Virtual University of Pakistan
FEDERAL GOVERNMENT UNIVERSITY

COMPLETE SLIDES

Network Security

CS315

Order Your VU Books Now

Get all Virtual University books in high-quality hard copy, delivered right to your doorstep. Visit vubookshoppk.com

Order Now

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.

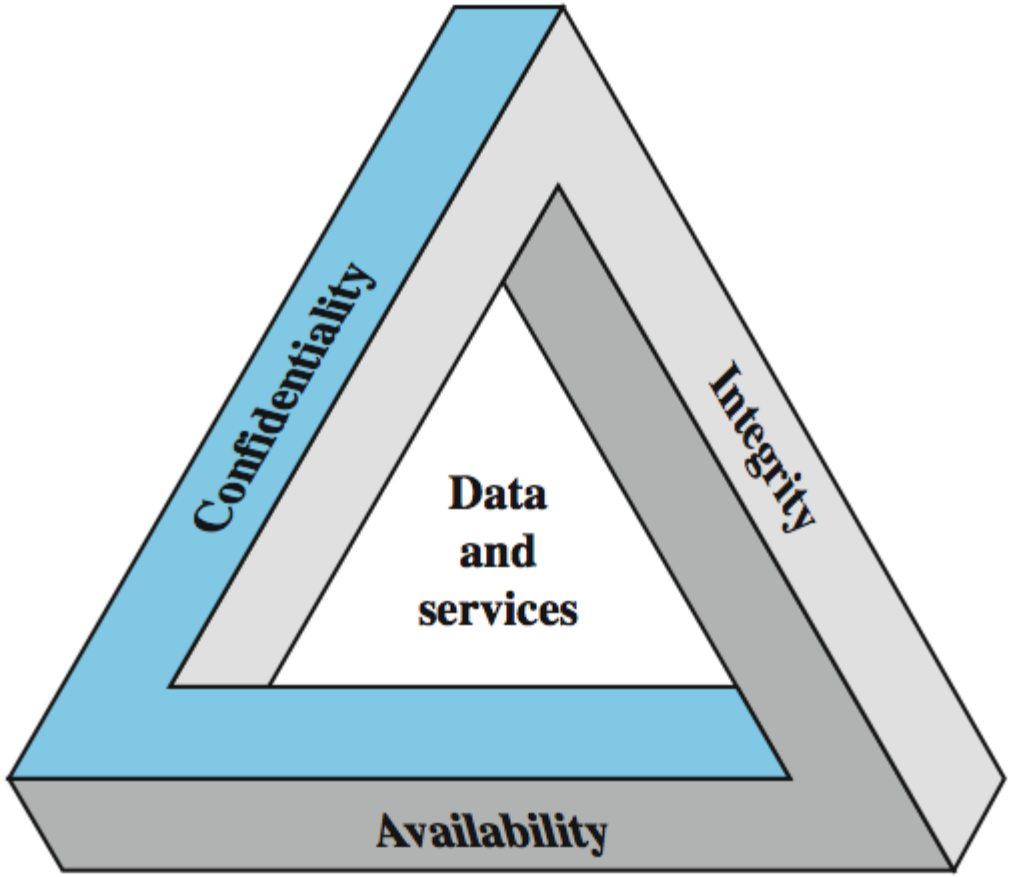
Definition Of Computer Security

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

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

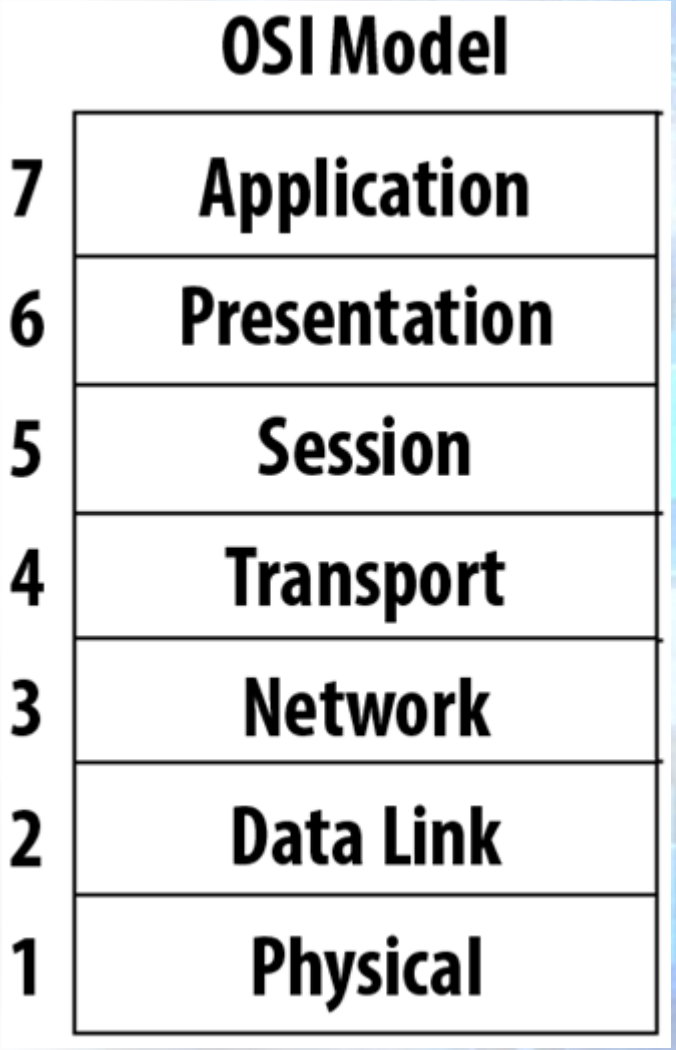
- (ITU-T) is a United Nations sponsored agency that develops standards, called Recommendations, relating to telecommunications and to open systems interconnection (OSI).

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.

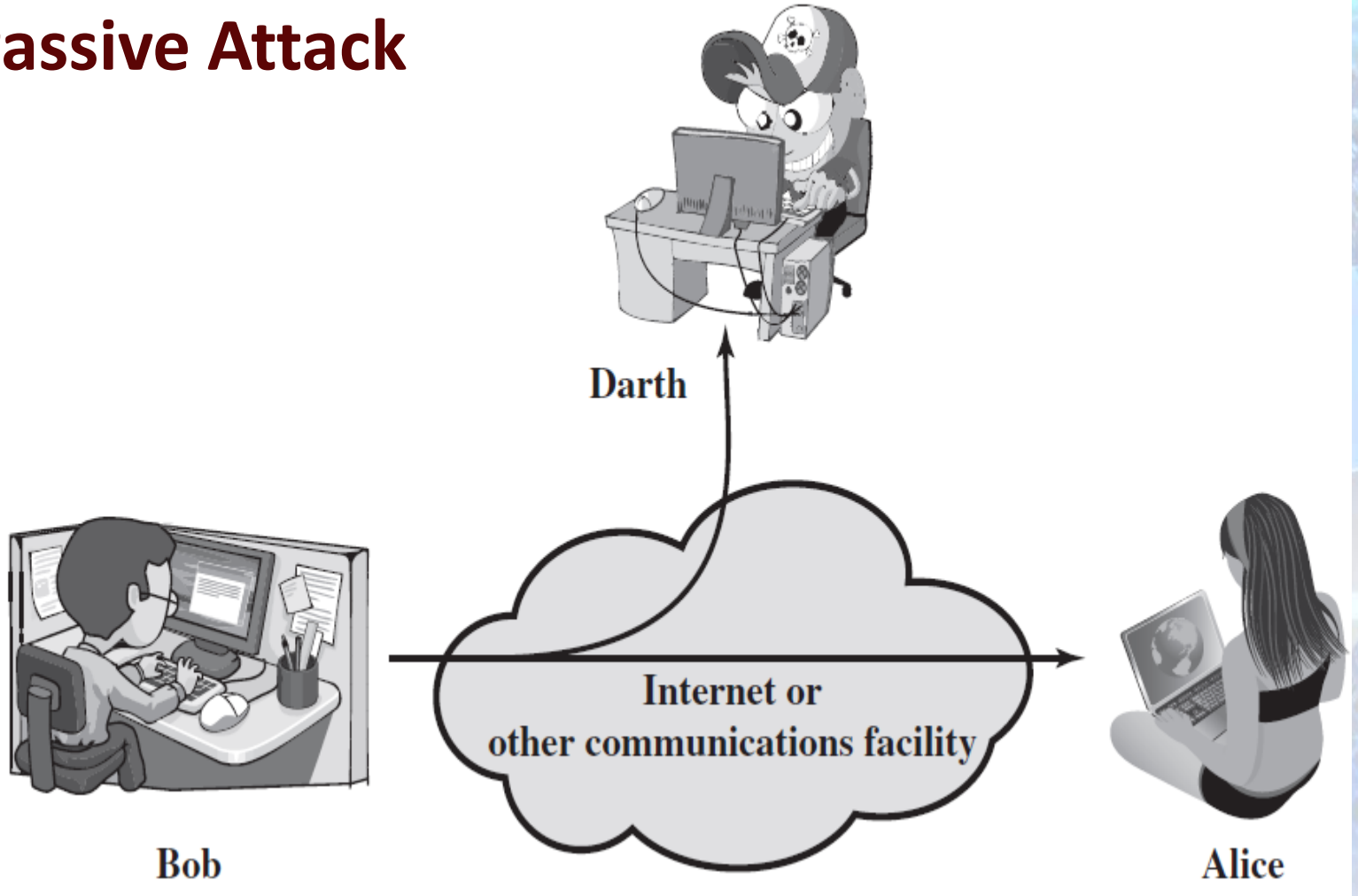
Security Attacks

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.

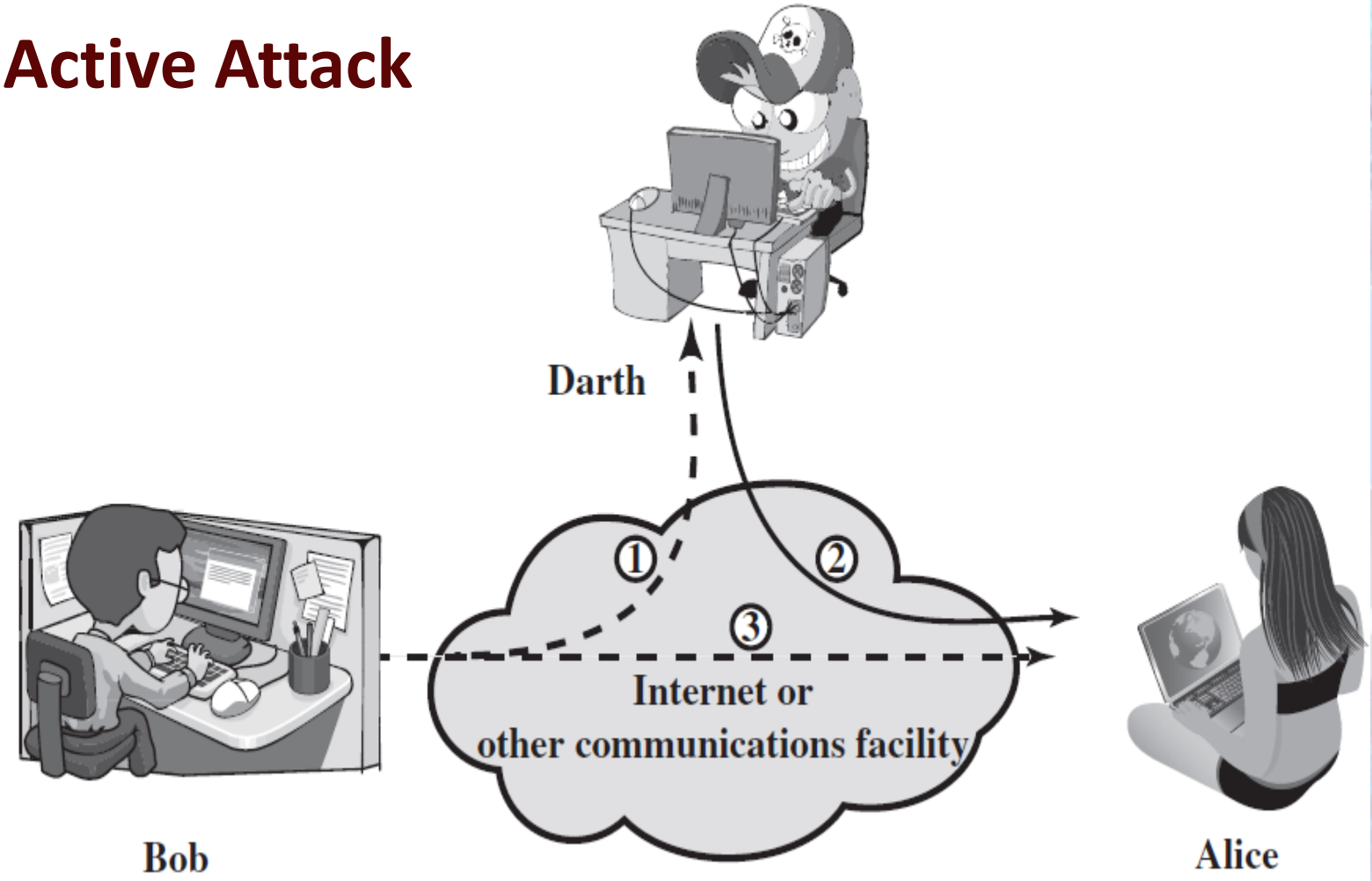
Security Attacks

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

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



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

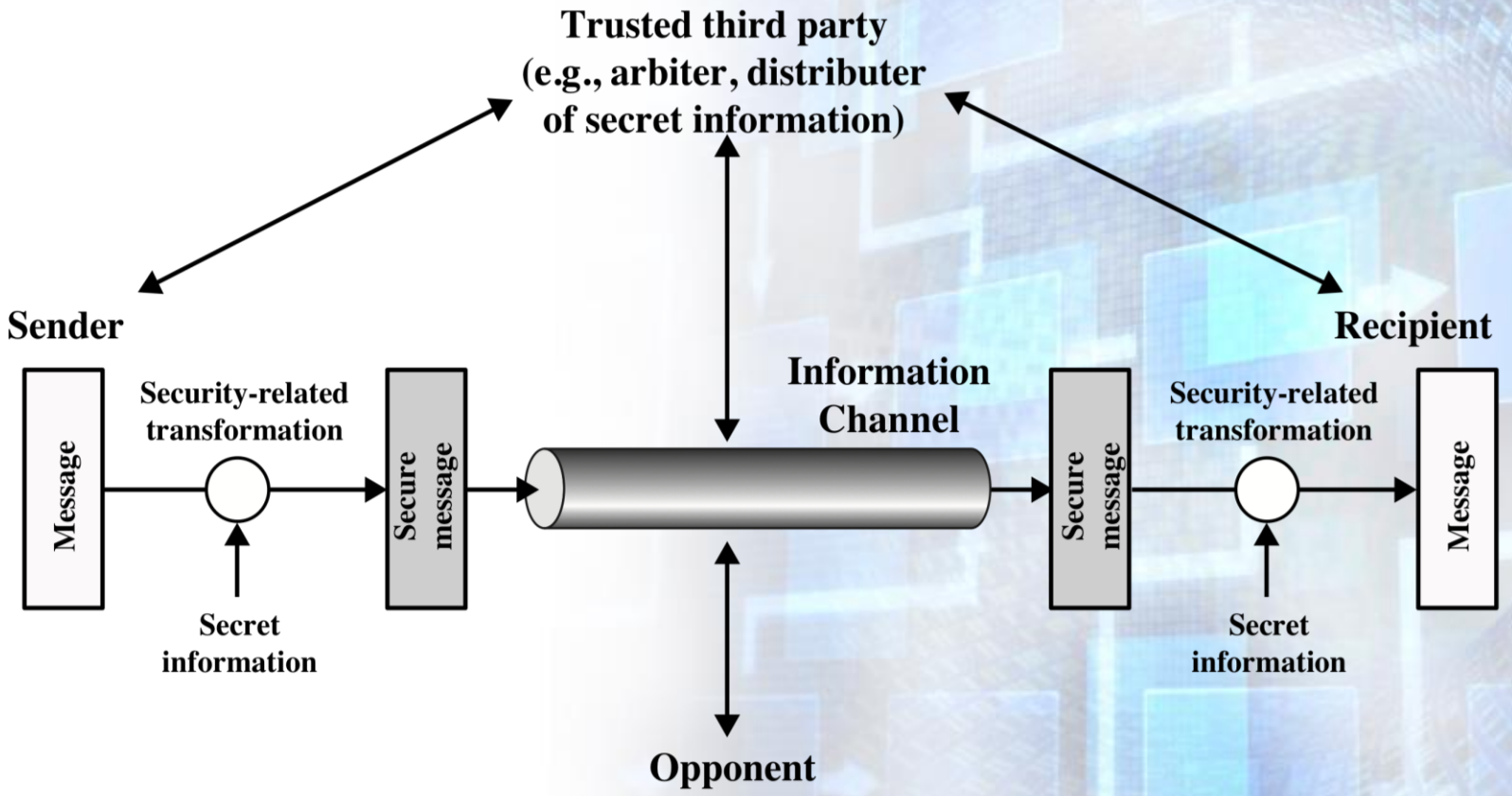


Figure 1.2 Model for Network Security

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

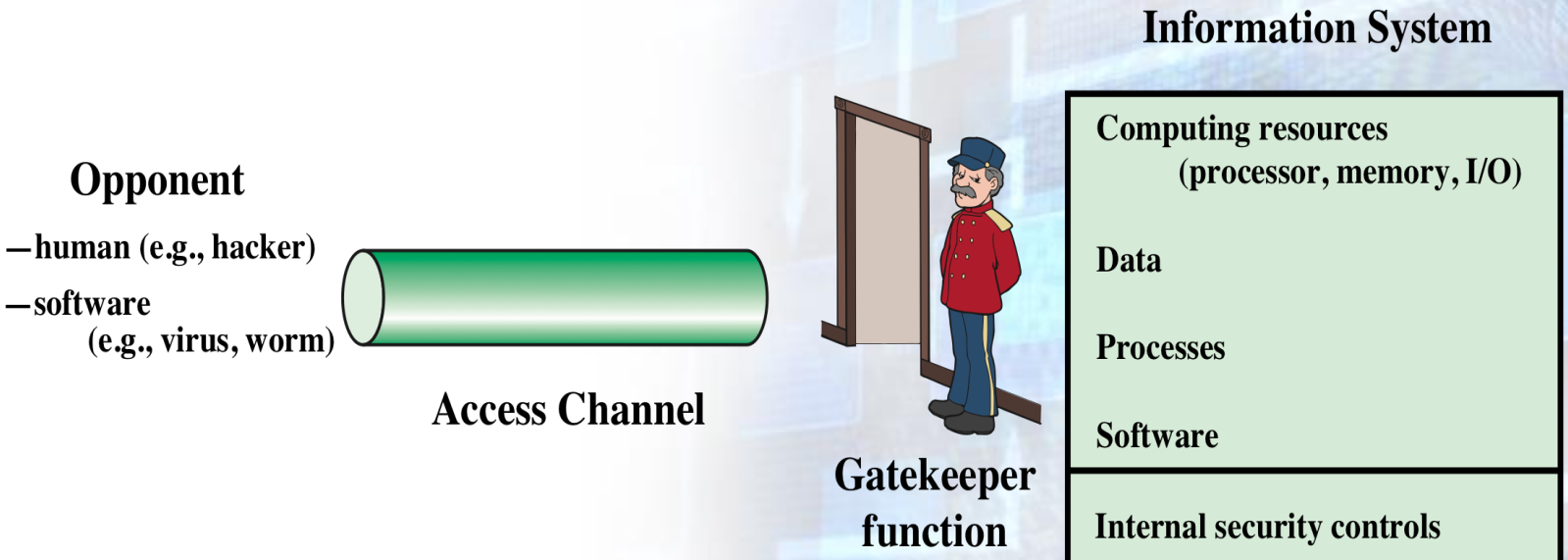


Figure 1.3 Network Access Security Model

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

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

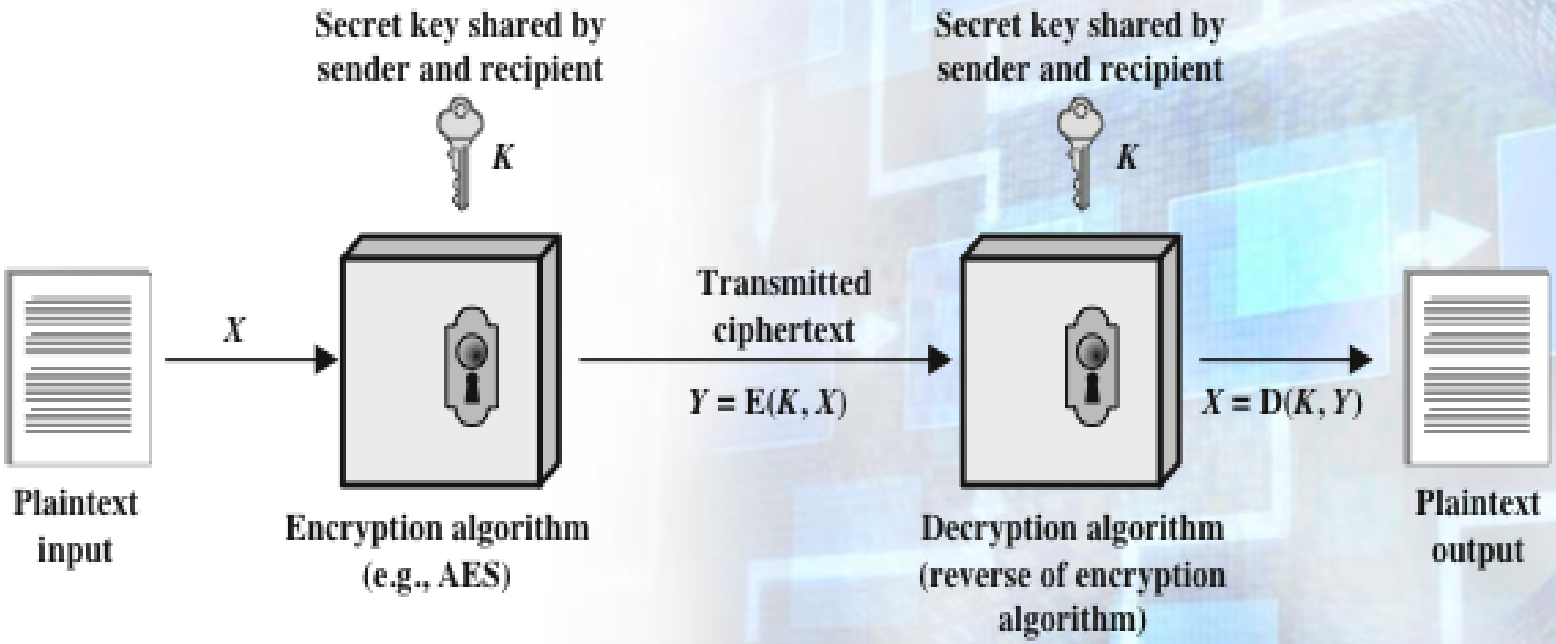


Figure 2.1 Simplified Model of Symmetric Encryption

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

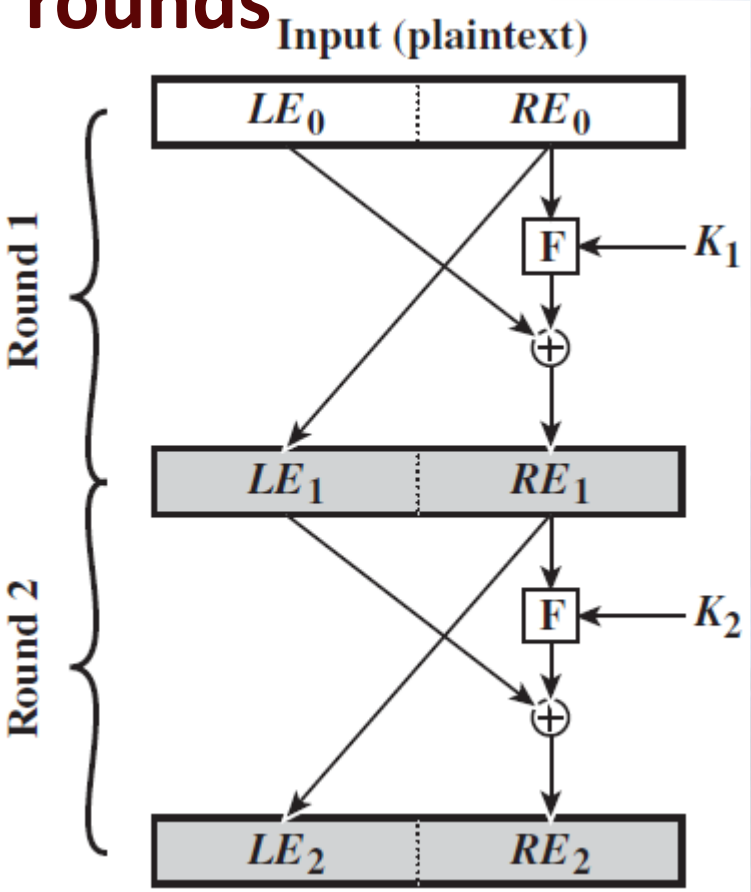
- In a given round, a substitution is performed on the left half of the data.
- Apply a round function F to the right half of the data and then take XOR of the output of that function and the left half of the data.

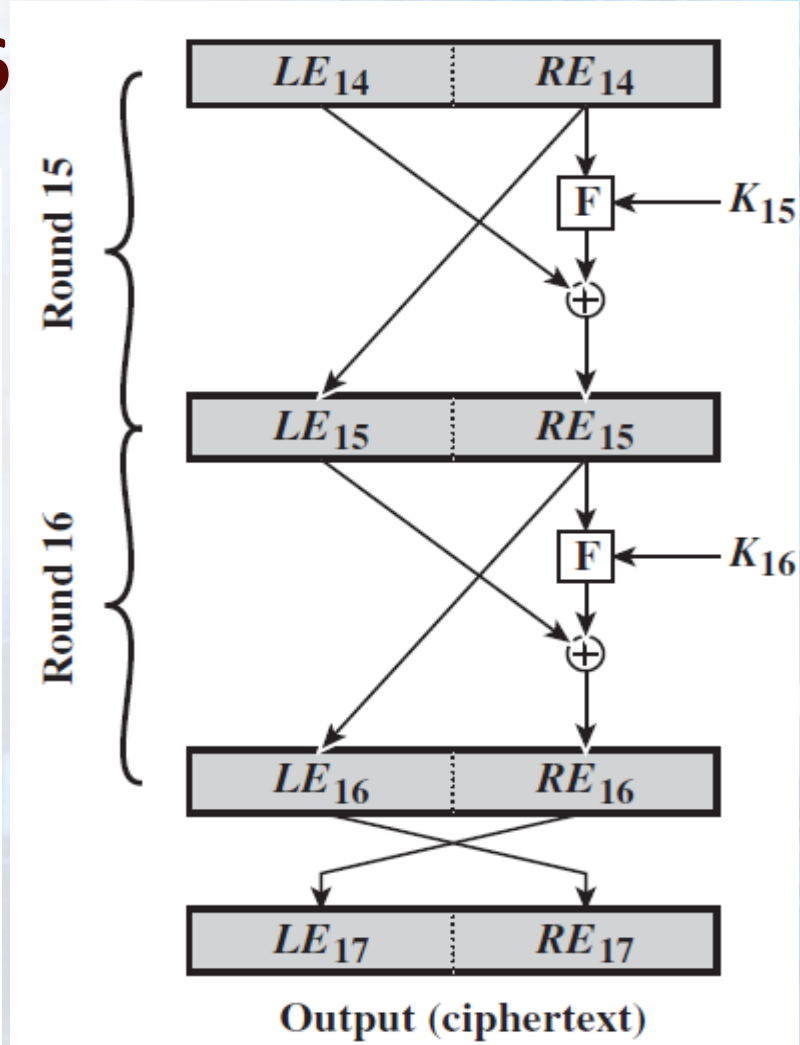
Feistel Cipher Structure

- 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





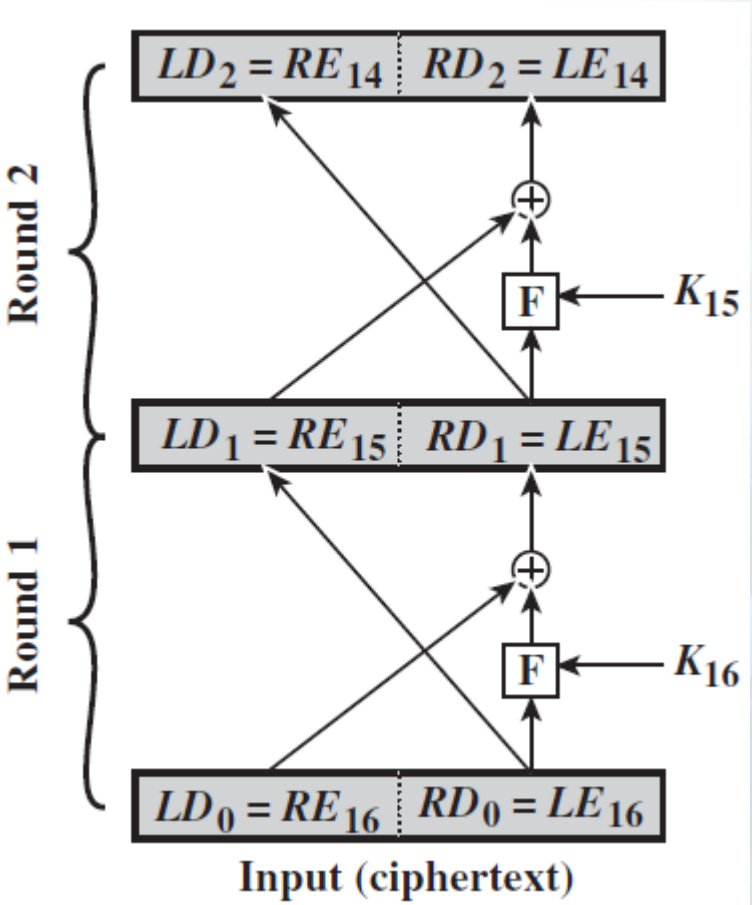
Feistel Cipher Structure

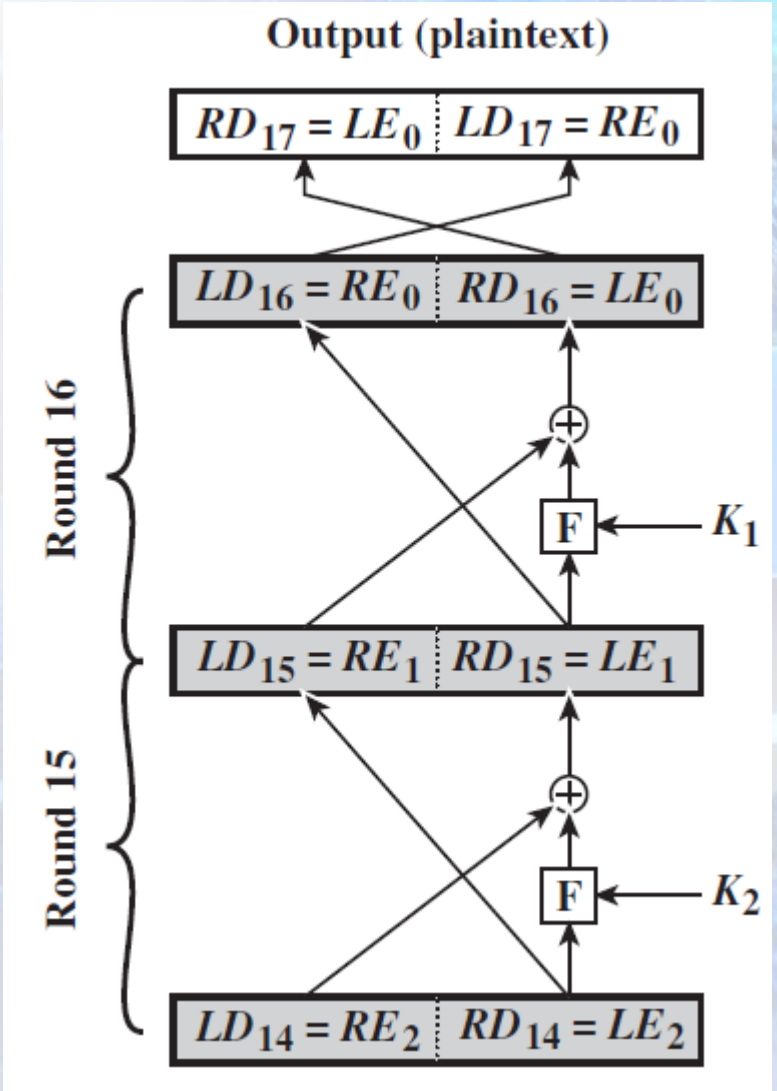
Feistel Decryption

- 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.

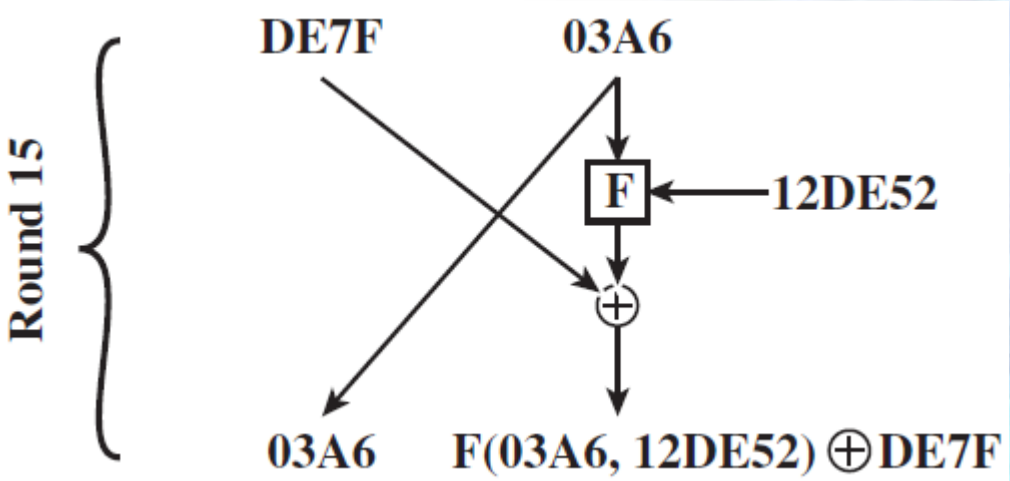
Feistel Cipher Structure

Feistel Decryption



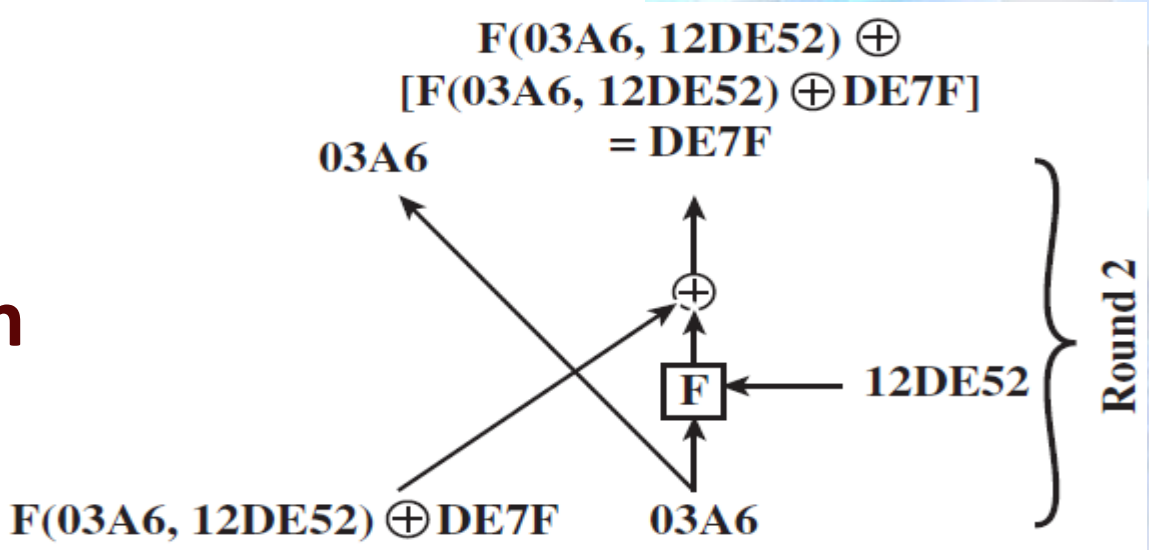


Feistel Cipher Structure



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.

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

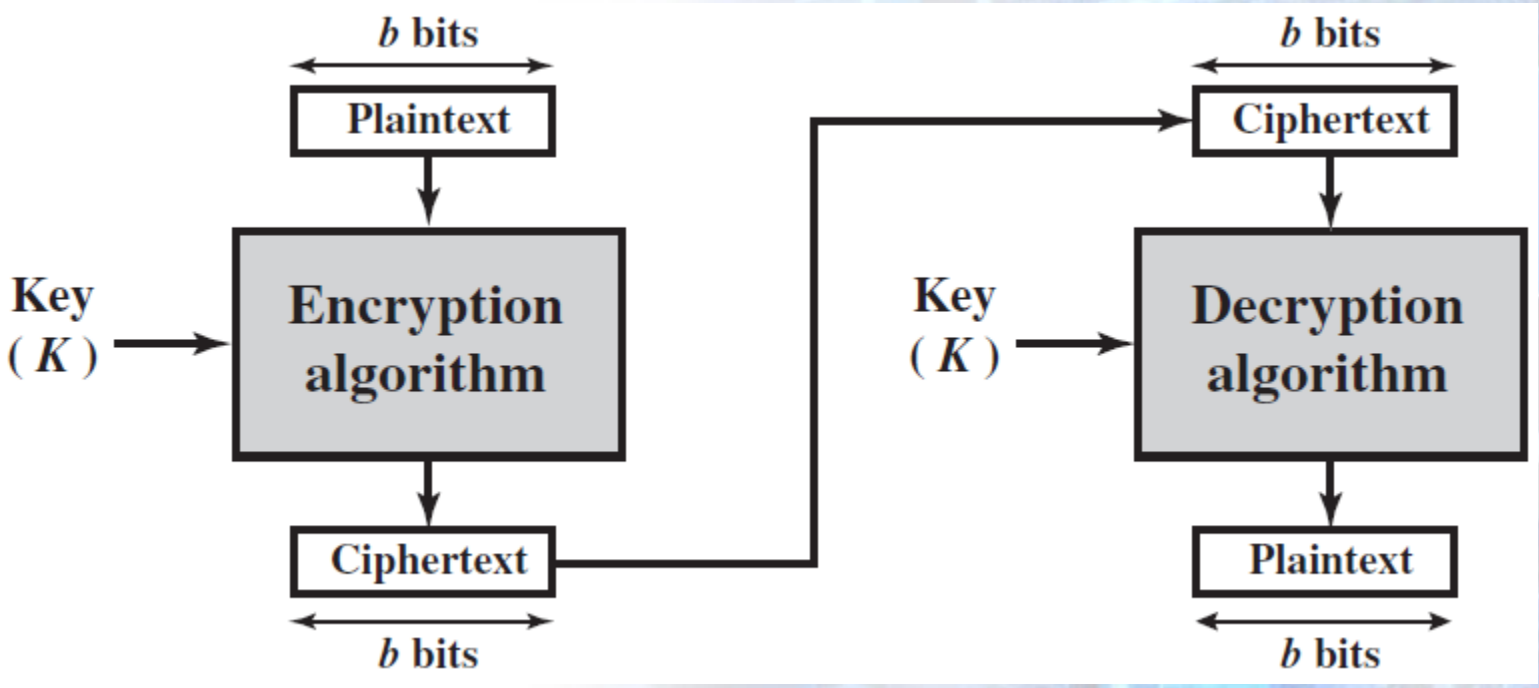
- adapted from *"Network Security Essentials: Applications and Standards"*, 2014, by William Stallings.

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

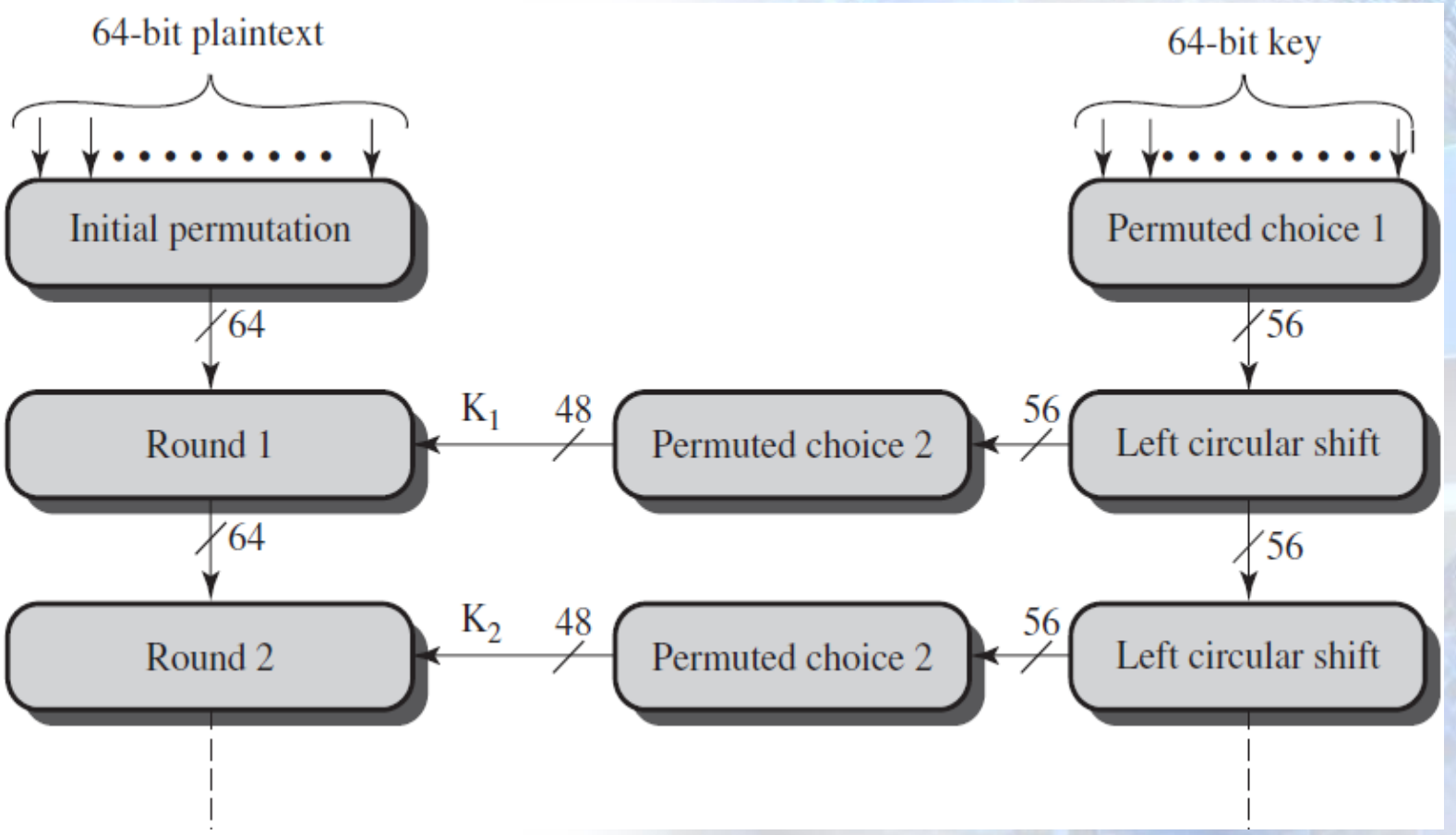
- Initially, the key is passed through a permutation function.
- Then, for each of the 16 rounds, a subkey (K_i) is produced by the combination of a left circular shift and a permutation.

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.

