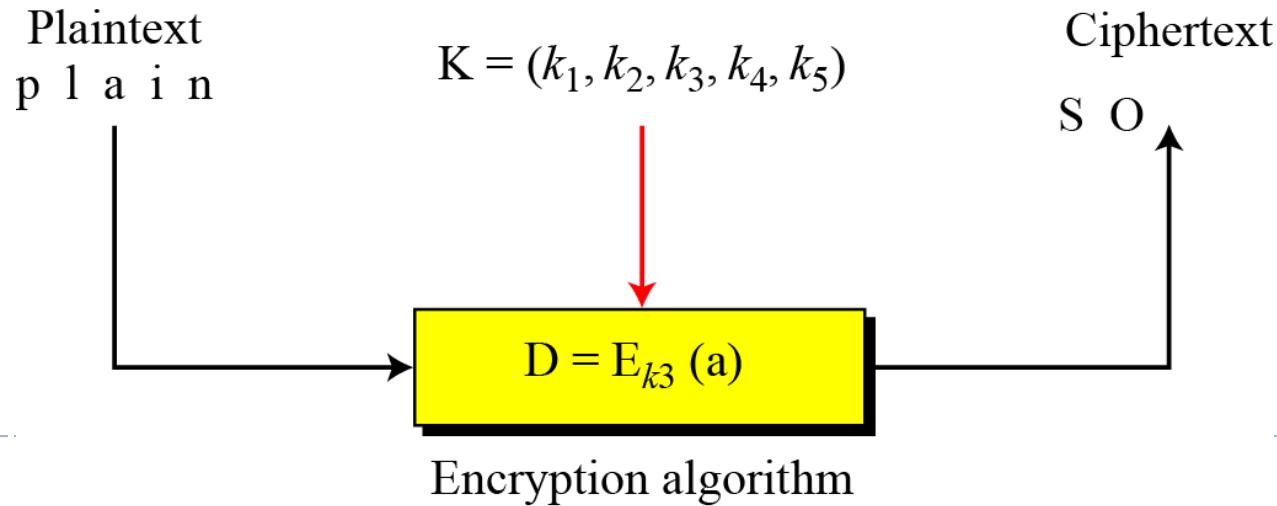
$$K = (k_1, k_2, k_3, \dots)$$

$$C_1 = E_{k1}(P_1)$$

$$C_2 = E_{k2}(P_2)$$

$$C_3 = E_{k3}(P_3) \dots$$

Figure Stream cipher



# Continued

## Example

Additive ciphers can be categorized **as stream ciphers** in which the key stream is the repeated value of the key. In other words, the key stream is considered as a predetermined stream of keys

or  $K = (k, k, \dots, k)$ . In this cipher, however, each character in the ciphertext depends only on the corresponding character in the plaintext, because the key stream is generated independently.

## Example

The monoalphabetic substitution ciphers discussed in this chapter are **also stream ciphers**. However, each value of the key stream in this case is the mapping of the current plaintext character to the corresponding ciphertext character in the mapping table.



# Continued

## Example

Vigenere ciphers are also stream ciphers according to the definition. In this case, the key stream is a repetition of  $m$  values, where  $m$  is the size of the keyword. In other words,

$$K = (k_1, k_2, \dots, k_m, k_1, k_2, \dots, k_m, \dots)$$

## Example

We can establish a criterion to divide stream ciphers based on their key streams. We can say that a stream cipher is a monoalphabetic cipher if the value of  $k_i$  does not depend on the position of the plaintext character in the plaintext stream; otherwise, the cipher is polyalphabetic.



# Continued

---

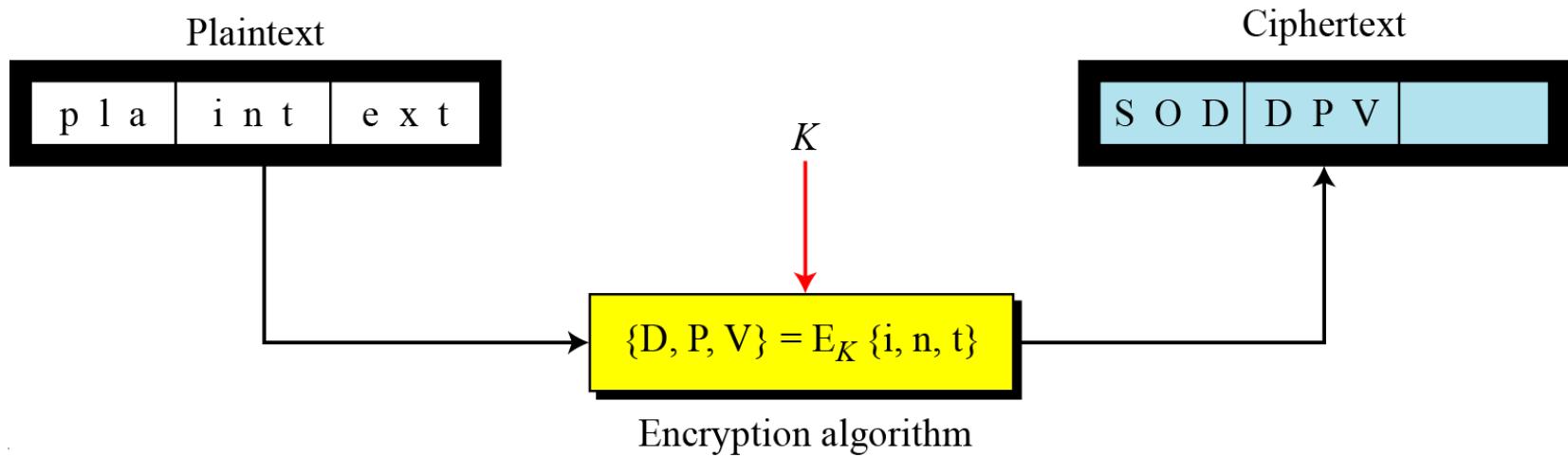
- Additive ciphers are definitely monoalphabetic because  $k_i$  in the key stream is fixed; it does not depend on the position of the character in the plaintext.
- Monoalphabetic substitution ciphers are monoalphabetic because  $k_i$  does not depend on the position of the corresponding character in the plaintext stream; it depends only on the value of the plaintext character.
- Vigenere ciphers are polyalphabetic ciphers because  $k_i$  definitely depends on the position of the plaintext character. However, the dependency is cyclic. The key is the same for two characters  $m$  positions apart.



# Stream Ciphers

In a block cipher, a group of plaintext symbols of size  $m$  ( $m > 1$ ) are encrypted together creating a group of ciphertext of the same size. A single key is used to encrypt the whole block even if the key is made of multiple values. Figure 3.27 shows the concept of a block cipher.

Figure 3.27 Block cipher



# Continued

## Example

Playfair ciphers are block ciphers. The size of the block is  $m = 2$ . Two characters are encrypted together.

## Example

Hill ciphers are block ciphers. A block of plaintext, of size 2 or more is encrypted together using a single key (a matrix). In these ciphers, the value of each character in the ciphertext depends on

all the values of the characters in the plaintext. Although the key is made of  $m \times m$  values, it is considered as a single key.

## Example

From the definition of the block cipher, it is clear that every block cipher is a polyalphabetic cipher because each character in a ciphertext block depends on all characters in the plaintext block.

# Combination

---

In practice, blocks of plaintext are encrypted individually, but they use a stream of keys to encrypt the whole message block by block. In other words, the cipher is a block cipher when looking at the individual blocks, but it is a stream cipher when looking at the whole message considering each block as a single unit.



# Product Ciphers

---

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



- ▶ The transposition cipher can be made significantly more secure by performing more than one stage of transposition. **If we apply previous mapping again:**

Key:	4	3	1	2	5	6	7
Input:	t	t	n	a	a	p	t
	m	t	s	u	o	a	o
	d	w	c	o	i	x	k
	n	l	y	p	e	t	z
Output:	NSCYAUOPTTWLTMDNAOIEPAXTTOKZ						

To visualize the result of this double transposition, designate the letters in the original plaintext message by the numbers designating their position.

01	02	03	04	05	06	07	08	09	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27	28

- After the first transposition, we have

03	10	17	24	04	11	18	25	02	09	16	23	01	08
15	22	05	12	19	26	06	13	20	27	07	14	21	28

- But after the second transposition, we have

17	09	05	27	24	16	12	07	10	02	22	20	03	25
15	13	04	23	19	14	11	01	26	21	18	08	06	28

- 
- ▶ The example just given suggests that multiple stages of encryption can produce an algorithm that is significantly more difficult to cryptanalyze
  - ▶ This is as true of substitution ciphers as it is of transposition ciphers.



# S-box

- We can extend the substitution box idea to binary words.
- Here's a  $4 \times 4$  S-box that maps 4 bits to 4 bits:

$s$	00	01	10	11
00	0011	1000	1111	0001
01	1010	0110	0101	1011
10	1110	1101	0100	0010
11	0111	0000	1001	1100

$s$	0	1	2	3
0	3	8	15	1
1	10	6	5	11
2	14	13	4	2
3	7	0	9	12

$$0000 \rightarrow 0011$$

- Examples:  $0001 \rightarrow 0100$   
 $1010 \rightarrow 0100$



## P-box

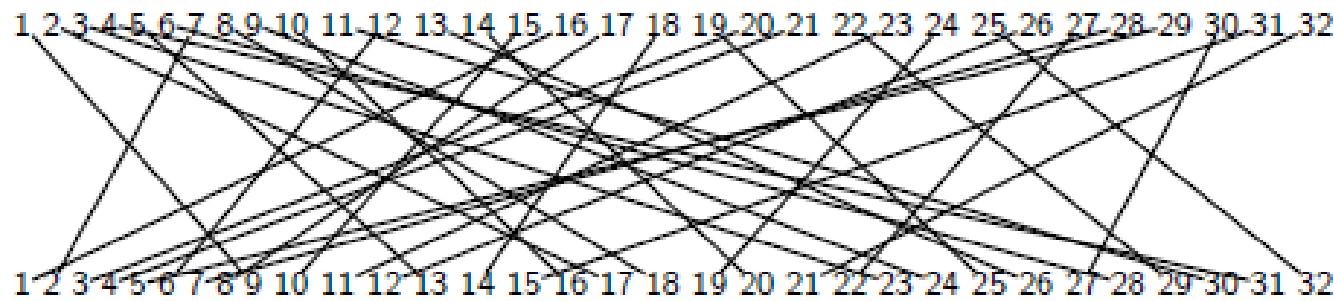
- We can extend the transposition cipher idea to binary words.
- Here's a 32-bit P-box that is used by the DES cipher:

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



# P-box

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



# Exclusive-OR

$0 \oplus 0 = 0$	$a \oplus a = 0$
$0 \oplus 1 = 1$	$a \oplus b \oplus b = a$
$1 \oplus 0 = 1$	$a \oplus a \oplus a = a$
$1 \oplus 1 = 0$	

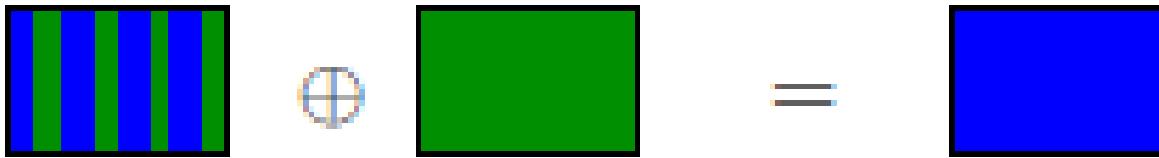
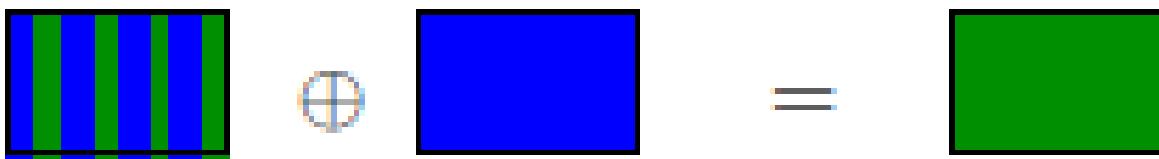
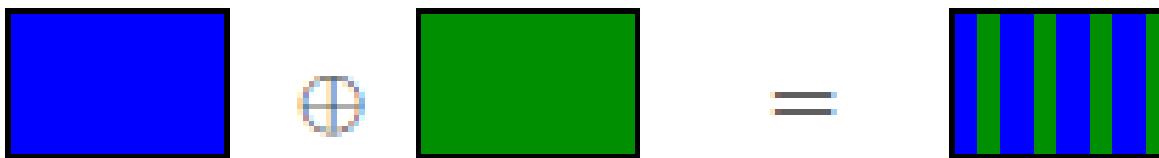
- Since xor-ing the same value twice gives us the original, we get a simple symmetric algorithm:

$$P \oplus K = C$$

$$C \oplus K = P$$



# Exclusive-OR



# Steganography & Cryptography

---

- ▶ The methods of **steganography** conceal the existence of the message, whereas the methods of **cryptography** render the message unintelligible to outsiders by various transformations of the text



# Steganography

---

- ▶ Hide a real message in a fake, but meaningful, message
- ▶ Assumes recipient knows the method of hiding
- ▶ Examples:
  - Selected letters in a document are marked to form the hidden message
  - Invisible ink (letters only become visible when exposed to a chemical or heat)
  - Using selected bits in images or videos to carry the message
- ▶ Advantages
  - Does not look like you are hiding anything
- ▶ Disadvantages
  - ▶ Once attacker knows your method, everything is lost
  - ▶ Can be inefficient (need to send lot of information to carry small message)



# Steganography Example

Dear George,

Greetings to all at Oxford. Many thanks for your letter and for the Summer examination package.

All Entry Forms and Fee Forms should be ready for final despatch to the Syndicate by Friday 20th or at the very latest, I'm told, by the 21st.

Admin has improved here, though there's room for improvement still; just give us all two or three more years and we'll really show you! Please don't let these wretched 16+ proposals destroy your basic O and A pattern. Certainly this sort of change, if implemented immediately, would bring chaos.

Sincerely yours.

---

# Review



## Probability distributions related to a cryptosystem

- Let us suppose that a cryptosystem is specified by  $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ .
- Assume that it is possible to define a probability distribution on the plaintext space  $\mathcal{P}$ , the key space  $\mathcal{K}$ .
- The probability distribution on  $\mathcal{P}$  and  $\mathcal{K}$  induces a probability distribution on  $\mathcal{C}$ .
- Let the random variables associated to plaintexts, keys and ciphertexts be  $X$ ,  $Y$  and  $K$  respectively.
- The probability that  $X = x$  is denoted by  $\Pr[X = x]$ .
- The probability that  $K = k$  is denoted by  $\Pr[K = k]$ .
- The probability that  $Y = y$  is denoted by  $\Pr[Y = y]$ .



## Distribution of ciphertexts

- $C(k) = \{e_k(x) : x \in \mathcal{P}\}$  is the set of all possible ciphertexts.
- The probability distribution of ciphertexts is

$$\Pr[Y = y] = \sum_{\{k : y \in C(k)\}} \Pr[K = k] \Pr[x = d_k(y)]$$

- The probability distribution of ciphertexts given a plaintext is

$$\Pr[Y = y | X = x] = \sum_{\{k : x = d_k(y)\}} \Pr[K = k].$$



# Distribution of plaintexts given a ciphertext

- The probability distribution of plaintexts conditional to ciphertexts is

$$\Pr[X = x | Y = y] = \frac{\Pr[(X = x) \cap (Y = y)]}{\Pr[Y = y]}$$
$$= \frac{\Pr[X = x] \Pr[Y = y | X = x]}{\Pr[Y = y]}$$

$$= \frac{\Pr[X=x] \times \sum_{\{k: x=d_k(y)\}} \Pr[K=k]}{\sum_{\{k: y \in C(k)\}} \Pr[K=k] \Pr[X=d_k(y)]}$$



## Computation of these probabilities

$$\begin{aligned} \Pr[Y=y] &= \sum_{\{k : y \in C(k)\}} \Pr[\underline{K=k} \cap \underline{y = e_k(x)}] \\ &= \sum_{\{k : y \in C(k)\}} \Pr[\underline{K=k} \cap \underline{x = d_k(y)}] \\ &= \sum_{\{k : y \in C(k)\}} \Pr[K=k] \Pr[X = \underline{x = d_k(y)}] \end{aligned}$$

## Computation of these probabilities

- Let  $\mathcal{P} = \{a, b\}$  with  $\Pr[X = a] = \frac{1}{4}$ ,  $\Pr[X = b] = \frac{3}{4}$ .
- Let  $\mathcal{K} = \{k_1, k_2, k_3\}$  with  $P[K = k_1] = \frac{1}{2}$ ,  $\Pr[K = k_2] = \Pr[K = k_3] = \frac{1}{4}$ .
- Let  $\mathcal{C} = \{1, 2, 3, 4\}$ .
- The cryptosystem is represented by the following encryption matrix:

	a	b
k1	1	2
k2	2	3
k3	3	4

$$\Pr[X = a] = \frac{1}{4}, \Pr[X = b] = \frac{3}{4}$$

$$P[K = k_1] = \frac{1}{2},$$

$$\Pr[K = k_2] = \Pr[K = k_3] = \frac{1}{4}.$$

$$\begin{aligned}\Pr[X = x | Y = y] &= \frac{\Pr[(X = x) \cap (Y = y)]}{\Pr[Y = y]} \\ &= \frac{\Pr[X = x] \Pr[Y = y | X = x]}{\Pr[Y = y]}\end{aligned}$$



## Computation of these probabilities

	a	b
k1	1	2
k2	2	3
k3	3	4

$$\Pr[X = a] = \frac{1}{4}, \Pr[X = b] = \frac{3}{4}$$

$$\Pr[K = k_1] = \frac{1}{2},$$

$$\Pr[K = k_2] = \Pr[K = k_3] = \frac{1}{4}.$$

$$\begin{aligned} & \Pr[X = x | Y = y] \\ &= \frac{\Pr[(X = x) \cap (Y = y)]}{\Pr[Y = y]} \\ &= \frac{\Pr[X = x] \Pr[Y = y | X = x]}{\Pr[Y = y]} \end{aligned}$$

$$\Pr[Y=1] = \Pr[K=k_1] \Pr[X=a]$$

$$= \frac{1}{2} \cdot \frac{1}{4} = \frac{1}{8}$$

$$\Pr[Y=2] = \underbrace{\Pr[X=b] \Pr[K=k_1]}_{\Pr[X=a] \Pr[K=k_2]} + \Pr[X=a] \Pr[K=k_2]$$

$$= \frac{3}{4} \cdot \frac{1}{2} + \frac{1}{4} \cdot \frac{1}{4} = \frac{3}{8} + \frac{1}{16}$$

## Computation of these probabilities

$$P_r[X=a|Y=2] = \frac{P_r[X=a] P_r[Y=2|X=a]}{P_r[Y=2]}$$

$$= \frac{\frac{1}{4} \cdot \frac{1}{4}}{\frac{7}{16}} = \frac{1}{7}.$$

## Perfect Secrecy

---

- Perfect secrecy means that an adversary (Oscar) cannot get any information about the plaintext by observing the ciphertext.
- A precise formulation of this was given by Claude Elwood Shannon which is as follows:

*A cryptosystem has perfect secrecy if*

$$\Pr[X = x | Y = y] = \Pr[X = x]$$

*for all  $x \in \mathcal{P}$ ,  $y \in \mathcal{C}$ .*



## Perfect secrecy and Shift Cipher

- Suppose that the 26 keys in the shift cipher are used with equal probability  $\frac{1}{26}$ . Then for any plaintext probability distribution, the Shift Cipher has perfect secrecy.
- $$\begin{aligned} \Pr[Y = y] &= \sum_{k \in \mathbb{Z}_{26}} \Pr[K = k] \Pr[X = d_k(y)] \\ &= \sum_{k \in \mathbb{Z}_{26}} \frac{1}{26} \Pr[X = y - k] = \frac{1}{26} \sum_{k \in \mathbb{Z}_{26}} \Pr[X = y - k] = \frac{1}{26}. \end{aligned}$$
- $\Pr[Y = y | X = x] = \Pr[K = (y - x) \bmod 26] = \frac{1}{26}.$
- $\Pr[X = x | Y = y] = \frac{\Pr[X=x]\Pr[Y=y|X=x]}{\Pr[Y=y]} = \frac{\Pr[X=x]\frac{1}{26}}{\frac{1}{26}} = \Pr[X = x].$

## Computation of these probabilities

---

- $\Pr[Y = 1] = \frac{1}{8}; \Pr[Y = 2] = \frac{3}{8} + \frac{1}{16} = \frac{7}{16}$   
 $\Pr[Y = 3] = \frac{3}{16} + \frac{1}{16} = \frac{1}{4}; \Pr[Y = 4] = \frac{3}{16}.$
- $\Pr[X = a|Y = 1] = 1; \Pr[X = a|Y = 2] = \frac{1}{7};$   
 $\Pr[X = a|Y = 3] = \frac{1}{4}; \Pr[X = a|Y = 4] = 0.$
- $\Pr[X = b|Y = 1] = 0; \Pr[X = b|Y = 2] = \frac{6}{7};$   
 $\Pr[X = b|Y = 3] = \frac{3}{4}; \Pr[X = b|Y = 4] = 1.$



# Summary

---

- have considered:
  - classical cipher techniques and terminology
  - monoalphabetic substitution ciphers
  - cryptanalysis using letter frequencies
  - Playfair cipher
  - polyalphabetic ciphers
  - transposition ciphers
  - product ciphers and rotor machines
  - steganography



# BLOCK CIPHERS

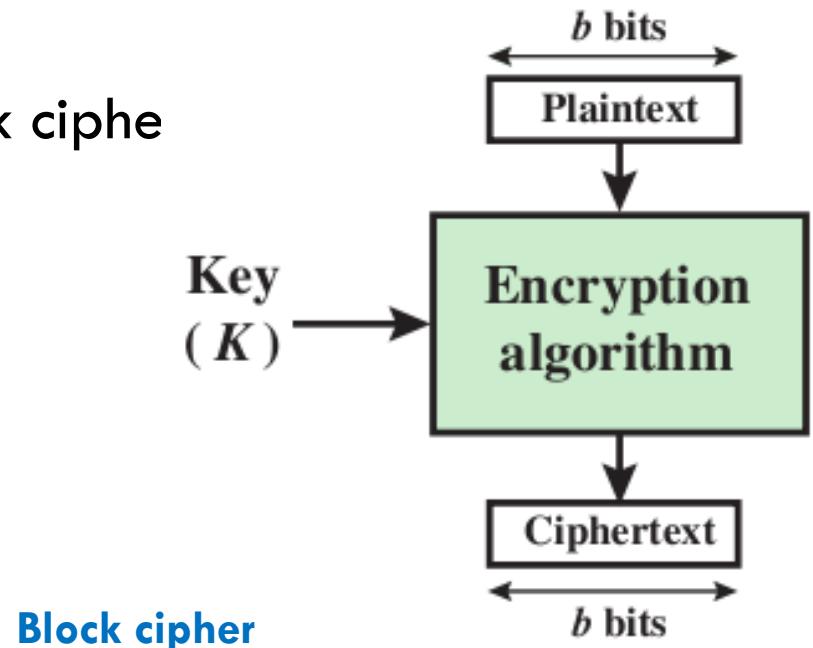


# Introduction

- Many symmetric block encryption algorithms in current use are based on a structure referred to as a Feistel block cipher
- For that reason, it is important to examine the design principles of the Feistel cipher.
- A comparison of stream ciphers and block ciphers will be made

# Block Ciphers

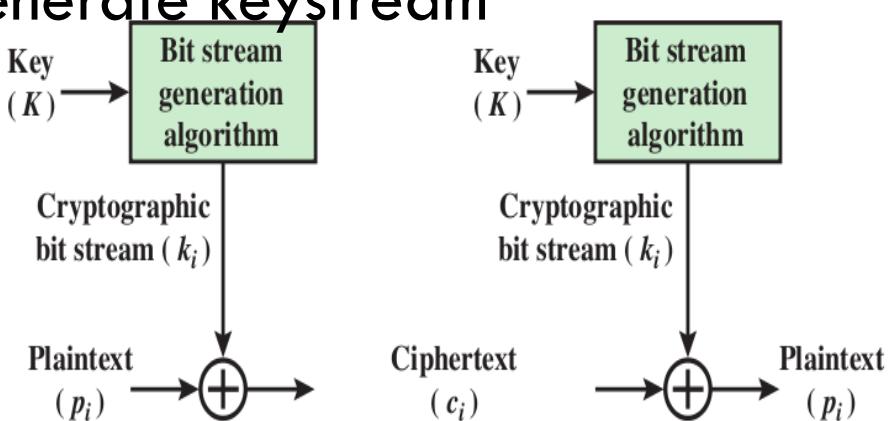
- Encrypt a block of plaintext as a whole to produce same sized cipher text
- Typical block sizes are 64 or 128 bits
- As with a stream cipher, the two users share a symmetric encryption key
- Using some modes of operation block cipher can have the same effect as a stream cipher.
- applicable to a broader range of applications than stream ciphers.



# Stream Ciphers

- Encrypts a digital data stream one bit or one byte at a time
- One time pad is example; but has practical limitations
- Typical approach for stream cipher:
  - Key ( $K$ ) used as input to bit-stream generator algorithm
  - Algorithm generates cryptographic bit stream ( $k_i$ ) used to encrypt plaintext
  - Users share a key; use it to generate keystream

Stream cipher using algorithmic  
bit-stream generator



# Motivation for the Feistel Cipher Structure : Reversible and irreversible Mappings

5

- n-bit block cipher takes n bit plaintext and produces n bit ciphertext
- In n bits,  $2^n$  possible different plaintext blocks
- Encryption to be reversible (i.e., for decryption to be possible), each must produce a unique ciphertext
- For  $n = 2$ ,

Reversible Mapping		Irreversible Mapping	
Plaintext	Ciphertext	Plaintext	Ciphertext
00	11	00	11
01	10	01	10
10	00	10	01
11	01	11	01

- If we limit ourselves to reversible mappings, the number of different transformations is  $(2^n)!$ .

# Ideal Block Cipher

- n-bit input maps to  $2^n$  possible input states
- Substitution used to produce  $2^n$  output states
- Output states map to n-bit output
- Feistel refers to this as Ideal block cipher because it allows maximum number of possible encryption mappings from plaintext block
- **Problems with ideal block cipher:**
  - Small block size: equivalent to classical substitution cipher; cryptanalysis based on statistical characteristics feasible
  - Large block size: key must be very large; performance/implementation problems

# Ideal block cipher example

P	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11	K12
00	00	00	00	00	00	01	01	10	10	11	11	
01	01	01	10	10	11	11	00	00	00	00	00	
10	10	11	01	11	01	10	10	11	01	11	01	10
11	11	10	11	01	10	01	11	10	11	01	10	01

P	K13	K14	K15	K16	K17	K18	K19	K20	K21	K22	K23	K24
00	01	01	10	10	11	11	01	01	10	10	11	11
01	10	11	01	11	01	10	10	11	01	11	01	10
10	00	00	00	00	00	00	11	10	11	01	10	01
11	11	10	11	01	10	01	00	00	00	00	00	00

2 bit block,  $2^2=4$  mappings

Input 01

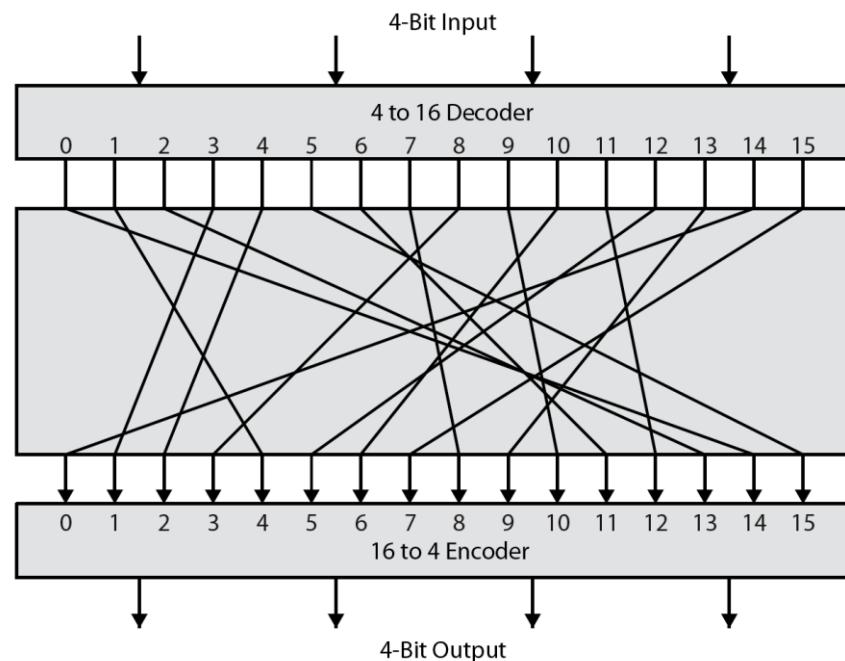
Output 01 if K17 is used, as  
K17=11 01 00 10

- Ideal : n-bit block,  $2^n$  Mappings.
- Total  $2^n !$  mappings
- Any Key length (to represent any mapping) n.  $2^n$  bits (each mapping contains n bits)
- Fiestel: n-bit block,  $2^K$  mappings, key length K

# Substitution/Block cipher

- 4-bit input produces one of 16 input states
- What is the possible number of different transformations?
- which is mapped by the substitution cipher into a unique one of 16 possible output states, each of which is represented by 4 ciphertext bits.
- This is the most general form of block cipher and can be used to define any reversible mapping between plaintext and ciphertext.

Figure illustrates the logic of a general substitution cipher for  $n = 4$ .



# Encryption and Decryption Tables for Substitution Cipher

Plaintext	Ciphertext
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

Ciphertext	Plaintext
0000	1110
0001	0011
0010	0100
0011	1000
0100	0001
0101	1100
0110	1010
0111	1111
1000	0111
1001	1101
1010	1001
1011	0110
1100	1011
1101	0010
1110	0000
1111	0101

# Substitution-permutation (S-P) networks

## Claude Shannon and Substitution-Permutation Ciphers

- Claude Shannon introduced idea of substitution-permutation (S-P) networks in 1949 paper
- This idea is the basis of modern block ciphers
- S-P nets are based on the two primitive cryptographic operations seen before:
  - *substitution* (S-box)
  - *permutation* (P-box)
- Provide *confusion & diffusion* of message & key

# Diffusion and Confusion

## Diffusion

- ▶ Dissipates **statistical structure** of plaintext over bulk of ciphertext
- ▶ E.g. A plaintext letter affects the value of many ciphertext letters
- ▶ How: repeatedly apply permutation (transposition) to data, and then apply function

## Confusion

- ▶ Makes **relationship** between **ciphertext** and **key** as complex as possible
- ▶ Even if attacker can find some statistical characteristics of ciphertext, still hard to find key
- ▶ How: apply complex (non-linear) substitution algorithm

# Diffusion

- How to achieve this?
  - Develop a many-to-many mapping between plain-ciphertext
  - Having each plaintext digit affect the value of many ciphertext digits; generally
  - this is equivalent to having each ciphertext digit be affected by many plaintext digits.
  - An example: encrypt a message of characters with an averaging operation:
  - adding k successive letters to get a ciphertext letter  $y_n$ .
  - One can show that the statistical structure of the plaintext has been dissipated

$$M = m_1, m_2, m_3, \dots$$

$$y_n = \left( \sum_{i=1}^k m_{n+i} \right) \bmod 26$$

# Confusion



- How to achieve this?
- Achieved by the **use of a complex substitution algorithm.**
- In contrast, a simple linear substitution function would add little confusion.

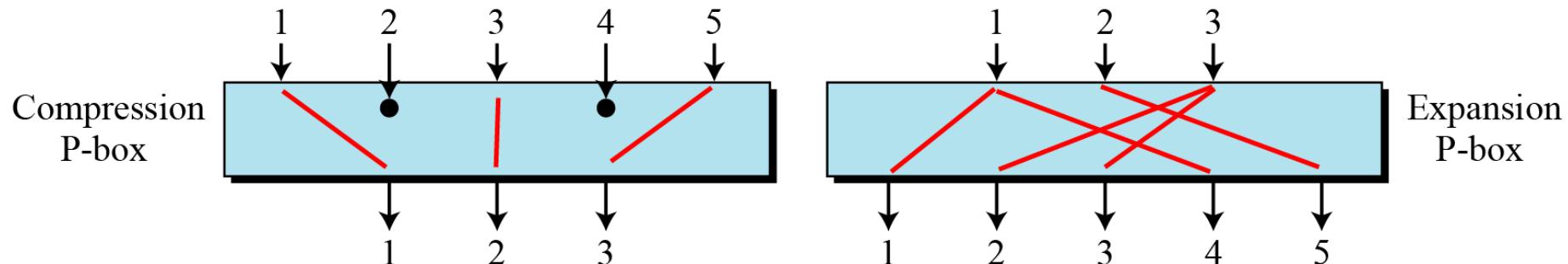
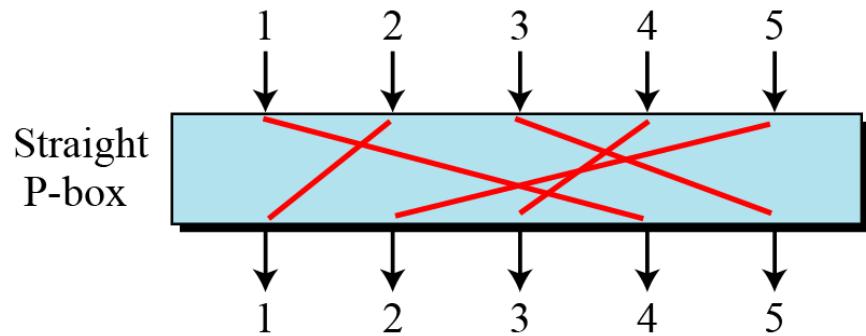
# Components of a Modern Block Cipher

Modern block ciphers normally are keyed substitution ciphers in which the key allows only partial mappings from the possible inputs to the possible outputs.

