Introduction to Course

Network Security

Introduction to Course

Objectives of the Topic

- After completing this topic, a student will be able to
  - get motivated, describe learning outcomes and describe the text and references books.

Introduction to Course

Motivation

- Before the widespread use of data processing equipment, security of information was provided primarily by physical and administrative means.
- E.g. rugged filing cabinets with locks

Introduction to Course

- Requirements for information security have undergone two major changes:
- a) As the computers were introduced, a need for protecting information stored on the shared computers was felt – Computer Security.

Introduction to Course

- b) As networks and communications facilities for carrying data from one computer to another were introduced, a need for protecting data during their transmission was felt – Network or Internet security.

Introduction to Course

Stored data:

- Business data must not be leaked to competitors
- Personal information
- Copyrighted software

Introduction to Course

Security Violations: Some Examples

- User A transmits a file to user B. User C, who is not authorized to read the file, is able to capture a copy of the file during its transmission – eavesdropping

Introduction to Course

- User D transmits a message to a computer E. User F intercepts the message, alters its contents and then forwards the message to E, which accepts the message as coming from D – Man-in-the-middle Attacks.

Introduction to Course

- It is also possible that user F constructs its own message and transmits that message to E as if it had come from computer D.

Introduction to Course

Some Other Common attacks

- Cryptanalysis
- Password Pilfering
- Intrusion
- Denial of Service Attacks
- Malicious software

Introduction to Course

Security Breaches can result in

- Financial loss for corporations
- Theft of intellectual property
- Lawsuits
- Threat to public safety

Introduction to Course

- The field of network and Internet security consists of measures to deter, prevent, detect, and correct security violations that involve the transmission of information.

Introduction to Course

Required Books

- W. Stallings, *"Network Security Essentials: Applications and Standards"*, Pearson Education, 2014
- *"CCNA Security 1.1 Student Packet Tracer Manual "* Cisco Networking Academy, 2012

Introduction to Course

Reference Books

- W. Stallings, "*Cryptography and Network Security Principles and Practice*", Pearson Education, 2014

Introduction to Course

Course Composition: Two parts

- Part 1 will provide a practical survey of network security applications and standards.
- It has been sub-divided into 3 subparts.

Introduction to Course

Subpart1: Cryptography

- Symmetric Encryption principles
- Public-Key Cryptography and message authentication

Introduction to Course

Subpart2: Network Security Applications

- Key distribution and user authentication
- Network Access Control and cloud Security
- Transport-level Security
- Wireless Net. Security

Introduction to Course

Subpart2: Network Security Applications

- Electronic Mail Security
- IP Security

Introduction to Course

Subpart3: System Security

- Malicious Software
- Intrusions
- Firewalls

Introduction to Course

- In Part 2, we will perform lab experiments to configure networks employing Cisco components for various security aspects.
- Packet Tracer will be used.

Introduction to Course

Grading Policy

- Assignments + Quizzes = 15%
- Mid Term Exam = 35%
- Final Term Exam = 50%

End

Definition Of Computer Security

Network Security

Definition Of Computer Security

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe a definition of the computer security.

Definition Of Computer Security

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

Definition Of Computer Security

- The National Institute of Standards and Technology (NIST) Computer Security Handbook defines the term computer security as follows:

Definition 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.

Definition Of Computer Security

- Resources include hardware, software, firmware, information/data, and telecommunications.

Definition Of Computer Security

Computer Security: Three Key Objectives

- Confidentiality
- Integrity
- Availability

Definition Of Computer Security

Confidentiality

- a) Data confidentiality: Assures that private or confidential information is not made available or disclosed to unauthorized individuals.

Definition Of Computer Security

Confidentiality

- b) 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.

Definition Of Computer Security

Integrity

- a) Data integrity: Assures that information and programs are changed only in a specified and authorized manner.

Definition Of Computer Security

Integrity

- b) System integrity: Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.

Definition Of Computer Security

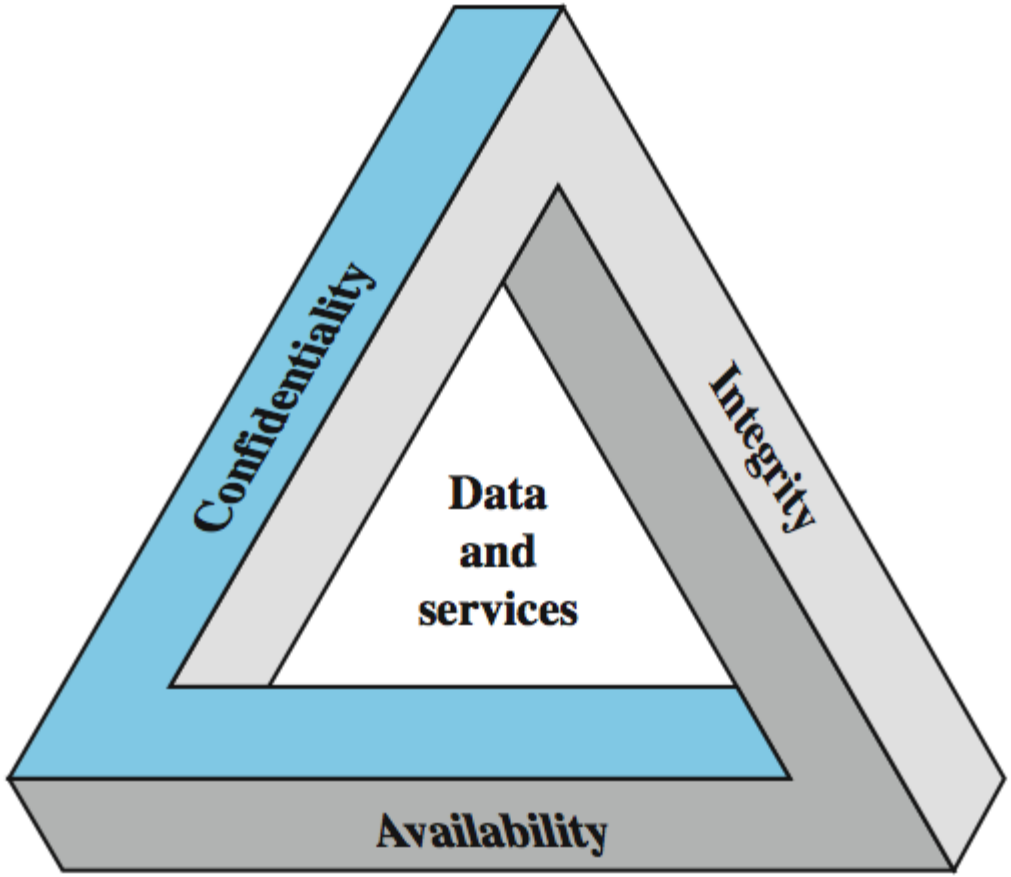
Availability

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

Definition Of Computer Security

The Security Requirements Triad:

CIA Triad



Definition Of Computer Security

Possible Additional Concepts:

- Authenticity
- Accountability

Definition Of Computer Security

Authenticity:

- The property of being genuine and being able to be verified and trusted.
- verifying that users are who they say they are and that each input arriving at the system came from a trusted source.

Definition Of Computer Security

Accountability:

- The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity.
- This supports nonrepudiation, intrusion detection and prevention etc.

Definition Of Computer Security

Accountability:

- Systems must keep records of their activities to permit later forensic analysis to trace security breaches or to aid in transaction disputes.

End

Impact Of A Security Breach

Network Security

Impact Of A Security Breach

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe levels of impact of a security breach on the system.

Impact Of A Security Breach

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

Impact Of A Security Breach

- In case, there be a breach of security (i.e., a loss of confidentiality, integrity, or availability), three levels of impact on organizations or individuals can be considered.

Impact Of A Security Breach

Low

- The loss could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals.
- A limited adverse effect means that, a security breach might

Impact Of A Security Breach

- (i) cause a degradation in mission capability to an extent and duration that the organization is able to perform its primary functions, but the effectiveness of the functions is noticeably reduced

Impact Of A Security Breach

- (ii) result in minor damage to organizational assets
- (iii) result in minor financial loss
- or (iv) result in minor harm to individuals.

Impact Of A Security Breach

Moderate

- The loss could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals.
- A serious adverse effect means that the loss might

Impact Of A Security Breach

- (i) cause a significant degradation in mission capability to an extent and duration that the organization is able to perform its primary functions, but the effectiveness of the functions is significantly reduced.

Impact Of A Security Breach

- (ii) result in significant damage to organizational assets
- (iii) result in significant financial loss
- or (iv) result in significant harm to individuals that does not involve loss of life or serious, life-threatening injuries.

Impact Of A Security Breach

High

- The loss could be expected to have a severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals.
- A catastrophic adverse effect means that,

Impact Of A Security Breach

- (i) cause a severe degradation in or loss of mission capability to an extent and duration that the organization is not able to perform one or more of its primary functions

Impact Of A Security Breach

- (ii) result in major damage to organizational assets
- (iii) result in major financial loss
- or (iv) result in severe or catastrophic harm to individuals involving loss of life or serious, life-threatening injuries

Impact Of A Security Breach

Examples of Security Requirements

- **a) Confidentiality:**
- Student grade information is an asset whose confidentiality is considered to be highly important by students.

Impact Of A Security Breach

- Grade information should only be available to students, their parents, and employees that require the information to do their job.
- Student enrollment information may have a moderate confidentiality rating.

Impact Of A Security Breach

- Student enrollment information is seen by more people on a daily basis, is less likely to be targeted than grade information and results in less damage if disclosed.

Impact Of A Security Breach

- Directory Information such as lists of students or faculty may be assigned a low or no confidentiality rating.

Impact Of A Security Breach

- **b) Integrity:**
- Assume a hospital patient's allergy information to be stored in a database.
- The doctor should be able to trust that the information is correct and current.

Impact Of A Security Breach

- Suppose a nurse who is authorized to update this info deliberately falsifies the data to cause harm to the hospital.
- Restore to a trusted basis quickly and to trace the error back to the person responsible.

Impact Of A Security Breach

- Patient allergy information requires high integrity.
- Inaccurate information could result in serious harm or death to a patient and expose the hospital to massive liability.

Impact Of A Security Breach

- 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.

Impact Of A Security Breach

Availability

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

Impact Of A Security Breach

- A system that provides authentication services.
- An interruption results in inability for customers to access computing resources and for staff to access resources to perform critical tasks.

Impact Of A Security Breach

- 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.

End

Challenges Of Network Security

Challenges Of Network Security

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain challenges of network security.

Challenges Of Network Security

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

Challenges Of Network Security

Computer Security

- Is the generic name for the collection of tools designed to protect data stored on computers and to thwart hackers.

Challenges Of Network Security

- The use of networks and communications facilities allows for carrying data from one computer to another.
- **Network or internet security** measures are needed to protect data during their transmission.

Challenges Of Network Security

- Network or internet security consists of measures to deter, prevent, detect, and correct security violations that involve the transmission of information.

Challenges Of Network Security

Deterrence

- is usually the first line of defense against intruders who may try to gain access.
- It works by creating an atmosphere intended to frighten intruders.

Challenges Of Network Security

Prevention

- is the process of trying to stop intruders from gaining access to the resources of the system.
- Barriers include firewalls, demilitalized zones (DMZs).

Challenges Of Network Security

Detection

- occurs when the intruder has succeeded or is in the process of gaining access to the system.
- Signals from the detection process include alerts to the existence of an intruder.

Challenges Of Network Security

Response

- is an aftereffect mechanism that tries to respond to the failure of the first three mechanisms.
- It works by trying to stop and/or prevent future damage or access to a facility.

Challenges Of Network Security

- No clear boundaries between Computer and Network forms of security.
- E.g., a virus may be introduced into a system physically when it arrives on an optical disk, or arrives over an internet.

Challenges Of Network Security

Challenges:

- 1) Security is not as simple as it might first appear to the novice.
- Mechanisms used to provide confidentiality, authentication, nonrepudiation, integrity are quite complex.

Challenges Of Network Security

- 2) In developing a security mechanism or algorithm, one must always consider potential attacks on security features.
- Successful attacks are designed by exploiting an unexpected weakness in the mechanism.

Challenges Of Network Security

- 3) Procedures used to provide particular services are often counterintuitive.
- It is only when the various aspects of the threat are considered that elaborate security mechanisms make sense.

Challenges Of Network Security

- 4) At what points in a network, are certain security mechanisms needed and at what layer(s) of an architecture such as TCP/IP should mechanisms be placed.

Challenges Of Network Security

- 5) Participants of security mechanisms may be in possession of some secret information (e.g., an encryption key), which raises questions about the creation, distribution, and protection of that secret information.

Challenges Of Network Security

- 6) Security is essentially a battle of wits bet. a perpetrator and the designer.
- The attacker needs only find a single weakness, while designer must find and eliminate all weaknesses to achieve perfect security.

Challenges Of Network Security

- 7) Security requires constant monitoring, and this is difficult in today's overloaded environment.
- 8) Little benefit from security investment is perceived until a security failure occurs.

Challenges Of Network Security

- 9) Strong security is often viewed as an impediment to efficient and user-friendly operation.

End

The OSI Security Architecture

Network Security

The OSI Security Architecture

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe the OSI security architecture and its usefulness.

The OSI Security Architecture

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

The OSI Security Architecture

- In an organization, the manager responsible for security has to effectively assess the security needs of an organization.
- He has to evaluate and choose various security products and policies.

The OSI Security Architecture

- Thus, the manager needs some systematic way of defining the requirements for security and characterizing the approaches to satisfying those requirements.

The OSI Security Architecture

- This is difficult enough in a centralized data processing environment.
- With the use of local and wide area networks, the problems are compounded.

The OSI Security Architecture

- Such a systematic approach was defined by the International Telecommunication Union (ITU) Telecommunication Standardization Sector (ITU-T).

The OSI Security Architecture

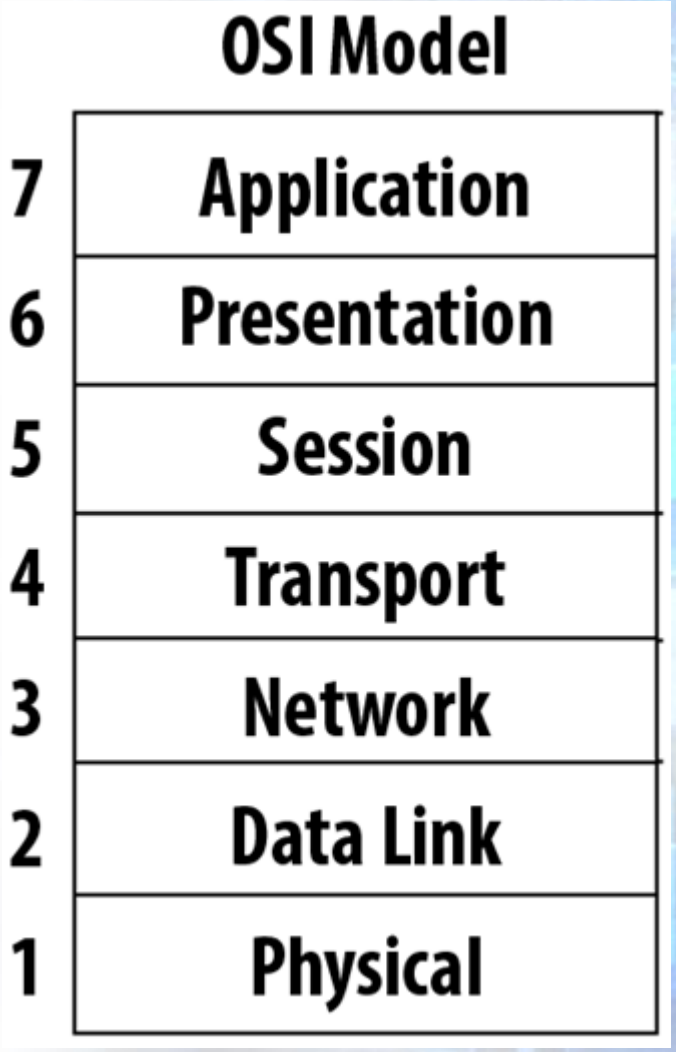
- (ITU-T) is a United Nations sponsored agency that develops standards, called Recommendations, relating to telecommunications and to open systems interconnection (OSI).

The OSI Security Architecture

- Recommendation X.800, Security Architecture for OSI.
- The open systems interconnection (OSI) security architecture was developed in the context of the OSI protocol architecture.

The OSI Security Architecture

OSI protocol architecture



The OSI Security Architecture

- The OSI security architecture is useful to managers as a way of organizing the task of providing security.
- It focuses on security attacks, mechanisms, and services.
- These are defined next:

The OSI Security Architecture

Security Attack:

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

The OSI Security Architecture

Security Mechanism:

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

The OSI Security Architecture

Security Service:

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

The OSI Security Architecture

- The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service.

The OSI Security Architecture

RFC 4949, Internet Security Glossary.

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.

The OSI Security Architecture

- The OSI architecture is an international standard, computer and communications vendors have developed security features for their products and services that relate to this structured definition of services and mechanisms.

End

Security Attacks

Network Security

Security Attacks

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe the security attacks.

Security Attacks

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

Security Attacks

- According to the OSI Architecture X.800, security attacks can be classified in two categories:
- passive attacks, and
- active attacks

Security 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.

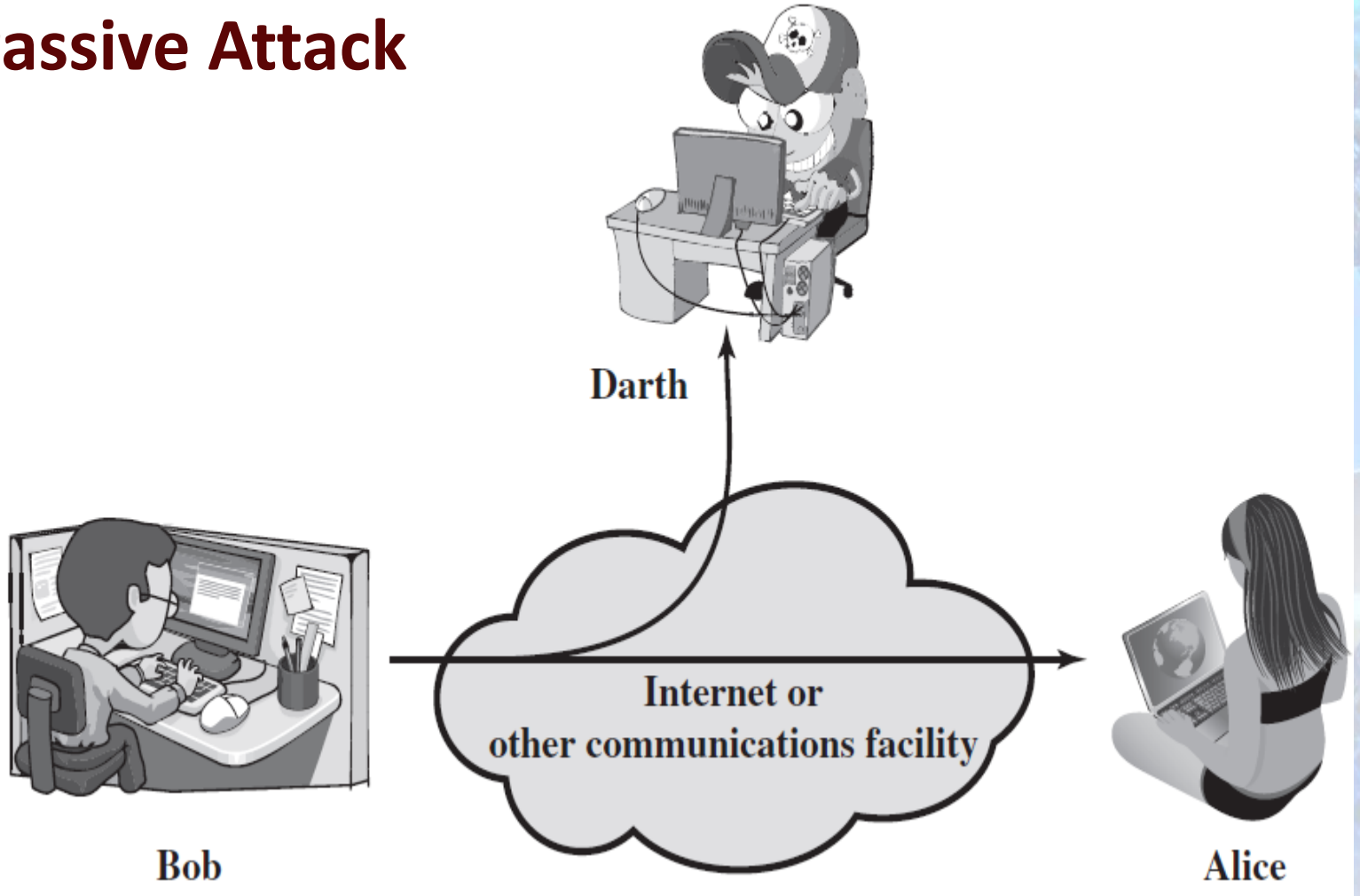
Security Attacks

Passive Attacks:

- Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions.
- The goal of the opponent is to obtain information that is being transmitted.

Security Attacks

Passive Attack



Security Attacks

- There are two types of passive attacks
- release of message contents, and
- traffic analysis.

Security Attacks

Release of message contents:

- A telephone conversation, an e-mail message, and a transferred file may contain confidential info. Prevent an opponent from learning contents of these transmissions.

Security Attacks

Traffic Analysis:

- Even if contents of messages are encrypted, an opponent might still be able to observe the pattern of these messages.

Security Attacks

- He could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged.
- He can guess the nature of the communication.

Security Attacks

- Passive attacks do not alter the data.
- Neither the sender nor receiver is aware that a third party has observed the traffic pattern.
- Emphasis is on prevention rather than detection.
- Use Encryption.

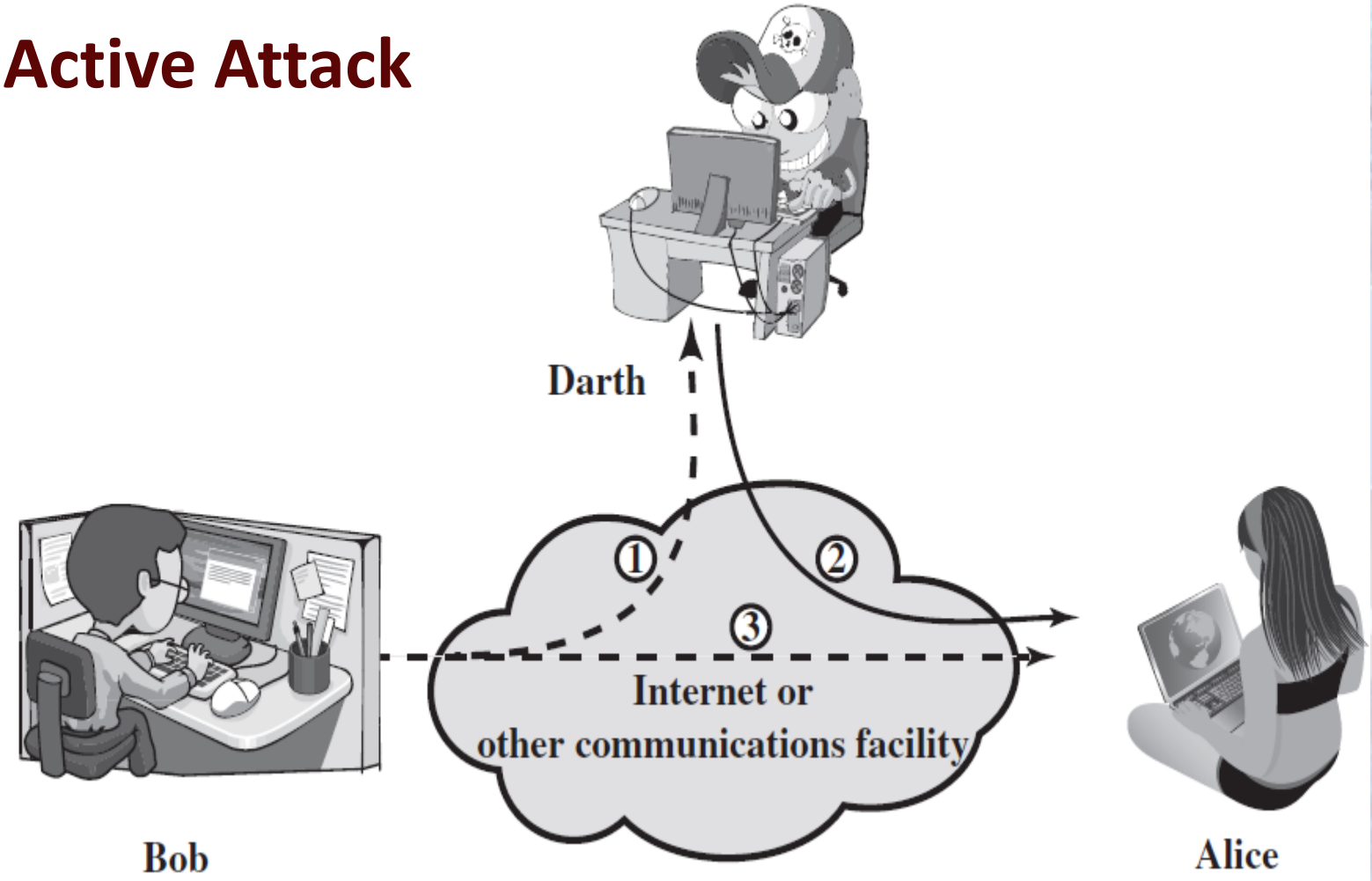
Security Attacks

Active Attacks:

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

Security Attacks

Active Attack



Security Attacks

- Active attacks can be subdivided into four categories:
- masquerade,
- replay,
- modification of messages, and
- denial of service.

Security Attacks

Masquerade:

- It takes place when one entity pretends to be a different entity.
- It usually includes one of the other forms of active attack.

Security Attacks

Replay:

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

Security Attacks

Modification of messages:

- It simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect.

Security Attacks

Denial of Service:

- It prevents or inhibits the normal use or management of communications facilities.
- E.g. an entity may suppress all messages directed to a particular destination.

Security Attacks

- Active Attacks are 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.

End

Authentication, Access Control

Network Security

Authentication, Access Control

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain authentication and access control services.

Authentication, Access Control

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

Authentication, Access Control

Security Services: Defined by X.800, OSI Security Architecture:

- a service provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or of data transfers.

Authentication, Access Control

Defined by RFC 4949:

- a processing or communication service provided by a system to give a specific kind of protection to system resources

Authentication, Access Control

- Security services implement security policies and are implemented by security mechanisms.

Authentication, Access Control

X.800 Service Categories:

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

Authentication, Access Control

Authentication :

- Concerned with assuring that a communication is authentic.
- In the case of a single message, assures the recipient that the message is from the source that it claims to be from.

Authentication, Access Control

- In the case of an ongoing interaction, two aspects are involved:
- First, at the time of connection initiation, the service assures that the two entities are authentic, that is, that each is the entity that it claims to be.

Authentication, Access Control

- Second, the service must assure that the connection is not interfered with in such a way that a third party can masquerade as one of the two legitimate parties for the purposes of unauthorized transmission or reception.

Authentication, Access Control

- Two specific authentication services are defined in X.800:
- Peer entity authentication
- Data origin authentication

Authentication, Access Control

Peer entity Authentication :

- Provides for the corroboration of the identity of a peer entity in an association.

Authentication, Access Control

- It’s provided for use at establishment of or during the data transfer phase of a connection.
- Provides confidence that an entity is neither performing a masquerade nor an unauthorized replay of a previous connection.

Authentication, Access Control

Data origin authentication:

- Provides for the corroboration of the source of a data unit.
- It does not provide protection against the duplication or modification of data units.

Authentication, Access Control

- This type of service supports applications like electronic mail, where there are no prior interactions between the communicating entities.

Authentication, Access Control

Access Control:

- It is the ability to limit and control the access to host systems and applications via communications links.

Authentication, Access Control

- 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.

End

Confidentiality, Integrity, Nonrepudiation

Network Security

Confidentiality, Integrity, Nonrepudiation

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain data confidentiality, data integrity and nonrepudiation services.

Confidentiality, Integrity, Nonrepudiation

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

Confidentiality, Integrity, Nonrepudiation

X.800 Service Categories:

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

Confidentiality, Integrity, Nonrepudiation

Data Confidentiality:

- is the protection of transmitted data from passive attacks.
- Assurance that data received are exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay).

Confidentiality, Integrity, Nonrepudiation

- Broadest service protects all user data transmitted between two users over a period of time.
- E.g. when a TCP connection is set up bet. two systems, it prevents the release of any user data transmitted.

Confidentiality, Integrity, Nonrepudiation

- Narrower forms of service include the protection of a single message or even specific fields within a message.

Confidentiality, Integrity, Nonrepudiation

- The other aspect is the protection of traffic flow from analysis.
- This requires that an attacker not be able to observe the source and destination, length, or other characteristics of the traffic.

Confidentiality, Integrity, Nonrepudiation

Data Integrity:

- can apply to a stream of messages, a single message, or selected fields within a message.
- the most useful and straightforward approach is total stream protection.

Confidentiality, Integrity, Nonrepudiation

- 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.

Confidentiality, Integrity, Nonrepudiation

- the connection-oriented integrity service addresses both message stream modification and denial of service.

Confidentiality, Integrity, Nonrepudiation

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

Confidentiality, Integrity, Nonrepudiation

- Because the integrity service relates to active attacks, we are concerned with detection rather than prevention.
- automated recovery mechanisms allow to recover from the loss of integrity of data.

Confidentiality, Integrity, Nonrepudiation

Nonrepudiation:

- prevents either sender or receiver from denying a transmitted message.

Confidentiality, Integrity, Nonrepudiation

- 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.

Confidentiality, Integrity, Nonrepudiation

Availability Service:

- 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.

Confidentiality, Integrity, Nonrepudiation

- 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.

End

By-Pin2

Security Mechanisms

Network Security

Security Mechanisms

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe security mechanisms.

Security Mechanisms

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

Security Mechanisms

- Security mechanisms defined by X.800, Security Architecture for OSI can be divided into two broad categories w.r.t their implementation.

Security Mechanisms

A) Specific Security Mechanisms

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

Security Mechanisms

Encipherment

- Use of mathematical algorithms to transform data into a form that is not readily intelligible.
- The transformation and subsequent recovery depend on the algorithm and encryption keys.

Security Mechanisms

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.

Security Mechanisms

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.

Security Mechanisms

Authentication Exchange

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

Security Mechanisms

Traffic Padding

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

Security Mechanisms

Routing Control

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

Security Mechanisms

Notarization

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

Security Mechanisms

B) Pervasive Security Mechanisms

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

Security Mechanisms

Trusted Functionality

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

Security Mechanisms

Security Label

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

Security Mechanisms

Event Detection

- Detection of security-relevant events.

Security Mechanisms

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 Mechanisms

Security Recovery

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

Security Mechanisms

Relationship

SERVICE \ MECHANISM	Encipherment	Digital signature	Access control	Data integrity	Authentication exchange	Traffic padding	Routing control	Notarization
Peer entity authentication	Y	Y			Y			
Data origin authentication	Y	Y						
Access control			Y					
Confidentiality	Y						Y	
Traffic flow confidentiality	Y					Y	Y	
Data integrity	Y	Y		Y				
Nonrepudiation		Y		Y				Y
Availability				Y	Y			

A Model For Network Security

Network Security

A Model For Network Security

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe a model for network security.

A Model For Network Security

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

A Model For Network Security

- Assume a message is to be transferred from one party to another across some sort of Internet service.
- The two parties, who are the principals in this transaction, must cooperate for the exchange to take place.

A Model For Network Security

- A logical information channel is established by defining a route through the Internet from source to destination and by the cooperative use of communication protocols (e.g., TCP/IP) by the two principals.

A Model For Network Security

- To protect the information from an opponent who may present a threat to confidentiality, authenticity, and so on, security comes into play.
- All of the security techniques have two components:

A Model For Network Security

- 1. A security-related transformation on the information to be sent.
- Example1: encryption of the message, which scrambles the message so that it is unreadable by the opponent.

A Model For Network Security

- Example2: the addition of a code based on the contents of the message, which can be used to verify the identity of the sender.

A Model For Network Security

- 2. Some secret information shared by the two principals and unknown to the opponent.
- E.g. encryption key used with the transformation to scramble the message before transmission and unscramble it on reception.

A Model For Network Security

- A trusted third party (TTP) may be needed to achieve secure transmission.
- E.g. a TTP may be responsible for distributing the secret information to the two principals while keeping it from any opponent.

A Model For Network Security

- This general model shows that there are four basic tasks in designing a particular security service:

A Model For Network Security

- 1. Design an algorithm for the security-related transformation. An opponent should not be able to defeat purpose of the algorithm.
- 2. Generate the secret information used by the algorithm.

A Model For Network Security

- 3. Develop methods for the distribution and sharing of the secret information.
- 4. Specify a protocol enabling the principals to use the security algorithm and the secret information for a particular security service.