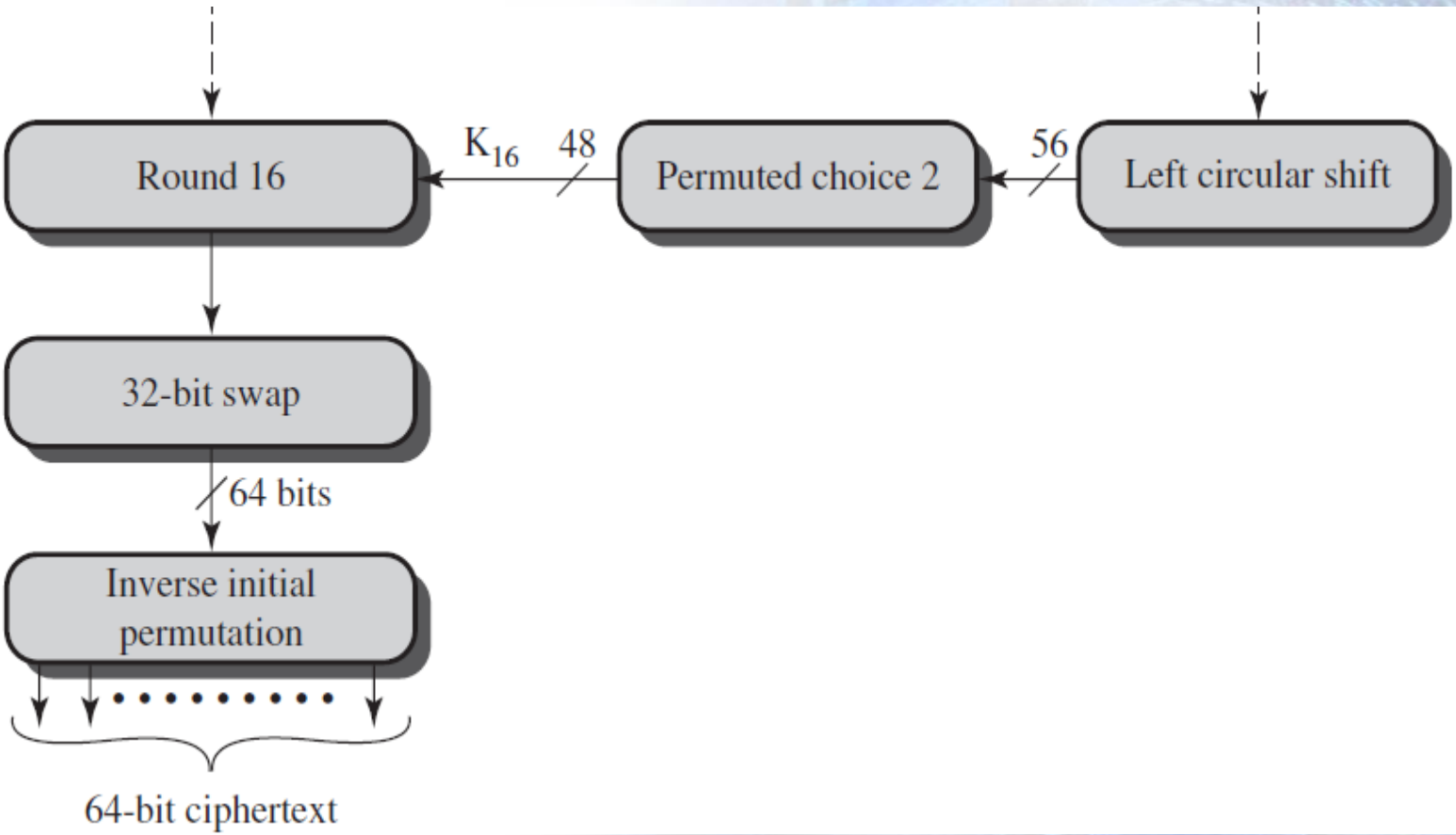
Data Encryption Standard (DES)

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

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

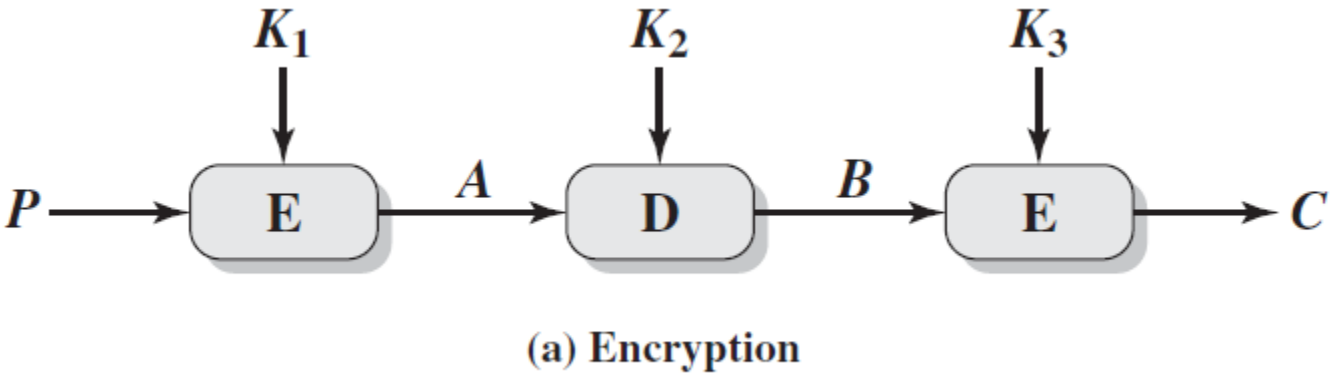
- Given a plaintext P, ciphertext C is generated as
- $C =$

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

- 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

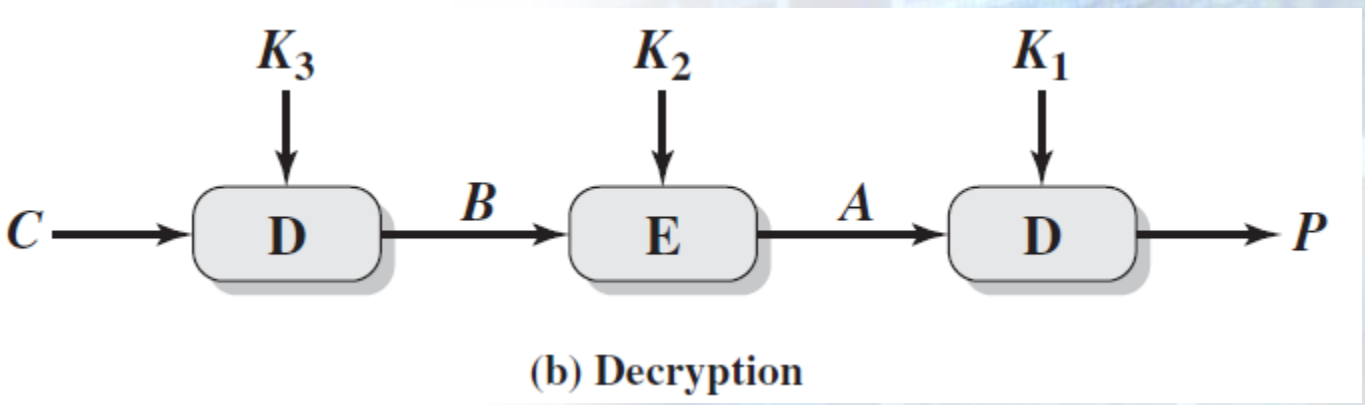
Triple DES

- 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 = 5.8×10^{33} years.
- Time required if PC works at 10^{13} decryptions/s, then 5.8×10^{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

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

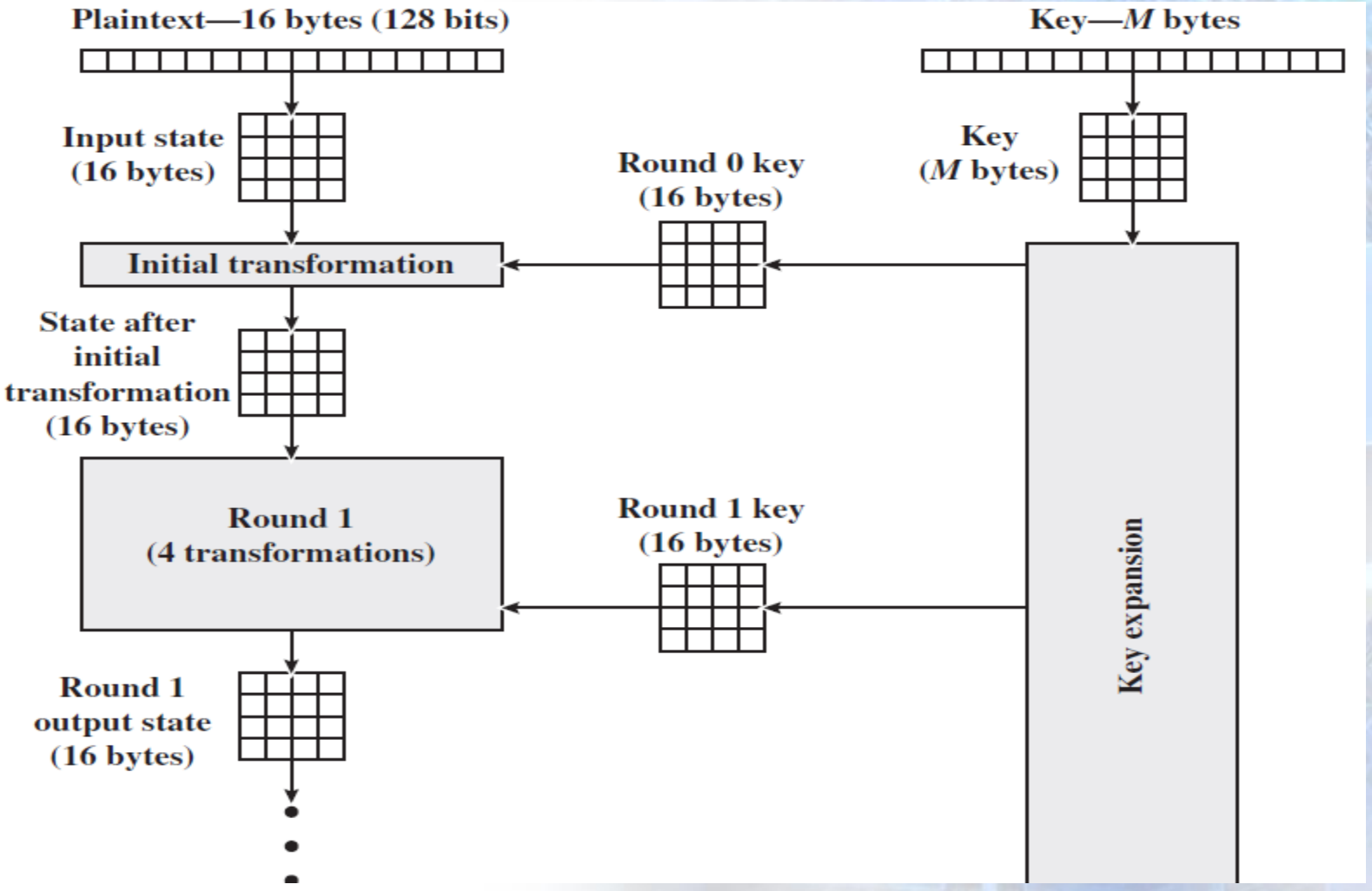
Advanced Encryption Standard

- **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

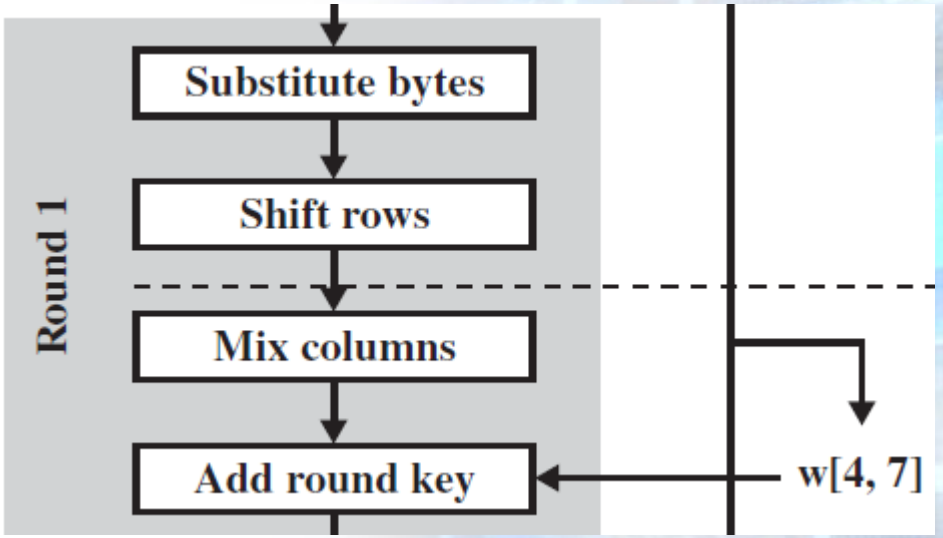
- **Mix columns:** A substitution that alters each byte in a column as a function of all of the bytes in the column.
- **Add round key:** A simple bitwise XOR of the current block with a portion of the expanded key.

Advanced Encryption Standard

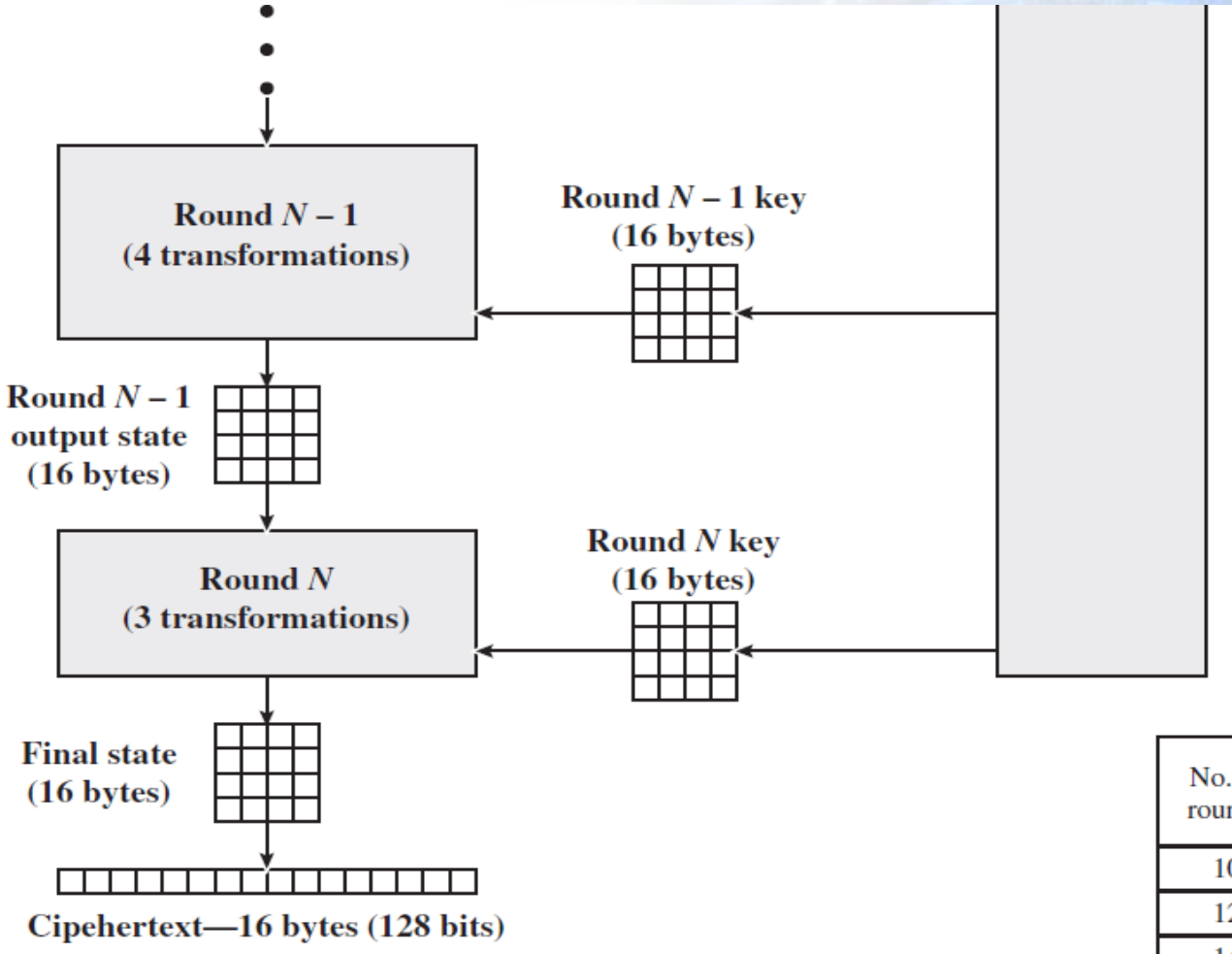


Advanced Encryption Standard

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.

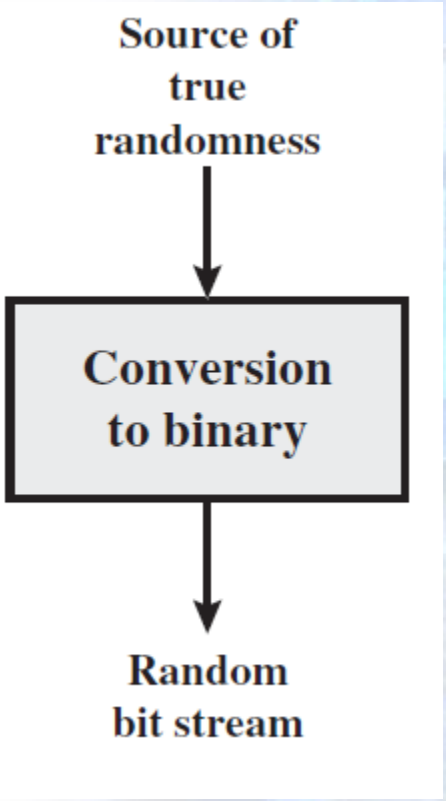
Pseudorandom Numbers

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

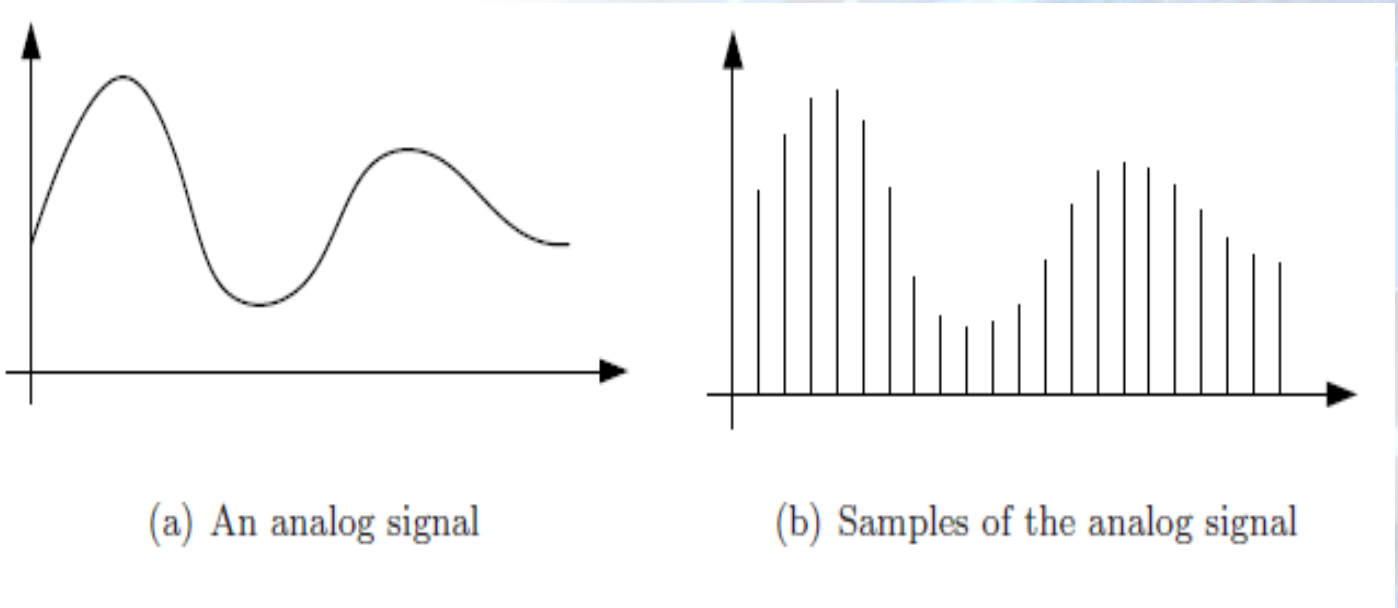
- The entropy source is drawn from physical environment of the computer and could include keystroke timing patterns, disk electrical activity, mouse movements, and instantaneous values of the system clock.

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

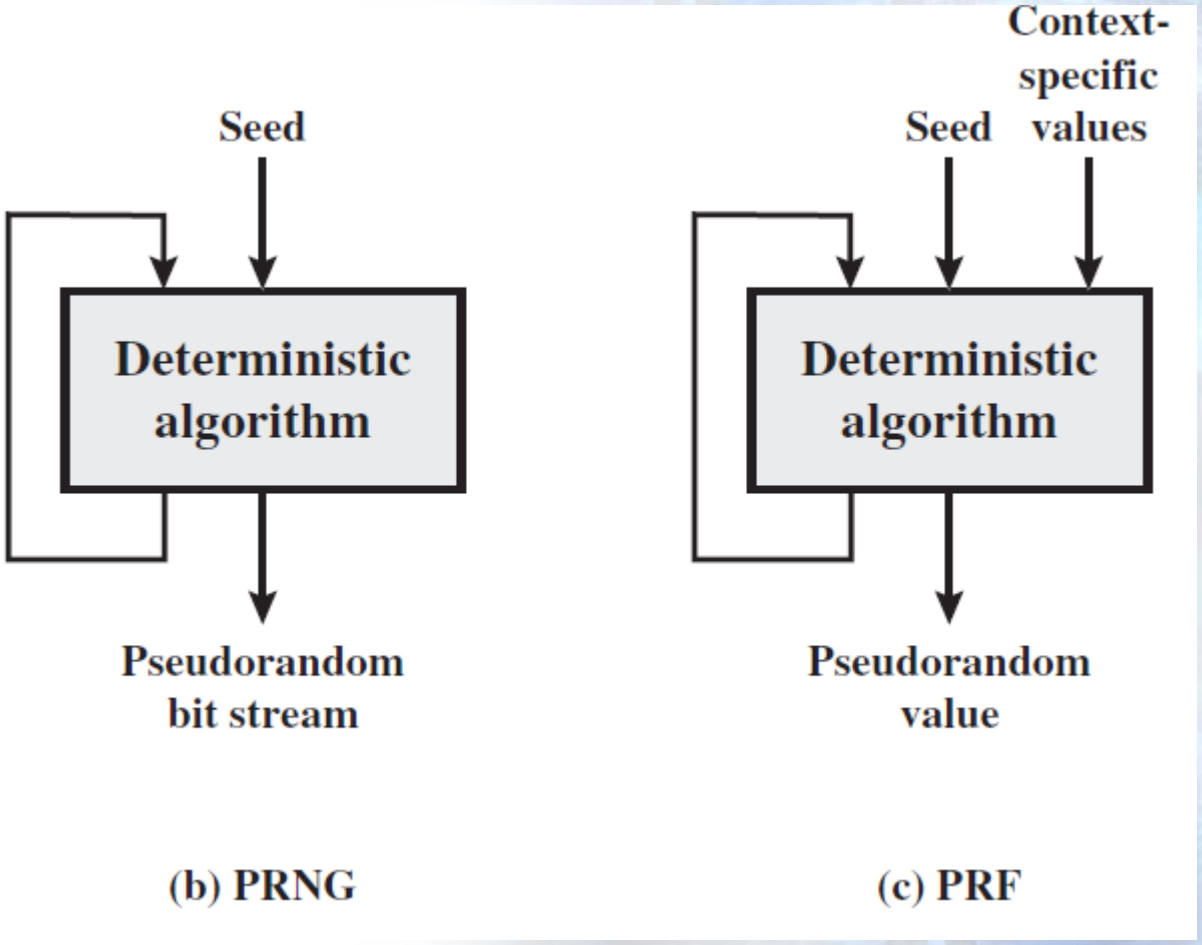
- Two different forms of PRNGs, based on application.
- **PRNG:** An algorithm used to produce an open-ended sequence of bits is referred to as a PRNG.
- App: input to a symmetric stream cipher.

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.

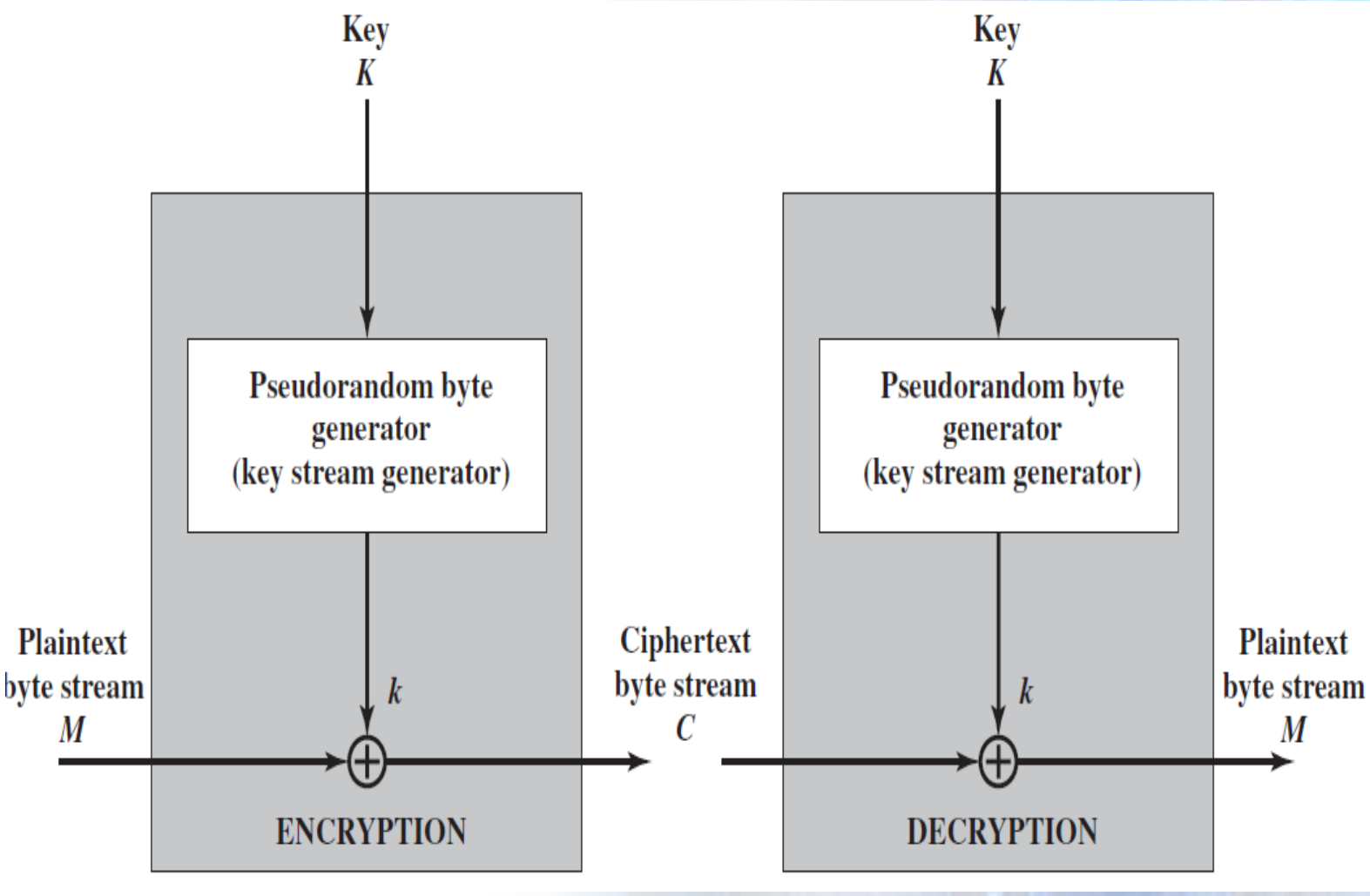
Stream Cipher Structure

- 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

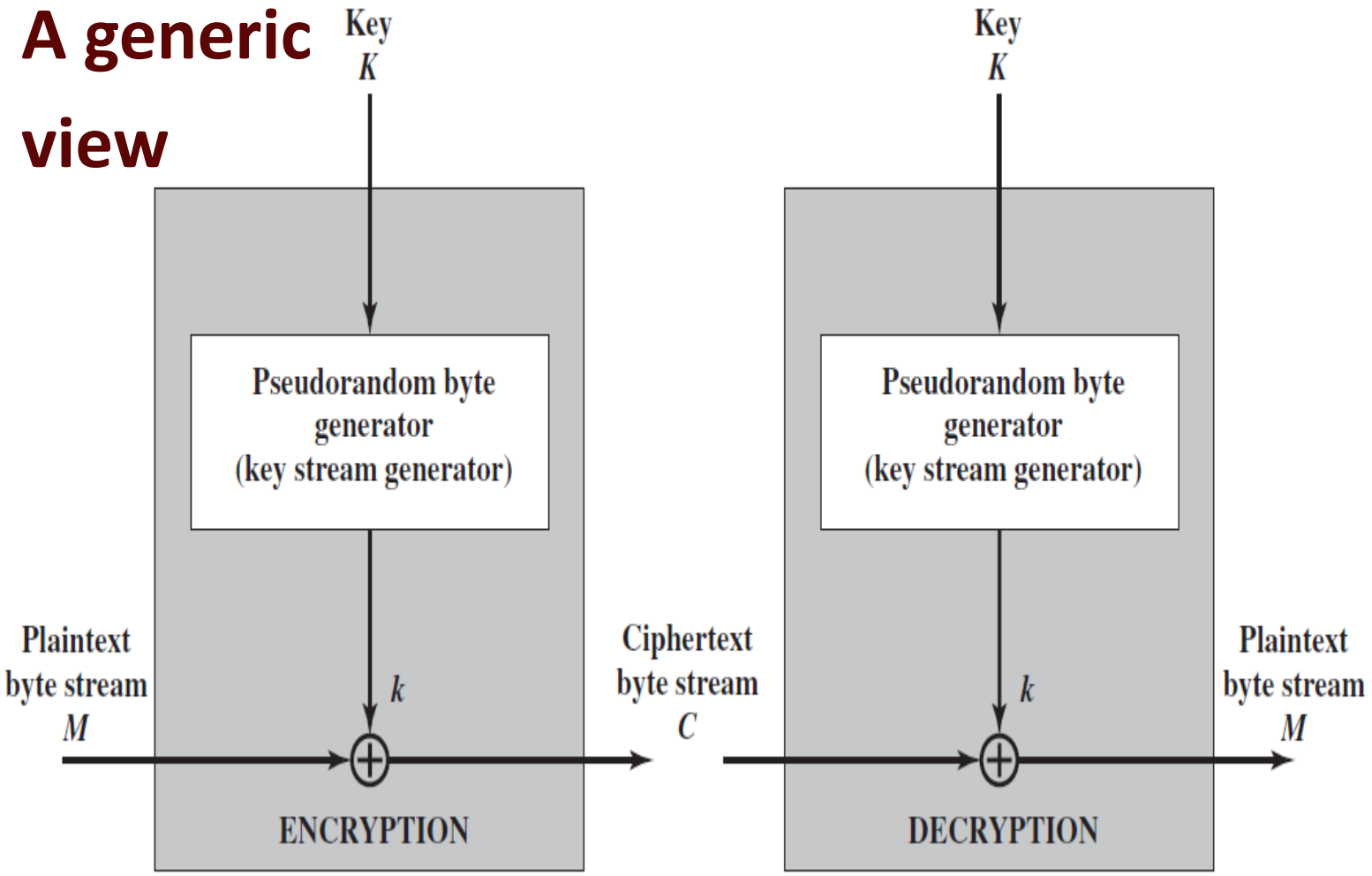
- RC4 is used in the Secure Sockets Layer/Transport Layer Security (SSL/TLS) standards that have been defined for communication between Web browsers and servers.