## P-Boxes

A P-box (permutation box) parallels the traditional transposition cipher for characters. It transposes bits.

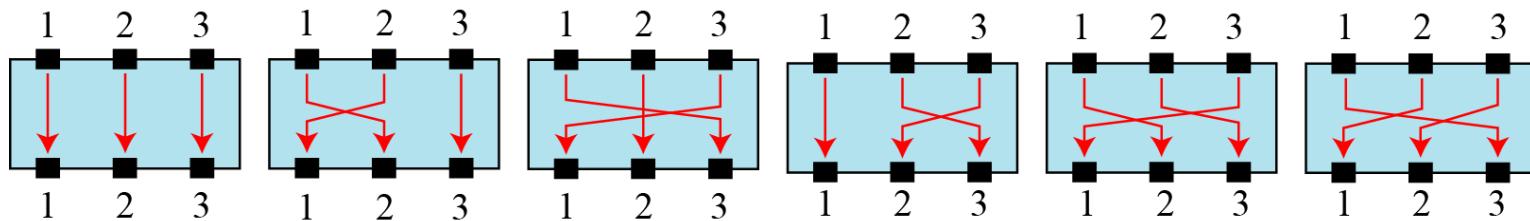
## Three types of P-boxes



## Example

Figure shows all 6 possible mappings of a  $3 \times 3$  P-box.

*The possible mappings of a  $3 \times 3$  P-box*



*Continued*

## Straight P-Boxes

**Example of a permutation table for a straight P-box**

58	50	42	34	26	18	10	02	60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06	64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01	59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05	63	55	47	39	31	23	15	07

## Example

Design an  $8 \times 8$  permutation table for a straight P-box that moves the two middle bits (bits 4 and 5) in the input word to the two ends (bits 1 and 8) in the output words. Relative positions of other bits should not be changed.

## Solution

We need a straight P-box with the table [4 1 2 3 6 7 8 5]. The relative positions of input bits 1, 2, 3, 6, 7, and 8 have not been changed, but the first output takes the fourth input and the eighth output takes the fifth input.

*Continued*

## Compression P-Boxes

A compression P-box is a P-box with  $n$  inputs and  $m$  outputs where  $m < n$ .

**Table** *Example of a  $32 \times 24$  permutation table*

01	02	03	21	22	26	27	28	29	13	14	17
18	19	20	04	05	06	10	11	12	30	31	32

## Expansion P-Box

*Continued*

An expansion P-box is a P-box with n inputs and m outputs where  $m > n$ .

**Table** *Example of a  $12 \times 16$  permutation table*

01	09	10	11	12	01	02	03	03	04	05	06	07	08	09	12
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

## P-Boxes: Invertibility

*Continued*

A straight P-box is invertible, but compression and expansion P-boxes are not.

## Example

Figure shows how to invert a permutation table represented as a one-dimensional table.

### Figure Inverting a permutation table

1. Original table

6	3	4	5	2	1
---	---	---	---	---	---

2. Add indices

6	3	4	5	2	1
1	2	3	4	5	6

3. Swap contents  
and indices

1	2	3	4	5	6
6	3	4	5	2	1

4. Sort based  
on indices

6	5	2	3	4	1
1	2	3	4	5	6

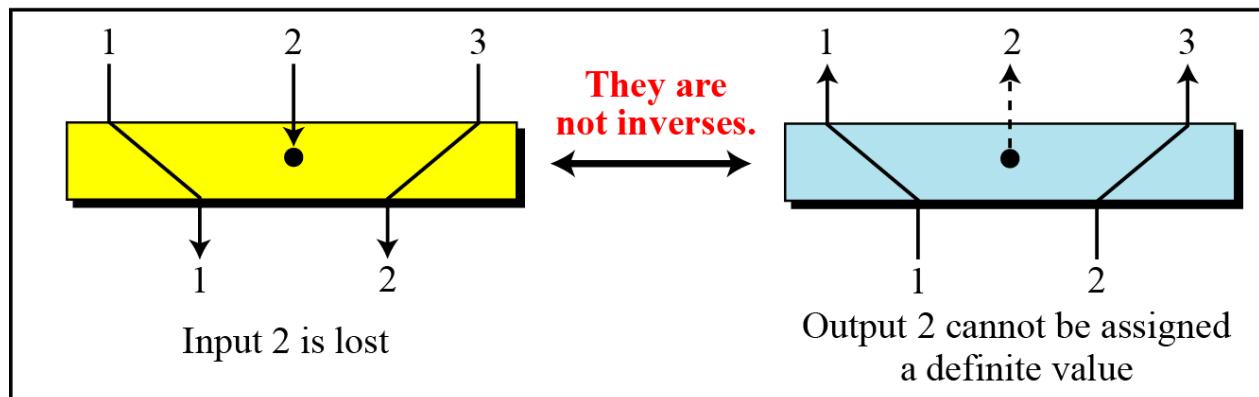
6	5	2	3	4	1
---	---	---	---	---	---

5. Inverted table

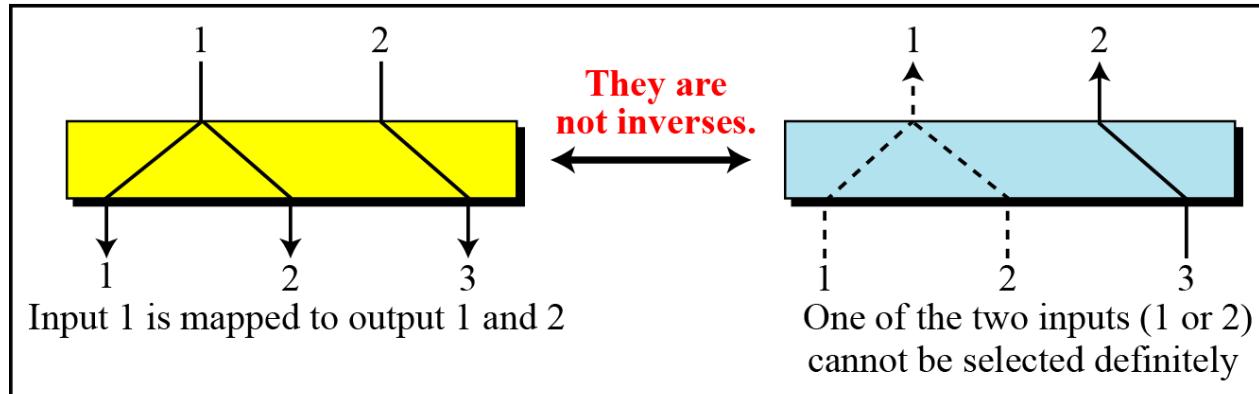
# *Continued*

**Figure Compression and expansion P-boxes are non-invertible**

Compression P-box



Expansion P-box



# *Continued*

## *S-Box*

*An S-box (substitution box) can be thought of as a miniature substitution cipher.*

An S-box is an  $m \times n$  substitution unit, where  $m$  and  $n$  are not necessarily the same.

## Example

In an S-box with three inputs and two outputs, we have

$$y_1 = x_1 \oplus x_2 \oplus x_3 \quad y_2 = x_1$$

The S-box is linear because  $a_{1,1} = a_{1,2} = a_{1,3} = a_{2,1} = 1$  and  $a_{2,2} = a_{2,3} = 0$ . The relationship can be represented by matrices, as shown below:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

## Example

In an S-box with three inputs and two outputs, we have

$$y_1 = (x_1)^3 + x_2 \quad y_2 = (x_1)^2 + x_1 x_2 + x_3$$

where multiplication and addition is in GF(2). The S-box is nonlinear because there is no linear relationship between the inputs and the outputs.

## Example

The following table defines the input/output relationship for an S-box of size  $3 \times 2$ . The leftmost bit of the input defines the row; the two rightmost bits of the input define the column. The two output bits are values on the cross section of the selected row and column.

Leftmost bit

Rightmost bits

Output bits

		00	01	10	11	
		0	00	10	01	11
		1	10	00	11	01
Output bits						

Based on the table, an input of 010 yields the output 01. An input of 101 yields the output of 00.

# *Continued*

## Example

Figure shows an example of an invertible S-box. For example, if the input to the left box is 001, the output is 101. The input 101 in the right table creates the output 001, which shows that the two tables are inverses of each other.

**Figure S-box tables for Example**

3 bits

Table used for encryption

3 bits

		00	01	10	11
0	011	101	111	100	
	000	010	001	110	
1	001	110	011	000	101

3 bits

Table used for decryption

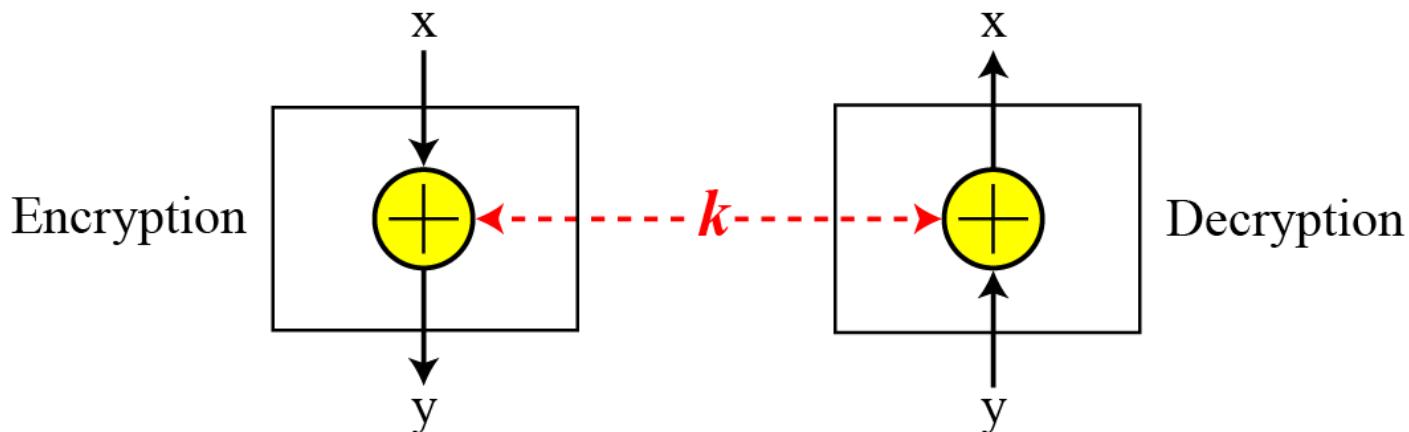
3 bits

		00	01	10	11
0	100	110	101	000	
	011	001	111	010	
1	001	110	011	000	101

## **Exclusive-Or**

*An important component in most block ciphers is the exclusive-or operation.*

**Figure** *Invertibility of the exclusive-or operation*



## *Continued*

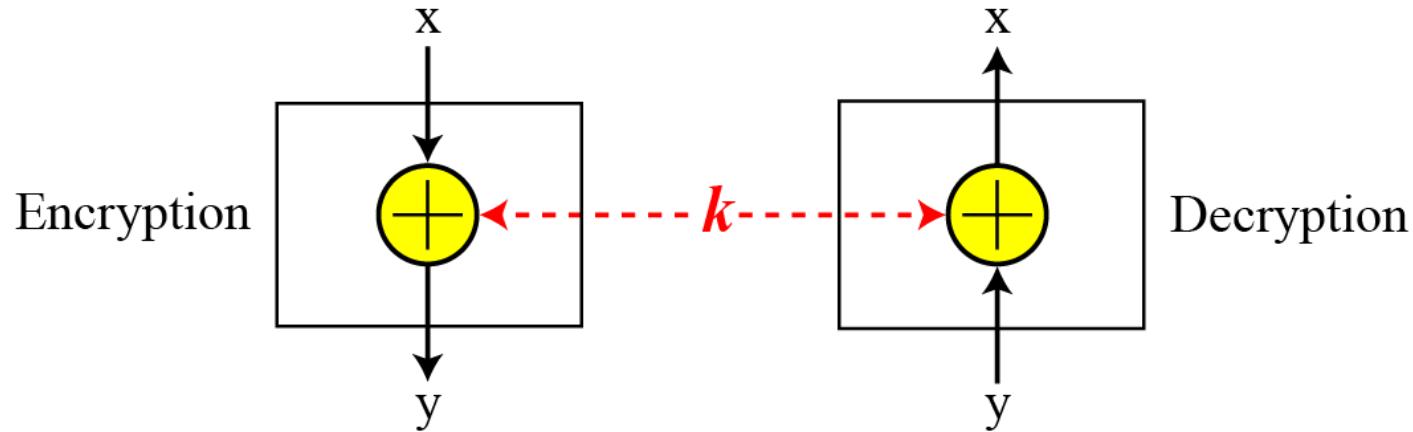
### **Exclusive-Or (Continued)**

*An important component in most block ciphers is the exclusive-or operation. Addition and subtraction operations in the  $GF(2^n)$  field are performed by a single operation called the exclusive-or (XOR).*

*The five properties of the exclusive-or operation in the  $GF(2^n)$  field makes this operation a very interesting component for use in a block cipher: **closure, associativity, commutativity, existence of identity, and existence of inverse.***

*Continued*

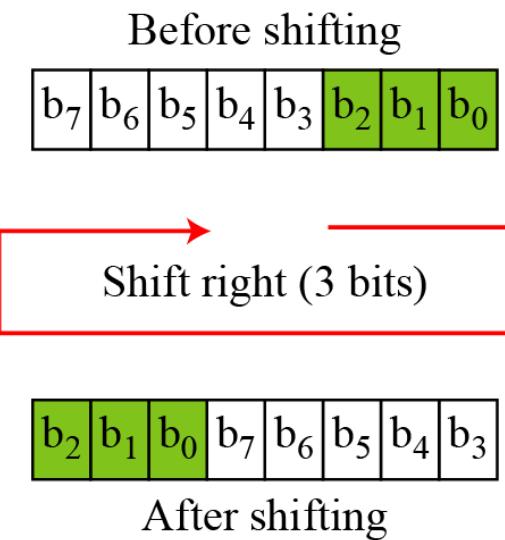
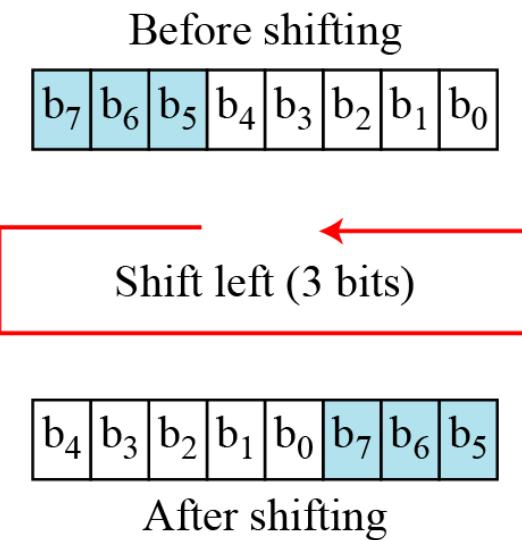
Figure Invertibility of the exclusive-or operation



## Circular Shift

*Another component found in some modern block ciphers is the circular shift operation.*

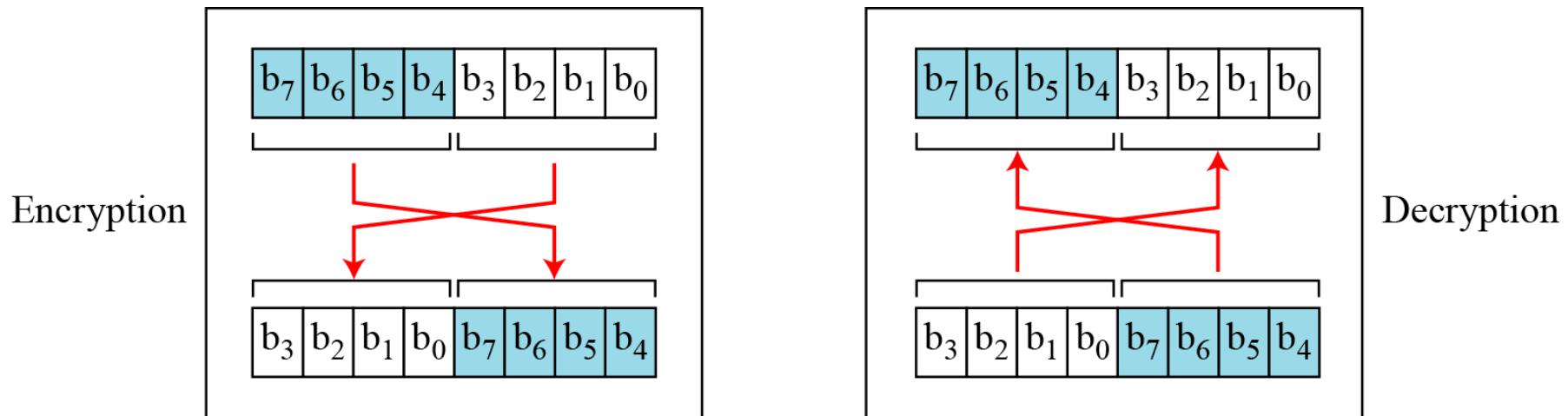
**Figure** *Circular shifting an 8-bit word to the left or right*



## Swap

The swap operation is a special case of the circular shift operation where  $k = n/2$ .

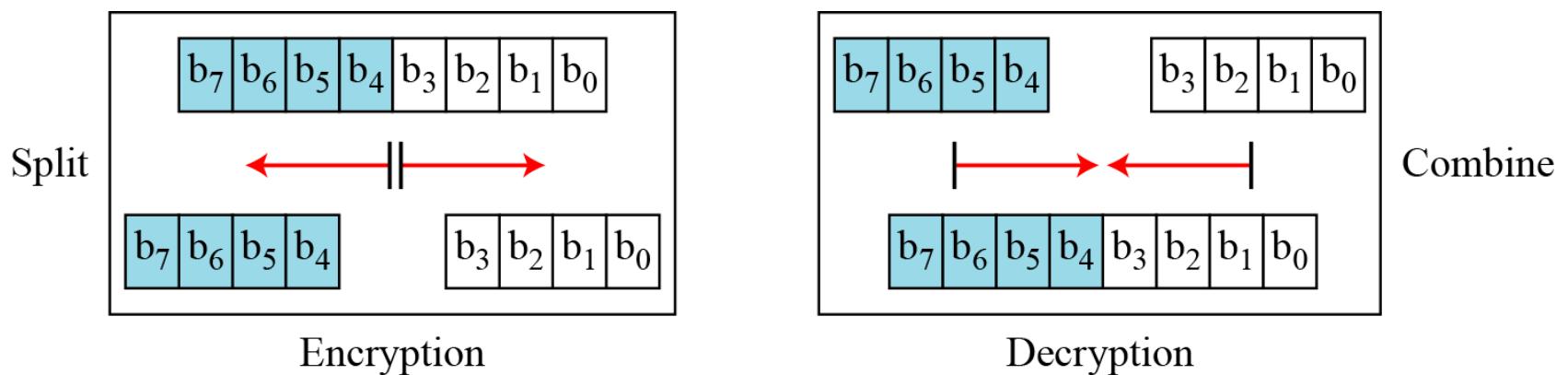
**Figure Swap operation on an 8-bit word**



## Split and Combine

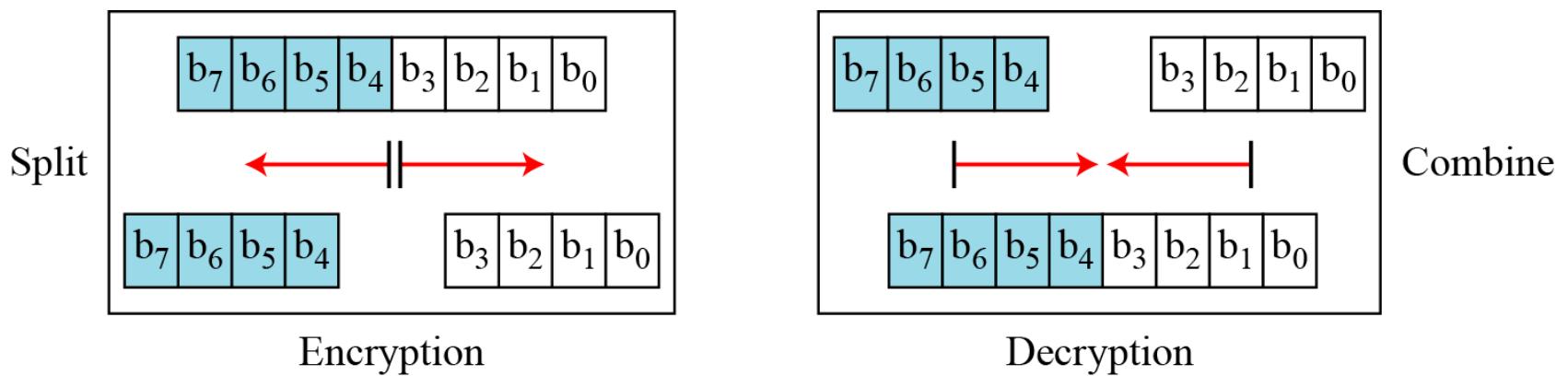
Two other operations found in some block ciphers are *split* and *combine*.

**Figure 5.12** *Split and combine operations on an 8-bit word*



# *Continued*

**Figure** *Split and combine operations on an 8-bit word*



# *Product Ciphers*

*Shannon introduced the concept of a product cipher. A product cipher is a complex cipher combining substitution, permutation, and other components.*

# *Continued*

## *Diffusion*

*The idea of diffusion is to hide the relationship between the ciphertext and the plaintext.*

Diffusion hides the relationship between the ciphertext and the plaintext.

# *Continued*

## **Confusion**

*The idea of confusion is to hide the relationship between the ciphertext and the key.*

---

**Confusion hides the relationship between the ciphertext  
and the key.**

---

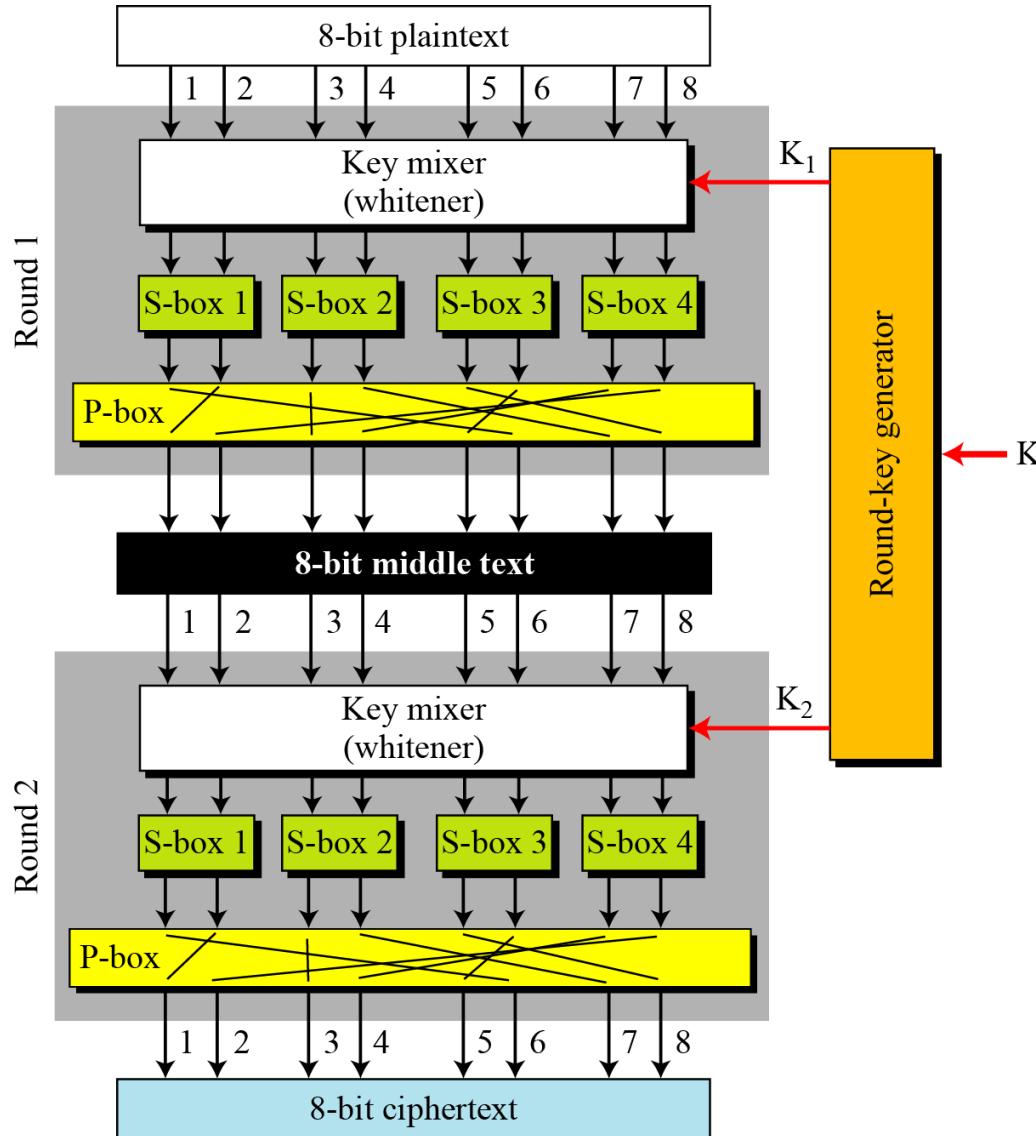
# *Continued*

## **Rounds**

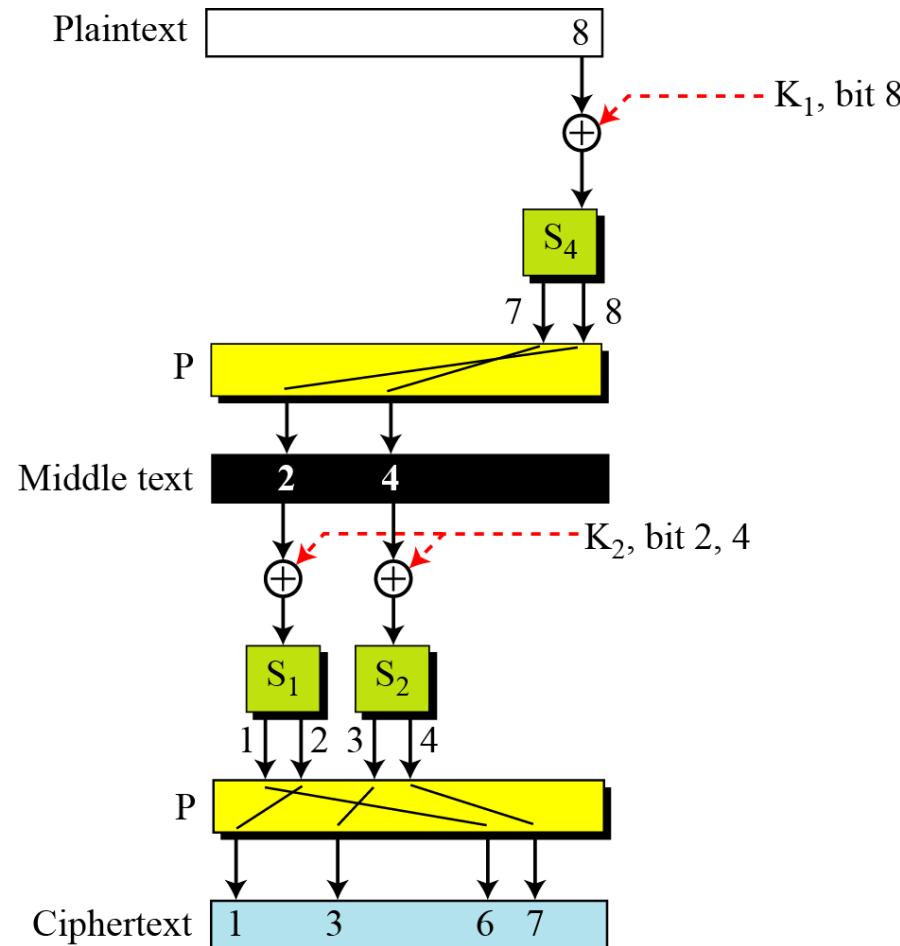
*Diffusion and confusion can be achieved using iterated product ciphers where each iteration is a combination of S-boxes, P-boxes, and other components.*

*Continued*

**Figure A product cipher made of two rounds**



**Figure Diffusion and confusion in a block cipher**



# Two Classes of Product Ciphers

Modern block ciphers are all product ciphers, but they are divided into two classes.

1. Feistel ciphers
2. Non-Feistel ciphers

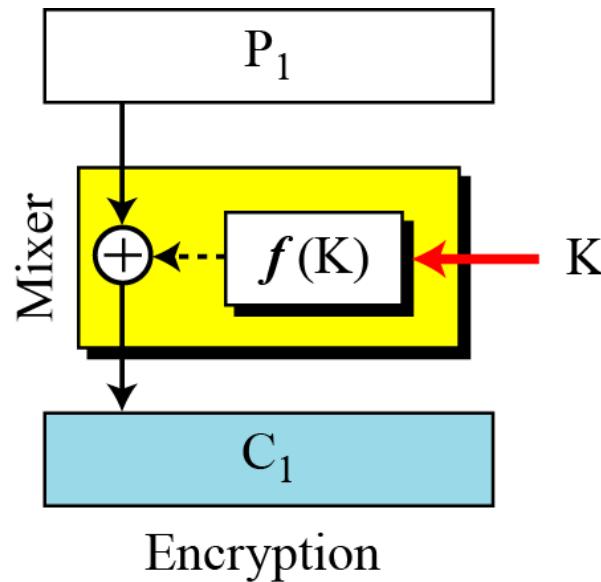
# Two Classes of Product Ciphers (cont.)

## Feistel Ciphers

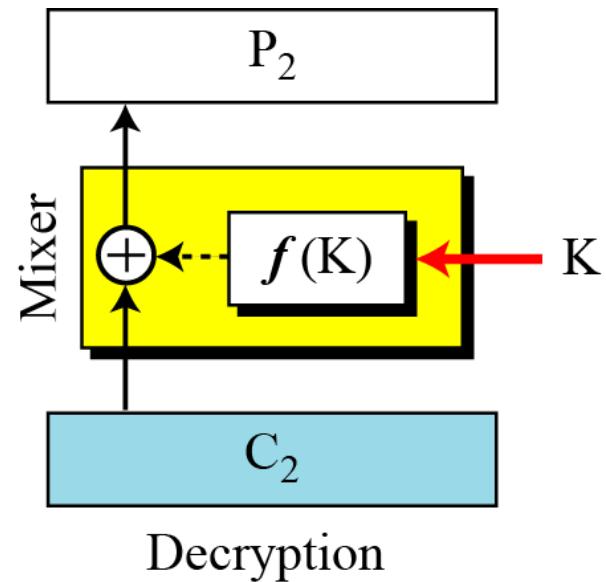
Feistel designed a very intelligent and interesting cipher that has been used for decades. A Feistel cipher can have three types of components: self-invertible, invertible, and noninvertible.

## The first thought in Feistel cipher design

Non-invertible elements cancels out when X-ored



Encryption



Decryption

Diffusion hides the relationship between the ciphertext and the plaintext.

Two algorithms are inverses of each other: If  
 $C_2 = C_1$  then  $P_2 = P_1$

Encryption:  $C_1 = P_1 \oplus f(K)$

Decryption:  $P_2 = C_2 \oplus f(K) = C_1 \oplus f(K) = P_1 \oplus f(K) \oplus f(K) = P_1 \oplus (00\dots0) = P_1$

---

The mixer in the Feistel design is self-invertible.

---

# *Continued*

## Example

This is a trivial example. The plaintext and ciphertext are each 4 bits long and the key is 3 bits long. Assume that the function takes the first and third bits of the key, interprets these two bits as a decimal number, squares the number, and interprets the result as a 4-bit binary pattern. Show the results of encryption and decryption if the original plaintext is 0111 and the key is 101.