A Model For Network Security

A Generic Model For Network Security

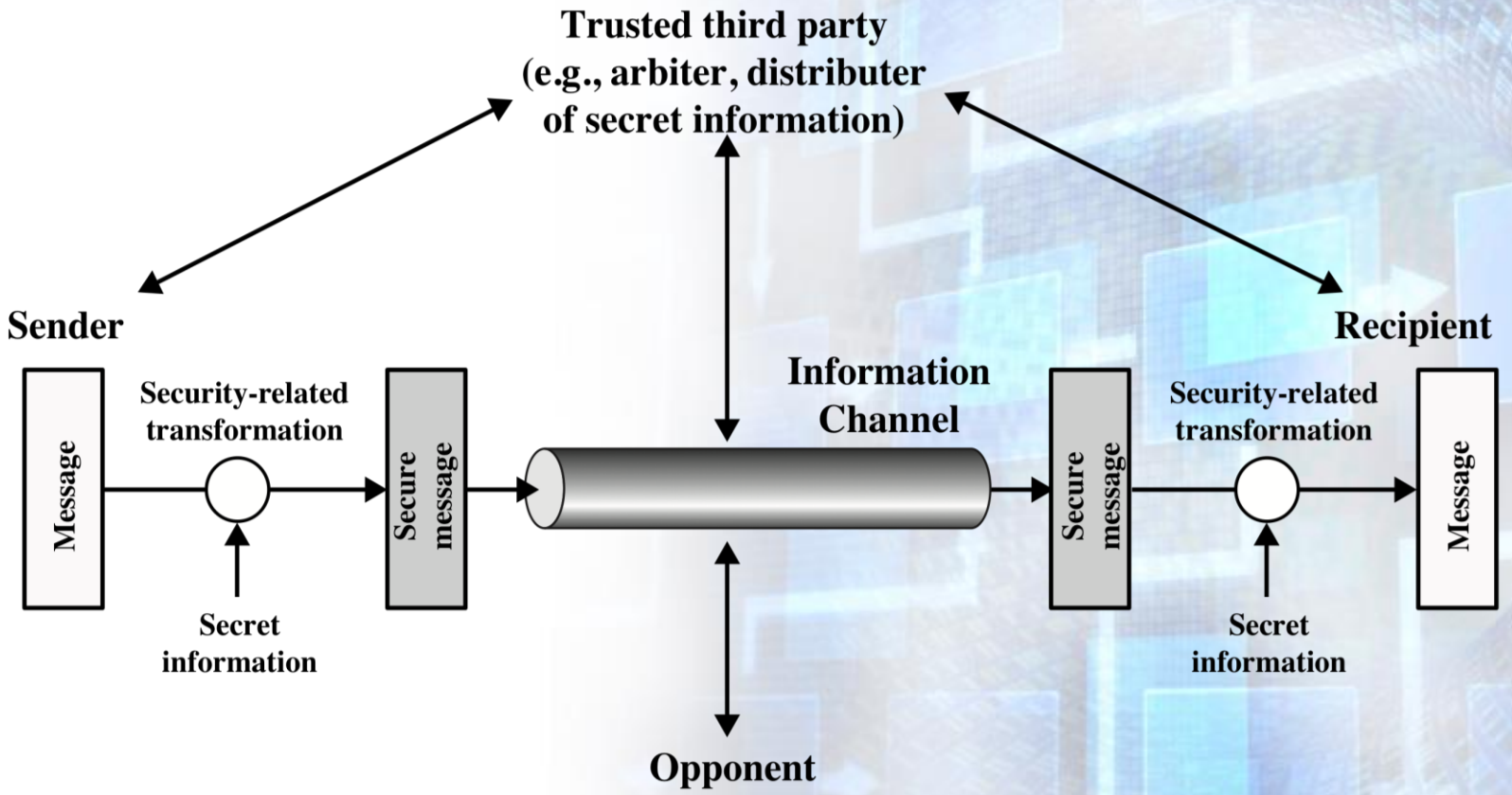


Figure 1.2 Model for Network Security

A Model For Network Security

- Next, we describe a general model which reflects a concern for protecting an information system from unwanted access.
- E.g. A hacker who attempts to penetrate system that can be accessed over a net.

A Model For Network Security

- An intruder can be a disgruntled employee who wishes to do damage or a criminal who seeks to exploit computer assets for financial gain (e.g., obtaining credit card numbers or performing illegal money transfers).

A Model For Network Security

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

A Model For Network Security

Network Access Security Model

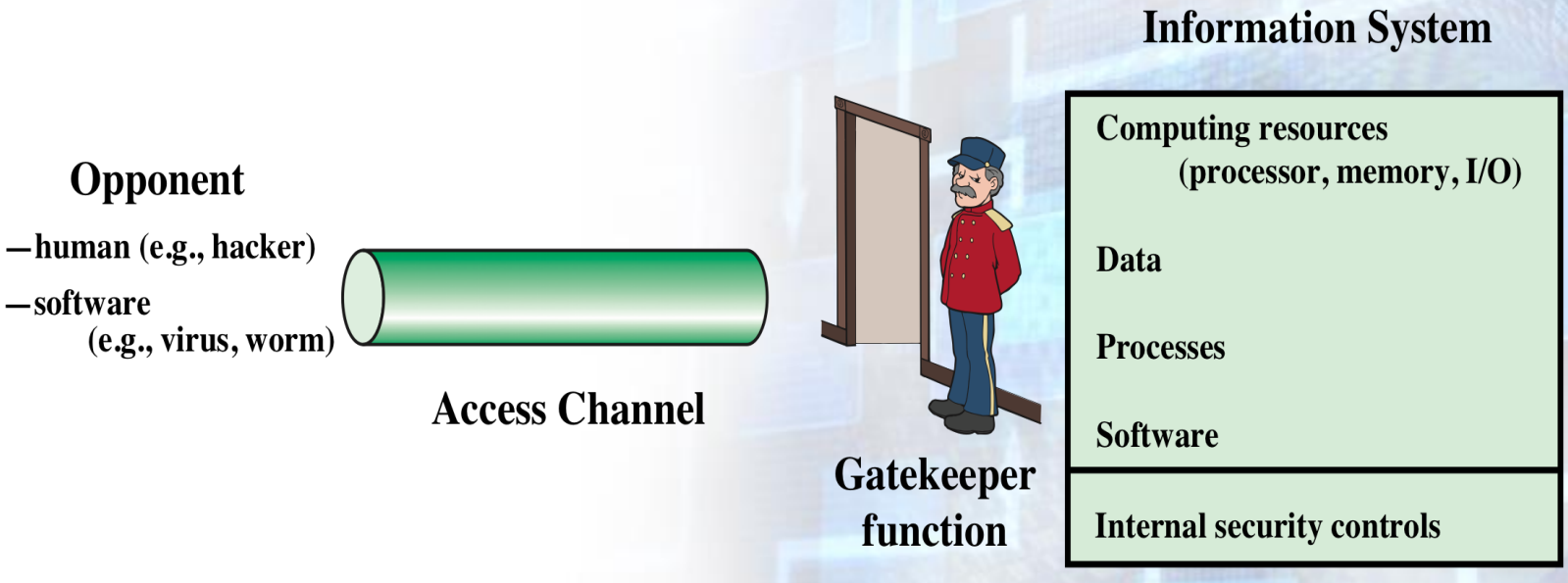


Figure 1.3 Network Access Security Model

A Model For Network Security

- Another type of unwanted access is the placement in a computer system of logic that exploits vulnerabilities in the system and that can affect application programs.

End

By-Pin2

Basics Of Symmetric Encryption

Network Security

Basics Of Symmetric Encryption

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe basics of symmetric encryption.

Basics Of Symmetric Encryption

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

Basics Of Symmetric Encryption

- Symmetric encryption, or single-key encryption, was the only type of encryption in use prior to the development of public key encryption in the 1970s.

Basics Of Symmetric Encryption

Some Basic Terminology

- Plaintext - original message
- Ciphertext - coded message
- Cipher - algorithm for transforming plaintext to ciphertext .

Basics Of Symmetric Encryption

- Key - info used in cipher known only to sender/receiver
- Encipher (encrypt) - converting plaintext to ciphertext
- Decipher (decrypt) - recovering ciphertext from plaintext

Basics Of Symmetric Encryption

- Cryptography - study of encryption principles/methods
- Cryptanalysis (code breaking) - study of principles/methods of deciphering ciphertext without knowing key
- Cryptology - field of both cryptography and cryptanalysis

Basics Of Symmetric Encryption

Symmetric Encryption Principles

- A symmetric encryption scheme has five ingredients

Basics Of Symmetric Encryption

Plaintext

- This is the original intelligible message or data that is fed into the algorithm as input.

Encryption Algorithm

- It performs various substitutions and transformations on the plaintext.

Basics Of Symmetric Encryption

Secret Key

- It is also input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm.

Basics Of Symmetric Encryption

Secret Key …

- The algorithm will produce a different output depending on the specific key being used at the time.
- The exact substitutions and transformations performed depend on the key.

Basics Of Symmetric Encryption

Ciphertext

- This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts (is unintelligible).

Basics Of Symmetric Encryption

Decryption algorithm

- This is essentially the encryption algorithm run in reverse.
- It takes the ciphertext and the secret key and produces the original plaintext.

Basics Of Symmetric Encryption

Model of Symmetric Encryption

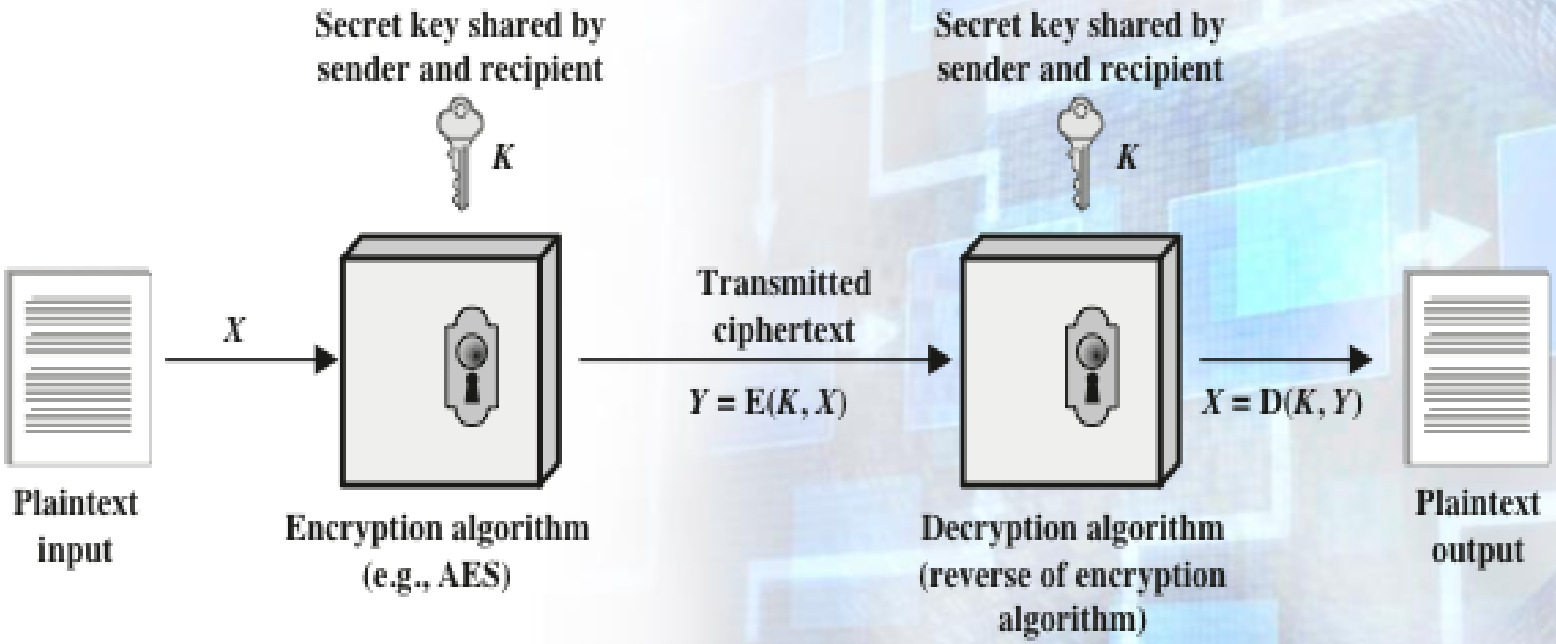


Figure 2.1 Simplified Model of Symmetric Encryption

Basics Of Symmetric Encryption

Requirements

- Two requirements for secure use of symmetric encryption

Basics Of Symmetric Encryption

- 1. We need a strong encryption algorithm.
- An opponent should be unable to decrypt ciphertext or discover the key even if he is in possession of a no. of ciphertexts together with the plaintext that produced each ciphertext.

Basics Of Symmetric Encryption

- 2. Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure.
- If someone discovers the key and knows the algorithm, all communication using this key is readable.

Basics Of Symmetric Encryption

- The security of symmetric encryption depends on the secrecy of the key, not the secrecy of the algorithm

Basics Of Symmetric Encryption

- It is impractical to decrypt a message on the basis of the ciphertext plus knowledge of the encryption/decryption algorithm.
- This makes it feasible for widespread use.

Basics Of Symmetric Encryption

End

- Manufacturers can and have developed low-cost chip implementations of data encryption algorithms.
- These chips are widely available and incorporated into a number of products.

Cryptanalysis

Network Security

Cryptanalysis

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain Cryptanalysis.

Cryptanalysis

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

Cryptanalysis

- Cryptography - study of encryption principles/methods.
- Cryptanalysis (code breaking) - study of principles/methods of deciphering ciphertext without knowing key.

Cryptanalysis

- Cryptographic systems are generically classified along three independent dimensions.

Cryptanalysis

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

- two general principles:
- Substitution
- Transposition

Cryptanalysis

Substitution:

- Each element (bit, letter, group of bits or letters) in the plaintext is mapped into another element.

Cryptanalysis

Transposition:

- Elements in the plaintext are rearranged.
- Fundamental requirement is that no information be lost.
- Product systems involve multiple stages of substitutions and transpositions.

Cryptanalysis

2. The number of keys used.

- Referred to as symmetric, single-key, secret-key, or conventional encryption if both sender and receiver use the same key.

Cryptanalysis

- Referred to as asymmetric, two-key, or public-key encryption if the sender and receiver each use a different key.

Cryptanalysis

3. The way in which the plaintext is processed.

- A block cipher processes the input one block of elements at a time, producing an output block for each input block.

Cryptanalysis

- A stream cipher processes the input elements continuously, producing output one element at a time, as it goes along.

Cryptanalysis

Cryptanalysis

- The strategy used by the cryptanalyst depends on the nature of the encryption scheme and the information available to the cryptanalyst.

Cryptanalysis

Ciphertext Only:

- The cryptanalyst knows ciphertext only.
- Uses brute-force approach - try all possible keys.
- Make the key space very large so it becomes impractical.
- Easiest to defend

Cryptanalysis

Known plaintext:

- The analyst may be able to capture one or more plaintext messages as well as their encryptions.
- Or he may know that certain plaintext patterns will appear in a message.
- May deduce the key.

Cryptanalysis

Probable-word:

- An opponent may know parts of the message, then he can obtain specific information.
- E.g. an accounting file is being transmitted, placement of certain key words in the header of the file.

Cryptanalysis

Chosen-plaintext:

- If the analyst is able to choose the messages to encrypt, the analyst may deliberately pick patterns that can be expected to reveal the structure of the key.

Cryptanalysis

Type of Attack	Known to Cryptanalyst
Ciphertext Only	• Encryption algorithm • Ciphertext
Known Plaintext	• Encryption algorithm • Ciphertext • One or more plaintext-ciphertext pairs formed with the secret key
Chosen Plaintext	• Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen Ciphertext	• Encryption algorithm • Ciphertext • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen Text	• Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

Cryptanalysis

- The last two (chosen ciphertext and chosen text) are less commonly employed as cryptanalytic techniques but are nevertheless possible avenues of attack.

Cryptanalysis

- Only a relatively weak algorithm will fail to withstand a ciphertext-only attack.
- Generally, an encryption algorithm is designed to withstand a known-plaintext attack.

Cryptanalysis

- An encryption scheme is **computationally secure** if ciphertext generated by the scheme meets one or both of the criteria:
- The cost of breaking the cipher exceeds the value of the encrypted information.

Cryptanalysis

- The time required to break the cipher exceeds the useful lifetime of the information.

Cryptanalysis

Brute Force attack:

- Involves trying every possible key until an intelligible translation of the ciphertext into plaintext is obtained
- On average, half of all possible keys must be tried to achieve success.

Cryptanalysis

End

- To supplement the brute-force approach
- Some degree of knowledge about the expected plaintext is needed.
- Some means of automatically distinguishing plaintext from garble is also needed.

Feistel Cipher Structure

Network Security

Feistel Cipher Structure

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain structure of Feistel Cipher.

Feistel Cipher Structure

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

Feistel Cipher Structure

- Many symmetric block encryption algorithms have a structure, which was first described by Horst Feistel of IBM in 1973.

Feistel Cipher Structure

Feistel Encryption

- The inputs to the encryption algorithm are a plaintext block of length 2w bits and a key K.
- The plaintext block is divided into two halves, LE_0 and RE_0.

Feistel Cipher Structure

- The two halves of the data pass through n rounds of processing and then combine to produce the ciphertext block.

Feistel Cipher Structure

- Each round *i* has as inputs LE_{i-1} and RE_{i-1} derived from the previous round, as well as a subkey K_i derived from the overall *K*.

Feistel Cipher Structure

- In general, the subkeys K_i are different from K and from each other and are generated from the key by a subkey generation algorithm.

Feistel Cipher Structure

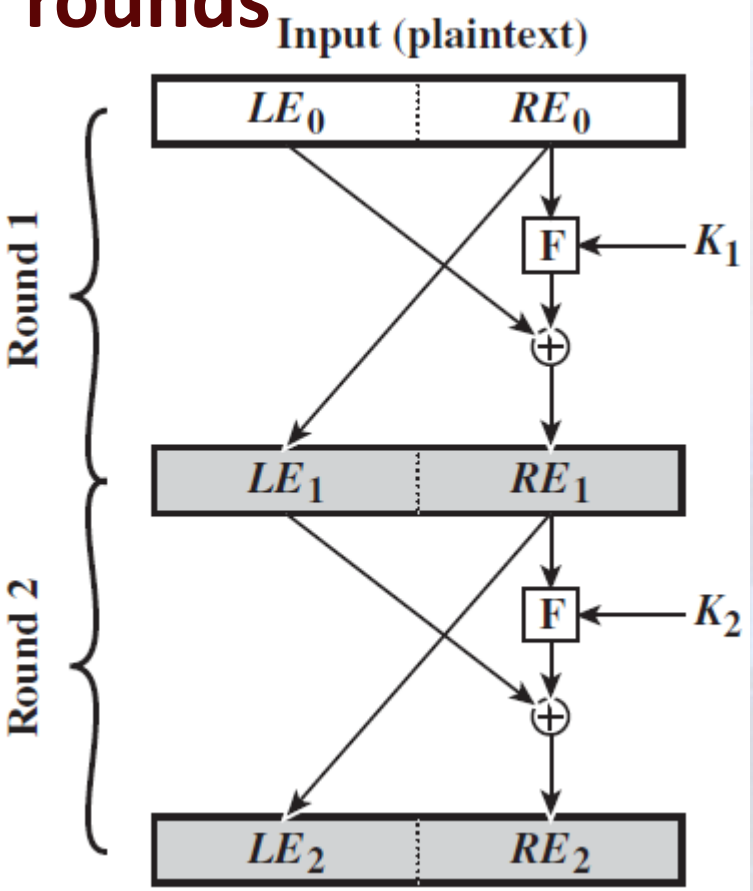
- In a given round, a substitution is performed on the left half of the data.
- Apply a round function F to the right half of the data and then take XOR of the output of that function and the left half of the data.

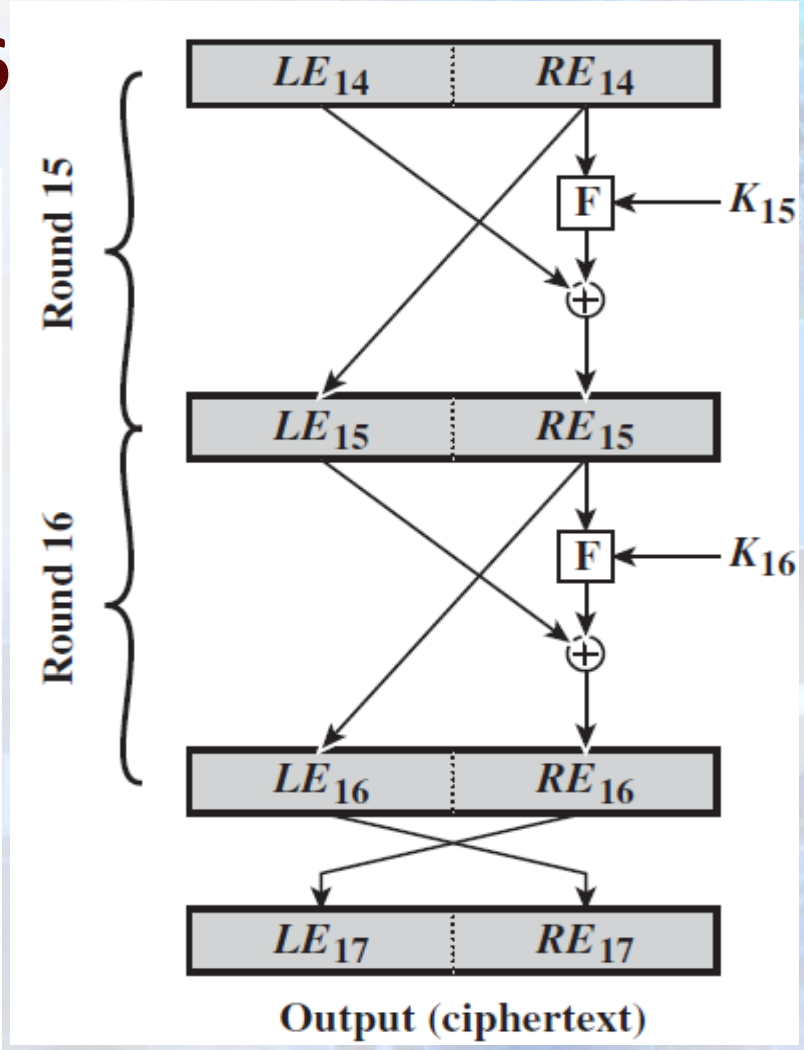
Feistel Cipher Structure

- The round function has the same general structure for each round but is parameterized by the round subkey K_i.
- A permutation is then performed to interchange the two halves of the data.

Feistel Cipher Structure

Feistel Encryption 16 rounds





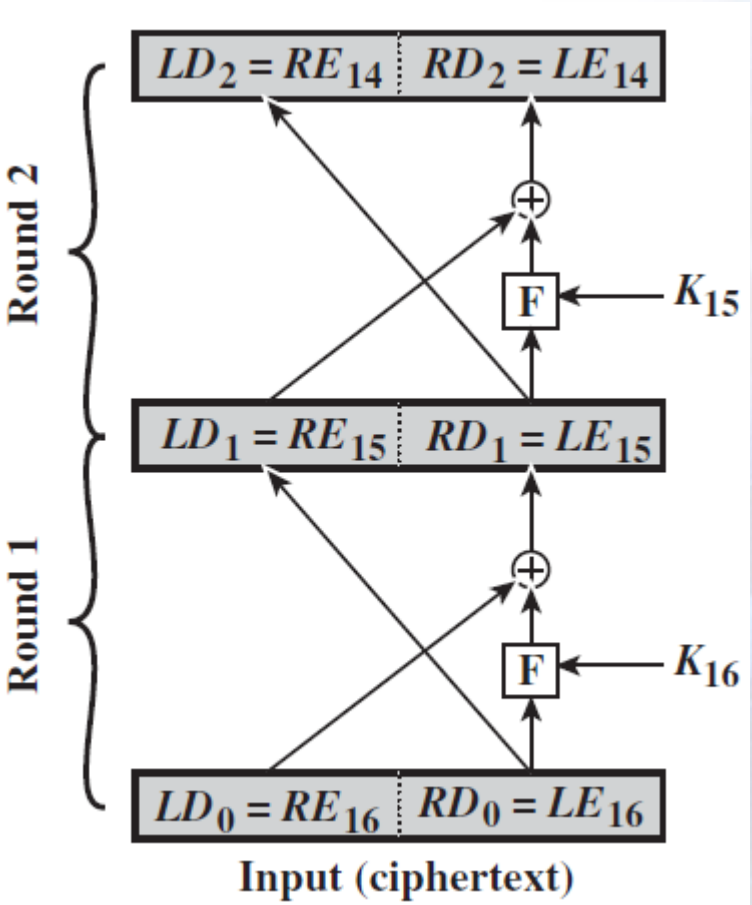
Feistel Cipher Structure

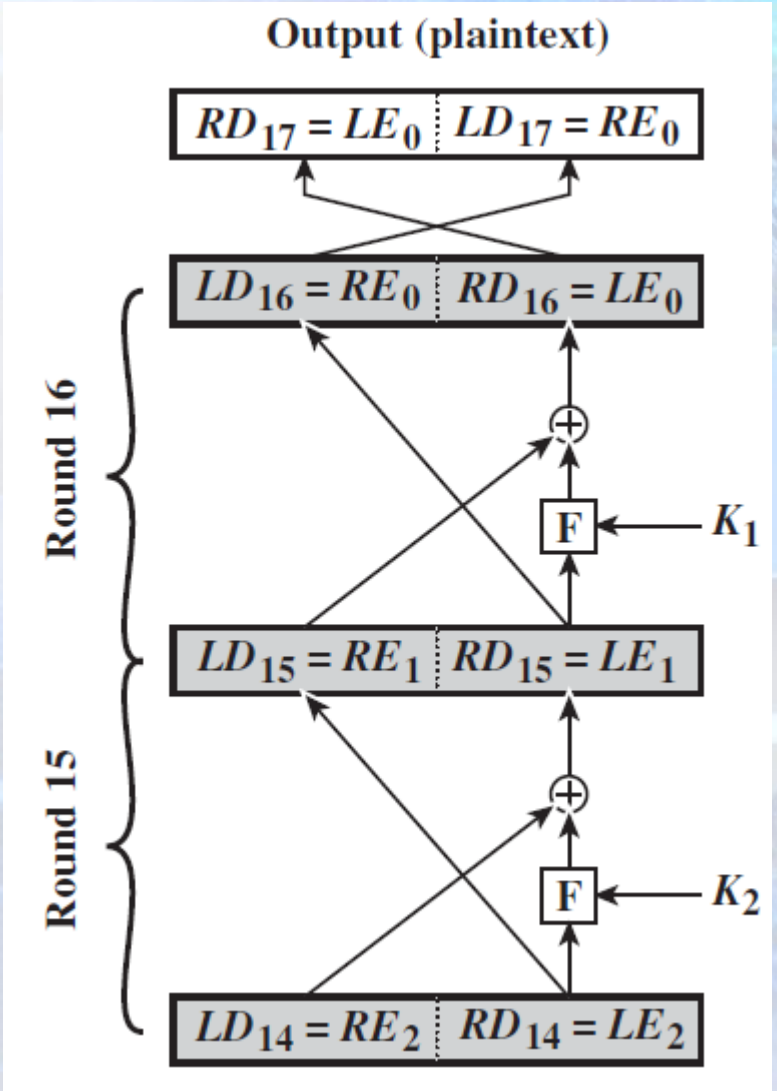
Feistel Decryption

- Use the ciphertext as input to the algorithm, but use the subkeys K_i in reverse order. That is, use K_n in the first round, K_{n-1} in the second round, and so on until K_1 is used in the last round.

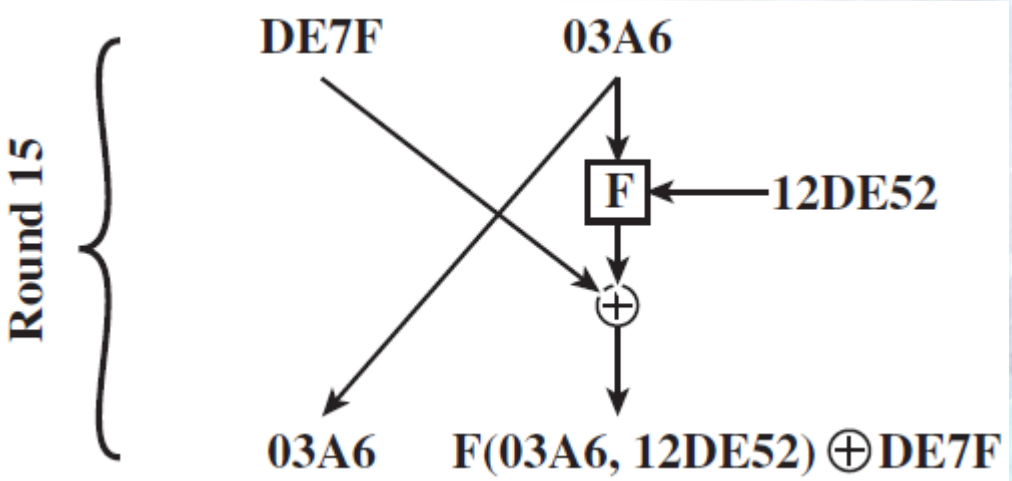
Feistel Cipher Structure

Feistel Decryption

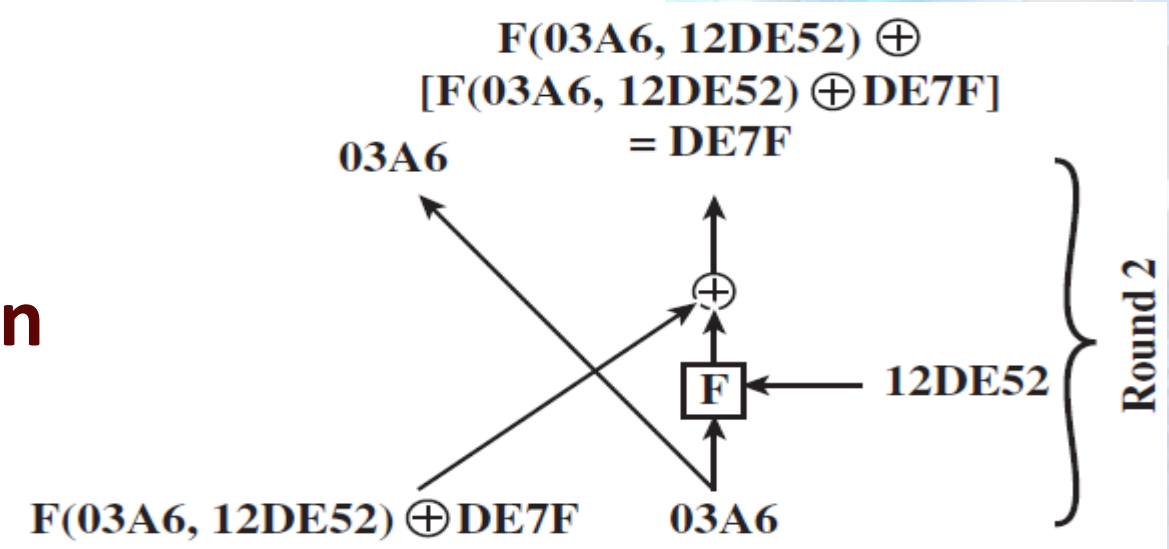




Feistel Cipher Structure



Encryption



Decryption

Feistel Cipher Structure

Feistel Cipher Design Features:

- **Block size:** Larger block sizes mean greater security (all other things being equal) but reduced encryption/decryption speed.

Feistel Cipher Structure

- **Key size:** Larger key size means greater security but may decrease encryption/decryption speed.

Feistel Cipher Structure

- **Number of rounds:** The essence of a symmetric block cipher is that a single round offers inadequate security but that multiple rounds offer increasing security.

Feistel Cipher Structure

- **Subkey generation algorithm:** Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.
- **Round function:** Greater complexity generally means greater resistance to cryptanalysis.

Feistel Cipher Structure

- **Fast software Algorithms:** Encryption is embedded in applications or utility functions in such a way as to preclude a hardware implementation. Thus, speed of execution of the algorithm becomes a concern.

Feistel Cipher Structure

End

- **Ease of analysis:** If the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities and develop a higher level of assurance as to its strength.

By-Pin2

Data Encryption Standard (DES)

Network Security

Data Encryption Standard (DES)

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe data encryption standard.