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

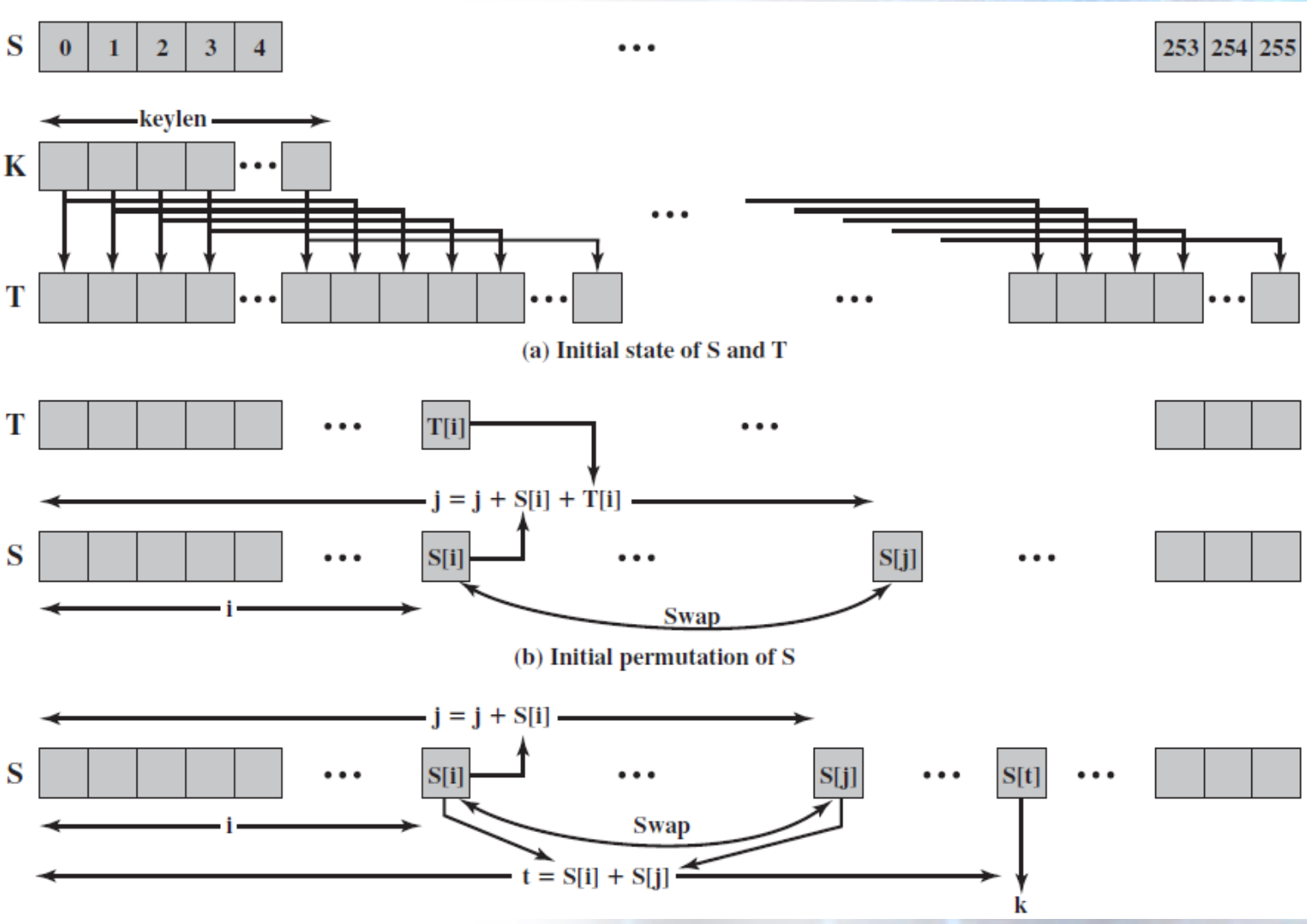
Stream Generation: Once the S vector is initialized, the input key is no longer used.

```
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

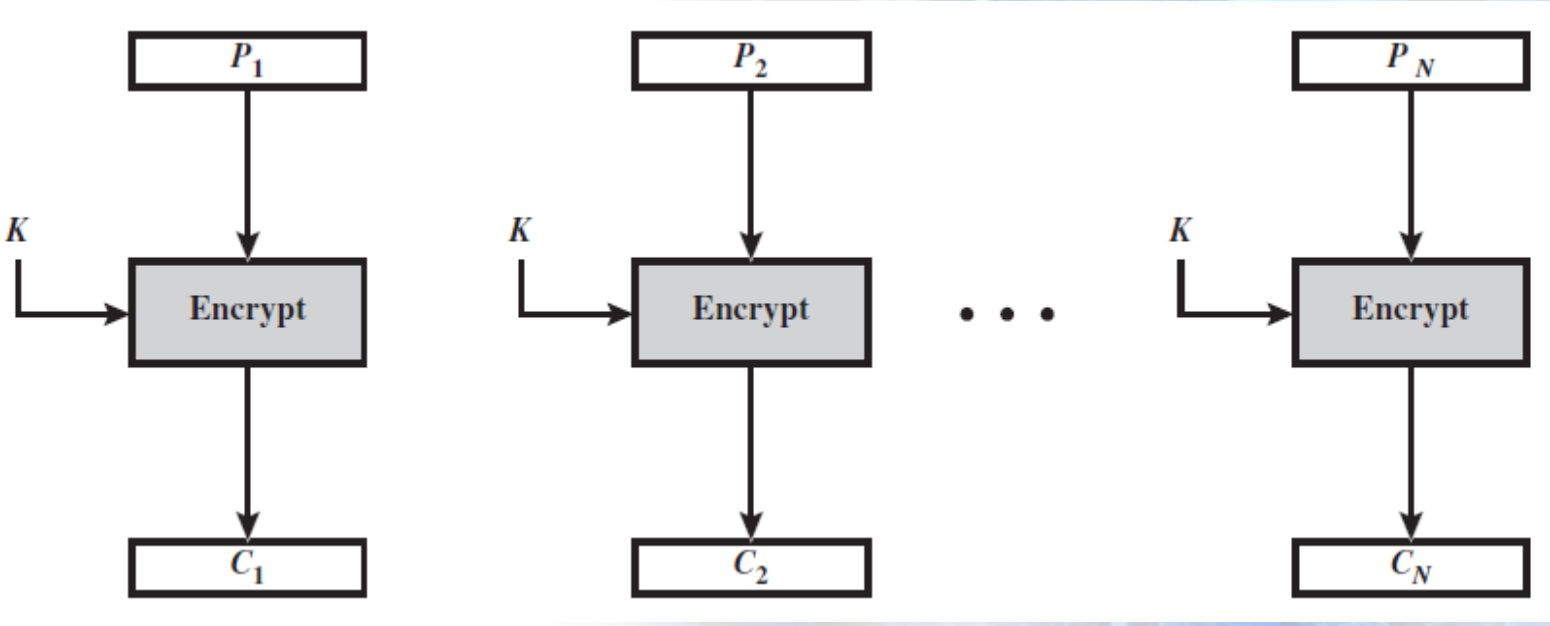
- The term codebook is used because, for a given key, there is a unique ciphertext for every **b**-bit block of plaintext.

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.

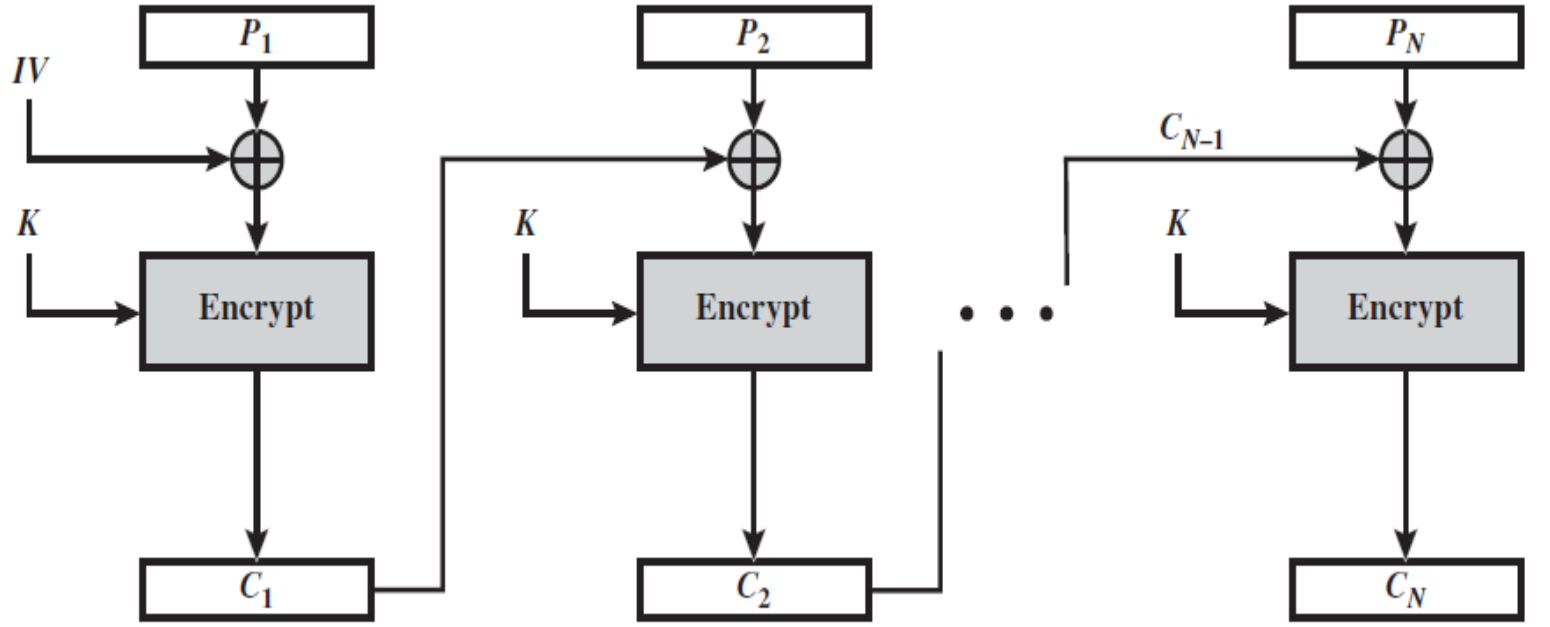
Elect. Codebook, Cipher Block Chaining

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

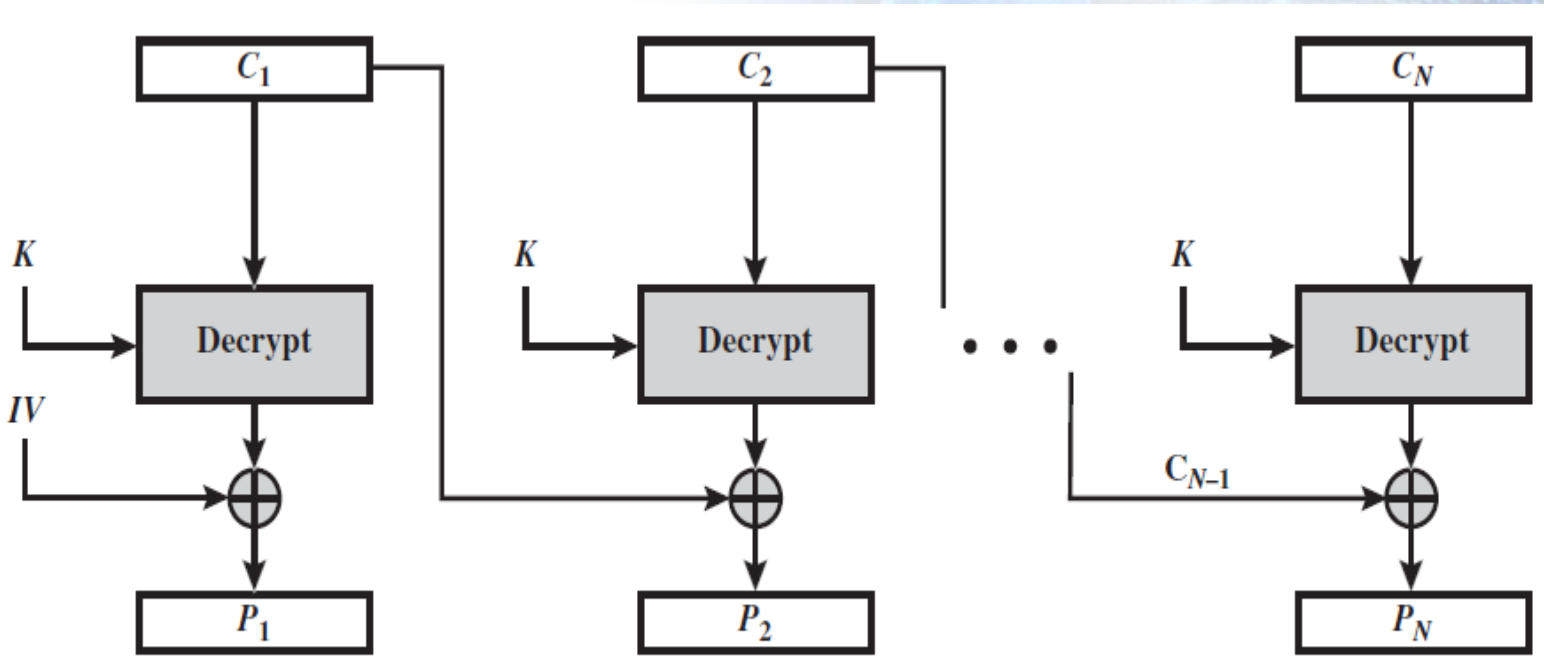
CBC Decryption:

- 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).

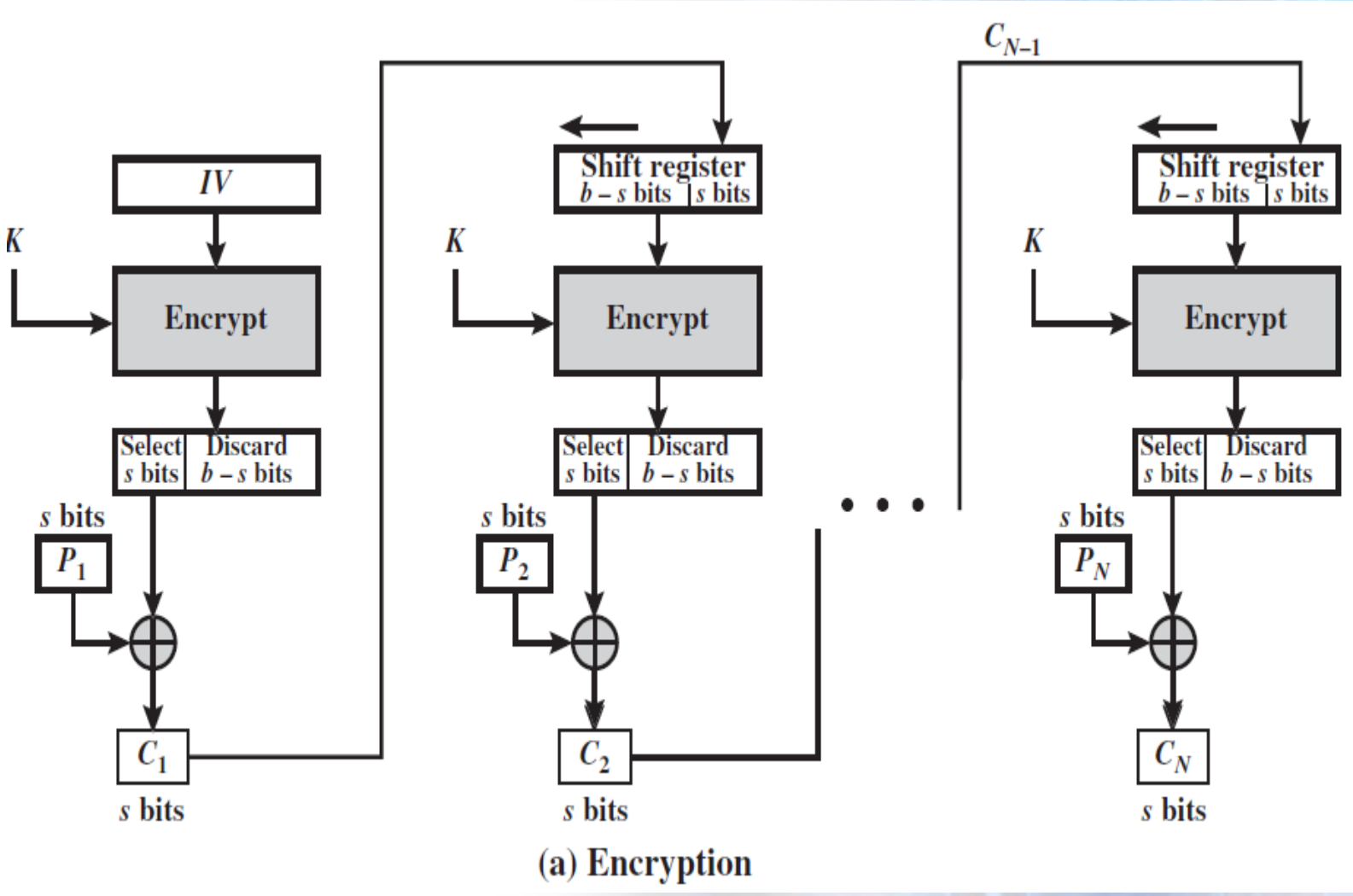
Cipher Feedback Mode

- 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

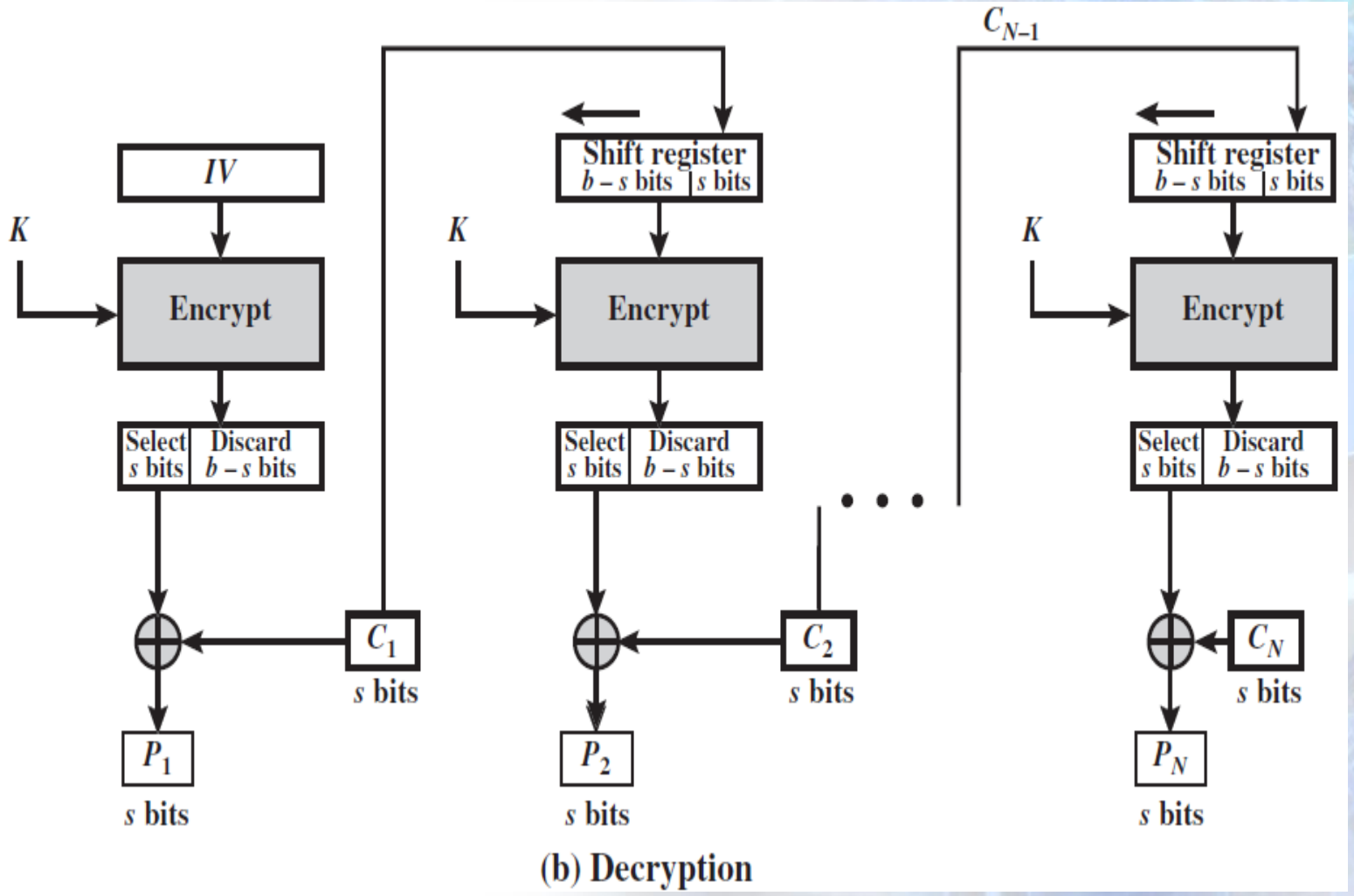
Decryption:

- 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.

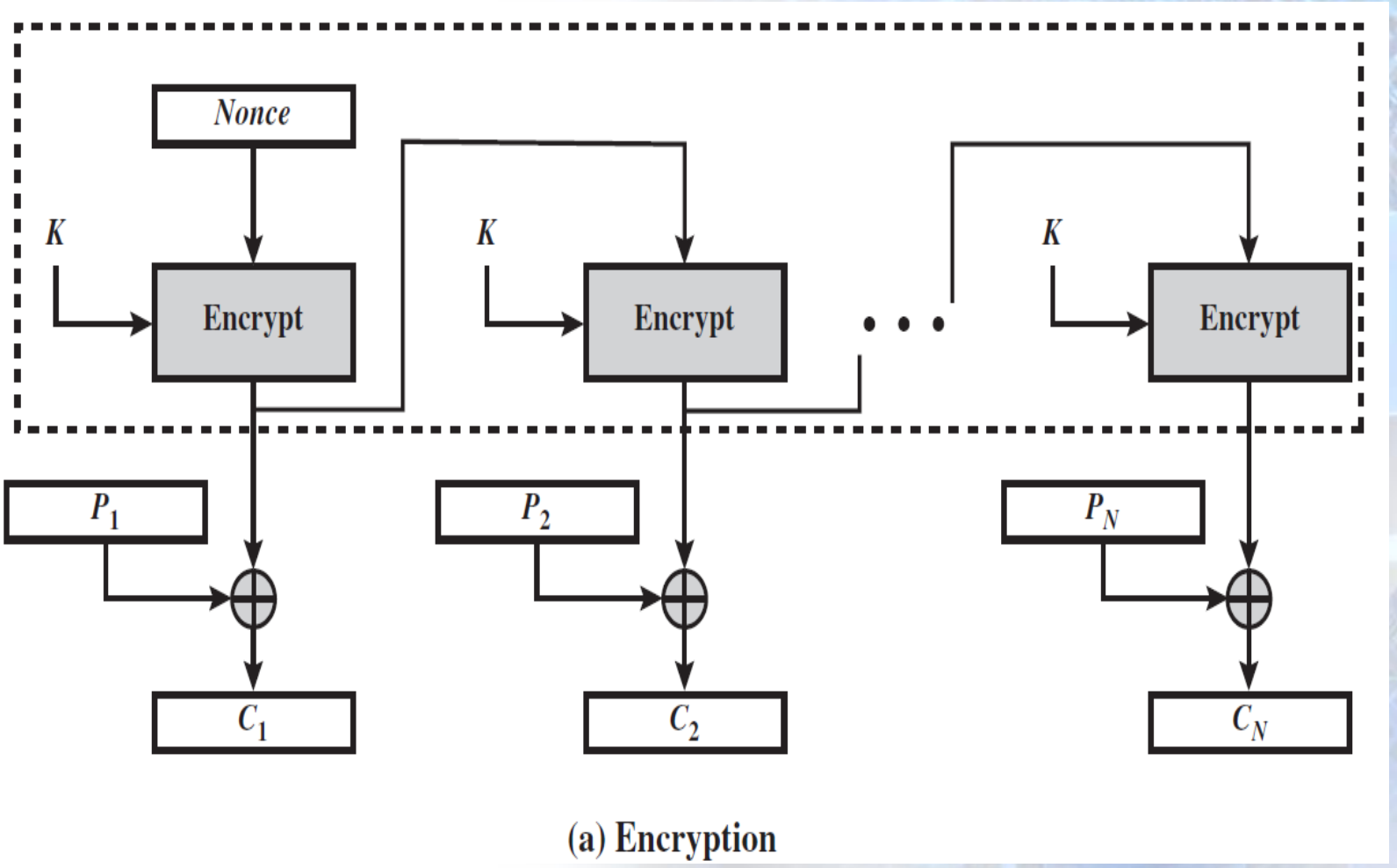
Counter Mode, Output Feedback Mode

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

Counter Mode, Output Feedback Mode

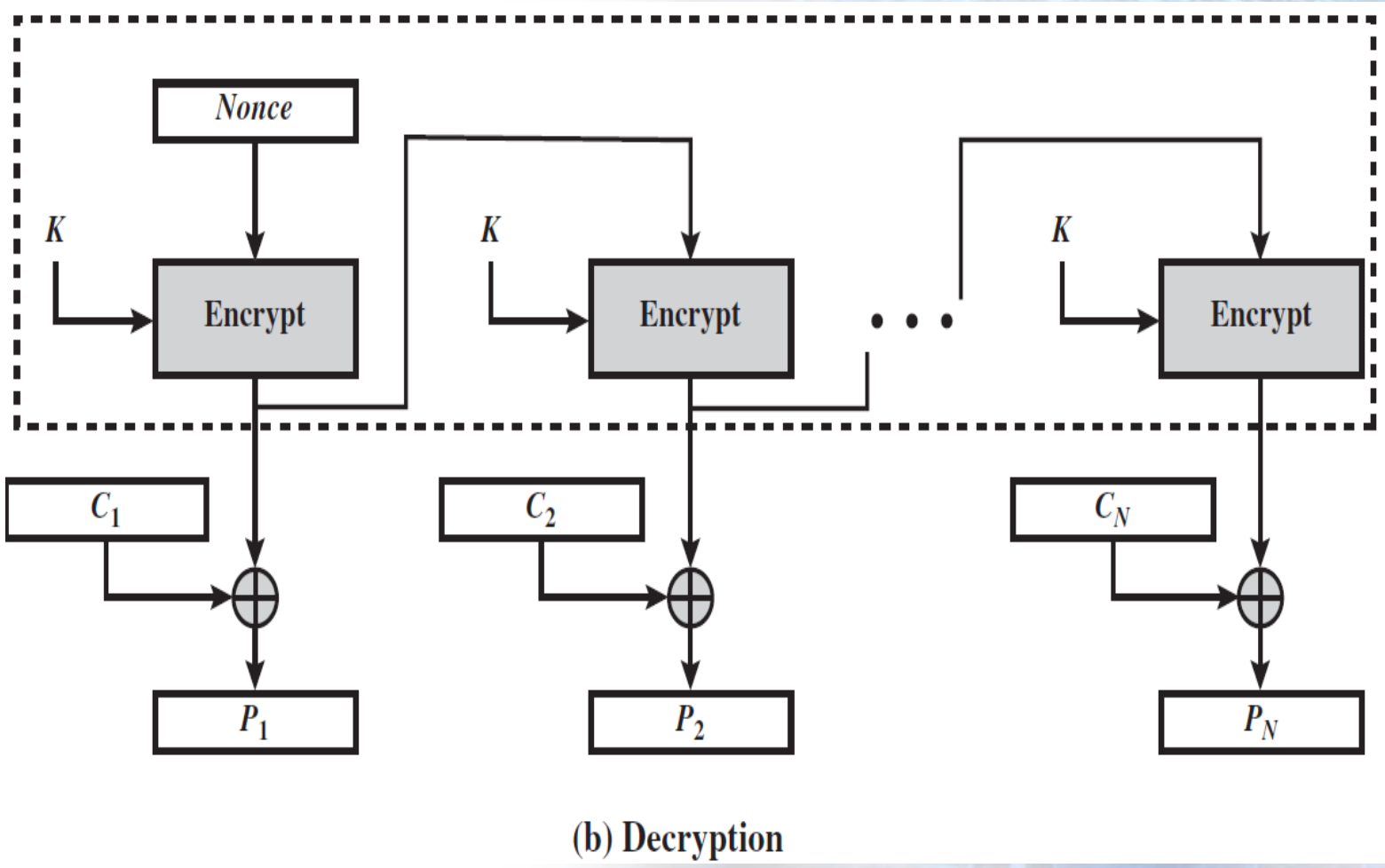
- The OFB mode requires a nonce; an initialization vector, unique to each execution of the encryption operation.
- Sequence of encryption output blocks depends only on the key and the IV and not on plaintext.

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.

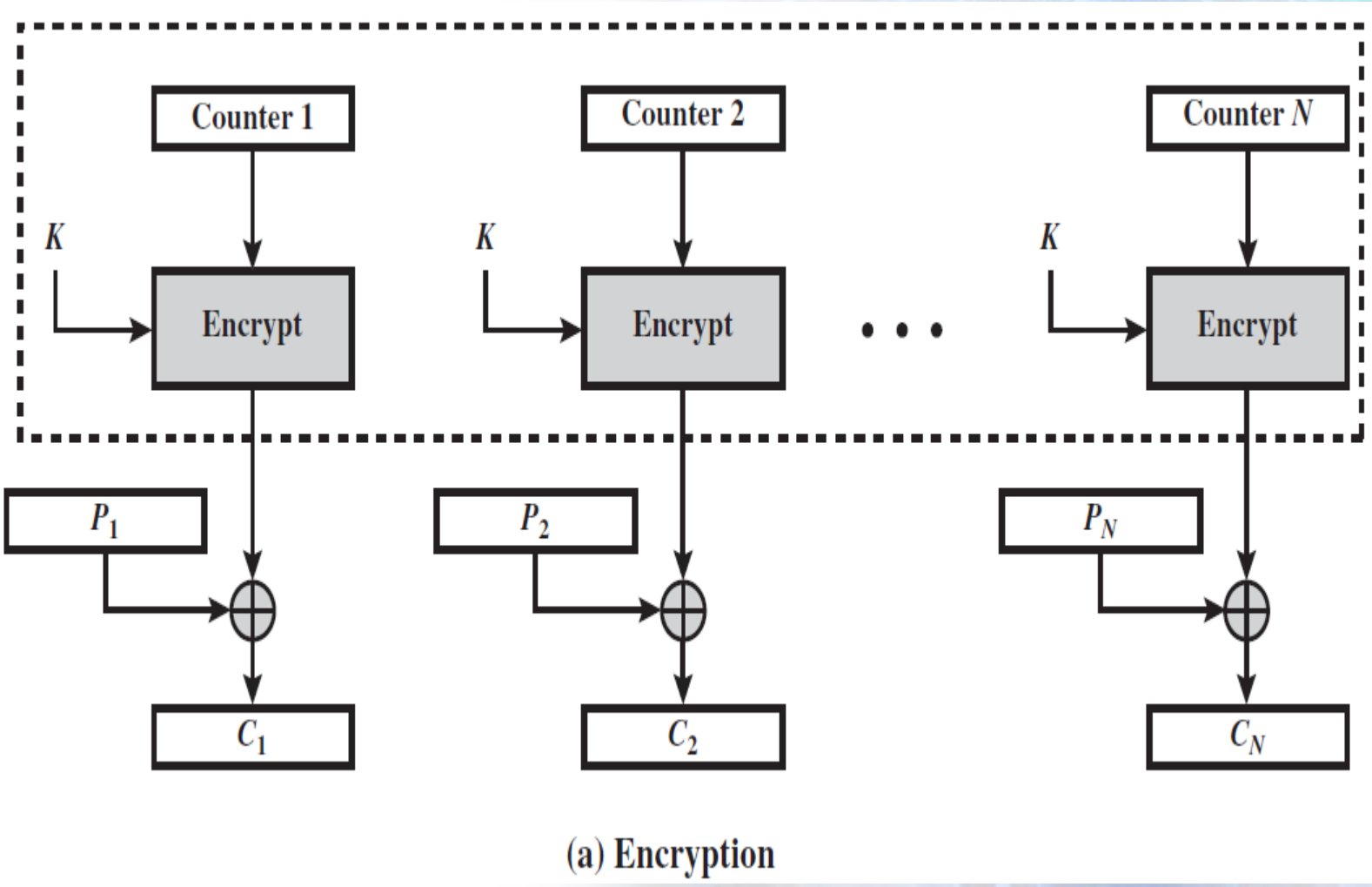
Counter Mode, Output Feedback Mode

- 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

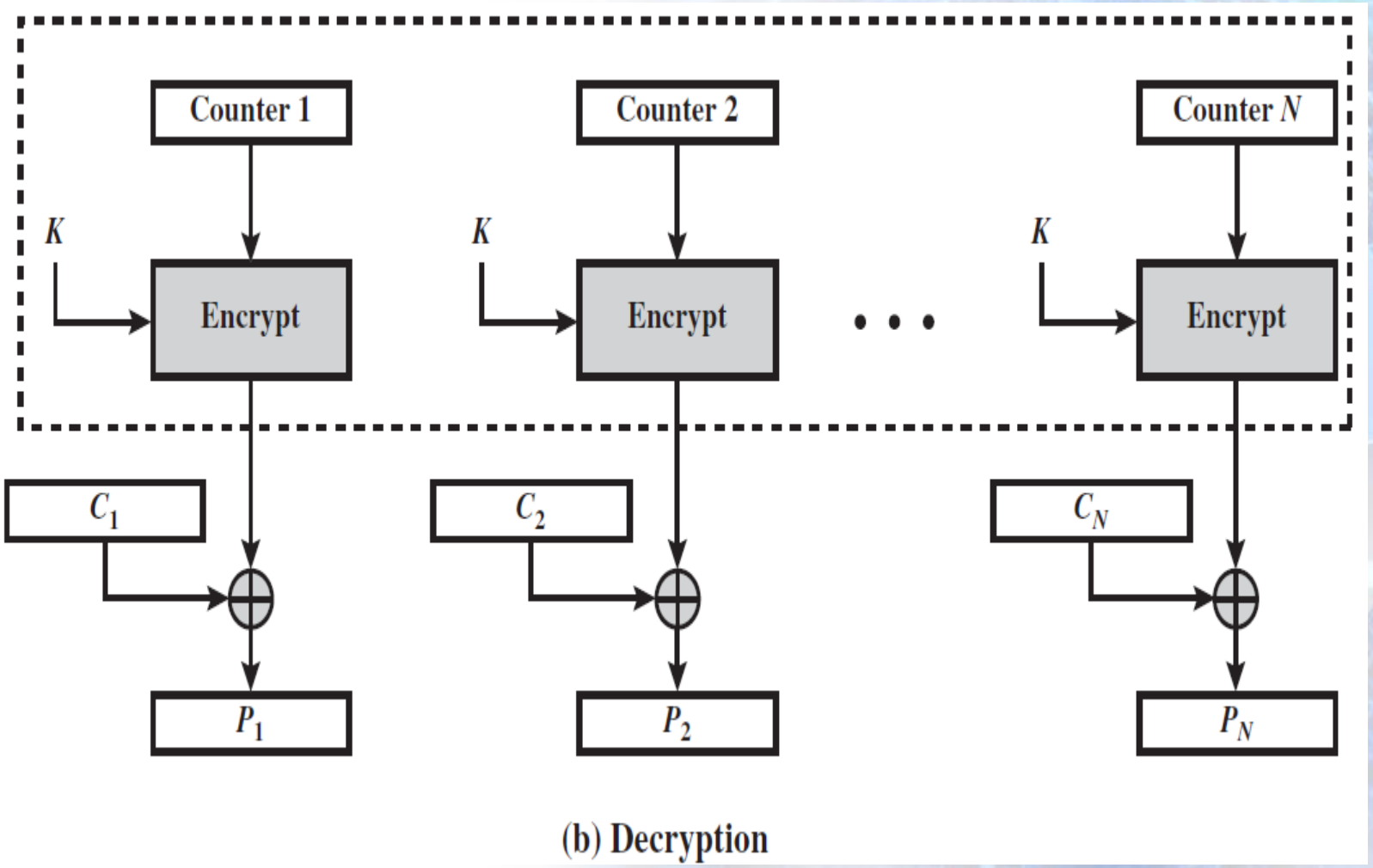


(a) Encryption

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

End

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

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: $MAC_M = F(K_{AB}, M)$.

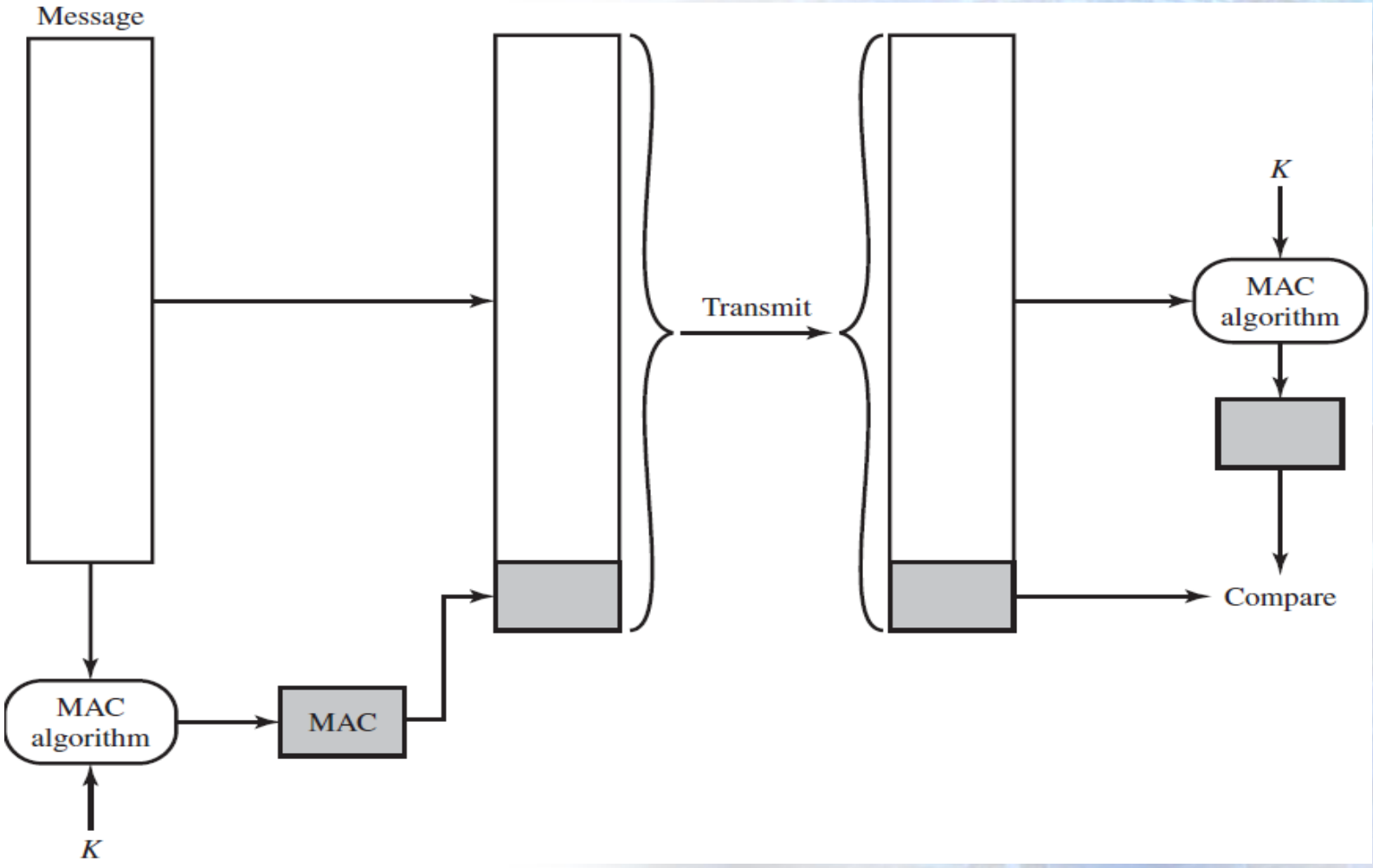
Message Authentication

- 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.

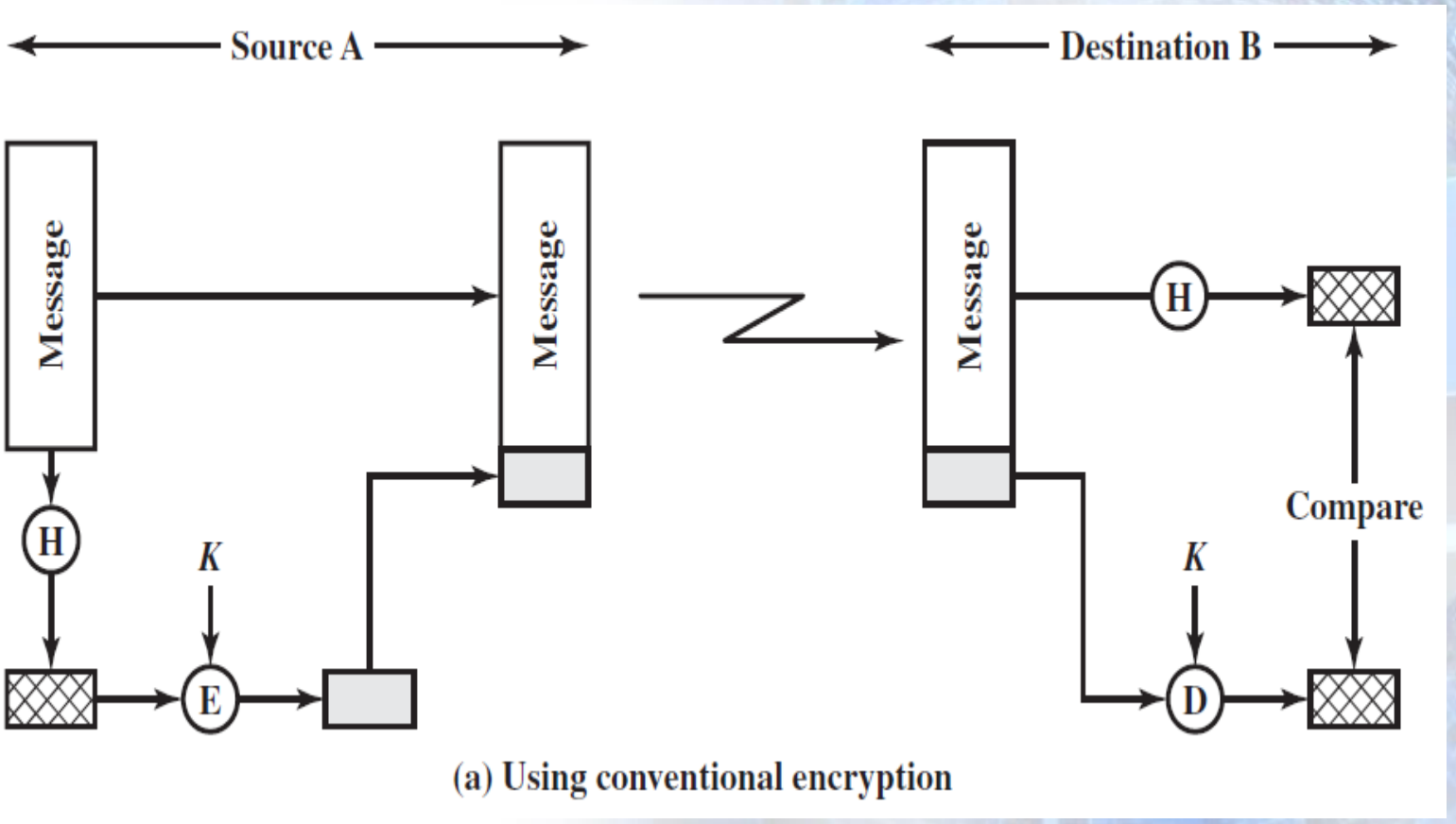
One-way Hash Function

- 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

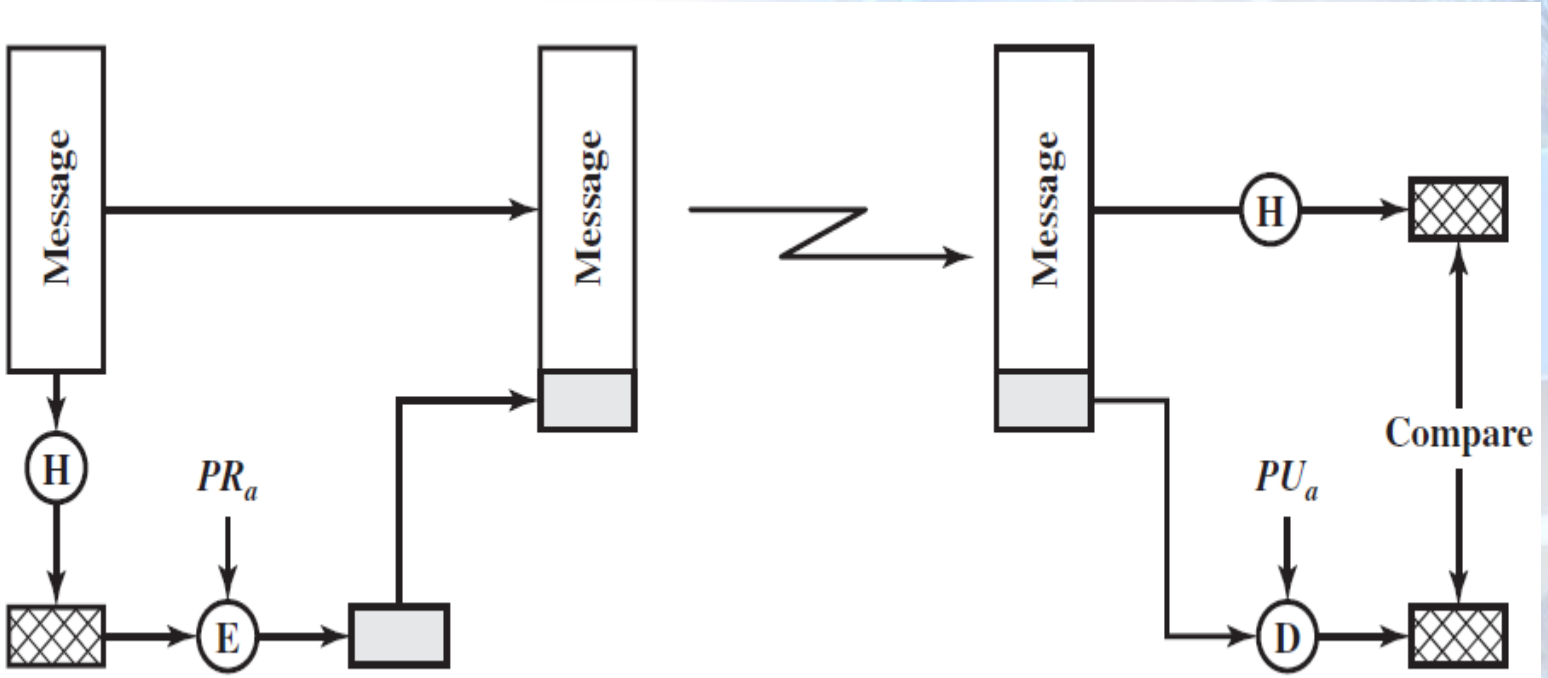
One-way Hash Function

- **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.

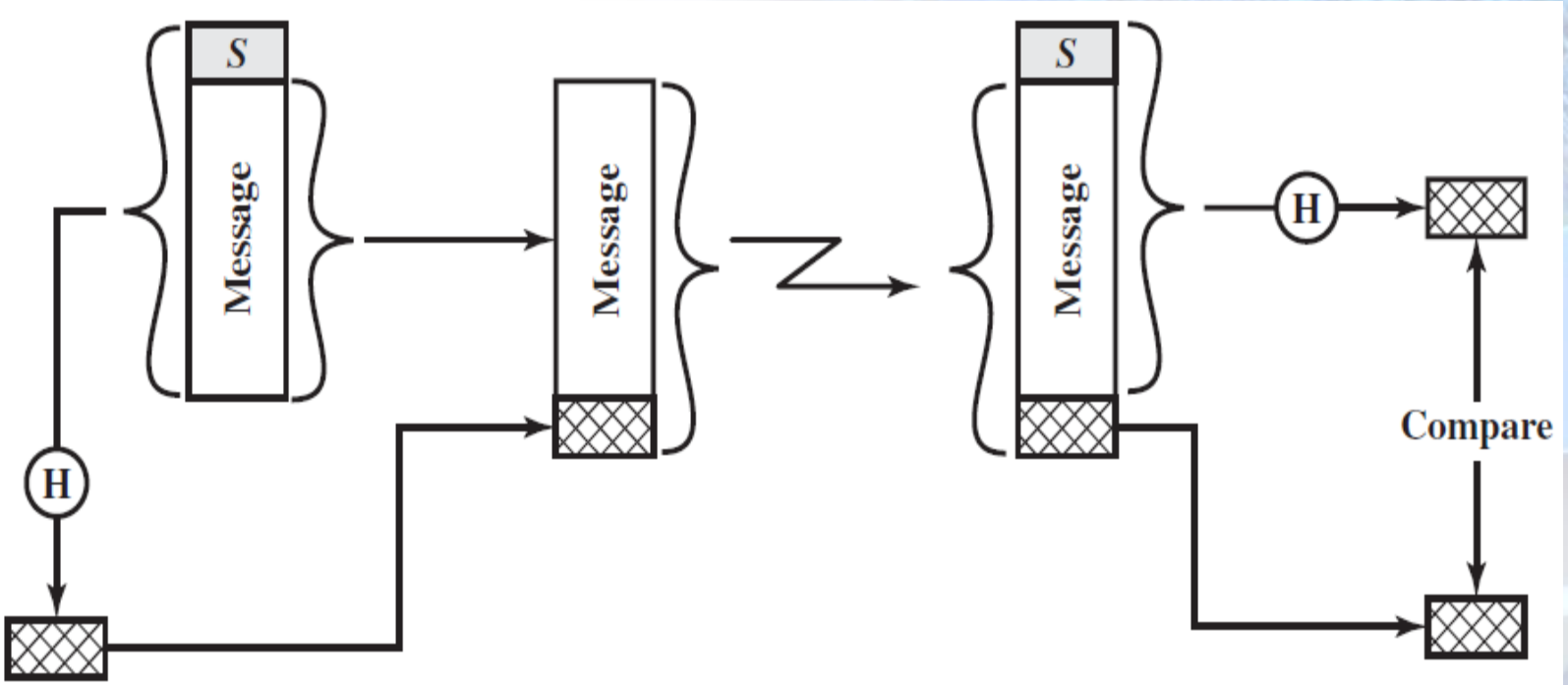
One-way Hash Function

- 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

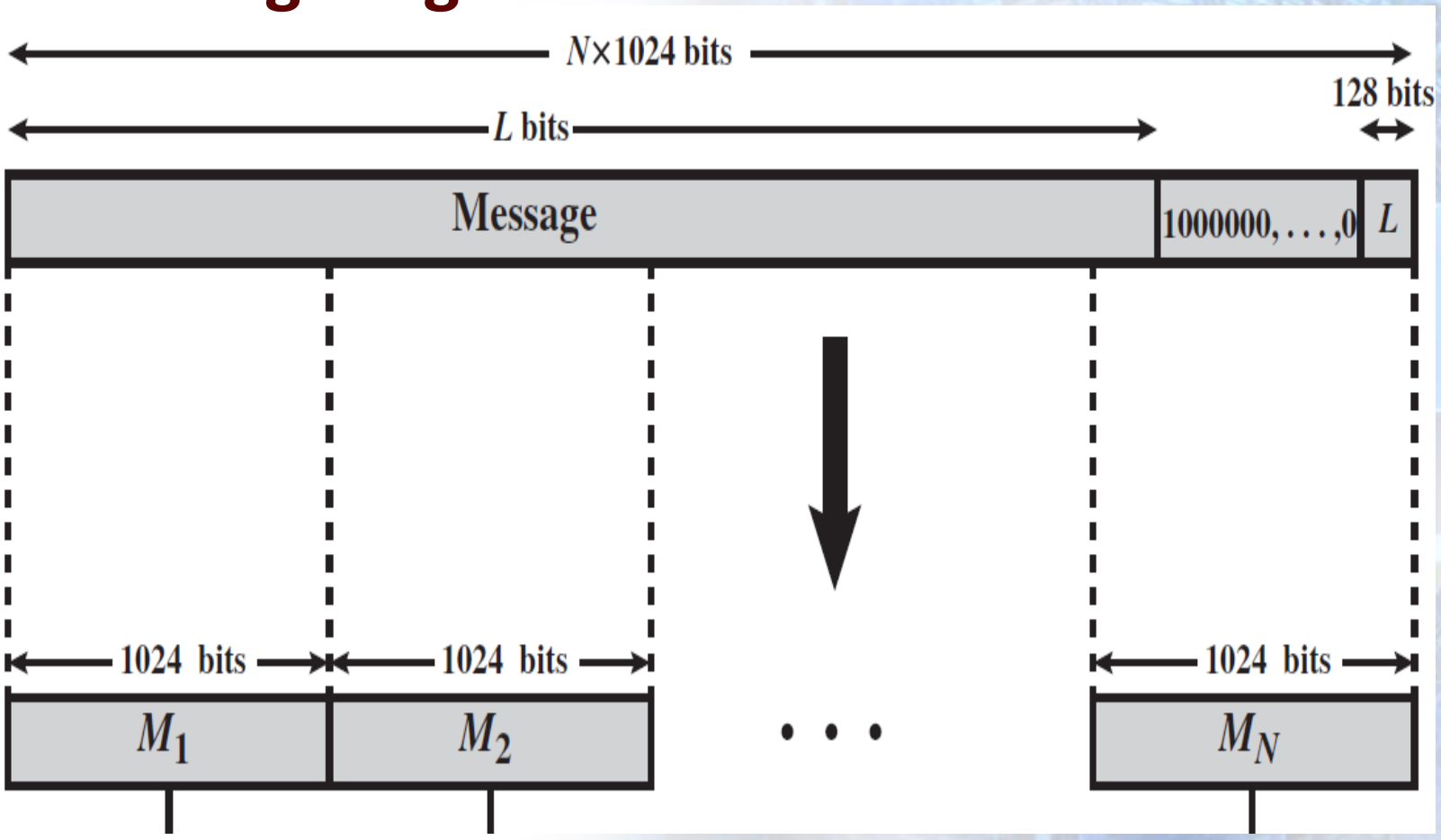
- A block of 128 bits is appended to the message.
- This block is treated as an unsigned 128-bit integer and contains the length of the original message (before the padding).

The Secure Hash Function (SHA)

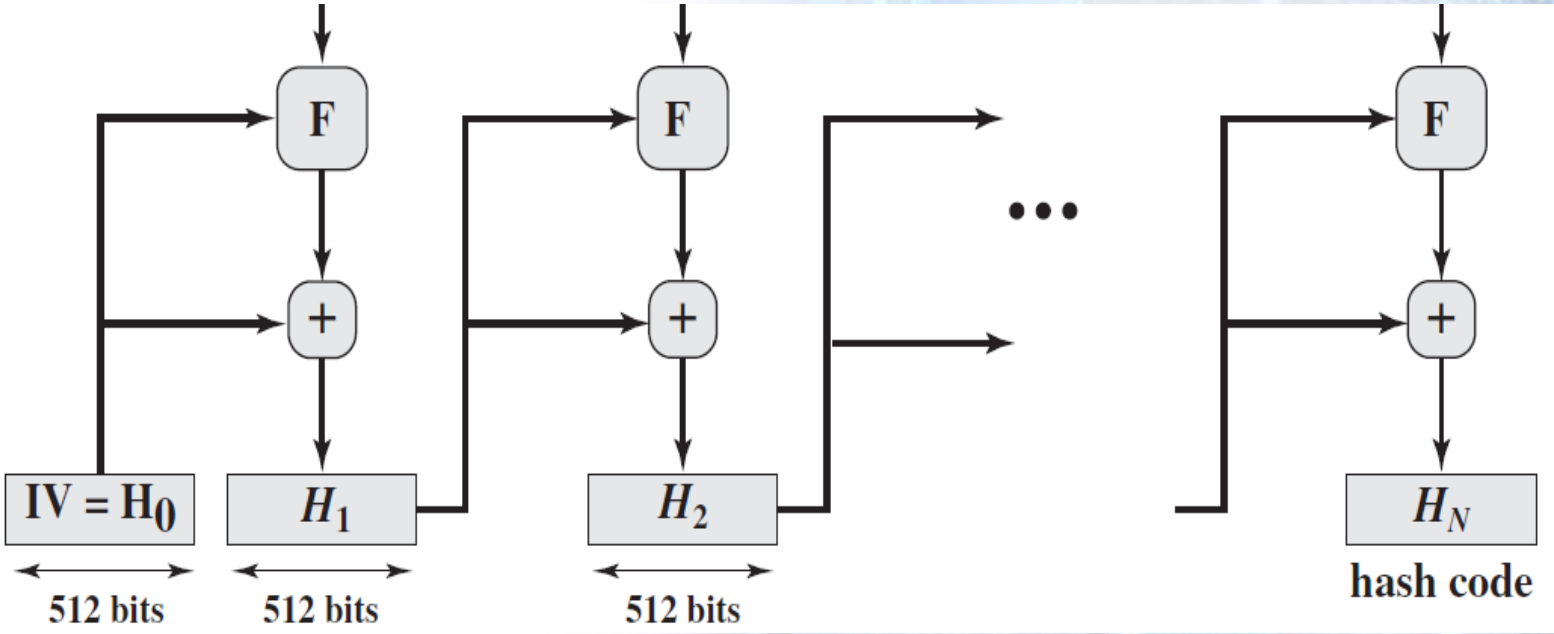
- Outcome of first two steps yields a message an integer multiple of 1024 bits in length.
- Total length of the expanded message is N × 1024 bits as the expanded message is a sequence of 1024-bit blocks M_1, M_2, . . ., $M_{N.}$

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

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

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)

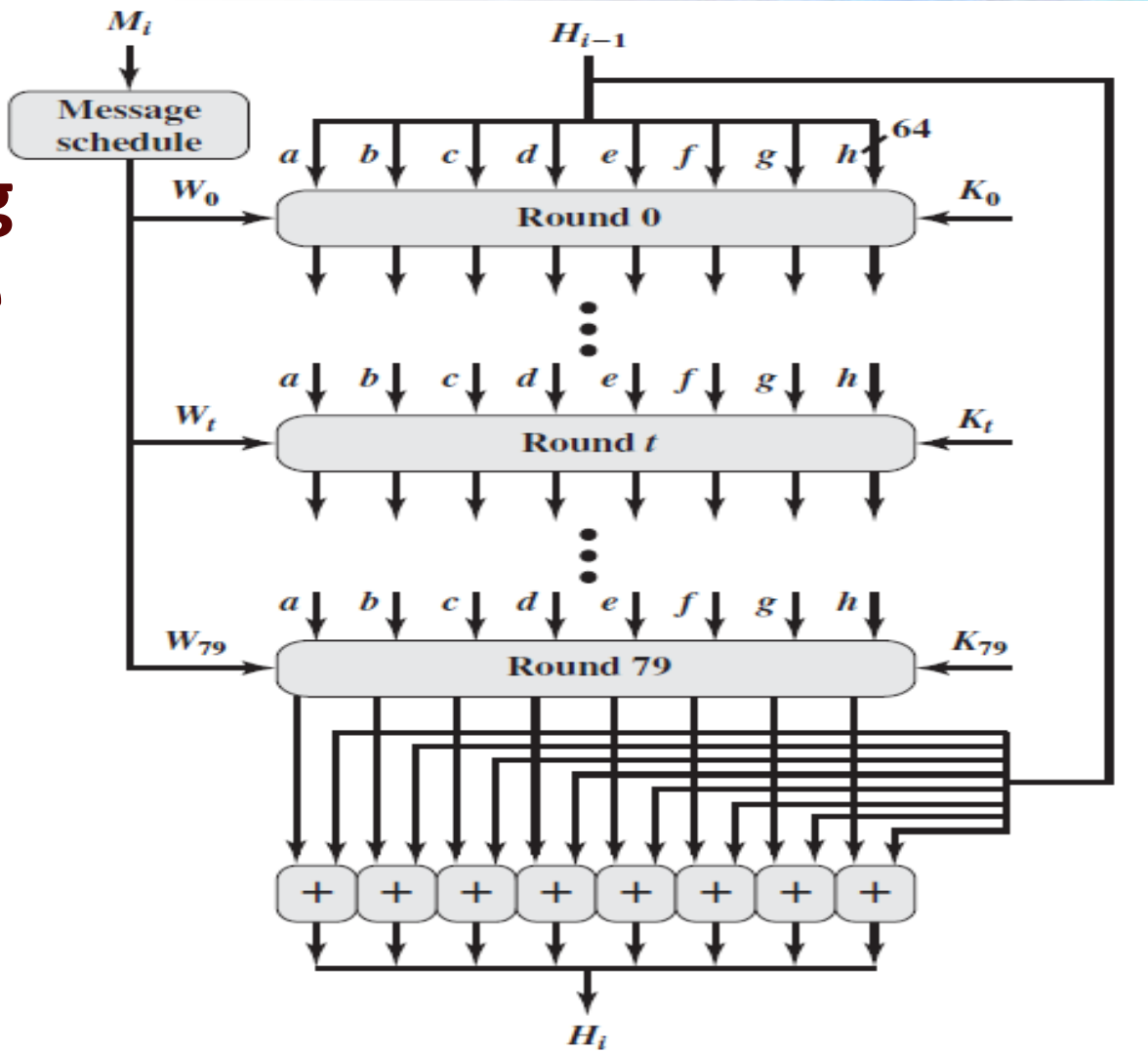
- Each round also makes use of an additive constant K_t, where t = 0 … … 79.
- The constants eliminate any regularities in the input data.

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



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

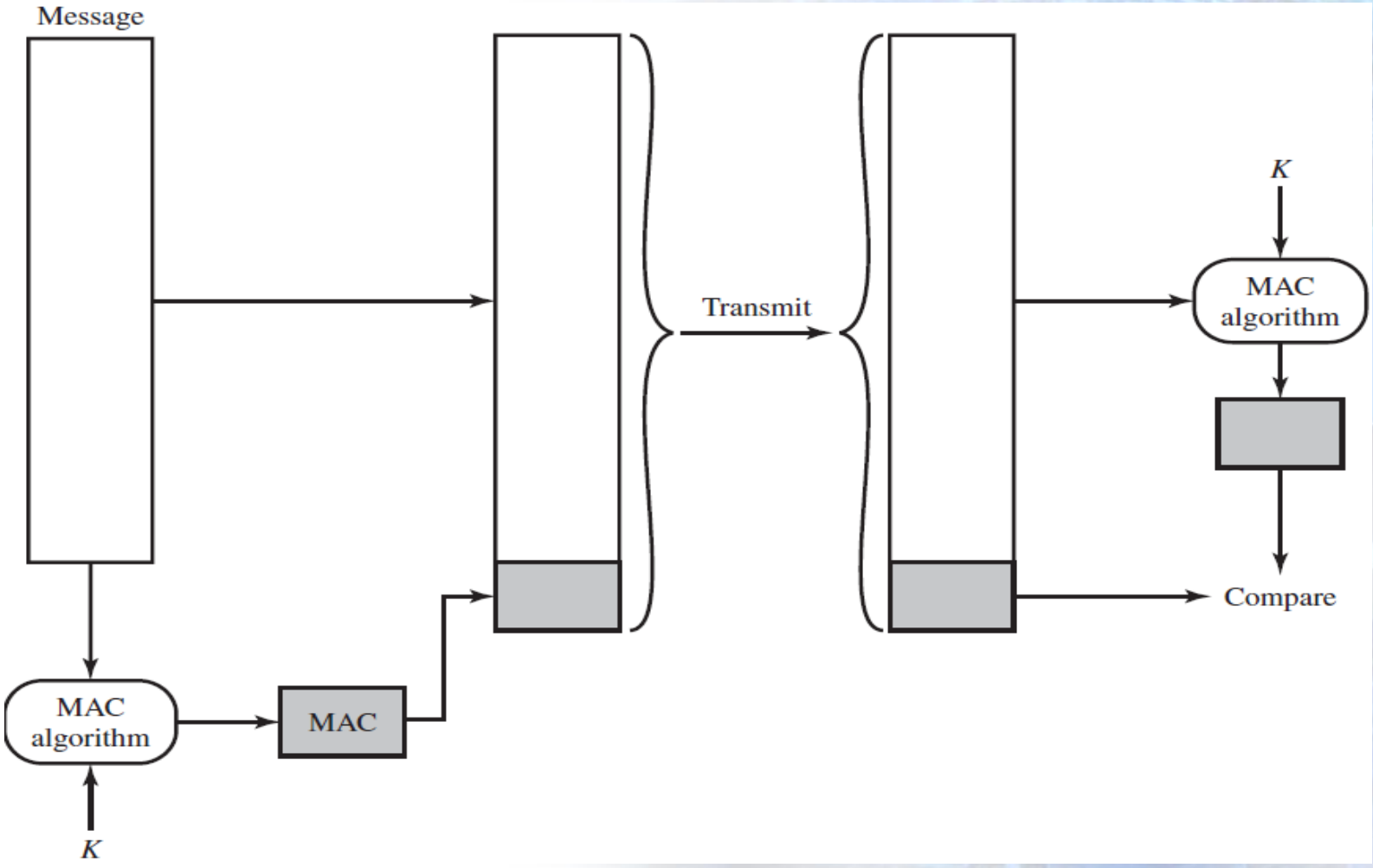
- Message authentication is a procedure that allows communicating parties to verify that received messages, file, document, or other collection of data are authentic.

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.

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

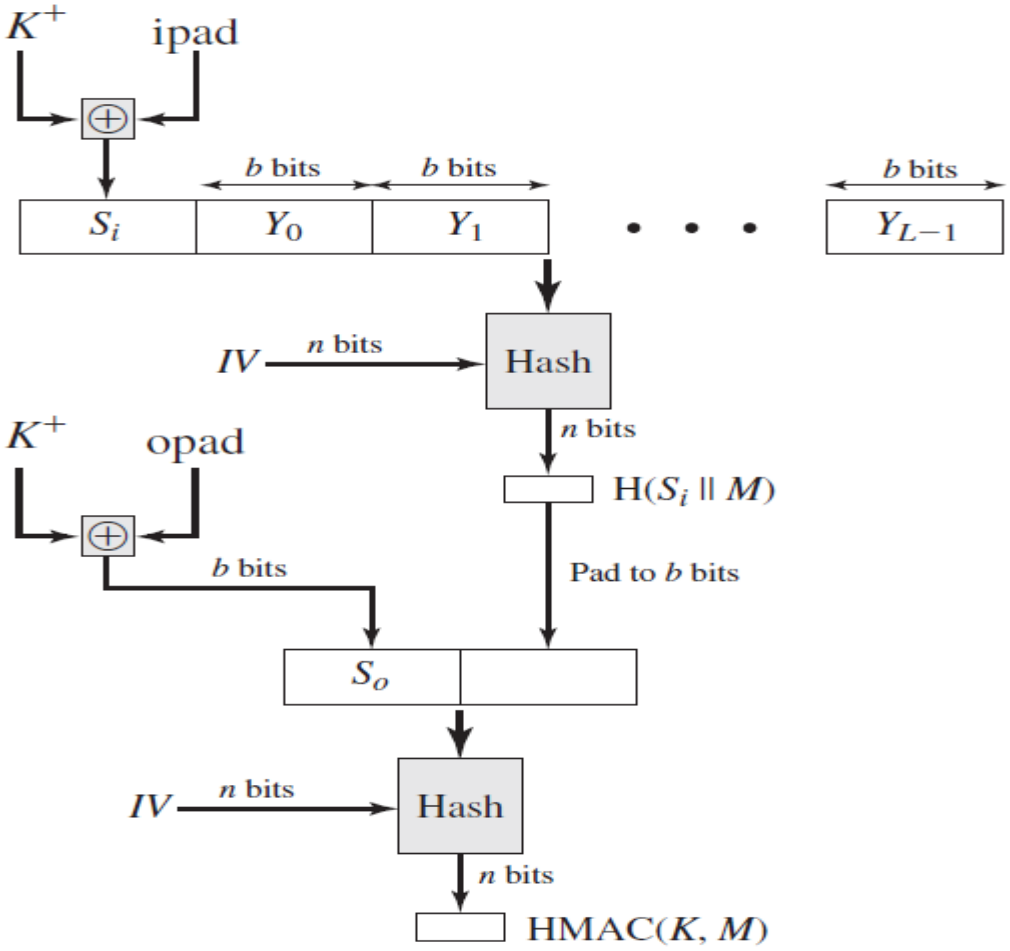
- 3. Append M to S_i.
- 4. Apply H to the stream generated in step 3.
- 5. XOR K^+ with opad to produce the b-bit block S_o.
- 6. Append the hash result from step 4 to S_o.

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

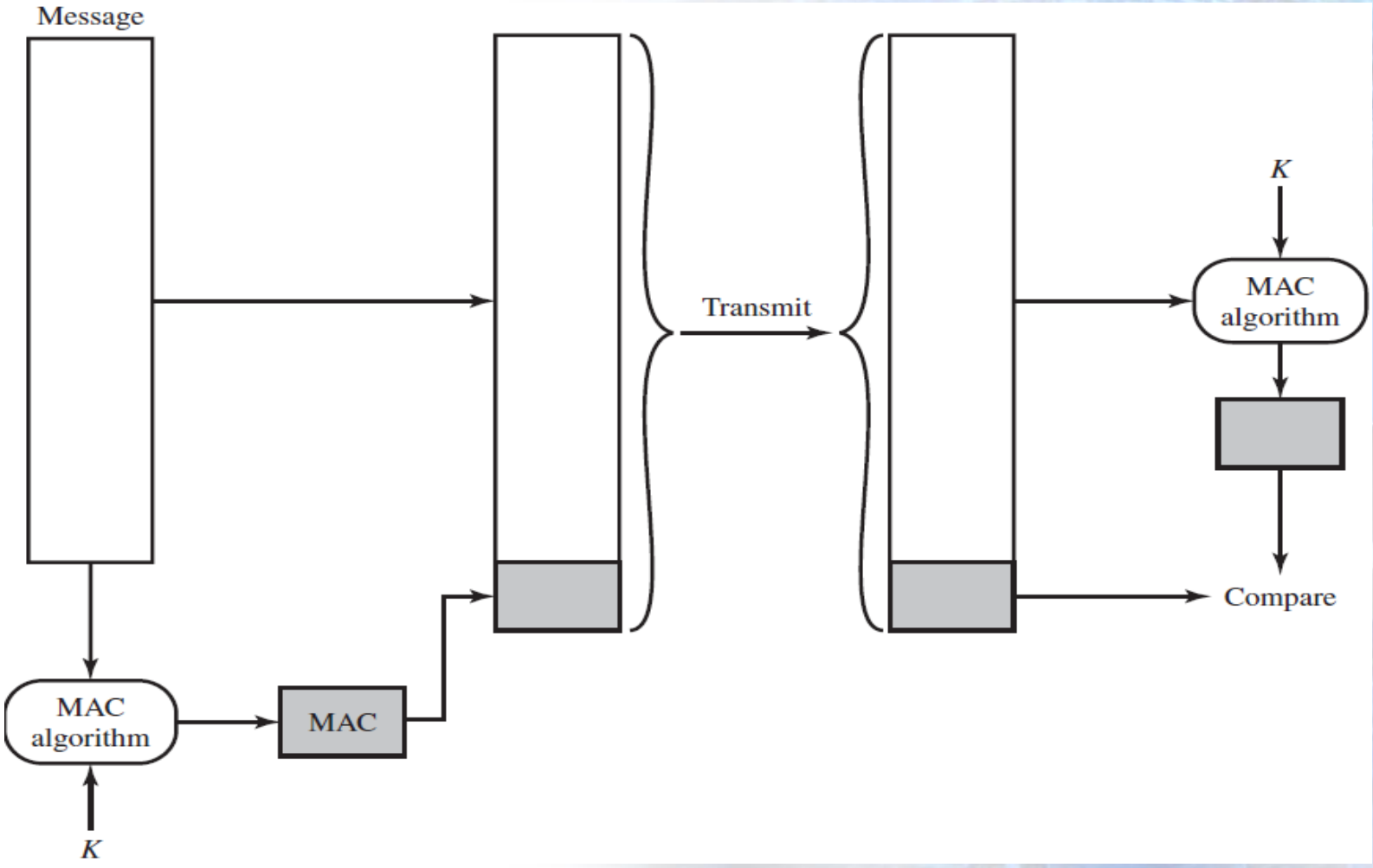
- *"Network Security Essentials : Applications and Standards"*, 2014, by William Stallings.

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.

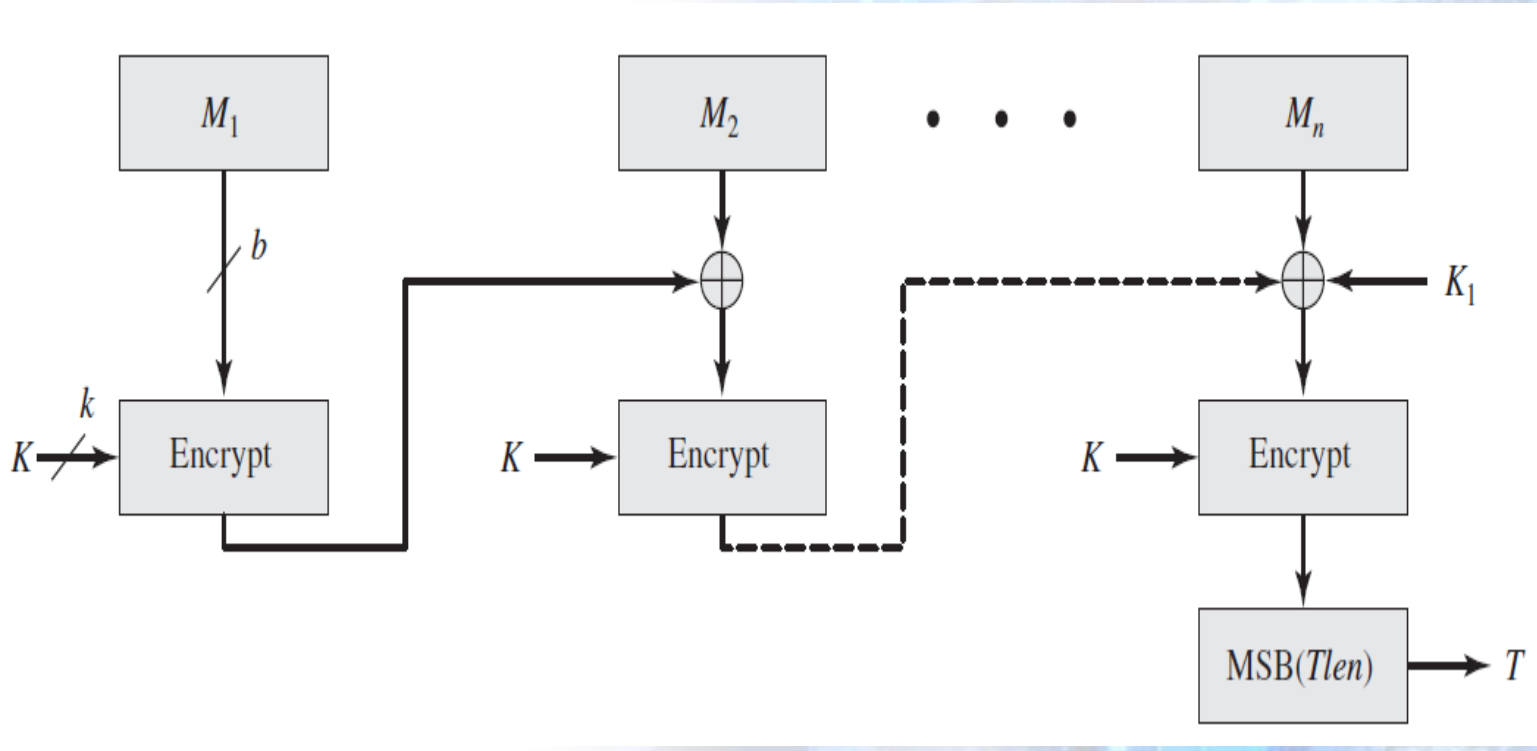
Cipher-Based MAC

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

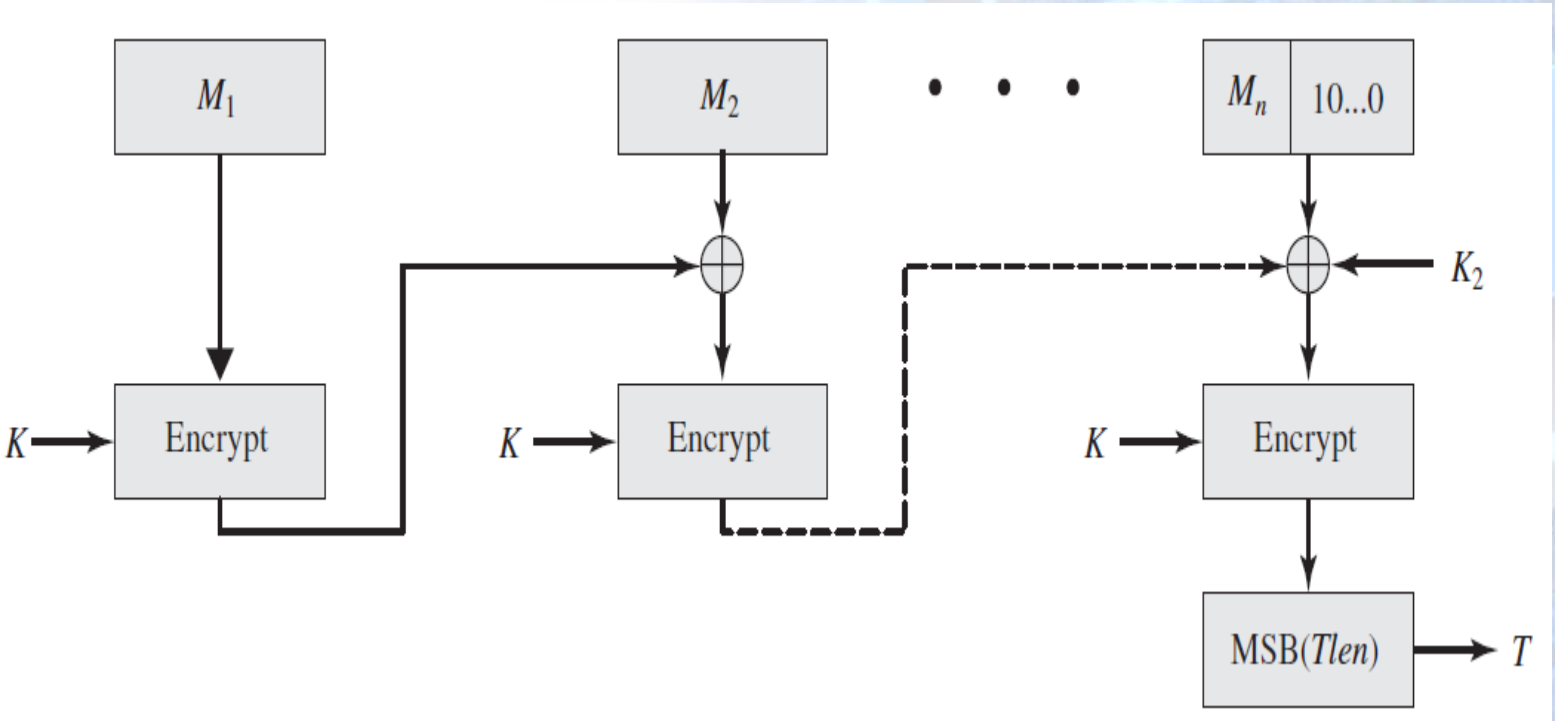
- The first subkey is derived from the resulting ciphertext by a left shift of one bit and, conditionally, by XORing a constant that depends on the block size.

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.

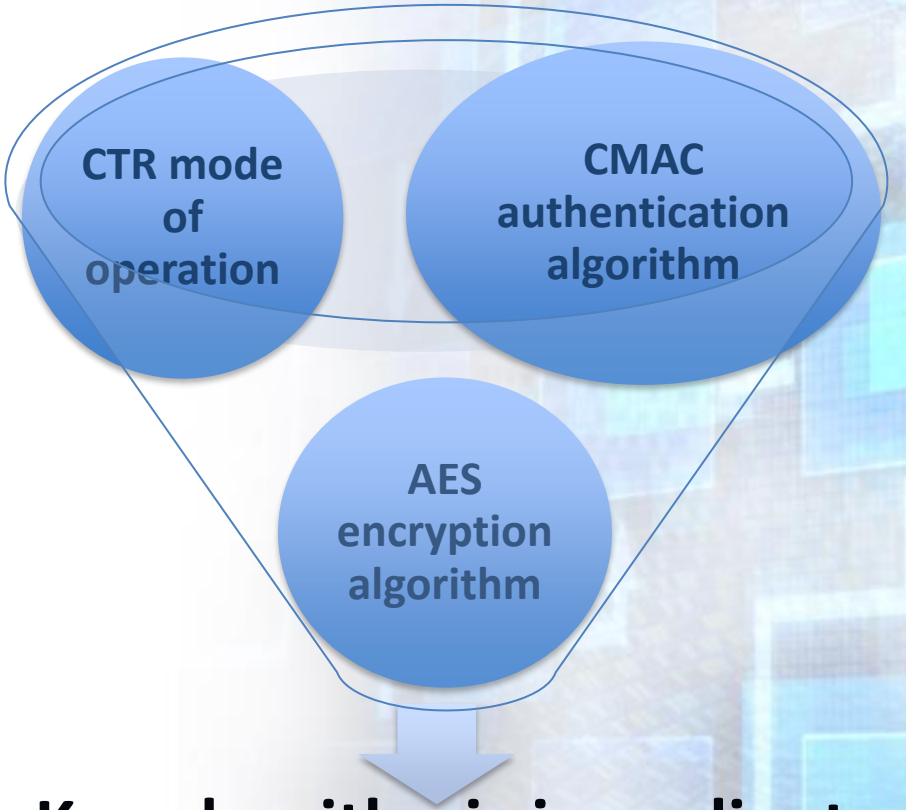
Counter With Cipher Block Chaining-MAC

- “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

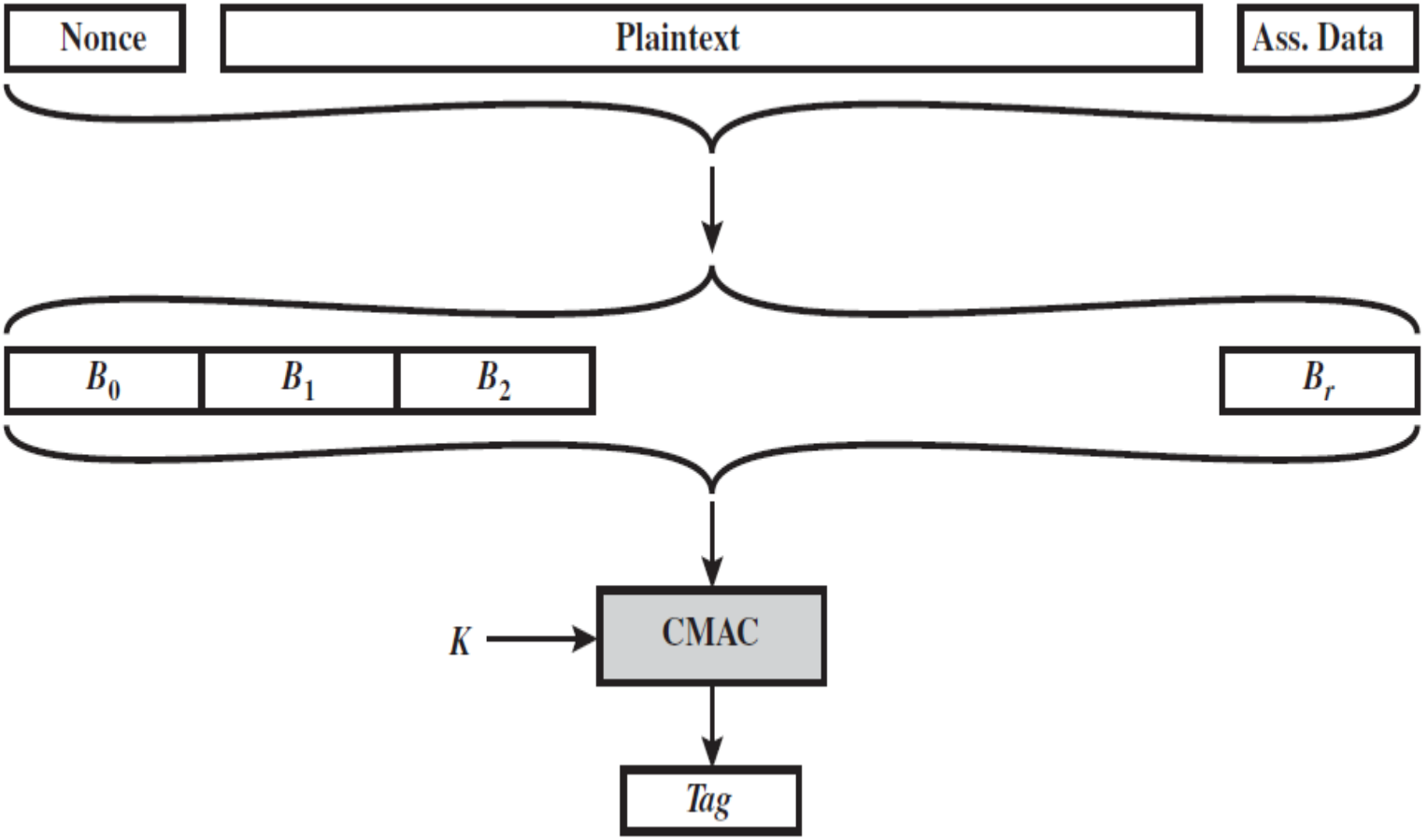
- The first block contains the nonce plus some formatting bits that indicate the lengths of the N, A, and P elements.
- This is followed by zero or more blocks that contain *A*, followed by zero of more blocks that contain *P.*

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

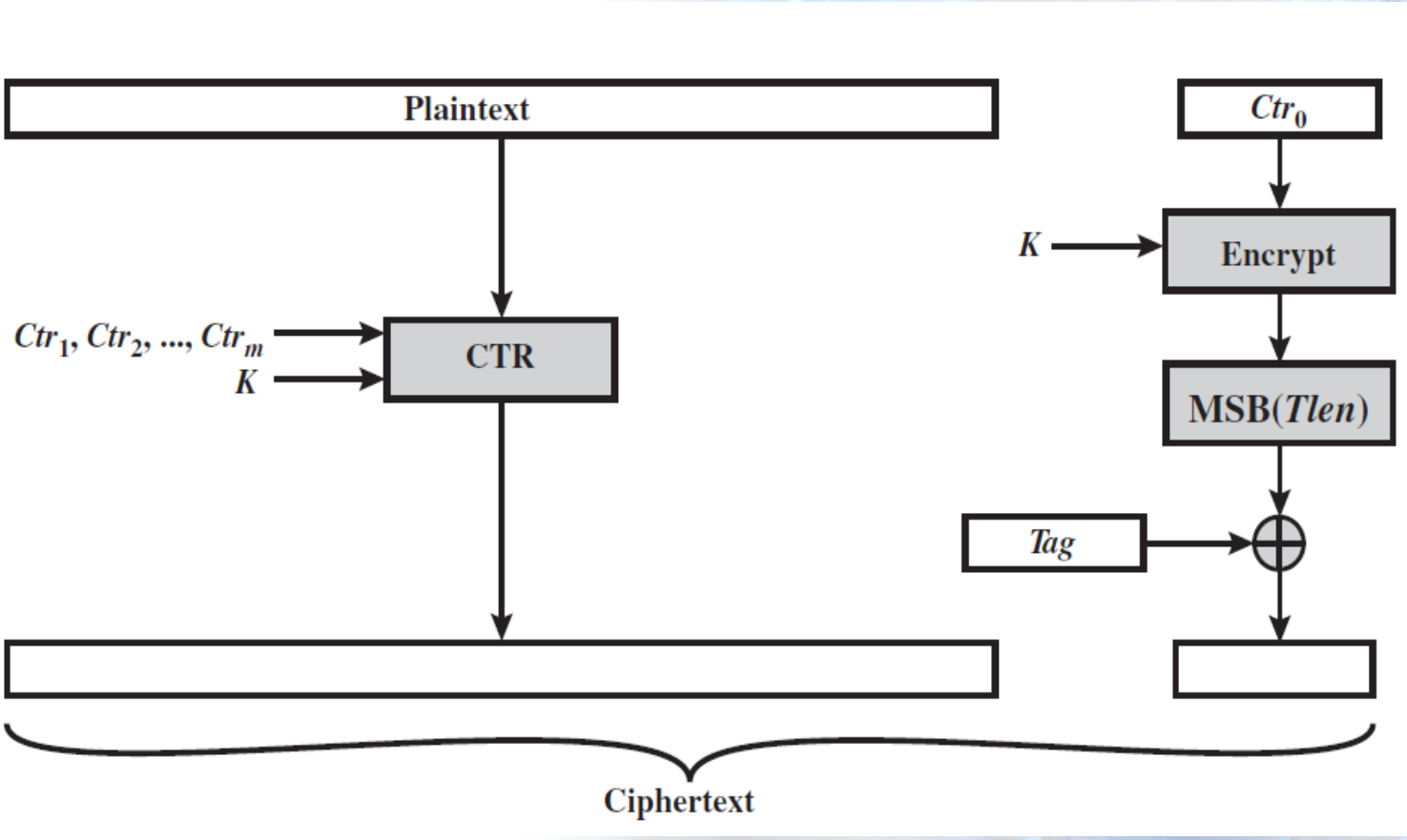
- The Tlen most significant bits of the output are XORed with the tag to produce an encrypted tag.
- The remaining counters are used for the CTR mode encryption of the plaintext.

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.

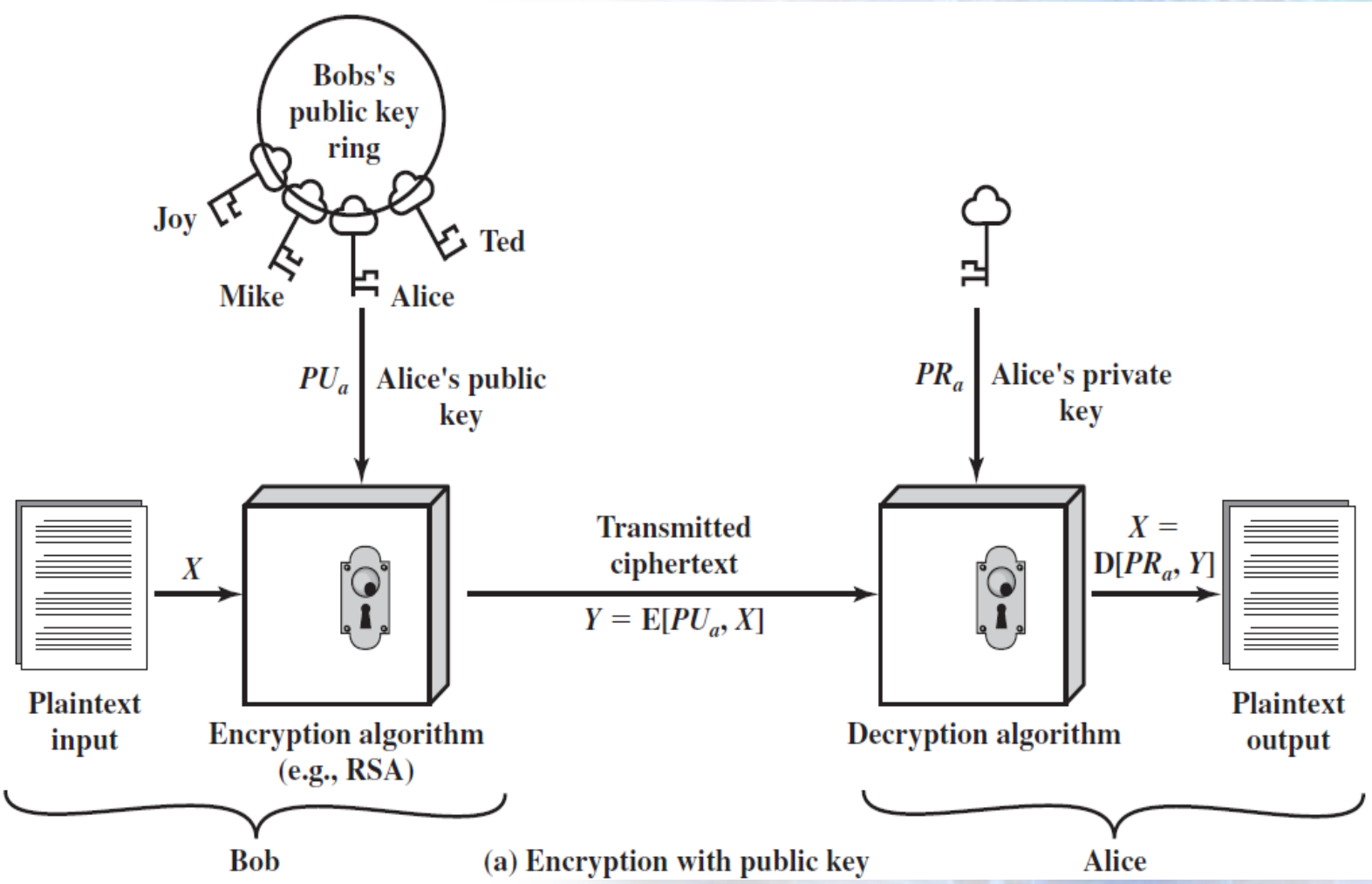
Public-Key Encryption Structure

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

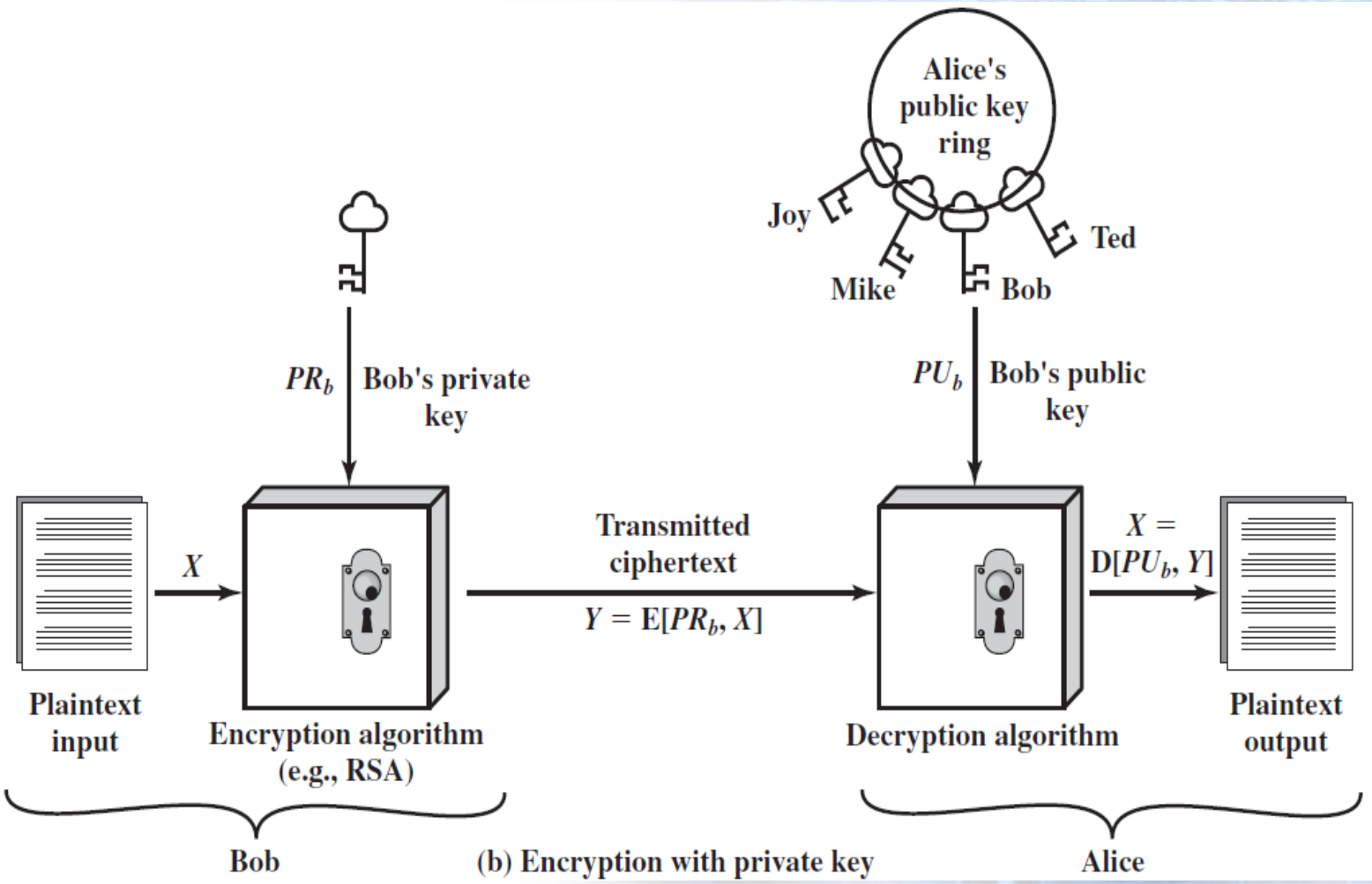
Public-Key Encryption Structure

- 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



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

Diffie-Hellman Algorithm

Network Security

Diffie-Hellman Algorithm

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe the Diffie-Hellman algorithm.

Diffie-Hellman Algorithm

Figures and material in this topic have been adapted from

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

Diffie-Hellman Algorithm

- The first published public-key algorithm that defined public-key cryptography was published by Diffie and Hellman.
- It is generally referred to as the Diffie – Hellman key exchange.

Diffie-Hellman Algorithm

- A number of commercial products employ this key exchange technique.

Diffie-Hellman Algorithm

- Purpose of the algorithm is to enable two users to exchange a secret key securely that then can be used for subsequent encryption of messages
- The algorithm itself is limited to the exchange of the keys.

Diffie-Hellman Algorithm

- Depends for its effectiveness on the difficulty of computing discrete logarithms.

Diffie-Hellman Algorithm

The Algorithm

- Lets assume that there are two publicly known numbers: a prime number q and an integer α that is a primitive root of q.

Diffie-Hellman Algorithm

- For a prime number p, if α is a primitive root of p, then α, α^2,…, α^{p-1} are distinct (mod p).
- E.g. for prime number 19, its primitive roots are 2, 3, 10, 13, 14, and 15.
- Suppose the users A and B wish to create a shared key.

Diffie-Hellman Algorithm

- User A selects a random integer $X_A < q$ and computes $Y_A = \alpha^{XA} \bmod q$.
- Similarly, user B independently selects a random integer $X_B < q$ and computes $Y_B = \alpha^{XB} \bmod q$.

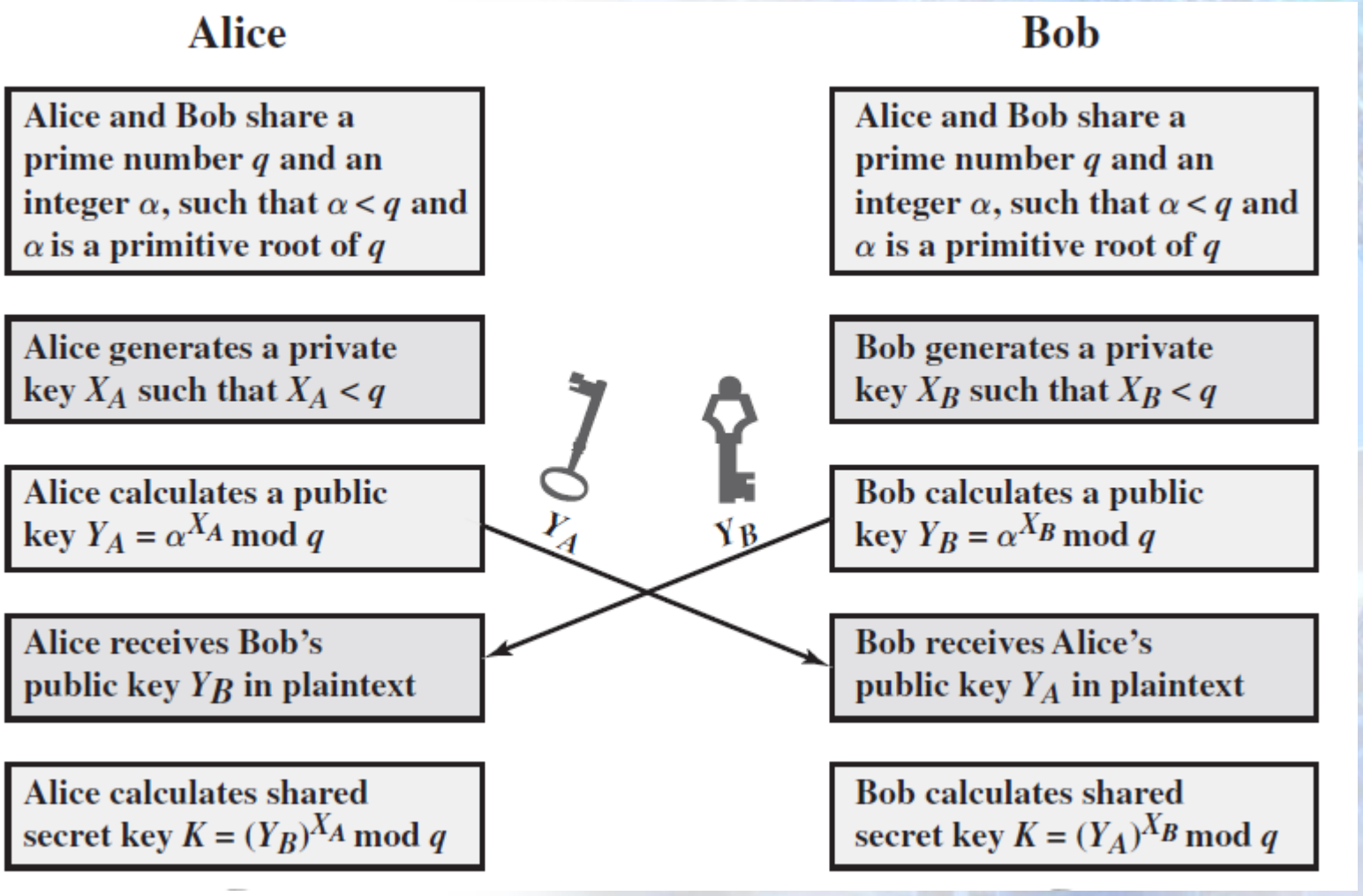
Diffie-Hellman Algorithm

- Each side keeps the X value private and makes the Y value available publicly to the other side.
- Thus, X_A is A's private key and Y_A is A's corresponding public key,
- The same applies for B.

Diffie-Hellman Algorithm

- User A computes the key as $K = (Y_B)^{XA} \bmod q$ and
- User B computes the key as $K = (Y_A)^{XB} \bmod q$.
- The result is that the two sides have exchanged a secret value.

Diffie-Hellman Algorithm



Diffie-Hellman Algorithm

Example:

- Lets take q = 353 and a primitive root of 353, $\alpha = 3$.
- A and B select private keys $X_A = 97$ and $X_B = 233$, respectively.

Diffie-Hellman Algorithm

- Each computes its public key:
- A computes $Y_A = 3^{97} \bmod 353 = 40$.
- B computes $Y_B = 3^{233} \bmod 353 = 248$.
- After they exchange public keys, each can compute the common secret key:

Diffie-Hellman Algorithm

- A computes $K = (Y_B)^{XA} \bmod 353 = 248^{97} \bmod 353 = 160$.
- B computes $K = (Y_A)^{XB} \bmod 353 = 40^{233} \bmod 353 = 160$.
- an attacker would have available the following information:
- $q = 353$; $\alpha = 3$; $Y_A = 40$; $Y_B = 248$.

Diffie-Hellman Algorithm

- In this simple example, it would be possible by brute force to determine the secret key 160.

Diffie-Hellman Algorithm

End

Security of the Diffie-Hellman algorithm:

- While it is relatively easy to calculate exponentials modulo a prime, it is very difficult to calculate discrete logarithms.
- For large primes, the latter task is considered infeasible.

Key Exchange, Man-in-the-Middle Attack

Network Security

Key Exchange, Man-in-the-Middle Attack

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe Man-in-the-Middle Attack while performing key exchange.

Key Exchange, Man-in-the-Middle Attack

Figures and material in this topic have been adapted from

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

Key Exchange, Man-in-the-Middle Attack

Key Exchange Protocol: An Example

- Suppose that user A wishes to set up a connection with user B and use a secret key to encrypt messages on that connection.

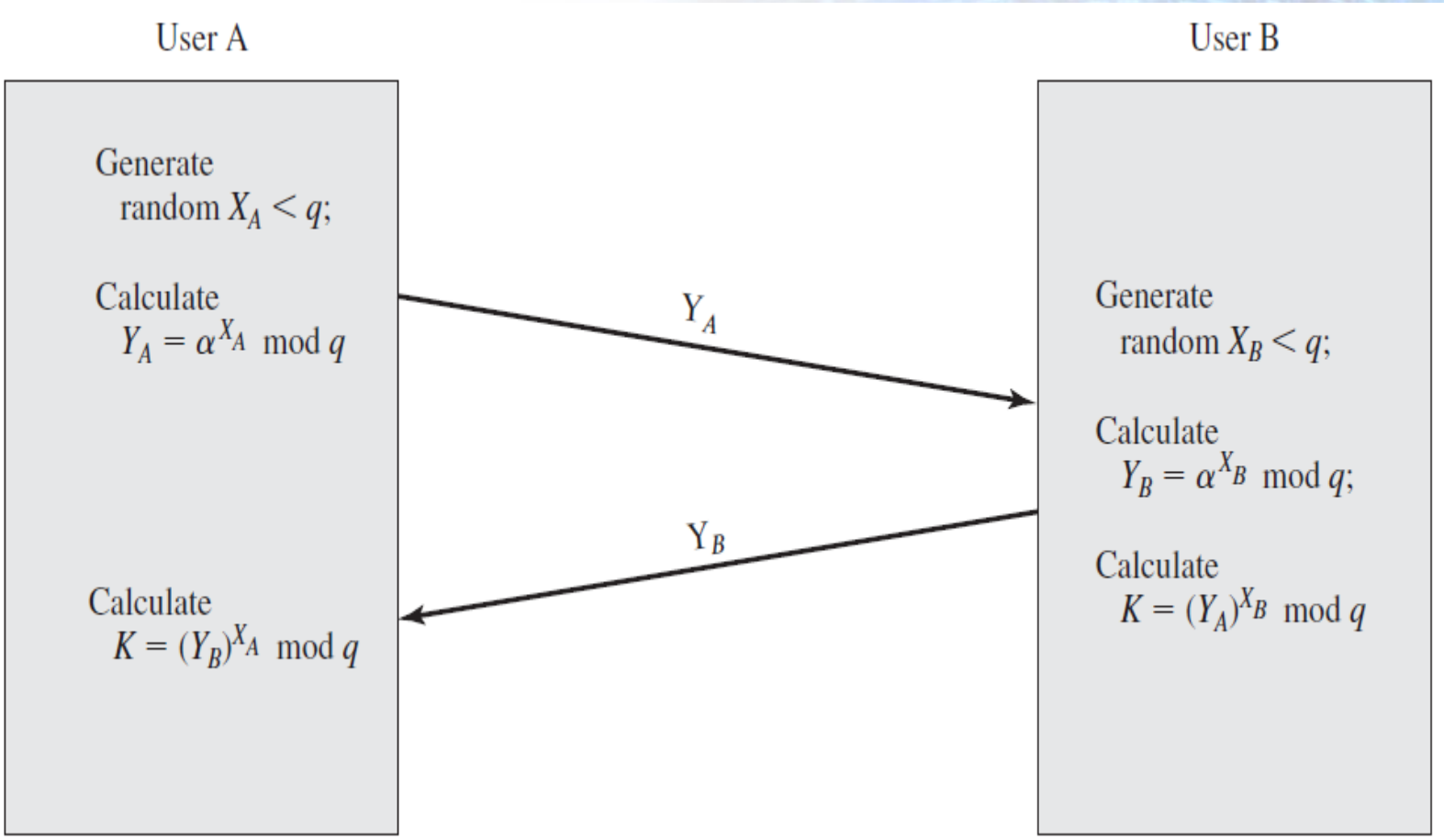
Key Exchange, Man-in-the-Middle Attack

- User A can generate a one-time private key X_A, calculate public value Y_A, and send that to user B.
- User B responds by generating a private value X_B, calculating Y_B, and sending Y_B to user A.

Key Exchange, Man-in-the-Middle Attack

- Both users can now calculate the key.
- The necessary public values q and α would need to be known ahead of time.

Key Exchange, Man-in-the-Middle Attack



Key Exchange, Man-in-the-Middle Attack

Key Exchange Protocol: Another Example

- Suppose that a group of users (e.g., all users on a LAN) each generate a long-lasting private value X_i (for user i) and calculate a public value Y_i.

Key Exchange, Man-in-the-Middle Attack

- These public values, together with global public values for q and α, are stored in some central directory.

Key Exchange, Man-in-the-Middle Attack

- At any time, user *j* can access user *i*'s public value, calculate a secret key, and use that to send an encrypted message to user *i*.

Key Exchange, Man-in-the-Middle Attack

- If the central directory is trusted, then this form of communication provides both confidentiality and a degree of authentication.

Key Exchange, Man-in-the-Middle Attack

- However, the technique does not protect against replay attacks.

Key Exchange, Man-in-the-Middle Attack

Man-in-the-Middle Attack

- Suppose Alice and Bob wish to exchange keys, and Darth is the adversary.
- The man-in-the-middle attack proceeds as follows

Key Exchange, Man-in-the-Middle Attack

- 1. Darth prepares for the attack by generating two random private keys X_{D1} and X_{D2} and then computing the corresponding public keys Y_{D1} and Y_{D2}.

Key Exchange, Man-in-the-Middle Attack

- 2. Alice transmits Y_A to Bob.
- 3. Darth intercepts Y_A and transmits Y_{D1} to Bob. Darth also calculates $K_2 = (Y_A)X_{D2} \bmod q$.

Key Exchange, Man-in-the-Middle Attack

- 4. Bob receives Y_{D1} and calculates $K_1 = (Y_{D1})X_B \bmod q$.
- 5. Bob transmits Y_B to Alice.
- 6. Darth intercepts Y_B and transmits Y_{D2} to Alice. Darth calculates $K_1 = (Y_B)X_{D1} \bmod q$.

Key Exchange, Man-in-the-Middle Attack

- 7. Alice receives Y_{D2} and calculates $K_2 = (Y_{D2})X_A \bmod q$.

Key Exchange, Man-in-the-Middle Attack

- At this point, Bob and Alice think that they share a secret key, but instead Bob and Darth share secret key K1 and Alice and Darth share secret key K2.
- All future communication bet. Bob and Alice works in following way.

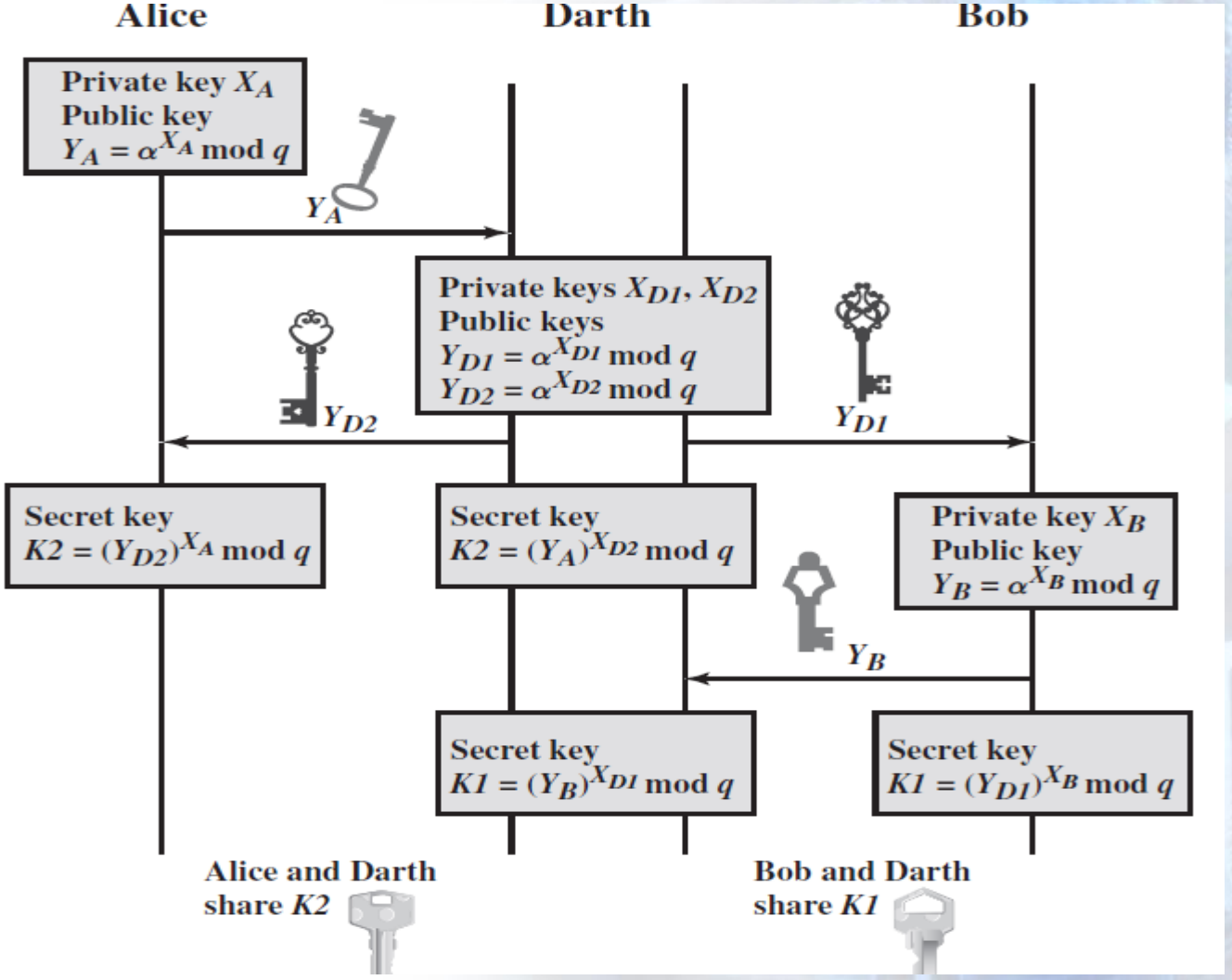
Key Exchange, Man-in-the-Middle Attack

- 1. Alice sends an encrypted message M: E(K2, M).
- 2. Darth intercepts the encrypted message and decrypts it to recover M.
- 3. Darth sends Bob E(K1, M) or E(K1, M'), where M' is any message.

Key Exchange, Man-in-the-Middle Attack

- In the first case, Darth simply wants to eavesdrop on the communication without altering it. In the second case, Darth wants to modify the message going to Bob.

Key Exchange, Man-in-the-Middle Attack



Key Exchange, Man-in-the-Middle Attack

- The key exchange protocol is vulnerable to such an attack because it does not authenticate the participants.

End

Digital Signature Standard (DSS)

Network Security

Digital Signature Standard (DSS)

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain Digital Signature Standard (DSS).

Digital Signature Standard (DSS)

Figures and material in this topic have been adapted from

- *W. Stalling's "Network Security Essentials "*, 2014.
- *W. Stalling's "Cryptography and Network Security Principles and Practice"*, 2014.

Digital Signature Standard (DSS)

- The most important development from the work on public-key cryptography is the digital signature.
- It provides a set of security capabilities that would be difficult to implement in any other way.

Digital Signature Standard (DSS)

- The National Institute of Standards and Technology (NIST) has published Federal Information Processing Standard FIPS 186, known as the Digital Signature Standard (DSS) or the Digital Signature Algorithm (DSA).

Digital Signature Standard (DSS)

- The DSA makes use of the Secure Hash Algorithm (SHA).
- The DSA was originally proposed in 1991 and revised in 1993 in response to public feedback concerning the security of the scheme.

Digital Signature Standard (DSS)

- There was a further minor revision in 1996.
- In 2000, an expanded version of the standard was issued as FIPS 186-2, subsequently updated to FIPS 186-3 in 2009.

Digital Signature Standard (DSS)

The DSA Approach

- employs an algorithm designed to provide only the digital signature function.
- it is a public-key technique.
- Unlike RSA, it cannot be used for encryption or key exchange.

Digital Signature Standard (DSS)

- **The RSA Approach for Digital Signatures:**
- When RSA is used for digital signatures, the message to be signed is input to a hash function that produces a secure hash code of fixed length.

Digital Signature Standard (DSS)

- This hash code is then encrypted using the sender's private key to form the signature.
- Both the message and the signature are then transmitted.
- The recipient takes the message and produces a hash code.

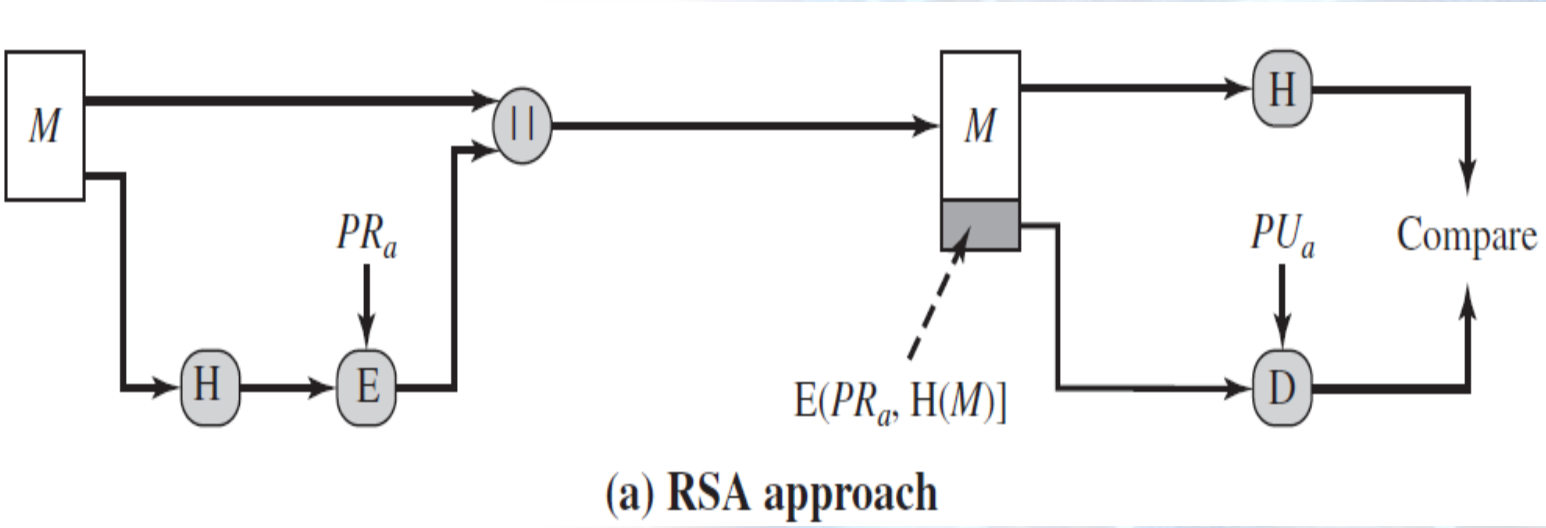
Digital Signature Standard (DSS)

- The recipient also decrypts the signature using the sender's public key.
- If the calculated hash code matches the decrypted signature, the signature is accepted as valid.

Digital Signature Standard (DSS)

- Because only the sender knows the private key, only the sender could have produced a valid signature.

Digital Signature Standard (DSS)



(a) RSA approach

Digital Signature Standard (DSS)

- **The DSA Approach:**
- also makes use of a hash function.
- The hash code is provided as input to a signature function along with a random number k generated for this particular signature.

Digital Signature Standard (DSS)

- The signature function also depends on the sender's private key (PRa) and a set of parameters known to a group of communicating principals.

Digital Signature Standard (DSS)

- We can consider this set to constitute a global public key (PUG).
- The result is a signature consisting of two components, labeled s and r.

Digital Signature Standard (DSS)

- At the receiving end, the hash code of the incoming message is generated.
- This plus the signature is input to a verification function.

Digital Signature Standard (DSS)

- The verification function also depends on the global public key as well as the sender's public key (PUa), which is paired with the sender's private key.

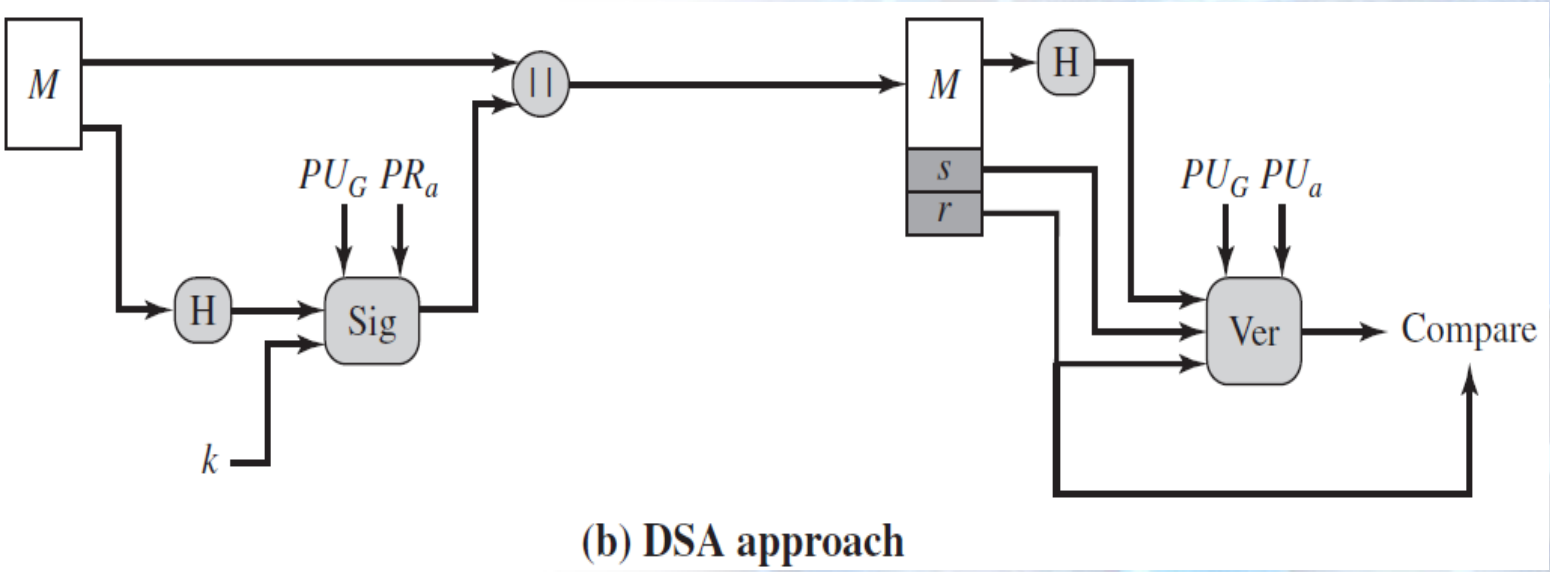
Digital Signature Standard (DSS)

- The output of the verification function is a value that is equal to the signature component r if the signature is valid.

Digital Signature Standard (DSS)

- The signature function is such that only the sender, with knowledge of the private key, could have produced the valid signature.

Digital Signature Standard (DSS)



(b) DSA approach

Elliptic-Curve Cryptography (ECC)

Network Security

Elliptic-Curve Cryptography (ECC)

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe Elliptic-Curve Cryptography (ECC).

Elliptic-Curve Cryptography (ECC)

Figures and material in this topic have been adapted from

- *W. Stalling's "Network Security Essentials "*, 2014.
- *W. Stalling's "Cryptography and Network Security Principles and Practice"*, 2014.

Elliptic-Curve Cryptography (ECC)

- Most of the products and standards that use public-key cryptography for encryption and digital signatures use RSA.

Elliptic-Curve Cryptography (ECC)

- The key length for secure RSA use has increased over recent years, and this has put a heavier processing load on applications using RSA.

Elliptic-Curve Cryptography (ECC)

- This burden has ramifications, especially for electronic commerce sites that conduct large numbers of secure transactions.

Elliptic-Curve Cryptography (ECC)

- A competing system challenges RSA: elliptic curve cryptography (ECC).
- It is showing up in standardization efforts, including the IEEE P1363 Standard for Public-Key Cryptography.

Elliptic-Curve Cryptography (ECC)

- The principal attraction of ECC, compared to RSA, is that it appears to offer equal security for a far smaller key size, thereby reducing processing overhead.

Elliptic-Curve Cryptography (ECC)

- Although the theory of ECC has been around for some time, it is only recently that products have begun to appear and that there has been sustained cryptanalytic interest in probing for weaknesses.

Elliptic-Curve Cryptography (ECC)

- Accordingly, the confidence level in ECC is not yet as high as that in RSA.

Elliptic-Curve Cryptography (ECC)

- An elliptic curve is defined by an equation in two variables with coefficients.
- For cryptography, the variables and coefficients are restricted to elements in a finite field.

Elliptic-Curve Cryptography (ECC)

- The addition operation in ECC is the counterpart of modular multiplication in RSA, and multiple addition is the counterpart of modular exponentiation.

Elliptic-Curve Cryptography (ECC)

Analog of Diffie-Hellman Key Exchange

- First pick a large integer q, which is either a prime number p or an integer of the form 2^m, and elliptic curve parameters a and b.

Elliptic-Curve Cryptography (ECC)

- This defines the elliptic group of points $E_q(a, b)$.
- Next, pick a base point $G = (x_1, y_1)$ in $E_p(a, b)$ whose order is a very large value n.

Elliptic-Curve Cryptography (ECC)

- 1. A selects an integer n_A less than n.
- This is A's private key.
- A then generates a public key $P_A = n_A \times G$; the public key is a point in $E_q(a, b)$.

Elliptic-Curve Cryptography (ECC)

- 2. B similarly selects a private key n_B and computes a public key P_B.
- 3. A generates the secret key $k = n_A \times P_B$. B generates the secret key $k = n_B \times P_A$.

Elliptic-Curve Cryptography (ECC)

- The two calculations in step 3 produce the same result because $n_A \times P_B = n_A \times (n_B \times G) = n_B \times (n_A \times G) = n_B \times P_A$
- To break this scheme, an attacker would need to be able to compute k given G and kG, which is assumed to be hard.

Elliptic-Curve Cryptography (ECC)

Elliptic Curve Encryption/Decryption:

- It requires a point G and an elliptic group $E_q(a, b)$ as parameters.
- Each user A selects a private key n_A and generates a public key $P_A = n_A \times G$.

Elliptic-Curve Cryptography (ECC)

- To encrypt and send a message P_m to B, A chooses a random positive integer k and produces the ciphertext C_m consisting of the pair of points:
- $C_m = \{kG, P_m + kP_B\}$
- Note that A has used B's public key P_B.

Elliptic-Curve Cryptography (ECC)

- To decrypt the ciphertext, B multiplies the first point in the pair by B's private key and subtracts the result from the second point:
- $P_m + kP_B - n_B(kG) = P_m + k(n_BG) - n_B(kG) = P_m$

Elliptic-Curve Cryptography (ECC)

- A has masked the message P_m by adding kP_B to it.
- Nobody but A knows the value of k, so even though P_B is a public key, nobody can remove the mask kP_B.

Elliptic-Curve Cryptography (ECC)

- There is a computational advantage to using ECC with a shorter key length than a comparably secure RSA.

End

Digital Signatures

Network Security

Digital Signatures

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain the use of digital signatures.

Digital Signatures

Figures and material in this topic have been adapted from

- *W. Stalling's "Network Security Essentials "*, 2014.
- *W. Stalling's "Cryptography and Network Security Principles and Practice"*, 2014.

Digital Signatures

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

Digital Signatures

- Message authentication protects two parties who exchange messages from any third party.
- It does not protect the two parties against each other. Several forms of dispute bet. the two are possible.

Digital Signatures

- Suppose that John sends an authenticated message to Mary.
- Consider the following disputes that could arise.

Digital Signatures

- 1. Mary may forge a different message and claim that it came from John. Mary would simply have to create a message and append an authentication code using the key that John and Mary share.

Digital Signatures

- E.g. an electronic funds transfer takes place, and the receiver increases the amount of funds transferred and claims that the larger amount had arrived from the sender.

Digital Signatures

- 2. John can deny sending the message. Because it is possible for Mary to forge a message, there is no way to prove that John did in fact send the message.

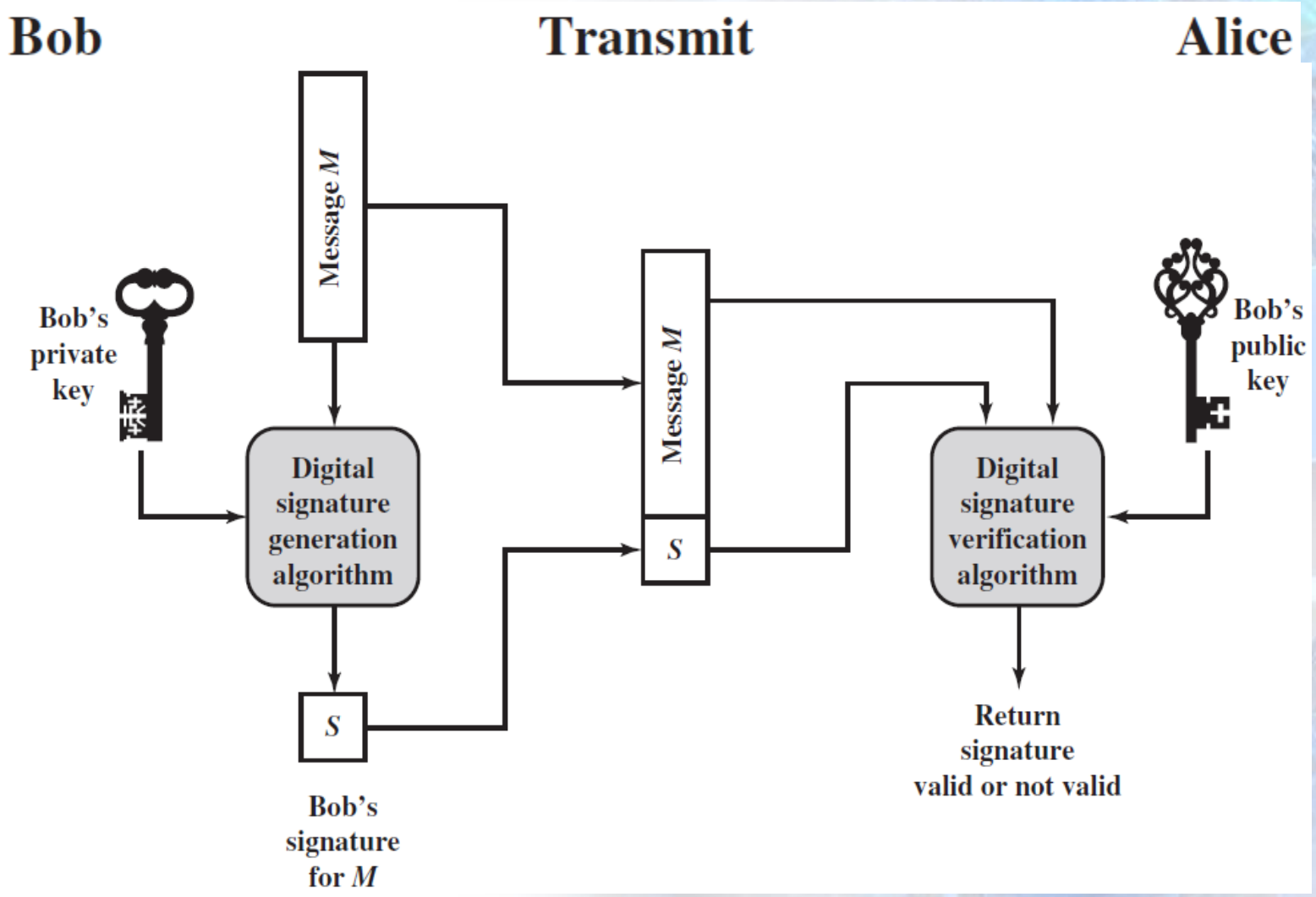
Digital Signatures

- E.g. an electronic mail message contains instructions to a stockbroker for a transaction that subsequently turns out badly. The sender pretends that the message was never sent.

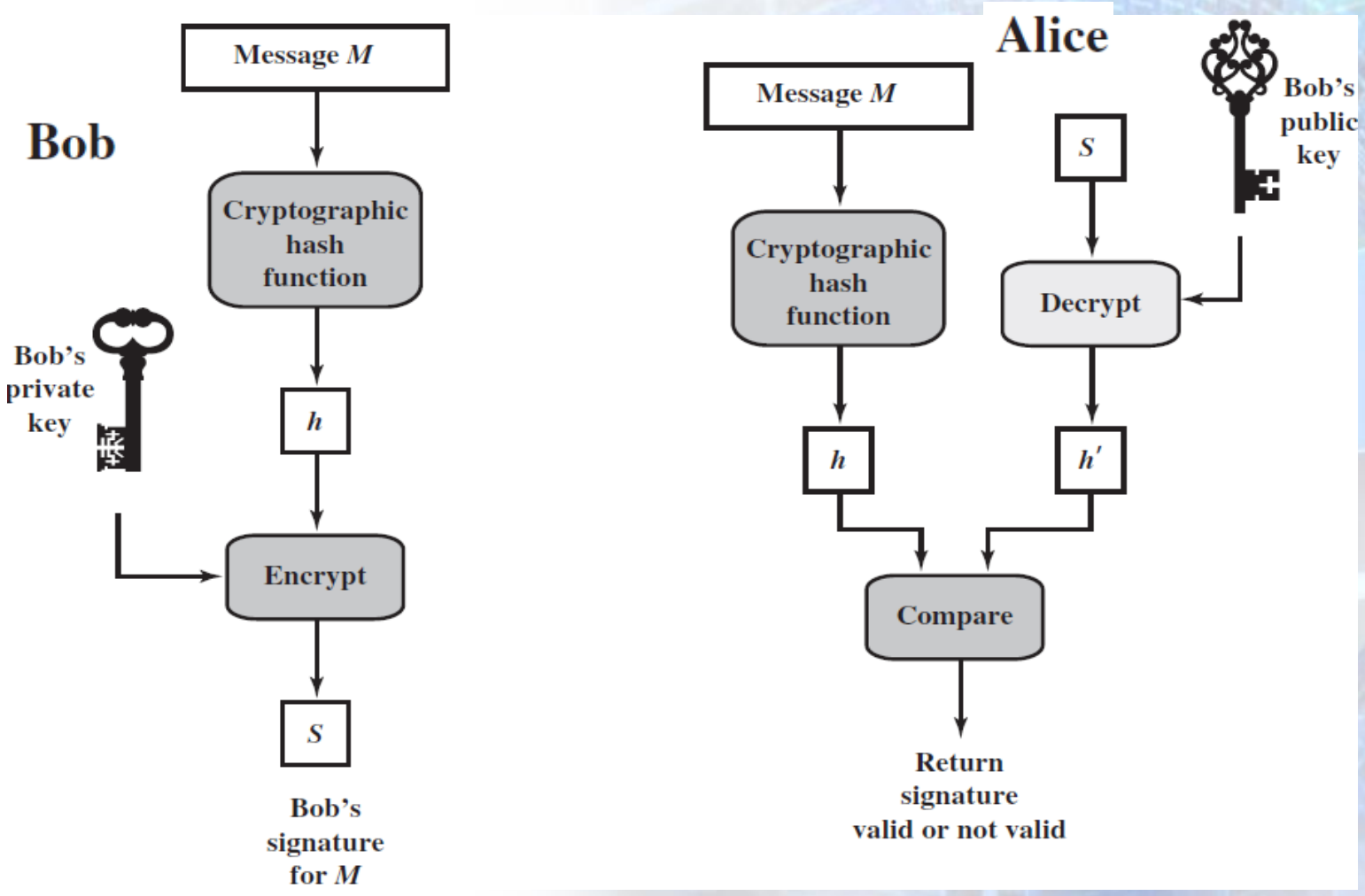
Digital Signatures

- In situations where there is not complete trust between sender and receiver, we use the digital signature.

Digital Signatures



Digital Signatures



Digital Signatures

Properties

- It must verify the author and the date and time of the signature.
- It must authenticate the contents at the time of the signature.
- It must be verifiable by third parties, to resolve disputes.

Digital Signatures

Attacks and Forgeries

- Assume that A denotes the user whose signature method is being attacked, and C denotes the attacker.