### Solution

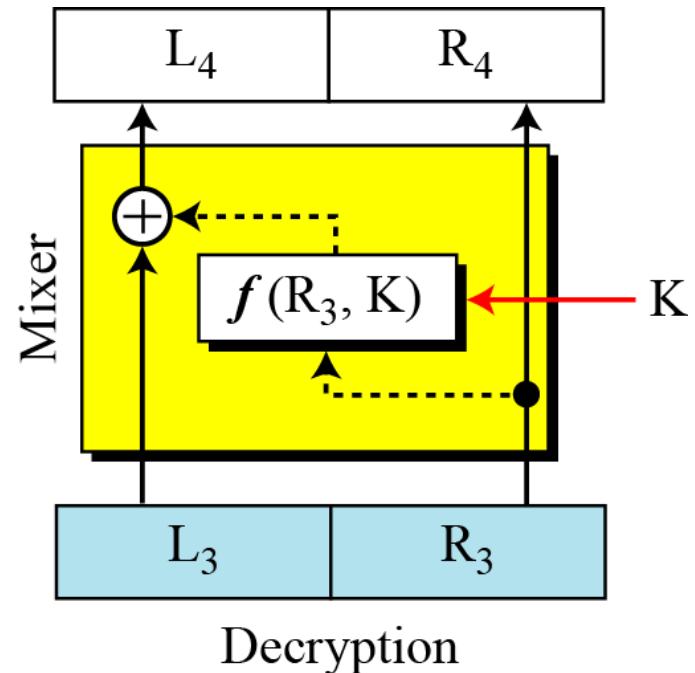
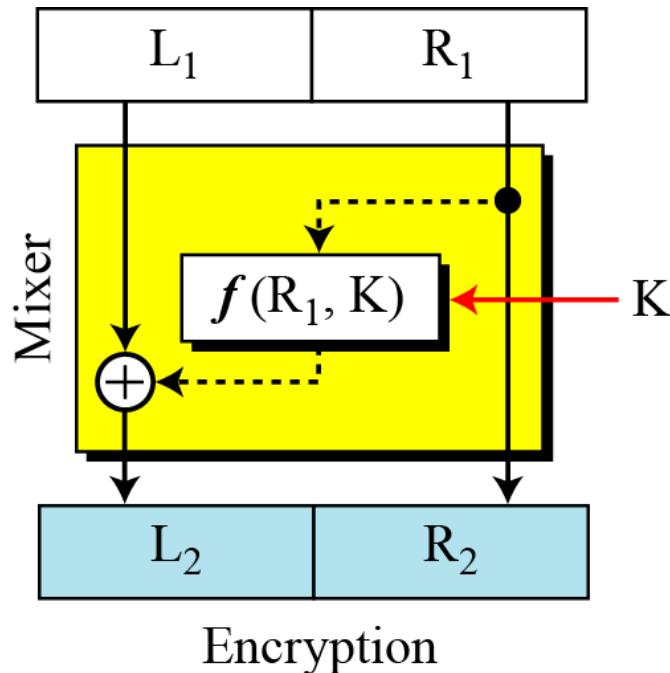
The function extracts the first and third bits to get 11 in binary or 3 in decimal. The result of squaring is 9, which is 1001 in binary.

$$\text{Encryption: } C = P \oplus f(K) = 0111 \oplus 1001 = 1110$$

$$\text{Decryption: } P = C \oplus f(K) = 1110 \oplus 1001 = 0111$$

*Continued*

## The improvement in Feistel cipher design



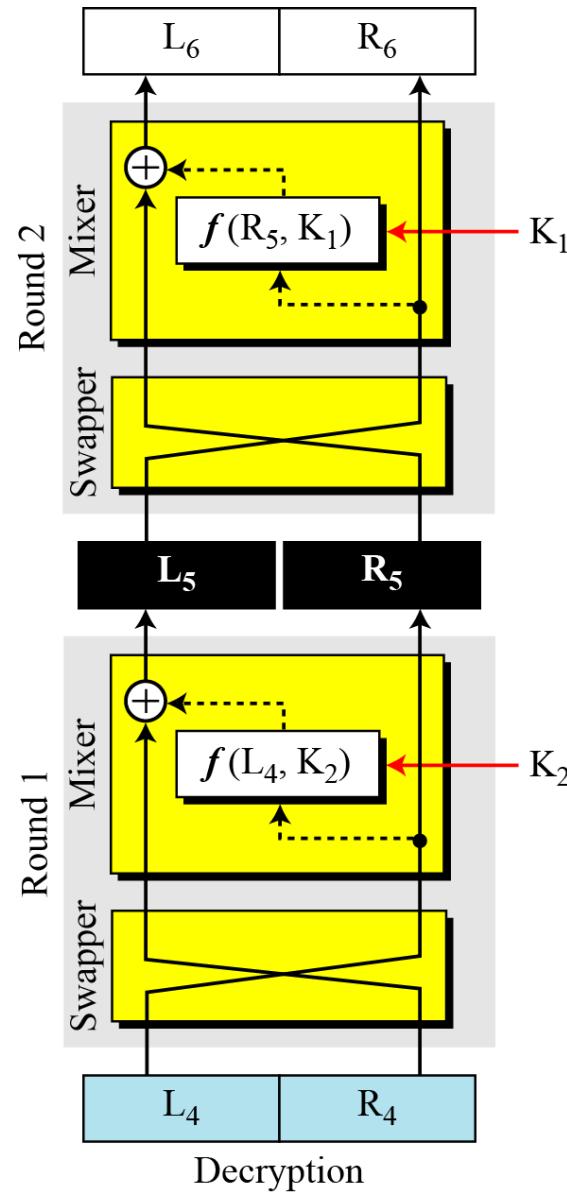
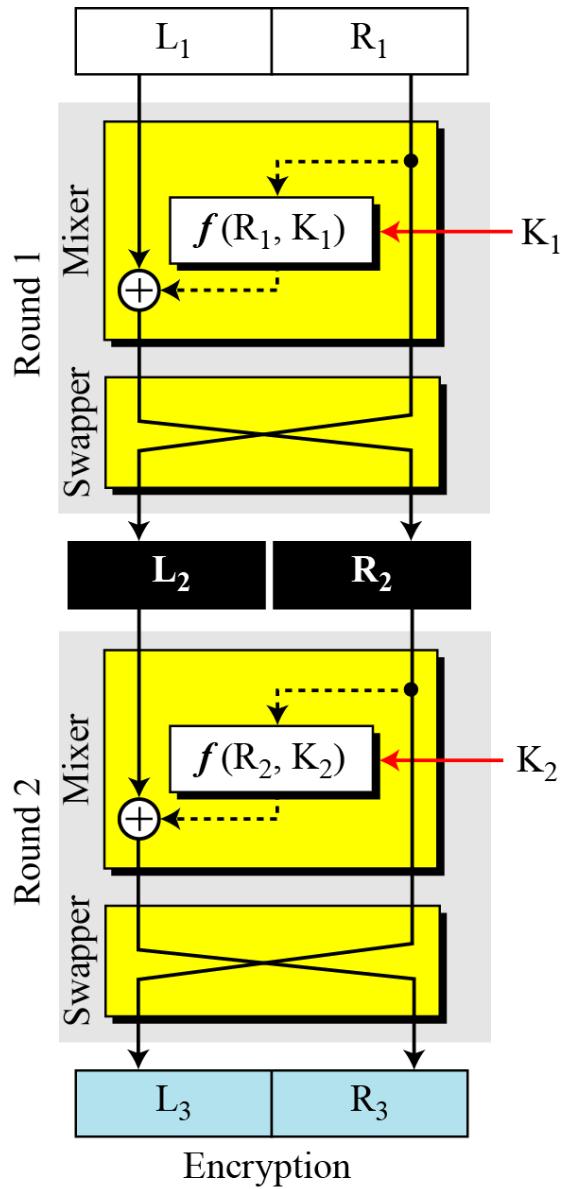
Two algorithms are inverses of each other: If  
 $L_3 = L_2$  and  $R_3 = R_2$

$$R_4 = R_3 = R_2 = R_1$$

$$L_4 = L_3 \oplus f(R_3, K) = L_2 \oplus f(R_2, K) = L_1 \oplus f(R_1, K) \oplus f(R_1, K) = L_1$$

# The final design of Feistel cipher

*Continued*



Final design  
Flaw: no change in Right half.  
Inc: rounds  
Add: swapper

Two algorithms are inverses of each other: If  
 $L_6=L_1$  and  $R_6=R_1$  assuming that  
 $L_4=L_3$  and  $R_4=R_3$

$$\begin{aligned}L_5 &= R_4 \oplus f(L_4, K_2) = R_3 \oplus f(R_2, K_2) = L_2 \oplus f(R_2, K_2) \oplus f(R_2, K_2) = L_2 \\R_5 &= L_4 = L_3 = R_2\end{aligned}$$

Then it is easy to prove that the holds for two  
plaintext blocks

$$\begin{aligned}L_6 &= R_5 \oplus f(L_5, K_1) = R_2 \oplus f(L_2, K_1) = L_1 \oplus f(R_1, K_1) \oplus f(R_1, K_1) = L_1 \\R_6 &= L_5 = L_2 = R_1\end{aligned}$$

## Non-Feistel Ciphers

A non-Feistel cipher uses **only invertible components**. A component in the encryption cipher has the corresponding component in the decryption cipher.

# Feistel Structure for Block Ciphers

- ▶ Feistel proposed applying two or more simple ciphers in sequence so final result is cryptographically stronger than component ciphers
- ▶ n-bit block length; k-bit key length;  $2^k$  transformations
- ▶ Feistel cipher alternates: substitutions, transpositions (permutations)
- ▶ Applies concepts of diffusion and confusion
- ▶ Applied in many ciphers today

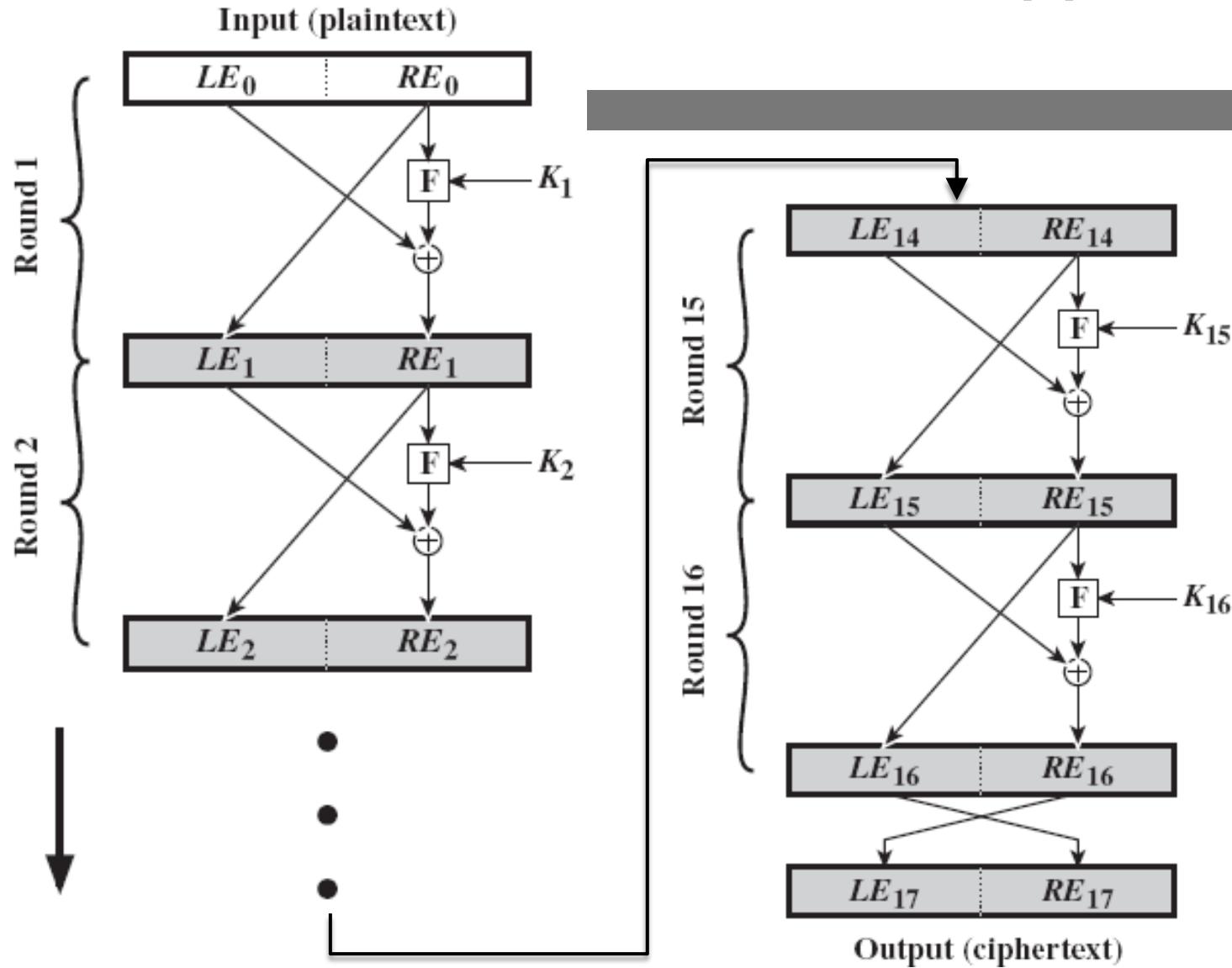
# Feistel Cipher Structure

- Horst Feistel devised the **Feistel cipher**
  - based on concept of **invertible product cipher**
- Partitions input block into two halves
  - ▶ Subkeys (or round keys) generated from key
  - ▶ Round function,  $F$ , applied to right half  $F(KE_i, K_{i+1})$
  - ▶ Apply substitution on left half using XOR
  - ▶ Apply permutation: interchange to halves
- Implements Shannon's S-P net concept

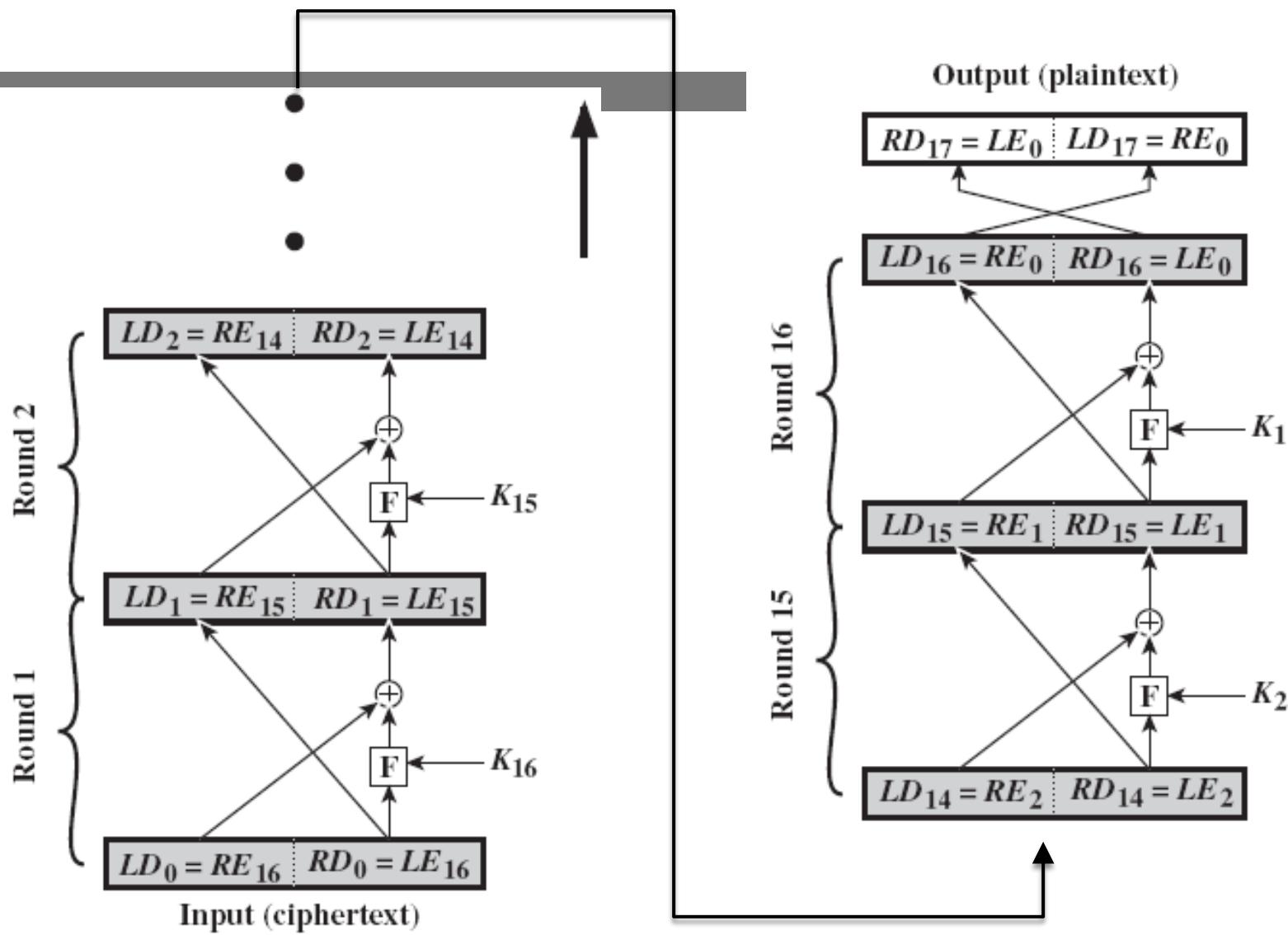
# Using the Feistel Structure

- ▶ Exact implementation depends on various design features
  - Block size, e.g. 64, 128 bits: larger values leads to more diffusion
  - Key size, e.g. 128 bits: larger values leads to more confusion, resistance against brute force
  - Number of rounds, e.g. 16 rounds
  - Subkey generation algorithm: should be complex
  - Round function F: should be complex
- ▶ Other factors include fast encryption in software and ease of analysis
- ▶ Trade-off: security vs. performance

# Feistel Cipher Structure Encryption



# Feistel Cipher Structure Decryption



# General Formula for Encryption/Decryption

- For the  $i$ th iteration of the encryption algorithm

$$LE_i = RE_{i-1}$$

$$RE_i = LE_{i-1} \oplus F(RE_{i-1}, K_i)$$

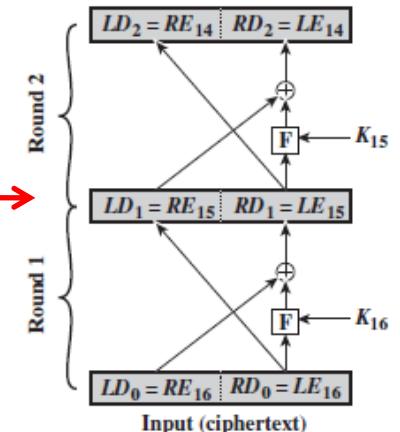
- Rearranging terms gives the decryption:

$$RE_{i-1} = LE_i$$

$$LE_{i-1} = RE_i \oplus F(RE_{i-1}, K_i) = RE_i \oplus F(LE_i, K_i)$$

# Relation between output and input

- Show that the **output** of the **first round** of the **decryption** process is equal to a 32-bit swap of the **input to the sixteenth round of the encryption process.**



- consider the encryption

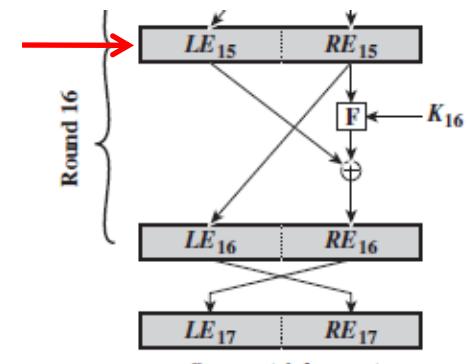
- decryption side

$$\begin{aligned}
 LD_1 &= RD_0 = LE_{16} = RE_{15} \\
 RD_1 &= LD_0 \oplus F(RD_0, K_{16}) \\
 &= RE_{16} \oplus F(RE_{15}, K_{16}) \\
 &= [LE_{15} \oplus F(RE_{15}, K_{16})] \oplus F(RE_{15}, K_{16})
 \end{aligned}$$

- Thus, we have

$$LD_1 = RE_{15} \text{ and } RD_1 = LE_{15}$$

- Therefore, the output of the first round of the decryption process is  $RE_{15} \parallel LE_{15}$ , which is the 32-bit swap of the input to the sixteenth round of the encryption



# Feistel Cipher Design Elements Discussions

- Block size
  - ▣ Larger block sizes mean greater security
- Key size
  - ▣ Larger key size means greater security but may decrease encryption/decryption speed
- Number of rounds
  - ▣ a single round offers inadequate security but that multiple rounds offer increasing security

# Feistel Cipher Design Elements Discussions

- Subkey generation algorithm
  - ▣ Greater complexity leads to greater difficulty of cryptanalysis
- Round function
  - ▣ Same as subkey gen.

# Feistel Cipher Design Elements Discussions

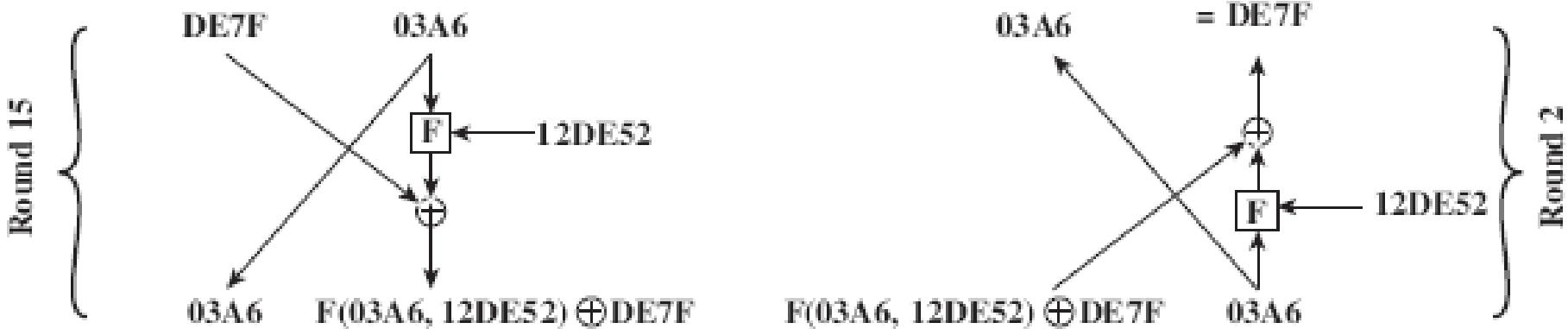
- Fast software en/decryption
  - ▣ the speed of execution of the algorithm becomes a concern
- Ease of analysis
  - ▣ if the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities and therefore develop a higher level of assurance as to its strength

## Dependency on function F

- The derivation **does not require** that F be a **reversible** function.
- For example, F produces a constant output (e.g., all ones) regardless of

Encryption round

Decryption round  
 $F(03A6, 12DE52) \oplus$   
 $[F(03A6, 12DE52) \oplus DE7F]$   
 $= DE7F$

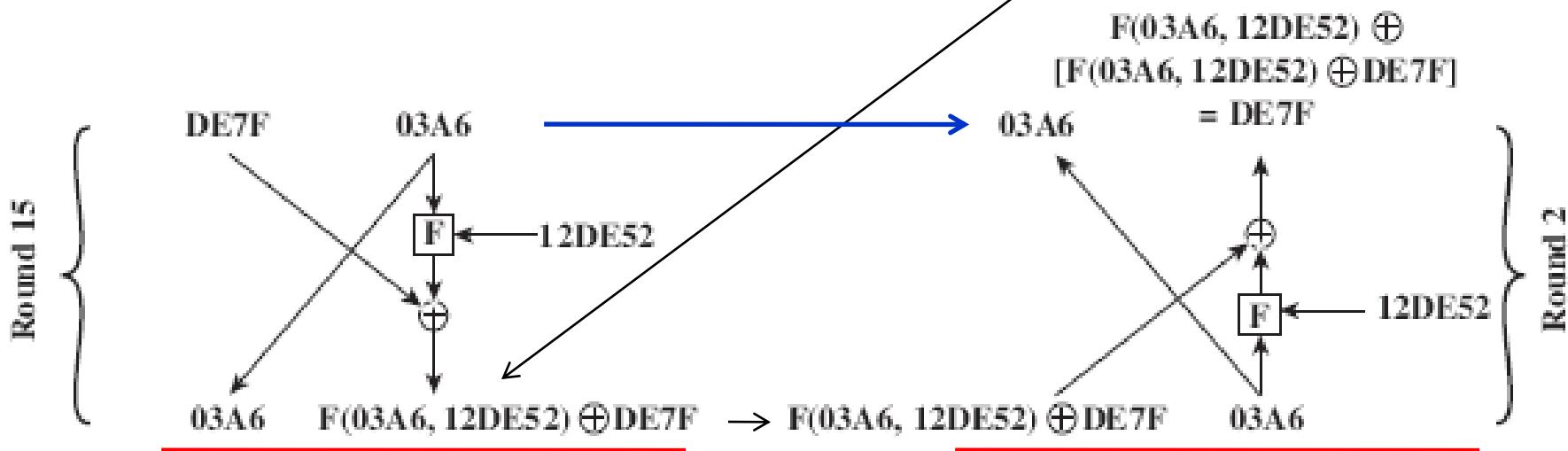


- 15<sup>th</sup> round of encryption corresponds to 2<sup>nd</sup> round of decryption
- Block size is 32 bits (two 16-bit halves) and key size is 24 bits

# Dependency on function F

the key size is 24 bits. Suppose that at the end of encryption round fourteen, the value of the intermediate block (in hexadecimal) is DE7F03A6. Then  $LE_{14} = DE7F$  and  $RE_{14} = 03A6$ . Also assume that the value of  $K_{15}$  is 12DE52. After round 15, we have  $LE_{15} = 03A6$  and  $RE_{15} = F(03A6, 12DE52) \oplus DE7F$ .

Now let's look at the decryption. We assume that  $LD_1 = RE_{15}$  and  $RD_1 = LE_{15}$ , as shown in Figure 3.3, and we want to demonstrate that  $LD_2 = RE_{14}$  and  $RD_2 = LE_{14}$ . So, we start with  $LD_1 = F(03A6, 12DE52) \oplus DE7F$  and  $RD_1 = 03A6$ . Then, from Figure 3.3,  $LD_2 = 03A6 = RE_{14}$  and  $RD_2 = F(03A6, 12DE52) \oplus [F(03A6, 12DE52) \oplus DE7F] = DE7F = LE_{14}$ .



# Symmetric Block Cipher Algorithms



- ▶ DES (Data Encryption Standard)
- ▶ 3DES (Triple DES)
- ▶ AES (Advanced Encryption Standard)

# Data Encryption Standard

- ▶ Symmetric block cipher
  - 56-bit key, 64-bit input block, 64-bit output block
- ▶ One of **most used** encryption systems in world
  - Developed in **1977** by NBS/NIST
  - Designed by **IBM** (Lucifer) with input from NSA
  - Principles used in other ciphers, e.g. 3DES, IDEA
- ▶ Simplified DES (S-DES)
  - ▶ Cipher using principles of DES
  - ▶ Developed for education (not real world use)

# Data Encryption Standard (DES)

- ▶ most widely used block cipher in world
- ▶ adopted in 1977 by NBS (now NIST)
  - as FIPS PUB 46
- ▶ encrypts 64-bit data using 56-bit key
- ▶ has widespread use
- ▶ has considerable controversy over its security

# DES History

- ▶ IBM developed Lucifer cipher
  - by team led by Feistel in late 60's
  - used 64-bit data blocks with 128-bit key
- ▶ then redeveloped as a commercial cipher with input from NSA and others
- ▶ in 1973 NBS issued request for proposals for a national cipher standard
- ▶ IBM submitted their revised Lucifer which was eventually accepted as the DES

# Triple DES

- ▶ Triple DES (3DES) was first standardized for use in financial applications in ANSI standard X9.17 in 1985.
- ▶ 3DES was incorporated as part of the Data Encryption Standard in 1999 with the publication of FIPS 46-3.

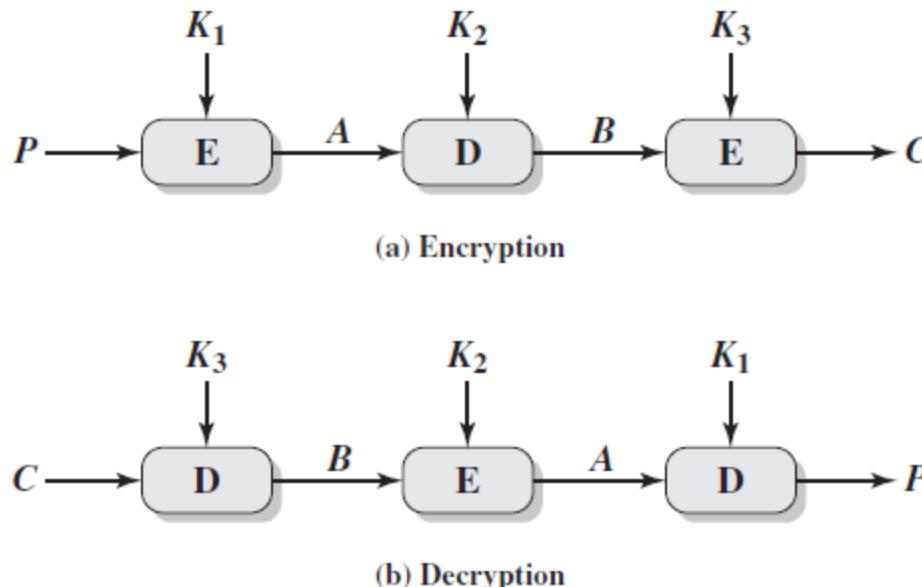


Figure 2.4 Triple DES

# Triple DES

3DES uses three keys and three executions of the DES algorithm. The function follows an encrypt-decrypt-encrypt (EDE) sequence

$$C = E(K_3, D(K_2, E(K_1, P)))$$

where

$C$  = ciphertext

$P$  = plaintext

$E[K, X]$  = encryption of  $X$  using key  $K$

$D[K, Y]$  = decryption of  $Y$  using key  $K$

$$P = D(K_1, E(K_2, D(K_3, C)))$$

There is no cryptographic significance to the use of decryption for the second stage of 3DES encryption.

# Triple DES comments

- ▶ 3DES is the FIPS approved symmetric encryption algorithm of choice.
- ▶ The original DES, which uses a single 56-bit key, is permitted under the standard for legacy systems only. New procurements should support 3DES.
- ▶ Government organizations with legacy DES systems are encouraged to transition to 3DES.
- ▶ It is anticipated that 3DES and the Advanced Encryption Standard (AES) will coexist as FIPS-approved algorithms, allowing for a gradual transition to AES.

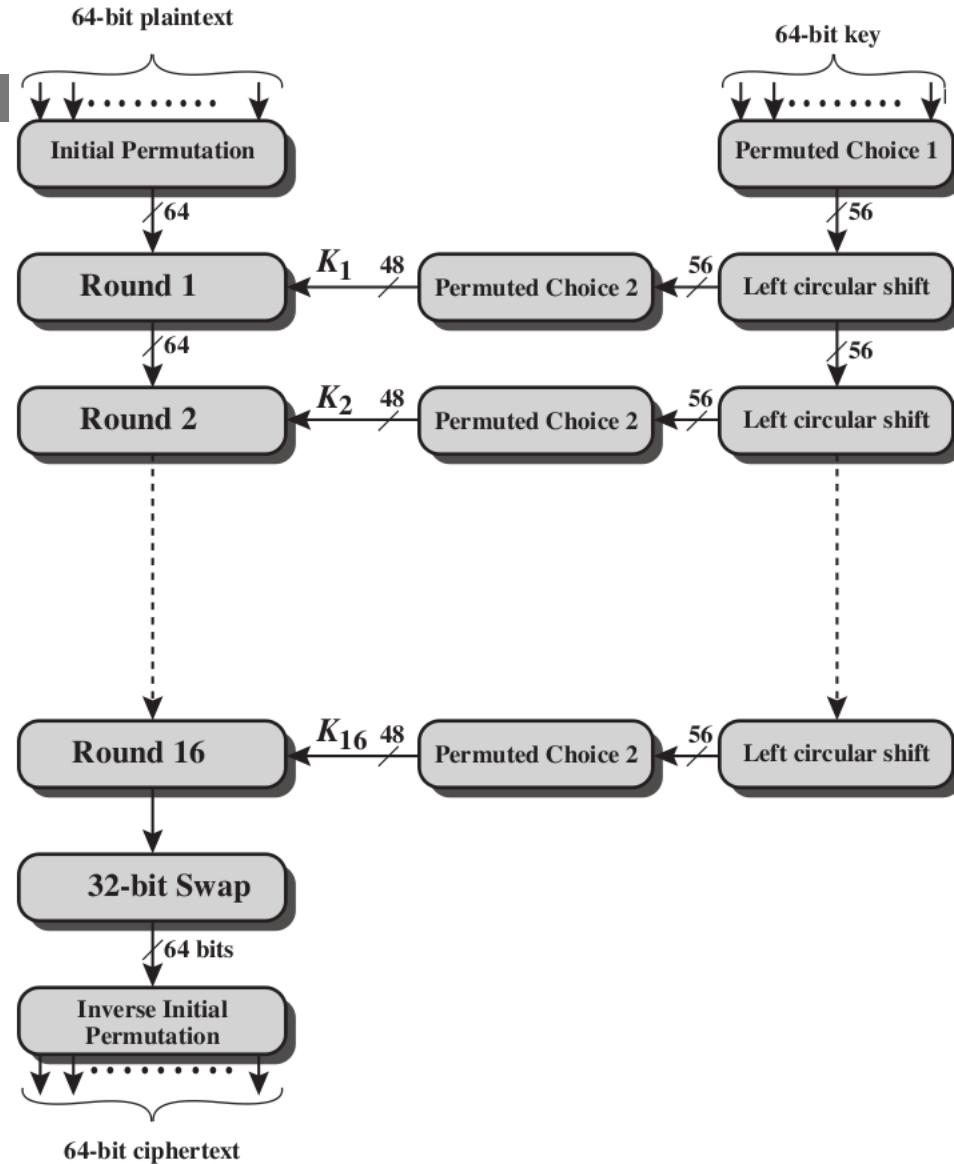
# Triple DES comments

- **FIPS: Federal Information Processing Standards**
- The purpose of FIPS is to ensure that all federal government and agencies adhere to the same guidelines regarding security and communication.

# DES

- ▶ For DES, data are encrypted in 64-bit blocks using a 56-bit key.
- ▶ The algorithm transforms 64-bit input in a series of steps into a 64-bit output.
- ▶ The same steps, with the same key, are used to reverse the encryption.
- ▶ With the exception of the initial and final permutations, DES has the exact structure of a Feistel cipher.