Data Encryption Standard (DES)

Figures and material in this topic have been

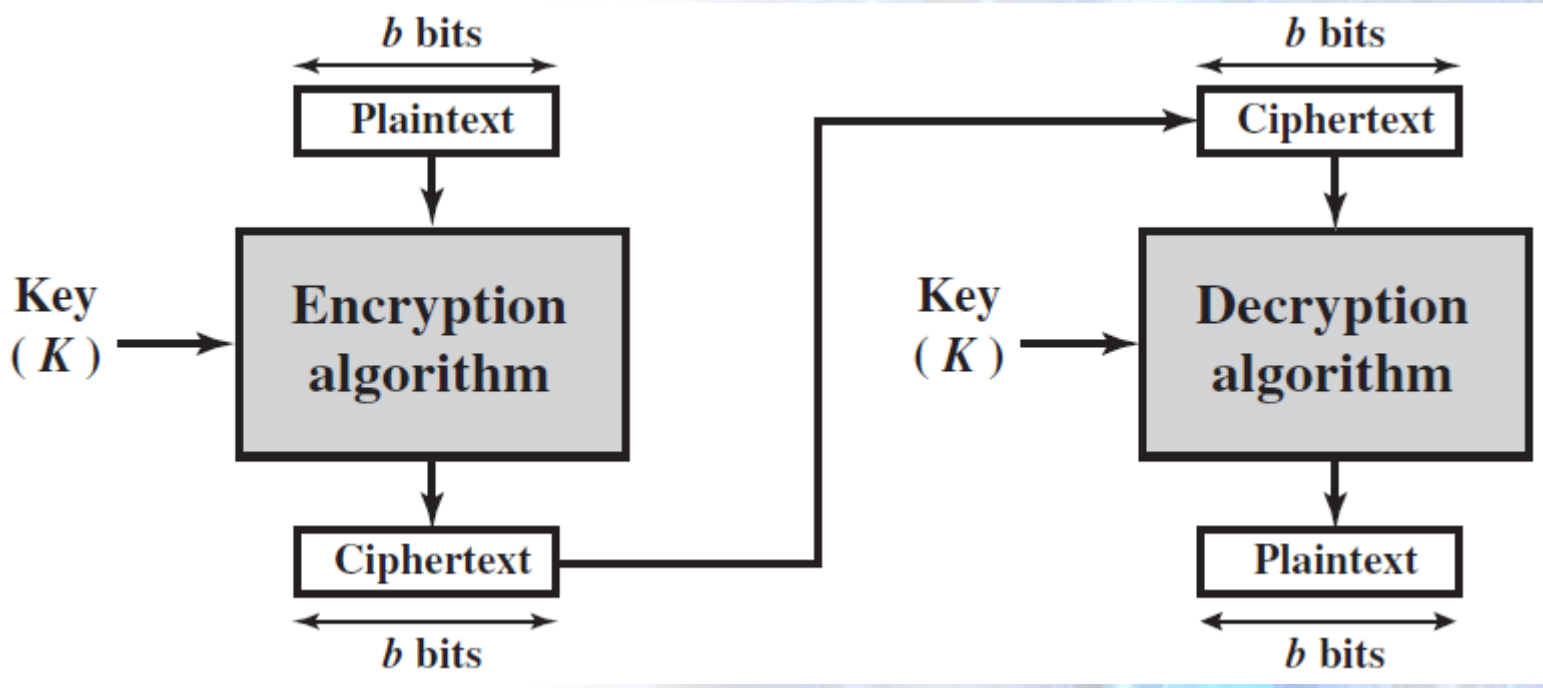
- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

Data Encryption Standard (DES)

- A block cipher processes plaintext input in fixed-sized blocks and produces a block of ciphertext of equal size for each plaintext block.
- The two users share a common encryption key.
- DES is an example.

Data Encryption Standard (DES)

Block Ciphers



Data Encryption Standard (DES)

- Data Encryption Standard (DES) was issued in 1977 as Federal Information Processing Standard 46 (FIPS 46) by the National Institute of Standards and Technology (NIST).

Data Encryption Standard (DES)

DES Encryption

- 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.

Data Encryption Standard (DES)

- There are two inputs to the encryption function: the plaintext to be encrypted and the key.
- The function expects a 64-bit key out of which only 56 are used; other 8 bits can be set arbitrarily.

Data Encryption Standard (DES)

- Plaintext proceeds in three phases.
- **First**, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the permuted input.

Data Encryption Standard (DES)

- The **2nd phase** consists of 16 rounds of the same function, which involves both permutation and substitution functions.
- The output of the last round consists of 64 bits that are a function of the input plaintext and the key.

Data Encryption Standard (DES)

- The left and right halves of the output are swapped to produce preoutput.
- **Finally**, the preoutput is passed through a permutation that is the inverse of the initial permutation function, to produce the 64-bit ciphertext.

Data Encryption Standard (DES)

Subkey Generation

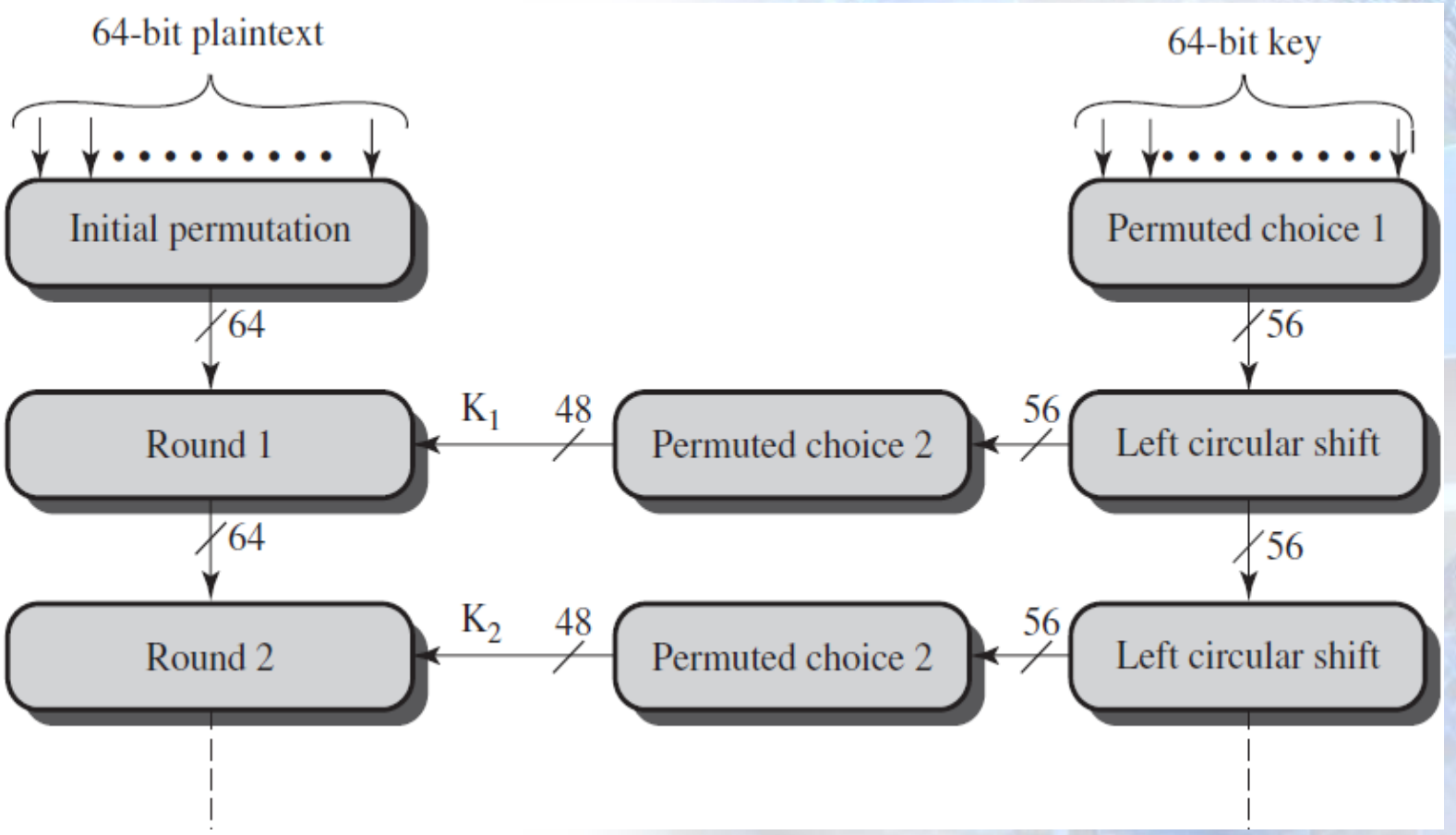
- Initially, the key is passed through a permutation function.
- Then, for each of the 16 rounds, a subkey (K_i) is produced by the combination of a left circular shift and a permutation.

Data Encryption Standard (DES)

- The permutation function is the same for each round, but a different subkey is produced because of the repeated shifts of the key bits.

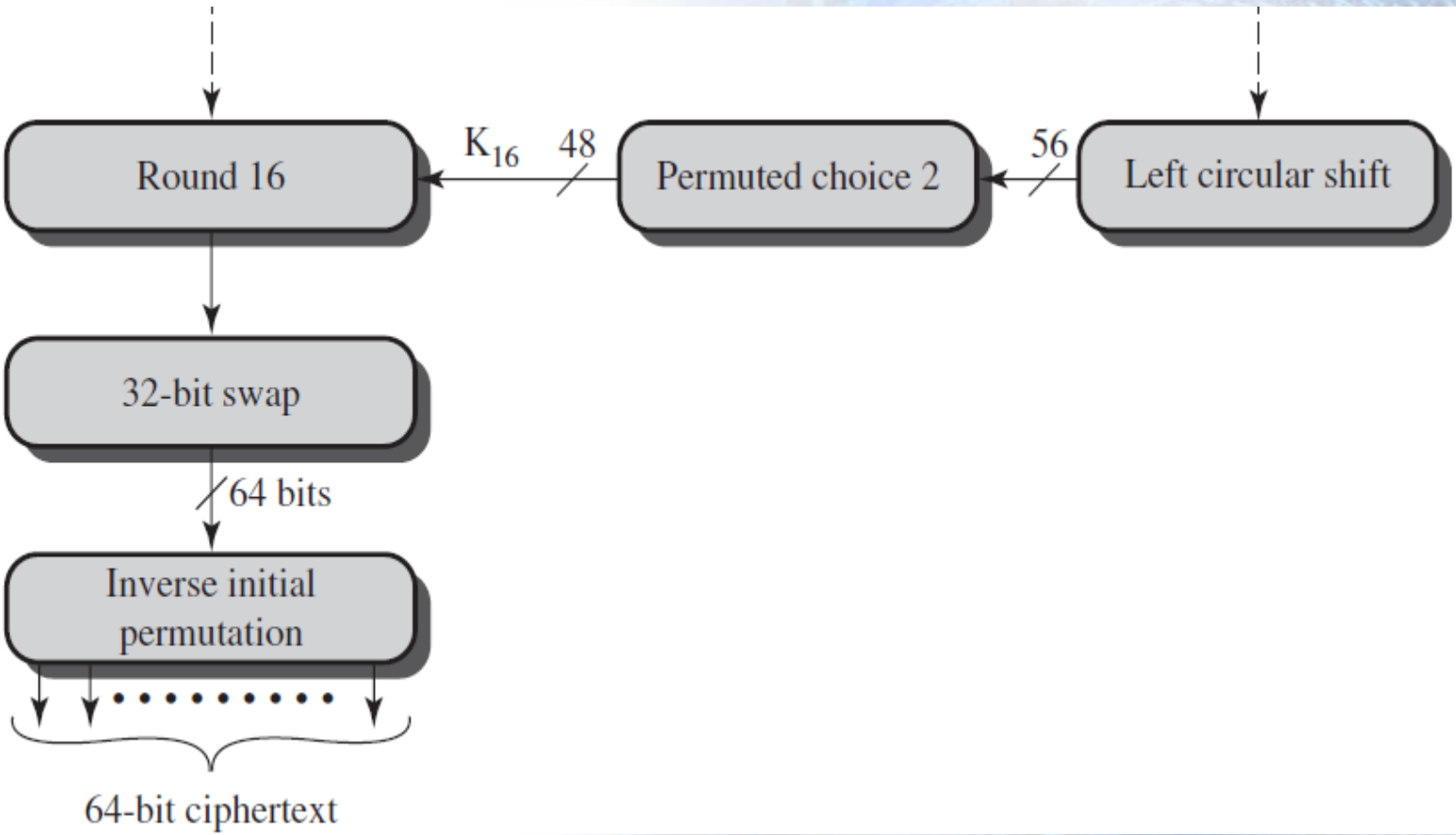
Data Encryption Standard (DES)

Working of DES



Data Encryption Standard (DES)

Working of DES



Data Encryption Standard (DES)

DES Decryption

- It uses the same algorithm as encryption, except that the application of the subkeys is reversed.
- Also, the initial and final permutations are reversed.

Data Encryption Standard (DES)

Concerns about DES

- 1.The algorithm itself
- Refers to the possibility that cryptanalysis is possible by exploiting the characteristics of the algorithm

Data Encryption Standard (DES)

- 2.The use of a 56-bit key
- $2^{56} = 7.2 \times 10^{16}$ keys
- Time required if PC works at 10^9 decryptions/s, then 2^{55} ns = 1.125 years.
- Time required if PC works at 10^{13} decryptions/s, then 1 hour.

Data Encryption Standard (DES)

- DES finally proved insecure in July 1998.
- Electronic Frontier Foundation (EFF) have broken it using a machine that took less than three days.

End

By-Pin2

Triple DES

Network Security

Triple DES

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe Triple DES.

Triple DES

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

Triple DES

- DES uses a 56-bit key
- $2^{56} = 7.2 \times 10^{16}$ keys
- Time required if PC works at 10^9 decryptions/s, then 2^{55} ns = 1.125 years.
- Time required if PC works at 10^{13} decryptions/s, then 1 hour.

Triple DES

- Given the potential vulnerability of DES to a brute-force attack, use of multiple encryption and multiple keys was suggested.
- Rationale was to preserve the existing investment in software, & hardware.

Triple DES

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

Triple DES

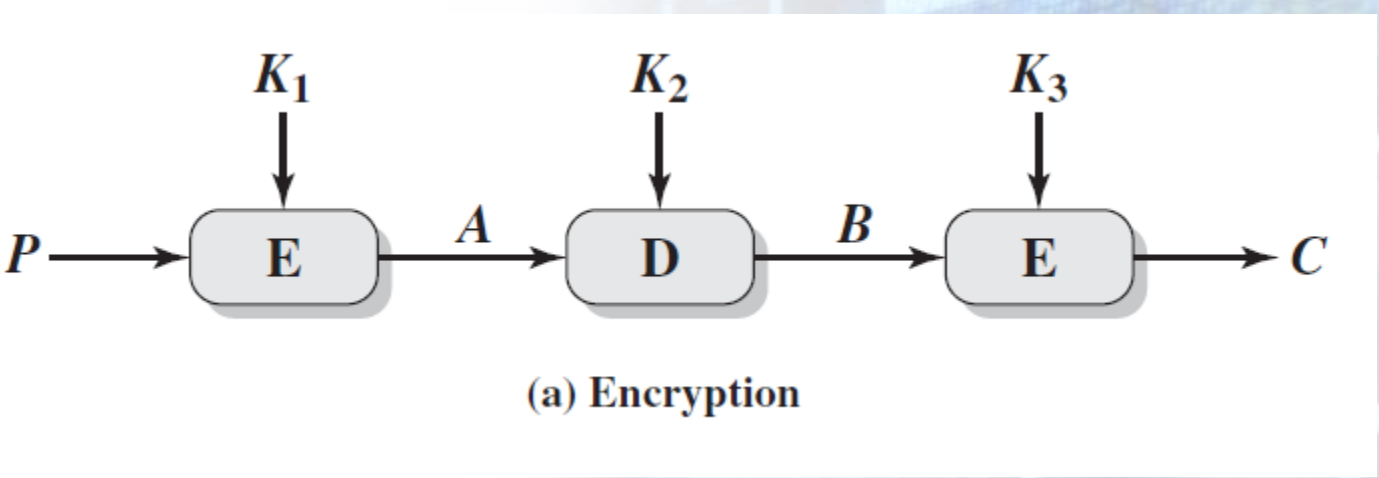
- Given a plaintext P, ciphertext C is generated as
- $C =$

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

- where E[K, X] encryption of X using key K
- D[K, Y] decryption of Y using key K

Triple DES

3DES Encryption



(a) Encryption

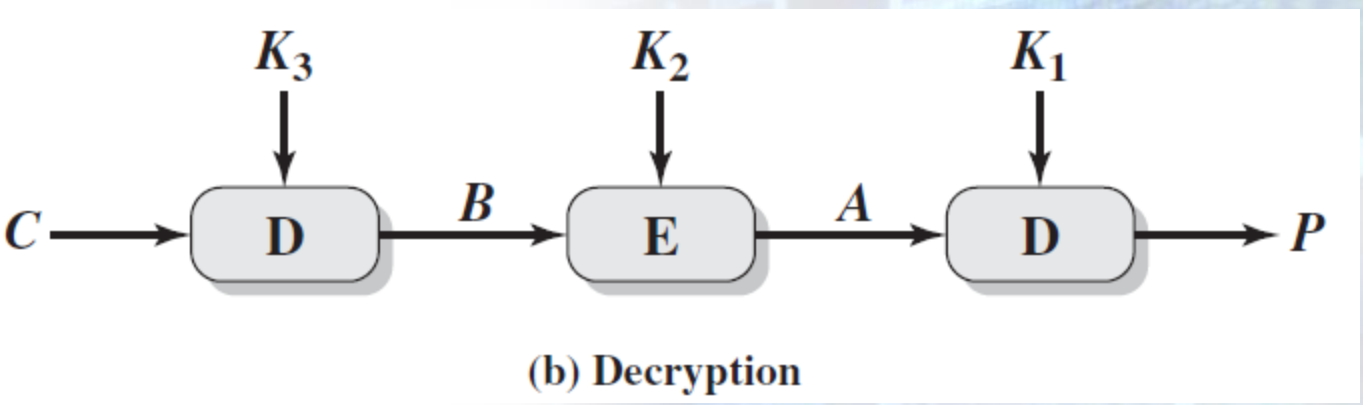
Triple DES

- Decryption is simply the same operation with the keys reversed:

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

Triple DES

3DES Decryption



(b) Decryption

Triple DES

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

Triple DES

- Its only advantage is that it allows users of 3DES to decrypt data encrypted by users of the older single DES:

- C =

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

  $= E[K, P]$

Triple DES

- Federal Information Processing Standards (FIPS) 46-3 also allows for the use of two keys, with $K_1 = K_3$; this provides for a key length of 112 bits.

Triple DES

- The cost of a brute-force key search on 3DES is on the order of $2^{112} = (5 * 10^{33})$.

Triple DES

- 3DES with two keys is a relatively popular alternative to DES and has been adopted for use in the key management standards ANSI X9.17 and ISO 8732.

Triple DES

Triple DES with Three Keys:

- Many researchers now feel that three-key 3DES is the preferred alternative.
- With three distinct keys, 3DES has an effective key length of 168 bits.

Triple DES

- $2^{168} = 3.7 \times 10^{50}$ keys
- Time required if PC works at 10^9 decryptions/s, then 2^{167} ns = 5.8 x 10^{33} years.
- Time required if PC works at 10^{13} decryptions/s, then 5.8 x 10^{29} years.

Triple DES

- Backward compatibility with DES is provided by putting $K_3 = K_2$ or $K_1 = K_2$.

Triple DES

Usage of 3DES:

- A number of Internet-based applications have adopted three-key 3DES:
- Pretty Good Privacy (PGP) and Secure/Multipurpose Internet Mail Extension (S/MIME).

Triple DES

FIPS 46-3 Guidelines for 3DES:

- 3DES is the approved symmetric encryption algorithm of choice.
- The original DES is permitted under the standard for legacy systems only; new procurements should support 3DES.

Triple DES

- Government organizations with legacy DES systems are encouraged to transition to 3DES.

End

By-Pin2

Advanced Encryption Standard

Network Security

Advanced Encryption Standard

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe Advanced Encryption Standard.

Advanced Encryption Standard

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

Advanced Encryption Standard

- The principal drawbacks of 3DES:
- 1.It has three times as many rounds as DEA and is correspondingly slower.
- 2.Both DEA and 3DES use a 64-bit block size.
- Its not a reasonable candidate for long term use.

Advanced Encryption Standard

- In 1997 NIST issued a call for proposals for a new AES:
- 1.Should have a security strength equal to or better than 3DES and significantly improved efficiency.

Advanced Encryption Standard

- 2. Must be a symmetric block cipher with a block length of 128 bits and support for key lengths of 128, 192, and 256 bits.

Advanced Encryption Standard

- 3. Evaluation criteria included security, computational efficiency, memory requirements, hardware and software suitability, and flexibility

Advanced Encryption Standard

- NIST selected Rijndael as the proposed AES algorithm
- Developers were two cryptographers from Belgium: Dr. Joan Daemen and Dr. Vincent Rijmen
- published as a final standard (FIPS PUB 197) in 2001.

Advanced Encryption Standard

- AES uses a block length of 128 bits and a key length that can be 128, 192, or 256 bits.
- For our discussion, we assume 128 bits in this topic.

Advanced Encryption Standard

- The input to the encryption and decryption algorithms is a single 128-bit block.
- In FIPS PUB 197, this block is depicted as a square matrix of bytes.

Advanced Encryption Standard

- The block is copied into the **State** array, which is modified at each stage of encryption or decryption.
- After the final stage, **State** is copied to an output matrix.

Advanced Encryption Standard

- Similarly, the 128-bit key is depicted as a square matrix of bytes.
- This key is then expanded into an array of key schedule words: Each word is four bytes and total key schedule is 44 words for 128-bit key.

Advanced Encryption Standard

- Ordering of bytes in a matrix is by column.
- First four bytes of a 128-bit plaintext input to the encryption cipher occupy the first column of the **in** matrix, the second four bytes occupy the second column, and so on.

Advanced Encryption Standard

- Similarly, the first four bytes of the expanded key, which form a word, occupy the first column of the **w** matrix.

Advanced Encryption Standard

AES's Working:

- Four different stages are used, one of permutation and three of substitution

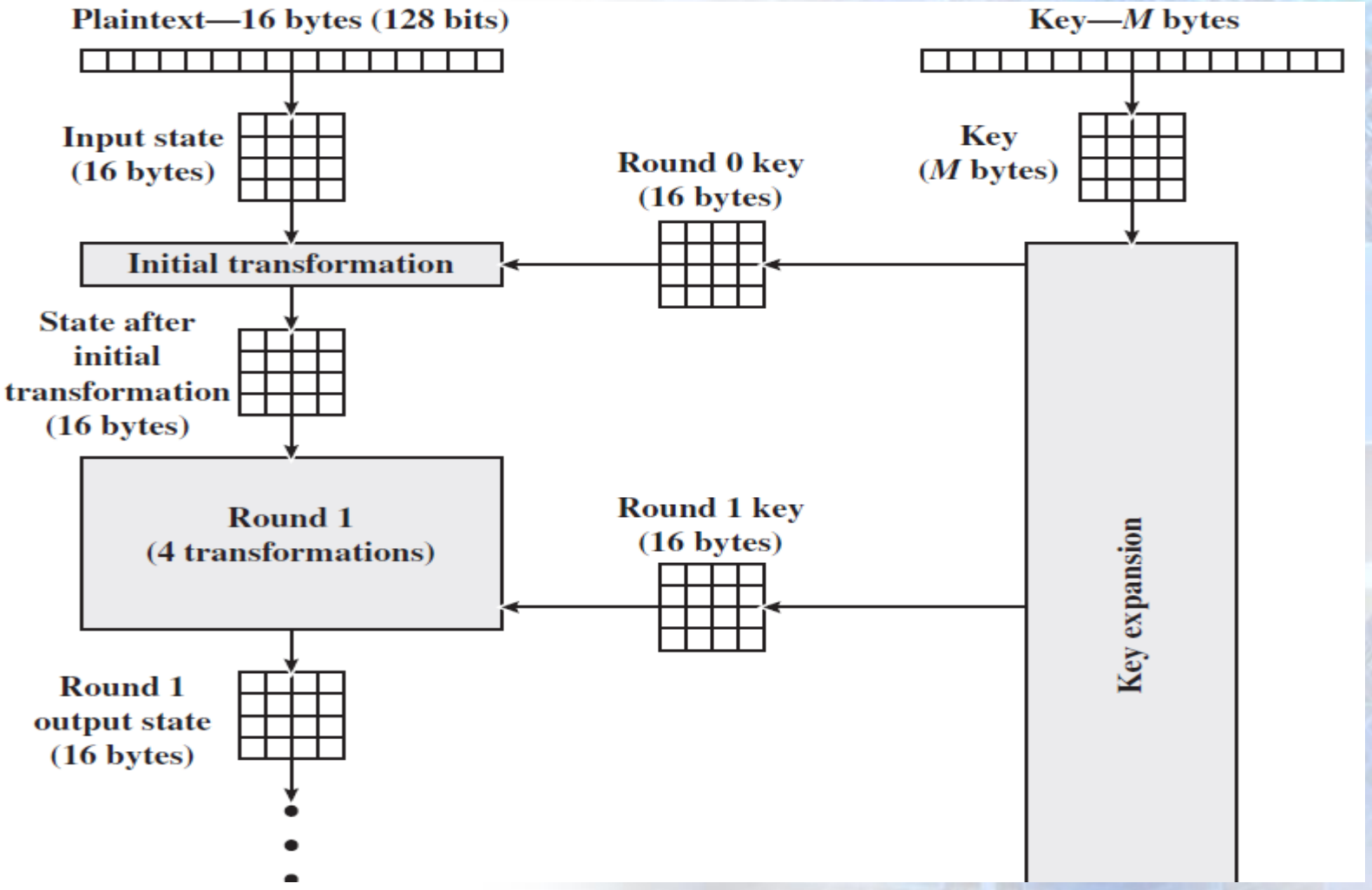
Advanced Encryption Standard

- **Substitute bytes:** Uses a table, referred to as an S-box, to perform a byte-by-byte substitution of the block.
- **Shift rows:** A simple permutation that is performed row by row.

Advanced Encryption Standard

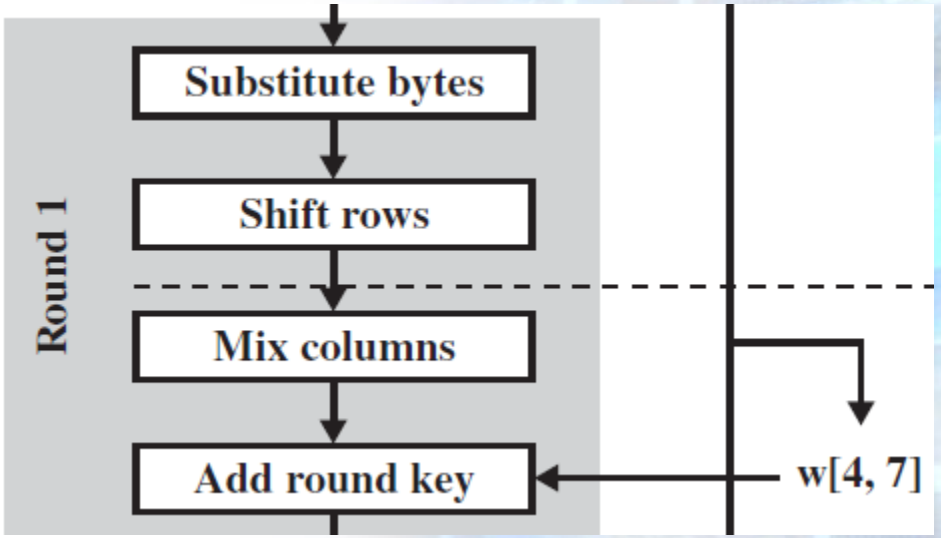
- **Mix columns:** A substitution that alters each byte in a column as a function of all of the bytes in the column.
- **Add round key:** A simple bitwise XOR of the current block with a portion of the expanded key.

Advanced Encryption Standard

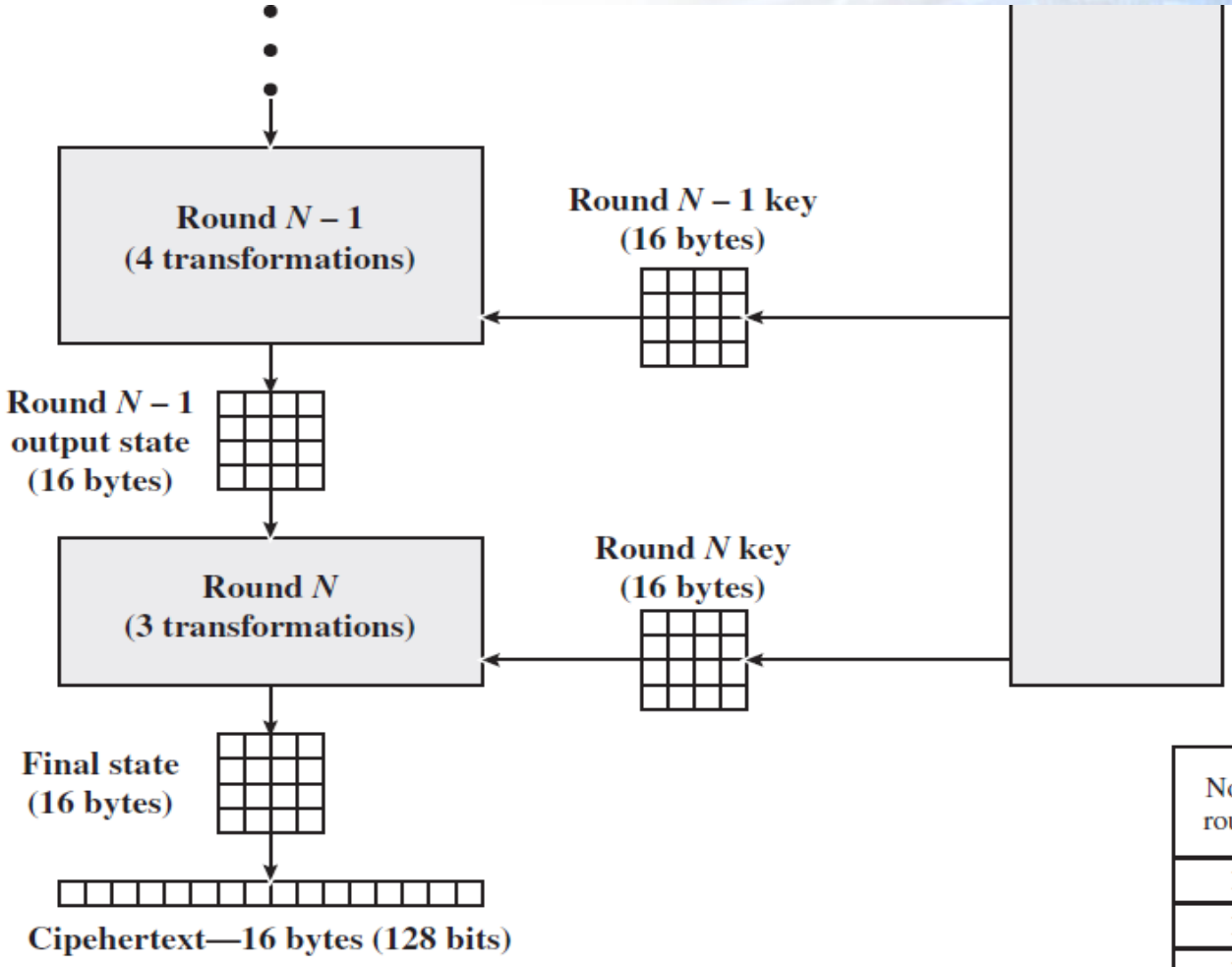


Advanced Encryption Standard

Internal Details of a round.



Advanced Encryption Standard



No. of rounds	Key Length (bytes)
10	16
12	24
14	32

Advanced Encryption Standard

Some comments:

- AES structure is not a Feistel structure.
- For both encryption and decryption, the cipher begins with an Add Round Key stage, followed by nine rounds that each includes all four stages

Advanced Encryption Standard

- Each stage is easily reversible.
- The final round of both encryption and decryption consists of only three stages.
- The decryption algorithm is not identical to the encryption algorithm.

Advanced Encryption Standard

- The decryption algorithm makes use of the expanded key in reverse order.

End

The Use of Random Numbers

Network Security

The Use of Random Numbers

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain use of random numbers.

The Use of Random Numbers

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

The Use of Random Numbers

- A number of network security algorithms based on cryptography make use of random numbers.

The Use of Random Numbers

Random Numbers

The Use of Random Numbers

- **Examples:**
- Generation of keys for the RSA public-key encryption algorithm and other public-key algorithms.

The Use of Random Numbers

- Generation of a symmetric key for use as a temporary session key; used in a number of networking applications such as Transport Layer Security, Wi-Fi, e-mail security, and IP security.

The Use of Random Numbers

- In a number of key distribution scenarios, such as Kerberos, random numbers are used for handshaking to prevent replay attacks.

The Use of Random Numbers

Requirements

- Two distinct and not necessarily compatible requirements for a sequence of random numbers are:
- Randomness
- Unpredictability

The Use of Random Numbers

Randomness

- The concern in the generation of a sequence of allegedly random numbers has been that the sequence of numbers be random in some well defined statistical sense.

The Use of Random Numbers

- The following criteria are used to validate that a sequence of numbers is random.

The Use of Random Numbers

- **Uniform distribution:** The distribution of bits in the sequence should be uniform; that is, the frequency of occurrence of ones and zeros should be approximately the same.

The Use of Random Numbers

- **Independence:**
- No one subsequence in the sequence can be inferred from the others.

The Use of Random Numbers

- There are well-defined tests for determining that a sequence of numbers matches a particular distribution, such as the uniform distribution.
- There is no such test to “prove” independence.

The Use of Random Numbers

- A number of tests can be applied to demonstrate if a sequence does not exhibit independence.
- The general strategy is to apply a number of such tests until the confidence that independence exists is sufficiently strong.

The Use of Random Numbers

Unpredictability

- In some applications, the requirement is not much that the sequence of numbers be statistically random but that the successive members of the sequence are unpredictable.