Digital Signatures

- **Key-only attack:** C only knows A's public key.
- **Known message attack:** C is given access to a set of messages and their signatures

Digital Signatures

- **Generic chosen message attack:** C chooses a list of messages before attempting to breaks A's signature scheme, independent of A's public key.
- C then obtains from A valid signatures for the chosen messages.

Digital Signatures

- The attack is generic, because it does not depend on A's public key; the same attack is used against everyone.

Digital Signatures

- **Directed chosen message attack:** Similar to the generic attack, except that the list of messages to be signed is chosen after C knows A's public key but before any signatures are seen.

Digital Signatures

- **Adaptive chosen message attack:** C is allowed to use A as an "oracle." This means that C may request from A signatures of messages that depend on previously obtained message-signature pairs.

Digital Signatures

Digital Signature Requirements

- The signature must be a bit pattern that depends on the message being signed.
- The signature must use some information unique to the sender to prevent both forgery and denial.

Digital Signatures

- It must be relatively easy to produce the digital signature.
- It must be relatively easy to recognize and verify the digital signature.

Digital Signatures

- It must be computationally infeasible to forge a digital signature, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message.

Digital Signatures

- It must be practical to retain a copy of the digital signature in storage.

End

Symmetric Key Distribution

Network Security

Symmetric Key Distribution

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe how symmetric key can be distributed with symmetric encryption.

Symmetric Key Distribution

Figures and material in this topic have been adapted from

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

Symmetric Key Distribution

- For symmetric encryption to work, the two parties to an exchange must share the same key, and that key must be protected from access by others.

Symmetric Key Distribution

- Frequent key changes are usually desirable to limit the amount of data compromised if an attacker learns the key.

Symmetric Key Distribution

- The strength of any cryptographic system rests with the “key distribution technique” --- the means of delivering a key to two parties that wish to exchange data, without allowing others to see the key.

Symmetric Key Distribution

Key Distribution Options

- For two parties A and B, there are the following options:
- 1. A key could be selected by A and physically delivered to B.

Symmetric Key Distribution

- 2. A third party could select the key and physically deliver it to A and B.
- 3. If A and B have recently used a key, one party could transmit the new key to the other, using the old key to encrypt the new key.

Symmetric Key Distribution

- 4. If A and B each have an encrypted connection to a third party C, C could deliver a key on the encrypted links to A and B.

Symmetric Key Distribution

- Options 1 and 2 call for manual delivery of a key.
- For link encryption, this is a reasonable requirement, because each link encryption device is only going to be exchanging data with its partner on the other end of the link.

Symmetric Key Distribution

- However, for end-to-end encryption over a network, manual delivery is awkward.

Symmetric Key Distribution

- In a distributed system, any given host may need to engage in exchanges with many other hosts over time.
- Each device needs a number of keys supplied dynamically.
- Difficult in a wide-area distributed system.

Symmetric Key Distribution

- Option 3 is a possibility for either link encryption or end-to-end encryption, but if an attacker ever succeeds in gaining access to one key, then all subsequent keys are revealed.

Symmetric Key Distribution

- To provide keys for end-to-end encryption, option 4 is preferable.
- For option 4 , two kinds of keys are used:

Symmetric Key Distribution

- **Session key:** When two end systems wish to communicate, they establish a logical connection.
- For the duration of that logical connection, called a session, all user data are encrypted with a one-time session key.

Symmetric Key Distribution

- At the conclusion of the session, the session key is destroyed.
- **Permanent key:** is a key used between entities for the purpose of distributing session keys.

Symmetric Key Distribution

- A necessary element of option 4 is a key distribution center (KDC) .
- The operation of a KDC proceeds as follows:

Symmetric Key Distribution

- 1. When host A wishes to set up a connection to host B, it transmits a connection request packet to the KDC.
- Communication bet. A and KDC is encrypted using a master key shared only by A and the KDC.

Symmetric Key Distribution

- 2. If KDC approves the connection request, it generates a unique one-time session key.
- It encrypts the session key using the permanent key it shares with A and delivers the encrypted session key to A.

Symmetric Key Distribution

- Similarly, it encrypts the session key using the permanent key it shares with B and delivers the encrypted session key to B.

Symmetric Key Distribution

- 3. A and B can now set up a logical connection and exchange messages and data, all encrypted using the temporary session key.

End

What is Kerberos?

Network Security

What is Kerberos?

Objectives of the Topic

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

What is Kerberos?

Figures and material in this topic have been adapted from

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

What is Kerberos?

- Kerberos is a key distribution and user authentication service developed at MIT as part of Project Athena.

What is Kerberos?

- The problem that Kerberos addresses is this:
- Assume an open distributed environment in which users at workstations wish to access services on servers distributed throughout network.

What is Kerberos?

- We would like for servers to be able to restrict access to authorized users and to be able to authenticate requests for service.
- A workstation cannot be trusted to identify its users correctly to network services.

What is Kerberos?

- In particular, the following three threats exist:
- 1. A user may gain access to a particular workstation and pretend to be another user operating from that workstation.

What is Kerberos?

- 2. A user may alter the network address of a workstation so that the requests sent from the altered workstation appear to come from the impersonated workstation.

What is Kerberos?

- 3. A user may eavesdrop on exchanges and use a replay attack to gain entrance to a server or to disrupt operations.

What is Kerberos?

- In any of these cases, an unauthorized user may be able to gain access to services and data that he or she is not authorized to access.

What is Kerberos?

- Rather than building elaborate authentication protocols at each server, Kerberos provides a centralized authentication server whose function is to authenticate users to servers and servers to users.

What is Kerberos?

- Kerberos relies exclusively on symmetric encryption, making no use of public-key encryption.

What is Kerberos?

Motivation

- If a set of users is provided with dedicated personal computers that have no network connections, then a user's resources and files can be protected by physically securing each computer.

What is Kerberos?

- When these users instead are served by a centralized time-sharing system, the time-sharing operating system can enforce access-control policies based on user identity and use the logon procedure to identify users.

What is Kerberos?

- More common is a distributed architecture consisting of dedicated users (clients) and distributed or centralized servers.

What is Kerberos?

- In this environment, there are three approaches to security.

What is Kerberos?

- 1. Rely on each individual client workstation to assure the identity of its user or users and rely on each server to enforce a security policy based on user identification (ID).

What is Kerberos?

- 2. Require that client systems authenticate themselves to servers, but trust the client system concerning the identity of its user.

What is Kerberos?

- 3. Require the user to prove his or her identity for each service invoked. Also require that servers prove their identity to clients.

What is Kerberos?

- In a small, closed environment in which all systems are owned and operated by a single organization, the first or perhaps the second strategy may suffice.

What is Kerberos?

- In a more open environment in which network connections to other machines are supported, the third approach is needed to protect user information and resources housed at the server.

What is Kerberos?

End

- Kerberos supports this third approach.
- Kerberos assumes a distributed client/server architecture and employs one or more Kerberos servers to provide an authentication service.

A Simple Authentication Dialogue

Network Security

A Simple Authentication Dialogue

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain a simple authentication dialogue.

A Simple Authentication Dialogue

Figures and material in this topic have been adapted from

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

A Simple Authentication Dialogue

- Version 4 of Kerberos makes use of DES, in a rather elaborate protocol, to provide the authentication service.
- To develop understanding about the full protocol, lets look at a simple dialogue.

A Simple Authentication Dialogue

- In an unprotected network environment, any client can apply to any server for service.
- The obvious security risk is that of impersonation.

A Simple Authentication Dialogue

- An opponent can pretend to be another client and obtain unauthorized privileges on server machines.
- To counter this threat, servers must be able to confirm the identities of clients who request service.

A Simple Authentication Dialogue

- Each server can be required to undertake this task for each client/server interaction, but in an open environment, this places a substantial burden on each server.

A Simple Authentication Dialogue

- An alternative is to use an authentication server (AS) that knows the passwords of all users and stores these in a centralized database.
- The AS shares a unique secret key with each server.

A Simple Authentication Dialogue

- If C = client, AS = authentication server, V = server, ID_C = identifier of user on C, ID_V = identifier of V, P_C = password of user on C, AD_C = network address of C, K_v = secret key shared by AS and V, then In a simple authentication scenario

A Simple Authentication Dialogue

- A user logs on to a workstation and requests access to server V.

A Simple Authentication Dialogue

- The client module C in the user's workstation requests the user's password and then sends a message to the AS that includes the user's ID, the server's ID, and the user's password.

A Simple Authentication Dialogue

- The AS checks its database to see if the user has supplied the proper password for this user ID and whether this user is permitted access to server V.

A Simple Authentication Dialogue

- If both tests are passed, the AS accepts the user as authentic and must now convince the server that this user is authentic.

A Simple Authentication Dialogue

- To do so, the AS creates a ticket that contains the user's ID and network address and the server's ID.
- This ticket is encrypted using the secret key shared by the AS and this server.
- This ticket is then sent back to C.

A Simple Authentication Dialogue

- As the ticket is encrypted, it cannot be altered by C or by an opponent.
- With this ticket, C can now apply to V for service.
- C sends a message to V containing C's ID and the ticket.

A Simple Authentication Dialogue

- V decrypts the ticket and verifies that the user ID in the ticket is the same as the unencrypted user ID in the message.
- If the two match, the server considers the user authenticated and grants the requested service.

A Simple Authentication Dialogue

$$\textbf{(1)}\ \mathrm{C} \rightarrow \mathrm{AS}: \quad ID_C \| P_C \| ID_V$$

$$\textbf{(2)}\ \mathrm{AS} \rightarrow \mathrm{C}: \quad Ticket$$

$$\textbf{(3)}\ \mathrm{C} \rightarrow \mathrm{V}: \quad ID_C \| Ticket$$

$$Ticket = E(K_v, [ID_C \| AD_C \| ID_V])$$

A Simple Authentication Dialogue

- In message (3), the ticket is encrypted to prevent alteration.
- Server's ID is included -- server can verify that it has decrypted the ticket properly.
- ID_C is included to indicate that this ticket has been issued on behalf of C.

A Simple Authentication Dialogue

- **A possible attack:** An opponent could capture the ticket transmitted in message (2), then use the name ID_C and transmit a message of form (3) from another workstation.

A Simple Authentication Dialogue

End

- The server would receive a valid ticket that matches the user ID and grant access to the user on that other workstation.
- To prevent this attack, the AS includes in the ticket the network address from which original request came.

A More Secure Authentication Dialogue



A More Secure Authentication Dialogue

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe a more secure authentication dialogue.

A More Secure Authentication Dialogue

Figures and material in this topic have been adapted from

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

A More Secure Authentication Dialogue

- Two issues with the simple authentication dialogue :
- 1. The number of times a user has to enter a password should be minimized.
- 2. A plaintext transmission of the password should be avoided.

A More Secure Authentication Dialogue

- Suppose each ticket can be used only once.
- If user C logs on to a workstation in the morning and checks his or her mail at a mail server, C has to supply a password to get a ticket for the mail server.

A More Secure Authentication Dialogue

- Similarly, if a user wishes to access a print server, a mail server, a file server, and so on, the first instance of each access would require a new ticket and hence require the user to enter the password.

A More Secure Authentication Dialogue

- This authentication scheme employs a new server, known as the **ticket-granting server** (TGS).
- TGS, issues tickets to users who have been authenticated to AS.
- The user first requests a ticket-granting ticket ($Ticket_{tgs}$) from the AS.

A More Secure Authentication Dialogue

- The client module in the user workstation saves this ticket.
- Each time the user requires access to a new service, the client applies to the TGS, using the ticket to authenticate itself.

A More Secure Authentication Dialogue

- The TGS then grants a ticket for the particular service.
- The client saves each service-granting ticket and uses it to authenticate its user to a server each time a particular service is requested.

A More Secure Authentication Dialogue

Details

- 1. The client requests a ticket-granting ticket on behalf of the user by sending its user's ID to the AS, together with the TGS ID, indicating a request to use the TGS service.

A More Secure Authentication Dialogue

- 2. The AS responds with a ticket that is encrypted with a key that is derived from the user's password (K_C), which is already stored at the AS.

A More Secure Authentication Dialogue

- When this response arrives at the client, the client prompts the user for his or her password, generates the key, and attempts to decrypt the incoming message.

A More Secure Authentication Dialogue

- If correct password is supplied, the ticket is successfully recovered.
- Thus, we have used the password to obtain credentials from Kerberos without having to transmit the password in plaintext.

A More Secure Authentication Dialogue

- The ticket-granting ticket is to be reusable.
- The client now has a reusable ticket and need not bother the user for a password for each new service request.

A More Secure Authentication Dialogue

- To avoid an opponent from capturing and using the ticket, the ticket includes a timestamp, indicating the date and time at which the ticket was issued, and a lifetime, indicating the length of time for which the ticket is valid.

A More Secure Authentication Dialogue

- 3. The client requests a service-granting ticket on behalf of the user. For this purpose, the client transmits a message to the TGS containing the user's ID, the ID of the desired service, and the ticket-granting ticket.

A More Secure Authentication Dialogue

- 4. The TGS decrypts the incoming ticket using a key shared only by the AS and the TGS (K_{tgs}) and verifies the success of the decryption by the presence of its ID. It checks to make sure that the lifetime has not expired.

A More Secure Authentication Dialogue

- Then it compares the user ID and network address with the incoming information to authenticate the user. If user is permitted access to the server V, the TGS issues a ticket to grant access to the requested service.

A More Secure Authentication Dialogue

- 5. The client requests access to a service on behalf of the user. For this purpose, the client transmits a message to the server containing the user's ID and the service granting ticket. The server authenticates by using the contents of the ticket.

A More Secure Authentication Dialogue

Once per user logon session:

(1) $C \rightarrow AS: \quad ID_C \| ID_{tgs}$

(2) $AS \rightarrow C: \quad E(K_c, Ticket_{tgs})$

Once per type of service:

(3) $C \rightarrow TGS: \quad ID_C \| ID_V \| Ticket_{tgs}$

(4) $TGS \rightarrow C: \quad Ticket_v$

Once per service session:

(5) $C \rightarrow V: \quad ID_C \| Ticket_v$

$$Ticket_{tgs} = E(K_{tgs}, [ID_C \| AD_C \| ID_{tgs} \| TS_1 \| Lifetime_1])$$

$$Ticket_v = E(K_v, [ID_C \| AD_C \| ID_v \| TS_2 \| Lifetime_2])$$

A More Secure Authentication Dialogue

- This new scenario satisfies the two requirements of only one password query per user session and protection of the user password.

End

The Version 4 Authentication Dialogue

Network Security

The Version 4 Authentication Dialogue

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain Kerberos V4 authentication dialogue.

The Version 4 Authentication Dialogue

Figures and material in this topic have been adapted from

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

The Version 4 Authentication Dialogue

- Two issues with the more secure authentication dialogue:
- First problem is the lifetime associated with the ticket-granting ticket.

The Version 4 Authentication Dialogue

- If this lifetime is very short (e.g., minutes), then the user will be repeatedly asked for a password.
- If the lifetime is long (e.g., hours), then an opponent has a greater opportunity for replay.

The Version 4 Authentication Dialogue

- An opponent could capture a copy of the ticket-granting ticket and then wait for the legitimate user to log out.
- Then opponent could forge the legitimate user's network address and send the message to the TGS.

The Version 4 Authentication Dialogue

- This would give the opponent unlimited access to the resources and files available to the legitimate user.
- A similar issue arises when an opponent captures a service-granting ticket.

The Version 4 Authentication Dialogue

- Thus, we arrive at an additional requirement.
- A network service (the TGS or an application service) must be able to prove that the person using a ticket is the same person to whom that ticket was issued.

The Version 4 Authentication Dialogue

- The second problem is that servers should authenticate themselves to users.
- An opponent can sabotage the configuration so that messages to a server were directed to another location.

The Version 4 Authentication Dialogue

Kerberos Solution:

- The threat is that an opponent will steal the ticket and use it before it expires.
- Let the AS provide both the client and the TGS with a secret piece of information in a secure manner.

The Version 4 Authentication Dialogue

- Then client can prove its identity to TGS by revealing secret information, again in a secure manner.
- Kerberos uses an encryption key as the secure information; this is referred to as a session key.

The Version 4 Authentication Dialogue

- The client sends a message to the AS requesting access to the TGS.
- The AS responds with a message, encrypted with a key derived from the user's password (K_C), that contains the ticket.

The Version 4 Authentication Dialogue

- The encrypted message also contains a copy of the session key, $K_{C,tgs}$.
- Because this session key is inside the message encrypted with K_C, only the user's client can read it.

The Version 4 Authentication Dialogue

- The same session key is included in the ticket, which can be read only by the TGS.
- Thus, the session key has been securely delivered to both C and the TGS.

The Version 4 Authentication Dialogue

(1) $C \rightarrow AS \quad ID_c \| ID_{tgs} \| TS_1$

(2) $AS \rightarrow C \quad E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$

$$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$$

(a) Authentication Service Exchange to obtain ticket-granting ticket

The Version 4 Authentication Dialogue

- Message (1) includes a timestamp, so that the AS knows that the message is timely.
- Message (2) includes several elements of the ticket in a form accessible to C.
- C learns that this ticket is for TGS and when it expires.

The Version 4 Authentication Dialogue

- Now C sends the TGS a message that includes the ticket plus the ID of the requested service.
- C also transmits an authenticator, which includes the ID and address of C's user and a timestamp.

The Version 4 Authentication Dialogue

- The authenticator is intended for use only once and has a very short lifetime.
- Now TGS can decrypt the ticket with the key that it shares with AS.
- This ticket indicates that user C has been provided with the session key $K_{C,tgs}$.

The Version 4 Authentication Dialogue

- The TGS uses the session key to decrypt the authenticator.
- The TGS can then check the name and address from the authenticator with that of the ticket and with the network address of the incoming message.

The Version 4 Authentication Dialogue

- If all match, then the TGS is assured that the sender of the ticket is indeed the ticket's real owner.

The Version 4 Authentication Dialogue

(3) C → TGS $ID_v \| Ticket_{tgs} \| Authenticator_c$

(4) TGS → C $E(K_{c,tgs}, [K_{c,v} \| ID_v \| TS_4 \| Ticket_v])$