# General DES Encryption Algorithm



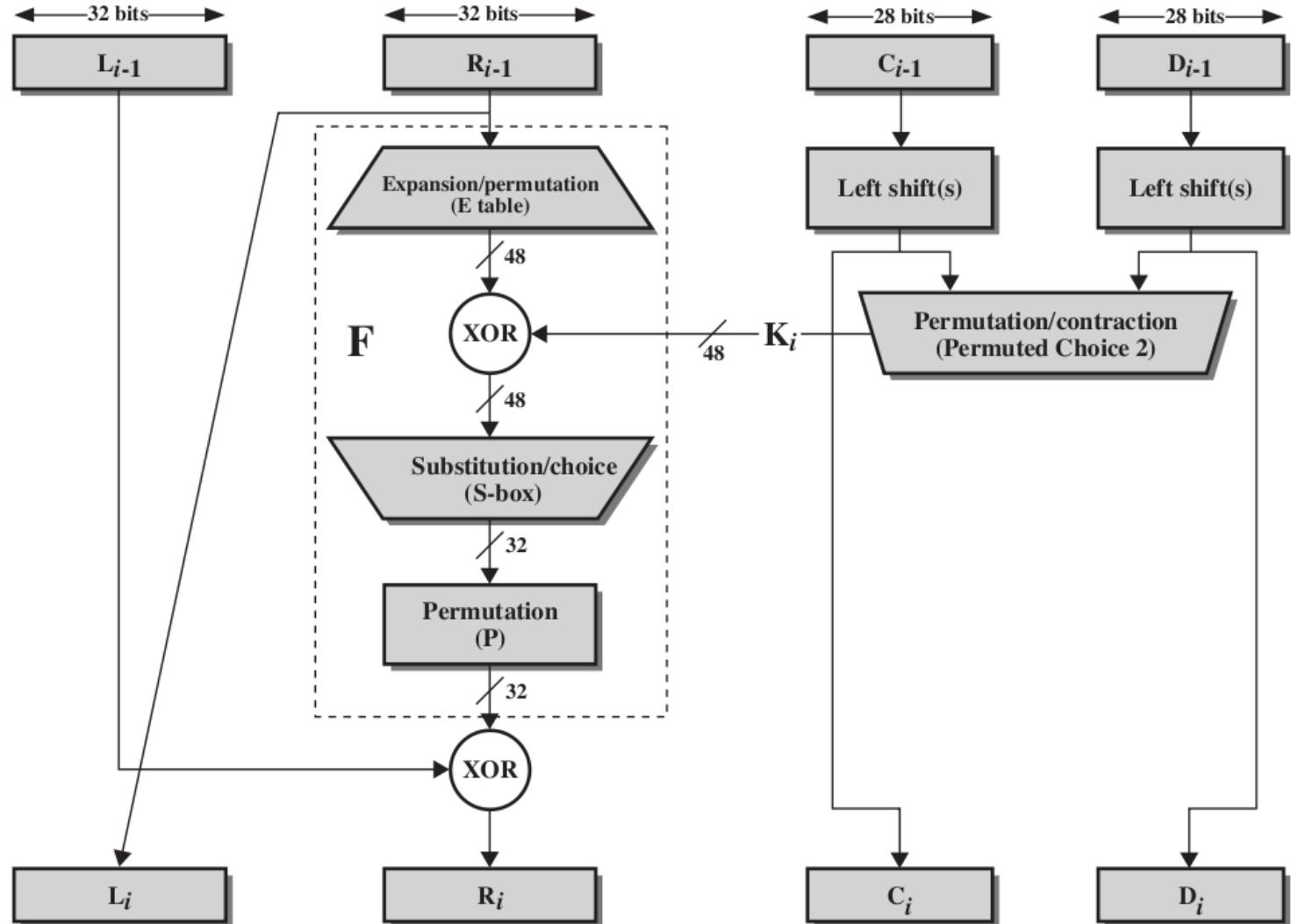
# DES Encryption

- As with any encryption scheme, there are **two inputs** to the encryption function: **the plaintext** to be encrypted and **the key**
- the processing of the plaintext proceeds in **three phases**.
  1. First, the **64-bit plaintext** passes through an **initial permutation (IP)** that rearranges the bits to produce the **permuted input**.
  2. This is followed by a phase consisting of **sixteen rounds** of the same function, which involves both **permutation** and **substitution** functions.
  3. The **left** and **right halves** of the output are **swapped** to produce the **preoutput**.
  4. Finally, the **preoutput** is passed through a **permutation [IP <sup>-1</sup>]** that is the inverse of the initial permutation function, to produce the **64-bit ciphertext**.

# Key generation

- ▶ Initially, the key is passed through a permutation function.
- ▶ Then, for each of the sixteen rounds, a subkey ( $K_i$ ) is produced by the combination of a left circular shift and a permutation.

# Single Round of DES Algorithm



# A DES Decryption

- ▶ 1. As with any Feistel cipher, decryption uses the **same algorithm as encryption**, except that the application of the **subkeys is reversed**.
- ▶ 2. Additionally, the initial and final permutations are **reversed**.

# Permutation Tables for DES

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

Input bit 58 goes to output bit 1

Input bit 50 goes to output bit 2, ...

Even bits to LH half, odd bits to RH half

Quite regular in structure (easy in h/w)

# Permutation Tables for DES

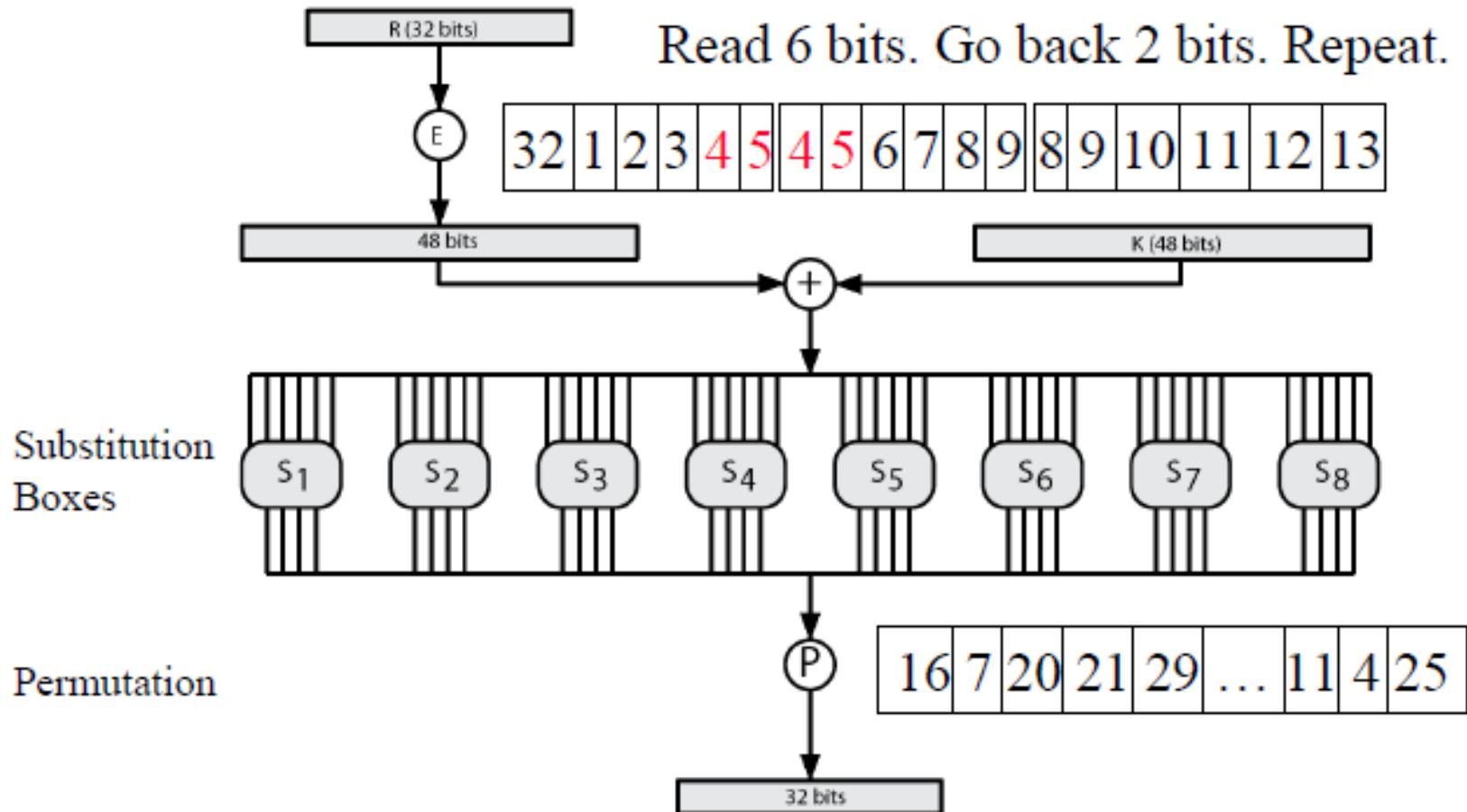
(c) Expansion Permutation (E)

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

(d) Permutation Function (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

# Calculation of F(R,K)



# Substitution boxes

- Map 6 to 4 bits
- Outer bits 1 & 6 (**row** bits) select one row of 4
- Inner bits 2-5 (**column** bits) are substituted
- Example:

Input bits 1 and 6		Input bits 2 thru 5														
↓	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1110	0100	1101	0001	0010	1111	1011	1000	0011	1010	0110	1100	0101	1001	0000	0111
01	0000	1111	0111	0100	1110	0010	1101	0001	1010	0110	1100	1011	1001	0101	0011	1000
10	0100	0001	1110	1000	1101	0110	0010	1011	1111	1100	1001	0111	0011	1010	0101	0000
11	1111	1100	1000	0010	0100	1001	0001	0111	0101	1011	0011	1110	1010	0000	0110	1101

# Definition of DES S-Boxes

$S_1$	14 4 13 1 2 15 11 8 3 10 6 12 5 9 0 7	0 15 7 4 14 2 13 1 10 6 12 11 9 5 3 8	4 1 14 8 13 6 2 11 15 12 9 7 3 10 5 0	15 12 8 2 4 9 1 7 5 11 3 14 10 0 6 13
$S_2$	15 1 8 14 6 11 3 4 9 7 2 13 12 0 5 10	3 13 4 7 15 2 8 14 12 0 1 10 6 9 11 5	0 14 7 11 10 4 13 1 5 8 12 6 9 3 2 15	13 8 10 1 3 15 4 2 11 6 7 12 0 5 14 9
$S_3$	10 0 9 14 6 3 15 5 1 13 12 7 11 4 2 8	13 7 0 9 3 4 6 10 2 8 5 14 12 11 15 1	13 6 4 9 8 15 3 0 11 1 2 12 5 10 14 7	1 10 13 0 6 9 8 7 4 15 14 3 11 5 2 12
$S_4$	7 13 14 3 0 6 9 10 1 2 8 5 11 12 4 15	13 8 11 5 6 15 0 3 4 7 2 12 1 10 14 9	10 6 9 0 12 11 7 13 15 1 3 14 5 2 8 4	3 15 0 6 10 1 13 8 9 4 5 11 12 7 2 14

# Definition of DES S-Boxes

$S_5$	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
$S_6$	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
$S_7$	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
$S_8$	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

# DES Key Schedule Calculation

- ❑ Permutation PC1 divides 56-bits in two 28-bit halves
- ❑ Rotate **each half** separately either 1 or 2 places depending on the **key rotation schedule K**
- ❑ Select 24-bits from each half & permute them by PC2

(a) Input Key

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	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

(b) Permuted Choice One (PC-1)

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

(c) Permuted Choice Two (PC-2)

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

(d) Schedule of Left Shifts

Round Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits Rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	1	

# The Avalanche Effect

- ▶ Aim: small change in key (or plaintext) produces large change in ciphertext
  - ▶ Avalanche effect is present in DES (good for security)
  - ▶ Following examples show the number of bits that change in output when two different inputs are used, **differing by 1 bit**
    - ▶ Plaintext 1: 02468aceeca86420
    - ▶ Plaintext 2: 12468aceeca86420
    - ▶ Ciphertext difference: 32 bits
      - Key 1: 0f1571c947d9e859
      - Key 2: 1f1571c947d9e859
      - **Ciphertext difference: 307**
- shows the result when the fourth bit of the plaintext is changed, so that the plaintext is **12468aceeca86420**.

# DES example

- ▶ For this example, the plaintext is a hexadecimal palindrome. The plaintext, key, and resulting ciphertext are as follows:

Plaintext:	02468aceeca86420
Key:	0f1571c947d9e859
Ciphertext:	da02ce3a89ecac3b

# Results

Table 3.2 DES Example

Round	$K_i$	$L_i$	$R_i$
<b>IP</b>		5a005a00	3cf03c0f
<b>1</b>	1e030f03080d2930	3cf03c0f	bad22845
<b>2</b>	0a31293432242318	bad22845	99e9b723
<b>3</b>	23072318201d0c1d	99e9b723	0bae3b9e
<b>4</b>	05261d3824311a20	0bae3b9e	42415649
<b>5</b>	3325340136002c25	42415649	18b3fa41
<b>6</b>	123a2d0d04262a1c	18b3fa41	9616fe23
<b>7</b>	021f120b1c130611	9616fe23	67117cf2
<b>8</b>	1c10372a2832002b	67117cf2	c11bfcc09
<b>9</b>	04292a380c341f03	c11bfcc09	887fbcc6c
<b>10</b>	2703212607280403	887fbcc6c	600f7e8b
<b>11</b>	2826390c31261504	600f7e8b	f596506e
<b>12</b>	12071c241a0a0f08	f596506e	738538b8
<b>13</b>	300935393c0d100b	738538b8	c6a62c4e
<b>14</b>	311e09231321182a	c6a62c4e	56b0bd75
<b>15</b>	283d3e0227072528	56b0bd75	75e8fd8f
<b>16</b>	2921080b13143025	75e8fd8f	25896490
<b>IP<sup>-1</sup></b>		da02ce3a	89ecac3b

Note: DES subkeys are shown as eight 6-bit values in hex format

shows the progression of the algorithm.

## Avalanche Effect in DES: Change in Plaintext

The second column of the table shows the intermediate 64-bit values at the end of each round for the two plaintexts.

The third column shows the number of bits that differ between the two intermediate values.

Round		$\delta$
	02468aceeca86420 12468aceeca86420	1
1	3cf03c0fbad22845 3cf03c0fbad32845	1
2	bad2284599e9b723 bad3284539a9b7a3	5
3	99e9b7230bae3b9e 39a9b7a3171cb8b3	18
4	0bae3b9e42415649 171cb8b3ccaca55e	34
5	4241564918b3fa41 ccaca55ed16c3653	37
6	18b3fa419616fe23 d16c3653cf402c68	33
7	9616fe2367117cf2 cf402c682b2cefbc	32
8	67117cf2c11bfc09	33

Round		$\delta$
9	c11bfc09887fbc6c 99f911532eed7d94	32
10	887fbc6c600f7e8b 2eed7d94d0f23094	34
11	600f7e8bf596506e d0f23094455da9c4	37
12	f596506e738538b8 455da9c47f6e3cf3	31
13	738538b8c6a62c4e 7f6e3cf34bc1a8d9	29
14	c6a62c4e56b0bd75 4bc1a8d91e07d409	33
15	56b0bd7575e8fd8f 1e07d4091ce2e6dc	31
16	75e8fd8f25896490 1ce2e6dc365e5f59	32
IP <sup>-1</sup>	da02ce3a89ecac3b	32

# Avalanche Effect in DES: Change in Key

shows a similar test using the original plaintext of with two keys that differ in only the fourth bit position:

Round		$\delta$
	02468aceeca86420 02468aceeca86420	0
1	3cf03c0fbad22845 3cf03c0f9ad628c5	3
2	bad2284599e9b723 9ad628c59939136b	11
3	99e9b7230bae3b9e 9939136b768067b7	25
4	0bae3b9e42415649 768067b75a8807c5	29
5	4241564918b3fa41 5a8807c5488dbe94	26
6	18b3fa419616fe23 488dbe94aba7fe53	26
7	9616fe2367117cf2 aba7fe53177d21e4	27
8	67117cf2c11bfc09 177d21e4548f1de4	32
IP <sup>-1</sup>	da02ce3a89ecac3b ee92b50606b62b0b	30
9	c11bfc09887fbc6c 548f1de471f64dfd	34
10	887fbc6c600f7e8b 71f64dfd4279876c	36
11	600f7e8bf596506e 4279876c399fdc0d	32
12	f596506e738538b8 399fdc0d6d208dbb	28
13	738538b8c6a62c4e 6d208dbbb9bdeeaad	33
14	c6a62c4e56b0bd75 b9bdeeaad2c3a56f	30
15	56b0bd7575e8fd8f d2c3a56f2765c1fb	33
16	75e8fd8f25896490 2765c1fb01263dc4	30

# Concerns of DES

## Key size and the nature of the algorithm

- ▶ Although 64 bit initial key, only 56 bits used in encryption (other 8 for parity check)
- ▶  $2^{56} = 7.2 * 10^{16}$ 
  - 1977: estimated cost \$US20m to build machine to break in 10 hours
  - 1998: EFF built machine for \$US250k to break in 3 days
  - Today: 56 bits considered too short to withstand brute force attack
- ▶ Recent offerings confirm this. Both Intel and AMD now offer hardware-based instructions to accelerate the use of AES. Test run on a contemporary multicore Intel machine resulted in an encryption rate of about **half a billion encryptions per second.**
- ▶ 3DES uses 128-bit keys

Table 3.5 Average Time Required for Exhaustive Key Search

Key Size (bits)	Cipher	Number of Alternative Keys	Time Required at $10^9$ Decryptions/s	Time Required at $10^{13}$ Decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	$2^{55} \text{ ns} = 1.125 \text{ years}$	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	$2^{127} \text{ ns} = 5.3 \times 10^{21} \text{ years}$	$5.3 \times 10^{17} \text{ years}$
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	$2^{167} \text{ ns} = 5.8 \times 10^{33} \text{ years}$	$5.8 \times 10^{29} \text{ years}$
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	$2^{191} \text{ ns} = 9.8 \times 10^{40} \text{ years}$	$9.8 \times 10^{36} \text{ years}$
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	$2^{255} \text{ ns} = 1.8 \times 10^{60} \text{ years}$	$1.8 \times 10^{56} \text{ years}$
26 characters (permutation)	Monoalphabetic	$2! = 4 \times 10^{26}$	$2 \times 10^{26} \text{ ns} = 6.3 \times 10^9 \text{ years}$	$6.3 \times 10^6 \text{ years}$

# DES Design Controversy (**Concerns**)

- ▶ although DES standard is public, considerable controversy over design (two concerns)
  - in choice of 56-bit key (vs Lucifer 128-bit)
  - and because design criteria were classified
- ▶ subsequent events and public analysis show in fact design was appropriate
- ▶ use of DES has flourished
  - especially in financial applications
  - still standardised for legacy application use

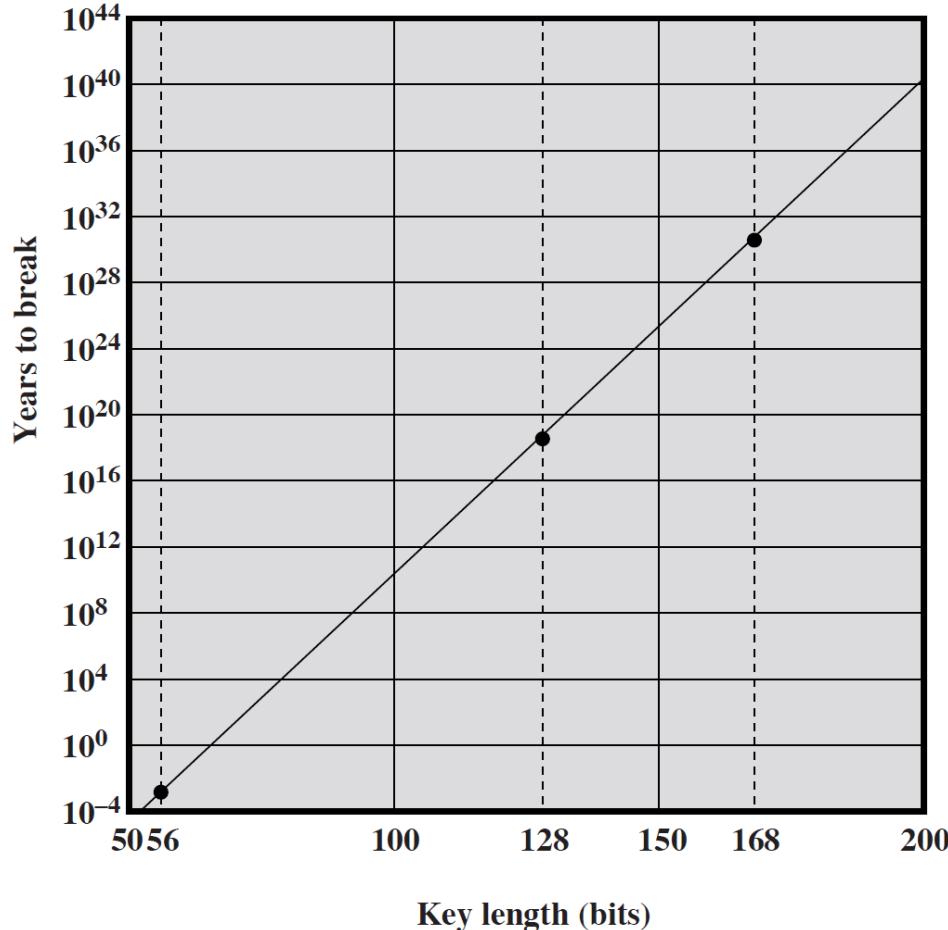
# Concern of DES

- ▶ **The Nature of the DES Algorithm**
- ▶ Another concern is the possibility that **cryptanalysis is possible** by exploiting the characteristics of the DES algorithm
- ▶ Because the design criteria for these S-boxes, and indeed for the entire algorithm, **were not made public**, there is a suspicion that the boxes were constructed in such a way that **cryptanalysis is possible** for an opponent who knows **the weaknesses in the S-boxes**.

# Time to Break a DES Code (assuming $10^6$ decryptions/ $\mu$ s)

Using Electronic  
Frontier  
Foundation (EFF)  
DES cracker

Appx 10 hrs.  
for DES



# Attacks on DES

## Timing Attacks

- ▶ Information gained about key/plaintext by observing how **long implementation takes to decrypt**
- ▶ No known useful attacks on DES

## Differential Cryptanalysis

- ▶ Observe how pairs of plaintext blocks evolve
- ▶ Break DES in 247 encryptions (compared to 255); but require 247 chosen plaintexts

## Linear Cryptanalysis

- ▶ Find linear approximations of the transformations
- ▶ Break DES using 243 known plaintexts

## Differential Cryptanalysis

- Chosen Plaintext attack: Get ciphertext for a given plaintext
- Get the  $(\Delta X, \Delta Y)$  pairs, where  $\Delta X$  is the difference in plaintext and  $\Delta Y$  is the difference in ciphertext
- Some  $(\Delta X, \Delta Y)$  pairs are more likely than others, if those pairs are found, some key values are more likely so you can reduce the amount of brute force search
- Straightforward brute force attack on DES requires  $2^{55}$  plaintexts
- Using differential cryptanalysis, DES can be broken with  $2^{47}$  plaintexts.

But finding appropriate plaintexts takes some trials and so the total amount of effort is  $2^{55.1}$  which is more than straight forward brute force attack

⇒ DES is resistant to differential cryptanalysis

# Linear Cryptanalysis

- Bits in plaintext, ciphertext, and keys may have a linear relationship. For example:

$$P_1 \oplus P_2 \oplus C_3 = K_2 \oplus K_5$$

- In a good cipher, the relationship should hold w probability  $\frac{1}{2}$ . If any relationship has probability 1, the cipher is easy to break.  
If any relationship has probability 0, the cipher is easy to break.
- Bias =  $|$ Probability of linear relationship – 0.5 $|$
- Find the linear approximation with the highest bias  
⇒ Helps reduce the brute force search effort.
- This method can be used to find the DES key given  $2^{43}$  plaintexts.

# Choosing F

- ▶ **Non-linearity** in rough terms, the more difficult it is to approximate F by a set of linear equations, the more nonlinear F is.
- ▶ A more stringent version of this is the **strict avalanche criterion (SAC)**, which states that any output bit  $j$  of an S-box should change with probability  $1/2$  when any single input bit  $i$  is inverted for all  $i, j$ .
- ▶ Another criterion proposed is the **bit independence criterion (BIC)**, which states that output bits  $j$  and  $k$  should change independently when any single input bit  $i$  is inverted for all  $i, j$ , and  $k$ .

# DES Algorithm Design

DES was designed in private; questions about the motivation

of the design

- ▶ S-Boxes provide non-linearity: important part of DES, generally considered to be secure
- ▶ S-Boxes provide increased confusion
- ▶ Permutation P chosen to increase diffusion

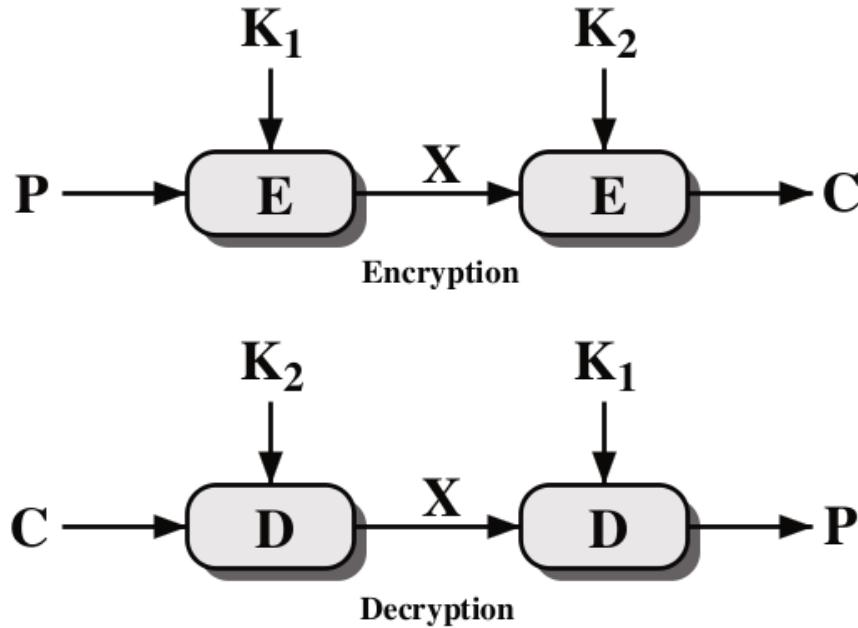
# Multiple Encryption with DES

- ▶ DES is vulnerable to brute force attack
- ▶ Alternative block cipher that makes use of DES software/equipment/knowledge: encrypt multiple times with different keys

## Options:

- ▶ 1. Double DES: not much better than single DES
- ▶ 2. Triple DES (3DES) with 2 keys: brute force  $2^{112}$
- ▶ 3. Triple DES with 3 keys: brute force  $2^{168}$

# Double Encryption

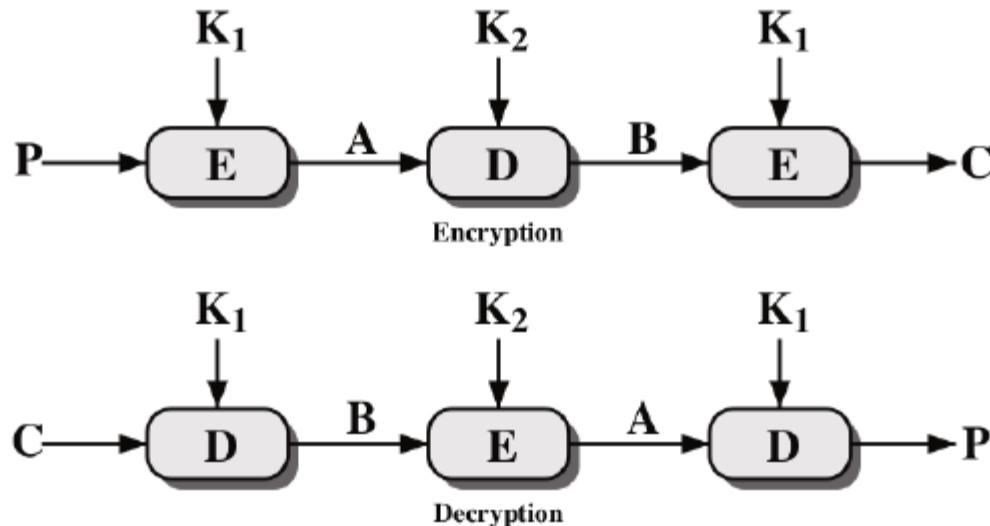


- ▶ For DES, 2 56-bit keys, meaning 112-bit key length
- ▶ Requires  $2^{111}$  operations for brute force?
- ▶ Meet-in-the-middle attack makes it easier

# Meet-in-the-Middle Attack

- ▶ Double DES Encryption:  $C = E(K_2, E(K_1, P))$
- ▶ Say  $X = E(K_1, P) = D(K_2, C)$
- ▶ Attacker knows two plaintext, ciphertext pairs  $(P_a, C_a)$  and  $(P_b, C_b)$ 
  1. Encrypt  $P_a$  using all  $2^{56}$  values of  $K_1$  to get multiple values of  $X$
  2. Store results in table and sort by  $X$
  3. Decrypt  $C_a$  using all  $2^{56}$  values of  $K_2$
  4. As each decryption result produced, check against table
  5. If match, check current  $K_1, K_2$  on  $C_b$ . If  $P_b$  obtained, then accept the keys
- ▶ With two known plaintext, ciphertext pairs, probability of successful attack is almost 1
- ▶ Encrypt/decrypt operations required:  $2^{56}$  (twice as many as single DES)

# Triple Encryption



- ▶ 2 keys, 112 bits
- ▶ 3 keys, 168 bits
- ▶ Why E-D-E? To be compatible with single DES:

$$C = E(K_1, D(K_1, E(K_1, P))) = E(K_1, P)$$

# Other Symmetric Encryption Algorithms

- ▶ Blowfish (Schneier, 1993): 64 bit blocks/32–448 bit keys; Feistel structure
- ▶ Twofish (Schneier et al, 1998): 128/128, 192, 256; Feistel structure
- ▶ Serpent (Anderson et al, 1998): 128/128, 192, 256; Substitution-permutation network
- ▶ Camellia (Mitsubishi/NTT, 2000): 128/128, 192, 256; Feistel structure
- ▶ IDEA (Lai and Massey, 1991): 64/128
- ▶ CAST-128 (Adams and Tavares, 1996): 64/40–128; Feistel structure
- ▶ CAST-256 (Adams and Tavares, 1998): 128/up to 256; Feistel structure
- ▶ RC5 (Rivest, 1994): 32, 64 or 128/up to 2040; Feistel-like structure
- ▶ RC6 (Rivest et al, 1998): 128/128, 192, 256; Feistel structure

# Cryptanalysis on Block Ciphers

Cipher	Method	Key space	Required resources:		
			Time	Memory	Known data
DES	Brute force	$2^{56}$	$2^{56}$	-	-
3DES	MITM	$2^{168}$	$2^{111}$	$2^{56}$	$2^2$
3DES	Lucks	$2^{168}$	$2^{113}$	$2^{88}$	$2^{32}$
AES 128	Biclique	$2^{128}$	$2^{126.1}$	$2^8$	$2^{88}$
AES 256	Biclique	$2^{256}$	$2^{254.4}$	$2^8$	$2^{40}$

- ▶ Known data: chosen pairs of (plaintext, ciphertext)
- ▶ MITM: Meet-in-the-middle
- ▶ Lucks: S. Lucks, Attacking Triple Encryption, in *Fast Software Encryption*, Springer, 1998
- ▶ Biclique: Bogdanov, Khovratovich and Rechberger, Biclique Cryptanalysis of the Full AES, in *ASIACRYPT2011*, Springer, 2011

# Multiple Encryption & DES

- ▶ clear a replacement for DES was needed
  - theoretical attacks that can break it
  - demonstrated exhaustive key search attacks
- ▶ AES is a new cipher alternative
  - ▶ prior to this alternative was to use multiple encryption with DES implementations
  - ▶ Triple-DES is the chosen form

# Double-DES?

- ▶ could use 2 DES encrypts on each block
  - $C = E_{K2}(E_{K1}(P))$
- ▶ issue of reduction to single stage
- ▶ and have “meet-in-the-middle” attack
  - ▶ works whenever use a cipher twice
  - ▶ since  $X = E_{K1}(P) = D_{K2}(C)$
  - ▶ attack by encrypting P with all keys and store
  - ▶ then decrypt C with keys and match X value
  - ▶ takes  $O(2^{56})$  steps

# Triple-DES with Two-Keys

- ▶ hence must use 3 encryptions
  - would seem to need 3 distinct keys
- ▶ but can use 2 keys with E-D-E sequence
  - $C = E_{K1}(D_{K2}(E_{K1}(P)))$
  - nb encrypt & decrypt equivalent in security
  - if  $K1=K2$  then can work with single DES
- ▶ standardized in ANSI X9.17 & ISO8732
- ▶ no current known practical attacks
  - several proposed impractical attacks might become basis of future attacks

# Triple-DES with Three-Keys

- ▶ although no practical attacks on two-key Triple-DES have some concerns
  - Two-key: key length =  $56*2 = 112$  bits
  - Three-key: key length =  $56*3 = 168$  bits
- ▶ can use Triple-DES with Three-Keys to avoid even these
  - $C = E_{K3}(D_{K2}(E_{K1}(P)))$
- ▶ has been adopted by some Internet applications, eg PGP, S/MIME



# Public-Key Cryptography

# Outline

1. Public Key Encryption
2. Symmetric vs. Public-Key
3. RSA Public Key Encryption
4. RSA Key Construction
5. Optimizing Private Key Operations
6. RSA Security

# Misconceptions Concerning Public-Key Encryption



- Public-key encryption is more secure from cryptanalysis than symmetric encryption
- Public-key encryption is a general-purpose technique that has made symmetric encryption obsolete
- There is a feeling that key distribution is trivial when using public-key encryption, compared to the cumbersome handshaking involved with key distribution centers for symmetric encryption

# Principles of Public-Key Cryptosystems

- The concept of public-key cryptography evolved from an attempt to attack two of the most difficult problems associated with symmetric encryption:

Key distribution

Digital signatures

- Whitfield Diffie and Martin Hellman from Stanford University achieved a breakthrough in 1976 by coming up with a method that addressed both problems and was radically different from all previous approaches to cryptography

# Public-Key Cryptosystems

- A public-key encryption scheme has six ingredients:

Plaintext

The readable message or data that is fed into the algorithm as input

Encryption algorithm

Performs various transformations on the plaintext

Public key

Used for encryption or decryption

Private key

Used for encryption or decryption

Ciphertext

The scrambled message produced as output

Decryption algorithm

Accepts the ciphertext and the matching key and produces the original plaintext

# Symmetric and Asymmetric-key Cryptography

Symmetric and asymmetric-key cryptography **will exist in parallel and continue to serve the community.** We actually believe that they are complements of each other; the advantages of one can compensate for the disadvantages of the other.