The Use of Random Numbers

- E.g. reciprocal authentication and session key generation.

The Use of Random Numbers

- With “true” random sequences, each number is statistically independent of other numbers in the sequence and therefore unpredictable.

The Use of Random Numbers

- Care must be taken that an opponent not be able to predict future elements of the sequence on the basis of earlier elements.

End

Pseudorandom Numbers

Network Security

Pseudorandom Numbers

Objectives of the Topic

- After completing this topic, a student will be able to
  - understand pseudorandom numbers.

Pseudorandom Numbers

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

Pseudorandom Numbers

- For cryptographic applications, algorithmic techniques for random number generation are deterministic and therefore produce sequences of numbers that are not statistically random.

Pseudorandom Numbers

- If the algorithm is good, the resulting sequences will pass many reasonable tests of randomness.
- Such numbers are referred to as **pseudorandom numbers**.

Pseudorandom Numbers

- You may be somewhat uneasy about the concept of using numbers generated by a deterministic algorithm as if they were random numbers.
- it generally works.

Pseudorandom Numbers

- Under most circumstances, pseudorandom numbers will perform as well as if they were random for a given use.

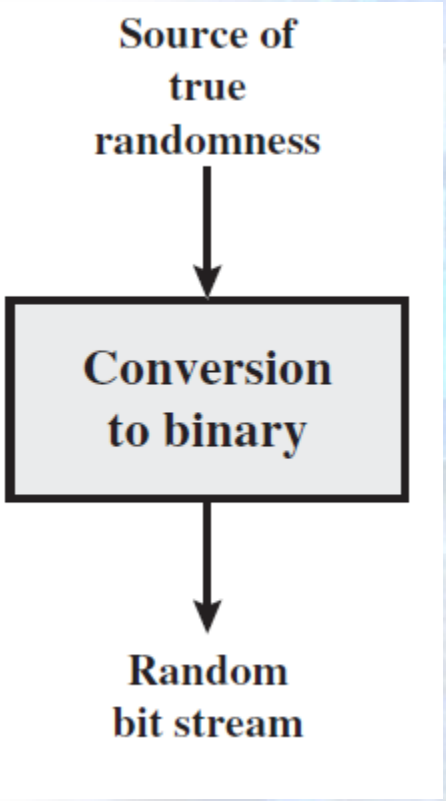
Pseudorandom Numbers

True Random Number Generator (TRNG)

- takes as input a source that is effectively random.
- the source is often referred to as an entropy source .

Pseudorandom Numbers

TRNG



Pseudorandom Numbers

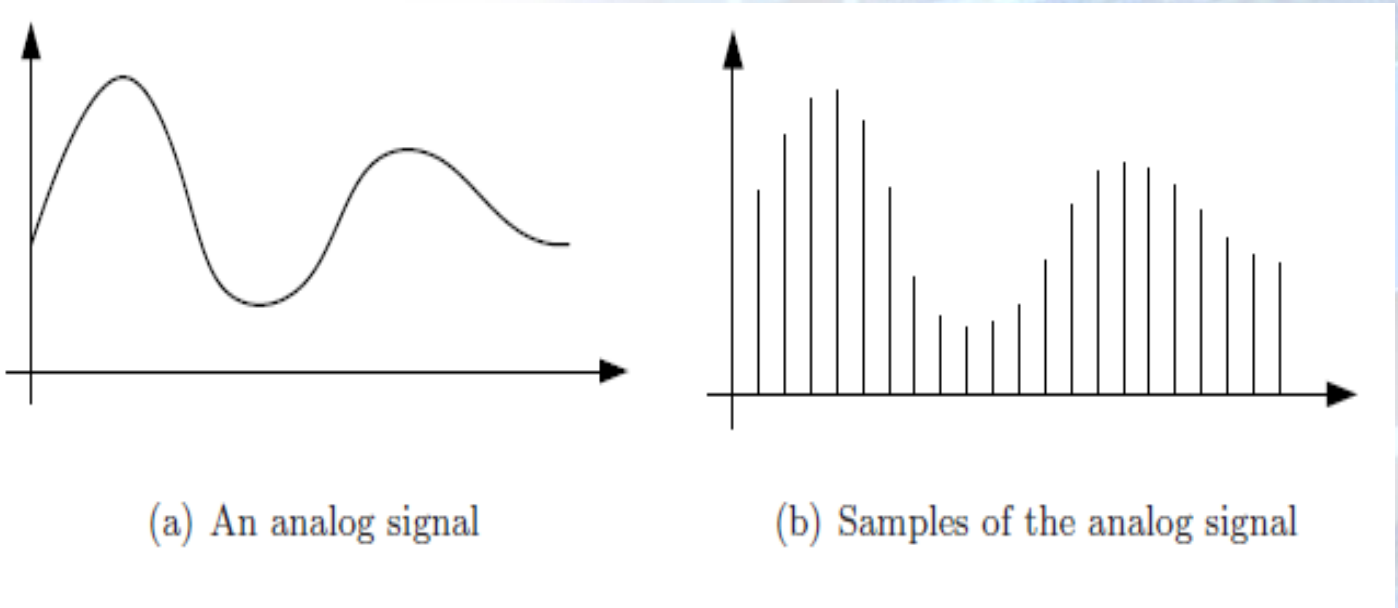
- The entropy source is drawn from physical environment of the computer and could include keystroke timing patterns, disk electrical activity, mouse movements, and instantaneous values of the system clock.

Pseudorandom Numbers

- The source, or combination of sources, serves as input to an algorithm that produces random binary output.
- The TRNG may simply involve conversion of an analog source to a binary output.

Pseudorandom Numbers

TRNG



(a) An analog signal

(b) Samples of the analog signal

Pseudorandom Numbers

Pseudorandom Number Generator (PRNG)

- takes as input a fixed value, called the seed, and produces a sequence of output bits using a deterministic algorithm.

Pseudorandom Numbers

- There is a feedback path by which some of the output are fed back as input.
- The output bit stream is determined solely by the input value, so that an adversary who knows the algorithm and the seed can reproduce bit stream.

Pseudorandom Numbers

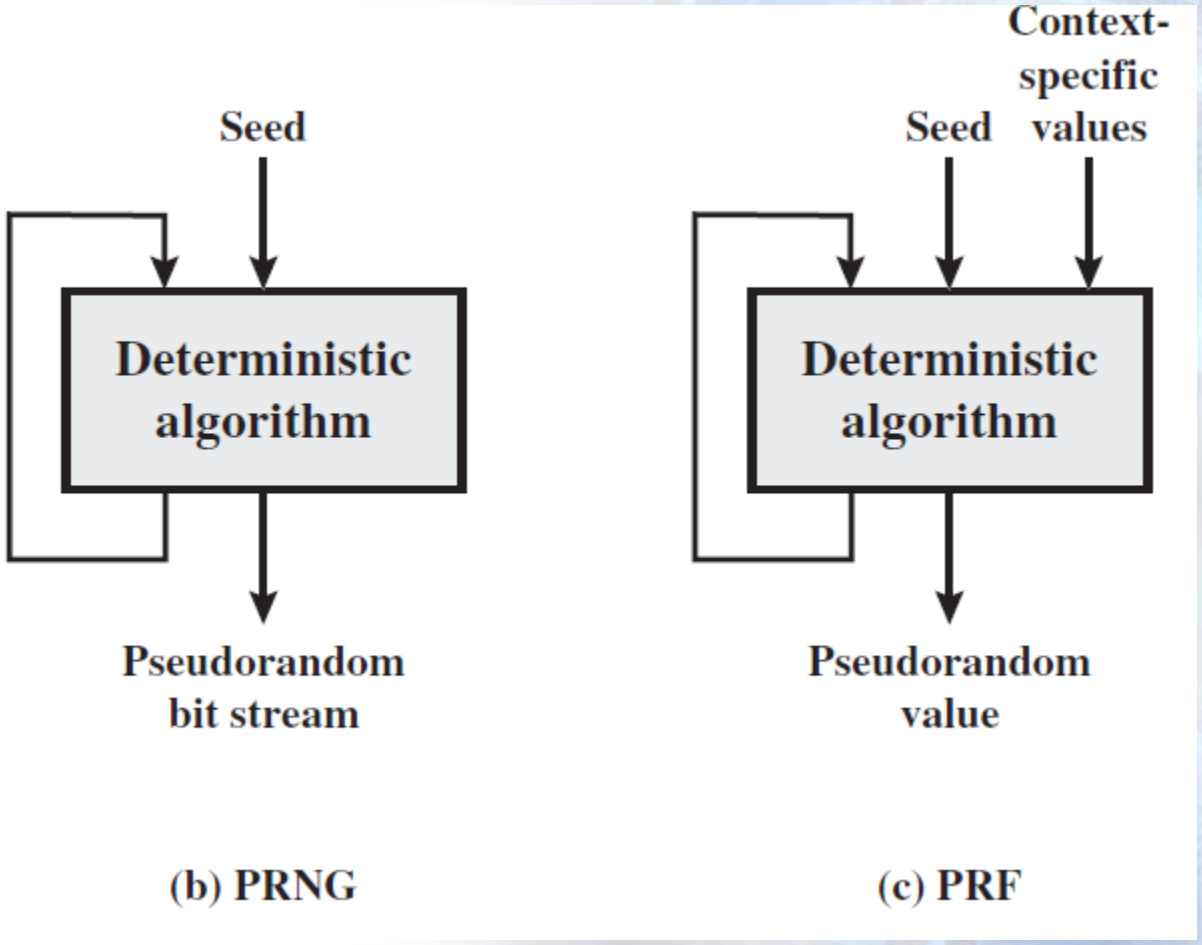
- Two different forms of PRNGs, based on application.
- **PRNG:** An algorithm used to produce an open-ended sequence of bits is referred to as a PRNG.
- App: input to a symmetric stream cipher.

Pseudorandom Numbers

- **Pseudorandom function (PRF):** produces a pseudorandom string of bits of some fixed length and takes as input seed plus some context values (a user or application ID).
- App: symmetric encrypt. keys, nonces.

Pseudorandom Numbers

PRNG and PRF



Pseudorandom Numbers

- Only difference between a PRNG and a PRF is the number of bits produced.
- The same algorithms can be used in both applications.
- Both require a seed and both must exhibit randomness and unpredictability.

Pseudorandom Numbers

- Cryptographic PRNGs have been the subject of much research over the years, and a wide variety of algorithms have been developed.
- These fall roughly into two categories:

Pseudorandom Numbers

Purpose-built algorithms

- Designed specifically and solely for the purpose of generating pseudorandom bit streams

Algorithms based on existing cryptographic algorithms

- Cryptographic algorithms have the effect of randomizing input
- Can serve as the core of PRNGs

Three broad categories of cryptographic algorithms are commonly used to create PRNGs:

- Symmetric block ciphers
- Asymmetric ciphers
- Hash functions and message authentication codes

Stream Cipher Structure

Network Security

Stream Cipher Structure

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe the basic structure of stream ciphers.

Stream Cipher Structure

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

Stream Cipher Structure

- A stream cipher processes the input elements continuously, producing output one element at a time as it goes along.

Stream Cipher Structure

Stream Cipher Structure

- A typical stream cipher encrypts plaintext one byte at a time, although a stream cipher may be designed to operate on one bit at a time or on units larger than a byte at a time.

Stream Cipher Structure

- In a stream cipher structure, a key is input to a pseudorandom bit generator that produces a stream of 8-bit numbers that are apparently random.

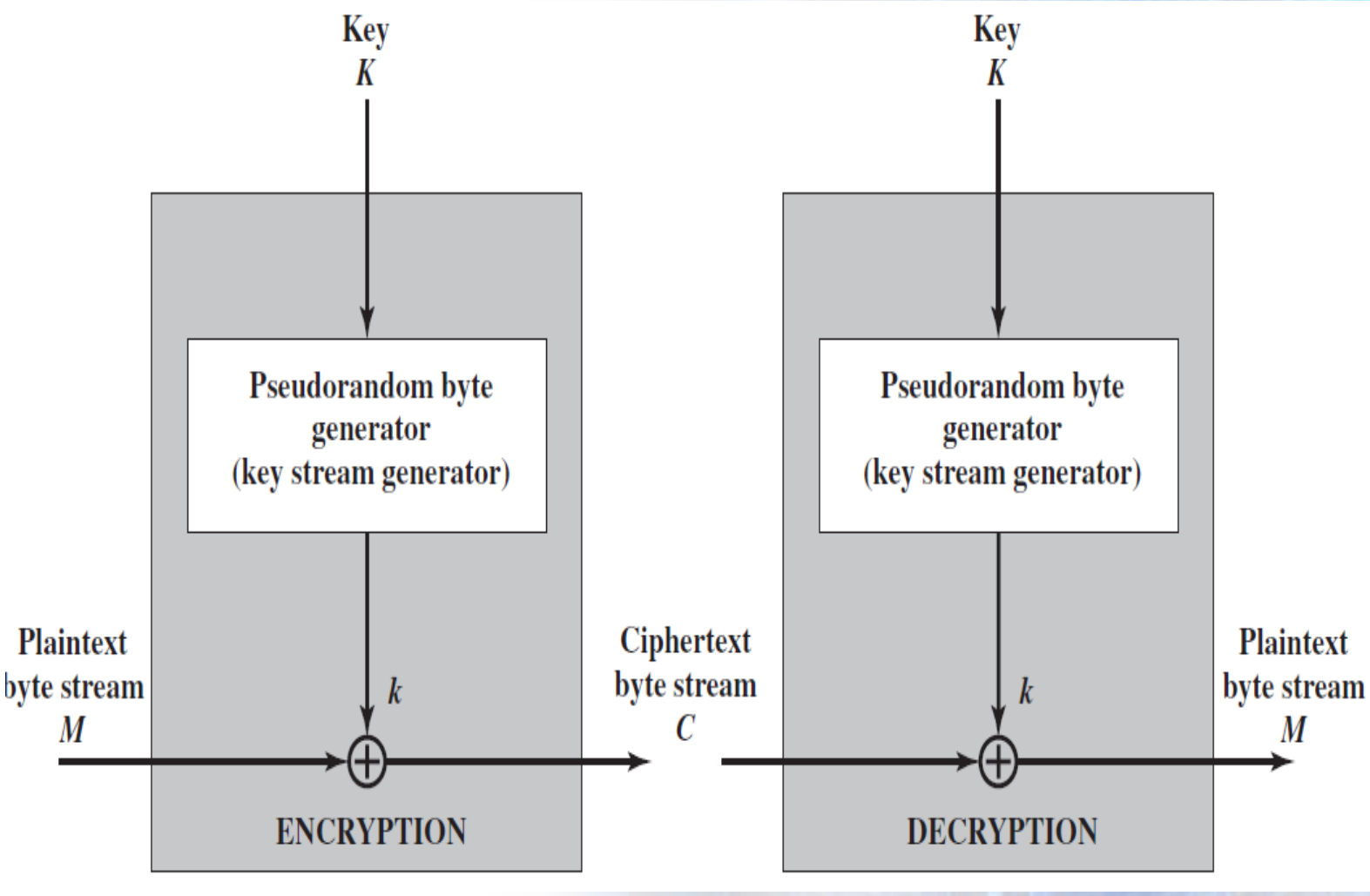
Stream Cipher Structure

- A pseudorandom stream is one that is unpredictable without knowledge of the input key and which has an apparently random character.

Stream Cipher Structure

- The output of the generator called a **keystream**, is combined one byte at a time with the plaintext stream using the bitwise exclusive-OR (XOR) operation.

Stream Cipher Structure



Stream Cipher Structure

Example

- if the next byte generated by the generator is 01101100 and the next plaintext byte is 11001100, then the resulting ciphertext byte is

```
  11001100  plaintext
⊕ 01101100  key stream
  10100000  ciphertext
```

Stream Cipher Structure

- Decryption requires the use of the same pseudorandom sequence

```
  10100000   ciphertext
⊕ 01101100   key stream
  11001100   plaintext
```

Stream Cipher Structure

Stream Cipher design considerations:

- 1. The encryption sequence should have a large period.

Stream Cipher Structure

- A pseudorandom number generator uses a function that produces a deterministic stream of bits that eventually repeats.
- The longer the period of repeat, the more difficult it will be to do cryptanalysis.

Stream Cipher Structure

- 2. The keystream should approximate the properties of a true random number stream as close as possible.

Stream Cipher Structure

- There should be an approximately equal number of 1s and 0s.
- If the keystream is treated as a stream of bytes, then all of the 256 possible byte values should appear approximately equally often.

Stream Cipher Structure

- The more random-appearing the keystream is, the more randomized the ciphertext is, making cryptanalysis more difficult.

Stream Cipher Structure

- 3. As the output of the pseudorandom number generator is conditioned on the value of the input key, to guard against brute-force attacks, the key needs to be sufficiently long.

Stream Cipher Structure

- With the current technology, a key length of at least 128 bits is desirable.
- The primary advantage of a stream cipher is that stream ciphers are almost always faster and use far less code than do block ciphers.

Stream Cipher Structure

- The advantage of a block cipher is that you can reuse keys.
- If two plaintexts are encrypted with the same key using a stream cipher, then cryptanalysis is often quite simple.

Stream Cipher Structure

End

- For applications that deal with stream of data, a stream cipher is preferred.
- For applications that deal with blocks of data (file transfer, e-mail), block ciphers may be more appropriate.

The RC4 Algorithm

Network Security

The RC4 Algorithm

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain working of RC4 algorithm.

The RC4 Algorithm

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

The RC4 Algorithm

- RC4 is a stream cipher designed in 1987 by Ron Rivest for RSA Security.
- It is a variable key-size stream cipher with byte-oriented operations.

The RC4 Algorithm

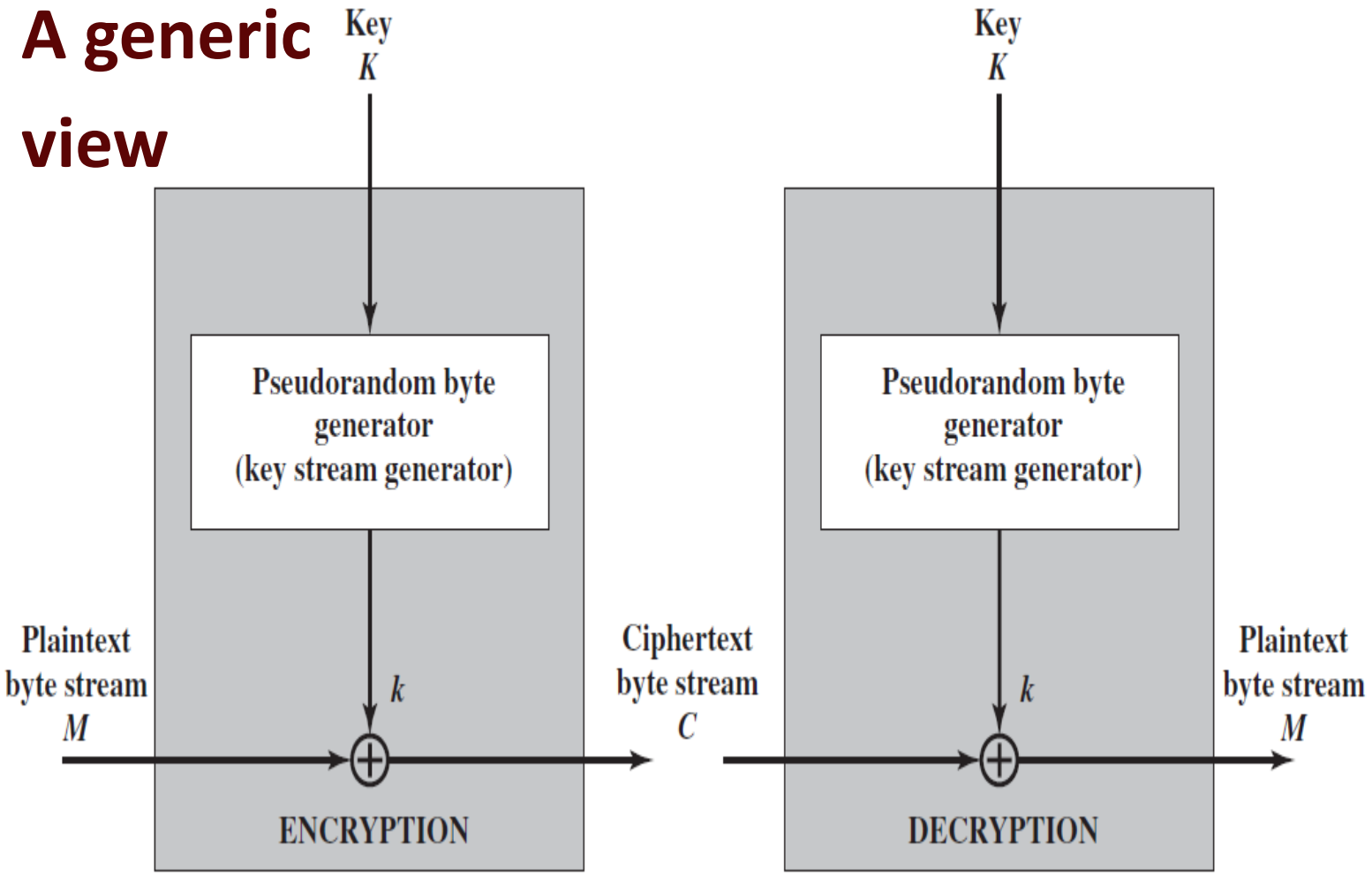
- RC4 is used in the Secure Sockets Layer/Transport Layer Security (SSL/TLS) standards that have been defined for communication between Web browsers and servers.

The RC4 Algorithm

- Also used in the Wired Equivalent Privacy (WEP) protocol and the newer WiFi Protected Access (WPA) protocol that are part of the IEEE 802.11 wireless LAN standard.

The RC4 Algorithm

A generic view



The RC4 Algorithm

- The RC4 algorithm is remarkably simple.
- A variable-length key of from 1 to 256 bytes (8 to 2048 bits) is used to initialize a 256-byte state vector S, with elements S[0], S[1], . . . , S[255].

The RC4 Algorithm

- For encryption and decryption, a byte k is generated from S by selecting one of the 255 entries in a systematic fashion.
- As each value of k is generated, the entries in S are once again permuted.

The RC4 Algorithm

Initialization of S:

- entries of S are set equal to the values from 0 through 255 in ascending order

```
for i = 0 to 255 do
S[i] = i;
T[i] = K[i mod keylen];
```

- Where T is a temporary vector.

The RC4 Algorithm

```
for i = 0 to 255 do
S[i] = i;
T[i] = K[i mod keylen];
```

- If the length of the key K is 256 bytes, then K is transferred to T.
- Otherwise, first keylen elements of T are copied from K, and then K is repeated as many times as necessary to fill out T.

The RC4 Algorithm

Next we use T to produce the initial permutation of S.

```
j = 0;
for i = 0 to 255 do
  j = (j + S[i] + T[i]) mod 256;
  Swap (S[i], S[j]);
```

The RC4 Algorithm

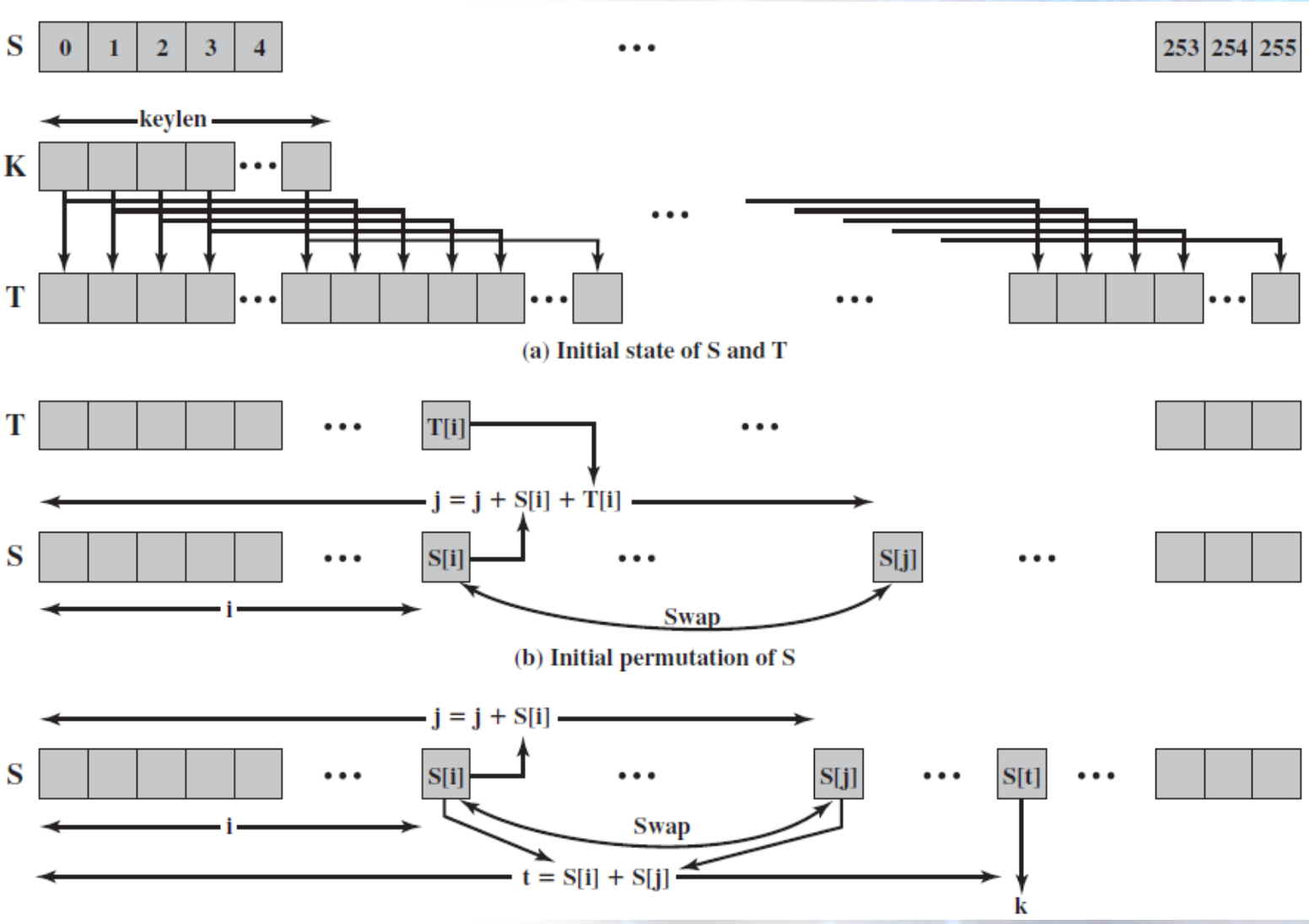
Stream Generation: Once the S vector is initialized, the input key is no longer used.

```
i, j = 0;
while (true)
  i = (i + 1) mod 256;
  j = (j + S[i]) mod 256;
  Swap (S[i], S[j]);
  t = (S[i] + S[j]) mod 256;
  k = S[t];
```

The RC4 Algorithm

- To encrypt, XOR the value k with the next byte of plaintext.
- To decrypt, XOR the value k with the next byte of ciphertext.

The RC4 Algorithm



The RC4 Algorithm

End

Strength of RC4:

- A number of papers have been published analyzing methods of attacking RC4.
- None of these approaches is practical against RC4 with a reasonable key length, such as 128 bits.

Elect. Codebook, Cipher Block Chaining

Network Security

Elect. Codebook, Cipher Block Chaining

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe electronic codebook and cipher block modes.

Elect. Codebook, Cipher Block Chaining

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

Elect. Codebook, Cipher Block Chaining

- A symmetric block cipher processes one block of data at a time.
- Block length is 64 bits for DES and 3DES
- For AES, the block length is 128 bits.

Elect. Codebook, Cipher Block Chaining

- If the amount of plaintext is greater than **b-**bits, then we can break the plaintext up into **b**-bit blocks.
- When multiple blocks of plaintext are encrypted using the same key, a number of security issues arise.

Elect. Codebook, Cipher Block Chaining

- Five modes of operation have been defined by NIST (SP(Special Publication) 800- 38A) so that a block cipher can be applied in a variety of applications.

Elect. Codebook, Cipher Block Chaining

- A mode of operation is a technique for enhancing the effect of a cryptographic algorithm or adapting the algorithm for an application, such as applying a block cipher to a sequence of data blocks or a data stream.

Elect. Codebook, Cipher Block Chaining

- Electronic Codebook Mode (ECB)
- Cipher Block Chaining Mode (CBC)
- Cipher Feedback Mode (CFB)
- Output Feedback (OFB)
- Counter Mode (CTR)

Elect. Codebook, Cipher Block Chaining

Electronic Codebook Mode:

- Simplest mode
- Plaintext is handled **b** bits at a time and each block of plaintext is encrypted using the same key.

Elect. Codebook, Cipher Block Chaining

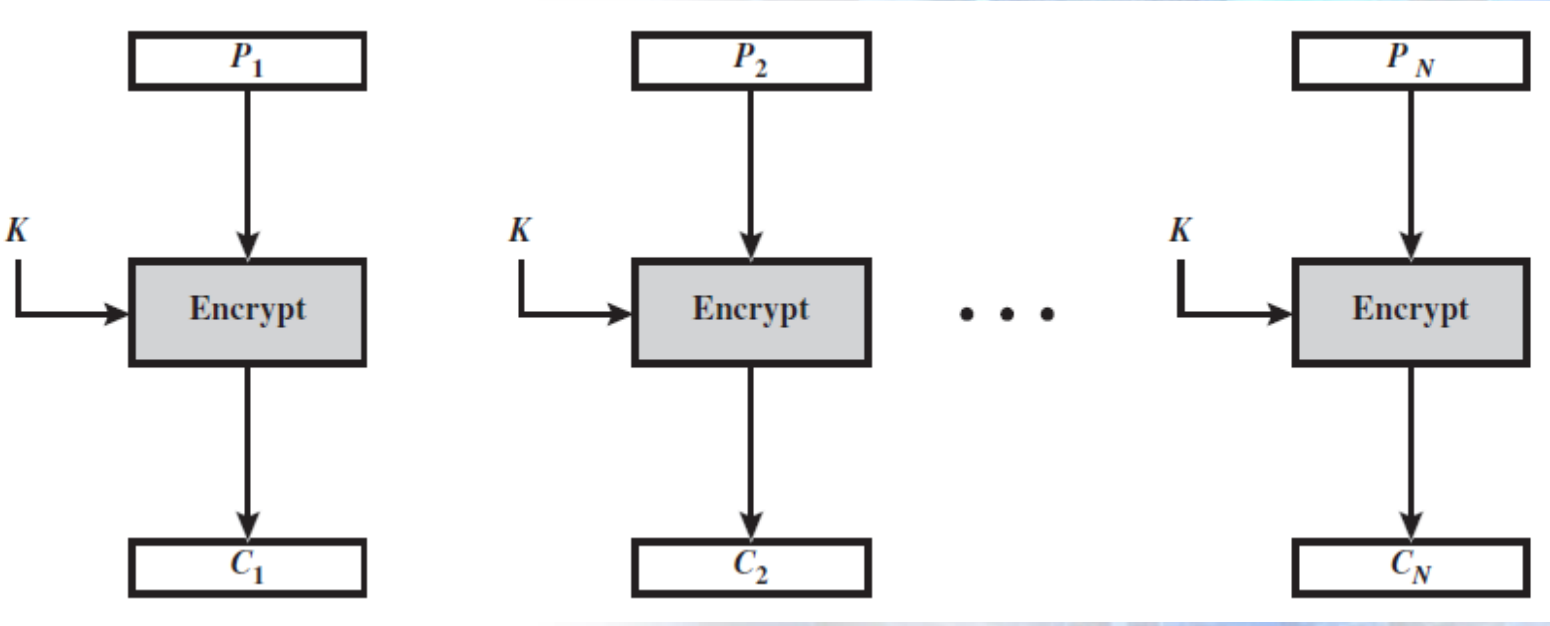
- The term codebook is used because, for a given key, there is a unique ciphertext for every **b**-bit block of plaintext.

Elect. Codebook, Cipher Block Chaining

- We can imagine a gigantic codebook in which there is an entry for every possible **b**-bit plaintext pattern showing its corresponding ciphertext.

Elect. Codebook, Cipher Block Chaining

For a message longer than b bits, the procedure is simply to break the message into b-bit blocks



Elect. Codebook, Cipher Block Chaining

- With ECB, if the same **b**-bit block of plaintext appears more than once in the message, it always produces the same ciphertext.
- Because of this, for lengthy messages, the ECB mode may not be secure.

Elect. Codebook, Cipher Block Chaining

- If the message has repetitive elements with a period of repetition a multiple of **b**-bits, these elements can be identified.
- We want to produce different ciphertext blocks for the same plaintext block if repeated .

Elect. Codebook, Cipher Block Chaining

Cipher Block Chaining Mode:

- The input to the encryption algorithm is the XOR of the current plaintext block and the preceding ciphertext block.
- The same key is used for each block.

Elect. Codebook, Cipher Block Chaining

- In effect, we have chained together the processing of the sequence of plaintext blocks.

Elect. Codebook, Cipher Block Chaining

- The input to the encryption function for each plaintext block bears no fixed relationship to the plaintext block.
- Therefore, repeating patterns of **b**-bits are not exposed.