$$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$$

$$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$$

$$Authenticator_c = E(K_{c,tgs}, [ID_C \| AD_C \| TS_3])$$

(b) Ticket-Granting Service Exchange to obtain service-granting ticket

The Version 4 Authentication Dialogue

- C now has a reusable service-granting ticket for V. When C presents this ticket, it also sends an authenticator.
- The server can decrypt the ticket, recover the session key, and decrypt the authenticator.

The Version 4 Authentication Dialogue

- If mutual authentication is required, the server returns the value of the timestamp from the authenticator, incremented by 1, and encrypted in the session key.
- C can decrypt this message.

The Version 4 Authentication Dialogue

(5) $C \rightarrow V$ $Ticket_v \| Authenticator_c$

(6) $V \rightarrow C$ $E(K_{c,v}, [TS_5 + 1])$ (for mutual authentication)

$$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$$

$$Authenticator_c = E(K_{c,v}, [ID_C \| AD_C \| TS_5])$$

(c) Client/Server Authentication Exchange to obtain service

Rationale For Elements Of Kerberos 4

Network Security

Rationale For Elements Of Kerberos 4

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe rationale for elements of Kerberos V4 authentication dialogue.

Rationale For Elements Of Kerberos 4

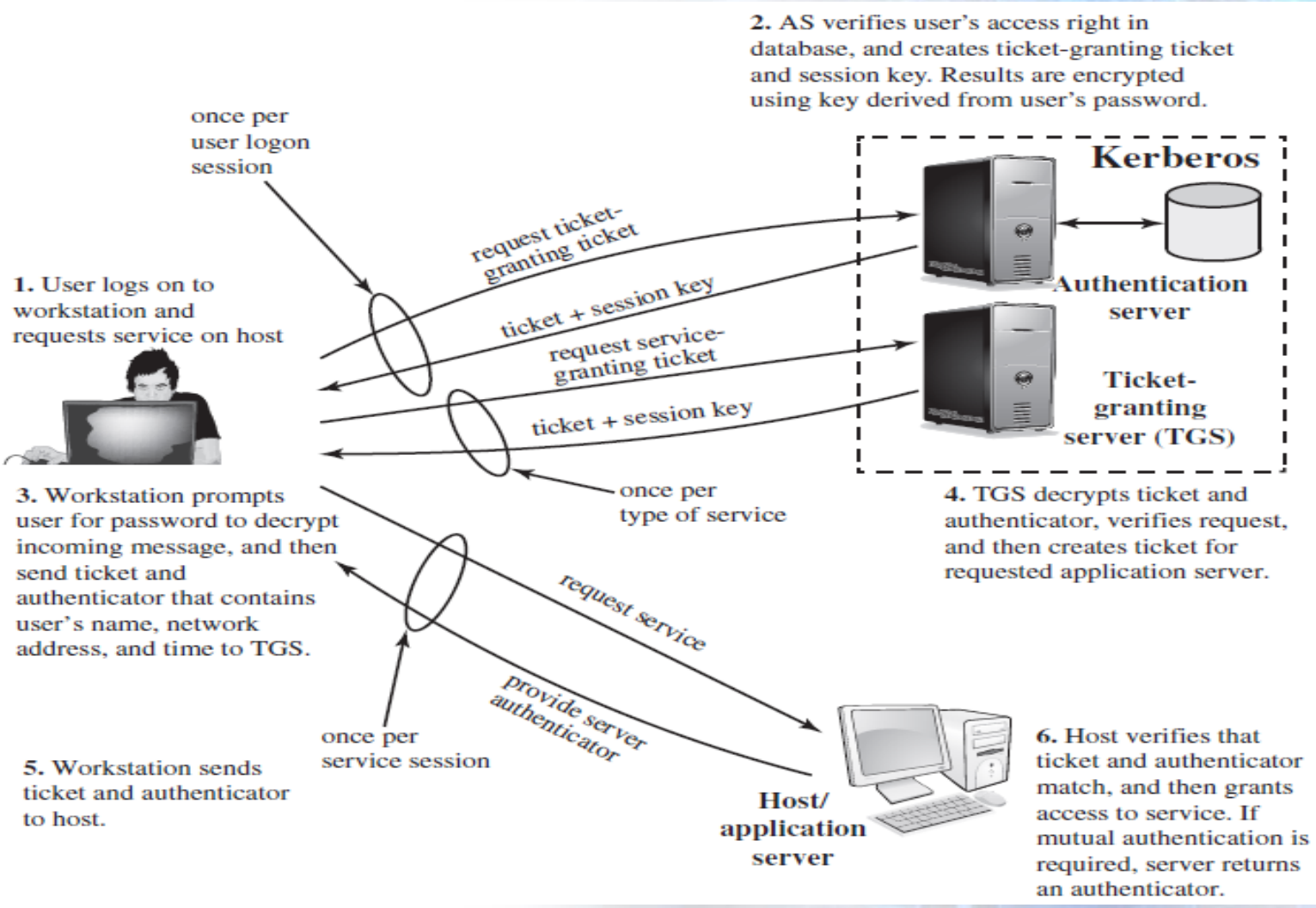
Figures and material in this topic have been adapted from

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

Rationale For Elements Of Kerberos 4

- Kerberos is a key distribution and user authentication service developed at MIT.

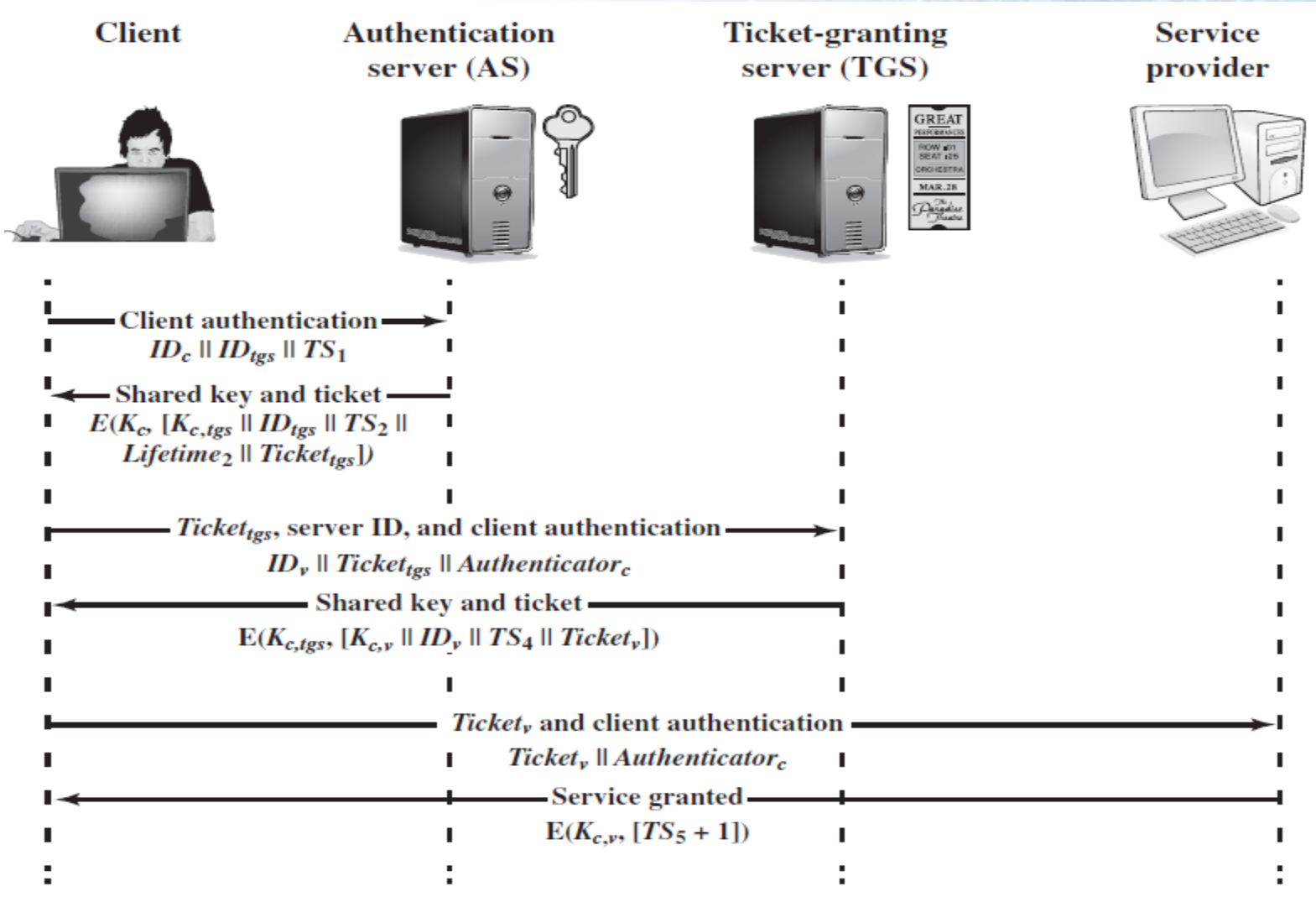
Rationale For Elements Of Kerberos 4



Rationale For Elements Of Kerberos 4

- Broadly, the Kerberos exchanges can be categorized as:
- Authentication Service Exchange
- Ticket-Granting Service Exchange
- Client/Server Authentication Exchange

Rationale For Elements Of Kerberos 4



Rationale For Elements Of Kerberos 4

Authentication Service Exchange

Message (1)	Client requests ticket-granting ticket.
ID_C	Tells AS identity of user from this client.
ID_{tgs}	Tells AS that user requests access to TGS.
TS_1	Allows AS to verify that client's clock is synchronized with that of AS.
Message (2)	AS returns ticket-granting ticket.
K_c	Encryption is based on user's password, enabling AS and client to verify password, and protecting contents of message (2).
$K_{c,tgs}$	Copy of session key accessible to client created by AS to permit secure exchange between client and TGS without requiring them to share a permanent key.
ID_{tgs}	Confirms that this ticket is for the TGS.
TS_2	Informs client of time this ticket was issued.
$Lifetime_2$	Informs client of the lifetime of this ticket.
$Ticket_{tgs}$	Ticket to be used by client to access TGS.

Rationale For Elements Of Kerberos 4

Kerberos Version 4 Message Exchanges

(1) $C \rightarrow AS \quad ID_c \| ID_{tgs} \| TS_1$

(2) $AS \rightarrow C \quad E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$

$$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$$

(a) Authentication Service Exchange to obtain ticket-granting ticket

Rationale For Elements Of Kerberos 4

Ticket-Granting Service Exchange

Message (3)	Client requests service-granting ticket.
ID_V	Tells TGS that user requests access to server V.
$Ticket_{tgs}$	Assures TGS that this user has been authenticated by AS.
$Authenticator_c$	Generated by client to validate ticket.
Message (4)	TGS returns service-granting ticket.
$K_{c,tgs}$	Key shared only by C and TGS protects contents of message (4).
$K_{c,v}$	Copy of session key accessible to client created by TGS to permit secure exchange between client and server without requiring them to share a permanent key.
ID_V	Confirms that this ticket is for server V.
TS_4	Informs client of time this ticket was issued.
$Ticket_V$	Ticket to be used by client to access server V.
$Ticket_{tgs}$	Reusable so that user does not have to reenter password.
K_{tgs}	Ticket is encrypted with key known only to AS and TGS, to prevent tampering.
$K_{c,tgs}$	Copy of session key accessible to TGS used to decrypt authenticator, thereby authenticating ticket.

Rationale For Elements Of Kerberos 4

Ticket-Granting Service Exchange

ID_C	Indicates the rightful owner of this ticket.
AD_C	Prevents use of ticket from workstation other than one that initially requested the ticket.
ID_{tgs}	Assures server that it has decrypted ticket properly.
TS_2	Informs TGS of time this ticket was issued.
$Lifetime_2$	Prevents replay after ticket has expired.
$Authenticator_c$	Assures TGS that the ticket presenter is the same as the client for whom the ticket was issued has very short lifetime to prevent replay.
$K_{c,tgs}$	Authenticator is encrypted with key known only to client and TGS, to prevent tampering.
ID_C	Must match ID in ticket to authenticate ticket.
AD_C	Must match address in ticket to authenticate ticket.
TS_3	Informs TGS of time this authenticator was generated.

Rationale For Elements Of Kerberos 4

Kerberos Version 4 Message Exchanges

(3) C → TGS $ID_v \| Ticket_{tgs} \| Authenticator_c$

(4) TGS → C $E(K_{c,tgs}, [K_{c,v} \| ID_v \| TS_4 \| Ticket_v])$

$$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$$

$$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$$

$$Authenticator_c = E(K_{c,tgs}, [ID_C \| AD_C \| TS_3])$$

(b) Ticket-Granting Service Exchange to obtain service-granting ticket

Rationale For Elements Of Kerberos 4

Client/Server Authentication Exchange

Message (5)	Client requests service.
$Ticket_V$	Assures server that this user has been authenticated by AS.
$Authenticator_c$	Generated by client to validate ticket.
Message (6)	Optional authentication of server to client.
$K_{c,v}$	Assures C that this message is from V.
$TS_5 + 1$	Assures C that this is not a replay of an old reply.
$Ticket_v$	Reusable so that client does not need to request a new ticket from TGS for each access to the same server.
K_v	Ticket is encrypted with key known only to TGS and server, to prevent tampering.
$K_{c,v}$	Copy of session key accessible to client; used to decrypt authenticator, thereby authenticating ticket.
ID_C	Indicates the rightful owner of this ticket.
AD_C	Prevents use of ticket from workstation other than one that initially requested the ticket.

Rationale For Elements Of Kerberos 4

Client/Server Authentication Exchange

ID_V	Assures server that it has decrypted ticket properly.
TS_4	Informs server of time this ticket was issued.
$Lifetime_4$	Prevents replay after ticket has expired.
$Authenticator_c$	Assures server that the ticket presenter is the same as the client for whom the ticket was issued; has very short lifetime to prevent replay.
$K_{c,v}$	Authenticator is encrypted with key known only to client and server, to prevent tampering.
ID_C	Must match ID in ticket to authenticate ticket.
AD_c	Must match address in ticket to authenticate ticket.
TS_5	Informs server of time this authenticator was generated.

Rationale For Elements Of Kerberos 4

Kerberos Version 4 Message Exchanges

(5) C → V $Ticket_v \| Authenticator_c$

(6) V → C $E(K_{c,v}, [TS_5 + 1])$ (for mutual authentication)

$$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$$

$$Authenticator_c = E(K_{c,v}, [ID_C \| AD_C \| TS_5])$$

(c) Client/Server Authentication Exchange to obtain service

Kerberos Realms And Multiple Kerberi

Network Security

Kerberos Realms And Multiple Kerberi

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe Kerberos Realms And Multiple Kerberi.

Kerberos Realms And Multiple Kerberi

Figures and material in this topic have been adapted from

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

Kerberos Realms And Multiple Kerberi

- A full-service Kerberos environment consisting of a Kerberos server, a number of clients, and a number of application servers requires the following:

Kerberos Realms And Multiple Kerberi

- 1. The Kerberos server must have the user ID and hashed passwords of all participating users in its database.
- All users are registered with the Kerberos server.

Kerberos Realms And Multiple Kerberi

- 2. The Kerberos server must share a secret key with each server.
- All servers are registered with the Kerberos server.
- Such an environment is referred to as a **Kerberos realm**.

Kerberos Realms And Multiple Kerberi

- A Kerberos realm is a set of managed nodes that share the same Kerberos database.
- The Kerberos database resides on the Kerberos master computer system, which should be kept in a physically secure room.

Kerberos Realms And Multiple Kerberi

- A read-only copy of the Kerberos database might also reside on other Kerberos computer systems.
- All changes to the database must be made on the master computer system with the Kerberos master password.

Kerberos Realms And Multiple Kerberi

- A Kerberos principal is a service or user that is known to Kerberos system.
- Each Kerberos principal is identified by its name, which consist of three parts: a service/user name, an instance name, and a realm name.

Kerberos Realms And Multiple Kerberi

- Networks of clients and servers under different administrative organizations typically constitute different realms.

Kerberos Realms And Multiple Kerberi

- Users in one realm may need access to servers in other realms, and some servers may be willing to provide service to users from other realms, provided that those users are authenticated.

Kerberos Realms And Multiple Kerberi

- For two realms to support interrealm authentication, a third requirement is added:
- **3.** The Kerberos server in each interoperating realm shares a secret key with the server in other realm. The two servers are registered with each other.

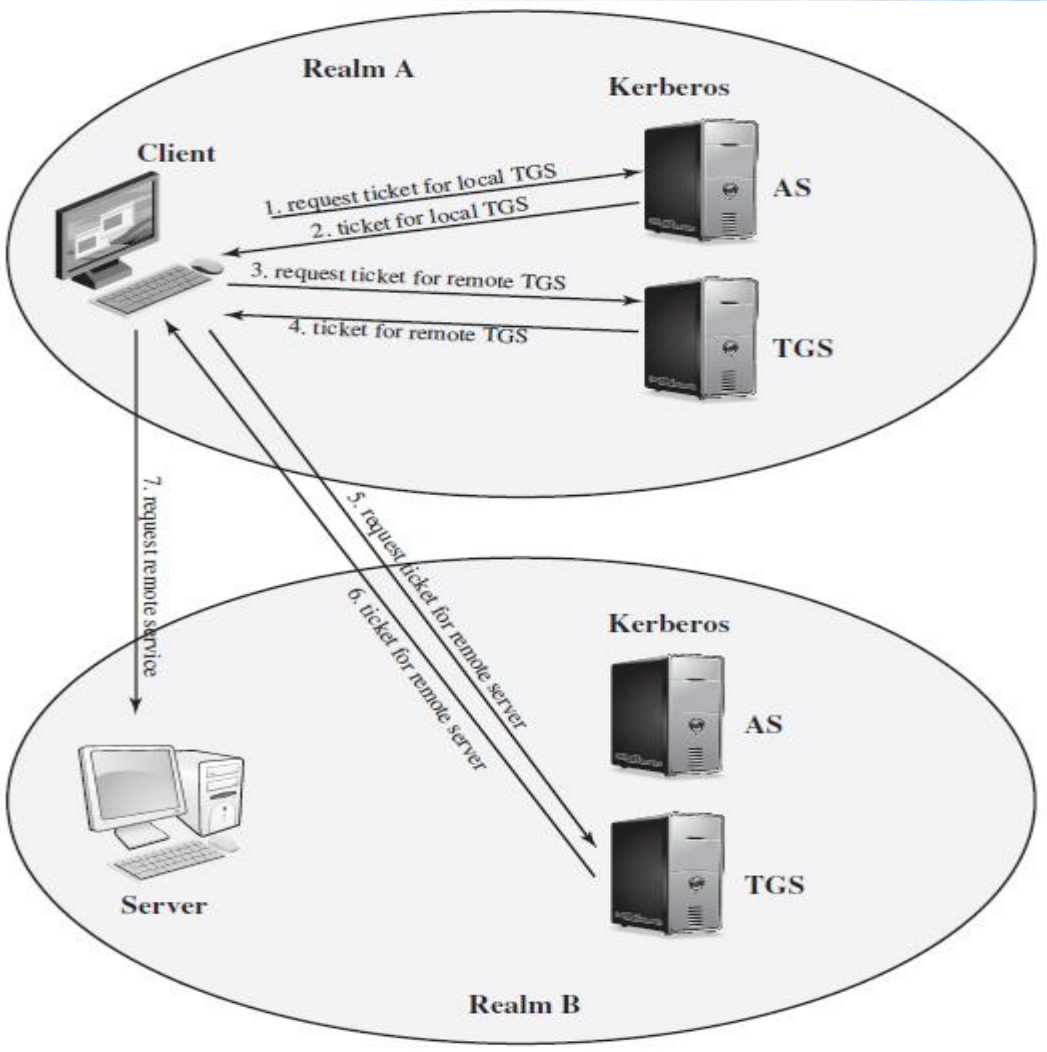
Kerberos Realms And Multiple Kerberi

- A user wishing service on a server in another realm needs a ticket for that server.
- The user's client follows the usual procedures to gain access to the local TGS and then requests a ticket-granting ticket for a remote TGS.

Kerberos Realms And Multiple Kerberi

- The client can then apply to the remote TGS for a service-granting ticket for the desired server in the realm of the remote TGS.

Kerberos Realms And Multiple Kerberi



Kerberos Realms And Multiple Kerberi

(1) $C \rightarrow AS$: $ID_c \| ID_{tgs} \| TS_1$

(2) $AS \rightarrow C$: $E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$

(3) $C \rightarrow TGS$: $ID_{tgsrem} \| Ticket_{tgs} \| Authenticator_c$

(4) $TGS \rightarrow C$: $E(K_{c,tgs}, [K_{c,tgsrem} \| ID_{tgsrem} \| TS_4 \| Ticket_{tgsrem}])$

(5) $C \rightarrow TGS_{rem}$: $ID_{vrem} \| Ticket_{tgsrem} \| Authenticator_c$

(6) $TGS_{rem} \rightarrow C$: $E(K_{c,tgsrem}, [K_{c,vrem} \| ID_{vrem} \| TS_6 \| Ticket_{vrem}])$

(7) $C \rightarrow V_{rem}$: $Ticket_{vrem} \| Authenticator_c$

Kerberos Realms And Multiple Kerberi

- The ticket presented to the remote server (Vrem) indicates the realm in which the user was originally authenticated.
- The server chooses whether to honor the remote request.

Kerberos Realms And Multiple Kerberi

End

- This foregoing approach does not scale well to many realms.
- If there are N realms, N(N - 1)/2 secure key exchanges so that each Kerberos realm can interoperate with all other Kerberos realms.

Differences Between Kerberos 4 And 5

Network Security

Differences Between Kerberos 4 And 5

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain differences between versions 4 and 5 of Kerberos.

Differences Between Kerberos 4 And 5

Figures and material in this topic have been adapted from

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

Differences Between Kerberos 4 And 5

- Version 5 is intended to address the limitations of version 4 in two areas:
- environmental shortcomings and
- technical deficiencies.

Differences Between Kerberos 4 And 5

Environmental Shortcomings:

- Kerberos version 4 was developed for use within the Project Athena environment and, accordingly, did not fully address the need to be of general purpose.

Differences Between Kerberos 4 And 5

- **1. Encryption system dependence:**
- Version 4 requires the use of DES.
- Export restriction on DES as well as doubts about the strength of DES were thus of concern.

Differences Between Kerberos 4 And 5

- In version 5, ciphertext is tagged with an encryption-type identifier so that any encryption technique may be used.

Differences Between Kerberos 4 And 5

- Encryption keys are tagged with a type and a length, allowing the same key to be used in different algorithms and allowing the specification of different variations on a given algorithm.

Differences Between Kerberos 4 And 5

- **2. Internet protocol dependence:**
- Version 4 requires the use of Internet Protocol (IP) addresses.
- Other address types, such as the ISO network address, are not accommodated.

Differences Between Kerberos 4 And 5

- Version 5 network addresses are tagged with type and length, allowing any network address type to be used.

Differences Between Kerberos 4 And 5

- **3. Message byte ordering:**
- In version 4, sender of a message employs a byte ordering of its own choosing and tags the message to indicate LSB in lowest address or MSB in lowest address.

Differences Between Kerberos 4 And 5

- This techniques works but does not follow established conventions.

Differences Between Kerberos 4 And 5

- In version 5, all message structures are defined using Abstract Syntax Notation One (ASN.1) and Basic Encoding Rules (BER), which provide an unambiguous byte ordering.

Differences Between Kerberos 4 And 5

- **4. Ticket lifetime:**
- Lifetime values in version 4 are encoded in an 8-bit quantity in units of five minutes.
- The maximum lifetime is $2^8 * 5 = 1280$ mins.
- This may be inadequate for a long-running simulation.

Differences Between Kerberos 4 And 5

- In version 5, tickets include an explicit start time and end time, allowing tickets with arbitrary lifetimes.

Differences Between Kerberos 4 And 5

- **5. Authentication forwarding:**
- Version 4 does not allow credentials issued to one client to be forwarded to some other host and used by some other client.

Differences Between Kerberos 4 And 5

- For example, a client issues a request to a print server that then accesses the client's file from a file server, using the client's credentials for access.
- Version 5 provides this capability.

Differences Between Kerberos 4 And 5

- **6. Interrealm authentication:**
- In version 4, interoperability among N realms requires N^2 Kerberos-to-Kerberos relationships.
- Version 5 requires fewer relationships.

Differences Between Kerberos 4 And 5

Technical Deficiencies:

- **1. Double encryption:** Kerberos Version 4 messages (2) and (4) are encrypted twice—The second encryption is not necessary and is computationally wasteful.

Differences Between Kerberos 4 And 5

- **2. PCBC encryption:**
- Encryption in version 4 makes use of a nonstandard mode of DES known as propagating cipher block chaining (PCBC).

Differences Between Kerberos 4 And 5

- This mode is vulnerable to an attack involving the interchange of ciphertext blocks.
- Version 5 provides explicit integrity mechanisms using the standard CBC mode for encryption.

Differences Between Kerberos 4 And 5

- **3. Session keys:**
- Each ticket includes a session key to be used by the client.
- Same ticket may be used repeatedly to gain service, an opponent may replay messages from an old session to client or the server.

Differences Between Kerberos 4 And 5

- In version 5, it is possible for a client and server to negotiate a subsession key, which is to be used only for that one connection.

Differences Between Kerberos 4 And 5

End

- **4. Password attacks:**
- Both versions are vulnerable to a password attack.
- Version 5 employs a preauthentication mechanism which makes password attacks more difficult, but it does not prevent them.

The Version 5 Authentication Dialogue

Network Security

The Version 5 Authentication Dialogue

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe authentication dialogue of Kerberos V5.

The Version 5 Authentication Dialogue

Figures and material in this topic have been adapted from

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

The Version 5 Authentication Dialogue

- Lets first, consider the **authentication service exchange**.
- The first message is a client request for a ticket-granting ticket.
- It includes the ID of the user and the TGS.
- In Version 5, following new elements are added:

The Version 5 Authentication Dialogue

- **Realm:** Indicates realm of user
- **Options:** Used to request that certain flags be set in the returned ticket

The Version 5 Authentication Dialogue

- **Times:** Used by the client to request the settings in the ticket:
- —from: the desired start time for the requested ticket
- —till: the requested expiration time for the requested ticket
- —rtime: requested renew-till time

The Version 5 Authentication Dialogue

- **Nonce:** A random value to be repeated in message (2) to assure that the response is fresh and has not been replayed by an opponent.

The Version 5 Authentication Dialogue

- Message (2) returns a ticket-granting ticket, identifying information for the client, and a block encrypted using the encryption key based on the user's password.

The Version 5 Authentication Dialogue

- This block includes the session key to be used between the client and the TGS, times specified in message (1), the nonce from message (1), and TGS identifying information.

The Version 5 Authentication Dialogue

- An important component in the ticket is the flags that reflect the status of this ticket and the requested options.
- These flags introduce significant new functionality in version 5.

The Version 5 Authentication Dialogue

Authentication Service Exchange to obtain ticket-granting ticket

(1) $C \rightarrow AS$ $Options \| ID_c \| Realm_c \| ID_{tgs} \| Times \| Nonce_1$

(2) $AS \rightarrow C$ $Realm_C \| ID_C \| Ticket_{tgs} \| E(K_c, [K_{c,tgs} \| Times \| Nonce_1 \| Realm_{tgs} \| ID_{tgs}])$

$Ticket_{tgs} = E(K_{tgs}, [Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times])$

The Version 5 Authentication Dialogue

- Let us now look at the **ticket-granting service exchange** for version 5.
- Message (3) includes an authenticator, a ticket, and the name of the requested service.

The Version 5 Authentication Dialogue

- For version 5, it also includes requested times and options for the ticket and a nonce—all with functions similar to those of message (1).

The Version 5 Authentication Dialogue

- Message (4) has the same structure as message (2).
- It returns a ticket plus information needed by the client, with the information encrypted using the session key now shared by the client and the TGS.

The Version 5 Authentication Dialogue

Ticket-Granting Service Exchange to obtain service-granting ticket

(3) $C \rightarrow TGS$ $Options \| ID_v \| Times \| Nonce_2 \| Ticket_{tgs} \| Authenticator_c$

(4) $TGS \rightarrow C$ $Realm_c \| ID_C \| Ticket_v \| E(K_{c,tgs}, [K_{c,v} \| Times \| Nonce_2 \| Realm_v \| ID_v])$

$$Ticket_{tgs} = E(K_{tgs}, [Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times])$$

$$Ticket_v = E(K_v, [Flags \| K_{c,v} \| Realm_c \| ID_C \| AD_C \| Times])$$

$$Authenticator_c = E(K_{c,tgs}, [ID_C \| Realm_c \| TS_1])$$

The Version 5 Authentication Dialogue

- Finally, for the **client/server authentication exchange**, several new features appear in version 5.
- In message (5), the client may request as an option that mutual authentication is required.

The Version 5 Authentication Dialogue

- The authenticator includes several new fields:
- **Subkey:** The client's choice for an encryption key to be used to protect this specific application session.

The Version 5 Authentication Dialogue

- **Sequence number:** An optional field.
- Messages may be sequence numbered to detect replays.
- If mutual authentication is required, the server responds with message (6).

The Version 5 Authentication Dialogue

- This message includes the timestamp from the authenticator.
- The subkey field, if present, overrides the subkey field, if present, in message (5).

The Version 5 Authentication Dialogue

Client/Server Authentication Exchange to obtain service

(5) $\mathbf{C} \rightarrow \mathbf{V}$ $Options \| Ticket_v \| Authenticator_c$

(6) $\mathbf{V} \rightarrow \mathbf{C}$ $E_{K_{c,v}}[TS_2 \| Subkey \| Seq\ \#]$

$$Ticket_v = E(K_v, [Flag \| K_{c,v} \| Realm_c \| ID_C \| AD_C \| Times])$$

$$Authenticator_c = E(K_{c,v}, [ID_C \| Relam_c \| TS_2 \| Subkey \| Seq\ \#])$$

Public-Key Certificates

Network Security

Public-Key Certificates

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe Public-Key Certificates.

Public-Key Certificates

Figures and material in this topic have been adapted from

- William Stallings, *"Network Security Essentials"*, 2014.
- William Stallings, *"Cryptography and Network Security Principles and Practice"*, 2014

Public-Key Certificates

- On the face of it, the point of public-key encryption is that the public key is public.
- Any participant can send his public key to any other participant or broadcast it to the community at large.

Public-Key Certificates

- Several techniques proposed for the distribution of public keys can be grouped as:
- Public announcement
- Publicly available directory
- Public-key authority
- Public-key certificates

Public-Key Certificates

- **1. Public Announcement:**
- Users append their public key to messages they send to public forums such as USENET newsgroups and Internet mailing lists.

Public-Key Certificates

Uncontrolled Public-Key Distribution



Public-Key Certificates

- **Major weakness:**
- Anyone can forge such a public announcement.
- Some user could pretend to be user A and send a public key to another participant or broadcast such a public key.

Public-Key Certificates

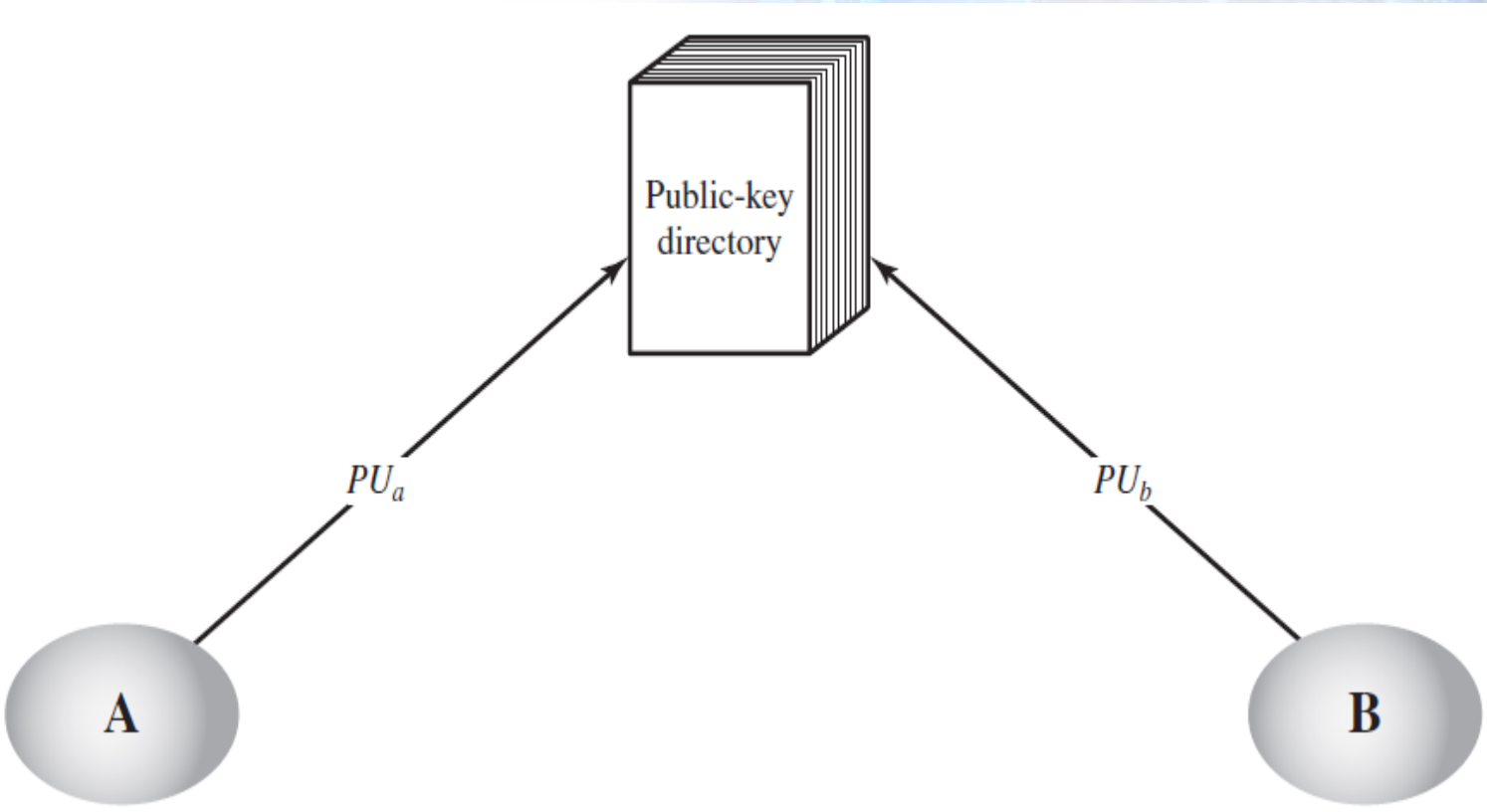
- **Publicly Available Directory**
- A greater degree of security can be achieved by maintaining a publicly available dynamic directory of public keys.
- Responsibility of some trusted organization.

Public-Key Certificates

- For each participant, a {name, public key}.
- Each participant registers a public key by secure authenticated communication.
- A participant may replace the existing key with a new one at anytime.

Public-Key Certificates

Public-Key Publication



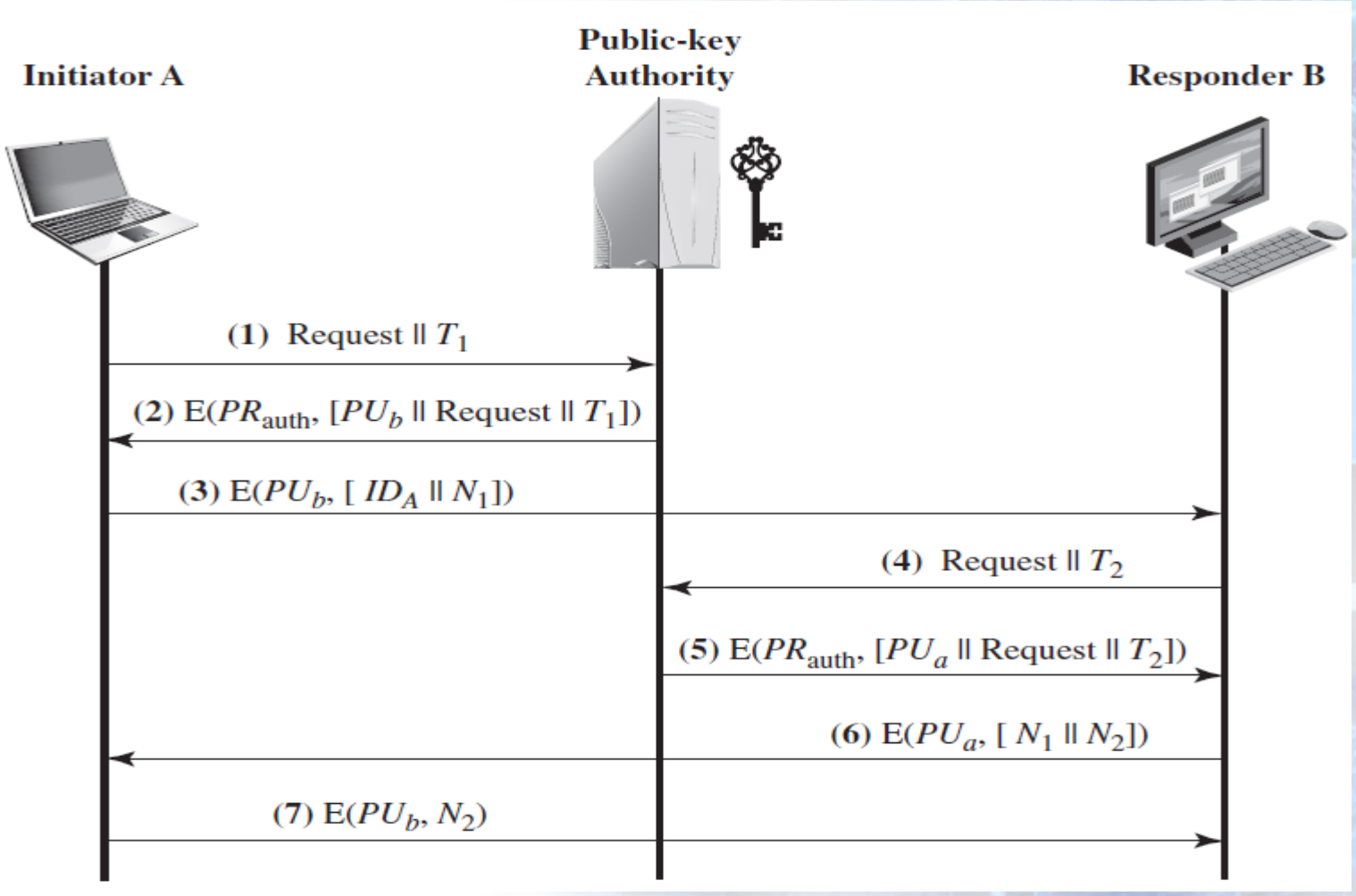
Public-Key Certificates

- If an adversary succeeds in obtaining the private key of the directory authority, he can authoritatively pass out counterfeit public keys and then impersonate a participant and eavesdrop on messages sent to him.