---

Symmetric-key cryptography is based on sharing secrecy; asymmetric-key cryptography is based on personal secrecy.

---

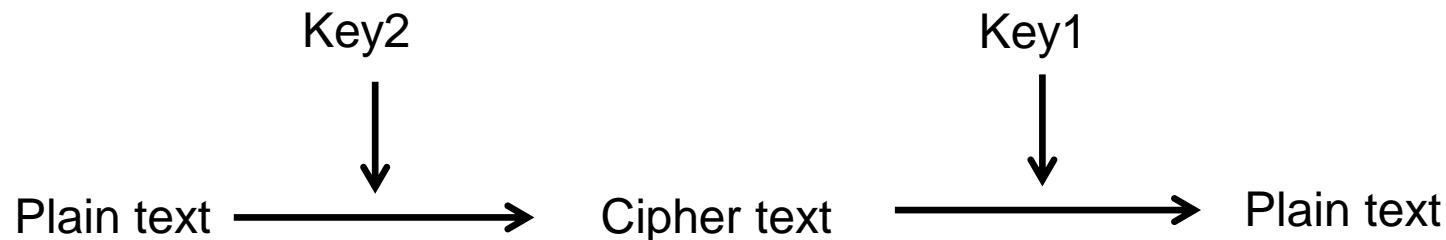
# Need for Both

There is a very important fact that is sometimes misunderstood: The advent of asymmetric-key cryptography does not eliminate the need for symmetric-key cryptography.

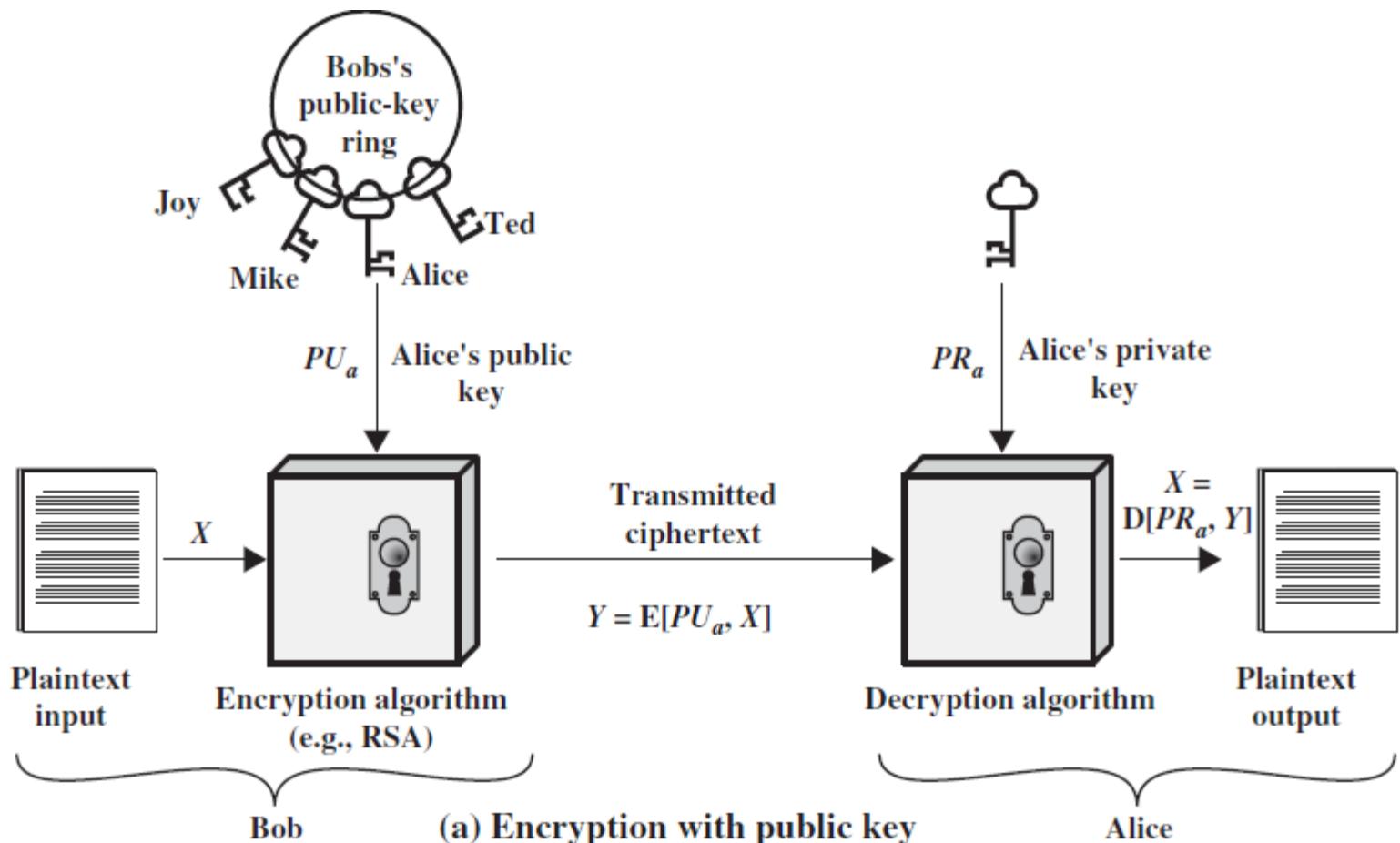
# Public Key Cryptography

- Invented in 1975 by Diffie and Hellman at Stanford
- Encrypted\_Message = Encrypt (Key1, Message)
- Message = Decrypt (Key2, Encrypted\_Message)
- Keys are **interchangeable**
- One key is made **public** while the other is kept **private**
- Sender knows only public key of the receiver =>**Asymmetric**

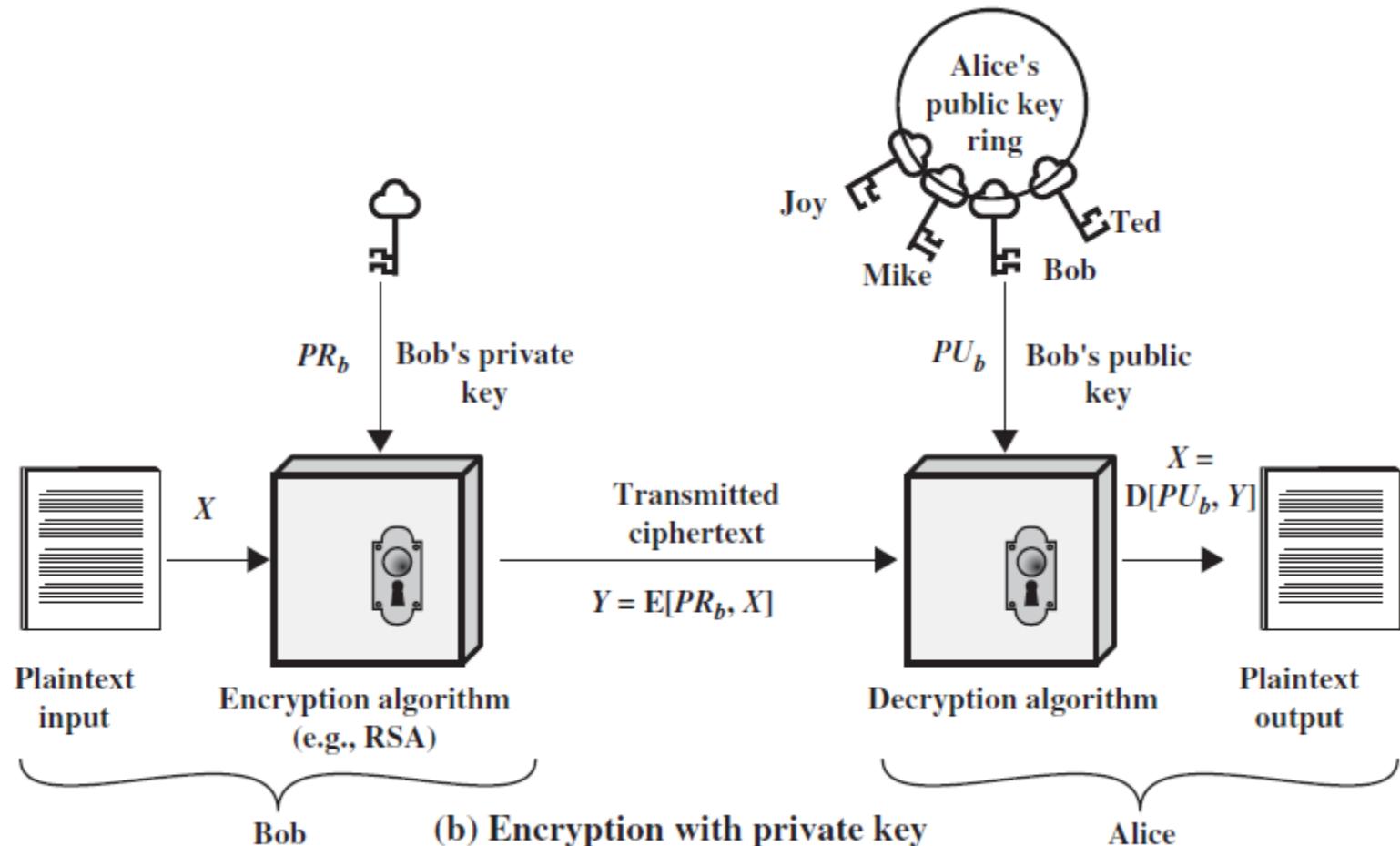
# Public Key Cryptography



# Encryption with Public Key



# Encryption with Private Key



# Why Public-Key Cryptography?

- Developed to address two key issues:
  - **key distribution** – how to have secure communications in general without having to trust a KDC with your key
  - **digital signatures** – how to verify a message comes intact from the claimed sender

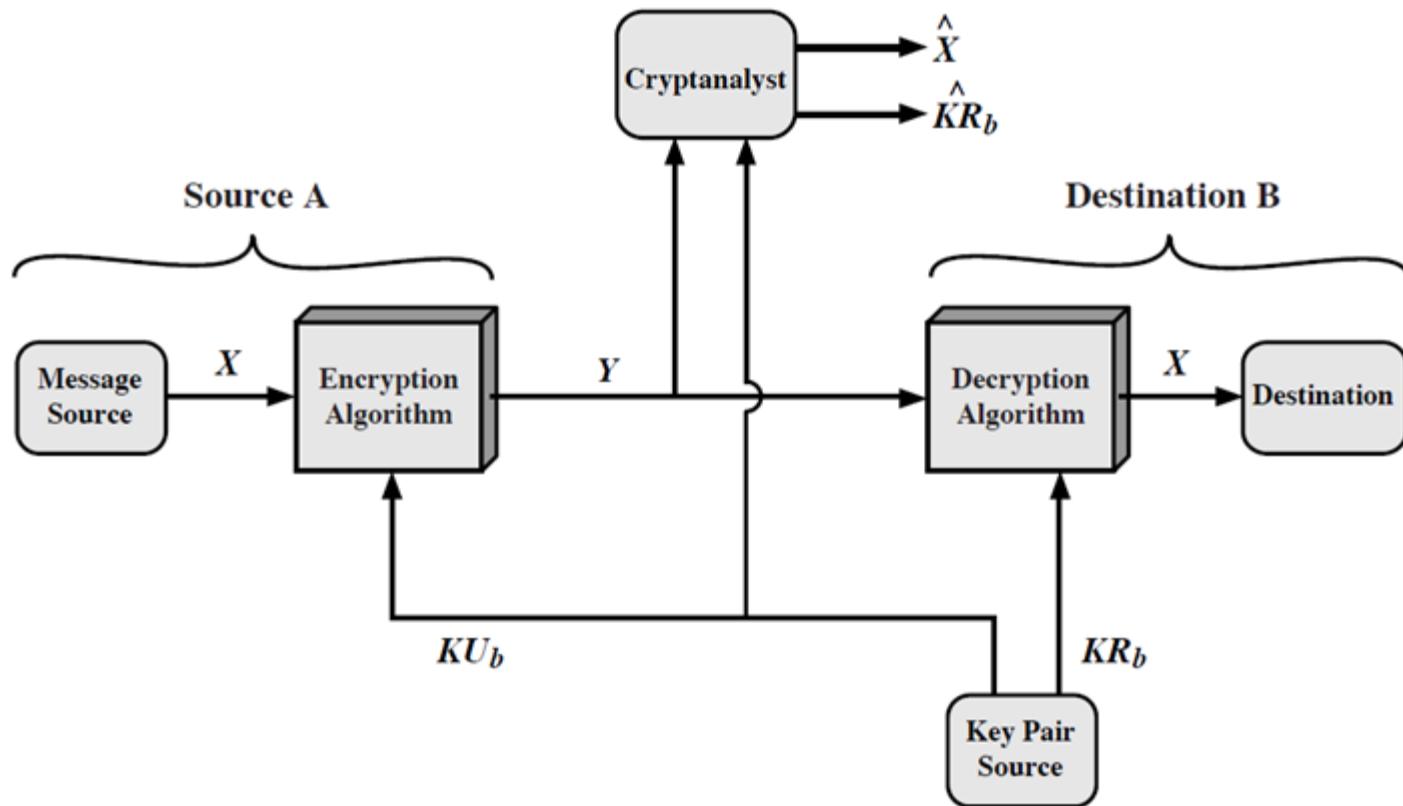
# Conventional and Public-Key Encryption

Conventional Encryption	Public-Key Encryption
<p><i>Needed to Work:</i></p> <ol style="list-style-type: none"><li>1. The same algorithm with the same key is used for encryption and decryption.</li><li>2. The sender and receiver must share the algorithm and the key.</li></ol> <p><i>Needed for Security:</i></p> <ol style="list-style-type: none"><li>1. The key must be kept secret.</li><li>2. It must be impossible or at least impractical to decipher a message if the key is kept secret.</li><li>3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.</li></ol>	<p><i>Needed to Work:</i></p> <ol style="list-style-type: none"><li>1. One algorithm is used for encryption and a related algorithm for decryption with a pair of keys, one for encryption and one for decryption.</li><li>2. The sender and receiver must each have one of the matched pair of keys (not the same one).</li></ol> <p><i>Needed for Security:</i></p> <ol style="list-style-type: none"><li>1. One of the two keys must be kept secret.</li><li>2. It must be impossible or at least impractical to decipher a message if one of the keys is kept secret.</li><li>3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.</li></ol>

# Public-Key Characteristics

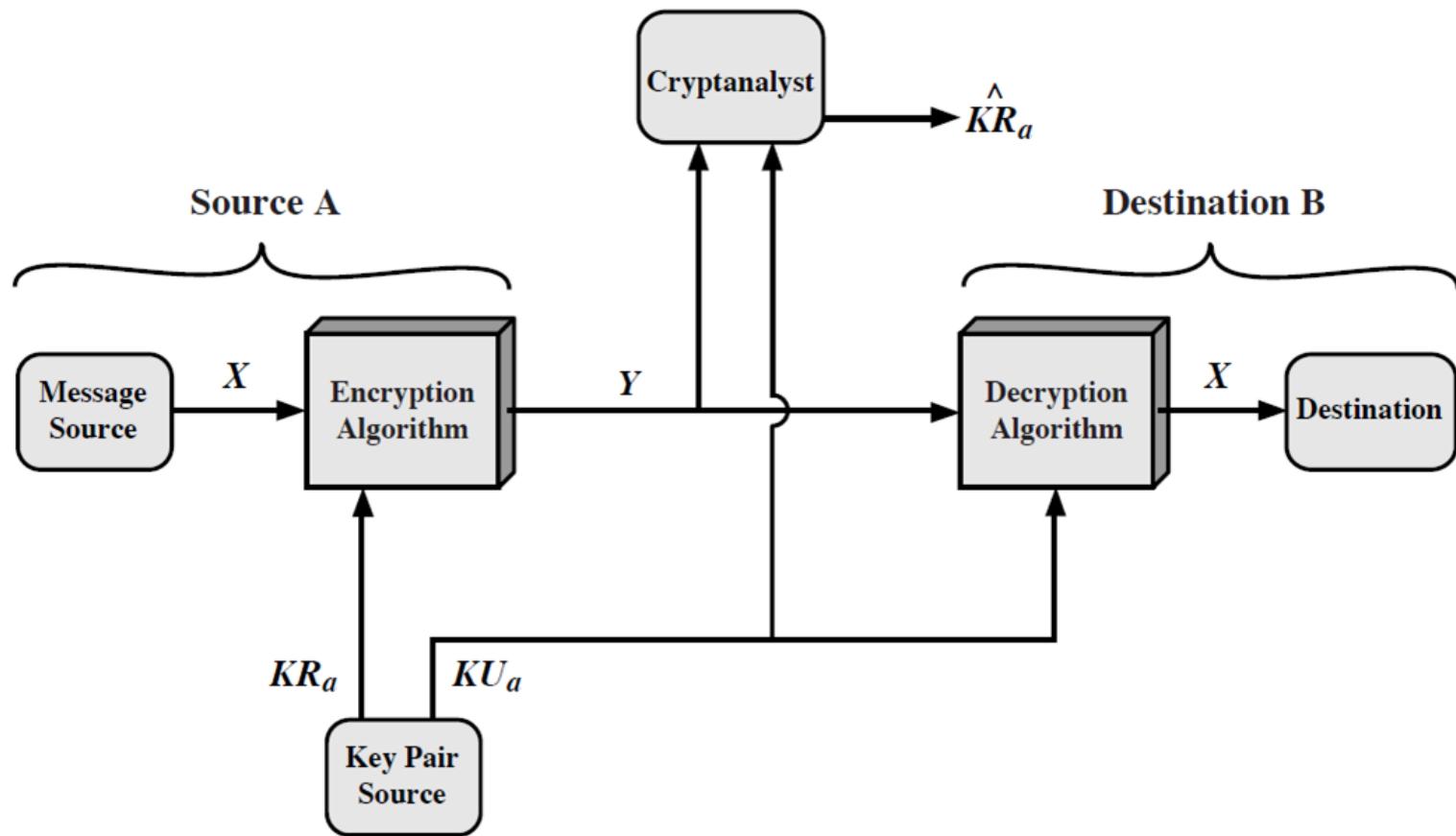
- Public-Key algorithms rely on two keys with the characteristics that it is:
  - computationally infeasible to find decryption key knowing only algorithm & encryption key
  - computationally easy to en/decrypt messages when the relevant (en/decrypt) key is known
  - either of the two related keys can be used for encryption, with the other used for decryption (in some schemes)

# Public-Key: Secrecy



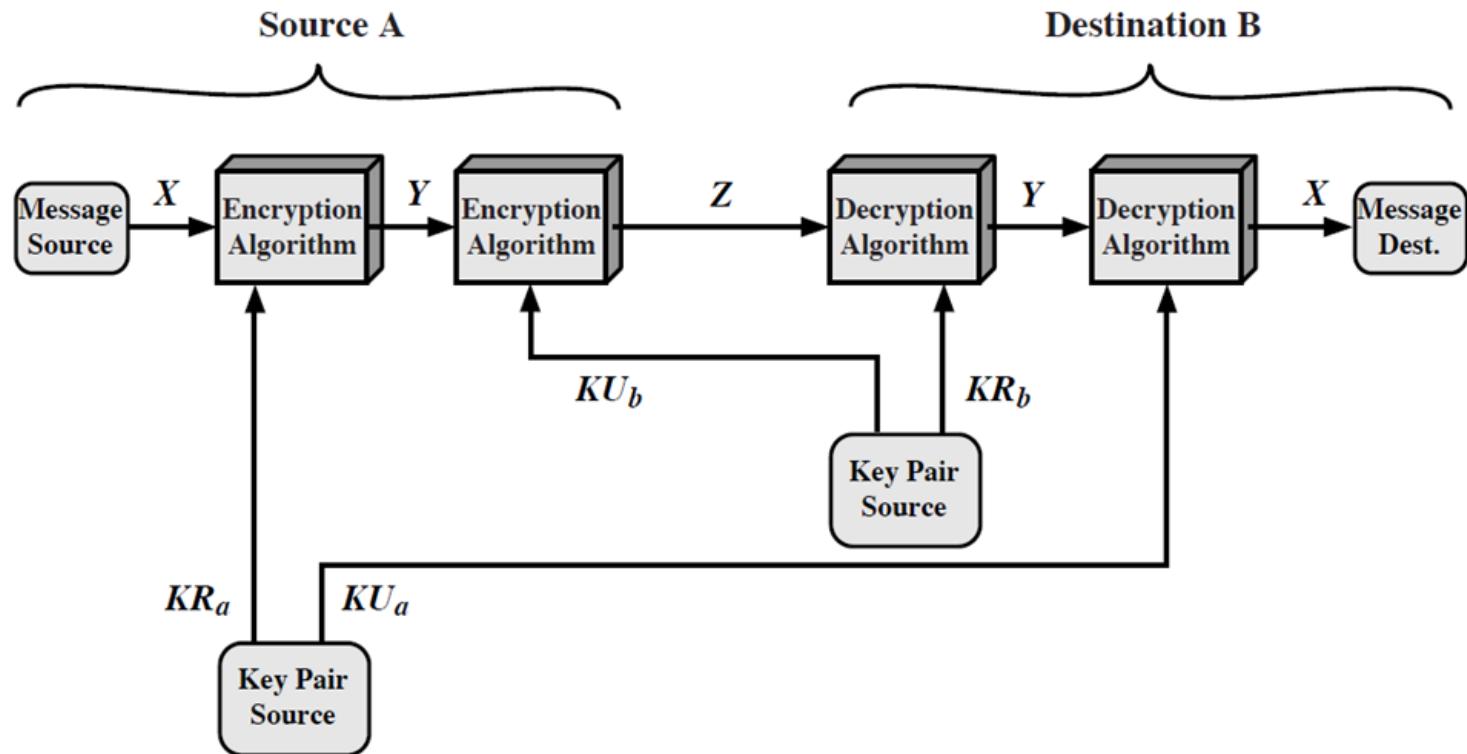
Public-Key Cryptosystem: Secrecy

# Public-Key: Authentication



Public-Key Cryptosystem: Authentication

# Public-Key: Secrecy and Authentication



Public-Key Cryptosystem: Secrecy and Authentication

# Public-Key Applications

- Can classify uses into 3 categories:
  - **encryption/decryption** (provide secrecy)
  - **digital signatures** (provide authentication)
  - **key exchange** (of session keys)
- Some algorithms are suitable for all uses, others are specific to one

# Public-Key Cryptography: Requirements

The main idea behind asymmetric-key cryptography is the concept of the **trapdoor one-way function**.

A **one-way function** is one that maps a domain into a range such that every function value has a unique inverse, with the condition that the calculation of the function is easy, whereas the calculation of the inverse is infeasible

$$Y = f(X) \text{ easy}$$

$$X = f^{-1}(Y) \text{ infeasible}$$

# Trapdoor One-Way Function



"Come in, come in. My door  
is always open for you!"

- A trap-door one-way function is a family of invertible functions  $f_k$ , such that
  - $Y = f_k(X)$  easy, if  $k$  and  $X$  are known
  - $X = f_k^{-1}(Y)$  easy, if  $k$  and  $Y$  are known
  - $X = f_k^{-1}(Y)$  infeasible, if  $Y$  known but  $k$  not known
- A practical public-key scheme depends on a suitable trap-door one-way function

# Security of Public Key Schemes

- Like private key schemes brute force exhaustive search attack is always theoretically possible
- But keys used are too large ( $>512$  bits)
- Security relies on a large enough difference in difficulty between easy (en/decrypt) and hard (cryptanalyse) problems
- More generally the hard problem is known, but is made hard enough to be impractical to break
- Requires the use of very large numbers
- Hence is slow compared to private key schemes

# Public-Key Cryptanalysis

- A public-key encryption scheme is vulnerable to a brute-force attack
  - Countermeasure: use large keys
  - Key size must be small enough for practical encryption and decryption
  - Key sizes that have been proposed result in encryption /decryption speeds that are too slow for general-purpose use
  - Public-key encryption is currently confined to key management and signature applications

# Public-Key Cryptanalysis (cont.)

- Another form of attack is to find some way to compute the **private key given the public key**
  - To date it has not been mathematically proven that this form of attack is infeasible for a particular public-key algorithm
- Finally, there is a **probable-message attack**
  - This attack can be thwarted by appending some random bits to simple messages

## Probable message attack

The public key is known- Encrypt all possible messages- Try to find a match between the ciphertext and one of the above encrypted messages

# RSA Cryptosystem

- Developed in 1977 at MIT by Ron Rivest, Adi Shamir & Len Adleman
- Most widely used general-purpose approach to public-key encryption
- Is a cipher in which the plaintext and ciphertext are integers between 0 and  $n - 1$  for some  $n$ 
  - A typical size for  $n$  is 1024 bits, or 309 decimal digits

# RSA Algorithm

- RSA makes use of an expression with exponentials
- Plaintext is encrypted in blocks with each block having a binary value less than some number  $n$
- Encryption and decryption are of the following form, for some plaintext block  $M$  and ciphertext block  $C$

$$C = M^e \bmod n$$

$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

## RSA Algorithm (cont)

- Both sender and receiver must know the value of  $n$
- The sender knows the value of  $e$ , and only the receiver knows the value of  $d$
- This is a public-key encryption algorithm with a public key of  $PU=\{e,n\}$  and a private key of  $PR=\{d,n\}$

# RSA Algorithm Requirements

- For this algorithm to be satisfactory for public-key encryption, the following requirements must be met:
  1. It is possible to find values of  $e$ ,  $d$ ,  $n$  such that  $M^{ed} \bmod n = M$  for all  $M < n$
  2. It is relatively easy to calculate  $M^e \bmod n$  and  $C^d \bmod n$  for all values of  $M < n$
  3. It is infeasible to determine  $d$  given  $e$  and  $n$

## Key Generation by Alice

Select  $p, q$

$p$  and  $q$  both prime,  $p \neq q$

Calculate  $n = p \times q$

Calcuate  $\phi(n) = (p - 1)(q - 1)$

Select integer  $e$

$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$

Calculate  $d$

$d \equiv e^{-1} \pmod{\phi(n)}$

Public key

$PU = \{e, n\}$

Private key

$PR = \{d, n\}$

## Encryption by Bob with Alice's Public Key

Plaintext:

$M < n$

Ciphertext:

$C = M^e \pmod{n}$

## Decryption by Alice with Alice's Public Key

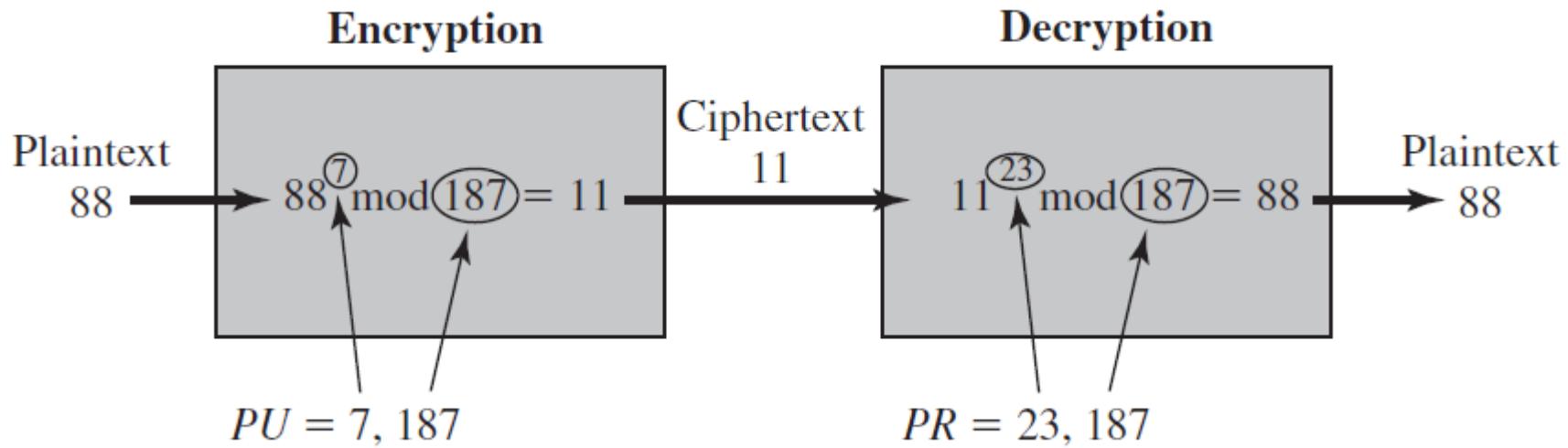
Ciphertext:

$C$

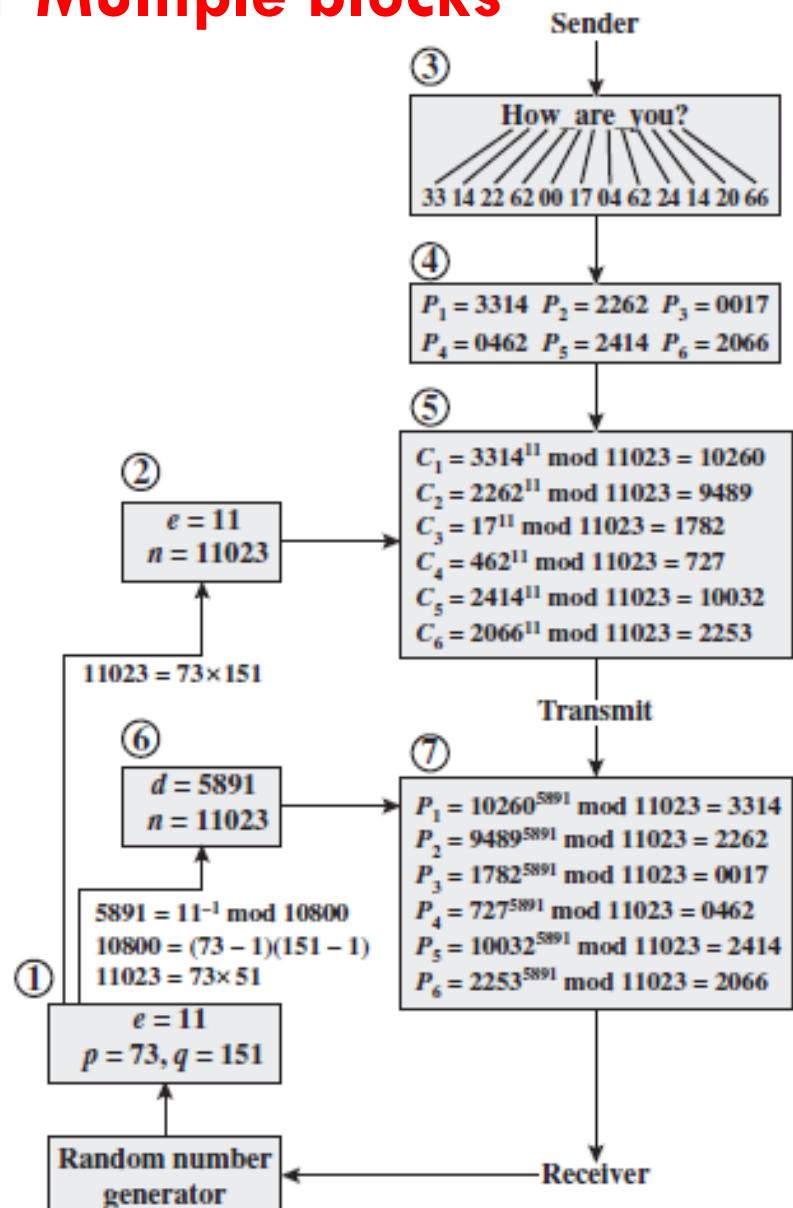
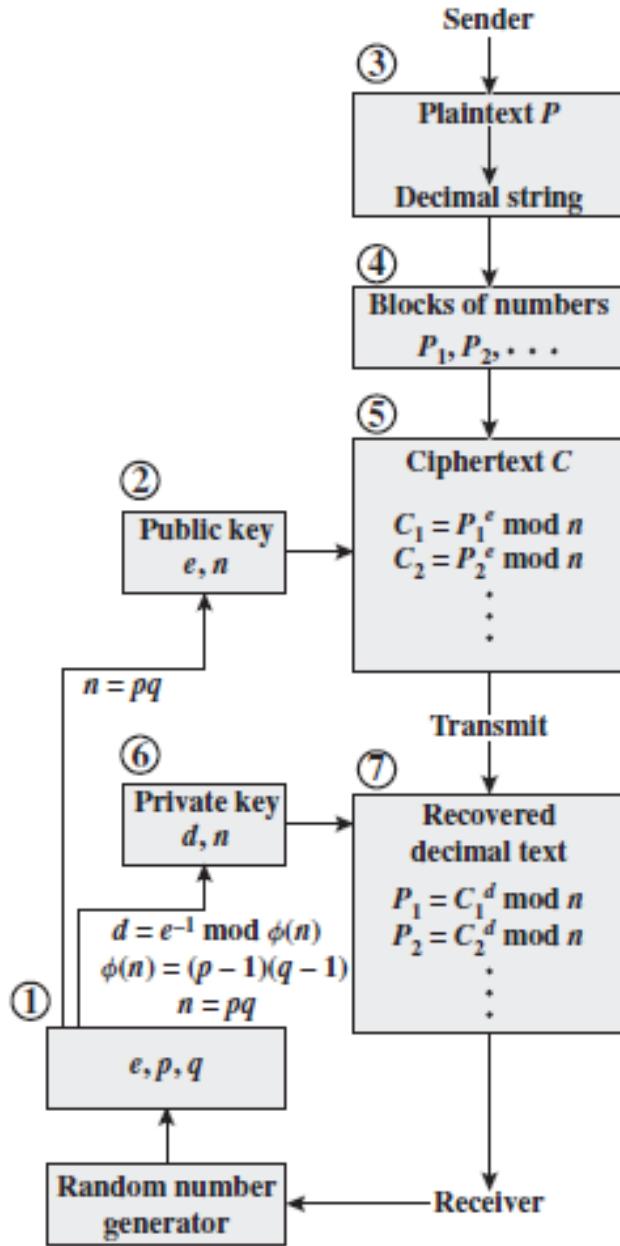
Plaintext:

$M = C^d \pmod{n}$

# RSA: Example



# RSA: Processing of Multiple blocks



# Exponentiation in Modular Arithmetic

- Both encryption and decryption in RSA involve raising an integer to an integer power, mod  $n$
- Can make use of a property of modular arithmetic:  
$$[(a \text{ mod } n) \times (b \text{ mod } n)] \text{ mod } n = (a \times b) \text{ mod } n$$
- With RSA you are dealing with potentially large exponents so efficiency of exponentiation is a consideration

# Exponentiation in Modular Arithmetic

$$a^b \bmod n$$

If we express  $b$  as a binary number  $b_k b_{k-1} \dots b_0$ , then

$$b = \sum_{b_i \neq 0} 2^i$$

Therefore,

$$a^b = a^{\left(\sum_{b_i \neq 0} 2^i\right)} = \prod_{b_i \neq 0} a^{(2^i)}$$

# Algorithm for Computing $a^b \bmod n$

$$a^b \bmod n = \left[ \prod_{b_i \neq 0} a^{(2^i)} \right] \bmod n = \left( \prod_{b_i \neq 0} [a^{(2^i)} \bmod n] \right) \bmod n$$

Note that  
the variable  $c$  is not needed; it  
is included for explanatory  
purposes. The final value  
of  $c$  is the value of the  
exponent.

$$x^{11} = x^{1+2+8} = (x)(x^2)(x^8)$$

```
c ← 0; f ← 1
for i ← k downto 0
    do   c ← 2 × c
          f ← (f × f) mod n
    if   bi = 1
        then c ← c + 1
              f ← (f × a) mod n
return f
```

# Algorithm for Computing $a^b \bmod n$

$i$	9	8	7	6	5	4	3	2	1	0
$b_i$	1	0	0	0	1	1	0	0	0	0
$c$	1	2	4	8	17	35	70	140	280	560
$f$	7	49	157	526	160	241	298	166	67	1

Result of the Fast Modular Exponentiation Algorithm for  $a^b \bmod n$ , where  $a = 7$ ,  $b = 560 = 1000110000$ , and  $n = 561$

# Efficient Operation Using the Public Key

- To speed up the operation of the RSA algorithm using the public key, a specific choice of  $e$  is usually made
- The most common choice is  $65537 (2^{16} + 1)$ 
  - Two other popular choices are  $e=3$  and  $e=17$
  - Each of these choices has only two 1 bits, so the number of multiplications required to perform exponentiation is minimized
  - With a very small public key, such as  $e = 3$ , RSA becomes vulnerable to a simple attack

# Key Generation

- Before the application of the public-key cryptosystem each participant must generate a pair of keys:
  - Determine two prime numbers  $p$  and  $q$
  - Select either  $e$  or  $d$  and calculate the other
- Because the value of  $n = pq$  will be known to any potential adversary, primes must be chosen from a sufficiently large set
  - The method used for finding large primes must be reasonably efficient

# Procedure for Picking a Prime Number

1. Pick an odd integer  $n$  at random
2. Pick an integer  $a < n$  at random
3. Perform the probabilistic primality test with  $a$  as a parameter. If  $n$  fails the test, reject the value  $n$  and go to step 1
4. If  $n$  has passed a sufficient number of tests, accept  $n$ ; otherwise, go to step 2

# Security of RSA

## Brute force

- Involves trying all possible private keys

## Mathematical attacks

- There are several approaches, all equivalent in effort to factoring the product of two primes

**Five possible approaches to attacking RSA are:**

## Hardware fault-based attack

- This involves inducing hardware faults in the processor that is generating digital signatures

## Timing attacks

- These depend on the running time of the decryption algorithm

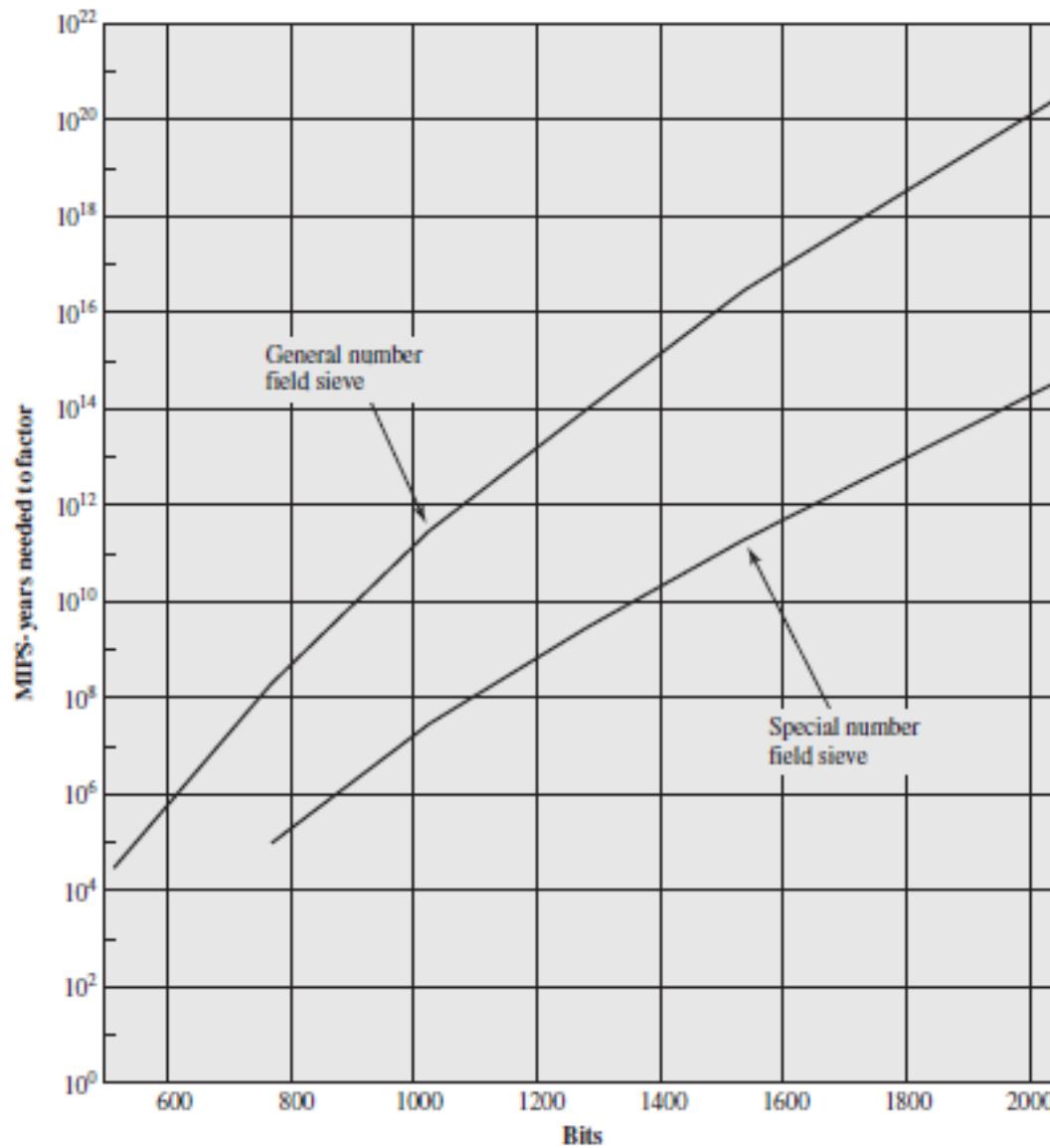
# Factoring Problem

- We can identify three approaches to attacking RSA mathematically:
  - Factor  $n$  into its two prime factors. This enables calculation of  $\phi(n) = (p - 1) \times (q - 1)$ , which in turn enables determination of  $d = e^{-1} \pmod{\phi(n)}$
  - Determine  $\phi(n)$  directly without first determining  $p$  and  $q$ . Again this enables determination of  $d = e^{-1} \pmod{\phi(n)}$
  - Determine  $d$  directly without first determining  $\phi(n)$

# Progress in RSA Factorization

Number of Decimal Digits	Number of Bits	Date Achieved
100	332	April 1991
110	365	April 1992
120	398	June 1993
129	428	April 1994
130	431	April 1996
140	465	February 1999
155	512	August 1999
160	530	April 2003
174	576	December 2003
200	663	May 2005
193	640	November 2005
232	768	December 2009

# MIPS Years needed to Factor



# Timing Attack

- Paul Kocher, a cryptographic consultant, demonstrated that a snooper can determine a private key by keeping track of how long a computer takes to decipher messages
- Are applicable not just to RSA but to other public-key cryptography systems
- Are alarming for two reasons:
  - It comes from a completely unexpected direction
  - It is a ciphertext-only attack

# Counter Measures

## Constant exponentiation time

- Ensure that all exponentiations take the same amount of time before returning a result; this is a simple fix but does degrade performance

## Random delay

- Better performance could be achieved by adding a random delay to the exponentiation algorithm to confuse the timing attack

## Blinding

- Multiply the ciphertext by a random number before performing exponentiation; this process prevents the attacker from knowing what ciphertext bits are being processed inside the computer and therefore prevents the bit-by-bit analysis essential to the timing attack

# Fault-based Attack

- An attack on a processor that is generating RSA digital signatures
  - Induces faults in the signature computation by reducing the power to the processor
  - The faults cause the software to produce invalid signatures which can then be analyzed by the attacker to recover the private key
- The attack algorithm involves inducing single-bit errors and observing the results
- While worthy of consideration, this attack does not appear to be a serious threat to RSA
  - It requires that the attacker have physical access to the target machine and is able to directly control the input power to the processor

# Chosen CipherText Attack

- The adversary chooses a number of ciphertexts and is then given the corresponding plaintexts, decrypted with the target's private key
  - Thus the adversary could select a plaintext, encrypt it with the target's public key, and then be able to get the plaintext back by having it decrypted with the private key
  - The adversary exploits properties of RSA and selects blocks of data that, when processed using the target's private key, yield information needed for cryptanalysis

# Chosen CipherText Attack

- To counter such attacks, RSA Security Inc. recommends modifying the plaintext using a procedure known as *optimal asymmetric encryption padding (OAEP)*

# Attacks on RSA

Attack based on the multiplicative property of RSA. Let Alice creates  $C=P^e \text{ mod } n$  and sends it to Bob. Bob will decrypt an arbitrary ciphertext for Eve other than C. Eve intercepts C and do following steps to get P

- a. Eve chooses a random integer X in  $Z_n^*$ .
- b. Eve calculates  $Y=C \times X^e \text{ mod } n$ .
- c. Eve sends Y to Bob for decryption and get  $Z= Y^d \text{ mod } n$ ;  
This step is an instance of a chosen-ciphertext attack.
- d. Eve can easily find P because

$$\begin{aligned} Z &= Y^d \text{ mod } n = (C \times X^e)^d \text{ mod } n \\ &= (C^d \times X^{ed}) \text{ mod } n = (C^d \times X) \text{ mod } n \\ &= (P \times X) \text{ mod } n \rightarrow P = Z \times X^{-1} \text{ mod } n \end{aligned}$$

# Optimal Asymmetric Encryption Padding (OAEP)

M: Padded message

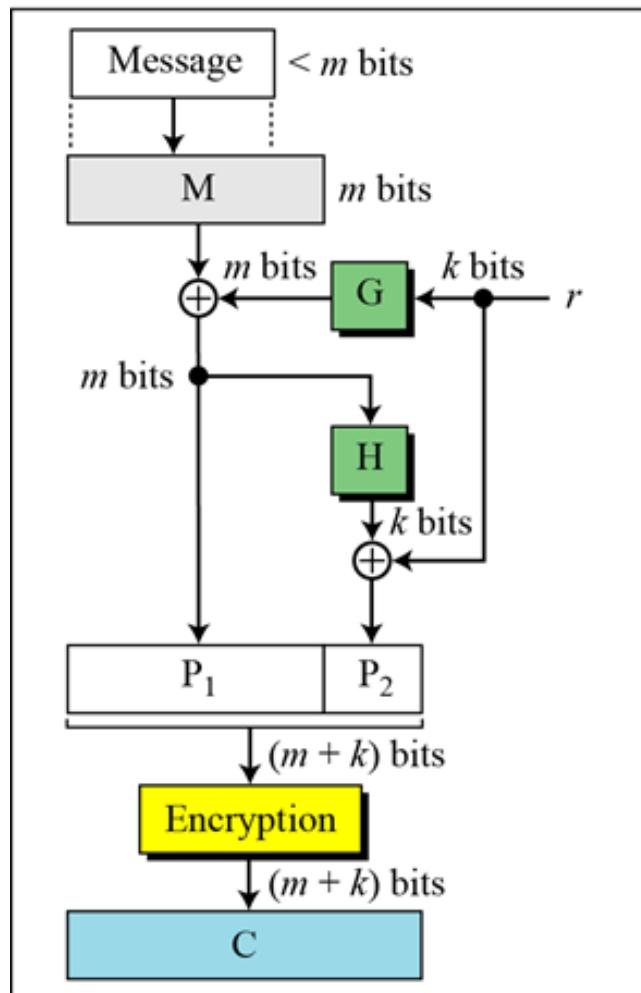
r: One-time random number

P: Plaintext ( $P_1 \parallel P_2$ )

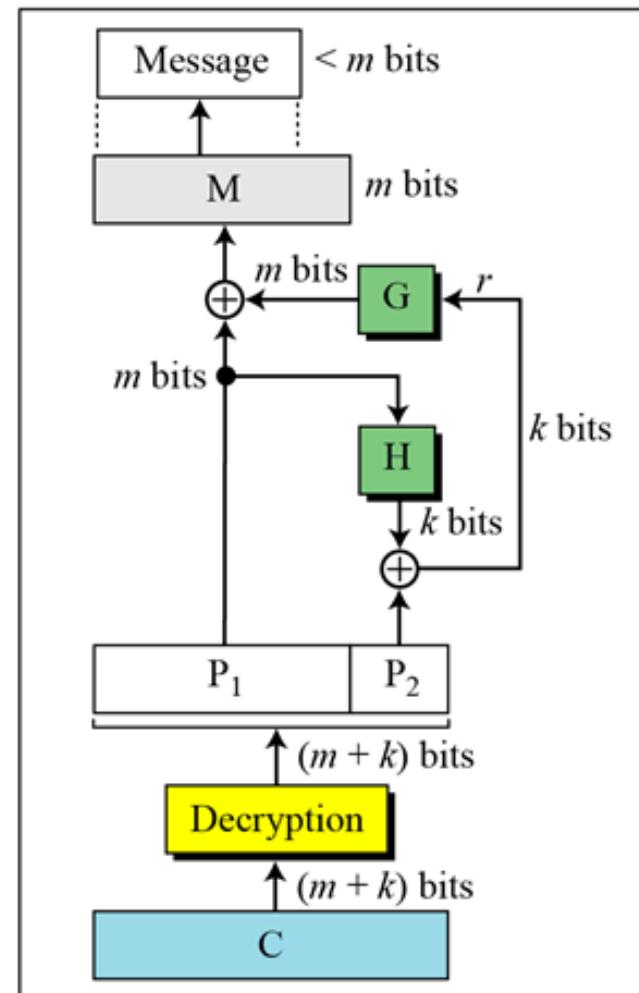
C: Ciphertext

G: Public function ( $k$ -bit to  $m$ -bit)

H: Public function ( $m$ -bit to  $k$ -bit)



Sender



Receiver

## Continued

### Proof of RSA

If  $n = p \times q$ ,  $a < n$ , and  $k$  is an integer, then  $a^{k \times \phi(n)+1} \equiv a \pmod{n}$ .

$$P_1 = C^d \pmod{n} = (P^e \pmod{n})^d \pmod{n} = P^{ed} \pmod{n}$$

$$ed = k\phi(n) + 1 \quad // d \text{ and } e \text{ are inverses modulo } \phi(n)$$

$$P_1 = P^{ed} \pmod{n} \rightarrow P_1 = P^{k\phi(n)+1} \pmod{n}$$

$$P_1 = P^{k\phi(n)+1} \pmod{n} = P \pmod{n} \quad // \text{Euler's theorem (second version)}$$

# Some Trivial Examples

## Example

Bob chooses 7 and 11 as p and q and calculates n = 77. The value of  $f(n) = (7 - 1)(11 - 1)$  or 60. Now he chooses two exponents, e and d, from  $Z_{60}^*$ . If he chooses e to be 13, then d is 37. Note that  $e \times d \bmod 60 = 1$  (they are inverses of each other). Now imagine that Alice wants to send the plaintext 5 to Bob. She uses the public exponent 13 to encrypt 5.

Plaintext: 5

$$C = 5^{13} = 26 \bmod 77$$

Ciphertext: 26

Bob receives the ciphertext 26 and uses the private key 37 to decipher the ciphertext:

Ciphertext: 26

$$P = 26^{37} = 5 \bmod 77$$

Plaintext: 5

# Some Trivial Examples

## Example

Now assume that another person, John, wants to send a message to Bob. John can use the same public key announced by Bob (probably on his website), 13; John's plaintext is 63. John calculates the following:

Plaintext: 63

$$C = 63^{13} \equiv 28 \pmod{77}$$

Ciphertext: 28

Bob receives the ciphertext 28 and uses his private key 37 to decipher the ciphertext:

Ciphertext: 28

$$P = 28^{37} \equiv 63 \pmod{77}$$

Plaintext: 63

## Some Trivial Examples

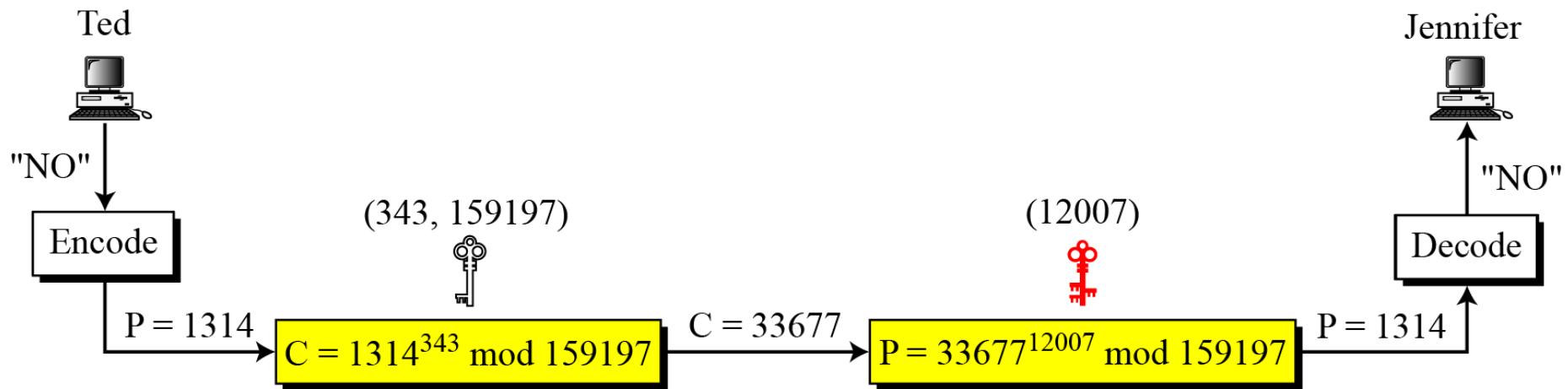
### Example

Jennifer creates a pair of keys for herself. She chooses  $p = 397$  and  $q = 401$ . She calculates  $n = 159197$ . She then calculates  $f(n) = 158400$ . She then chooses  $e = 343$  and  $d = 12007$ . Show how Ted can send a message to Jennifer if he knows  $e$  and  $n$ .

Suppose Ted wants to send the message “NO” to Jennifer. He changes each character to a number (from 00 to 25), with each character coded as two digits. He then concatenates the two coded characters and gets a four-digit number. The plaintext is 1314. Figure 10.7 shows the process.

# Continued

Figure Encryption and decryption in Example



## Attacks on RSA

Attack based on the multiplicative property of RSA. Let Alice creates  $C = P^e \text{ mod } n$  and sends it to Bob. Bob will decrypt an arbitrary ciphertext for Eve other than C. Eve intercepts C and do following steps to get P

- a. Eve chooses a random integer X in  $Z_n^*$ .
- b. Eve calculates  $Y = C \times X^e \text{ mod } n$ .
- c. Eve sends Y to Bob for decryption and get  $Z = Y^d \text{ mod } n$ ; This step is an instance of a chosen-ciphertext attack.
- d. Eve can easily find P because

$$\begin{aligned}Z &= Y^d \text{ mod } n = (C \times X^e)^d \text{ mod } n = (C^d \times X^{ed}) \text{ mod } n = (C^d \times X) \text{ mod } n = (P \times X) \text{ mod } n \\Z &= (P \times X) \text{ mod } n \implies P = Z \times X^{-1} \text{ mod } n\end{aligned}$$

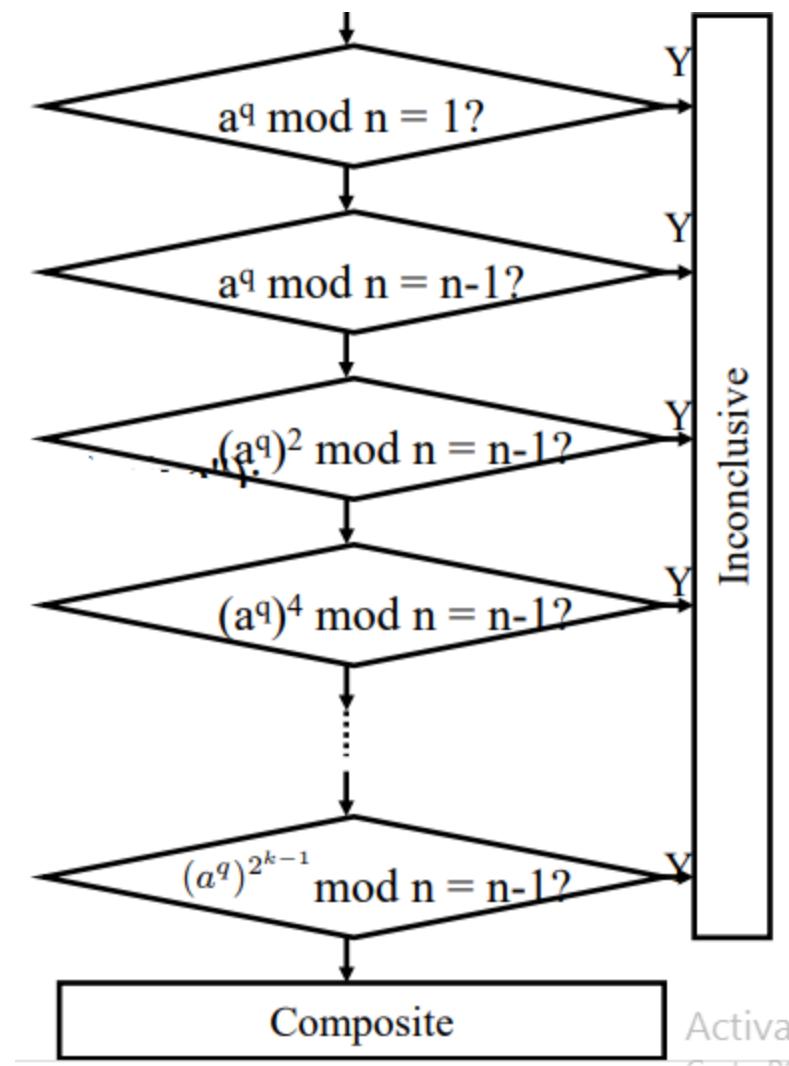
# Miller-Rabin Algorithm for Primality Test

The procedure TEST takes a candidate integer  $n$  as input and returns the result **composite** if  $n$  is definitely **not a prime**, and the result **inconclusive** if  $n$  may or may not be a prime.

TEST ( $n$ )

1. Find integers  $k, q$ , with  $k > 0$ ,  $q$  odd, so that  $(n - 1 = 2^kq)$ ;
2. Select a random integer  $a$ ,  $1 < a < n - 1$ ;
3. **if**  $a^q \text{mod } n = 1$  **then** return("inconclusive") ;
4. **for**  $j = 0$  **to**  $k - 1$  **do**
5.   **if**  $a^{2^j q} \text{mod } n = n - 1$  **then** return("inconclusive") ;
6. **return**("composite") ;

# Miller-Rabin Algorithm for Primality Test



## Miller-Rabin Algorithm for Primality Test

Test 29 for primality

- $29-1 = 28 = 2^2 \times 7 = 2^k q \Rightarrow k=2, q=7$
- Let  $a = 10$ 
  - $10^7 \bmod 29 = 17$
  - $10^{2 \times 7} \bmod 29 = 17^2 \bmod 29 = 28 \Rightarrow$  Inconclusive

Test 221 for primality

- $221-1=220=2^2 \times 55$
- Let  $a=5$ 
  - $5^{55} \bmod 221 = 112$
  - $5^{2 \times 55} \bmod 221 = 112^2 \bmod 221 = 168 \Rightarrow$  Composite

## Chinese Remainder Theorem (cont.)

The Chinese remainder theorem (CRT) is used to solve a set of congruent equations with one variable but different moduli, which are relatively prime, as shown below:

$$x \equiv a_1 \pmod{m_1}$$

$$x \equiv a_2 \pmod{m_2}$$

...

$$x \equiv a_k \pmod{m_k}$$

# Chinese Remainder Theorem (cont.)

## Example

The following is an example of a set of equations with different moduli:

$$x \equiv 2 \pmod{3}$$

$$x \equiv 3 \pmod{5}$$

$$x \equiv 2 \pmod{7}$$

The solution to this set of equations is given in the next section; for the moment, note that the answer to this set of equations is  $x = 23$ . This value satisfies all equations:  $23 \equiv 2 \pmod{3}$ ,  $23 \equiv 3 \pmod{5}$ , and  $23 \equiv 2 \pmod{7}$ .

# Chinese Remainder Theorem (cont.)

## Solution To Chinese Remainder Theorem

1. Find  $M = m_1 \times m_2 \times \dots \times m_k$ . This is the common modulus.
2. Find  $M_1 = M/m_1, M_2 = M/m_2, \dots, M_k = M/m_k$ .
3. Find the multiplicative inverse of  $M_1, M_2, \dots, M_k$  using the corresponding moduli ( $m_1, m_2, \dots, m_k$ ). Call the inverses  $M_1^{-1}, M_2^{-1}, \dots, M_k^{-1}$ .
4. The solution to the simultaneous equations is

$$x = (a_1 \times M_1 \times M_1^{-1} + a_2 \times M_2 \times M_2^{-1} + \dots + a_k \times M_k \times M_k^{-1}) \text{ mod } M$$

# Chinese Remainder Theorem (cont.)

## Example

Find the solution to the simultaneous equations:

$$x \equiv 2 \pmod{3}$$

$$x \equiv 3 \pmod{5}$$

$$x \equiv 2 \pmod{7}$$

## Solution

We follow the four steps.

$$1. M = 3 \times 5 \times 7 = 105$$

$$2. M_1 = 105 / 3 = 35, M_2 = 105 / 5 = 21, M_3 = 105 / 7 = 15$$

$$3. \text{The inverses are } M_1^{-1} = 2, M_2^{-1} = 1, M_3^{-1} = 1$$

$$4. x = (2 \times 35 \times 2 + 3 \times 21 \times 1 + 2 \times 15 \times 1) \pmod{105} = 23 \pmod{105}$$

# Chinese Remainder Theorem (cont.)

## Example

Find an integer that has a remainder of 3 when divided by 7 and 13, but is divisible by 12.

## Solution

This is a CRT problem. We can form three equations and solve them to find the value of  $x$ .

$$x = 3 \bmod 7$$

$$x = 3 \bmod 13$$

$$x = 0 \bmod 12$$

If we follow the four steps, we find  $x = 276$ . We can check that  $276 = 3 \bmod 7$ ,  $276 = 3 \bmod 13$  and 276 is divisible by 12 (the quotient is 23 and the remainder is zero).

## Chinese Remainder Theorem (cont.)

- $\text{mod } M = m_1 m_2 .. m_k$
- Chinese Remainder theorem lets us work in each moduli  $m_i$  separately
- Since computational cost is proportional to size, this is faster

$$A \bmod M = \sum_{i=1}^k (A \bmod m_i) \frac{M}{m_i} \left( \left[ \frac{M}{m_i} \right]^{-1} \bmod m_i \right)$$

## Chinese Remainder Theorem (cont.)

$$A \bmod M = \sum_{i=1}^k (A \bmod m_i) \frac{M}{m_i} \left( \left[ \frac{M}{m_i} \right]^{-1} \bmod m_i \right)$$

$$35^{-1} = x \bmod 3$$

$$35x = 1 \bmod 3 \Rightarrow x = 2$$

$$21x = 1 \bmod 5 \Rightarrow x = 1$$

$$15x = 1 \bmod 7 \Rightarrow x = 1$$

Example:  $452 \bmod 105$

$$= (452 \bmod 3)(105/3)\{(105/3)^{-1} \bmod 3\}$$

$$+ (452 \bmod 5)(105/5)\{(105/5)^{-1} \bmod 5\}$$

$$+ (452 \bmod 7)(105/7)\{(105/7)^{-1} \bmod 7\}$$

$$= 2 \times 35 \times (35^{-1} \bmod 3) + 2 \times 21 \times (21^{-1} \bmod 5) + 4 \times 15 \times (15^{-1} \bmod 7)$$

$$= 2 \times 35 \times 2 + 2 \times 21 \times 1 + 4 \times 15 \times 1$$

$$= (140 + 42 + 60) \bmod 105 = 242 \bmod 105 = 32$$

## Chinese Remainder Theorem (cont.)

$$x = a_1 \bmod m_1$$

$$x = a_2 \bmod m_2$$

$$x = a_k \bmod m_k$$

where  $m_1, m_2, \dots, m_k$  are relatively prime is found as follows:

$$M = m_1 m_2 \dots m_k \text{ then}$$

$$x = \sum_{i=1}^k a_i \frac{M}{m_i} \left( \left[ \frac{M}{m_i} \right]^{-1} \bmod m_i \right)$$

## Chinese Remainder Theorem (cont.)

$$x \equiv 1 \pmod{7}$$

$$x \equiv 2 \pmod{8}$$

$$x \equiv 3 \pmod{9}$$

$$N = 7 \times 8 \times 9 = 504$$

$$\begin{aligned}x &= \left( 1 \times \frac{504}{7} \times \left[ \frac{504}{7} \right]_7^{-1} + 2 \times \frac{504}{8} \times \left[ \frac{504}{8} \right]_8^{-1}\right. \\&\quad \left. + 3 \times \frac{504}{9} \times \left[ \frac{504}{9} \right]_9^{-1} \right) \pmod{7 \times 8 \times 9} \\&= (1 \times 72 \times (72^{-1} \pmod{7}) + 2 \times 63 \times (63^{-1} \pmod{8}) \\&\quad + 3 \times 56 \times (56^{-1} \pmod{9})) \pmod{504} \\&= (1 \times 72 \times 4 + 2 \times 63 \times 7 + 3 \times 56 \times 5) \pmod{504} \\&= (288 + 882 + 840) \pmod{504} \\&= 2010 \pmod{504} \\&= 498\end{aligned}$$

# Fermat's Little Theorem

Given a prime number p:

$$a^{p-1} \equiv 1 \pmod{p}$$

For all integers  $a \neq p$

Or  $a^p \equiv a \pmod{p}$

Example:

- $1^4 \pmod{5} = 1$
- $2^4 \pmod{5} = 1$
- $3^4 \pmod{5} = 1$
- $4^4 \pmod{5} = 1$

# Euler's Theorem

A generalisation of Fermat's Theorem

$$a^{\phi(n)} = 1 \pmod{n}$$

➤ for any  $a, n$  where  $\gcd(a,n)=1$

Example:

$$a=3; n=10; \phi(10)=4;$$

$$\text{hence } 3^4 = 81 = 1 \pmod{10}$$

$$a=2; n=11; \phi(11)=10;$$

$$\text{hence } 2^{10} = 1024 = 1 \pmod{11}$$

Also have:  $a^{\phi(n)+1} = a \pmod{n}$

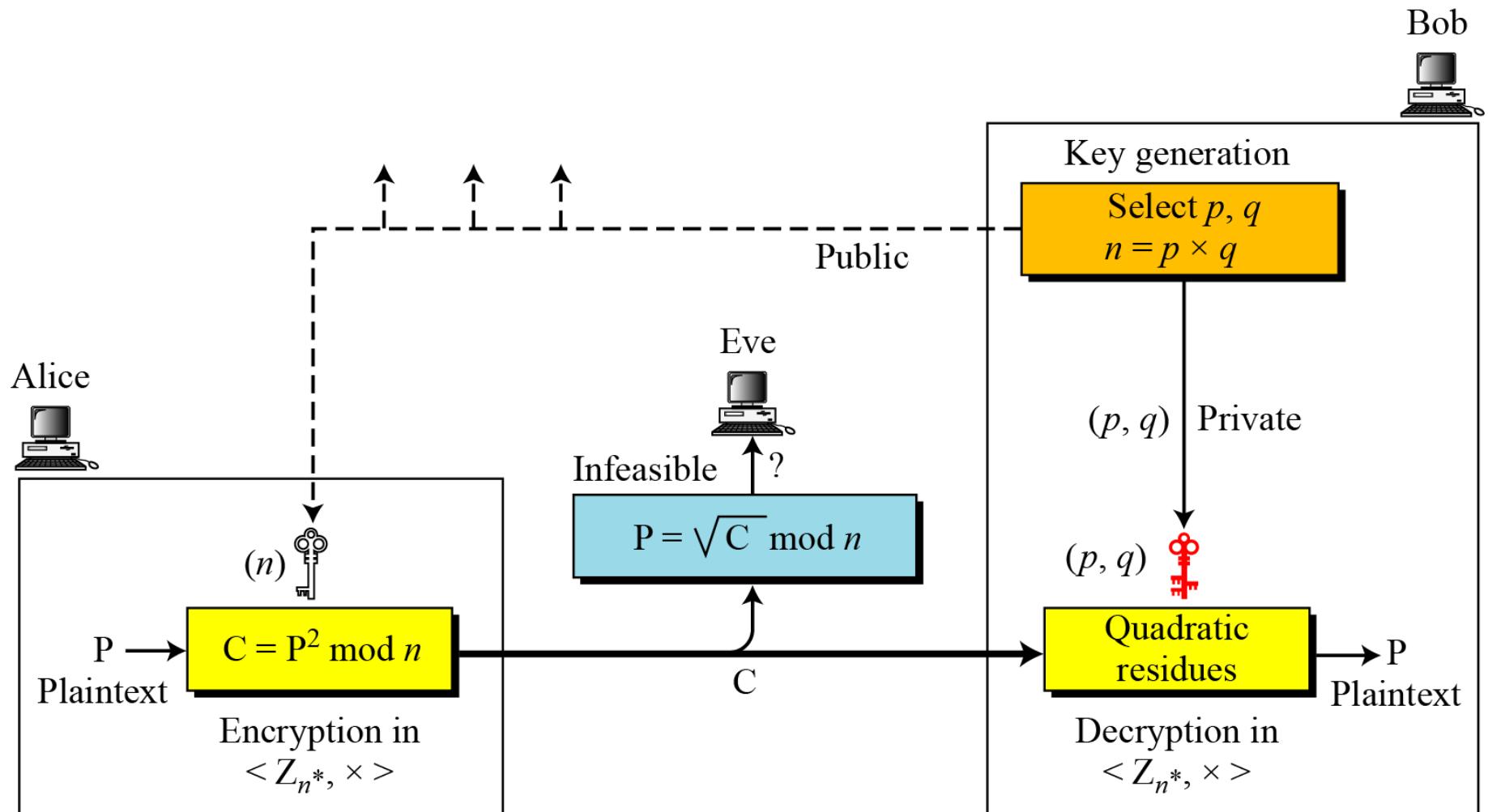
# RABIN CRYPTOSYSTEM

The Rabin cryptosystem can be thought of as an RSA cryptosystem in which the value of e and d are fixed. The encryption is  $C \equiv P^2 \pmod{n}$  and the decryption is  $P \equiv C^{1/2} \pmod{n}$ .