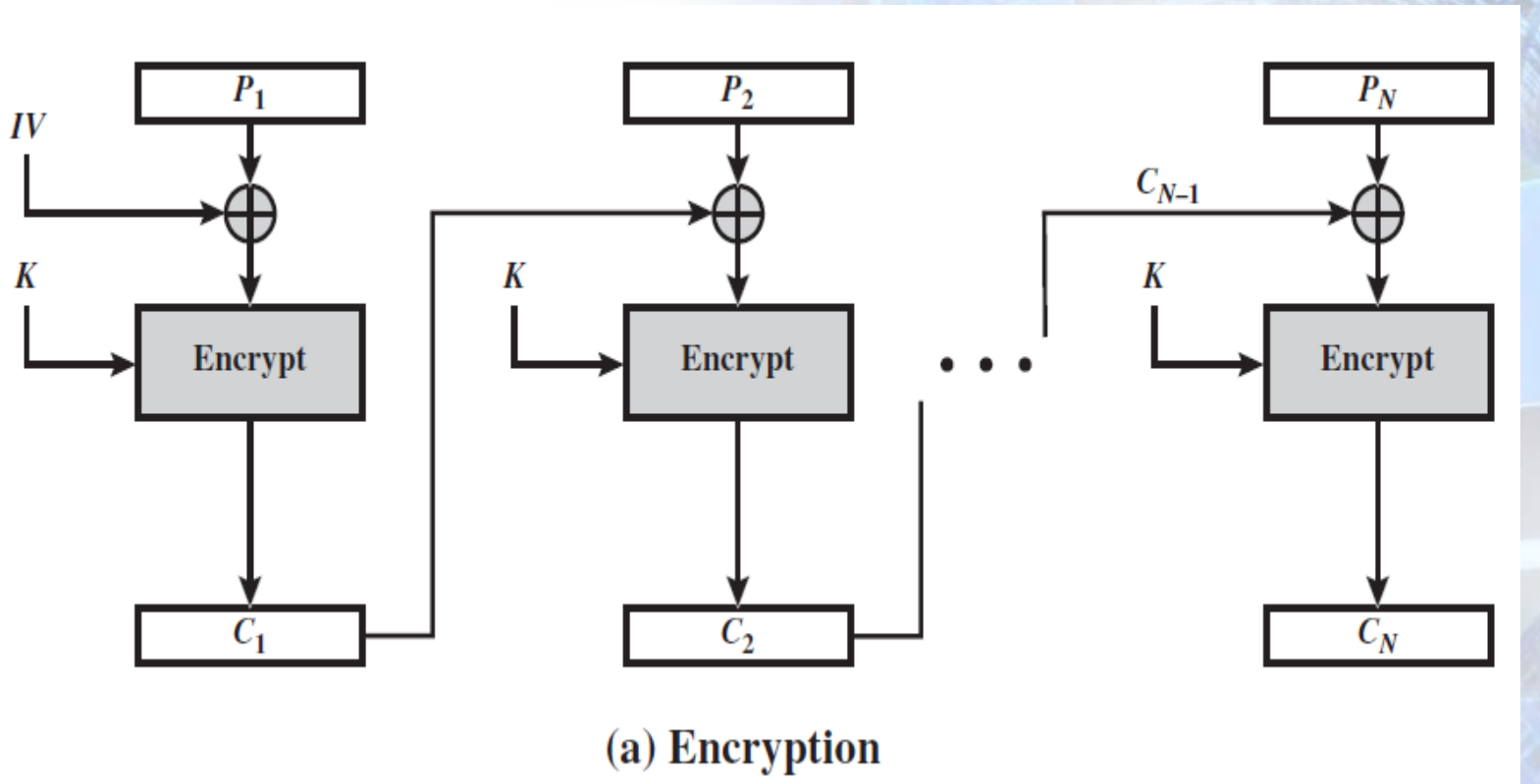
Elect. Codebook, Cipher Block Chaining

CBC Encryption:

- To produce the first block of ciphertext, an initialization vector (IV) is XORed with the first block of plaintext.
- For the jth output

$$C_j = \mathrm{E}(K, [C_{j-1} \oplus P_j])$$

Elect. Codebook, Cipher Block Chaining



(a) Encryption

Elect. Codebook, Cipher Block Chaining

- The IV must be known to both the sender and receiver but be unpredictable by a third party.

Elect. Codebook, Cipher Block Chaining

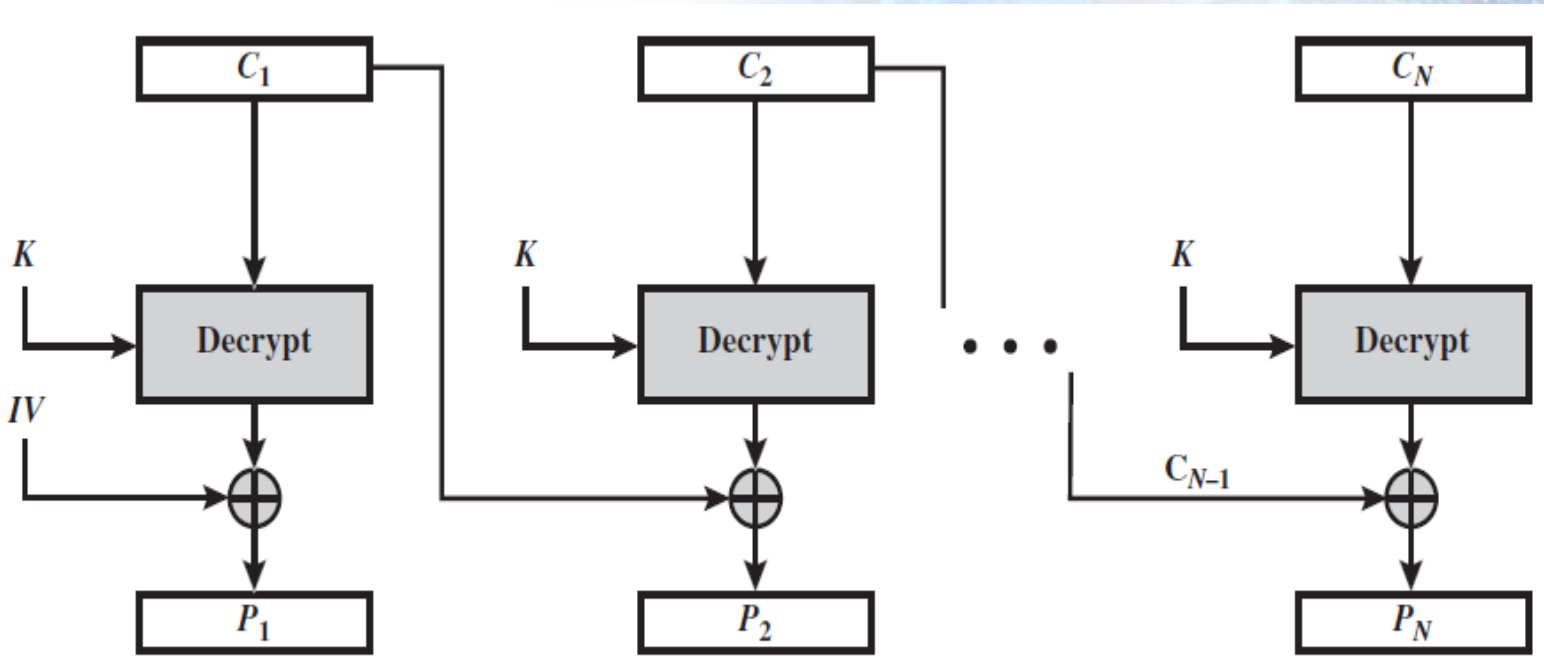
CBC Decryption:

- For decryption, each cipher block is passed through the decryption algorithm.
- The result is XORed with the preceding ciphertext block to produce the plaintext block.

Elect. Codebook, Cipher Block Chaining

- On decryption, the IV is XORed with the output of the decryption algorithm to recover the first block of plaintext.

Elect. Codebook, Cipher Block Chaining



(b) Decryption

Elect. Codebook, Cipher Block Chaining

End

- Because of the chaining mechanism of CBC, it is an appropriate mode for encrypting messages of length greater than **b**-bits.

Cipher Feedback Mode

Network Security

Cipher Feedback Mode

Objectives of the Topic

- After completing this topic, a student will be able to
  - understand cipher feedback mode.

Cipher Feedback Mode

Figures and material in this topic have been

- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

Cipher Feedback Mode

- A block cipher takes a fixed-length block of text of length ***b***-bits and a key as input and produces a ***b***-bit block of ciphertext.
- NIST has defined five modes of operation so that block ciphers can be applied in a variety of applications.

Cipher Feedback Mode

- Electronic Codebook Mode (ECB)
- Cipher Block Chaining Mode (CBC)
- Cipher Feedback Mode (CFB)
- Output Feedback (OFB)
- Counter Mode (CTR)

Cipher Feedback Mode

- It is possible to convert a block cipher into a stream cipher, using one of the three CFB, OFB, and CTR modes.
- A stream cipher eliminates the need to pad a message to be an integral number of blocks.

Cipher Feedback Mode

- It also can operate in real time.
- Thus, if a character stream is being transmitted, each character can be encrypted and transmitted immediately using a character-oriented stream cipher.

Cipher Feedback Mode

- One desirable property of a stream cipher is that the ciphertext be of the same length as the plaintext.

Cipher Feedback Mode

- If 8-bit characters are being transmitted, each character should be encrypted using 8 bits.
- If more than 8 bits are used, transmission capacity is wasted.

Cipher Feedback Mode

- Assume that the unit of transmission is **s** bits; a common value is s = 8.
- Rather than blocks of **b**-bits, the plaintext is divided into segments of **s**-bits.
- As with CBC, the units of plaintext are chained together.

Cipher Feedback Mode

Encryption:

- The input to the encryption function is a b-bit shift register that is initially set to some initialization vector (IV).

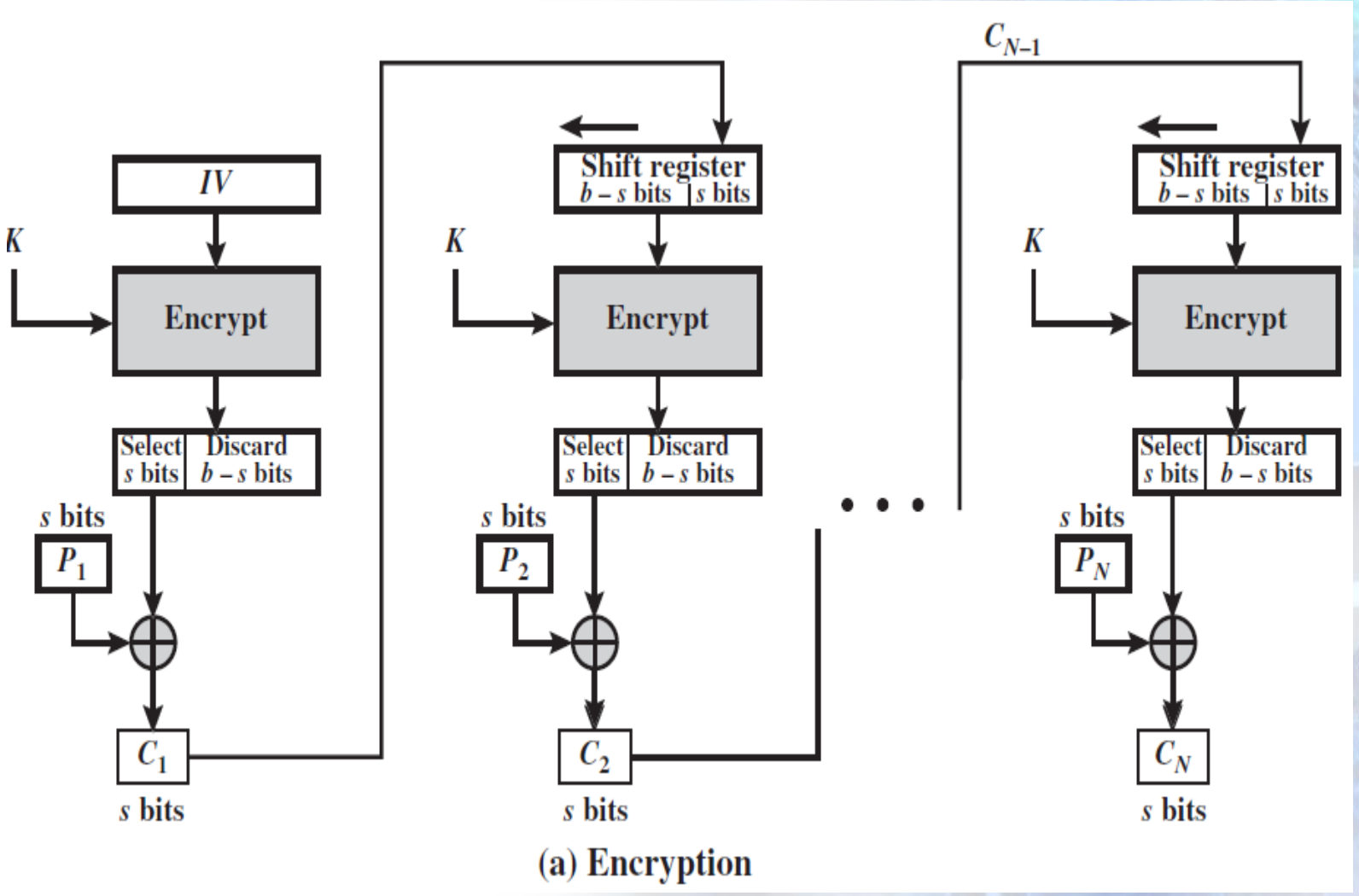
Cipher Feedback Mode

- The leftmost (most significant) s bits of the output of the encryption function are XORed with the first segment of plaintext P_1 to produce the first unit of ciphertext C_1, which is then transmitted.

Cipher Feedback Mode

- In addition, the contents of the shift register are shifted left by s bits, and C_1 is placed in the rightmost (least significant) s bits of the shift register.
- This process continues until all plaintext units have been encrypted.

Cipher Feedback Mode



(a) Encryption

Cipher Feedback Mode

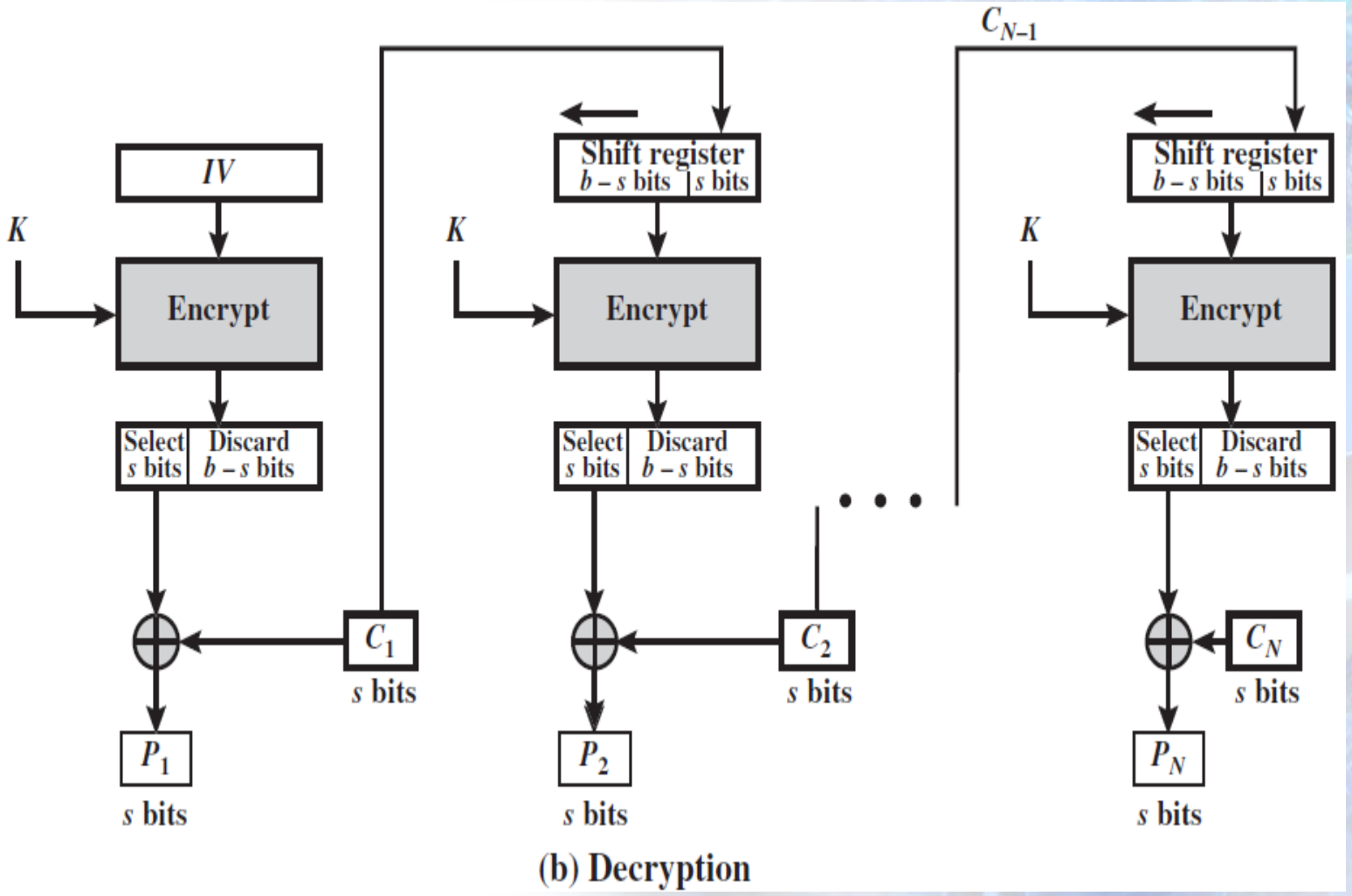
Decryption:

- the same scheme is used, except that the received ciphertext unit is XORed with the output of encryption function to produce the plaintext unit.

$$C_1 = P_1 \oplus \text{MSB}_s[\text{E}(K, \text{IV})]$$

$$P_1 = C_1 \oplus \text{MSB}_s[\text{E}(K, \text{IV})]$$

Cipher Feedback Mode



(b) Decryption

Cipher Feedback Mode

- In a typical stream cipher, the cipher takes as input some initial value and a key and generates a stream of bits, which is then XORed with the plaintext bits.

Cipher Feedback Mode

- In the case of CFB, the stream of bits that is XORed with the plaintext also depends on the plaintext.

End

Counter Mode, Output Feedback Mode

Network Security

Counter Mode, Output Feedback Mode

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain counter and output feedback modes.

Counter Mode, Output Feedback Mode

Figures and material in this topic have been adapted from

- *"Network Security Essentials"*, 2014, by William Stallings.
- W. Stallings, "*Crypto. and Network Security Principles and Practice*", Pearson Education, 2014

Counter Mode, Output Feedback Mode

- Electronic Codebook Mode (ECB)
- Cipher Block Chaining Mode (CBC)
- Cipher Feedback Mode (CFB)
- Output Feedback (OFB)
- Counter Mode (CTR)

Counter Mode, Output Feedback Mode

Output Feedback Mode

- Similar in structure to that of CFB.
- For OFB, the output of encryption function is fed back to become the input for encrypting the next block of plaintext.

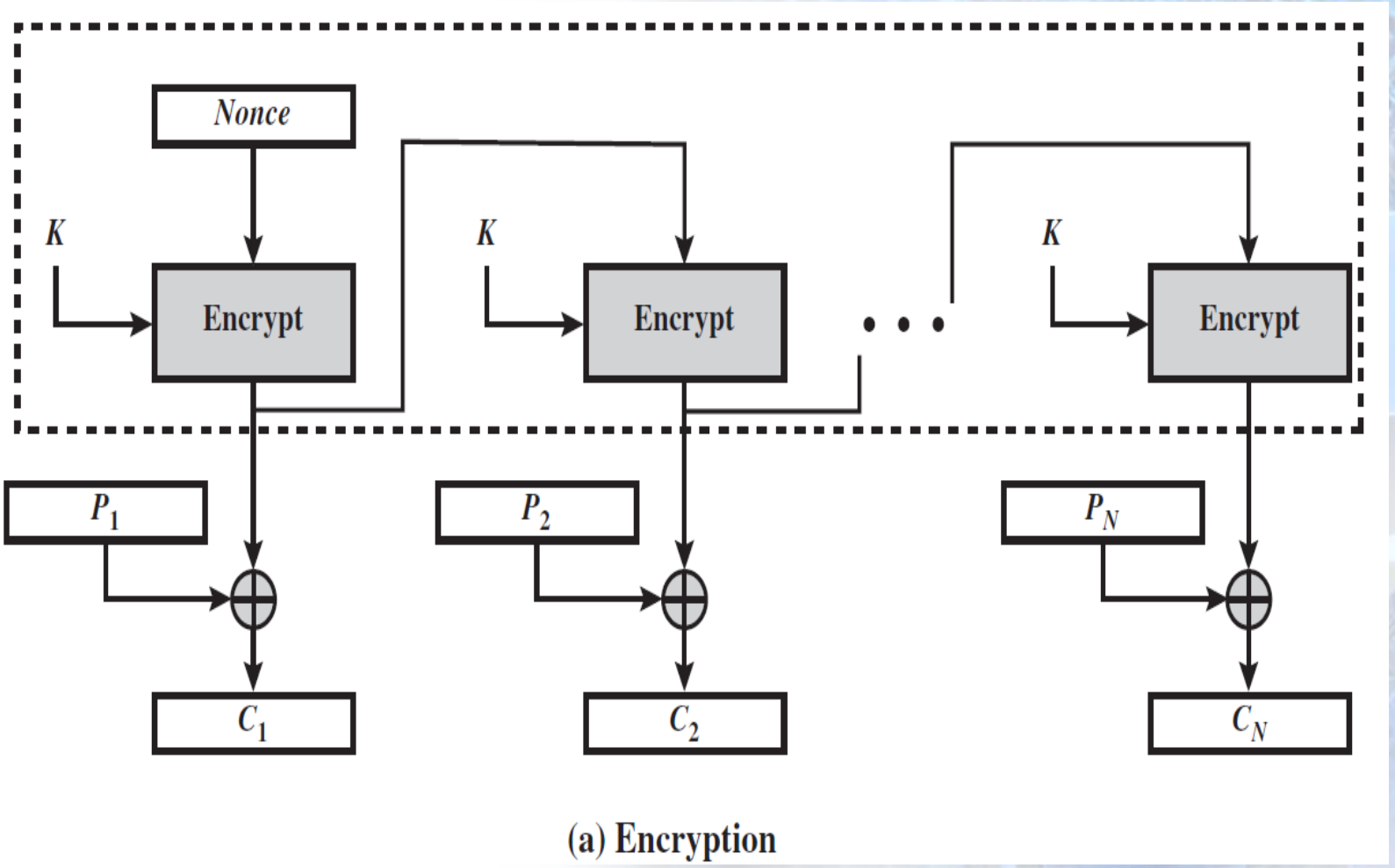
Counter Mode, Output Feedback Mode

- The OFB mode operates on full blocks of plaintext and ciphertext, whereas CFB operates on an *s*-bit subset.

Counter Mode, Output Feedback Mode

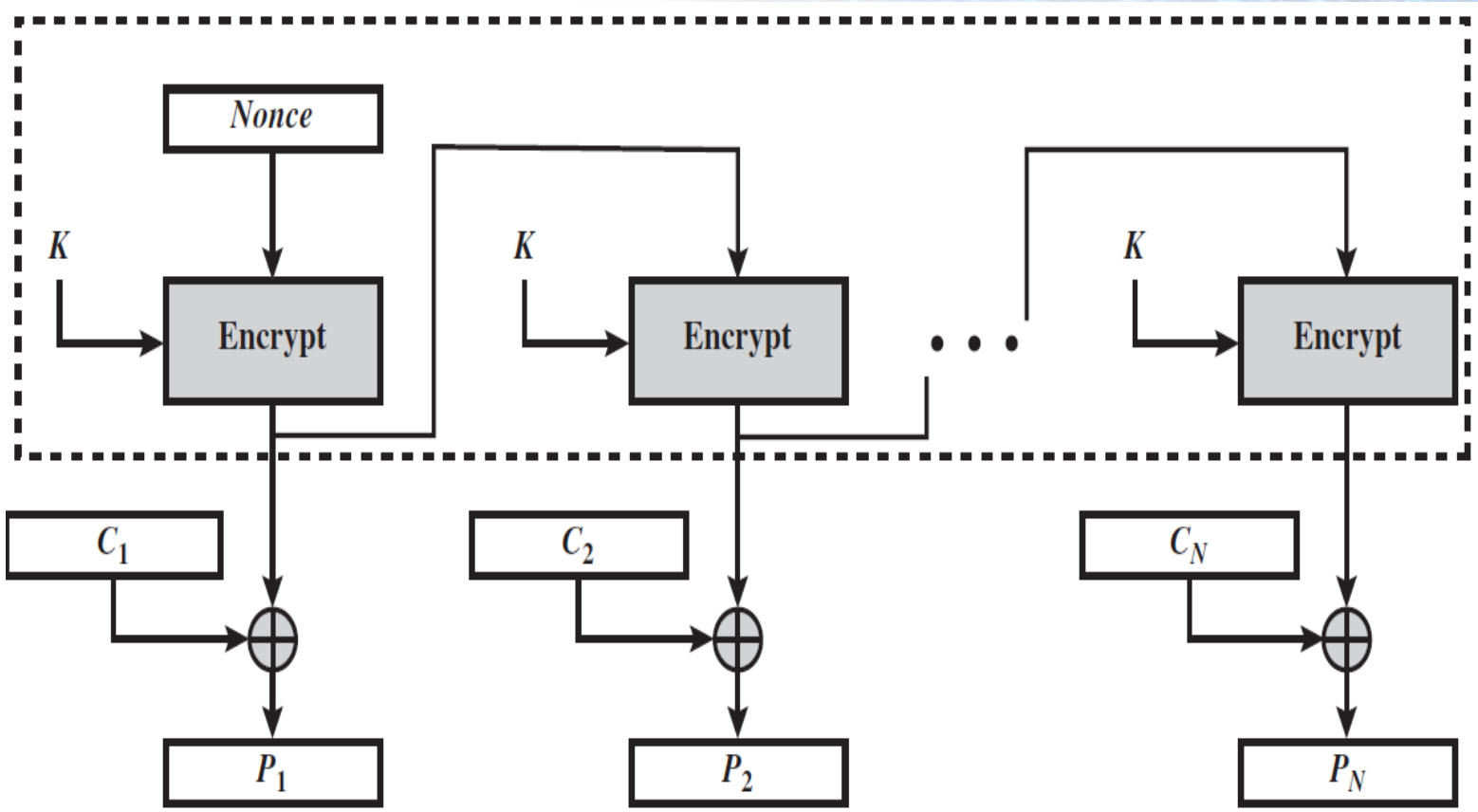
- The OFB mode requires a nonce; an initialization vector, unique to each execution of the encryption operation.
- Sequence of encryption output blocks depends only on the key and the IV and not on plaintext.

Counter Mode, Output Feedback Mode



(a) Encryption

Counter Mode, Output Feedback Mode



(b) Decryption

Counter Mode, Output Feedback Mode

- An advantage of the OFB method is that bit errors in transmission do not propagate.
- The disadvantage of OFB is that it is more vulnerable to a message stream modification attack than is CFB.

Counter Mode, Output Feedback Mode

Counter Mode

- Employed in applications to ATM (asynchronous transfer mode), network security and IPSec (IP security).
- A counter equal to the plaintext block size is used in this mode.

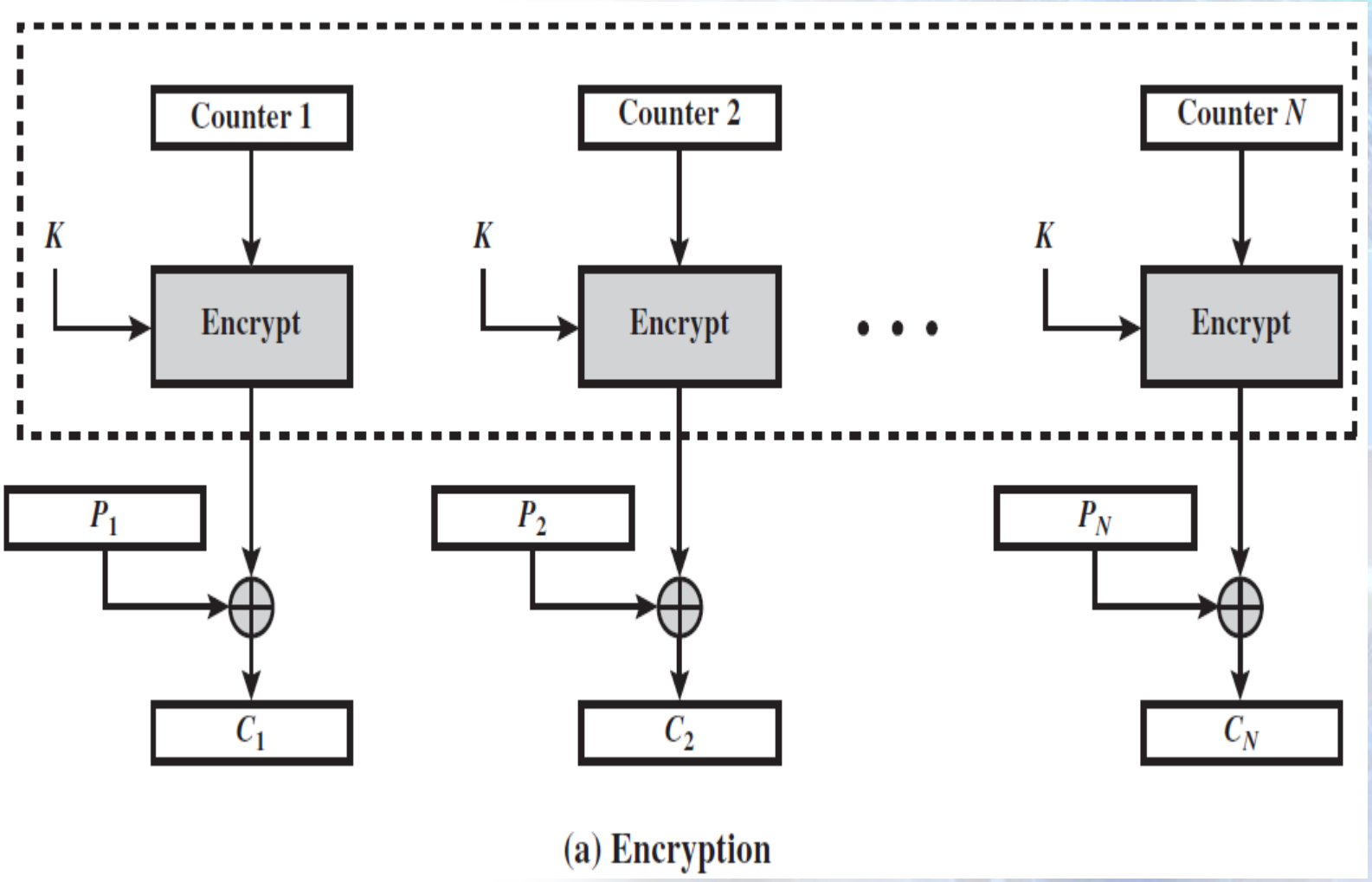
Counter Mode, Output Feedback Mode

- The counter value must be different for each plaintext block that is encrypted.
- Typically, the counter is initialized to some value and then incremented by 1 for each subsequent block (modulo 2^b, where b is block size).

Counter Mode, Output Feedback Mode

- For encryption, the counter is encrypted and then XORed with the plaintext block to produce the ciphertext block.
- There is no chaining.