Public-Key Certificates

- **Public-Key Authority**
- Each participant reliably knows a public key for the authority, with only the authority knowing the corresponding private key.

Public-Key Certificates



Public-Key Certificates

- **Public-Key Certificates:**
- In previous case, a user has to appeal to authority for a public key for every other user that it wishes to contact.
- This is somewhat of a bottleneck in the system.

Public-Key Certificates

- Also, the directory of names and public keys maintained by the authority is vulnerable to tampering.
- An alternative approach is to use certificates.

Public-Key Certificates

- Certificates can be used by participants to exchange keys without contacting a public-key authority, in a way that is as reliable as if the keys were obtained directly from a public-key authority.

Public-Key Certificates

- A certificate consists of a public key, an identifier of the key owner, and the whole block signed by a trusted third party.
- TTP is a certificate authority (CA), such as a government agency or a financial institution.

Public-Key Certificates

- A user can present his public key to CA in a secure manner and obtain a certificate.
- The user can then publish the certificate.
- Anyone needing this user's public key can obtain certificate and verify it's validity – attached signature.

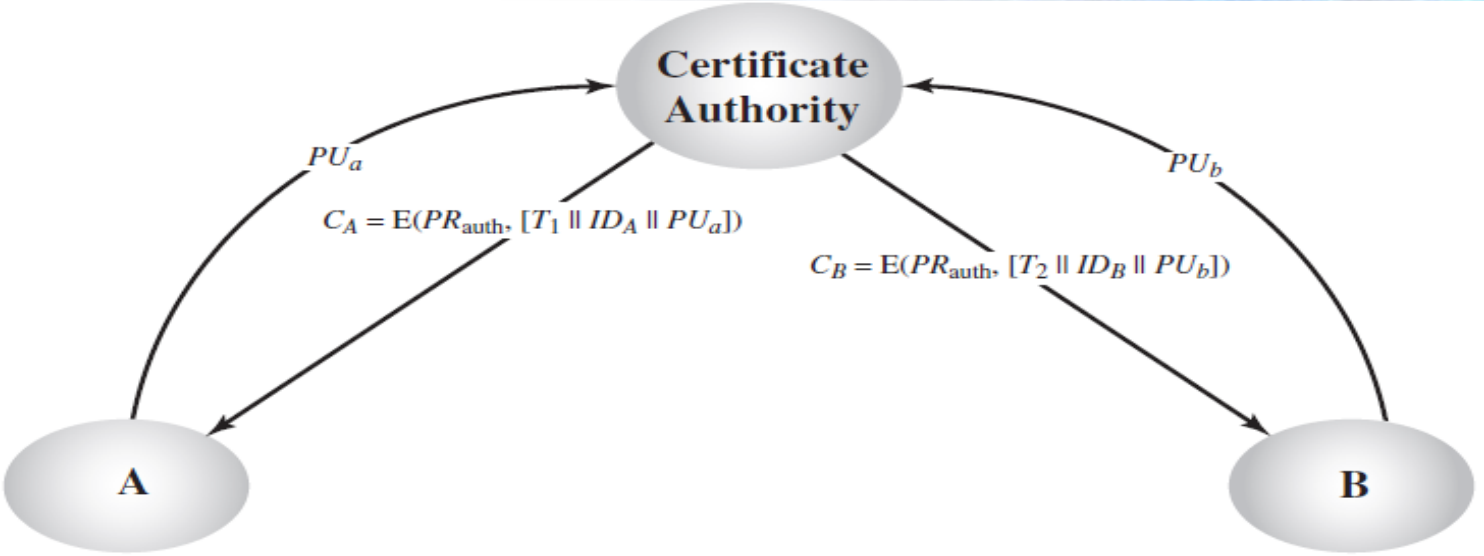
Public-Key Certificates

- **Requirements:**
- 1. Any participant can read a certificate to determine the name and public key of the certificate's owner.
- 2. Any participant can verify that certificate originated from the certificate authority and is not counterfeit.

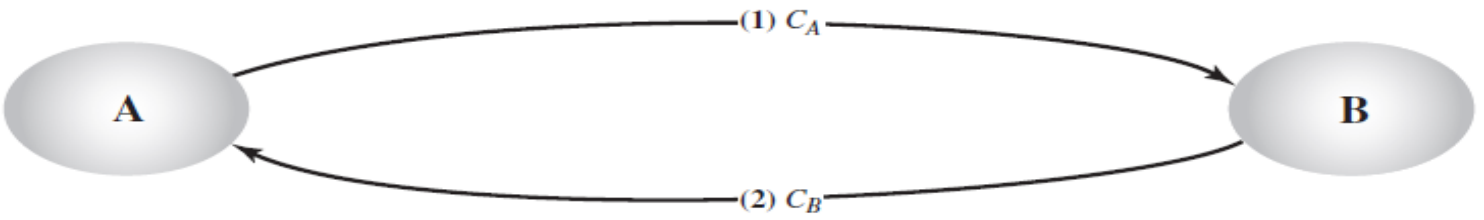
Public-Key Certificates

- 3. Only the certificate authority can create and update certificates.

Public-Key Certificates



(a) Obtaining certificates from CA



(b) Exchanging certificates

Public-Key Certificates

- One scheme has become universally accepted for public-key certificates: the X.509 standard.

End

Public-Key Distribution of Secret Keys

Network Security

Public-Key Distribution of Secret Keys

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe Public-Key Distribution of Secret Keys.

Public-Key Distribution of Secret Keys

Figures and material in this topic have been adapted from

- William Stallings, *"Network Security Essentials"*, 2014.
- William Stallings, *"Cryptography and Network Security Principles and Practice"*, 2014

Public-Key Distribution of Secret Keys

- Because of the inefficiency of public-key cryptosystems, they are almost never used for the direct encryption of sizable block of data, but are limited to relatively small blocks.

Public-Key Distribution of Secret Keys

- One of the most important uses of a public-key cryptosystem is to encrypt secret keys for distribution.

Public-Key Distribution of Secret Keys

Simple Secret Key Distribution:

- If A wishes to communicate with B, the following procedure is employed:

Public-Key Distribution of Secret Keys

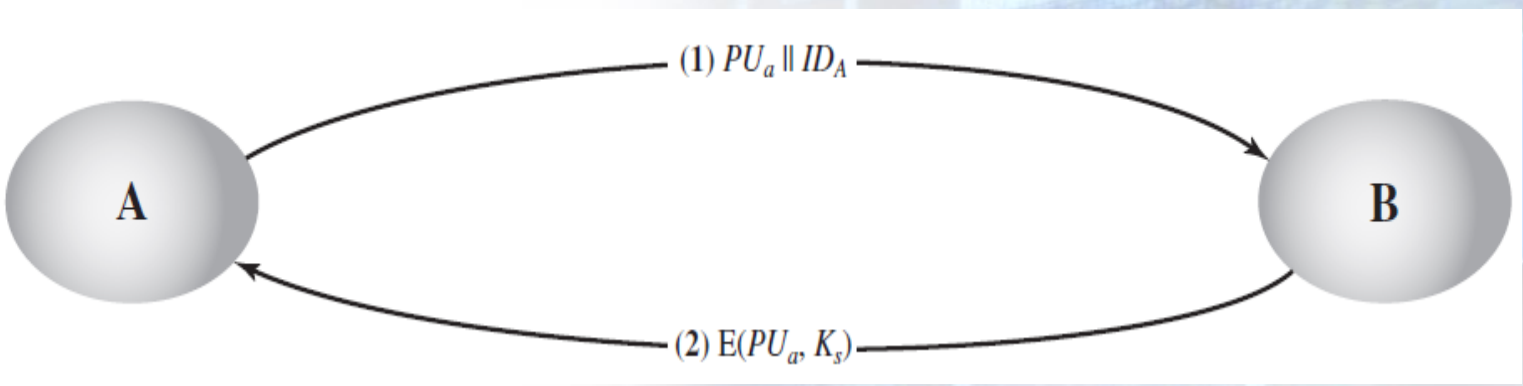
- 1. A generates a {PU_a, PR_a} key pair and transmits a message to B consisting of PU_a and an identifier of A, ID_A.
- 2. B generates a secret session key, K_s, and transmits it to A, which is encrypted with A's public key.

Public-Key Distribution of Secret Keys

- 3. A computes $D(PR_a, E(PU_a, K_s))$ to recover the secret key.
- Because only A can decrypt the message, only A and B will know K_s.
- 4. A discards PU_a and PR_a and B discards PU_a.

Public-Key Distribution of Secret Keys

Simple Use of Public-Key Encryption to Establish a Session Key



Public-Key Distribution of Secret Keys

- A and B can now securely communicate using conventional encryption and the session key K_s.
- At the completion of the exchange, both A and B discard K_s.

Public-Key Distribution of Secret Keys

- No keys exist before the start of the communication and none exist after the completion of communication.
- Risk of compromise of the keys is minimal.
- Communication is also secure from eavesdropping.

Public-Key Distribution of Secret Keys

- However, if an adversary, D, has control of the intervening communication channel, then D can compromise the communication in the following fashion without being detected:

Public-Key Distribution of Secret Keys

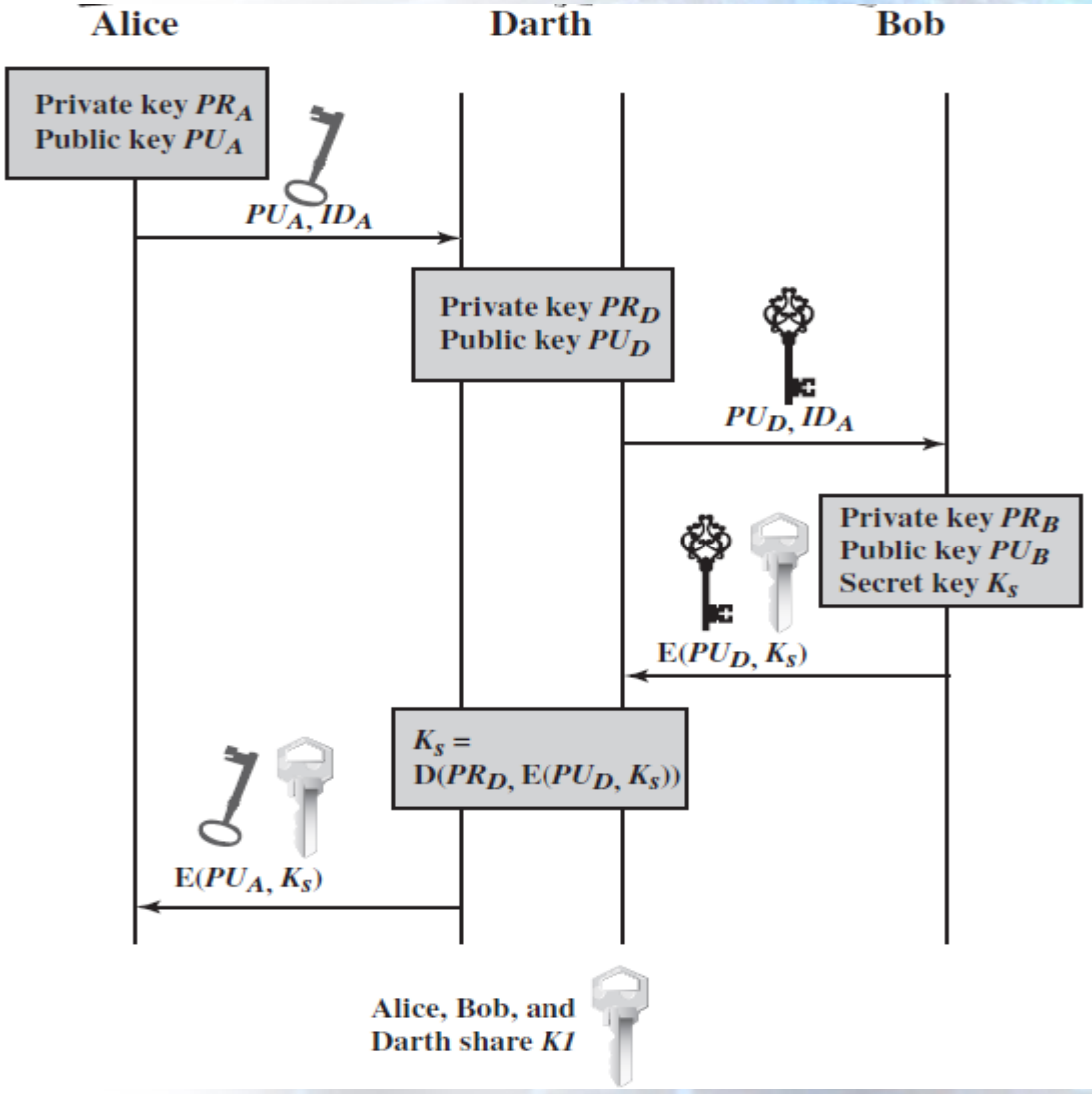
- 1. A generates a {PU_a, PR_a} key pair and transmits a message intended for B consisting of PU_a and an identifier of A, ID_A.
- 2. D intercepts the message, creates its own {PU_d, PR_d} key pair and transmits PU_d || ID_A to B.

Public-Key Distribution of Secret Keys

- 3. B generates a secret key, K_s, and transmits $E(PU_d, K_s)$.
- 4. D intercepts the message and learns K_s by computing $D(PR_d, E(PU_d, K_s))$.
- 5. D transmits $E(PU_a, K_s)$ to A.

Public-Key Distribution of Secret Keys

Another Man-in-the-Middle Attack



Public-Key Distribution of Secret Keys

- The result is that both A and B know K_s and are unaware that K_s has also been revealed to D.
- A and B can now exchange messages using K_s.
- D simply eavesdrops.

Public-Key Distribution of Secret Keys

Public-Key Distribution of Secret Keys:

- We need to protect distribution of the secret key against both active and passive attacks.
- Assume A and B have exchanged public keys by public certificates.

Public-Key Distribution of Secret Keys

- **1.** A uses B's public key to encrypt a message to B containing an identifier of A (ID_A) and a nonce ($N1$), which is used to identify this transaction uniquely.

Public-Key Distribution of Secret Keys

- **2.** B sends a message to A encrypted with *PUa* and containing A's *N*1 and a new nonce (*N*2) generated by B.
- As only B could have decrypted message (1), presence of *N*1 in message (2) assures A that sender is B.

Public-Key Distribution of Secret Keys

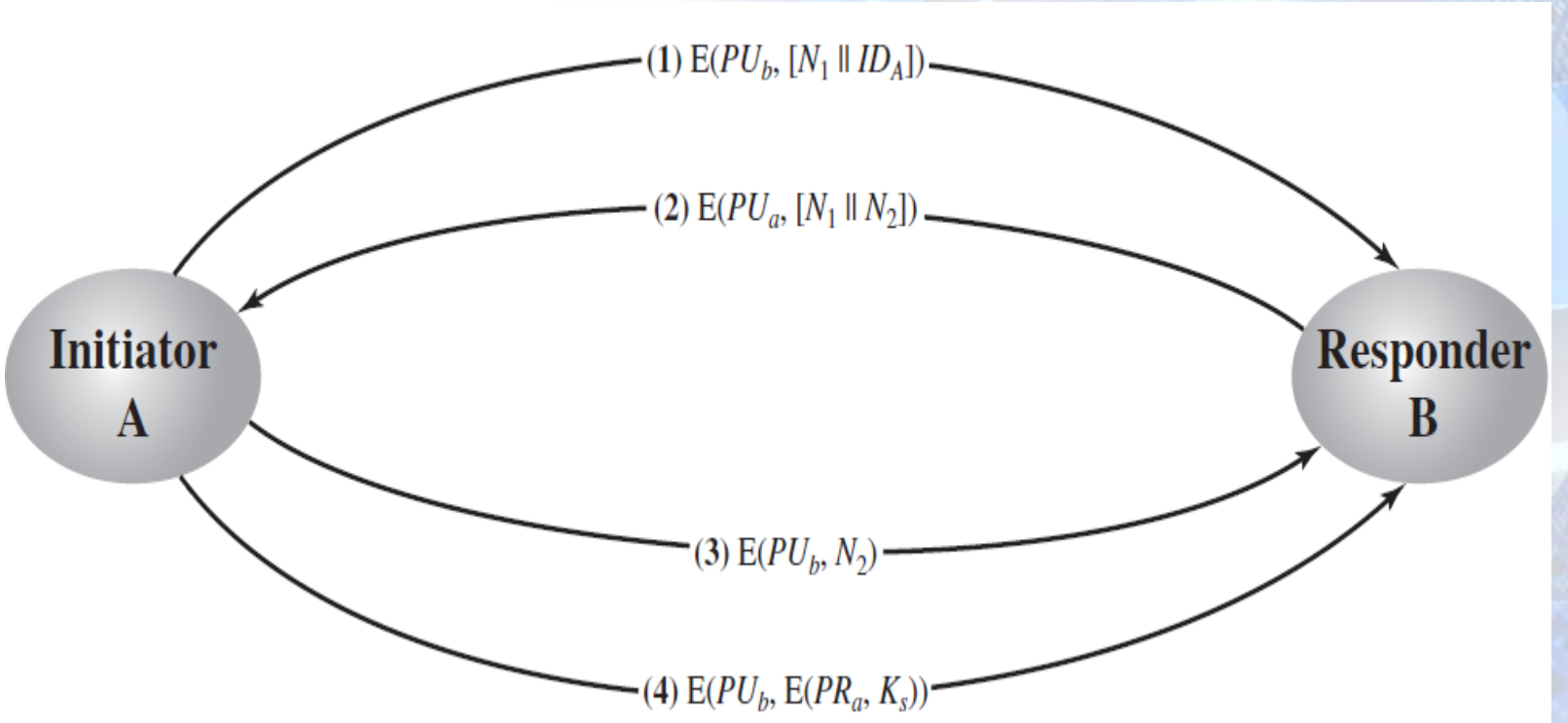
- **3.** A returns *N*2, encrypted using B's public key, to assure B that its correspondent is A.
- **4.** A selects a secret key *Ks* and sends *M* = E(*PUb*, E(*PRa*, *Ks*)) to B.

Public-Key Distribution of Secret Keys

- Encryption of this message with B's public key ensures that only B can read it; encryption with A's private key ensures that only A could have sent it.
- **5.** B computes D(*PUa*, D(*PRb*, *M*)) to recover the secret key.

Public-Key Distribution of Secret Keys

Public-Key Distribution of Secret Keys



Certificates of X.509 Certificates

Network Security

Certificates of X.509 Certificates

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain basics of X.509 Certificates.

Certificates of X.509 Certificates

Figures and material in this topic have been adapted from

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

Certificates of X.509 Certificates

- ITU-T recommendation X.509 is part of the X.500 series of recommendations that define a directory service, which is a server or distributed set of servers that maintains a database of information about users.

Certificates of X.509 Certificates

- X.509 defines a framework for the provision of authentication services by the X.500 directory to its users.
- The directory may serve as a repository of public-key certificates.

Certificates of X.509 Certificates

- Also, X.509 defines alternative authentication protocols based on the use of public-key certificates.
- X.509 certificate format is used in S/MIME, IP Security, and SSL/TLS .

Certificates of X.509 Certificates

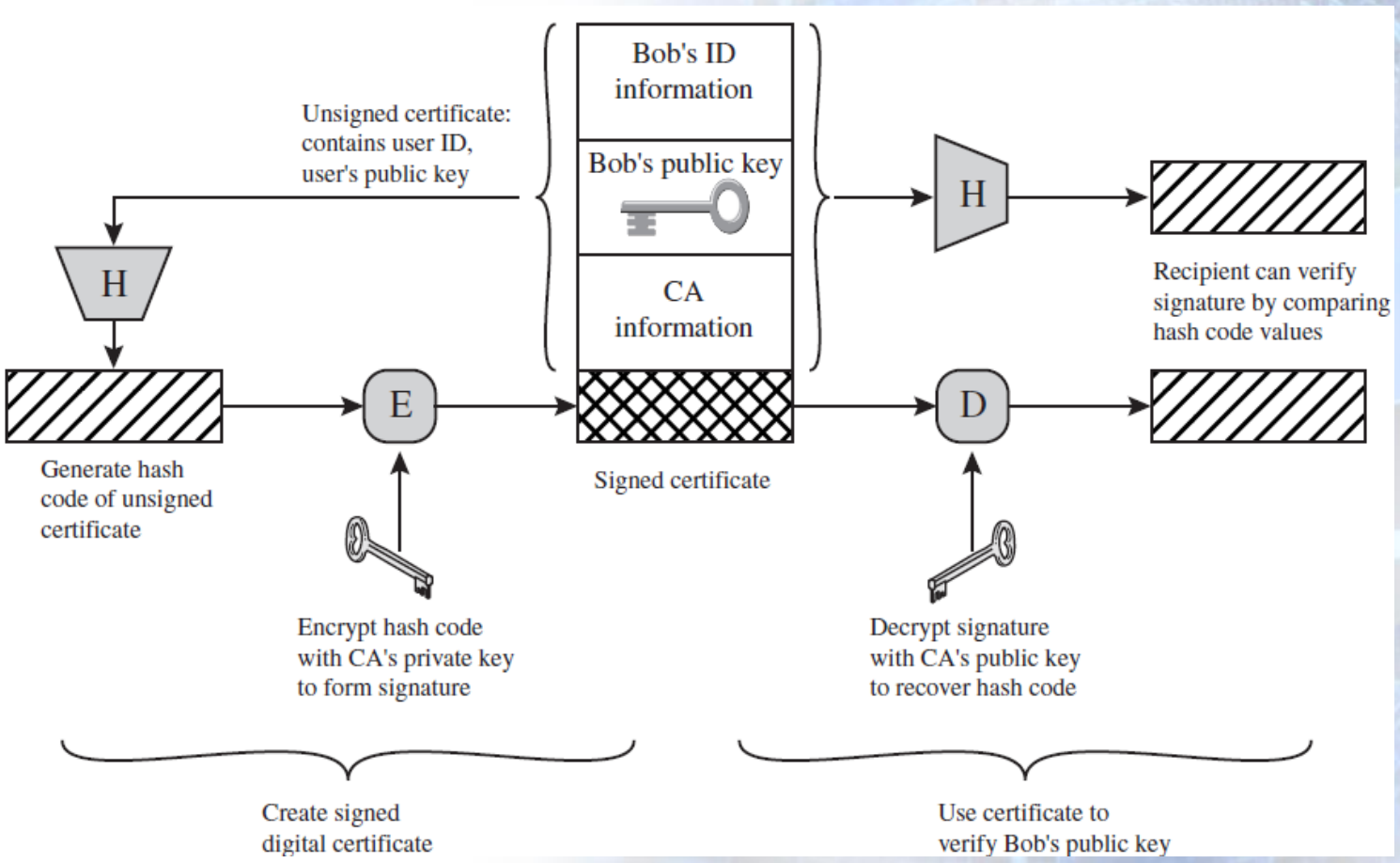
- X.509 was initially issued in 1988.
- It was subsequently revised in 1993 to address some of the security concerns.
- A third version was issued in 1995 and revised in 2000.

Certificates of X.509 Certificates

- X.509 is based on the use of public-key cryptography and digital signatures.
- The standard does not dictate the use of a specific algorithm but recommends RSA.

Certificates of X.509 Certificates

Public-Key Certificate Use



Certificates of X.509 Certificates

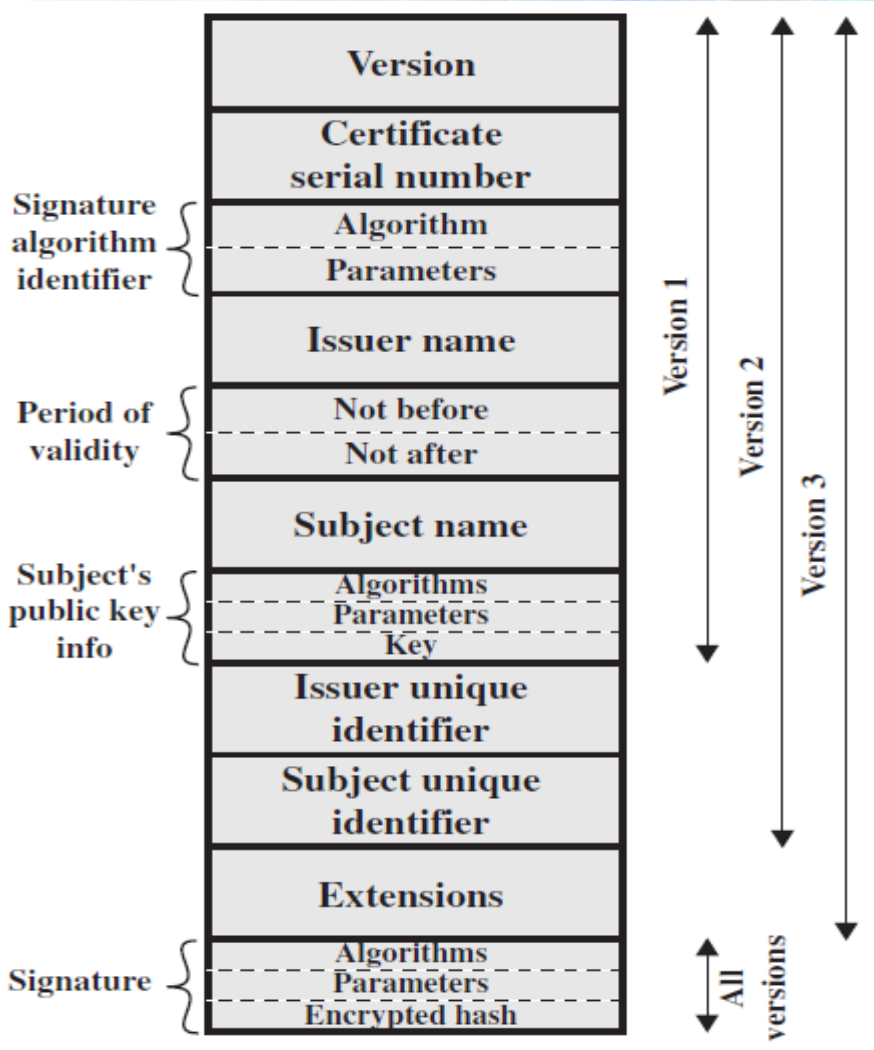
- The heart of X.509 scheme is public-key certificate associated with each user.
- These user certificates are created by some trusted certification authority (CA) and placed in the directory by the CA or by the user.

Certificates of X.509 Certificates

- The directory server only provides an easily accessible location for users to obtain certificates.

Certificates of X.509 Certificates

X.509 Format



Certificates of X.509 Certificates

- **Version:** Differentiates among versions of the certificate format.
- Default version is 1.
- If the issuer unique identifier or subject unique identifier are present, the value must be version 2.

Certificates of X.509 Certificates

- If one or more extensions are present, the version must be version 3.
- **Serial number:** An integer value unique within the issuing CA that is unambiguously associated with this certificate.

Certificates of X.509 Certificates

- **Signature algorithm identifier:** The algorithm used to sign the certificate together with any associated parameters.
- This information is also repeated in the signature field at the end of the certificate.

Certificates of X.509 Certificates

- **Issuer name:** X.500 name of the CA that created and signed this certificate.
- **Period of validity:** Consists of two dates: the first and last on which the certificate is valid.

Certificates of X.509 Certificates

- **Subject name:** The name of the user to whom this certificate refers.
- This certificate certifies the public key of the subject who holds corresponding private key.

Certificates of X.509 Certificates

- **Subject's public-key information:** The public key of the subject, plus an identifier of the algorithm for which this key is to be used, together with any associated parameters.

Certificates of X.509 Certificates

- **Issuer unique identifier:** An optional-bit string field used to identify uniquely the issuing CA in the event the X.500 name has been reused for different entities.

Certificates of X.509 Certificates

- **Subject unique identifier:** An optional-bit string field used to identify uniquely the subject in the event the X.500 name has been reused for different entities.

Certificates of X.509 Certificates

- **Extensions:** A set of one or more extension fields. Extensions were added in version 3.
- **Signature:** contains hash code of other fields encrypted with CA's private key. It includes signature algorithm ID.

Certificates of X.509 Certificates

Notation to define a certificate

$$CA \ll A \gg = CA \{V, SN, AI, CA, UCA, A, UA, Ap, T^A\}$$

where

$Y \ll X \gg$ = the certificate of user X issued by certification authority Y

$Y \{I\}$ = the signing of I by Y. It consists of I with an encrypted hash code appended

V = version of the certificate

SN = serial number of the certificate

AI = identifier of the algorithm used to sign the certificate

CA = name of certificate authority

UCA = optional unique identifier of the CA

A = name of user A

UA = optional unique identifier of the user A

Ap = public key of user A

T^A = period of validity of the certificate

Certificates of X.509 Certificates

End

- The CA signs the certificate with its private key.
- If the corresponding public key is known to a user, then that user can verify that a certificate signed by the CA is valid.

Obtaining And Revoking A Certificate

Network Security

Obtaining And Revoking A Certificate

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe how to obtain and revoke a certificate.

Obtaining And Revoking A Certificate

Figures and material in this topic have been adapted from

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

Obtaining And Revoking A Certificate

- User certificates generated by a CA have the following characteristics:
- 1. Any user with access to the public key of the CA can verify the user public key that was certified.

Obtaining And Revoking A Certificate

- 2. No party other than the certification authority can modify the certificate without this being detected.

Obtaining And Revoking A Certificate

- If there is a large community of users, it is more practical to have a number of CAs, each of which securely provides its public key to some fraction of the users.

Obtaining And Revoking A Certificate

- Now suppose A has obtained a certificate from CA X1 and B has obtained a certificate from CA X2.
- If A does not securely know the public key of X2, then B's certificate, issued by X2, is useless to A.

Obtaining And Revoking A Certificate

- A can read B's certificate, but A cannot verify the signature.
- However, if the two CAs have securely exchanged their own public keys, the following procedure will enable A to obtain B's public key.

Obtaining And Revoking A Certificate

- **Step 1:** A obtains from directory the certificate of X2 signed by X1.
- As A securely knows X1's public key, A can obtain X2's public key from its certificate and verify it by means of X1's signature on certificate.

Obtaining And Revoking A Certificate

- **Step 2:** A then goes back to the directory and obtains the certificate of B signed by X2.
- Because A now has a trusted copy of X2's public key, A can verify the signature and securely obtain B's public key.

Obtaining And Revoking A Certificate

- A has used a chain of certificates to obtain B's public key:
- X1 <<X2>> X2 << B>>
- B can obtain A's public key with reverse chain
- X2 <<X1>> X1<< A >>
- A chain with *N* elements is:
- X1 <<X2>> X2<< X3 >> … X*N* << B >>

Obtaining And Revoking A Certificate

- In this case, each pair of CAs in the chain (Xi, Xi+1) must have created certificates for each other.

Obtaining And Revoking A Certificate

- All these certificates of CAs by CAs need to appear in the directory, and the user needs to know how they are linked to follow a path to another user's public key certificate.

Obtaining And Revoking A Certificate

- X.509 suggests that CAs be arranged in a hierarchy so that navigation is straightforward.
- Lets indicate the hierarchical relationship among the CAs with connected circles.

Obtaining And Revoking A Certificate

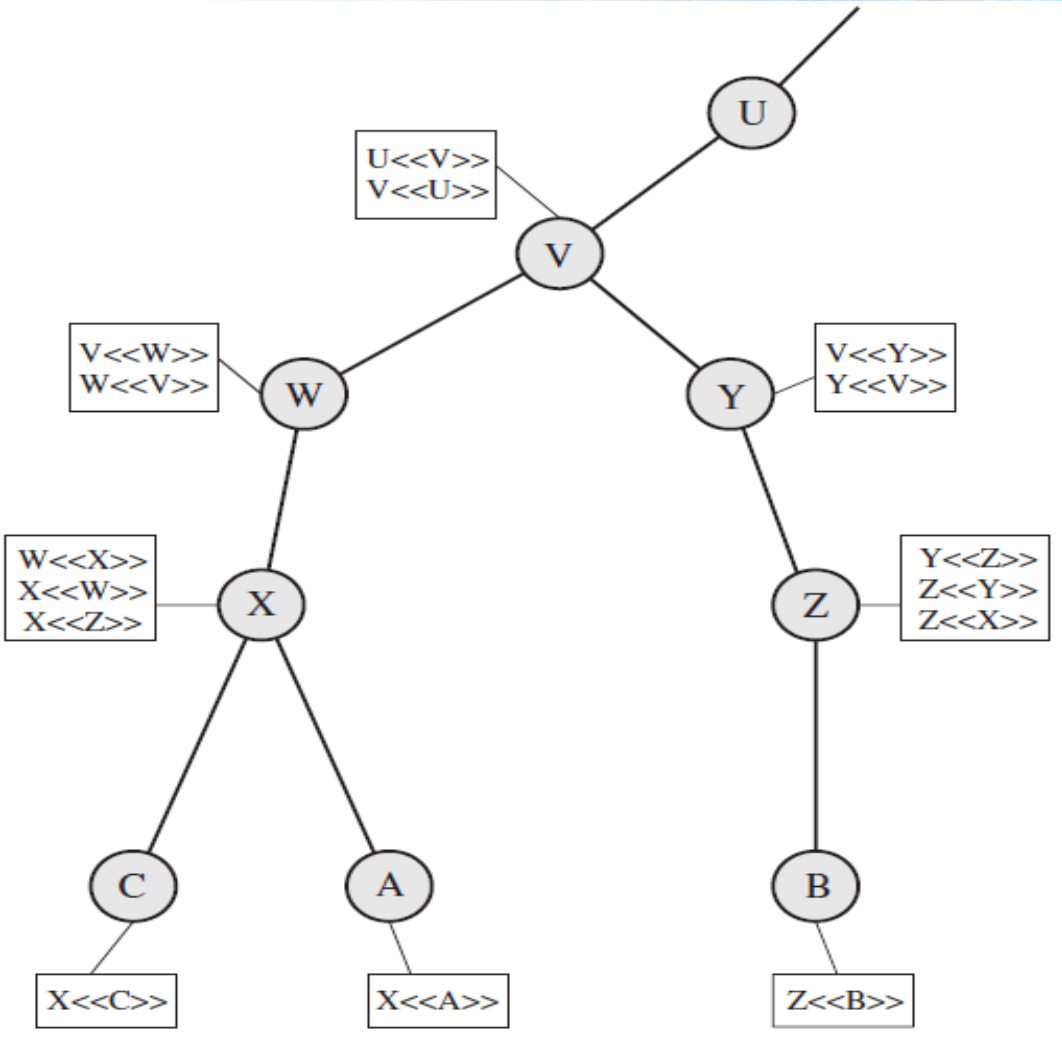
- Let certificates maintained in the directory for each CA entry be indicated by associated boxes.
- The directory entry for each CA includes two types of certificates:

Obtaining And Revoking A Certificate

- **Forward certificates:** Certificates of X generated by other CAs.
- **Reverse certificates:** Certificates generated by X that are the certificates of other CAs.

Obtaining And Revoking A Certificate

X.509 Hierarchy: An Example



Obtaining And Revoking A Certificate

- User A can establish a certification path to B:
- X <<W>> W<< V>> V <<Y>> Y<<Z>> Z<< B>>
- When A has obtained these certificates, it can unwrap the certification path in sequence to recover a B's public key.

Obtaining And Revoking A Certificate

Revocation of Certificates:

- Each certificate includes a period of validity.
- A new certificate is issued just before the expiration of the old one.

Obtaining And Revoking A Certificate

- It may be desirable on occasion to revoke a certificate before it expires, for one of the following reasons.
- **1.** The user's private key is assumed to be compromised.

Obtaining And Revoking A Certificate

- **2.** The user is no longer certified by this CA
- The subject's name has changed, or the certificate was not issued in conformance with the CA's policies.
- **3.** The CA's certificate is assumed to be compromised.

Obtaining And Revoking A Certificate

End

- Each CA must maintain a list consisting of all revoked but not expired certificates issued by that CA.
- Each certificate revocation list (CRL) should be posted on the directory.

X.509 Version 3

Network Security

X.509 Version 3

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain version 3 of X.509.

X.509 Version 3

Figures and material in this topic have been adapted from

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

X.509 Version 3

- X.509 was initially issued in 1988.
- It was subsequently revised in 1993 to address some of the security concerns.
- A third version was issued in 1995 and revised in 2000.

X.509 Version 3

- Following are the requirements not satisfied by the X.509 version 2 format:

X.509 Version 3

- 1. The subject field is inadequate to convey the identity of a key owner to a public-key user.
- X.509 names may be relatively short and lacking in obvious identification details that may be needed by the user.

X.509 Version 3

- 2. The subject field is also inadequate for many applications, which typically recognize entities by an Internet e-mail address, a URL, or some other Internet related identification.

X.509 Version 3

- 3. There is a need to indicate security policy information.
- This enables a security application or function, such as IPSec, to relate an X.509 certificate to a given policy.

X.509 Version 3

- 4. There is a need to limit the damage that can result from a faulty or malicious CA by setting constraints on the applicability of a particular certificate.

X.509 Version 3

- 5. We should be able to identify different keys used by the same owner at different times.

X.509 Version 3

- Version 3 includes a number of optional extensions that may be added to the version 2 format.
- The certificate extensions fall into three main categories:

X.509 Version 3

1. Key and Policy Information:

- These extensions convey additional information about the subject and issuer keys, plus indicators of certificate policy.

X.509 Version 3

- A certificate policy is a named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements.

X.509 Version 3

- **Authority key identifier:** Identifies the public key to be used to verify the signature on this certificate or CRL.
- **Subject key identifier:** Identifies the public key being certified.

X.509 Version 3

- **Key usage:** Indicates a restriction imposed as to the purposes for which, and the policies under which, the certified public key may be used.

X.509 Version 3

- **Private-key usage period:** Indicates the period of use of the private key corresponding to the public key.

X.509 Version 3

- **Certificate policies:** lists policies that the certificate is recognized as supporting, together with optional qualifier information.

X.509 Version 3

- **Policy mappings:** allow an issuing CA to indicate that one or more of that issuer's policies can be considered equivalent to another policy used in the subject CA's domain.

X.509 Version 3

2. Certificate Subject and Issuer Attributes:

- These extensions support alternative names, in alternative formats, for a certificate subject or certificate issuer and can convey additional information about the certificate subject.

X.509 Version 3

- **Subject alternative name:**
- Contains one or more alternative names, using any of a variety of forms.
- **Issuer alternative name:** Contains one or more alternative names, using any of a variety of forms.

X.509 Version 3

- **Subject directory attributes:** Conveys any desired X.500 directory attribute values for the subject of this certificate.

X.509 Version 3

3. Certification Path Constraints:

- These extensions allow constraint specifications to be included in certificates issued for CAs by other CAs.

X.509 Version 3

- **Basic constraints:** a certification path length constraint may be specified.
- **Name constraints:** Indicates a name space within which all subject names in subsequent certificates must be located.

X.509 Version 3

End

- **Policy constraints:** Specifies constraints that may require explicit certificate policy identification or inhibit policy mapping for the remainder of the certification path.

Public Key Infrastructure X.509 (PKIX)

Network Security

Public Key Infrastructure X.509 (PKIX)

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain elements of Public Key Infrastructure X.509 (PKIX).

Public Key Infrastructure X.509 (PKIX)

Figures and material in this topic have been adapted from

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

Public Key Infrastructure X.509 (PKIX)

- The principal objective for developing a PKI is to enable secure, convenient, and efficient acquisition of public keys.

Public Key Infrastructure X.509 (PKIX)

- RFC 4949 (Internet Security Glossary) defines public-key infrastructure (PKI) as:

Public Key Infrastructure X.509 (PKIX)

- The set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates based on asymmetric cryptography.

Public Key Infrastructure X.509 (PKIX)

- The Internet Engineering Task Force (IETF) Public Key Infrastructure X.509 (PKIX) working group has been the driving force behind setting up a generic model based on X.509 on the Internet.

Public Key Infrastructure X.509 (PKIX)

- The elements of the PKIX model are:
- End entity
- Certification authority (CA)
- Registration authority (RA)
- CRL issuer
- Repository

Public Key Infrastructure X.509 (PKIX)

- **End entity:**
- A generic term used to denote end users, devices (e.g., servers, routers), or any other entity that can be identified in the subject field of a public-key certificate.

Public Key Infrastructure X.509 (PKIX)

- End entities typically consume and/or support PKI-related services.

Public Key Infrastructure X.509 (PKIX)

- **Certification authority (CA):**
- The issuer of certificates and (usually) certificate revocation lists (CRLs).

Public Key Infrastructure X.509 (PKIX)

- It may also support a variety of administrative functions, although these are often delegated to one or more Registration Authorities.

Public Key Infrastructure X.509 (PKIX)

- **Registration authority (RA):**
- An optional component that can assume a number of administrative functions from the CA.

Public Key Infrastructure X.509 (PKIX)

- The RA is often associated with the end entity registration process but can assist in a number of other areas as well.

Public Key Infrastructure X.509 (PKIX)

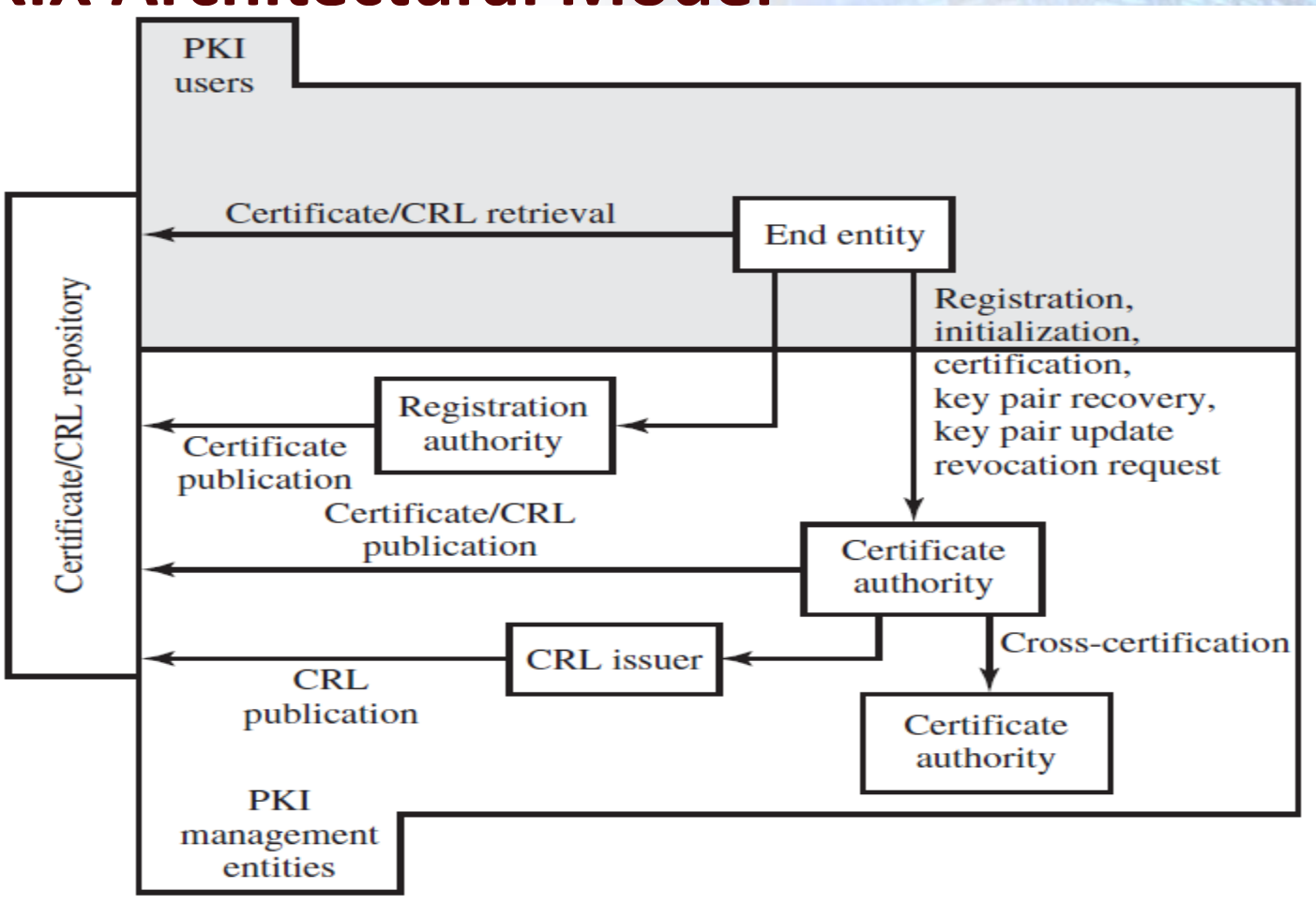
- **CRL issuer:**
- An optional component that a CA can delegate to publish CRLs.

Public Key Infrastructure X.509 (PKIX)

- **Repository:**
- A generic term used to denote any method for storing certificates and CRLs so that they can be retrieved by end entities.

Public Key Infrastructure X.509 (PKIX)

PKIX Architectural Model



PKIX Management Functions

Network Security

PKIX Management Functions

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe PKIX Management functions.

PKIX Management Functions