## Topics :

- 1      **Procedure**
- 2      **Security of the Rabin System**

# Continued



# Continued

**Algorithm** : *Key generation for Rabin cryptosystem*

## Rabin\_Key\_Generation

```
{  
    Choose two large primes  $p$  and  $q$  in the form  $4k + 3$  and  $p \neq q$ .  
     $n \leftarrow p \times q$   
    Public_key  $\leftarrow n$                                 // To be announced publicly  
    Private_key  $\leftarrow (q, n)$                           // To be kept secret  
    return Public_key and Private_key  
}
```

# *Continued*

## *Encryption*

**Algorithm** *Encryption in Rabin cryptosystem*

```
Rabin_Encryption (n, P)          // n is the public key; P is the ciphertext from  $\mathbf{Z}_n^*$ 
{
    C ← P2 mod n           // C is the ciphertext
    return C
}
```

# *Continued*

## *Decryption*

**Algorithm** *Decryption in Rabin cryptosystem*

```
Rabin_Decryption (p, q, C)           // C is the ciphertext; p and q are private keys
{
     $a_1 \leftarrow +(\text{C}^{(p+1)/4}) \text{ mod } p$ 
     $a_2 \leftarrow -(\text{C}^{(p+1)/4}) \text{ mod } p$ 
     $b_1 \leftarrow +(\text{C}^{(q+1)/4}) \text{ mod } q$ 
     $b_2 \leftarrow -(\text{C}^{(q+1)/4}) \text{ mod } q$ 
    // The algorithm for the Chinese remainder algorithm is called four times.
     $P_1 \leftarrow \text{Chinese\_Remainder}(a_1, b_1, p, q)$ 
     $P_2 \leftarrow \text{Chinese\_Remainder}(a_1, b_2, p, q)$ 
     $P_3 \leftarrow \text{Chinese\_Remainder}(a_2, b_1, p, q)$ 
     $P_4 \leftarrow \text{Chinese\_Remainder}(a_2, b_2, p, q)$ 
    return  $P_1, P_2, P_3$ , and  $P_4$ 
}
```

**The Rabin cryptosystem is not deterministic:  
Decryption creates four plaintexts.**

# *Continued*

## Example

Here is a very trivial example to show the idea.

1. Bob selects  $p = 23$  and  $q = 7$ . Note that both are congruent to 3 mod 4.
2. Bob calculates  $n = p \times q = 161$ .
3. Bob announces  $n$  publicly; he keeps  $p$  and  $q$  private.
4. Alice wants to send the plaintext  $P = 24$ . Note that 161 and 24 are relatively prime; 24 is in  $\mathbb{Z}_{161}^*$ . She calculates  $C = 24^2 = 93$  mod 161, and sends the ciphertext 93 to Bob.

# *Continued*

## Example

5. Bob receives 93 and calculates four values:

$$a_1 = +(93^{(23+1)/4}) \bmod 23 = 1 \bmod 23$$

$$a_2 = -(93^{(23+1)/4}) \bmod 23 = 22 \bmod 23$$

$$b_1 = +(93^{(7+1)/4}) \bmod 7 = 4 \bmod 7$$

$$b_2 = -(93^{(7+1)/4}) \bmod 7 = 3 \bmod 7$$

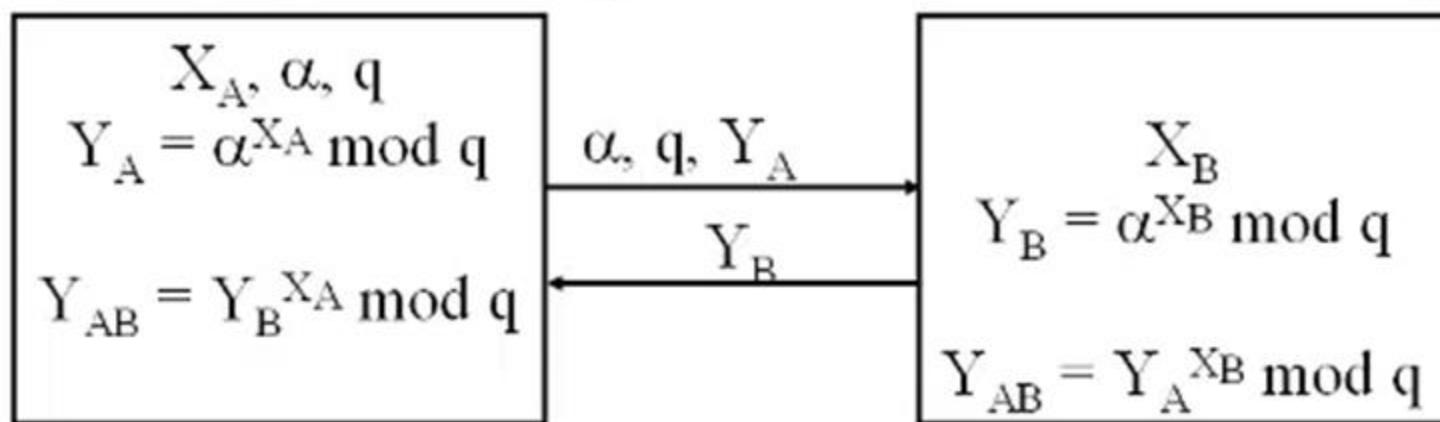
6. Bob takes four possible answers,  $(a_1, b_1)$ ,  $(a_1, b_2)$ ,  $(a_2, b_1)$ , and  $(a_2, b_2)$ , and uses the Chinese remainder theorem to find four possible plaintexts: 116, 24, 137, and 45. Note that only the second answer is Alice's plaintext.

# Diffie-Hellman Key Exchange

- ▶ First public-key type scheme proposed
- ▶ By Diffie & Hellman in 1976 along with the exposition of public key concepts
  - note: now know that Williamson (UK CESG) secretly proposed the concept in 1970
- ▶ Is a practical method for public exchange of a secret key
- ▶ Used in a number of commercial products

## Diffie-Hellman Key Exchange

- Allows two parties to agree on a secret key using a public channel
- A selects  $q = \text{large prime}$ , and  $\alpha = \text{a primitive root of } q$
- A selects a random  $\# X_A$ , B selects another random  $\# X_B$



- Eavesdropper can see  $Y_A, \alpha, q$  but cannot compute  $X_A$
- Computing  $X_A$  requires discrete logarithm - a difficult problem

# Diffie-Hellman Key Exchange

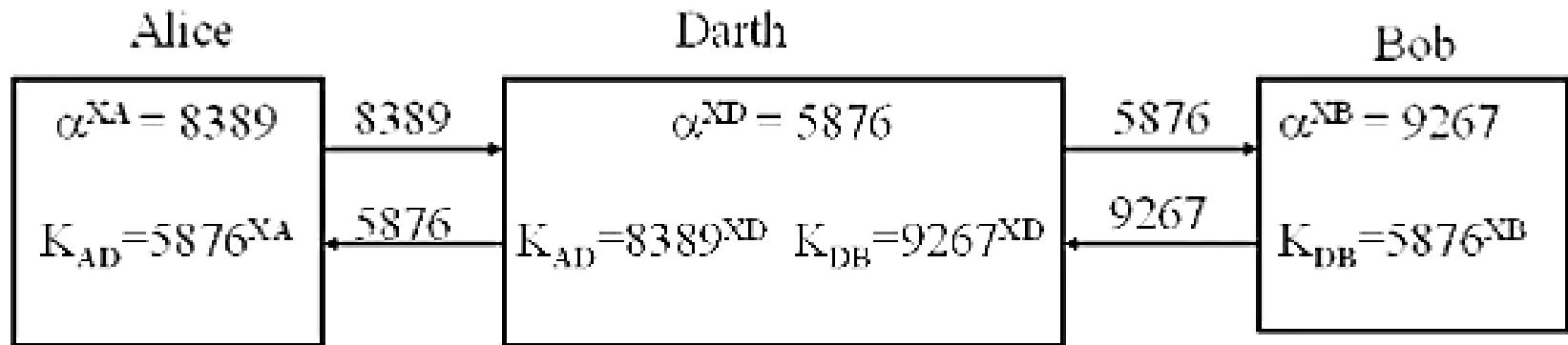
$a^i \bmod 7$	$a^1$	$a^2$	$a^3$	$a^4$	$a^5$	$a^6$	$\dots$
1	1	1	1	1	1	1	$\times$
2	4	1	2	4	1	4	$\times$
3	2	6	4	5	1	2	$\checkmark$
4	2	1	4	2	1	2	$\times$
5	4	6	2	3	1	4	$\checkmark$
6	1	6	1	6	1	6	$\times$

## Diffie-Hellman Key Exchange (cont.)

- Example:  $\alpha=5$ ,  $q=19$ 
  - A selects 6 and sends  $5^6 \bmod 19 = 7$
  - B selects 7 and sends  $5^7 \bmod 19 = 16$
  - A computes  $K = 16^6 \bmod 19 = 7$
  - B computes  $K = 7^7 \bmod 19 = 7$
- Preferably  $(q-1)/2$  should also be a prime.
- Such primes are called safe prime.

# Man-in-the-Middle Attack

- Diffie-Hellman does not provide authentication



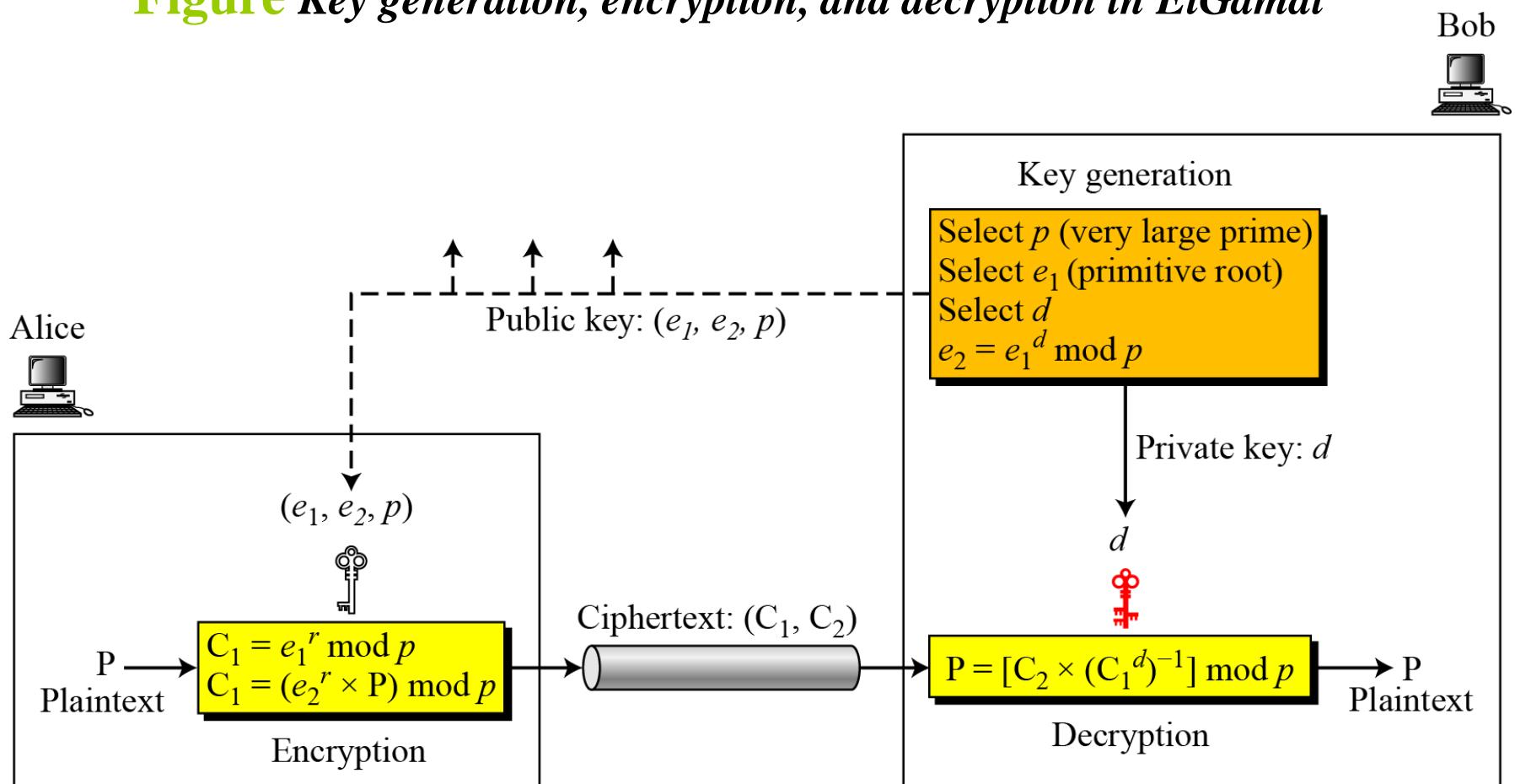
- X can then intercept, decrypt, re-encrypt, forward all messages between Alice & Bob
- You can use RSA authentication and other alternatives

# ELGAMAL CRYPTOSYSTEM

*Besides RSA and Rabin, another public-key cryptosystem is ElGamal. ElGamal is based on the discrete logarithm problem.*

# Procedure

**Figure Key generation, encryption, and decryption in ElGamal**



# *Continued*

## Example

*Here is a trivial example. Bob chooses  $p = 11$  and  $e_1 = 2$ . and  $d = 3$   $e_2 = e_1^d = 8$ . So the public keys are  $(2, 8, 11)$  and the private key is 3. Alice chooses  $r = 4$  and calculates  $C_1$  and  $C_2$  for the plaintext 7.*

**Plaintext:** 7

$$C_1 = e_1^r \bmod 11 = 16 \bmod 11 = 5 \bmod 11$$

$$C_2 = (P \times e_2^r) \bmod 11 = (7 \times 4096) \bmod 11 = 6 \bmod 11$$

**Ciphertext:** (5, 6)

*Bob receives the ciphertexts (5 and 6) and calculates the plaintext*

$$[C_2 \times (C_1^d)^{-1}] \bmod 11 = 6 \times (5^3)^{-1} \bmod 11 = 6 \times 3 \bmod 11 = 7 \bmod 11$$

**Plaintext:** 7

# **Digital Signature**

# COMPARISON

*Let us begin by looking at the differences between conventional signatures and digital signatures.*

## *Topics discussed in this section:*

Inclusion

Verification Method

Relationship

Duplicity

# Inclusion

A conventional signature is included in the document; it is part of the document. But when we sign a document digitally, we send **the signature as a separate document**.

# Verification Method

For a conventional signature, when the recipient receives a document, he/she compares the signature on the document **with the signature on file**. For a digital signature, the recipient receives the message and the signature. The recipient needs to **apply a verification technique to** the combination of the message and the signature to verify the authenticity.

## Relationship

For a conventional signature, there is normally a one-to-many relationship between a signature and documents (same signature to many documents). For a digital signature, there is a one-to-one relationship between a signature and a message.

# Duplicity

In conventional signature, a copy of the signed document can be distinguished from the original one on file. In digital signature, there is no such distinction unless there is a factor of time on the document.

# PROCESS

Figure in the next slide shows the digital signature process. The sender uses a signing algorithm to sign the message. The message and the signature are sent to the receiver. The receiver receives the message and the signature and applies the verifying algorithm to the combination. If the result is true, the message is accepted; otherwise, it is rejected.

## Topics

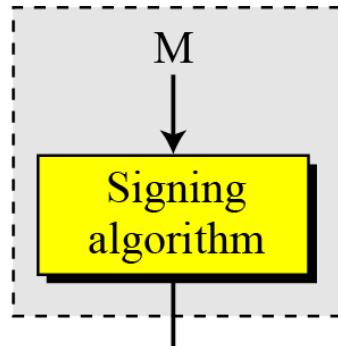
Need for Keys

Signing the Digest

# Continued

**Figure** Digital signature process

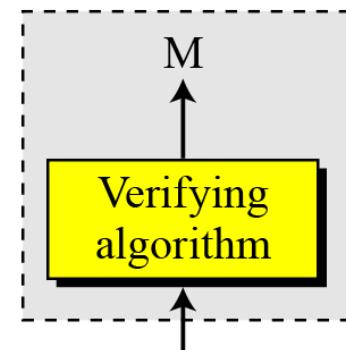
Alice



M: Message  
S: Signature

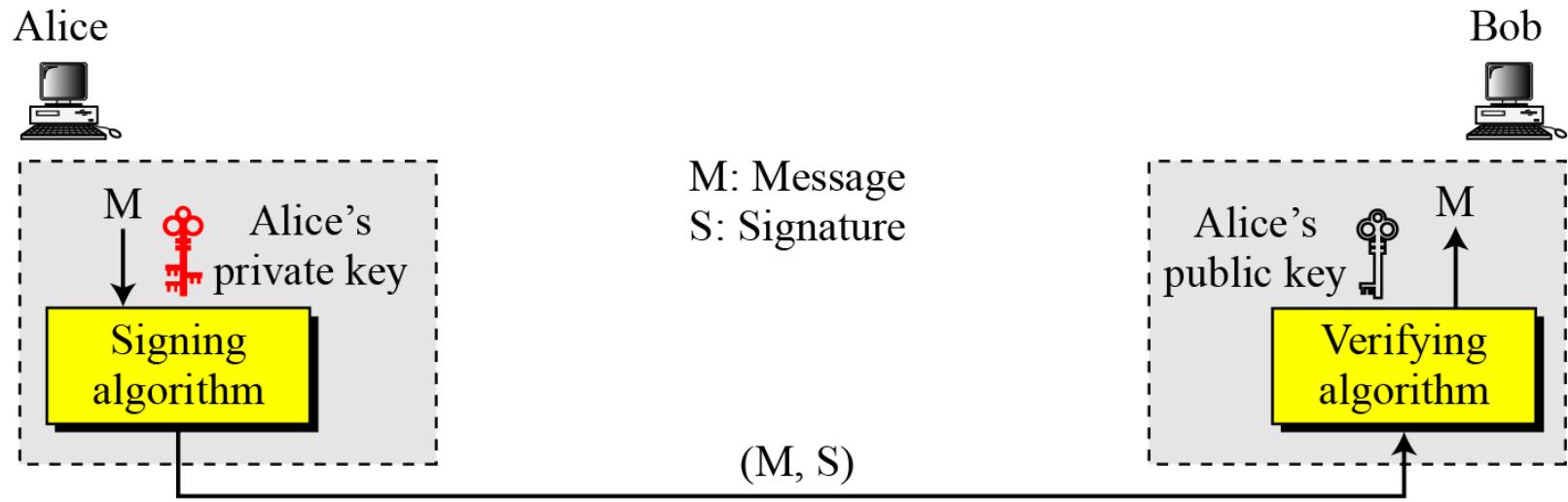
(M, S)

Bob



# Need for Keys

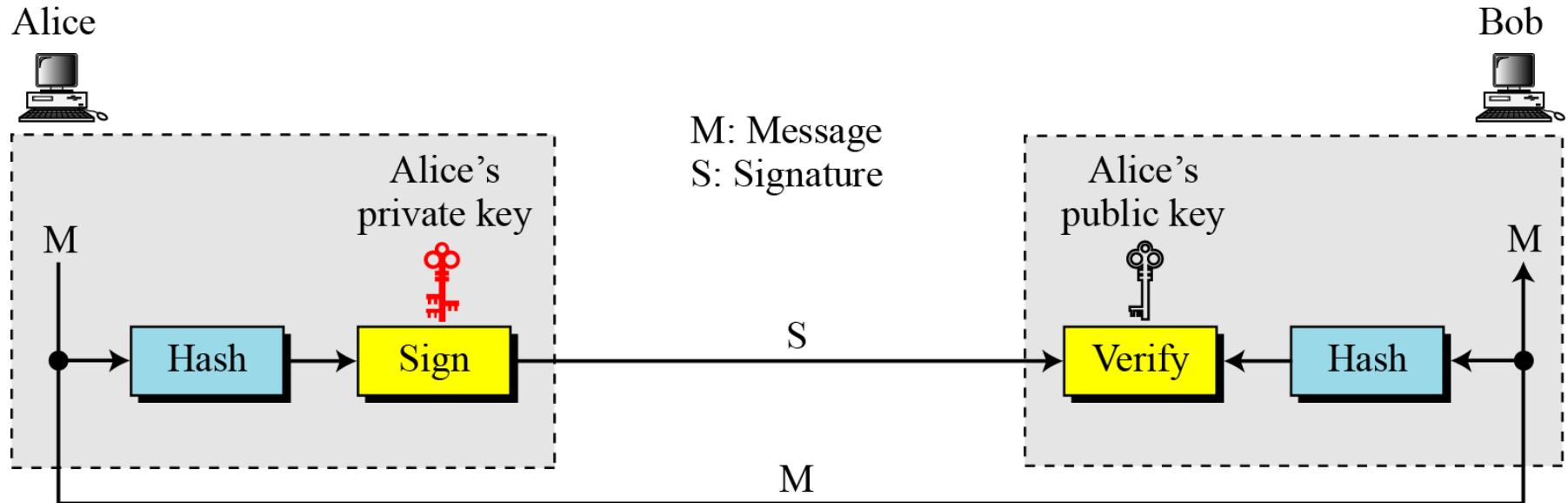
**Figure** Adding key to the digital signature process



A digital signature needs a public-key system.  
The signer signs with her private key; the verifier  
verifies with the signer's public key.

# Signing the Digest

Figure Signing the digest



# Attacks

- In the order of Increasing severity.
  - C=Attacker, A=Victim
1. **Key-only attack:** C only knows A's public key
  2. **Known message attack:** C has a set of messages, signatures
  3. **Generic chosen message attack:** C obtains A's signatures on messages selected without knowledge of A's public key
  4. **Directed chosen message attack:** C obtains A's signatures on messages selected after knowing A's public key
  5. **Adaptive chosen message attack:** C may request signatures on messages depending upon previous message-signature pairs

# Forgeries

1. **Total break:** C knows A's private key
2. **Universal forgery:** C can generate A's signatures on any message
3. **Selective forgery:** C can generate A's signature for a particular message chosen by C
4. **Existential forgery:** C can generate A's signature for a message not chosen by C

# Digital Signature Requirements

- Must depend on the message signed
- Must use information unique to sender
  - To prevent both forgery and denial
- Must be relatively easy to produce
- Must be relatively easy to recognize & verify
  - Directed ⇒ Recipient can verify
  - Arbitrated ⇒ Anyone can verify
- Be computationally infeasible to forge
  - With new message for existing digital signature
  - With fraudulent digital signature for given message
- Be able to retain a copy of the signature in storage

# SERVICES

We discussed several security services *including message confidentiality, message authentication, message integrity, and nonrepudiation*. A digital signature can directly provide the last three; for message confidentiality we still need encryption/decryption.

## Topics

Message Authentication

Message Integrity

Nonrepudiation

Confidentiality

## *Message Authentication*

*A secure digital signature scheme, like a secure conventional signature can provide message authentication.*

A digital signature provides message authentication.

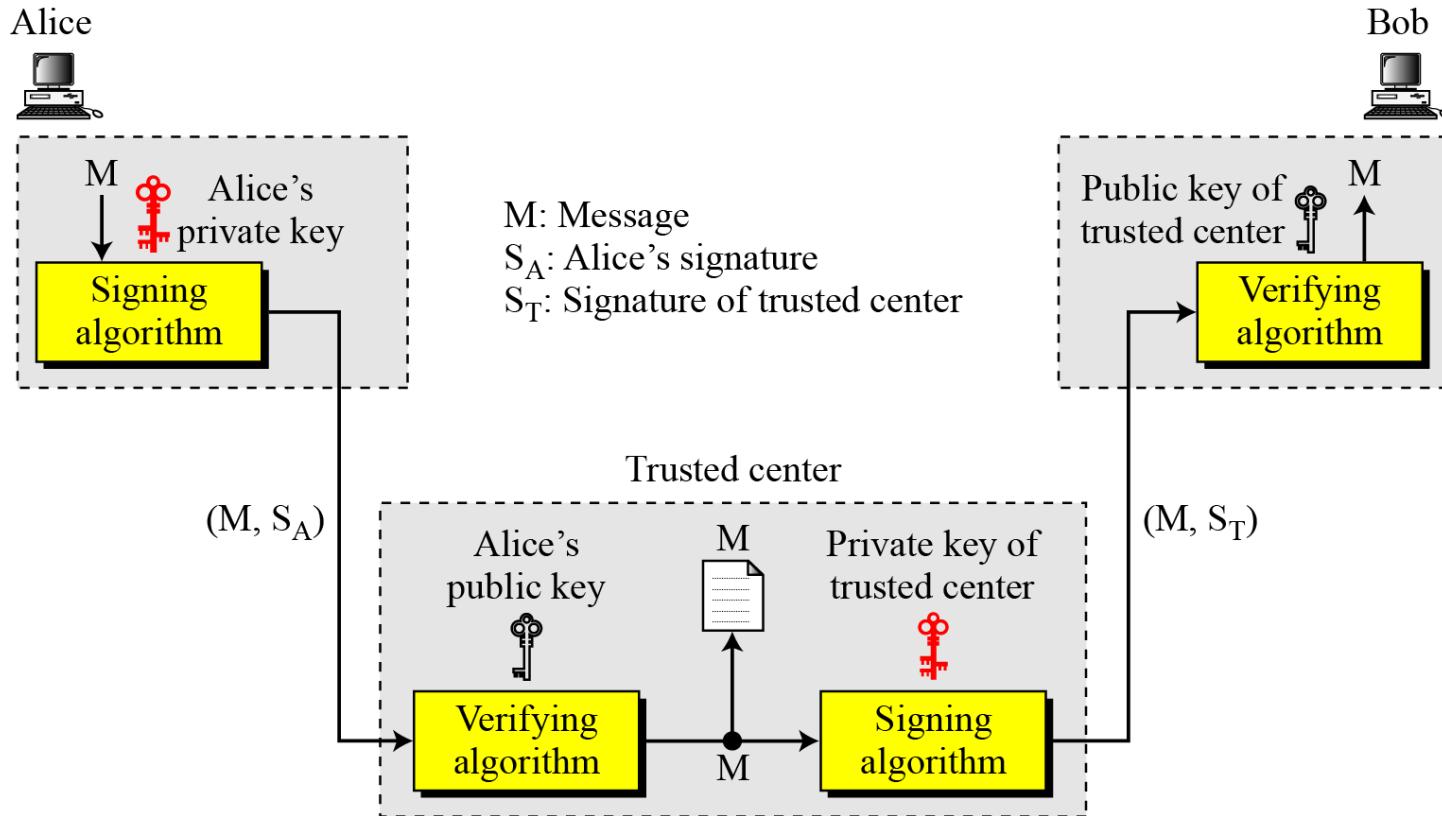
## *Message Integrity*

*The integrity of the message is preserved even if we sign the whole message because we cannot get the same signature if the message is changed.*

A digital signature provides message integrity.

# Nonrepudiation

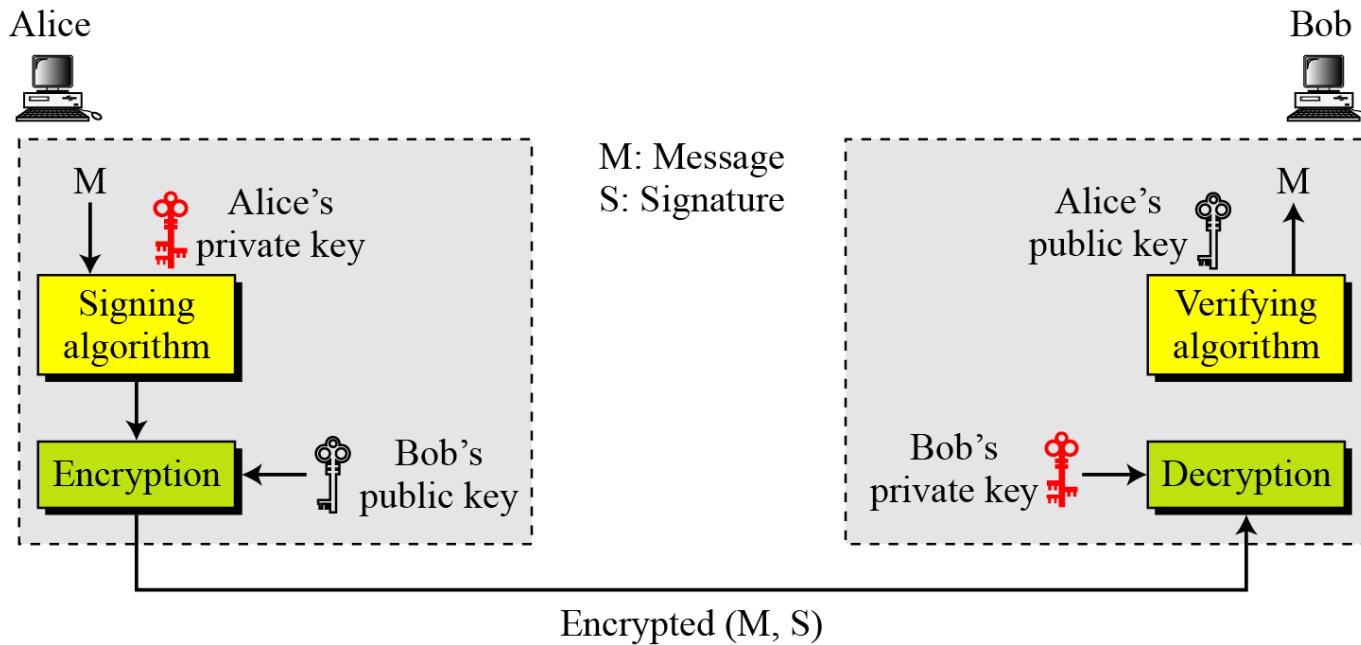
Figure Using a trusted center for nonrepudiation



Nonrepudiation can be provided using a trusted party.

# Confidentiality

**Figure** Adding confidentiality to a digital signature scheme



A digital signature does not provide privacy.  
If there is a need for privacy, another layer of encryption/decryption must be applied.