Counter Mode, Output Feedback Mode

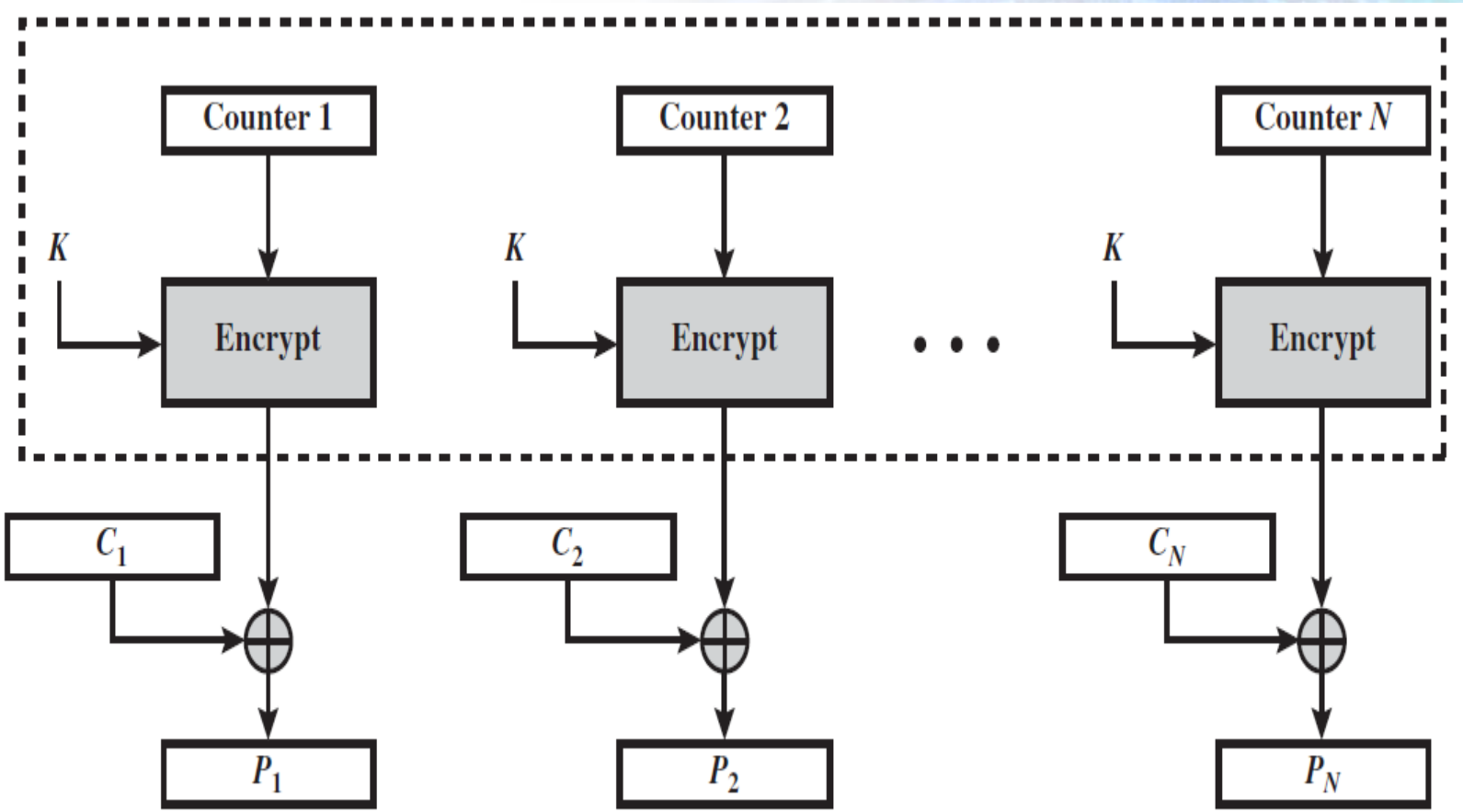


(a) Encryption

Counter Mode, Output Feedback Mode

- For decryption, the same sequence of counter values is used, with each encrypted counter XORed with a ciphertext block to recover the corresponding plaintext block.

Counter Mode, Output Feedback Mode



(b) Decryption

Counter Mode, Output Feedback Mode

Advantages of Counter Mode:

- **Preprocessing**
- when the plaintext or ciphertext input is presented, the only computation is a series of XORs, greatly enhancing throughput.

Counter Mode, Output Feedback Mode

- **Random access**
- The ith block of plaintext or ciphertext can be processed in random-access fashion
- **Provable security**
- CTR can be shown to be at least as secure as the other modes.

Counter Mode, Output Feedback Mode

- **Hardware efficiency**
- Encryption/decryption can be done in parallel on multiple blocks of plaintext or ciphertext
- **Software efficiency**
- Processors that support parallel features can be effectively utilized.

Counter Mode, Output Feedback Mode

- **Simplicity**
- Requires only the implementation of the encryption algorithm and not the decryption algorithm

End

Message Authentication

Network Security

Message Authentication

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain message authentication approaches.

Message Authentication

Figures and material in this topic have been adapted from

- *"Network Security Essentials : Applications and Standards"*, 2014, by William Stallings.

Message Authentication

- Encryption protects against passive attack (eavesdropping).
- Protection against active attack (falsification of data and transactions) is known as message authentication.

Message Authentication

- Message authentication is a procedure that allows communicating parties to verify that received messages, file, document, or other collection of data are authentic.

Message Authentication

- There are two important aspects:
- to verify that the contents of the message have not been altered, and
- to verify that the source is authentic.

Message Authentication

- Also, we would like to verify a message's timeliness (it has not been artificially delayed and replayed) and sequence relative to other messages flowing between two parties.
- These are related to data integrity.

Message Authentication

Authentication Using Encryption

- We can perform authentication by the use of symmetric encryption.

Message Authentication

- We assume that only the sender and receiver share a key, so only the genuine sender would be able to encrypt a message successfully.

Message Authentication

- The receiver assumes that no alterations have been made and that sequencing is proper if the message includes an error detection code and a sequence number.

Message Authentication

- If the message includes a timestamp, the receiver is assured that the message has not been delayed beyond that normally expected for network transit.

Message Authentication

Authentication without Encryption

- An authentication tag is generated and appended to each message for transmission.

Message Authentication

- The message itself is not encrypted and can be read at destination independent of the authentication function.
- Because the message is not encrypted, message confidentiality is not provided.

Message Authentication

- We can combine encryption of a message and its authentication tag in a single algorithm.
- Typically, message authentication is provided as a separate function from message encryption.

Message Authentication

Message Authentication Code (MAC)

- Is a technique that involves the use of a secret key to generate a small block of data, known as a message authentication code , that is appended to the message.

Message Authentication

- MAC assumes that two communicating parties, say A and B, share a common secret key K_{AB}.

Message Authentication

- When A has a message to send to B, it calculates the message authentication code as a function of the message and the key: $MAC_M = F(K_{AB}, M)$.

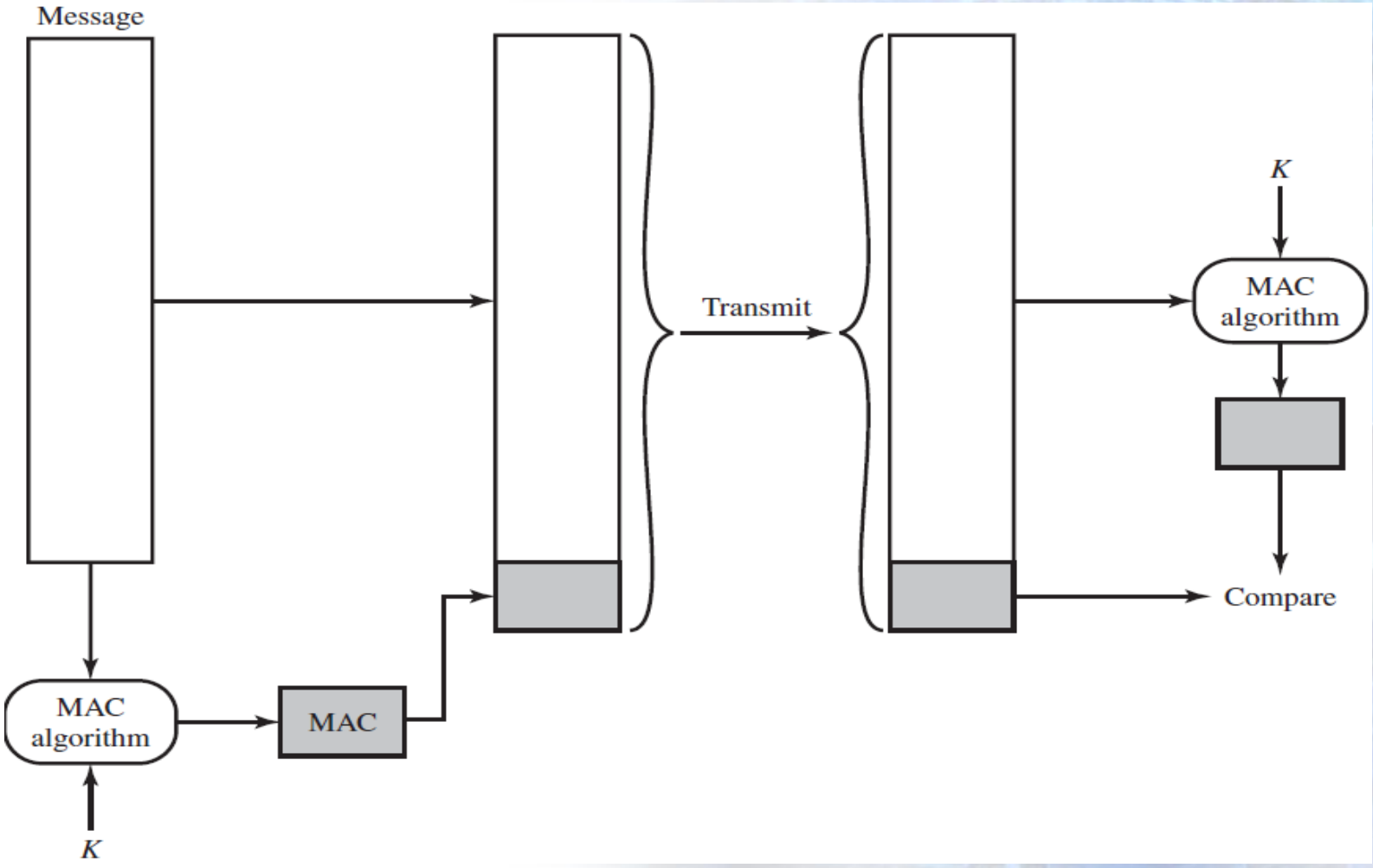
Message Authentication

- The message plus code are transmitted to the intended recipient.

Message Authentication

- The recipient performs the same calculation on the received message, using the same secret key, to generate a new message authentication code.
- The received code is compared to the calculated code.

Message Authentication



Message Authentication

- If we assume that only the receiver and the sender know the identity of the secret key, and if the received code matches the calculated code, then:

Message Authentication

- 1. The receiver is assured that message has not been altered.
- Attacker does not know the secret key. If message is altered but code remains the same, then receiver's calculation of the code will differ from the received code.

Message Authentication

- 2. The receiver is assured that the message is from the alleged sender.
- 3. If the message includes a sequence number, then the receiver can be assured of the proper sequence.

Message Authentication

End

- The NIST specification, FIPS PUB 113, recommends the use of DES.
- DES is used to encrypt the message, and the last number of bits of ciphertext are used as the code.
- A 16- or 32-bit code is typical.

One-way Hash Function

Network Security

One-way Hash Function

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain working of one-way hash function.

One-way Hash Function

Figures and material in this topic have been adapted from

- *"Network Security Essentials : Applications and Standards"*, 2014, by William Stallings.

One-way Hash Function

- Message authentication is a procedure that allows communicating parties to verify that received messages, file, document, or other collection of data are authentic.

One-way Hash Function

- There are two important aspects:
- to verify that the contents of the message have not been altered, and
- to verify that the source is authentic.

One-way Hash Function

- Also, we would like to verify a message's timeliness (it has not been artificially delayed and replayed) and sequence relative to other messages flowing between two parties.
- These are related to data integrity.

One-way Hash Function

One-way Hash Function:

- Is an alternative to the message authentication code (MAC).

One-way Hash Function

- A hash function accepts a variable-size message *M* as input and produces a fixed-size hash value h = H(M).

One-way Hash Function

- When a hash function is used to provide message authentication, the hash function value is often referred to as a message digest.

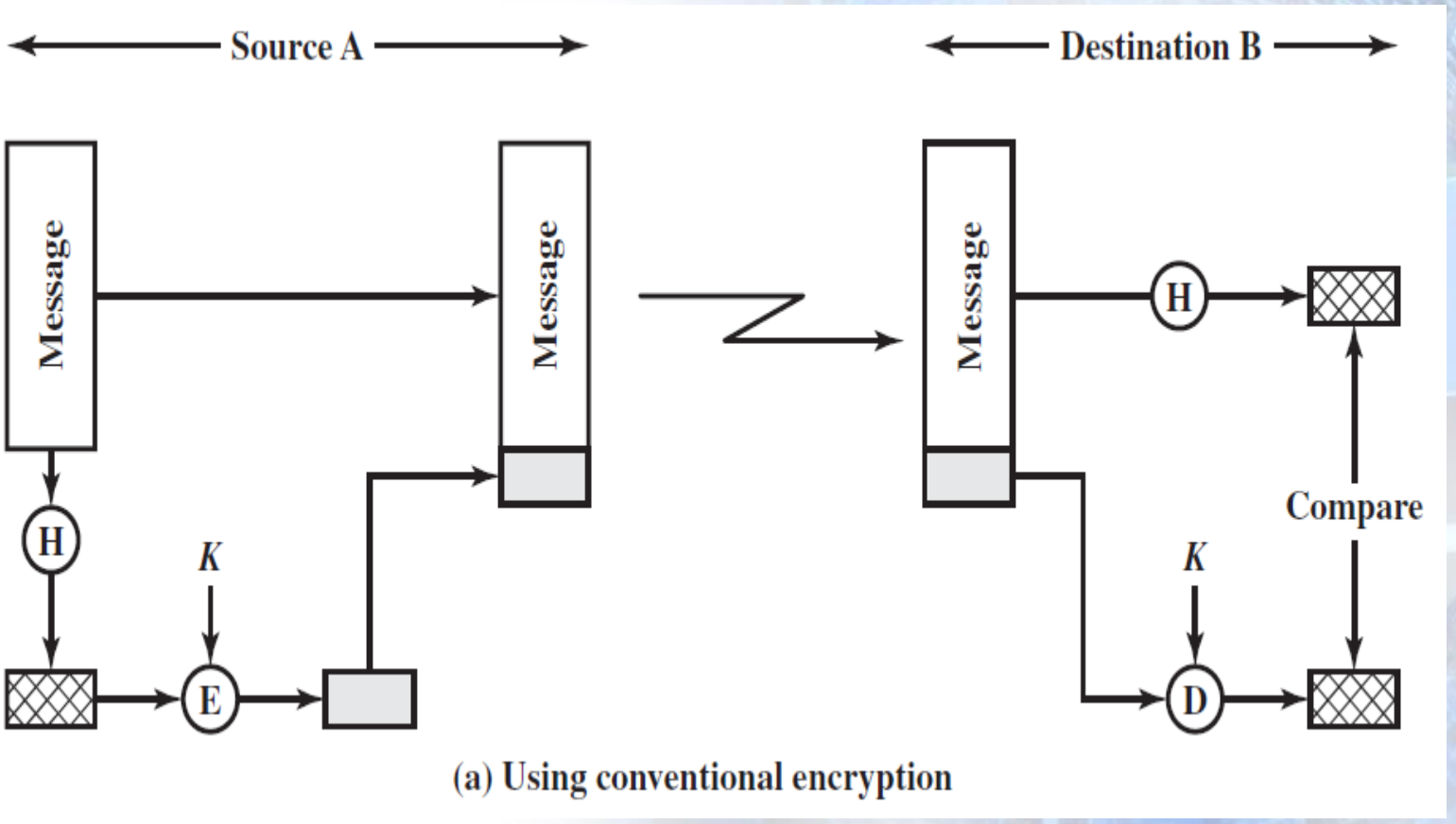
One-way Hash Function

- A hash function does not take a secret key as input.
- To authenticate a message, the message digest is sent with the message in such a way that the message digest is authentic.

One-way Hash Function

- There are three ways in which the message can be authenticated.
- **A)** The message digest can be encrypted using encryption if it is assumed that only the sender and receiver share the encryption key, then authenticity is assured.

One-way Hash Function



(a) Using conventional encryption

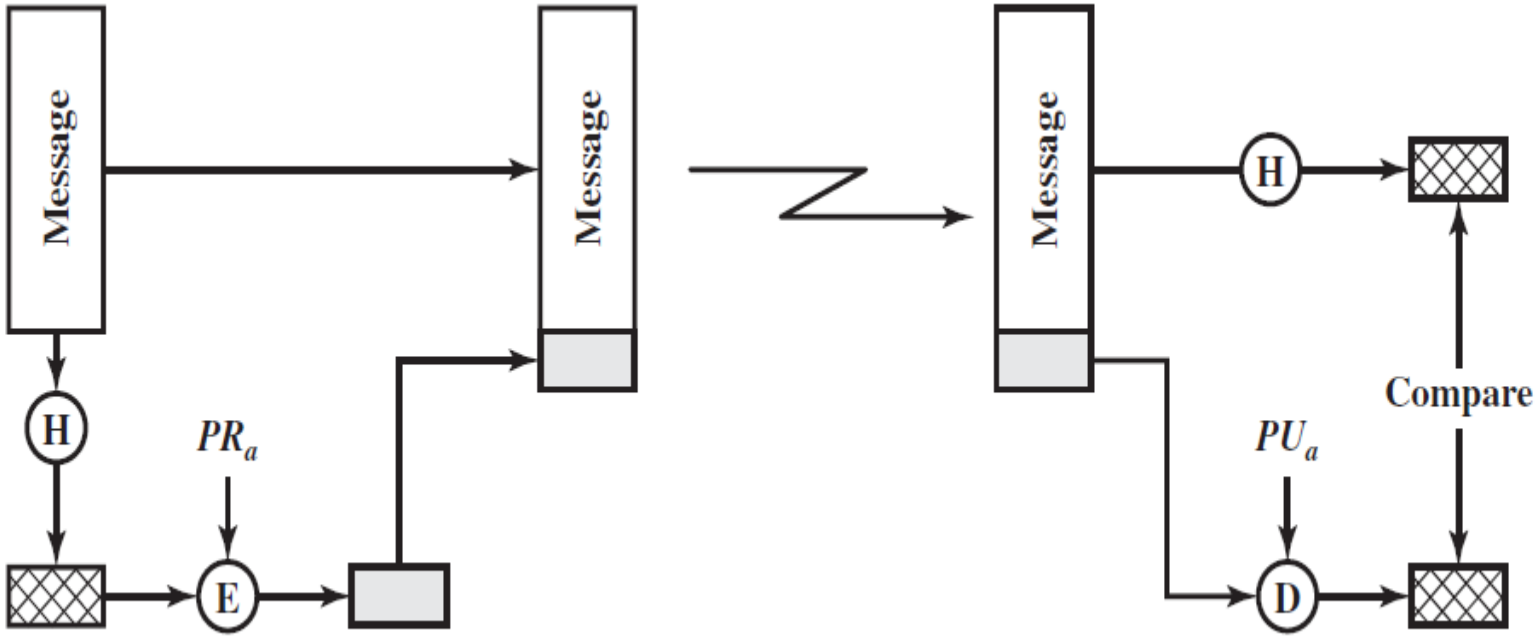
One-way Hash Function

- **B)** The message digest can be encrypted using public-key encryption.
- This approach has two advantages:
- (1) It provides a digital signature as well as message authentication.

One-way Hash Function

- (2) It does not require the distribution of keys to communicating parties.

One-way Hash Function



(b) Using public-key encryption

One-way Hash Function

- These two approaches require less computations over approaches that encrypt the entire message.
- There has been interest in developing a technique that avoids encryption altogether.

One-way Hash Function

- **C)** uses a hash function but no encryption for message authentication.
- This technique assumes that two communicating parties, say A and B, share a common secret value S_{AB}.

One-way Hash Function

- When A has a message to send to B, it calculates the hash function over the concatenation of the secret value and the message:
  $MD_M = H(S_{AB}||M)$.
- It then sends $[M||MD_M]$ to B.

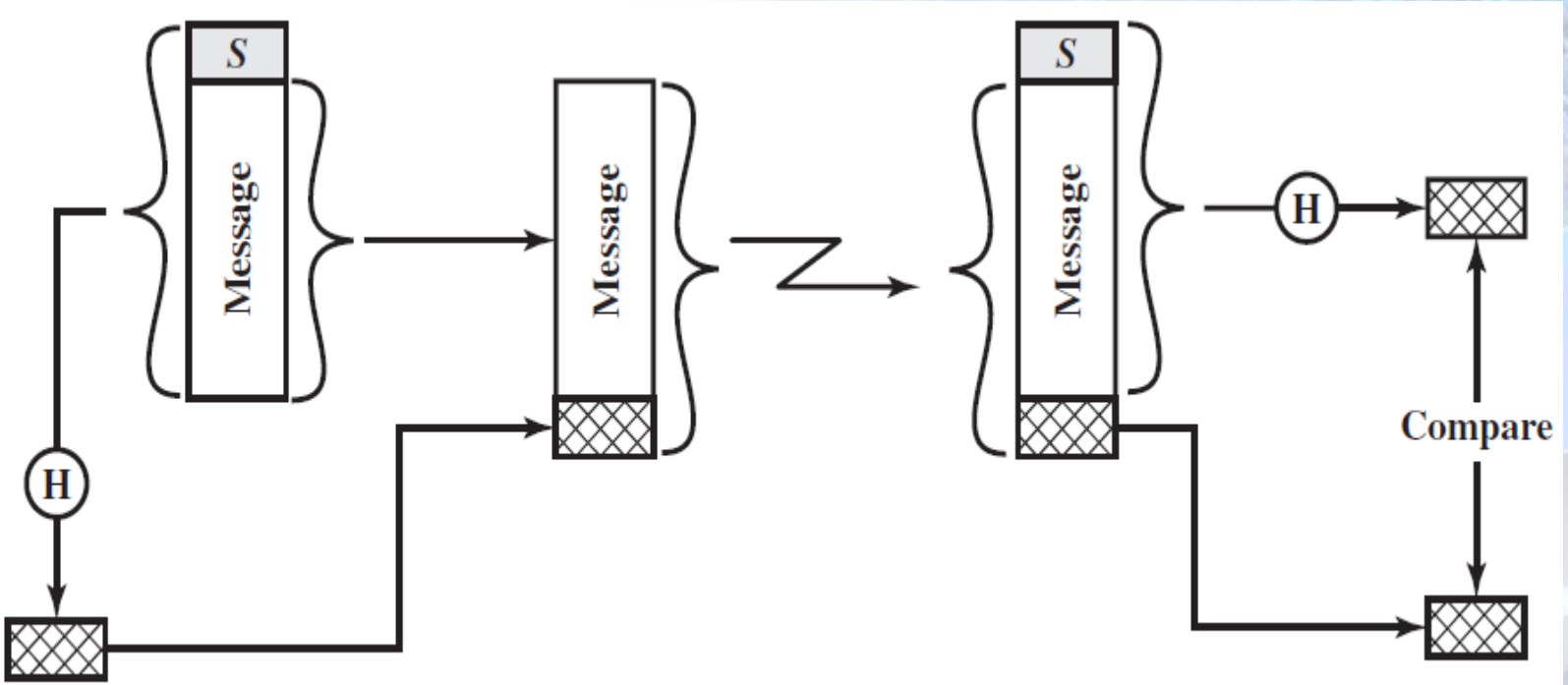
One-way Hash Function

- Because B possesses S_{AB}, it can recompute $H(S_{AB}||M)$ and verify MD_M.
- Because the secret value itself is not sent, it is not possible for an attacker to modify an intercepted message.

One-way Hash Function

- As long as the secret value remains secret, it is also not possible for an attacker to generate a false message.

One-way Hash Function



(c) Using secret value

One-way Hash Function

End

- A variation on the third technique is the one adopted for IP security
- It also has been specified for Simple Network Management Protocol (SNMP)v3.

Hash Function Requirements

Network Security

Hash Function Requirements

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain hash function requirements.

Hash Function Requirements

Figures and material in this topic have been adapted from

- *"Network Security Essentials : Applications and Standards"*, 2014, by William Stallings.

Hash Function Requirements

- The purpose of a hash function is to produce a “fingerprint” of a file, message, or other block of data.
- It accepts a variable-length block of data M as input and produces a fixed-size hash value h = H(M).

Hash Function Requirements

- A “good” hash function has the property that the results of applying the function to a large set of inputs will produce outputs that are evenly distributed and apparently random.

Hash Function Requirements

- A change to any bit or bits in M results, with high probability, in a change to the hash code.
- The principal object of a hash function is data integrity.

Hash Function Requirements

- For a hash value h = H(x), we say that x is the **preimage** of h.
- In other words, x is a data block whose hash function, using the function H, is h.

Hash Function Requirements

- Because H is a many-to-one mapping, for any given hash value h, there will in general be multiple preimages.

Hash Function Requirements

- A collision occurs if we have $x \neq y$ and $H(x) = H(y)$.
- Because we are using hash functions for data integrity, collisions are clearly undesirable.

Hash Function Requirements

- To be useful for message authentication, a hash function H must have the following properties:
- 1. H can be applied to a block of data of any size.
- 2. H produces a fixed-length output.

Hash Function Requirements

- 3. H(x) is relatively easy to compute for any given x, making both hardware and software implementations practical.

Hash Function Requirements

- 4. For any given code h, it is computationally infeasible to find x such that H(x) = h.
- A hash function with this property is referred to as **one-way** or **preimage resistant**.

Hash Function Requirements

- 5. For any given block x, it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$.
- A hash function with this property is referred to as **second preimage resistant**.
- This is also referred to as **weak collision resistant**.

Hash Function Requirements

- 6. It is computationally infeasible to find any pair (x, y) such that H(x) = H(y).
- Such a hash function is referred to as **collision resistant**.
- This is sometimes referred to as **strong collision resistant**.

Hash Function Requirements

- First three properties are requirements for the practical application of a hash function to message authentication.
- The 4th property is important if the authentication technique involves the use of a secret value.

Hash Function Requirements

- The fourth property, preimage resistant, is the “one-way” property: It is easy to generate a code given a message, but virtually impossible to generate a message given a code.

Hash Function Requirements

- The 5th property prevents forgery when an encrypted hash code is used.

Hash Function Requirements

- A hash function that satisfies the first five properties in the preceding list is referred to as a weak hash function.
- If the sixth property is also satisfied, then it is referred to as a strong hash function.

Hash Function Requirements

Security of Hash Functions:

- Two approaches to attacking a secure hash function are:
- Cryptanalysis, and
- brute-force attack

Hash Function Requirements

- Cryptanalysis of a hash function involves exploiting logical weaknesses in the algorithm.
- The strength of a hash function against brute-force attacks depends on length of hash code produced by algorithm.

Hash Function Requirements

- For a hash code of length n, the level of effort required is proportional to the following:

Preimage resistant	2^n
Second preimage resistant	2^n
Collision resistant	$2^{n/2}$

End

Simple Hash Functions

Network Security

Simple Hash Functions

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain working of simple hash functions.

Simple Hash Functions

Figures and material in this topic have been adapted from

- *"Network Security Essentials : Applications and Standards"*, 2014, by William Stallings.

Simple Hash Functions

- A hash function accepts a variable-length block of data M as input and produces a fixed-size hash value h = H(M).

Simple Hash Functions

- A “good” hash function has the property that the results of applying the function to a large set of inputs will produce outputs that are evenly distributed and apparently random.

Simple Hash Functions

- A change to any bit or bits in M results, with high probability, in a change to the hash code.
- The principal object of a hash function is data integrity.

Simple Hash Functions

Simple Hash Functions:

- All hash functions operate using the following general principles.
- The input (message, file, etc.) is viewed as a sequence of n-bit blocks.

Simple Hash Functions

- The input is processed one block at a time in an iterative fashion to produce an n-bit hash function.
- One of the simplest hash functions is the bit-by-bit exclusive-OR (XOR) of every block.

Simple Hash Functions

- If

C_i = ith bit of the hash code, $1 \le i \le n$

m = number of n-bit blocks in the input

b_{ij} = ith bit in jth block

$\oplus$ = XOR operation

- Then,

$$C_i = b_{i1} \oplus b_{i2} \oplus \ldots \oplus b_{im}$$

Simple Hash Functions

Simple Hash Function Using Bitwise XOR

	bit 1	bit 2	• • •	bit n
Block 1	b_{11}	b_{21}		b_{n1}
Block 2	b_{12}	b_{22}		b_{n2}
	•	•	•	•
	•	•	•	•
	•	•	•	•
Block m	b_{1m}	b_{2m}		b_{nm}
Hash code	C_1	C_2		C_n

Simple Hash Functions

- This operation produces a simple parity for each bit position and is known as a longitudinal redundancy check.
- It is reasonably effective for random data as a data integrity check.

Simple Hash Functions

- Each n-bit hash value is equally likely.
- Thus, the probability that a data error will result in an unchanged hash value is 2^{-n}.
- With more predictably formatted data, the function is less effective.

Simple Hash Functions

- For example, in most normal text files, the high-order bit of each octet is always zero.
- With a 128-bit hash value, effectiveness of the hash function is reduced from 2^{-128} to 2^{-112} on this type of data.

Simple Hash Functions

- A simple way to improve matters is to perform a 1-bit circular shift, or rotation, on the hash value after each block is processed.
- The procedure can be summarized as:

Simple Hash Functions

- 1. Initially set the n-bit hash value to zero.
- 2. Process each successive n-bit block of data:
- a. Rotate the current hash value to the left by one bit.
- b. XOR the block into the hash value.

Simple Hash Functions

- This has the effect of "randomizing" the input more completely and overcoming any regularities that appear in the input.
- Data security is at stake when an encrypted hash code is used with a plaintext message.

Simple Hash Functions

- A technique originally proposed by the National Bureau of Standards used the simple XOR applied to 64-bit blocks of the message and then an encryption of the entire message using the cipher block chaining (CBC) mode.

Simple Hash Functions

- Given a message consisting of a sequence of 64-bit blocks $X_1, X_2, \ldots, X_N$, define the hash code C as the block-by-block XOR or all blocks and append the hash code as the final block:

$$C = X_{N+1} = X_1 \oplus X_2 \oplus \ldots \oplus X_N$$

Simple Hash Functions

End

- Next, encrypt the entire message plus hash code using CBC mode to produce the encrypted message Y_1, Y_2, . . ., Y_{N-1}
- Ciphertext of this message can be manipulated so that it is not detectable by the hash code.

The Secure Hash Function (SHA)

Network Security

The Secure Hash Function (SHA)

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain working of the secure hash algorithm (SHA).

The Secure Hash Function (SHA)

Figures and material in this topic have been adapted from

- *"Network Security Essentials : Applications and Standards"*, 2014, by William Stallings.

The Secure Hash Function (SHA)

Secure Hash Algorithm (SHA)

- Is the most widely used hash function in recent years.
- Developed by the National Institute of Standards and Technology (NIST)
- FIPS 180 in 1993.

The Secure Hash Function (SHA)

- The actual standards document is entitled "Secure Hash Standard."
- SHA is based on the hash function (Message-Digest) MD4, and its design closely models MD4.

The Secure Hash Function (SHA)

- When weaknesses were discovered in SHA (now known as SHA-0), a revised version was issued as FIPS 180-1 in 1995 and is referred to as SHA-1.

The Secure Hash Function (SHA)

- In 2002, NIST produced FIPS 180-2.
- Three new versions of SHA with hash value lengths of 256, 384, and 512 bits known as SHA-256, SHA-384, and SHA-512 were defined.
- Collectively, these are known as SHA-2.

The Secure Hash Function (SHA)

Comparison of SHA Parameters

	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Message Digest Size	160	224	256	384	512
Message Size	$< 2^{64}$	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
Block Size	512	512	512	1024	1024
Word Size	32	32	32	64	64
Number of Steps	80	64	64	80	80

Note: All sizes are measured in bits.

The Secure Hash Function (SHA)

- In 2005 NIST announced the intention to phase out approval of SHA-1 and move to a reliance on SHA-2 by 2010.
- We focus on SHA-512.

The Secure Hash Function (SHA)

SHA-512 Logic:

- The algorithm takes as input a message with a maximum length of less than 2^{128} bits and produces as output a 512-bit message digest.
- The input is processed in 1024-bit blocks.

The Secure Hash Function (SHA)

Step 1: Append padding bits

- Padding is added, even if the message is already of the desired length. No. of Padding bits = [1 1024]
- Padding consists of a single 1 bit followed by the necessary number of 0 bits.