Figures and material in this topic have been adapted from

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

PKIX Management Functions

- The principal objective for developing a PKI is to enable secure, convenient, and efficient acquisition of public keys.

PKIX Management Functions

- PKIX identifies a number of management functions that potentially need to be supported by management protocols.

PKIX Management Functions

- **Registration:**
- This is the process whereby a user first makes itself known to a CA (directly or through an RA), prior to that CA issuing a certificate or certificates for that user.

PKIX Management Functions

- Registration begins the process of enrolling in a PKI.
- Registration usually involves some offline or online procedure for mutual authentication.

PKIX Management Functions

- Typically, the end entity is issued one or more shared secret keys used for subsequent authentication.

PKIX Management Functions

- **Initialization:** Before a client system can operate securely, it is necessary to install key materials that have the appropriate relationship with keys stored elsewhere in the infrastructure.

PKIX Management Functions

- For example, the client needs to be securely initialized with the public key and other assured information of the trusted CA(s), to be used in validating certificate paths.

PKIX Management Functions

- **Certification:** This is the process in which a CA issues a certificate for a user's public key, returns that certificate to the user's client system, and/or posts that certificate in a repository.

PKIX Management Functions

- **Key pair recovery:**
- Key pairs can be used to support digital signature creation and verification, encryption and decryption, or both.

PKIX Management Functions

- We have to provide a mechanism to recover the necessary decryption keys when normal access to the keying material is no longer possible, otherwise it will not be possible to recover the encrypted data.

PKIX Management Functions

- Loss of access to the decryption key can result from forgotten passwords/PINs, corrupted disk drives, damage to hardware tokens, and so on.

PKIX Management Functions

- Key pair recovery allows end entities to restore their encryption/decryption key pair from an authorized key backup facility (typically, the CA that issued the end entity's certificate).

PKIX Management Functions

- **Key pair update:**
- All key pairs need to be updated regularly (i.e., replaced with a new key pair) and new certificates issued.
- Update is required when the certificate lifetime expires and as a result of certificate revocation.

PKIX Management Functions

- **Revocation request:** An authorized person advises a CA of an abnormal situation requiring certificate revocation.
- Reasons include private key compromise, change in affiliation, and name change.

PKIX Management Functions

End

- **Cross certification:** Two CAs exchange information used in establishing a cross-certificate.
- A cross-certificate is a certificate issued by one CA to another CA that contains a CA signature key used for issuing certificates.

Identity Management Architecture

Network Security

Identity Management Architecture

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain identity management architecture.

Identity Management Architecture

Figures and material in this topic have been adapted from

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

Identity Management Architecture

- Identity management is a centralized, automated approach to provide enterprise-wide access to resources by employees and other authorized individuals.

Identity Management Architecture

- The focus of identity management is defining an identity for each user (human or process), associating attributes with the identity, and enforcing a means by which a user can verify identity.

Identity Management Architecture

- The central concept of an identity management system is the use of single sign-on (SSO).
- SSO enables a user to access all network resources after a single authentication.

Identity Management Architecture

- In following, we list the principal elements of an identity management system.
- **Authentication:** Confirmation that a user corresponds to the user name provided.

Identity Management Architecture

- **Authorization:** Granting access to specific services and/or resources based on the authentication.
- **Accounting:** A process for logging access and authorization.

Identity Management Architecture

- **Provisioning:** The enrollment of users in the system.
- **Workflow automation:** Movement of data in a business process.

Identity Management Architecture

- **Delegated administration:** The use of role-based access control to grant permissions.
- **Password synchronization:** Creating a process for single sign-on (SSO) or reduced sign-on (RSO).

Identity Management Architecture

- Single sign-on enables a user to access all network resources after a single authentication. RSO may involve multiple sign-ons but requires less user effort than if each resource and service maintained its own authentication facility.

Identity Management Architecture

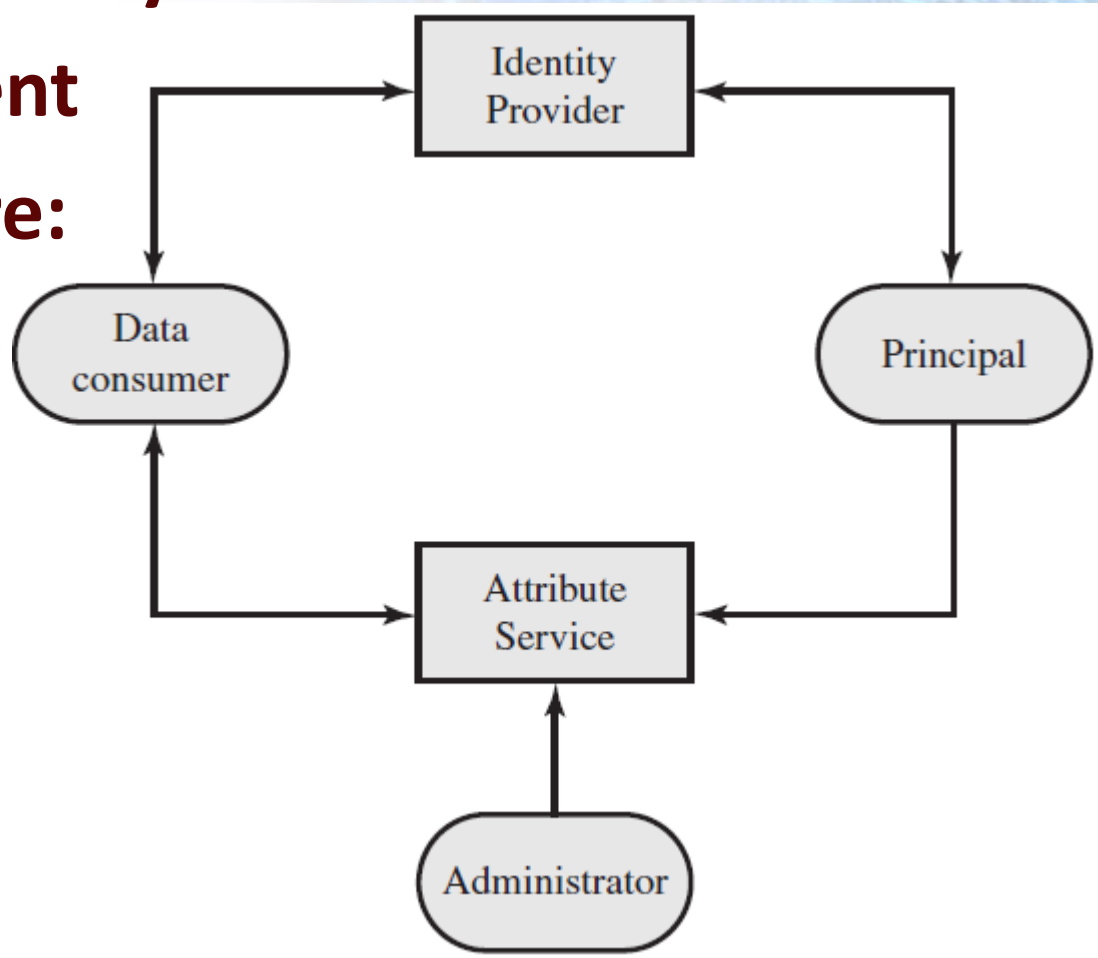
- **Self-service password reset:** Enables the user to modify his or her password.

Identity Management Architecture

- **Federation:** A process where authentication and permission will be passed on from one system to another—usually across multiple enterprises, thereby reducing the number of authentications needed by the user.

Identity Management Architecture

Generic Identity Management Architecture: Entities and data flows



Identity Management Architecture

- A **principal** is an identity holder.
- Typically, this is a human user that seeks access to resources and services on the network.
- User devices, agent processes, and server systems may also function as principals.

Identity Management Architecture

- Principals authenticate themselves to an **identity provider**.
- The identity provider associates authentication information with a principal, as well as attributes and one or more identifiers.

Identity Management Architecture

- Digital identities incorporate attributes other than simply an identifier and authentication information (such as passwords and biometric information).

Identity Management Architecture

- An **attribute service** manages the creation and maintenance of such attributes.
- E.g, a user needs to provide a shipping address each time an order is placed at a new Web merchant.
- It needs to be revised when the user moves.

Identity Management Architecture

- Identity management enables the user to provide this information once, so that it is maintained in a single place and released to data consumers in accordance with authorization and privacy policies.

Identity Management Architecture

- Users may create some of the attributes to be associated with their digital identity, such as address.
- **Administrators** may also assign attributes to users, such as roles, access permissions, and employee information.

Identity Management Architecture

- **Data consumers** are entities that obtain and employ data maintained and provided by identity and attribute providers, which are often used to support authorization decisions and to collect audit information.

Identity Management Architecture

End

- For example, a database server or file server is a data consumer that needs a client's credentials so as to know what access to provide to that client.

Federated Identity Operation

Network Security

Federated Identity Operation

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain federated identity operation and the standards used.

Federated Identity Operation

Figures and material in this topic have been adapted from

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

Federated Identity Operation

- Identity federation is an extension of identity management to multiple security domains such as autonomous internal business units, external business partners, and other third-party applications and services.

Federated Identity Operation

- The goal is to provide the sharing of digital identities so that a user can be authenticated a single time and then access applications and resources across multiple domains.

Federated Identity Operation

- Being autonomous, a centralized control is not suitable.
- The cooperating organizations must form a federation based on agreed standards and mutual levels of trust to securely share digital identities.

Federated Identity Operation

- Federated identity management refers to the agreements, standards, and technologies that enable the portability of identities, identity attributes, and entitlements across multiple enterprises.

Federated Identity Operation

- This will allow an employee to log onto her corporate intranet and be authenticated to perform authorized functions and access authorized services on that intranet.

Federated Identity Operation

- The employee could then access their health benefits from an outside health-care provider without having to reauthenticate.

Federated Identity Operation

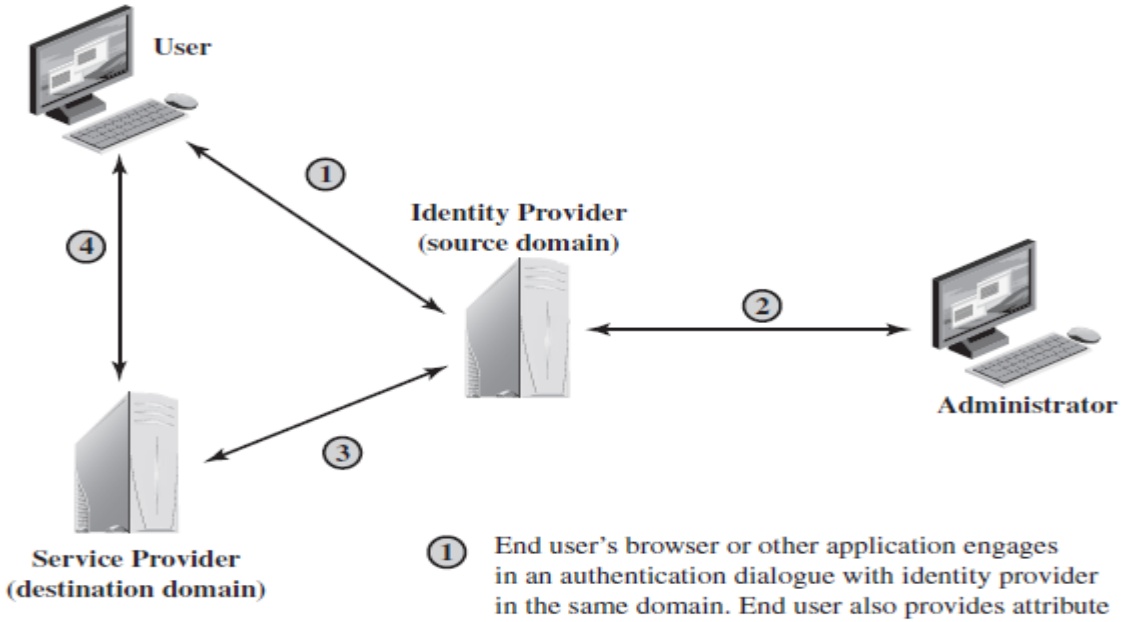
- Beyond SSO, federated identity management provides other capabilities:
- A standardized means of representing attributes such as account numbers, organizational roles, physical location, and file ownership.

Federated Identity Operation

- Another key function is identity mapping.
- Different security domains may represent identities and attributes differently.

Federated Identity Operation

Federated Identity Operation



1. End user's browser or other application engages in an authentication dialogue with identity provider in the same domain. End user also provides attribute values associated with user's identity.
2. Some attributes associated with an identity, such as allowable roles, may be provided by an administrator in the same domain.
3. A service provider in a remote domain, which the user wishes to access, obtains identity information, authentication information, and associated attributes from the identity provider in the source domain.
4. Service provider opens session with remote user and enforces access control restrictions based on user's identity and attributes.

Federated Identity Operation

- The power of this approach is for federated identity management, in which the service provider is in a different domain as the user and identity provider.

Federated Identity Operation

- Federated identity management uses a number of standards as the building blocks for secure identity exchange across different domains or heterogeneous systems.

Federated Identity Operation

- **The Extensible Markup Language (XML):** A markup language that uses sets of embedded tags to characterize text elements within a document to indicate their appearance, function, meaning, or context.

Federated Identity Operation

- **The Simple Object Access Protocol (SOAP):**
- enables applications to request services from one another with XML-based requests and receive responses as data formatted with XML.

Federated Identity Operation

- **WS-Security:**
- A set of SOAP extensions for implementing message integrity and confidentiality in Web services.

Federated Identity Operation

End

- **Security Assertion Markup Language (SAML):**
- An XML-based language for the exchange of security information between online business partners.

Federated Identity Scenarios

Network Security

Federated Identity Scenarios

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe federated identity scenarios.

Federated Identity Scenarios

Figures and material in this topic have been adapted from

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

Federated Identity Scenarios

- **Scenario 1:**
- Workplace.com contracts with Health.com to provide employee health benefits.
- The organizations are part of a federation that cooperatively exchanges user identifiers.

Federated Identity Scenarios

- An employee uses a Web interface to sign on to Workplace.com and goes through authentication there.
- This enables the employee to access authorized services and resources at Workplace.com.

Federated Identity Scenarios

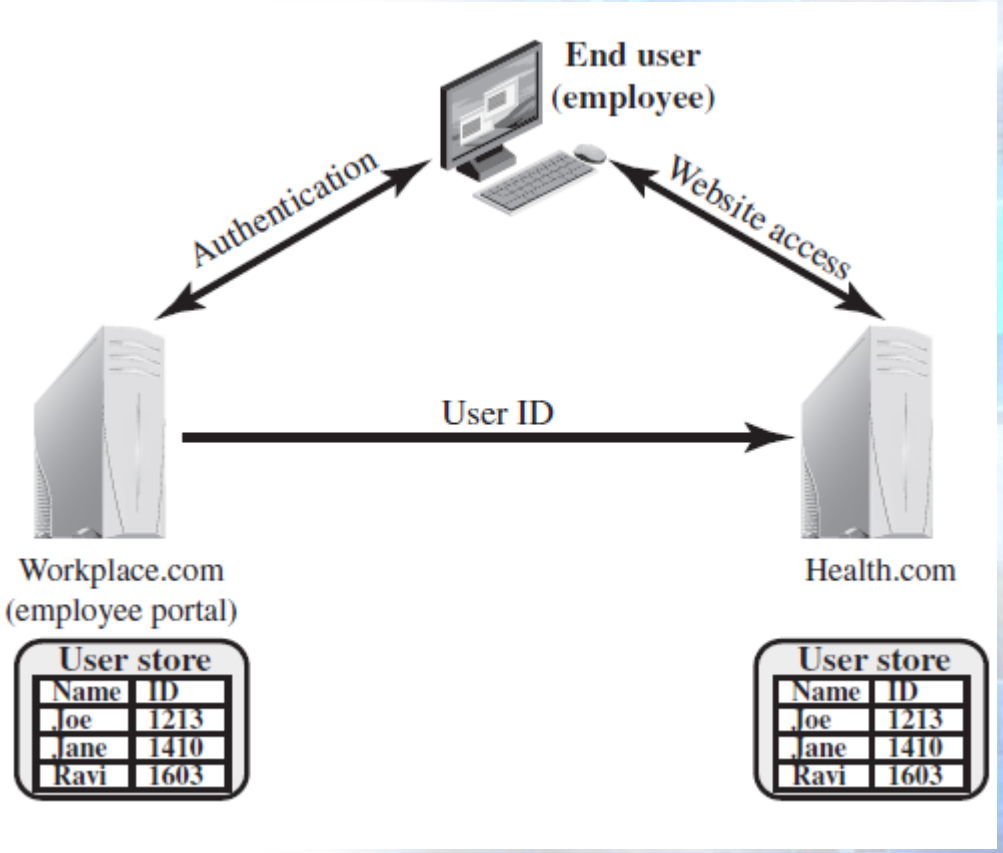
- When employee clicks on a link to access health benefits, her browser is redirected to Health.com.
- At the same time, the Workplace.com software passes the user's identifier to Health.com in a secure manner.

Federated Identity Scenarios

- Health.com maintains user identities for every employee at Workplace.com and associates with each identity health-benefits information and access rights.
- Linkage between two companies is based on account information.

Federated Identity Scenarios

Federation based on account linking



Federated Identity Scenarios

- **Scenario 2:**
- PartsSupplier. com is a regular supplier of parts to Workplace.com.
- A role-based access-control (RBAC) scheme is used for access to information.

Federated Identity Scenarios

- An engineer of Workplace.com authenticates at the employee portal at Workplace.com and clicks on a link to access information at PartsSupplier.com.

Federated Identity Scenarios

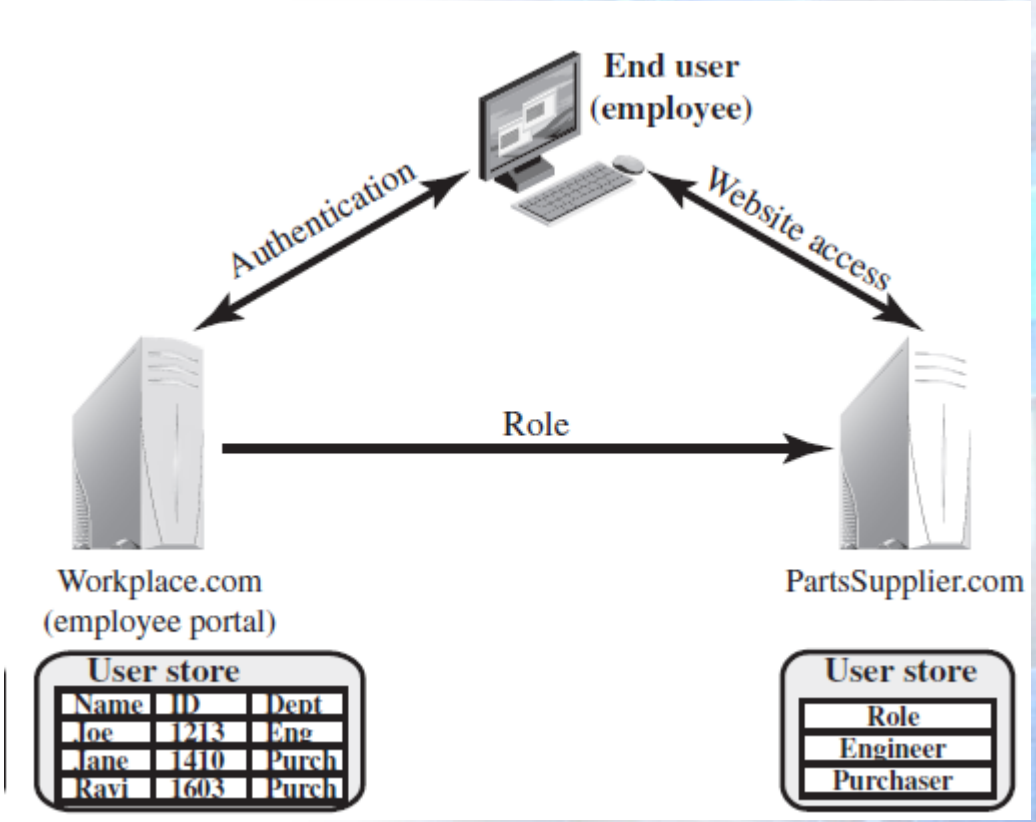
- Because the user is authenticated in the role of an engineer, he is taken to the technical documentation of PartsSupplier.com's Web site without having to sign on.

Federated Identity Scenarios

- For this scenario, PartsSupplier.com does not have identity information for individual employees at Workplace.com.
- The linkage between the two federated partners is in terms of roles.

Federated Identity Scenarios

Federation based on roles



Federated Identity Scenarios

- **Scenario 3:**
- Workplace.com has a purchasing agreement with PinSupplies.com, and PinSupplies.com has a business relationship with E-Ship.com.

Federated Identity Scenarios

- An employee of WorkPlace.com signs on and is authenticated.
- He goes to a procurement application that provides a list of WorkPlace.com's suppliers and parts that can be ordered.

Federated Identity Scenarios

- The user clicks on the PinSupplies button and is presented with a purchase order Web page (HTML page).
- The employee fills out the form and clicks the submit button.

Federated Identity Scenarios

- The procurement application generates an XML/SOAP document that it inserts into the envelope body of an XML-based message.

Federated Identity Scenarios

- The procurement application then inserts user's credentials and Workplace.com's organizational identity in header of message.
- It posts the message to PinSupplies.com's purchasing Web service.

Federated Identity Scenarios

- This service authenticates the incoming message and processes the request.
- The purchasing Web service then sends a SOAP message to its shipping partner to fulfill the order.

Federated Identity Scenarios

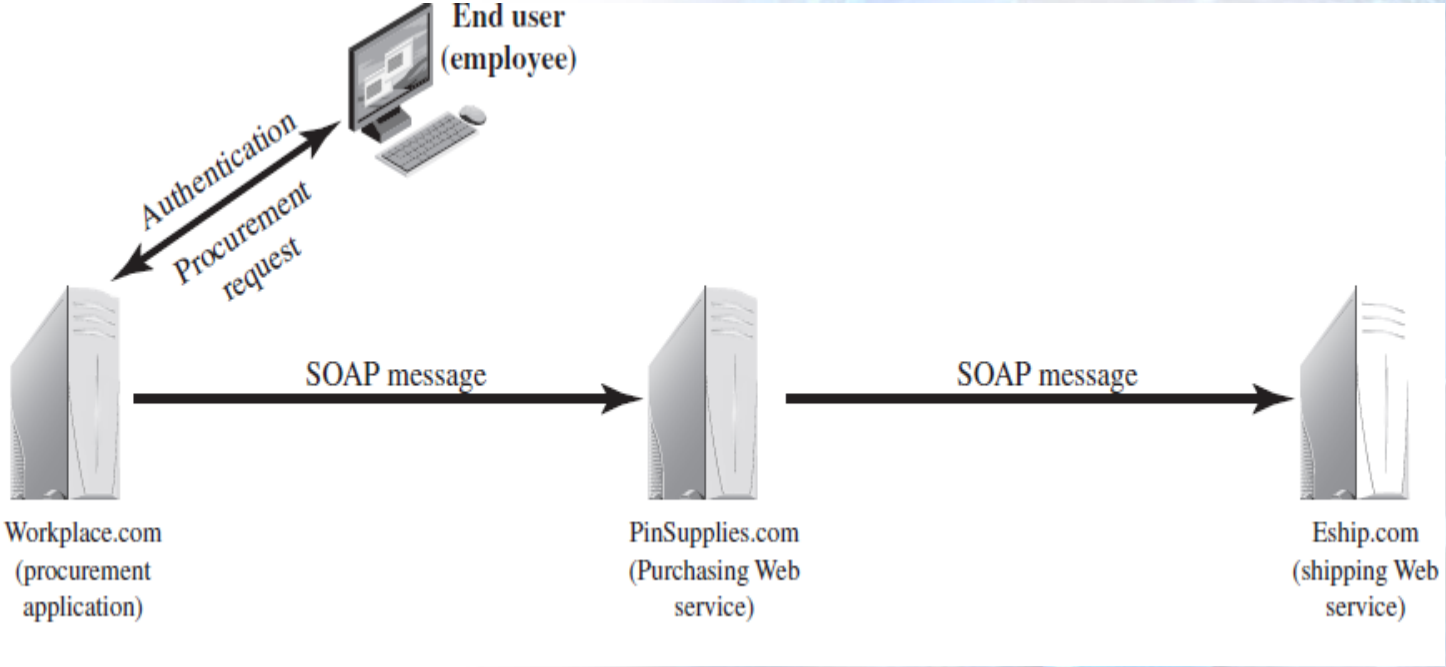
- The message includes a PinSupplies.com security token in the envelope header and the list of items to be shipped as well as the end user's shipping information in the envelope body.

Federated Identity Scenarios

- The shipping Web service authenticates the request and processes the shipment order.

Federated Identity Scenarios

Federation based on roles



Elements of Network Access Control

Network Security

Elements of Network Access Control

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe the elements of network access control.

Elements of Network Access Control

Figures and material in this topic have been adapted from

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

Elements of Network Access Control

- Network access control (NAC) is an umbrella term for managing access to a network.

Elements of Network Access Control

- NAC authenticates users logging into the network and determines what data they can access and actions they can perform.
- NAC also examines the health of the user's computer (the endpoints).

Elements of Network Access Control

- **Elements of a Network Access Control System:**
- NAC systems deal with three categories of components:

Elements of Network Access Control

- **1. Access requestor (AR):** The AR is node that is attempting to access network and may be any device that is managed by the NAC system.
- E.g. workstations, servers, printers, cameras, and other IP-enabled devices.

Elements of Network Access Control

- ARs are also referred to as **supplicants**, or simply, clients.

Elements of Network Access Control

- **2. Policy server:**
- Based on the AR's posture and an enterprise's defined policy, the policy server determines what access should be granted.

Elements of Network Access Control

- The policy server often relies on backend systems, including antivirus, patch management, or a user directory, to help determine the host's condition.

Elements of Network Access Control

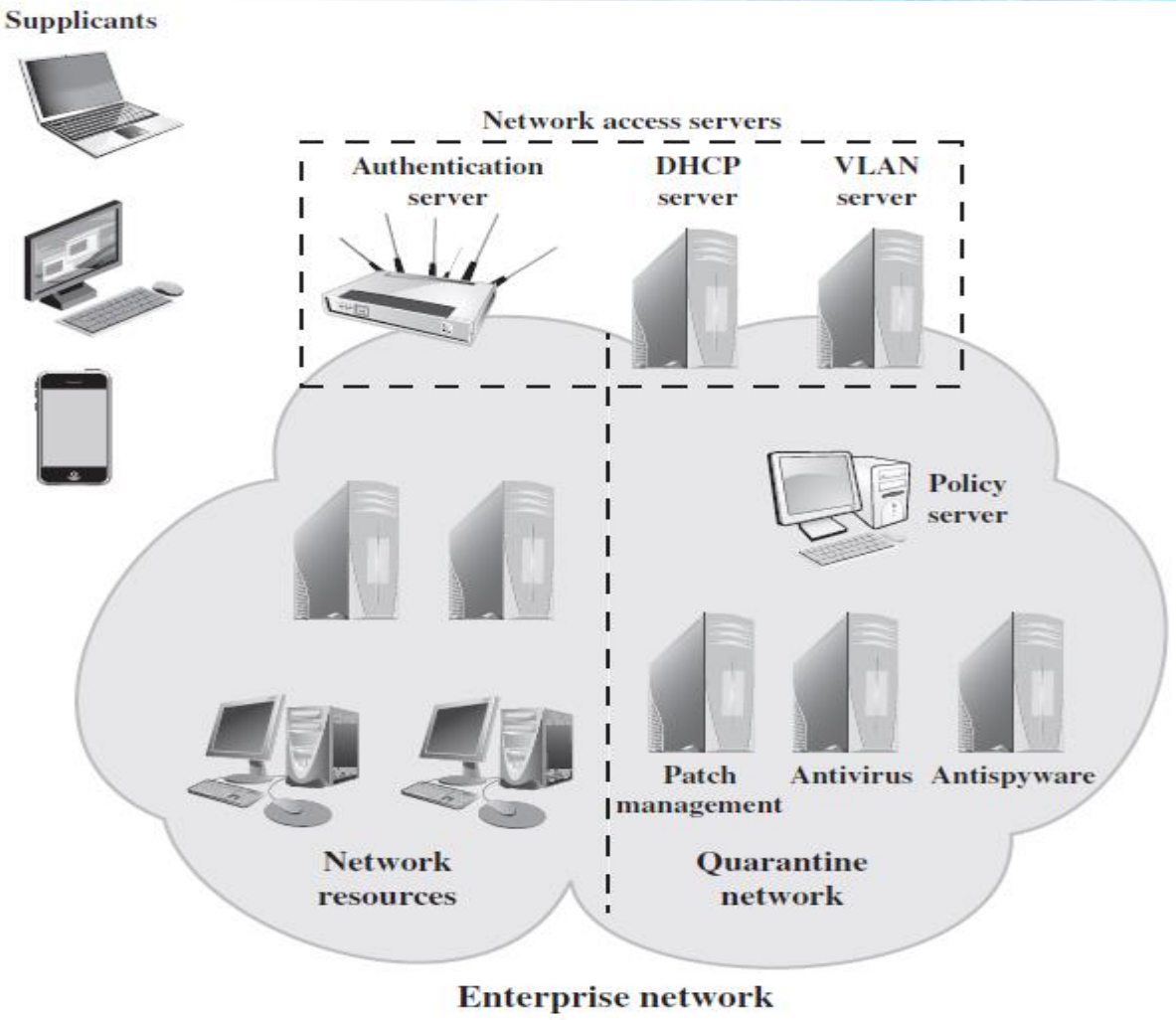
- **3. Network access server (NAS):** The NAS functions as an access control point for users in remote locations connecting to an enterprise's internal network.

Elements of Network Access Control

- Also called a **media gateway**, a **remote access server (RAS)**, or a **policy server**, an NAS may include its own authentication services or rely on a separate authentication service from the policy server.

Elements of Network Access Control

Network Access Control Context



Elements of Network Access Control

- A variety of different ARs seek access to an enterprise network by applying to some type of NAS.
- The first step is to authenticate the AR.
- This involves some sort of secure protocol and the use of cryptographic keys.

Elements of Network Access Control

- Authentication may be performed by the NAS, or the NAS may mediate the authentication process.

Elements of Network Access Control

- In the latter case, authentication takes place between the supplicant and an authentication server that is part of the policy server or that is accessed by the policy server.
- Authentication has a number of purposes:

Elements of Network Access Control

- The authentication verifies a supplicant's claimed identity, which enables the policy server to determine what access privileges, if any, the AR may have.

Elements of Network Access Control

- The authentication exchange may result in the establishment of session keys to enable future secure communication between the supplicant and resources on the enterprise network.

Elements of Network Access Control

- The policy server or a supporting server will perform checks on the AR to determine if it should be permitted interactive remote access connectivity.
- For example, user's antimalware software must be up-to-date, OS must be patched.

Elements of Network Access Control

- These checks should be performed before granting the AR access to the enterprise network.

Elements of Network Access Control

- If an authorized user's remote computer does not pass the health check, the user and remote computer should be denied network access or have limited access to a quarantine network so that authorized personnel can fix the security deficiencies.

Elements of Network Access Control

End

- Once an AR has been authenticated and cleared for a certain level of access to the enterprise network, the NAS can enable the AR to interact with resources in the enterprise network.

Network Access Enforcement Methods

Network Security

Network Access Enforcement Methods

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe network access enforcement methods.

Network Access Enforcement Methods

Figures and material in this topic have been adapted from

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

Network Access Enforcement Methods

- Network access control (NAC) is an umbrella term for managing access to a network.

Network Access Enforcement Methods

- NAC authenticates users logging into network and determines what data they can access and actions they can perform.
- NAC also examines health of endpoints (e.g. computers, mobile devices).

Network Access Enforcement Methods

- NAC systems deal with three categories of components:
- **1. Access requestor (AR):** is node that is attempting to access network and may be any device that is managed by the NAC system. E.g. servers, printers, cameras etc.

Network Access Enforcement Methods

- **2. Policy server:**
- determines what access should be granted.
- It often relies on backend systems, including antivirus, patch management, or a user directory, to help determine the host's condition.

Network Access Enforcement Methods

- **3. Network access server (NAS):**
- It functions as an access control point for users in remote locations connecting to an enterprise's internal network.

Network Access Enforcement Methods

- **Network Access Enforcement Methods:**
- are the actions that are applied to ARs to regulate access to the enterprise network.

Network Access Enforcement Methods

- Many vendors support multiple enforcement methods simultaneously, allowing the customer to tailor the configuration by using one or a combination of methods.

Network Access Enforcement Methods

- The following are common NAC enforcement methods.

Network Access Enforcement Methods

- **IEEE 802.1X:** This is a link layer protocol that enforces authorization before a port is assigned an IP address.
- It makes use of the Extensible Authentication Protocol for authentication.

Network Access Enforcement Methods

- **Virtual local area networks (VLANs):**
- In this approach, the enterprise network, consisting of an interconnected set of LANs, is segmented logically into a number of virtual LANs.

Network Access Enforcement Methods

- The NAC system decides to which of the network's VLANs it will direct an AR, based on whether the device needs security remediation, Internet access only, or some level of network access to enterprise resources.

Network Access Enforcement Methods

- VLANs can be created dynamically and VLAN membership, of both enterprise servers and ARs, may overlap.
- That is, an enterprise server or an AR may belong to more than one VLAN.

Network Access Enforcement Methods

- **Firewall:** A firewall provides a form of NAC by allowing or denying network traffic between an enterprise host and an external user.

Network Access Enforcement Methods

- **DHCP management:**
- The Dynamic Host Configuration Protocol (DHCP) is an Internet protocol that enables dynamic allocation of IP addresses to hosts.
- A DHCP server intercepts DHCP requests and assigns IP addresses instead.

Network Access Enforcement Methods