# DIGITAL SIGNATURE SCHEMES

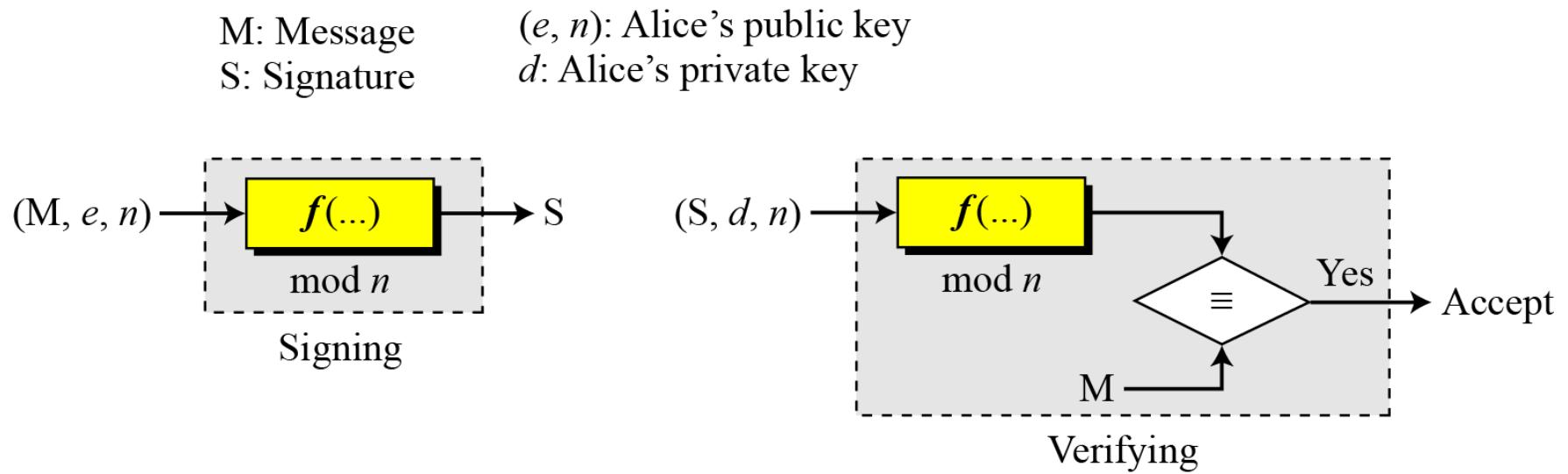
Several digital signature schemes have evolved during the last few decades. Some of them have been implemented.

## Topics

1. RSA Digital Signature Scheme
2. ElGamal Digital Signature Scheme
3. Schnorr Digital Signature Scheme
4. Digital Signature Standard (DSS)

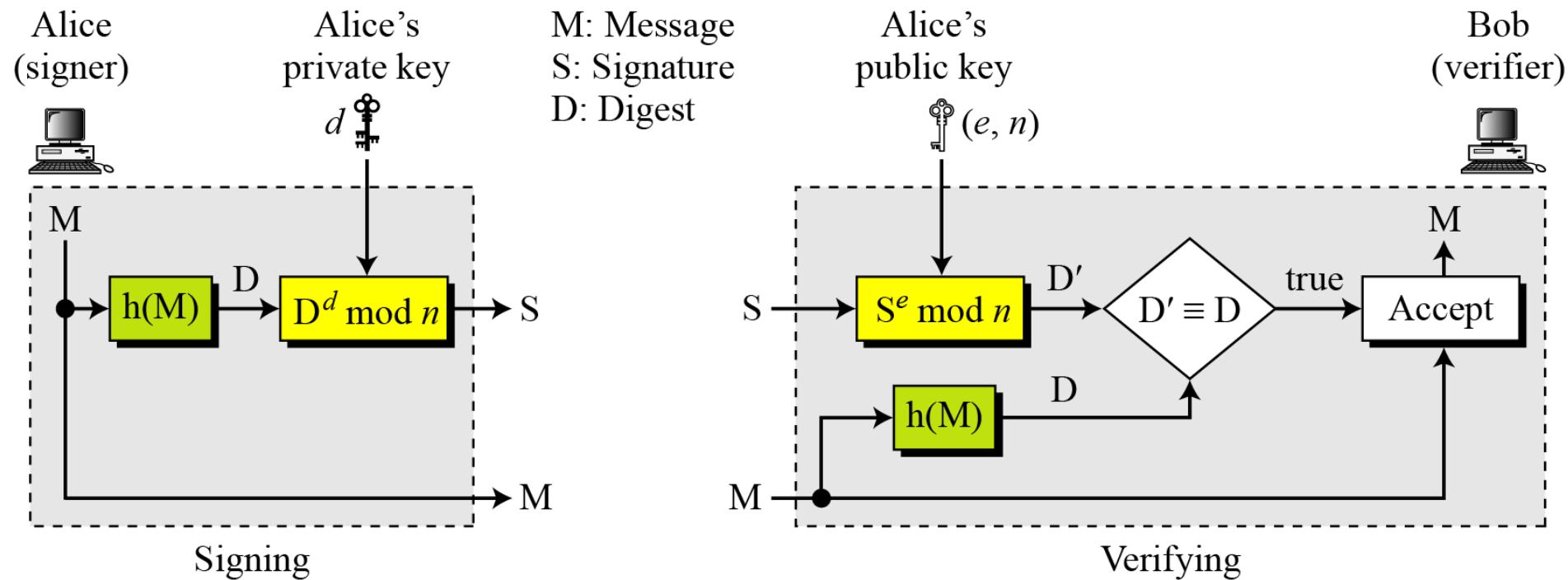
# RSA Digital Signature Scheme

Figure General idea behind the RSA digital signature scheme



# Continued

## Signing and with Message digest



# Continued

## ElGamal signature Scheme

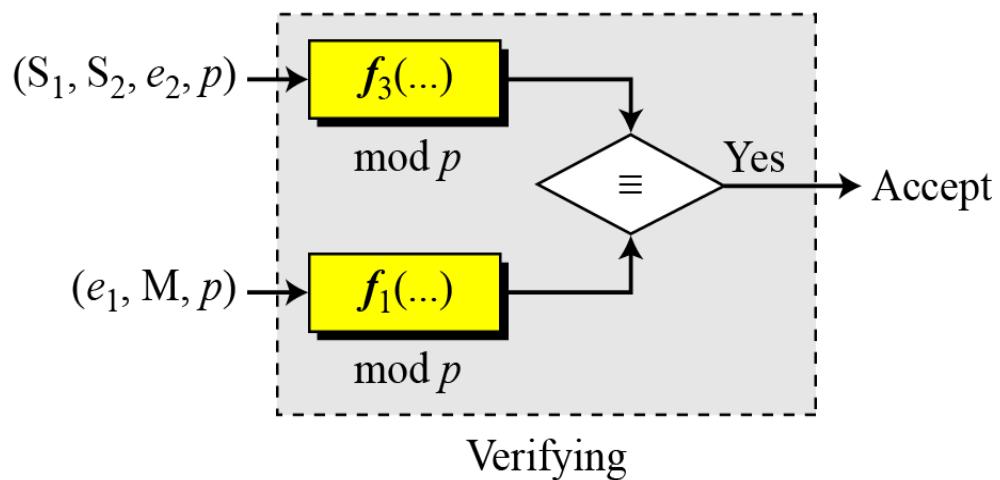
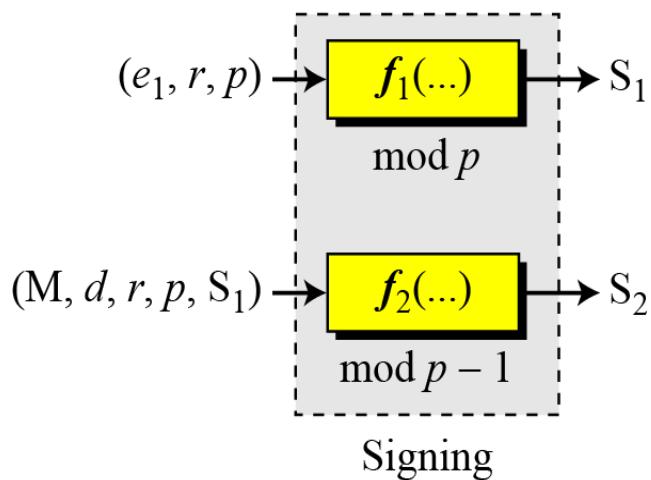
$S_1, S_2$ : Signatures

M: Message

$(e_1, e_2, p)$ : Alice's public key

$d$ : Alice's private key

$r$ : Random secret



# Continued

## ElGamal signature Scheme

M: Message

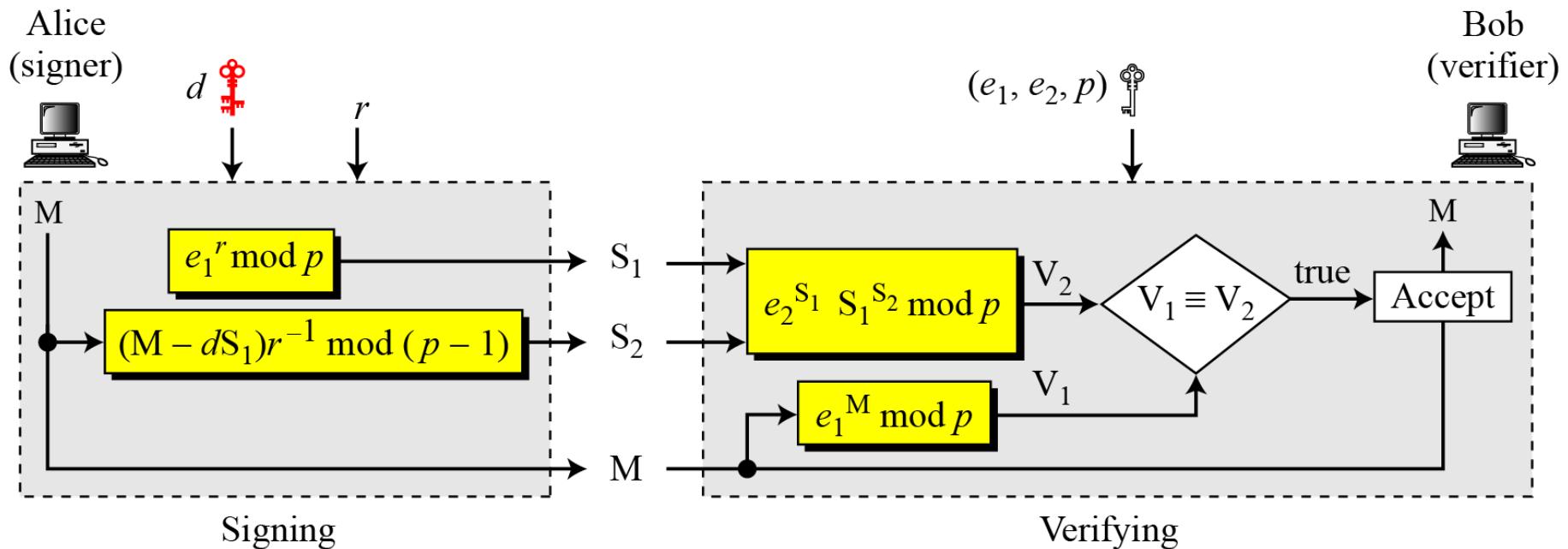
$S_1, S_2$ : Signatures

$V_1, V_2$ : Verifications

$r$ : Random secret

$d$ : Alice's private key

$(e_1, e_2, p)$ : Alice's public key



# Schnorr Digital Signature Scheme

Figure General idea behind the Schnorr digital signature scheme.  
Based on ElGamal: reduce signature size

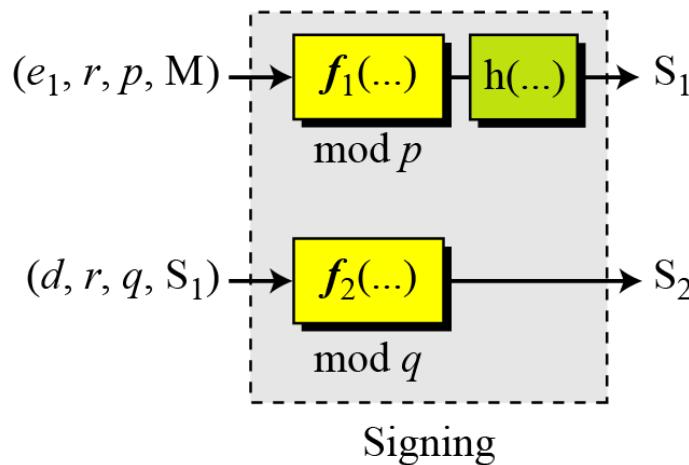
$S_1, S_2$ : Signatures

$(d)$ : Alice's private key

$M$ : Message

$r$ : Random secret

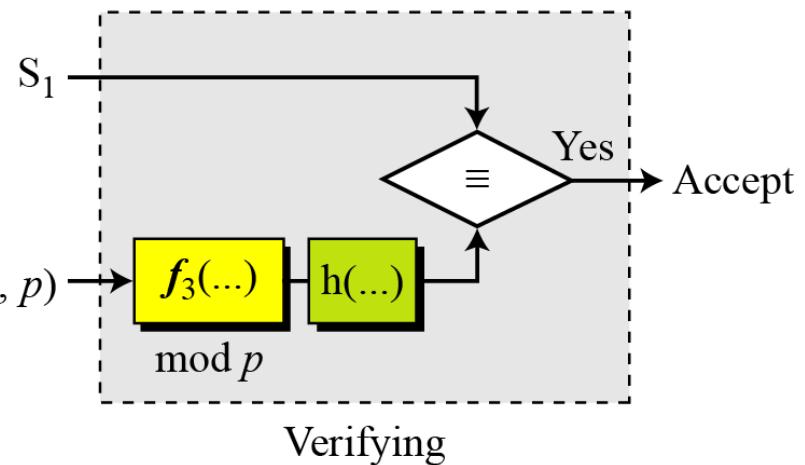
$(e_1, e_2, p, q)$ : Alice's public key



$S_2$

$(S_1, S_2, M, e_1, e_2, p)$

Signing



Verifying

# Digital Signature Standard (DSS) (NIST 1994)

Figure *General idea behind DSS scheme*  
*Based on ElGamal and Schnorr : reduce signature size*

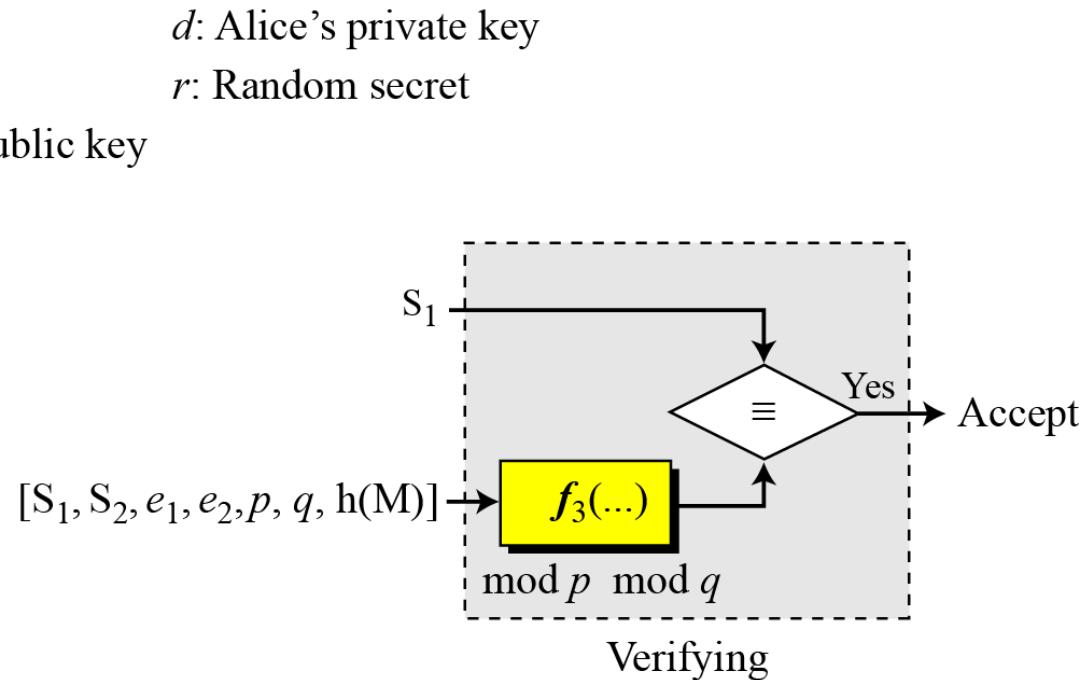
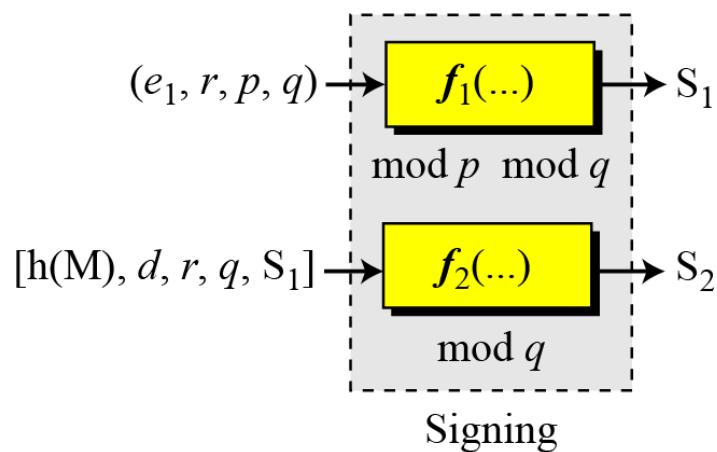
$S_1, S_2$ : Signatures

$d$ : Alice's private key

M: Message

$r$ : Random secret

$(e_1, e_2, p, q)$ : Alice's public key



# Continued

## DSS Versus RSA

Computation of DSS signatures is faster than computation of RSA signatures when using the same  $p$ .

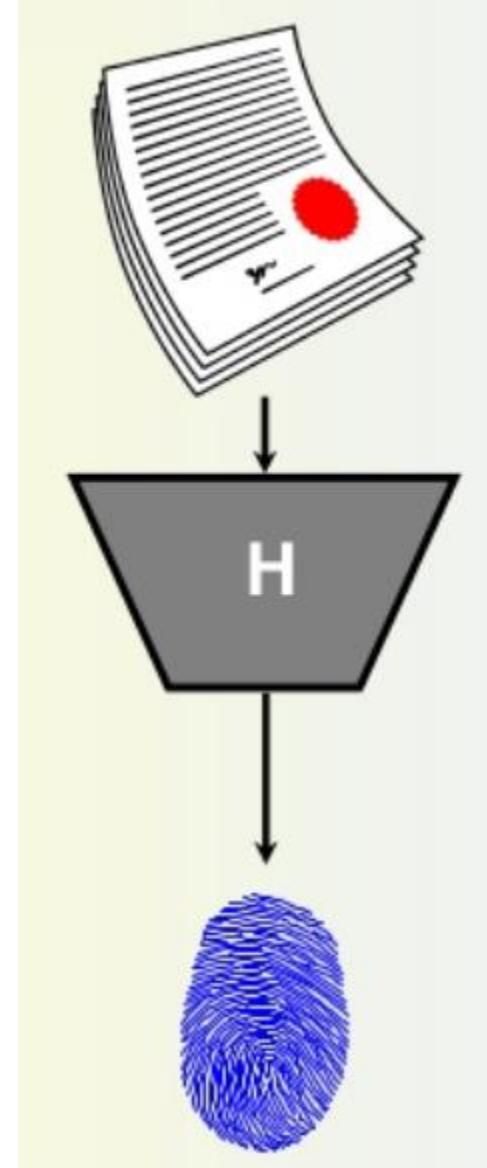
## DSS Versus ElGamal

DSS signatures are smaller than ElGamal signatures because  $q$  is smaller than  $p$ .

# Summary

- Public-key cryptosystems
- Applications for public-key cryptosystems
- Requirements for public-key cryptography
- Public-key cryptanalysis
- The RSA algorithm
  - Description of the algorithm
  - Computational aspects
  - Security of RSA

# Cryptographic Hash and MAC



## Outline

1. To introduce general ideas behind cryptographic hash functions
2. To discuss the Merkle-Damgard scheme as the basis for iterated hash functions
3. To distinguish between two categories of hash functions.
4. To discuss the structure of SHA-512.

## Outline

- ❑ To define message integrity
- ❑ To define message authentication
- ❑ To define criteria for a cryptographic hash function
- ❑ To define the Random Oracle Model and its role in evaluating the security of cryptographic hash functions
- ❑ To distinguish between an MDC and a MAC
- ❑ To discuss some common MACs

# **Message Authentication Requirements**

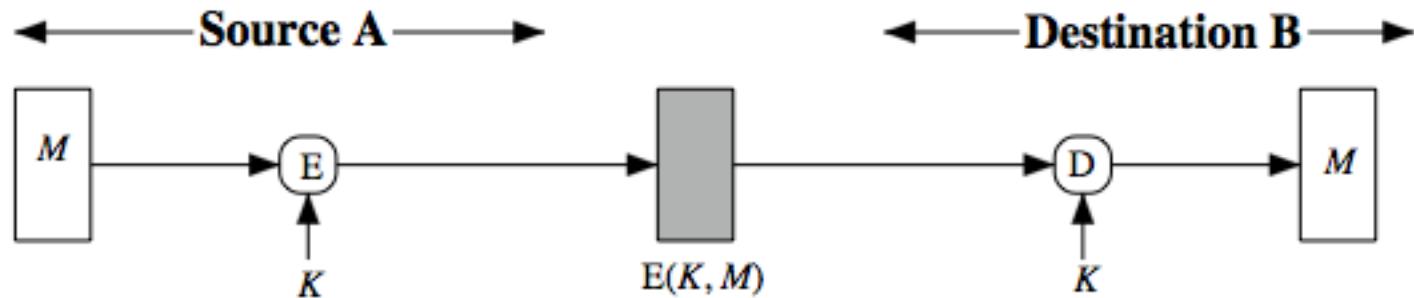
1. Disclosure
2. Traffic analysis
3. Masquerade
4. Content modification
5. Sequence modification
6. Timing modification
7. Source repudiation
8. Destination repudiation

# Message Authentication Functions

- ▶ **Hash function:** A function that maps a message of any length into a fixed length hash value, which serves as the authenticator
- ▶ **Message encryption:** The ciphertext of the entire message serves as its authenticator
- ▶ **Message authentication code (MAC):** A function of the message and a secret key that produces a fixed-length value that serves as the authenticator

# Symmetric Message Encryption

- ▶ encryption can also provides authentication
- ▶ if symmetric encryption is used then:
  - receiver know sender must have created it
  - since only sender and receiver now key used
  - know content cannot have been altered
  - if message has suitable structure, redundancy or a checksum to detect any changes



(a) Symmetric encryption: confidentiality and authentication

# Message Encryption

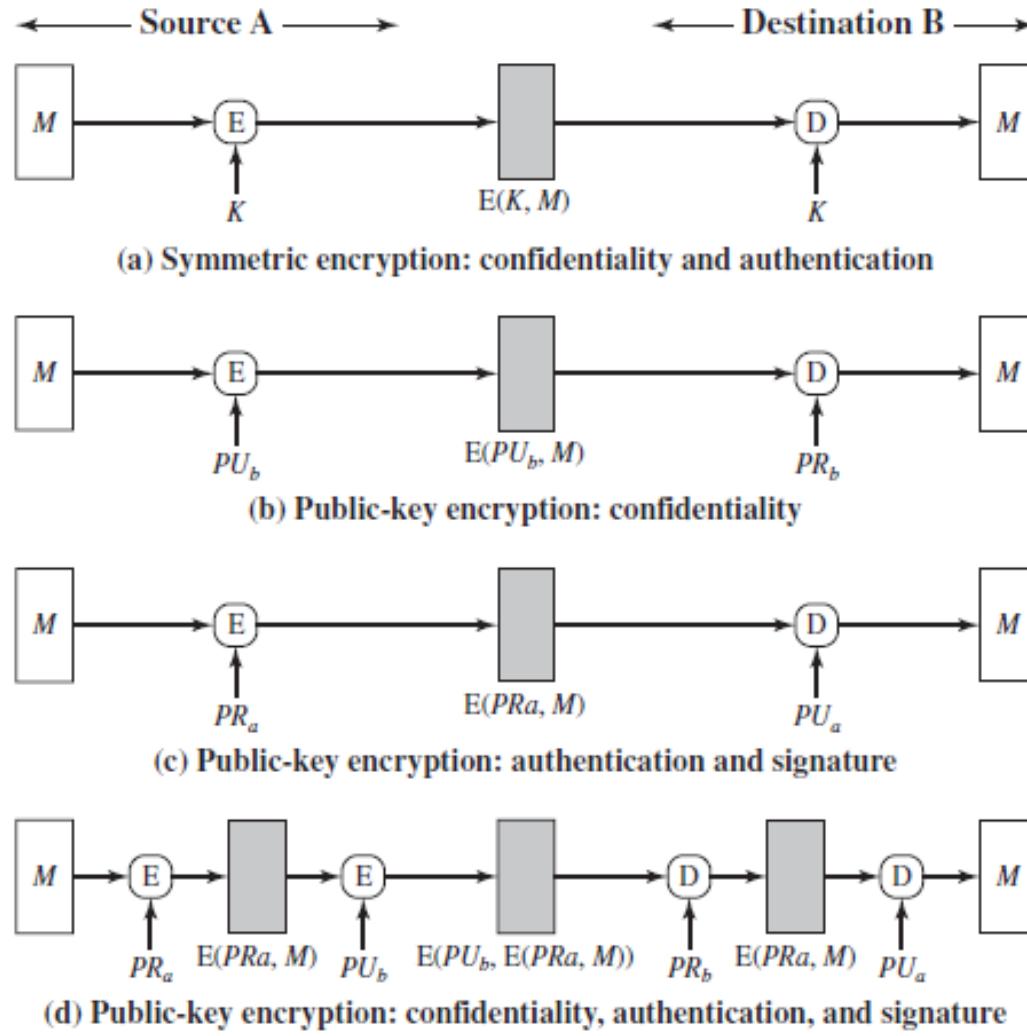


Figure 12.1 Basic Uses of Message Encryption

# Message Integrity

The cryptography systems that we have studied so far provide secrecy, or confidentiality, but not **integrity**. However, there are occasions where we may not even need secrecy but instead **must have integrity**.

## Topics:

- 1 Document and Fingerprint
- 2 Message and Message Digest
- 3 Difference
- 4 Checking Integrity
- 5 Cryptographic Hash Function Criteria

## Document and Fingerprint

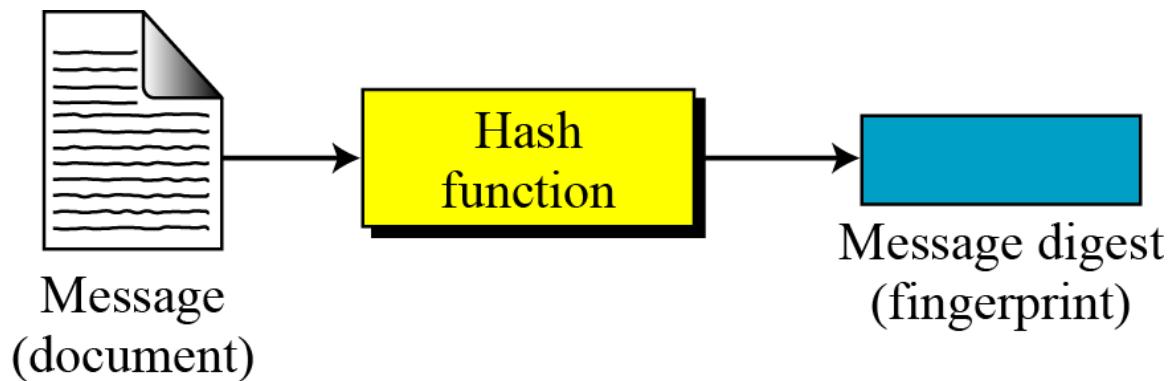
One way to preserve the integrity of a document is through the **use of a fingerprint**. If Alice needs to be sure that the contents of her document will not be changed, she can put her fingerprint at the bottom of the document.

## Message and Message Digest

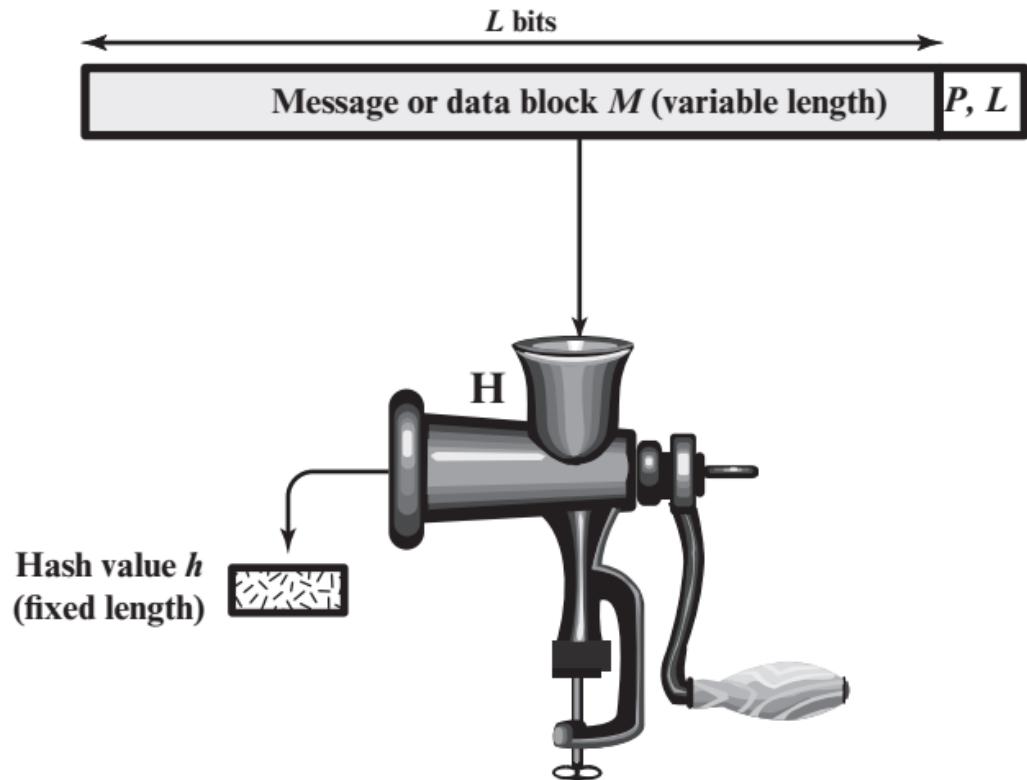
The electronic equivalent of the document and fingerprint pair is the message and digest pair.

A hash function  $H$  accepts a variable-length block of data  $M$  as input and produces a fixed-size hash value  $h = H(M)$ .

Figure Message and digest



# Cryptographic Hash Function



$P, L$  = padding plus length field

Figure Cryptographic Hash Function;  $h = H(M)$

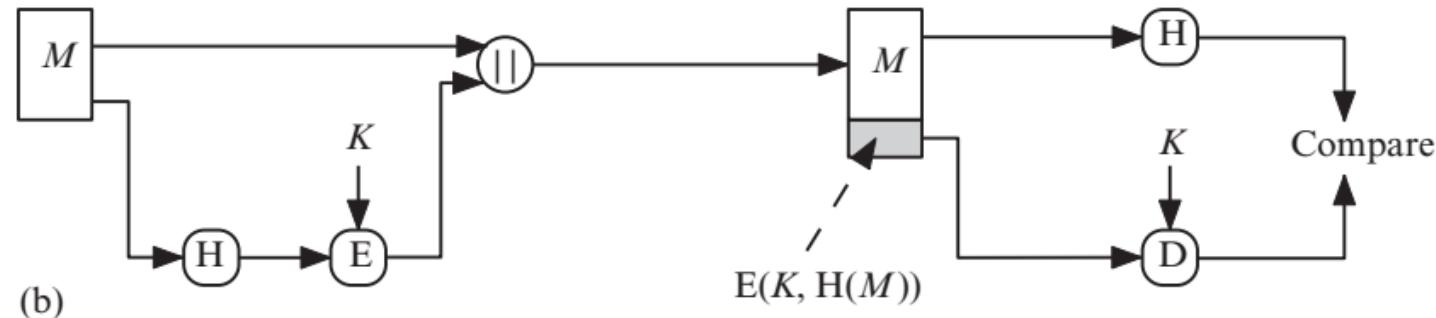
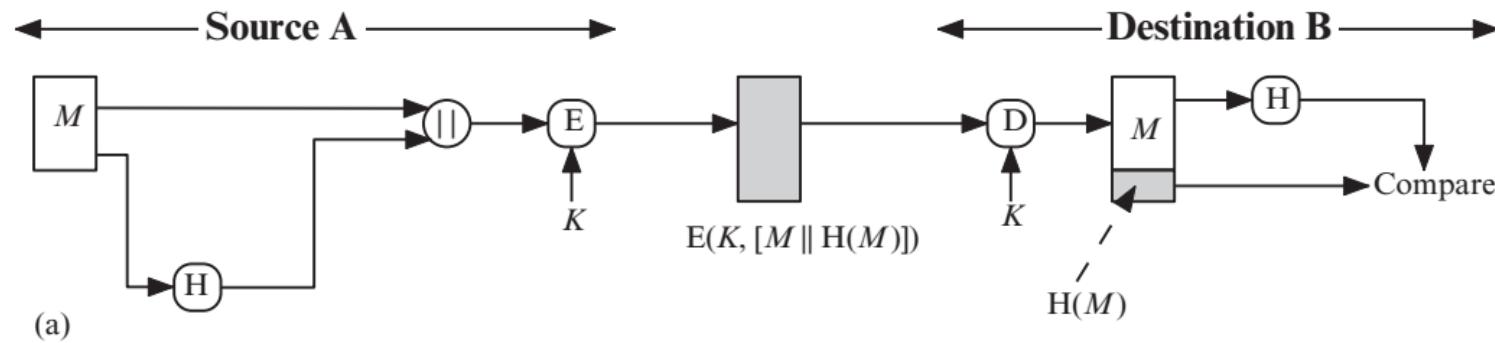
# Application of Cryptographic Hash Function

**Message Authentication:** Message authentication is a mechanism or service used to verify the integrity of a message.

When a hash function is used to provide message authentication, the hash function value is often referred to as a message digest.

# Application of Cryptographic Hash Function

Figures illustrates a variety of ways in which a hash code can be used to provide message authentication, as follows.



# Application of Cryptographic Hash Function