The Secure Hash Function (SHA)

Step 2: Append length

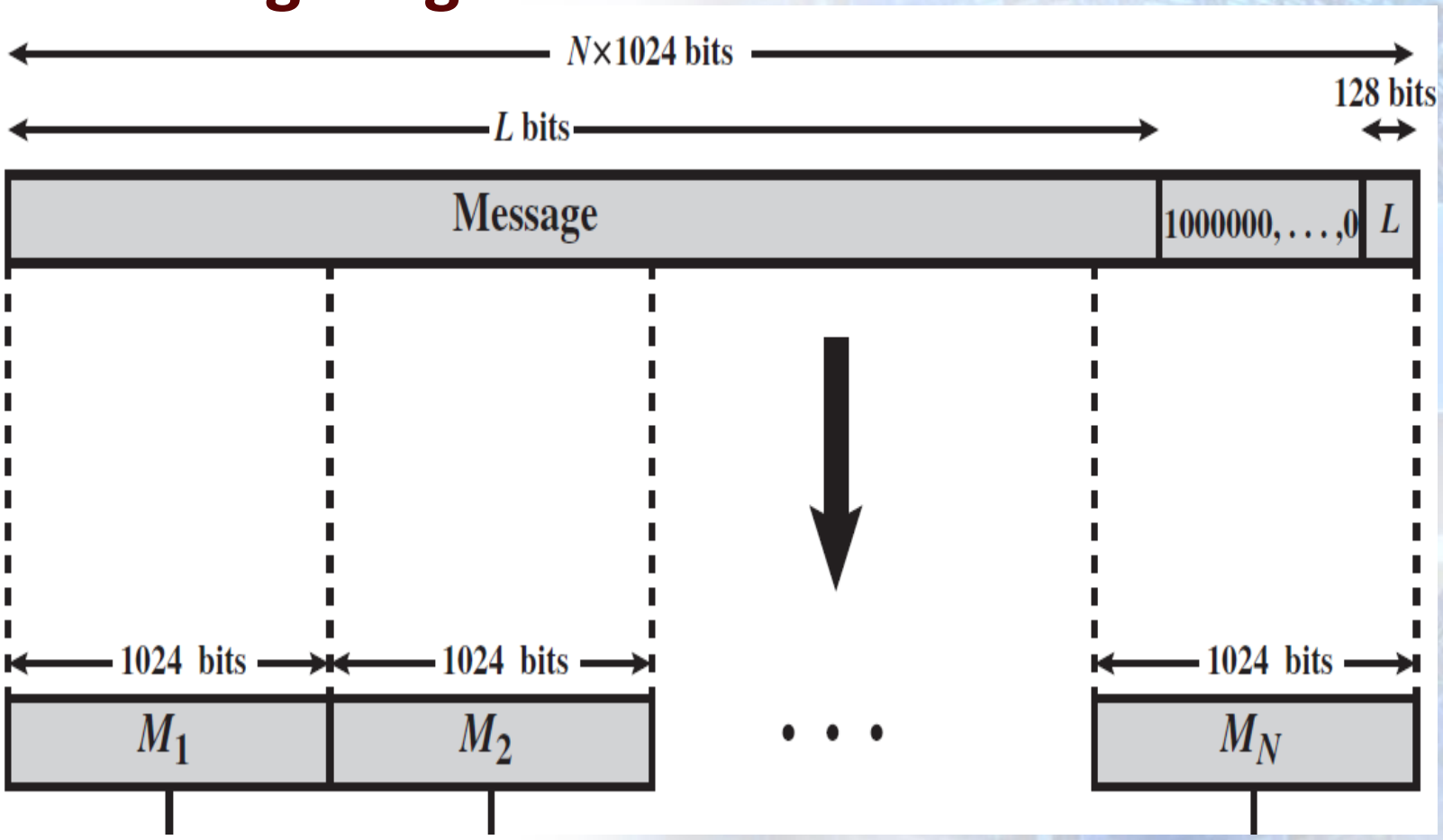
- A block of 128 bits is appended to the message.
- This block is treated as an unsigned 128-bit integer and contains the length of the original message (before the padding).

The Secure Hash Function (SHA)

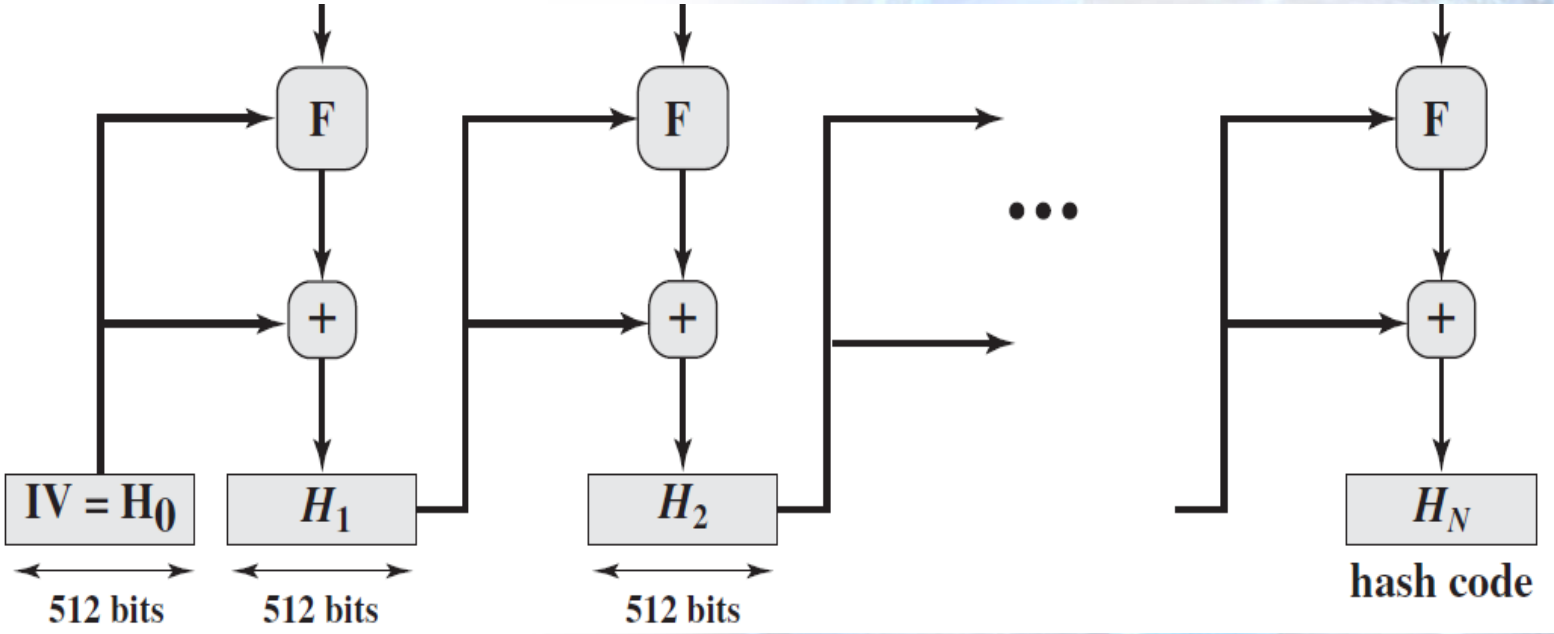
- Outcome of first two steps yields a message an integer multiple of 1024 bits in length.
- Total length of the expanded message is N × 1024 bits as the expanded message is a sequence of 1024-bit blocks M_1, M_2, . . ., M_N.

The Secure Hash Function (SHA)

Message Digest Generation of SHA-512



The Secure Hash Function (SHA)



The Secure Hash Function (SHA)

Step 3: Initialize hash buffer

- A 512-bit buffer is used to hold intermediate and final results of the hash function.
- The buffer can be represented as eight 64-bit registers (*a*, *b*, *c*, *d*, *e*, *f*, *g*, *h*).

The Secure Hash Function (SHA)

- Initialize these registers by taking the first sixty-four bits of the fractional parts of the square roots of the first eight prime numbers.

The Secure Hash Function (SHA)

Initialization of the registers

$a = 6A09E667F3BCC908$	$e = 510E527FADE682D1$
$b = BB67AE8584CAA73B$	$f = 9B05688C2B3E6C1F$
$c = 3C6EF372FE94F82B$	$g = 1F83D9ABFB41BD6B$
$d = A54FF53A5F1D36F1$	$h = 5BE0CD19137E2179$

The Secure Hash Function (SHA)

Step 4: Process message in 1024-bit (128-word) blocks

- The module labeled F consists of 80 rounds.
- Each round takes as input the 512-bit buffer value *abcdefgh* and updates the contents of the buffer.

The Secure Hash Function (SHA)

- At input to the first round, the buffer has the value of the intermediate hash value, H_{i-1}.
- Each round **t** makes use of a 64-bit value W_t derived from the current 1024-bit block being processed (M_i).

The Secure Hash Function (SHA)

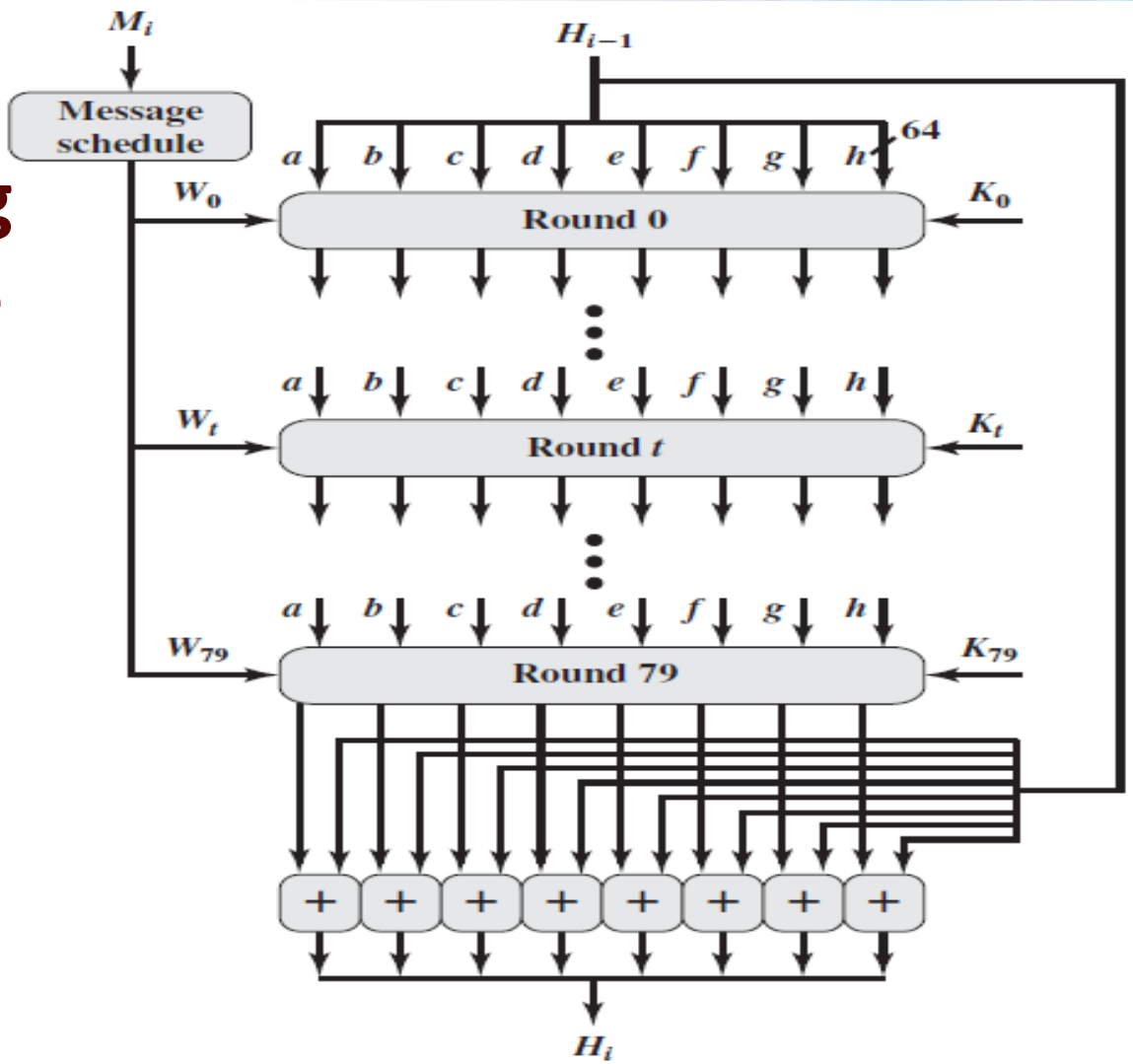
- Each round also makes use of an additive constant K_t, where t = 0 … … 79.
- The constants eliminate any regularities in the input data.

The Secure Hash Function (SHA)

- The output of the 80th round is added to the input to the first round (H_{i-1}) to produce H_i .

The Secure Hash Function (SHA)

Processing of a Single 1024-Bit Block



The Secure Hash Function (SHA)

End

Step 5 Output:

- After all N 1024-bit blocks have been processed, the output from the Nth stage is the 512-bit message digest.
- In 2012, NIST formally published SHA-3.

HMAC and its Design Objectives

Network Security

HMAC and its Design Objectives

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain design objectives of hash-based message authentication code(HMAC).

HMAC and its Design Objectives

Figures and material in this topic have been adapted from

- *"Network Security Essentials : Applications and Standards"*, 2014, by William Stallings.

HMAC and its Design Objectives

- Message authentication is a procedure that allows communicating parties to verify that received messages, file, document, or other collection of data are authentic.

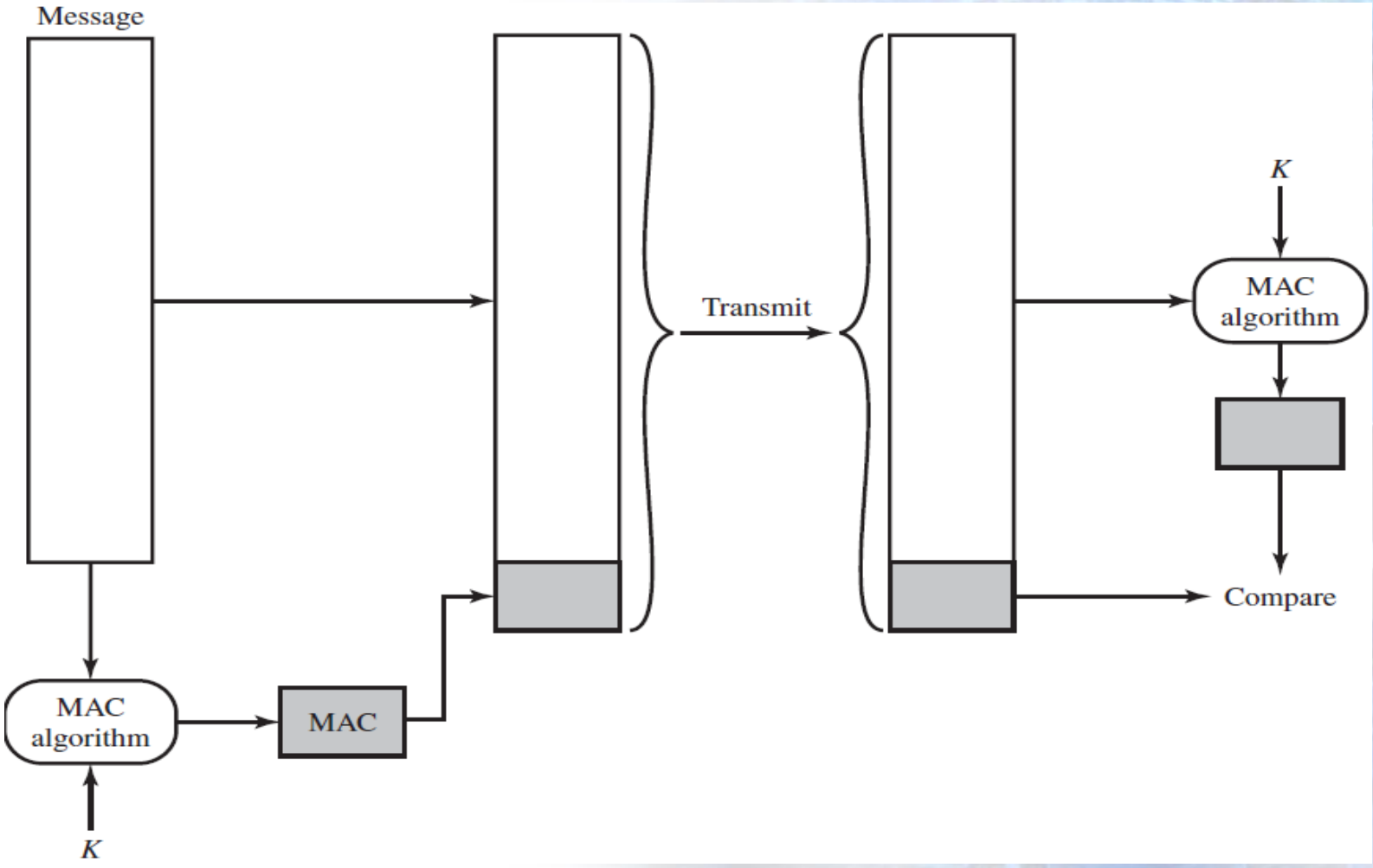
HMAC and its Design Objectives

Message Authentication Code (MAC)

- is a technique that involves the use of a secret key to generate a small block of data, known as a message authentication code , that is appended to the message.

By-Pin2

HMAC and its Design Objectives



HMAC and its Design Objectives

- There has been a growing interest in developing a MAC derived from a cryptographic hash code, such as SHA-1.
- The motivations are:

HMAC and its Design Objectives

- 1. Cryptographic hash functions generally execute faster in software than conventional encryption algorithms such as DES.
- 2. Library code for cryptographic hash functions is widely available.

HMAC and its Design Objectives

- A hash function such as SHA was not designed for use as a MAC and cannot be used directly for that purpose, because it does not rely on a secret key.

HMAC and its Design Objectives

- Among the proposals for the incorporation of a secret key into an existing hash algorithm, HMAC is the approach that has received the most support.

HMAC and its Design Objectives

- HMAC has been issued as RFC 2104,
- as a NIST standard (FIPS 198).
- as mandatory-to-implement MAC for IP Security.
- Also used in Transport Layer Security (TLS) and Secure Electronic Transaction (SET).

HMAC and its Design Objectives

- **HMAC Design Objectives:**
- RFC 2104 lists the following design objectives for HMAC.

HMAC and its Design Objectives

- 1. To use, without modifications, available hash functions.
- In particular, hash functions that perform well in software, and for which code is freely and widely available.

HMAC and its Design Objectives

- 2. To allow for easy replaceability of the embedded hash function in case faster or more secure hash functions are found or required.

HMAC and its Design Objectives

- 3. To preserve the original performance of the hash function without incurring a significant degradation.
- 4. To use and handle keys in a simple way.

HMAC and its Design Objectives

- 5. To have a well-understood cryptographic analysis of the strength of the authentication mechanism based on reasonable assumptions on the embedded hash function.

HMAC and its Design Objectives

- The first two objectives are important to the acceptability of HMAC.
- HMAC treats the hash function as a “black box.” This has two benefits.

HMAC and its Design Objectives

- First, an existing implementation of a hash function can be used as a module in implementing HMAC.

HMAC and its Design Objectives

- Second, if it is ever desired to replace a given hash function in an HMAC implementation, all that is required is to remove the existing hash function module and drop in the new module.

HMAC and its Design Objectives

- The last design objective in the preceding list is, in fact, the main advantage of HMAC over other proposed hash-based schemes.

End

HMAC Algorithm

Network Security

HMAC Algorithm

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe working of HMAC algorithm.

HMAC Algorithm

Figures and material in this topic have been adapted from

- *"Network Security Essentials : Applications and Standards"*, 2014, by William Stallings.

HMAC Algorithm

Notation:

- H = embedded hash function (e.g., SHA-1)
- IV = initial value input to hash function
- M = message input to HMAC (including the padding specified in the embedded hash function)

HMAC Algorithm

- Y_i = ith block of M, $0 \leq i \leq (L - 1)$
- L = number of blocks in M
- b = number of bits in a block
- n = length of hash code produced by embedded hash function

HMAC Algorithm

- K = secret key; recommended length is ≥ n; if key length is greater than b, the key is input to the hash function to produce an n-bit key
- K^+ = K padded with zeros on the left so that the result is b bits in length

HMAC Algorithm

- ipad = 00110110 (36 in hexadecimal) repeated b/8 times
- opad = 01011100 (5C in hexadecimal) repeated b/8 times

HMAC Algorithm

- 1. Append zeros to the left end of K to create a b-bit string K^+ (e.g., if K is of length 160 bits and b = 512, K will be appended with 44 zero bytes).
- 2. Bitwise exclusive-OR K^+ with ipad to produce the b-bit block S_i.

HMAC Algorithm

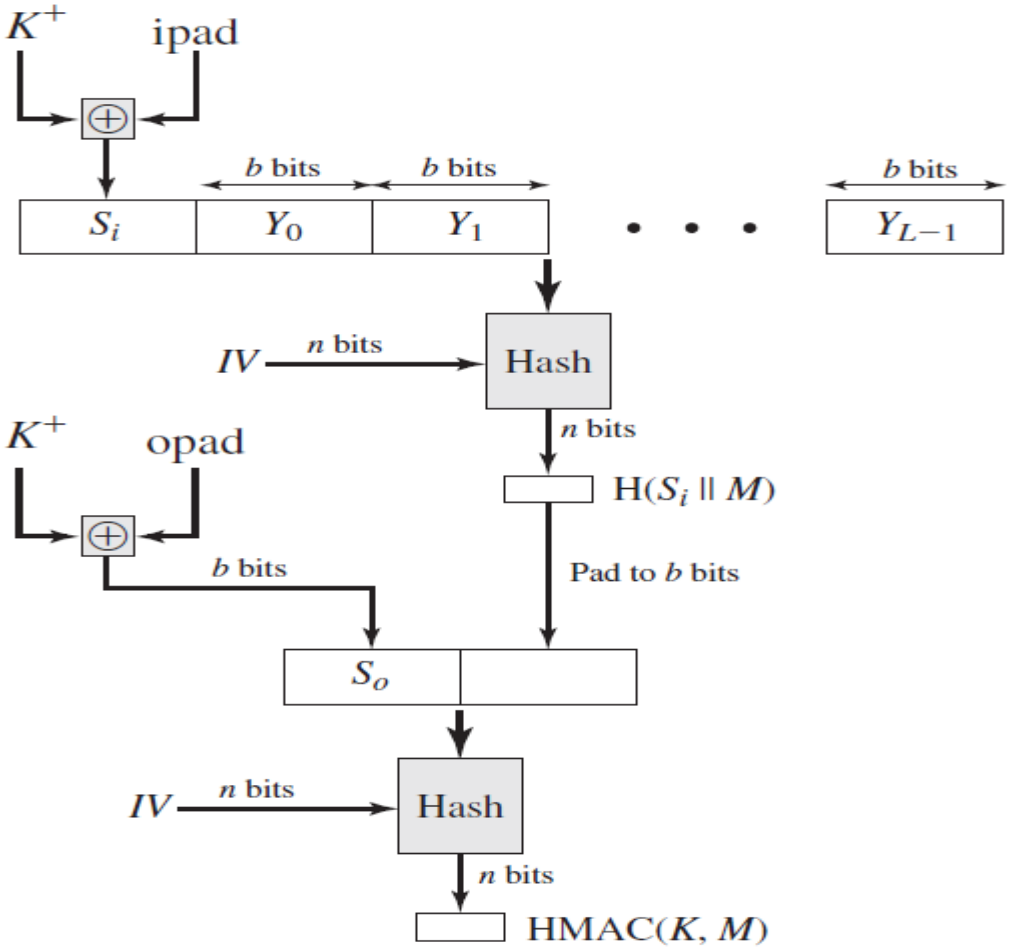
- 3. Append M to S_i.
- 4. Apply H to the stream generated in step 3.
- 5. XOR K^+ with opad to produce the b-bit block S_o.
- 6. Append the hash result from step 4 to S_o.

HMAC Algorithm

- 7. Apply H to the stream generated in step 6 and output the result.

HMAC Algorithm

$$\text{HMAC}(K, M) = \text{H}[(K^+ \oplus \text{opad}) \| \text{H}[(K^+ \oplus \text{ipad}) \| M]]$$



HMAC Algorithm

- Note that the XOR with ipad results in flipping one-half of the bits of K.
- Similarly, the XOR with opad results in flipping one-half of the bits of K, using a different set of bits.

HMAC Algorithm

- In effect, by passing S_i and S_o through the hash algorithm, we have pseudorandomly generated two keys from K.

HMAC Algorithm

- HMAC should execute in approximately the same time as the embedded hash function for long messages.

HMAC Algorithm

- HMAC adds three executions of the basic hash function (for S_i, S_o, and the block produced from the inner hash).

HMAC Algorithm

End

Security of HMAC:

- The appeal of HMAC is that its designers have been able to prove an exact relationship between the strength of the embedded hash function and the strength of HMAC.

Cipher-Based MAC

Network Security

Cipher-Based MAC

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe working of Cipher-based message authentication code.

Cipher-Based MAC

Figures and material in this topic have been adapted from

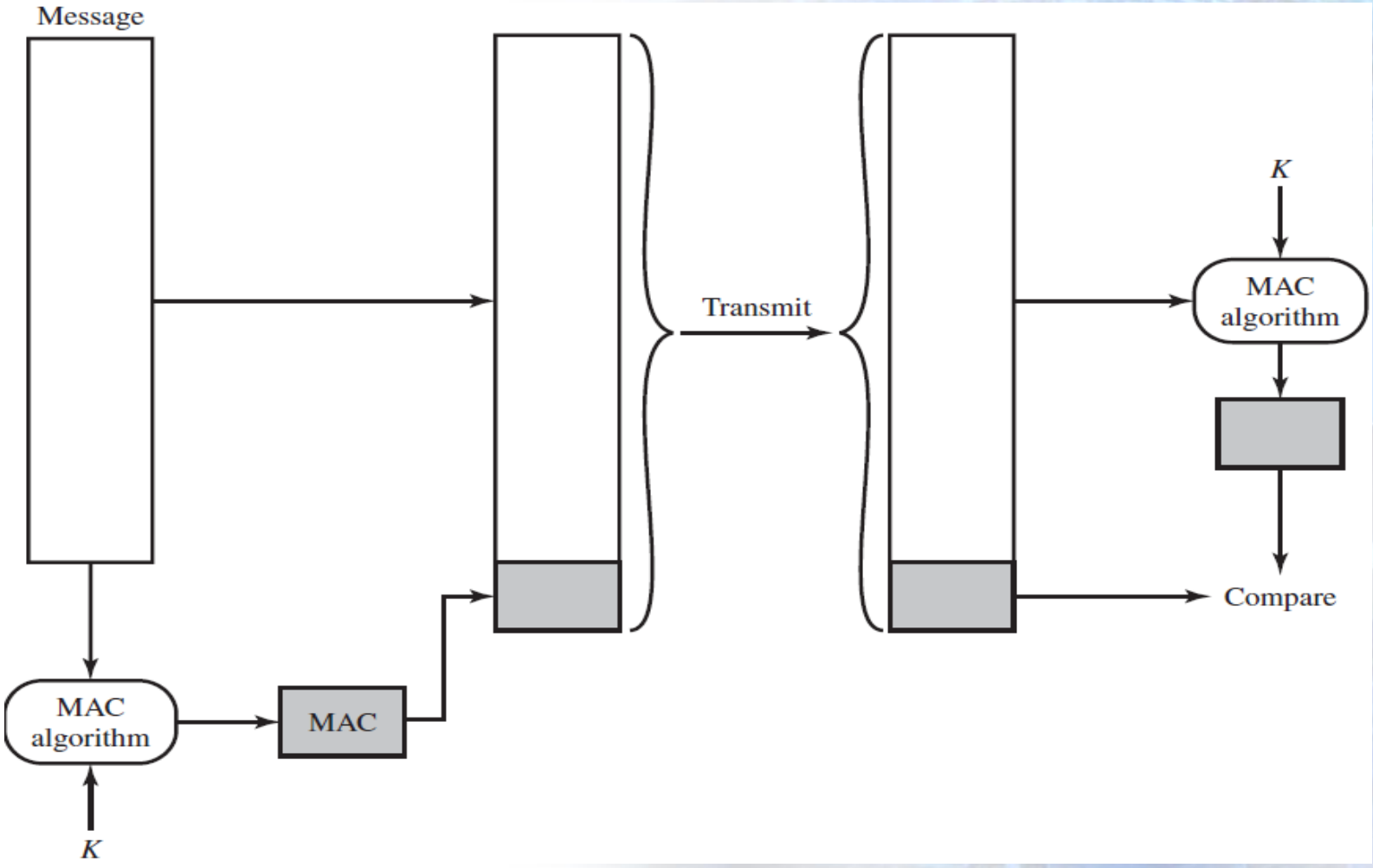
- *"Network Security Essentials : Applications and Standards"*, 2014, by William Stallings.

Cipher-Based MAC

Message Authentication Code (MAC)

- is a technique that involves the use of a secret key to generate a small block of data, known as a message authentication code , that is appended to the message.

Cipher-Based MAC



Cipher-Based MAC

Cipher-based Message Authentication Code (CMAC):

- Is a message authentication code based on AES and triple DES.
- It is specified in NIST Special Publication 800-38B.

Cipher-Based MAC

Message Length is Integer Multiple of Block Size:

- First, let us consider the operation of CMAC when the message is an integer multiple **n** of the cipher block length **b**.

Cipher-Based MAC

- For AES, b = 128, and for triple DES, b=64.
- The message is divided into **n** blocks (M_1,M_2, . . .,M_n).

Cipher-Based MAC

- The algorithm makes use of a k-bit encryption key K and an n-bit key, K_1.
- For AES, the key size k is 128, 192, or 256 bits.
- For triple DES, the key size is 112 or 168 bits.

Cipher-Based MAC

- Lets assume that T is the message authentication code, also referred to as the tag
- Tlen = bit length of T
- $MSB_s(X)$ = the s leftmost bits of the bit string X.

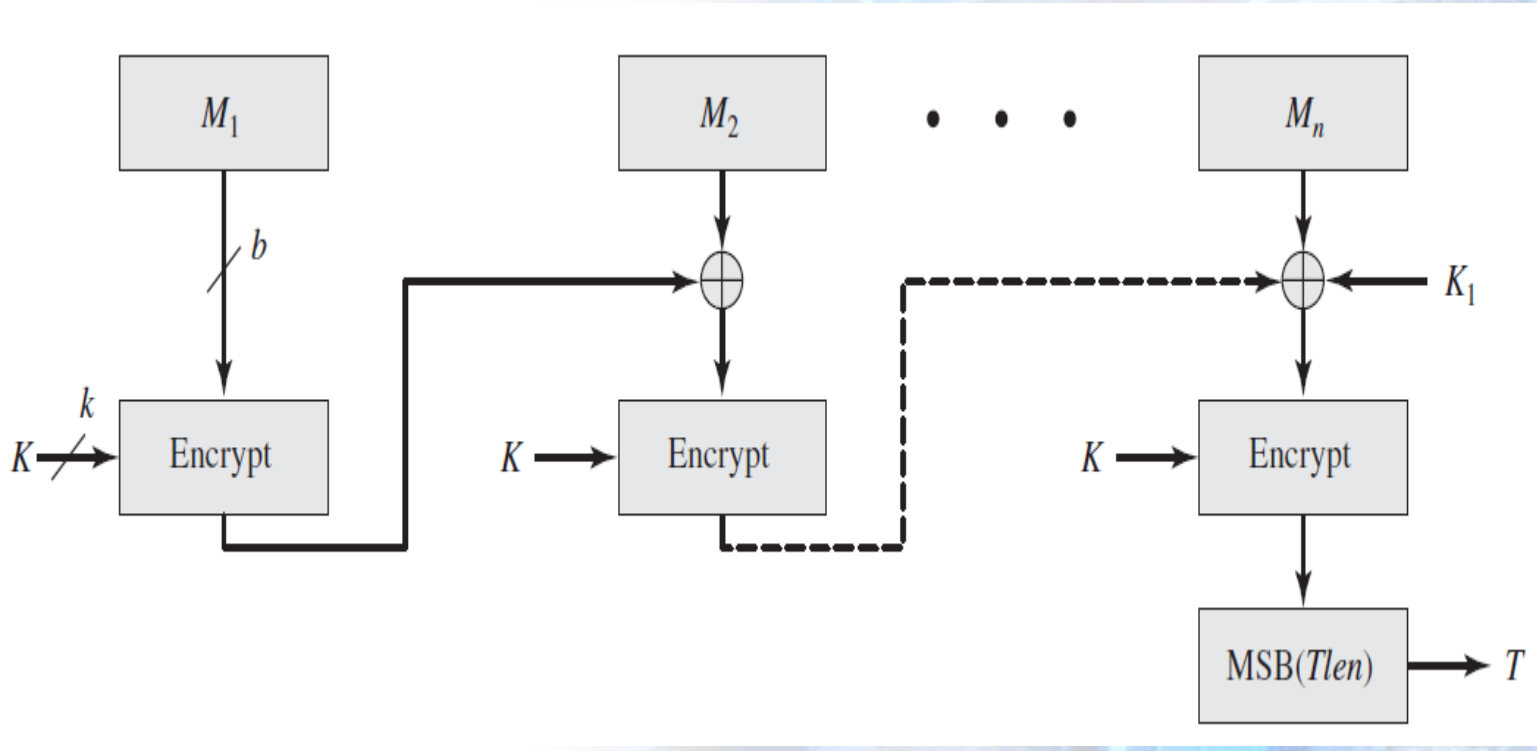
Cipher-Based MAC

Calculation of CMAC

$$
\begin{aligned}
C_1 &= \mathrm{E}(K, M_1) \\
C_2 &= \mathrm{E}(K, [M_2 \oplus C_1]) \\
C_3 &= \mathrm{E}(K, [M_3 \oplus C_2]) \\
&\cdot \\
&\cdot \\
&\cdot \\
C_n &= \mathrm{E}(K, [M_n \oplus C_{n-1} \oplus K_1]) \\
T &= \mathrm{MSB}_{Tlen}(C_n)
\end{aligned}
$$

Cipher-Based MAC

Message Length is Integer Multiple of Block Size



Cipher-Based MAC

Message Length is not Integer Multiple of Block Size:

- In this case, the final block is padded to the right (least significant bits) with a 1 and as many 0s as necessary so that the final block is also of length b.

Cipher-Based MAC

- The CMAC operation then proceeds as before, except that a different n-bit key K_2 is used instead of K_1.
- To generate the two *n*-bit keys, the block cipher is applied to the block that consists entirely of 0 bits.

Cipher-Based MAC

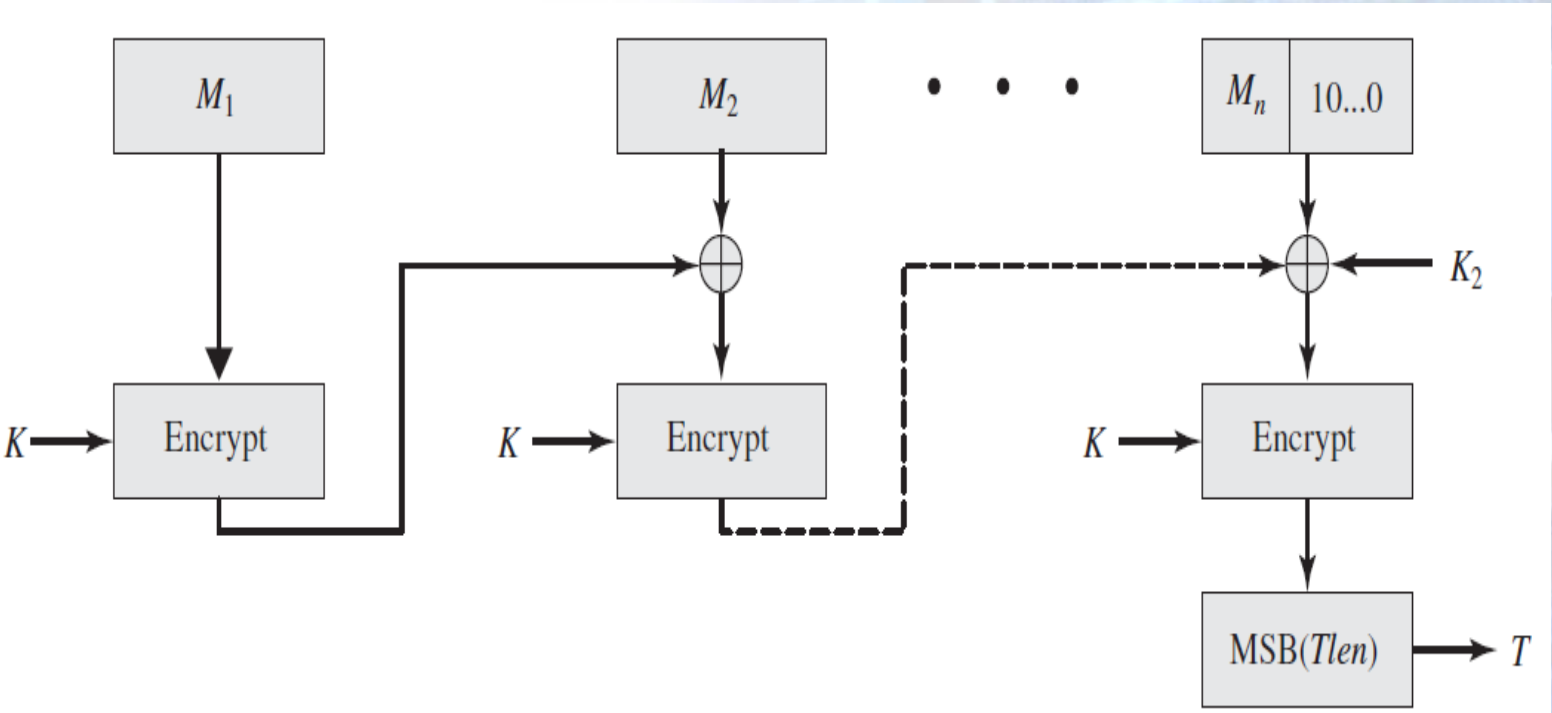
- The first subkey is derived from the resulting ciphertext by a left shift of one bit and, conditionally, by XORing a constant that depends on the block size.

Cipher-Based MAC

- The second subkey is derived in the same manner from the first subkey.

Cipher-Based MAC

Message Length is not Integer Multiple of Block Size



Counter With Cipher Block Chaining-MAC

Network Security

Counter With Cipher Block Chaining-MAC

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe working of Counter with Cipher Block Chaining-MAC.

Counter With Cipher Block Chaining-MAC

Figures and material in this topic have been adapted from

- *"Network Security Essentials : Applications and Standards"*, 2014, by William Stallings.

Counter With Cipher Block Chaining-MAC

- The Counter with Cipher Block Chaining-Message Authentication Code (CCM) mode of operation, was standardized by NIST specifically to support security requirements of IEEE 802.11 WiFi wireless local area networks.

Counter With Cipher Block Chaining-MAC

- It can be used in any networking application requiring authenticated encryption.
- It is defined in NIST SP 800-38C.

Counter With Cipher Block Chaining-MAC

- CCM is a variation of the encrypt-and-MAC approach to authenticated encryption.
- It is referred to as an authenticated encryption mode.

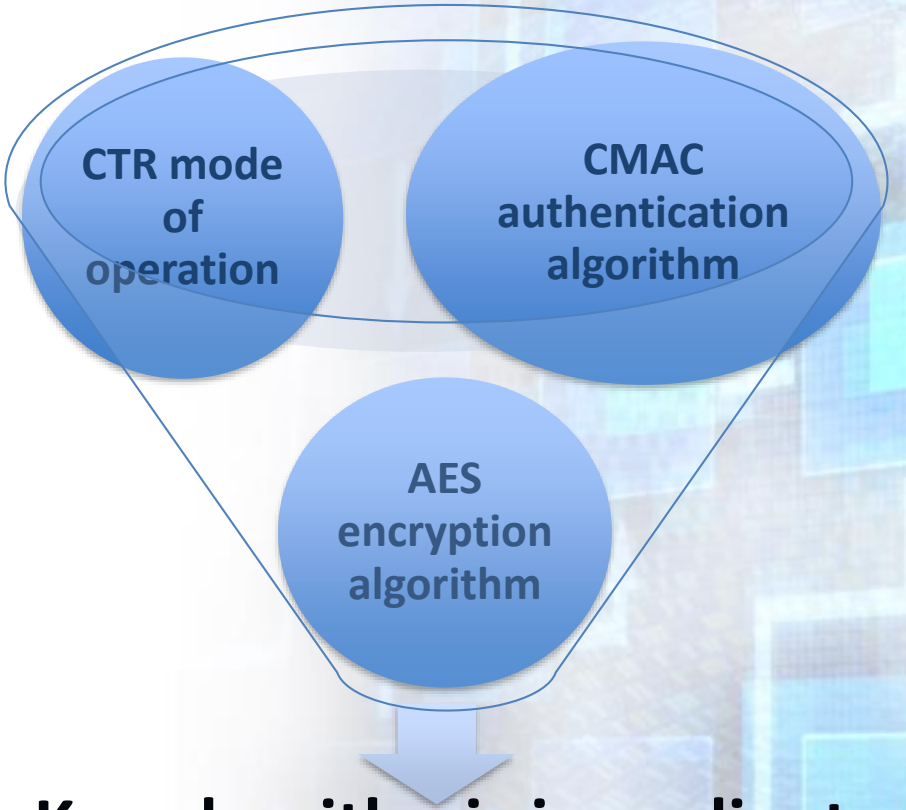
Counter With Cipher Block Chaining-MAC

- “Authenticated encryption” is a term used to describe encryption systems that protect confidentiality and authenticity of communications simultaneously .

Counter With Cipher Block Chaining-MAC

- The key algorithmic ingredients of CCM are AES encryption algorithm, the Counter mode (CTR) of operation, and the CMAC authentication algorithm.
- A single key K is used for both encryption and MAC algorithms.

Counter With Cipher Block Chaining-MAC



Counter With Cipher Block Chaining-MAC

- The input to the CCM encryption process consists of three elements.
- 1. Data that will be both authenticated and encrypted. This is the plaintext message P of data block.

Counter With Cipher Block Chaining-MAC

- 2. Associated data A that will be authenticated but not encrypted.
- An example is a protocol header that must be transmitted in the clear for proper protocol operation but which needs to be authenticated.

Counter With Cipher Block Chaining-MAC

- 3. A nonce N that is assigned to the payload and the associated data.
- This is a unique value that is different for every instance during lifetime of a protocol association and is intended to prevent replay attacks.

Counter With Cipher Block Chaining-MAC

Authentication

- For authentication, the input includes the nonce, the associated data, and the plaintext.
- This input is formatted as a sequence of blocks B_0 through B_r.

Counter With Cipher Block Chaining-MAC

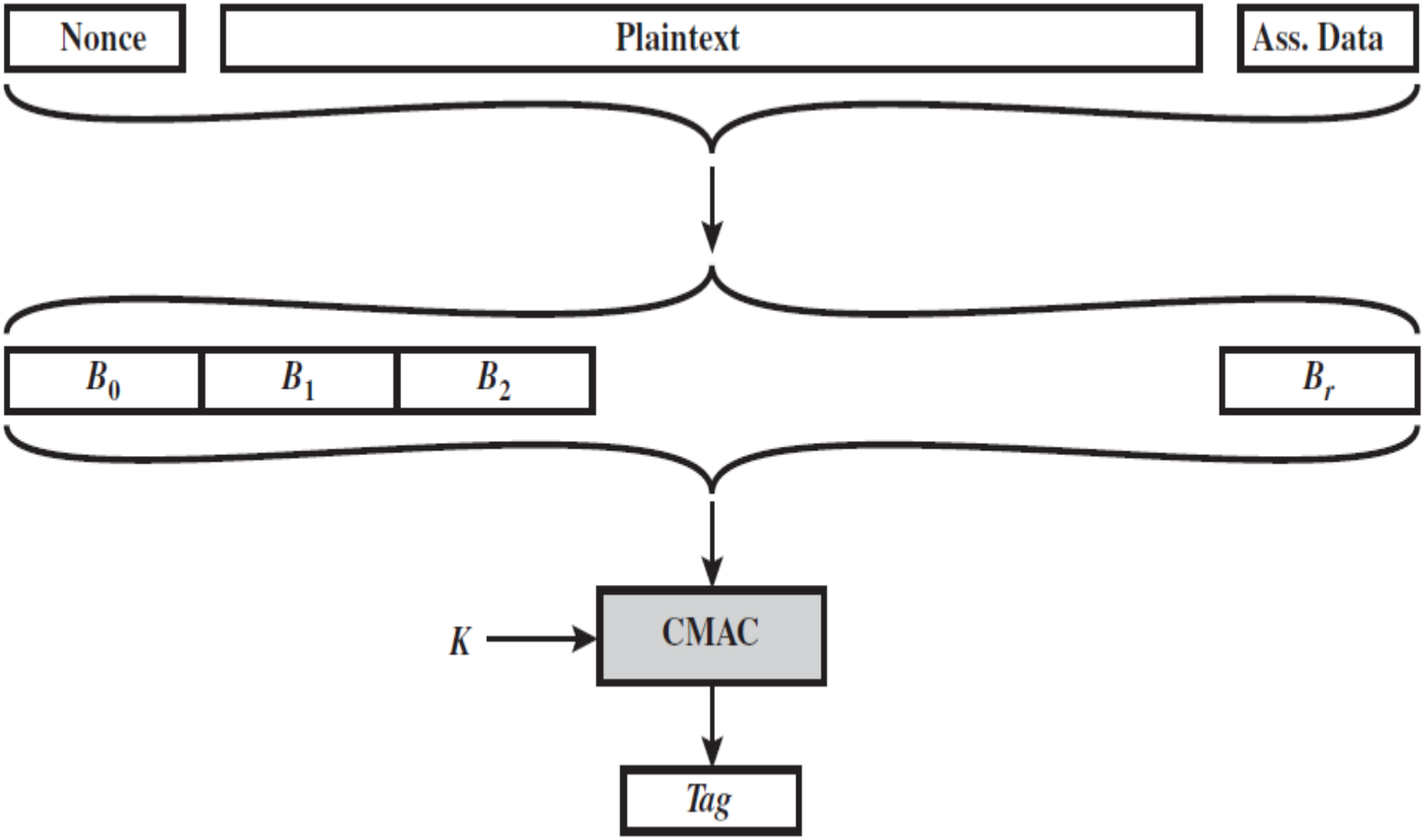
- The first block contains the nonce plus some formatting bits that indicate the lengths of the N, A, and P elements.
- This is followed by zero or more blocks that contain *A*, followed by zero of more blocks that contain *P.*

Counter With Cipher Block Chaining-MAC

- The resulting sequence of blocks serves as input to the CMAC algorithm, which produces a MAC value with length *Tlen*, which is less than or equal to the block length.

Counter With Cipher Block Chaining-MAC

Authentication



Counter With Cipher Block Chaining-MAC

Encryption

- For encryption, a sequence of counters is generated that must be independent of the nonce.
- The authentication tag is encrypted in CTR mode using the single counter Ctr0 .

Counter With Cipher Block Chaining-MAC

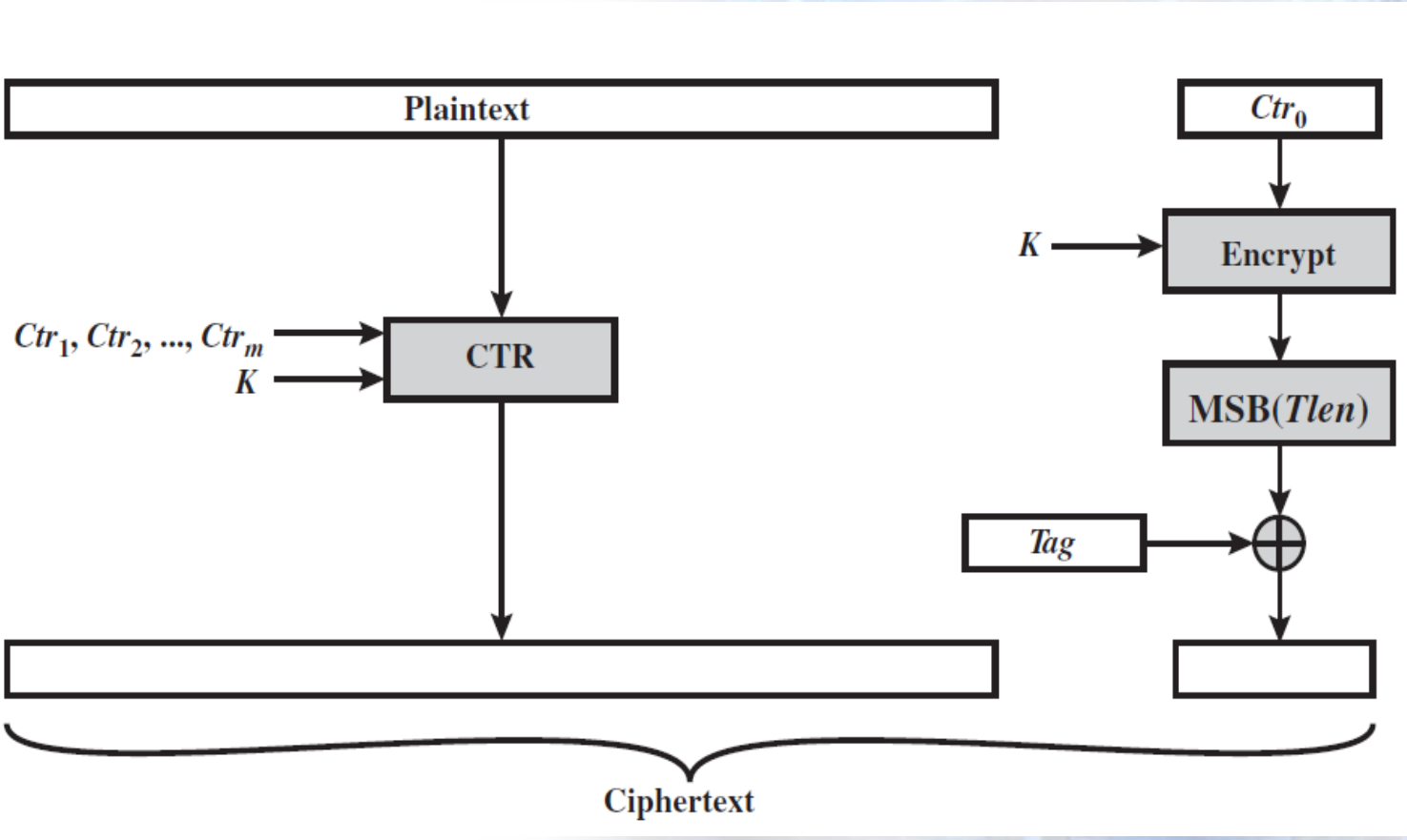
- The Tlen most significant bits of the output are XORed with the tag to produce an encrypted tag.
- The remaining counters are used for the CTR mode encryption of the plaintext.

Counter With Cipher Block Chaining-MAC

- The encrypted plaintext is concatenated with the encrypted tag to form the ciphertext output.

Counter With Cipher Block Chaining-MAC

Encryption



Public-Key Encryption Structure

Network Security

Public-Key Encryption Structure

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain structure of Public-key encryption.

Public-Key Encryption Structure

Figures and material in this topic have been adapted from

- *"Network Security Essentials : Applications and Standards"*, 2014, by William Stallings.

Public-Key Encryption Structure

- Public-key encryption, first publicly proposed by Diffie and Hellman in 1976.

Public-Key Encryption Structure

- Public-key algorithms are based on mathematical functions rather than on simple operations on bit patterns, such as are used in symmetric encryption algorithms.

Public-Key Encryption Structure

- More important, public-key cryptography is asymmetric, involving the use of two separate keys—in contrast to the symmetric conventional encryption, which uses only one key.

Public-Key Encryption Structure

- The use of two keys has profound consequences in the areas of confidentiality, key distribution, and authentication.

Public-Key Encryption Structure

Structure

- A public-key encryption scheme has six ingredients.
- **Plaintext:** This is the readable message or data that is fed into the algorithm as input.

Public-Key Encryption Structure

- **Encryption algorithm:** The encryption algorithm performs various transformations on the plaintext.

Public-Key Encryption Structure

- **Public and private key:** This is a pair of keys; one is used for encryption, the other is used for decryption.
- The exact transformations performed by the encryption algorithm depend on the public or private key.

Public-Key Encryption Structure

- The public key of the pair is made public for others to use, while the private key is known only to its owner.

Public-Key Encryption Structure

- **Ciphertext:** This is the scrambled message produced as output.
- It depends on the plaintext and the key.
- For a given message, two different keys will produce two different ciphertexts.

Public-Key Encryption Structure

- **Decryption algorithm:** This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

Public-Key Encryption Structure

- A general-purpose public-key cryptographic algorithm relies on one key for encryption and a different but related key for decryption.

Public-Key Encryption Structure

Working

- 1. Each user generates a pair of keys to be used for the encryption and decryption of messages.

Public-Key Encryption Structure

- 2. Each user places one of the two keys in a public register or other accessible file.
- This is the public key. The companion key is kept private.
- Each user maintains a collection of public keys obtained from others.

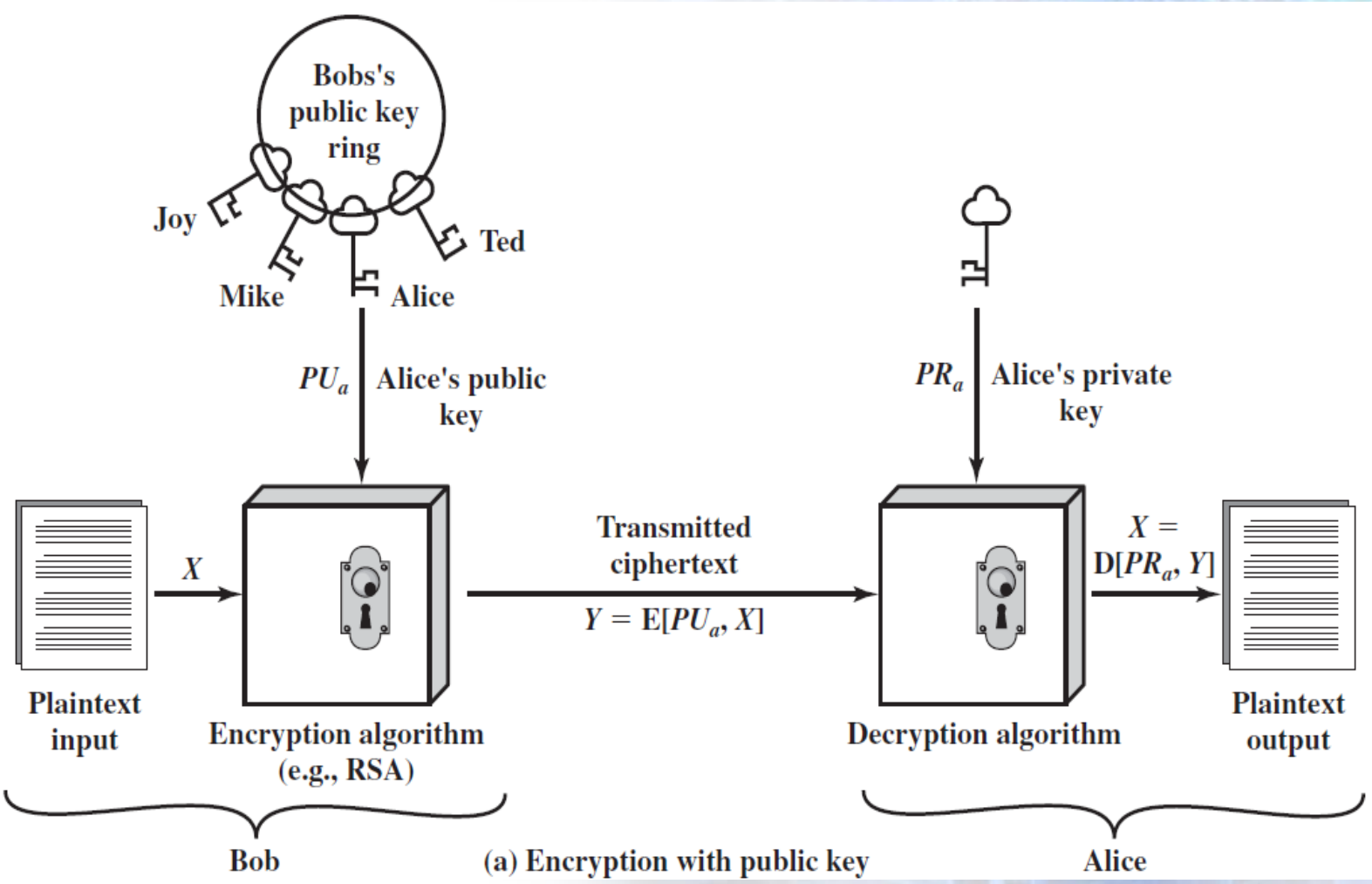
Public-Key Encryption Structure

- 3. If Bob wishes to send a private message to Alice, Bob encrypts the message using Alice's public key.

Public-Key Encryption Structure

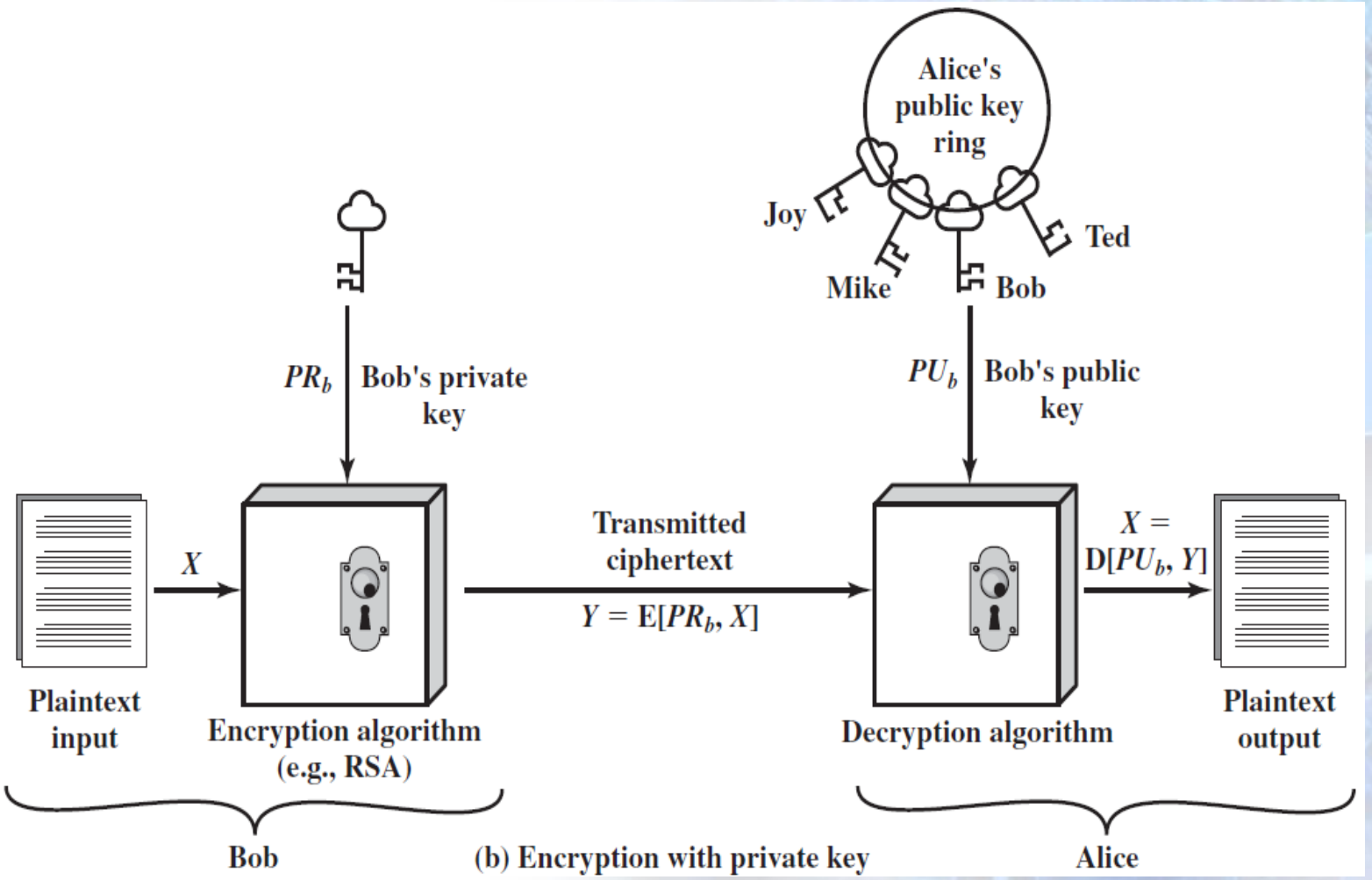
- 4. When Alice receives the message, she decrypts it using her private key.
- No other recipient can decrypt the message because only Alice knows Alice's private key.

Public-Key Encryption Structure



(a) Encryption with public key

Public-Key Encryption Structure



Applications/Requirements for Public-key

Network Security

Applications/Requirements for Public-key

Objectives of the Topic

- After completing this topic, a student will be able to
  - Describe applications and requirements for public-key cryptosystems.

Applications/Requirements for Public-key

Figures and material in this topic have been adapted from

- *"Network Security Essentials : Applications and Standards"*, 2014, by William Stallings.

Applications/Requirements for Public-key

- Public-key cryptography is asymmetric, involving the use of two separate keys.

Applications/Requirements for Public-key

- With this approach, all participants have access to public keys, and private keys are generated locally by each participant and therefore need never be distributed.

Applications/Requirements for Public-key

- As long as a user protects his or her private key, incoming communication is secure.
- At any time, a user can change the private key and publish its companion public key to replace the old public key.

Applications/Requirements for Public-key

Misconceptions

- 1. Public-key encryption is more secure from cryptanalysis than conventional encryption.

Applications/Requirements for Public-key

- 2. Public-key encryption is a general-purpose technique that has made conventional encryption obsolete.

Applications/Requirements for Public-key

- 3. There is a feeling that key distribution is trivial when using public-key encryption, compared to the rather cumbersome handshaking involved with key distribution centers for conventional encryption.

Applications/Requirements for Public-key

Applications

- Public-key systems are characterized by the use of a cryptographic type of algorithm with two keys, one held private and one available publicly.

Applications/Requirements for Public-key

- Depending on the application, the sender uses either the sender's private key, the receiver's public key, or both to perform some type of cryptographic function.

Applications/Requirements for Public-key

- We can classify the use of public-key cryptosystems into three categories:
- Encryption/Decryption
- Digital Signatures
- Key Exchange

Applications/Requirements for Public-key

- **Encryption/decryption:**
- The sender encrypts a message with the recipient’s public key.

Applications/Requirements for Public-key

- **Digital signature:**
- The sender “signs” a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.

Applications/Requirements for Public-key

- **Key exchange:**
- Two sides cooperate to exchange a session key.
- Several different approaches are possible, involving the private key(s) of one or both parties.

Applications/Requirements for Public-key

- Some public-key algorithms are suitable for all three applications, whereas others can be used only for one or two of these applications.

Applications/Requirements for Public-key

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

Applications/Requirements for Public-key

Requirements

- A public-key cryptosystem depends on a cryptographic algorithm based on two related keys.

Applications/Requirements for Public-key

- Diffie and Hellman lay out the conditions that such algorithms must fulfill
- 1. It is computationally easy for a party B to generate a pair (public key PU_b, private key PR_b).

Applications/Requirements for Public-key

- 2. It is computationally easy for a sender A, knowing the public key and the message to be encrypted, M, to generate the corresponding ciphertext:
- $C = E(PU_b, M)$

Applications/Requirements for Public-key

- 3. It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message:
- $M = D(PR_b, C) = D[PR_b, E(PU_b, M)]$

Applications/Requirements for Public-key

- 4. It is computationally infeasible for an opponent, knowing the public key, PU_b, to determine the private key, PR_b.

Applications/Requirements for Public-key

- 5. It is computationally infeasible for an opponent, knowing the public key, PU_b, and a ciphertext, C, to recover the original message, M.

Applications/Requirements for Public-key

End

- 6. Either of the two related keys can be used for encryption, with the other used for decryption.
- $M = D[PU_b, E(PR_b, M)] = D[PR_b, E(PU_b,M)]$
- This requirement is useful but not necessary.

The RSA Public-Key Encryption Algorithm

Network Security

The RSA Public-Key Encryption Algorithm

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain working of the RSA Public-Key encryption algorithm.

The RSA Public-Key Encryption Algorithm

Figures and material in this topic have been adapted from

- *"Network Security Essentials : Applications and Standards"*, 2014, by William Stallings.

The RSA Public-Key Encryption Algorithm

- RSA is the best known, and widely used general public key encryption scheme.
- It was first published by Rivest, Shamir & Adleman of MIT in 1978.

The RSA Public-Key Encryption Algorithm

- The RSA scheme 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. That is, n is less than 2^{1024}.

The RSA Public-Key Encryption Algorithm

- RSA is based on exponentiation in a finite (Galois) field over integers modulo a prime.
- Security due to cost of factoring large numbers.

The RSA Public-Key Encryption Algorithm

Description

- Plaintext is encrypted in blocks, with each block having a binary value less than some number n.
- The block size must be less than or equal to $\log_2(n) + 1$; in practice, the block size is i bits, where $2^i < n \leq 2^{i+1}$.

The RSA Public-Key Encryption Algorithm

- Encryption and decryption are of the following form, for some plaintext block M and ciphertext block C.
- $C = M^e \bmod n$
- $M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$

The RSA Public-Key Encryption Algorithm

- 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.

The RSA Public-Key Encryption Algorithm

- Here, the public key of PU = {e, n} and the private key of PR = {d, n}.
- For this algorithm to be satisfactory for public-key encryption, following requirements must be met:

The RSA Public-Key Encryption Algorithm

- 1. It is possible to find values of e, d, and 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.

The RSA Public-Key Encryption Algorithm

Summary

Key Generation

Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calculate d	$de \bmod \phi(n) = 1$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod{n}$

Decryption

Ciphertext:	C
Plaintext:	$M = C^d \pmod{n}$

The RSA Public-Key Encryption Algorithm

Example

- **Key Generation**
- 1. Select two prime numbers, p= 17 and q= 11.
- 2. Calculate n = pq = 17 × 11 = 187.
- 3. Calculate $\varphi(n) = (p-1)(q-1) = 16 \times 10 = 160$.

The RSA Public-Key Encryption Algorithm

- 4. Select e such that e is relatively prime to $\varphi(n) = 160$ and less than $\varphi(n)$; we choose e=7.
- 5. Determine d such that de mod 160 =1 and d < 160.The correct value is d=23, because 23 × 7=161= (1 × 160)+1.

The RSA Public-Key Encryption Algorithm

- The resulting keys are public key PU={7, 187} and private key PR= {23, 187}.
- **Encryption**
- Lets use these keys for a plaintext input of *M* = 88.
- Here, we need to calculate C= 88^7 mod 187.

The RSA Public-Key Encryption Algorithm

- **Decryption**
- We need to calculate $M = 11^{23} \bmod 187$
- Above expressions can be evaluated by exploiting the properties of modular arithmetic.

The RSA Public-Key Encryption Algorithm

The Security of RSA

- There are two possible approaches to defeating the RSA algorithm.
- The first is the brute-force approach: Try all possible private keys.

The RSA Public-Key Encryption Algorithm

- Thus, the larger the number of bits in e and d, the more secure the algorithm.
- However, because the calculations involved are complex, the larger the size of the key, the slower the system will run.

The RSA Public-Key Encryption Algorithm

- Most discussions of the cryptanalysis of RSA have focused on the task of factoring n into its two prime factors.
- For a large n with large prime factors, factoring is a hard problem.

The RSA Public-Key Encryption Algorithm

- A large key size such as a 1024-bit key size (about 300 decimal digits) is considered strong enough for virtually all applications.

End