- NAC enforcement occurs at the IP layer based on subnet and IP assignment.
- A DCHP server is easy to install and configure, but is subject to various forms of IP spoofing, providing limited security.

Network Access Enforcement Methods

End

- There are a number of other enforcement methods available from vendors.
- The ones mentioned above are perhaps the most common, and IEEE 802.1X is by far the most commonly implemented solution.

Extensible Authentication Protocol

Network Security

Extensible Authentication Protocol

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe basics of extensible authentication protocol.

Extensible Authentication Protocol

Figures and material in this topic have been adapted from

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

Extensible Authentication Protocol

- EAP provides a generic transport service for the exchange of authentication information between a client system and an authentication server.

Extensible Authentication Protocol

- EAP is defined in RFC 3748.
- It acts as a framework for network access and authentication protocols.

Extensible Authentication Protocol

- EAP provides a set of protocol messages that can encapsulate various authentication methods to be used between a client and an authentication server.

Extensible Authentication Protocol

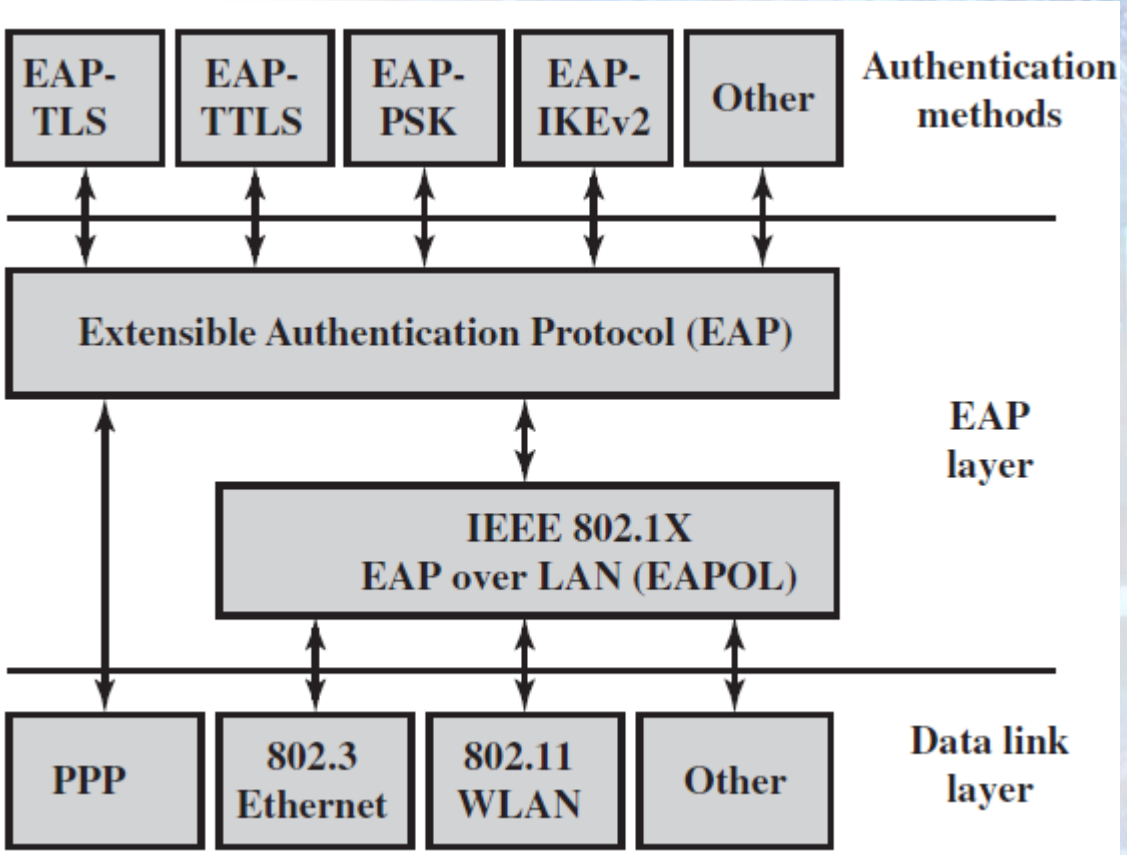
- EAP can operate over a variety of network and link level facilities.
- E.g. point-to-point links, LANs, and other networks.

Extensible Authentication Protocol

- The protocol layers that form the context for EAP are shown next.

Extensible Authentication Protocol

EAP Layered Context



Extensible Authentication Protocol

- **Authentication Methods:**
- EAP supports multiple authentication methods.
- This is what is meant by referring to EAP as *extensible*.
- Commonly supported EAP methods are:

Extensible Authentication Protocol

- **EAP-TLS (EAP Transport Layer Security):**
- It is defined RFC 5216.
- It defines how the TLS protocol can be encapsulated in EAP messages.
- It uses the handshake protocol in TLS, not its encryption method.

Extensible Authentication Protocol

- Client and server authenticate each other using digital certificates.
- Client generates a pre-master secret key by encrypting a random number with the server's public key and sends it to the server.

Extensible Authentication Protocol

- Both client and server use the pre-master to generate the same secret key.

Extensible Authentication Protocol

- **EAP-TTLS (EAP Tunneled TLS):**
- It is defined in RFC 5281.
- EAP-TTLS is like EAP-TLS, except only the server has a certificate to authenticate itself to the client first.

Extensible Authentication Protocol

- As in EAP-TLS, a secure connection (the “tunnel”) is established with secret keys.
- This connection is used to continue the authentication process.

Extensible Authentication Protocol

- The client and possibly the server are authenticated again using any EAP method or legacy method such as PAP (Password Authentication Protocol) and CHAP (Challenge-Handshake Authentication Protocol).

Extensible Authentication Protocol

- **EAP-GPSK (EAP Generalized Pre-Shared Key):**
- It is defined in RFC 5433.
- It is an EAP method for mutual authentication and session key derivation using a Pre-Shared Key (PSK).

Extensible Authentication Protocol

- This method is efficient in terms of message flows and computational costs, but requires the existence of pre-shared keys between each peer and EAP server.

Extensible Authentication Protocol

- The set up of these pairwise secret keys is part of the peer registration.
- It is designed for authentication over insecure networks such as IEEE 802.11.

Extensible Authentication Protocol

End

- **EAP-IKEv2:** is based on the Internet Key Exchange protocol version 2 (IKEv2).
- It supports mutual authentication and session key establishment using a variety of methods.
- It is defined in RFC 5106.

EAP Protocol Exchanges

Network Security

EAP Protocol Exchanges

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe working of EAP protocol exchanges.

EAP Protocol Exchanges

Figures and material in this topic have been adapted from

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

EAP Protocol Exchanges

- The authentication information and authentication protocol information are carried in Extensible Authentication Protocol (EAP) messages.

EAP Protocol Exchanges

- A successful authentication is the one for which the authenticator decides to allow access by the peer, and the peer decides to use this access.
- Following components are involved:

EAP Protocol Exchanges

- **EAP peer:** Client computer that is attempting to access a network.
- **EAP authenticator:** An access point or NAS that requires EAP authentication prior to granting access to a network.

EAP Protocol Exchanges

- **Authentication server:** negotiates use of a specific EAP method with an EAP peer, validates EAP peer's credentials and authorizes access to the network.
- is a Remote Authentication Dial-In User Service (RADIUS)

EAP Protocol Exchanges

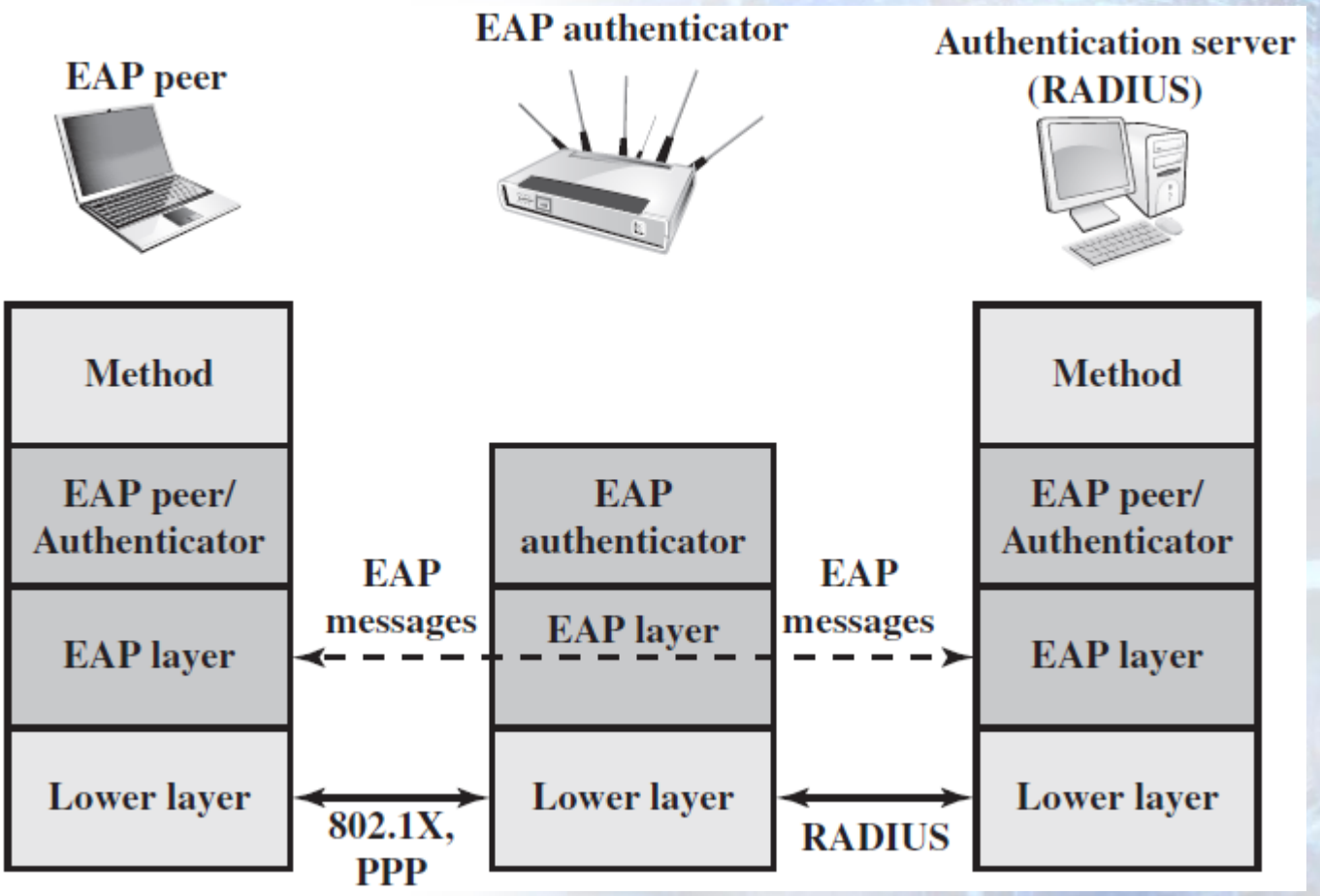
- The authentication server functions as a backend server that can authenticate peers as a service to a number of EAP authenticators.

EAP Protocol Exchanges

- The EAP authenticator then makes the decision of whether to grant access.
- This is referred to as the **EAP pass-through mode**.

EAP Protocol Exchanges

EAP Protocol Exchanges



EAP Protocol Exchanges

- As a first step, a lower-level protocol, such as PPP or IEEE 802.1X, is used to connect to the EAP authenticator.
- The software entity in the EAP peer that operates at this level is referred to as the supplicant.

EAP Protocol Exchanges

- EAP messages containing the appropriate information for a chosen EAP method are then exchanged between the EAP peer and the authentication server.

EAP Protocol Exchanges

- EAP messages include the following fields:
- **Code:** Identifies the Type of EAP message. The codes are Request (1), Response (2), Success (3), and Failure (4).
- **Identifier:** Used to match Responses with Requests.

EAP Protocol Exchanges

- **Length:** Indicates the length, in octets, of the EAP message, including the Code, Identifier, Length, and Data fields.
- **Data:** Contains information related to authentication.

EAP Protocol Exchanges

- After a lower-level exchange that established the need for an EAP exchange, the authenticator sends a Request to the peer to request an identity, and the peer sends a Response with the identity information.

EAP Protocol Exchanges

- This is followed by a sequence of Requests by the authenticator and Responses by the peer for the exchange of authentication information.

EAP Protocol Exchanges

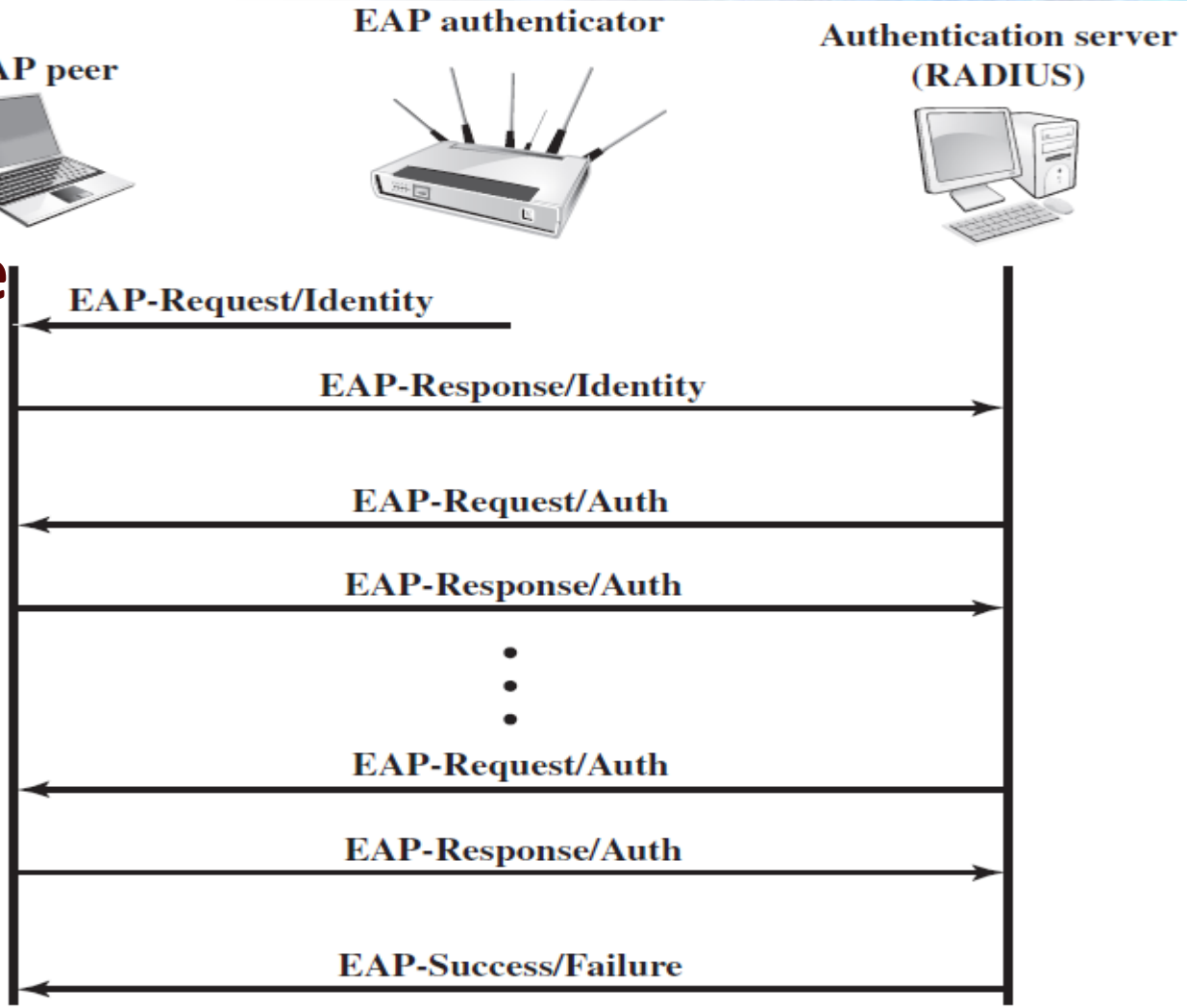
- The information exchanged and the number of Request–Response exchanges needed depend on the authentication method.

EAP Protocol Exchanges

- This continues until either (1) the authenticator determines that it cannot authenticate peer and transmits an EAP Failure or (2) the authenticator transmits an EAP Success indicating a successful authentication.

EAP Protocol Exchanges

EAP Message Flow in Pass-Through Mode



IEEE 802.1X Port-Based NAC

Network Security

IEEE 802.1X Port-Based NAC

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain basics of IEEE 802.1X access control.

IEEE 802.1X Port-Based NAC

Figures and material in this topic have been adapted from

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

IEEE 802.1X Port-Based NAC

- IEEE 802.1X Port-Based Network Access Control was designed to provide access control functions for LANs.

IEEE 802.1X Port-Based NAC

Terminology Related to IEEE 802.1X

Authenticator

An entity at one end of a point-to-point LAN segment that facilities authentication of the entity to the other end of the link.

Authentication exchange

The two-party conversation between systems performing an authentication process.

Authentication process

The cryptographic operations and supporting data frames that perform the actual authentication.

Authentication server (AS)

An entity that provides an authentication service to an authenticator. This service determines, from the credentials provided by supplicant, whether the supplicant is authorized to access the services provided by the system in which the authenticator resides.

IEEE 802.1X Port-Based NAC

Terminology Related to IEEE 802.1X

Authentication transport

The datagram session that actively transfers the authentication exchange between two systems.

Bridge port

A port of an IEEE 802.1D or 802.1Q bridge.

Edge port

A bridge port attached to a LAN that has no other bridges attached to it.

Network access port

A point of attachment of a system to a LAN. It can be a physical port, such as a single LAN MAC attached to a physical LAN segment, or a logical port, for example, an IEEE 802.11 association between a station and an access point.

IEEE 802.1X Port-Based NAC

Terminology Related to IEEE 802.1X

Port access entity (PAE)

The protocol entity associated with a port. It can support the protocol functionality associated with the authenticator, the supplicant, or both.

Supplicant

An entity at one end of a point-to-point LAN segment that seeks to be authenticated by an authenticator attached to the other end of that link.

IEEE 802.1X Port-Based NAC

- The terms supplicant, network access point, and authentication server correspond to the EAP terms peer, authenticator, and authentication server, respectively.

IEEE 802.1X Port-Based NAC

- Until the AS authenticates a supplicant (using an authentication protocol), the authenticator only passes control and authentication messages between the supplicant and the AS.

IEEE 802.1X Port-Based NAC

- The 802.1X control channel is unblocked, but the 802.11 data channel is blocked.

IEEE 802.1X Port-Based NAC

- Once a supplicant is authenticated and keys are provided, the authenticator can forward data from the supplicant, subject to predefined access control limitations for supplicant to network.
- Now data channel is unblocked.

IEEE 802.1X Port-Based NAC

- 802.1X uses the concepts of controlled and uncontrolled ports.
- Ports are logical entities defined within the authenticator and refer to physical network connections.

IEEE 802.1X Port-Based NAC

- Each logical port is mapped to one of these two types of physical ports.

IEEE 802.1X Port-Based NAC

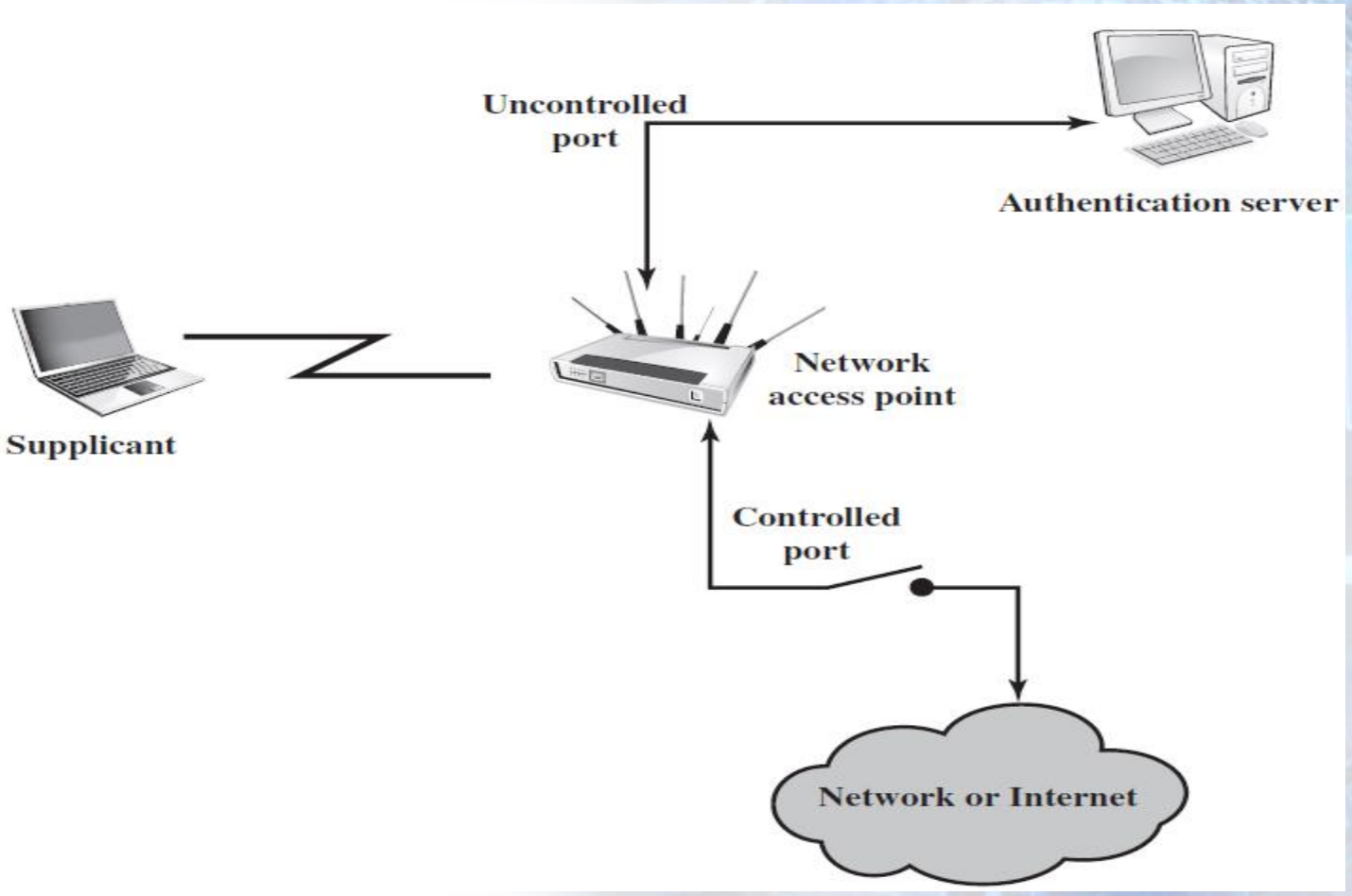
- An uncontrolled port allows the exchange of protocol data units (PDUs) between the supplicant and the AS, regardless of the authentication state of the supplicant.

IEEE 802.1X Port-Based NAC

- A controlled port allows the exchange of PDUs between a supplicant and other systems on the network only if the current state of the supplicant authorizes such an exchange.

IEEE 802.1X Port-Based NAC

802.1X Access Control



Working of IEEE 802.1X Port-Based NAC

Network Security

Working of IEEE 802.1X Port-Based NAC

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain working of IEEE 802.1X access control.

Working of IEEE 802.1X Port-Based NAC

Figures and material in this topic have been adapted from

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

Working of IEEE 802.1X Port-Based NAC

- IEEE 802.1X Port-Based Network Access Control was designed to provide access control functions for LANs.

Working of IEEE 802.1X Port-Based NAC

- The essential element defined in 802.1X is a protocol known as EAPOL (EAP over LAN).
- EAPOL operates at the network layers and makes use of an IEEE 802 LAN, such as Ethernet or Wi-Fi, at the link level.

Working of IEEE 802.1X Port-Based NAC

- EAPOL enables a supplicant to communicate with an authenticator and supports the exchange of EAP packets for authentication.
- The most common EAPOL packets are listed next:

Working of IEEE 802.1X Port-Based NAC

Common EAPOL Frame Types

Frame Type	Definition
EAPOL-EAP	Contains an encapsulated EAP packet.
EAPOL-Start	A supplicant can issue this packet instead of waiting for a challenge from the authenticator.
EAPOL-Logoff	Used to return the state of the port to unauthorized when the supplicant if finished using the network.
EAPOL-Key	Used to exchange cryptographic keying information.

Working of IEEE 802.1X Port-Based NAC

- When the supplicant first connects to the LAN, it does not know the MAC address of the authenticator.
- Actually it doesn’t know whether there is an authenticator present at all.

Working of IEEE 802.1X Port-Based NAC

- By sending an **EAPOL-Start** packet to a special group-multicast address reserved for IEEE 802.1X authenticators, a supplicant can determine presence of authenticator and let it know that the supplicant is ready.

Working of IEEE 802.1X Port-Based NAC

- In many cases, the authenticator will already be notified that a new device has connected from some hardware notification.
- In this case, the authenticator may preempt the **Start** message with its own message.

Working of IEEE 802.1X Port-Based NAC

- In either case the authenticator sends an **EAP-Request Identity** message encapsulated in an **EAPOL-EAP** packet.
- The EAPOL-EAP is the EAPOL frame type used for transporting EAP packets.

Working of IEEE 802.1X Port-Based NAC

- Authenticator uses the **EAP-Key** packet to send cryptographic keys to the supplicant once it has decided to admit it to network.
- The **EAP-Logoff** packet indicates that supplicant wishes to be disconnected from the network.

Working of IEEE 802.1X Port-Based NAC

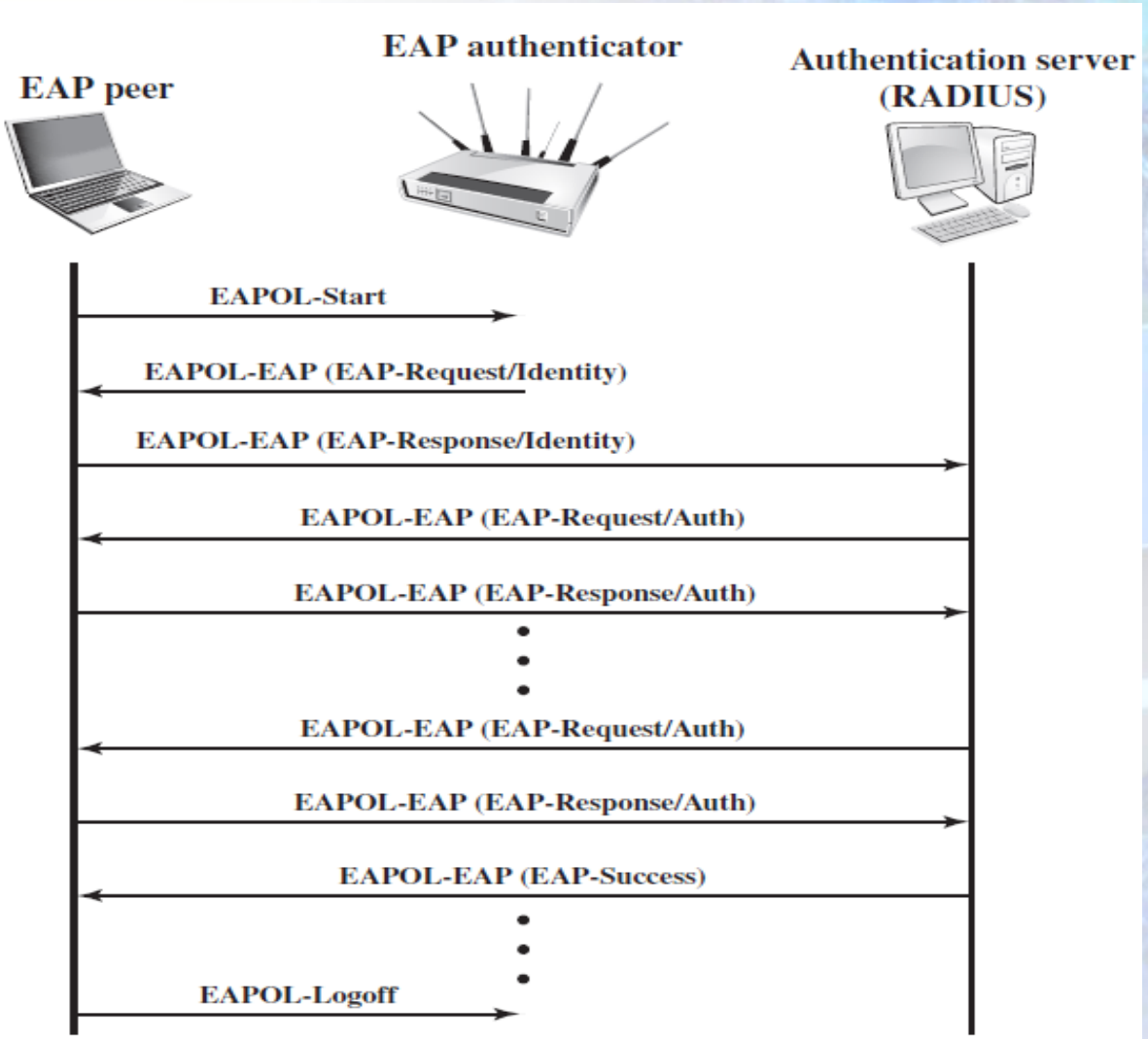
- The EAPOL packet format includes the following fields:
- **Protocol version:** version of EAPOL.
- **Packet type:** indicates start, EAP, key, logoff, etc.

Working of IEEE 802.1X Port-Based NAC

- **Packet body length:** If the packet includes a body, this field indicates the body length.
- **Packet body:** The payload for this EAPOL packet. An example is an EAP packet.

Working of IEEE 802.1X Port-Based NAC

Example Timing Diagram



Network Security

Cloud Computing and its Elements

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain basics of Cloud Computing.

Cloud Computing and its Elements

Figures and material in this topic have been adapted from

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

Cloud Computing and its Elements

- There is a growing interest in many organizations to move information technology (IT) operations to an Internet-connected infrastructure known as enterprise cloud computing.

Cloud Computing and its Elements

The NIST Definition of Cloud Computing: NIST SP-800-145

Cloud computing: A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models.

Cloud Computing and its Elements

Elements:

- **Broad network access:** Capabilities are available over network and accessed through standard mechanisms that promote use by client platforms (e.g., mobile phones, laptops, and PDAs).

Cloud Computing and its Elements

- **Rapid elasticity:** the ability to expand and reduce resources according to your specific service requirement.
- **Measured service:** automatically control and optimize resource use by leveraging a metering capability.

Cloud Computing and its Elements

- **On-demand self-service:** A consumer can unilaterally provision computing capabilities.
- **Resource pooling:** The provider's computing resources are pooled to serve multiple consumers.

Cloud Computing and its Elements

Service Models:

- **1. Software as a service (SaaS):**
- The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure.
- E.g. Gmail.

Cloud Computing and its Elements

- **2. Platform as a service (PaaS):** the capability provided to consumer is to deploy onto the cloud consumer-created or acquired applications created using programming languages and tools supported by provider.

Cloud Computing and its Elements

- **3. Infrastructure as a service (IaaS):** The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources.

Cloud Computing and its Elements

Deployment models

- **1. Public cloud:** The cloud infrastructure is made available to the general public.
- The cloud provider is responsible both for cloud infrastructure and for the control of data and operations within the cloud.

Cloud Computing and its Elements

- **2. Private cloud:** The cloud infrastructure is operated solely for an organization.
- The cloud provider (CP) is responsible only for the infrastructure and not for the control.

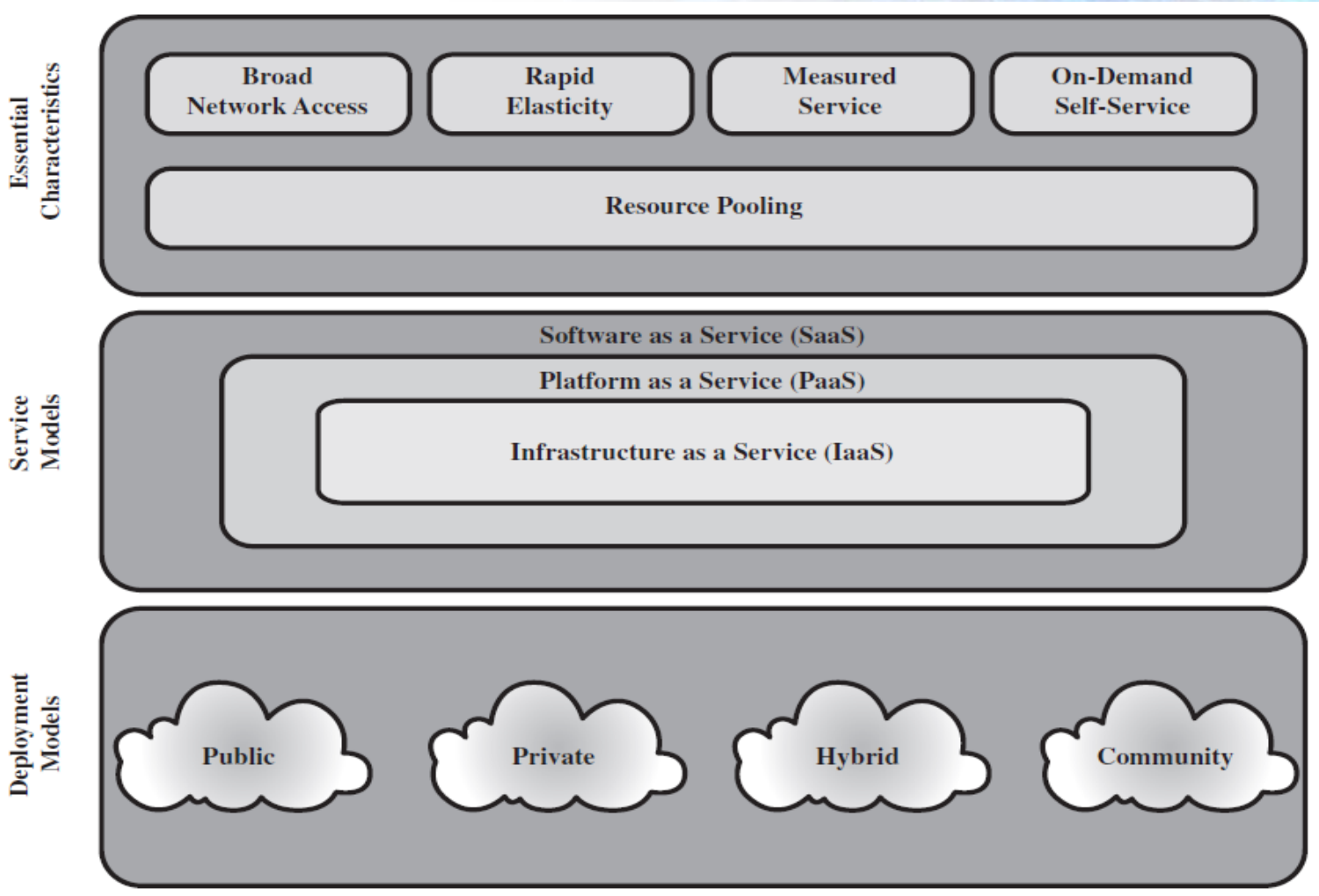
Cloud Computing and its Elements

- **3. Community cloud:** The cloud infrastructure is shared by several organizations and supports a specific community.
- may be managed by organizations or third party and may exist on or off premise.

Cloud Computing and its Elements

- **4. Hybrid cloud:** The cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by technology that enables data and application portability.

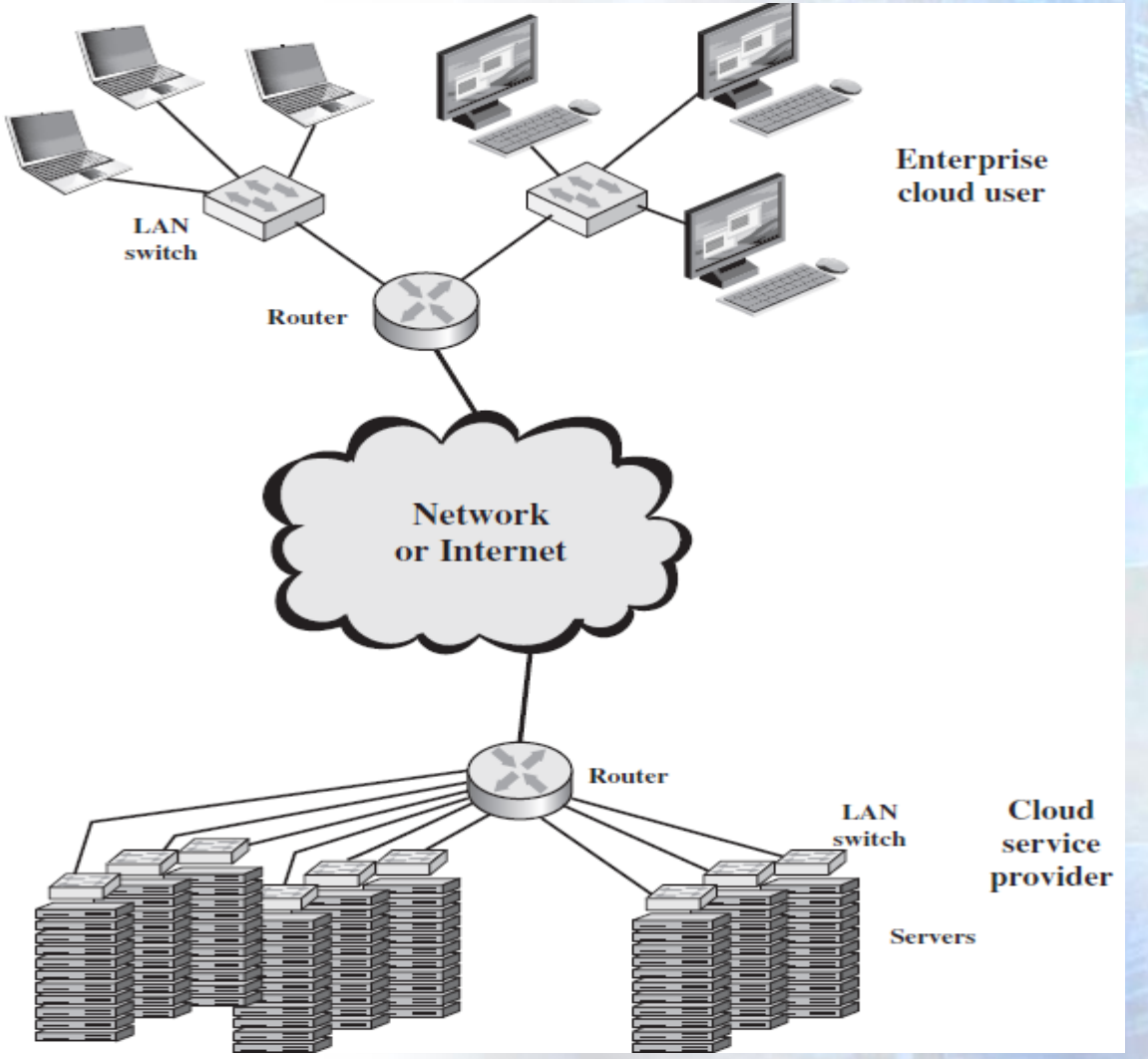
Cloud Computing and its Elements



Cloud Computing and its Elements

- Next an enterprise maintaining workstations within LAN(s), which are connected by a router through the Internet to the cloud service provider is shown.

Cloud Computing and its Elements



Cloud Computing Reference Architecture

Network Security

Cloud Computing Reference Architecture

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe the cloud computing reference architecture.

Cloud Computing Reference Architecture

Figures and material in this topic have been adapted from

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

Cloud Computing Reference Architecture

NIST SP 500-292 (NIST Cloud Computing Reference Architecture)

The NIST cloud computing reference architecture focuses on the requirements of "what" cloud services provide, not a "how to" design solution and implementation. The reference architecture is intended to facilitate the understanding of the operational intricacies in cloud computing. It does not represent the system architecture of a specific cloud computing system; instead it is a tool for describing, discussing, and developing a system-specific architecture using a common framework of reference.

Cloud Computing Reference Architecture

- NIST developed the reference architecture with the following **objectives**:
- to illustrate and understand the various cloud services in the context of an overall cloud computing conceptual model.

Cloud Computing Reference Architecture

- to provide a technical reference for consumers to understand, discuss, categorize, and compare cloud services

Cloud Computing Reference Architecture

- to facilitate the analysis of candidate standards for security, interoperability, and portability and reference implementations

Cloud Computing Reference Architecture

- The reference architecture defines **five major actors** in terms of the roles and responsibilities:
- **Cloud consumer:** A person or organization that maintains a business relationship with, and uses service from, cloud providers.

Cloud Computing Reference Architecture

- **Cloud provider:** A person, organization, or entity responsible for making a service available to interested parties.

Cloud Computing Reference Architecture

- **Cloud auditor:** A party that can conduct independent assessment of cloud services, information system operations, performance, and security of the cloud implementation.

Cloud Computing Reference Architecture

- **Cloud broker:** An entity that manages the use, performance, and delivery of cloud services, and negotiates relationships between CPs and cloud consumers.

Cloud Computing Reference Architecture

- **Cloud carrier:** An intermediary that provides connectivity and transport of cloud services from CPs to cloud consumers.

Cloud Computing Reference Architecture

- A **cloud provider** can provide one or more of the cloud services to meet IT and business requirements of **cloud consumers**.

Cloud Computing Reference Architecture

- For each of the three service models (SaaS, PaaS, IaaS), the CP provides the storage and processing facilities needed to support that service model, together with a cloud interface for cloud service consumers.

Cloud Computing Reference Architecture

- The **cloud carrier** is a networking facility that provides connectivity and transport of cloud services between cloud consumers and CPs.

Cloud Computing Reference Architecture

- A **cloud broker** is useful when cloud services are too complex for a cloud consumer to easily manage.
- Three areas of support can be offered by a cloud broker:

Cloud Computing Reference Architecture

- A **cloud broker** is useful when cloud services are too complex for a cloud consumer to easily manage.
- Three areas of support can be offered by a cloud broker:

Cloud Computing Reference Architecture

- **Service intermediation:** These are value-added services, such as identity management, performance reporting, and enhanced security.

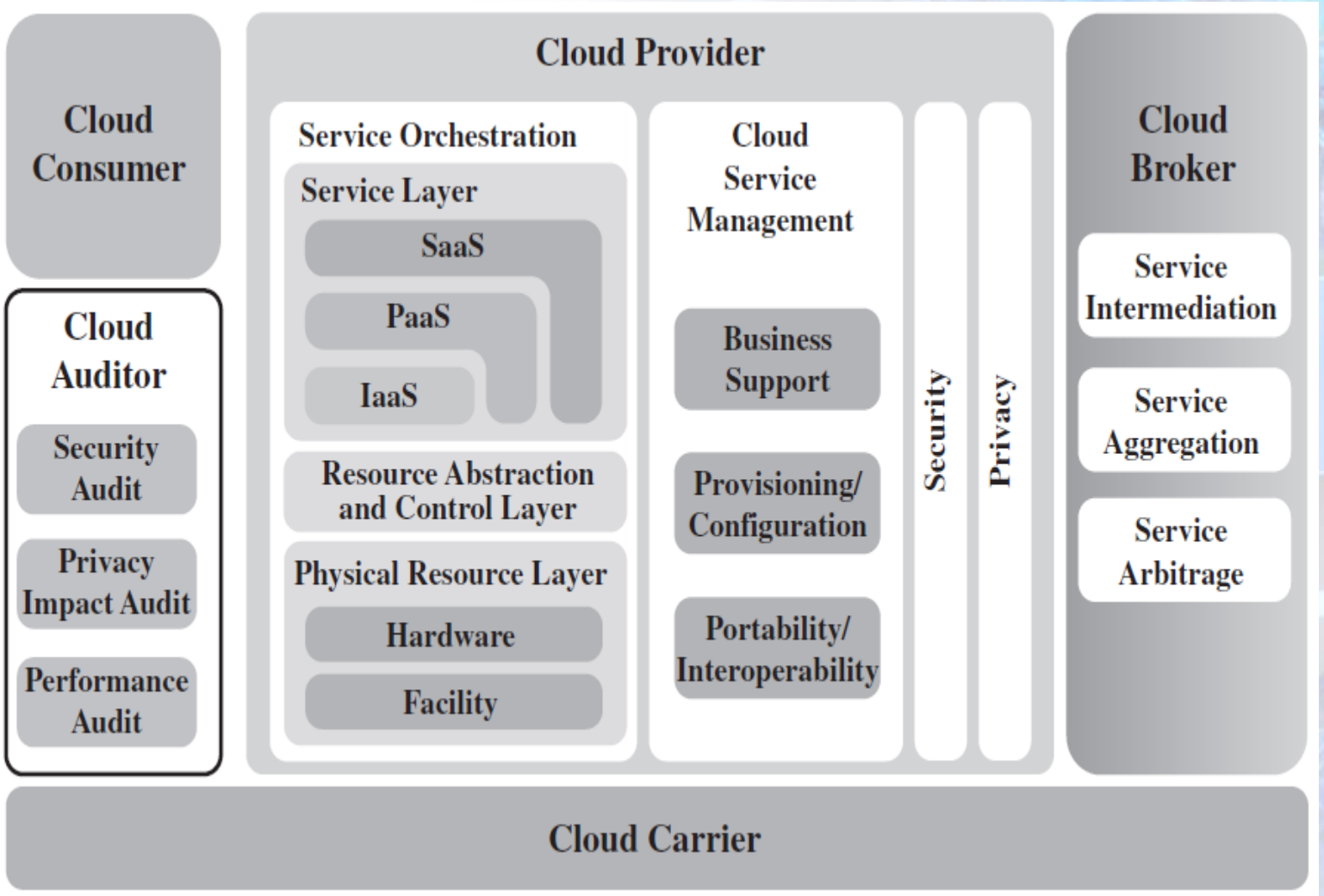
Cloud Computing Reference Architecture

- **Service aggregation:** The broker combines multiple cloud services to meet consumer needs not specifically addressed by a single CP, or to optimize performance or minimize cost.

Cloud Computing Reference Architecture

- **Service arbitrage:** means a broker has the flexibility to choose services from multiple agencies.
- A **cloud auditor** can evaluate the services provided by a CP in terms of security controls, privacy impact, performance.

Cloud Computing and its Elements



Cloud Security Risks, Countermeasures

Network Security

Cloud Security Risks, Countermeasures

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe cloud security risks and countermeasures.

Cloud Security Risks, Countermeasures

Figures and material in this topic have been adapted from

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

Cloud Security Risks, Countermeasures

- In general terms, security controls in cloud computing are similar to the security controls in any IT environment.

Cloud Security Risks, Countermeasures

- However, because of the operational models and technologies used to enable cloud service, cloud computing may present risks that are specific to the cloud environment.

Cloud Security Risks, Countermeasures

- In this regard, the essential concept is that enterprise loses a substantial amount of control over resources, services, and applications but must maintain accountability for security and privacy policies.

Cloud Security Risks, Countermeasures

- The Cloud Security Alliance lists the following as the top cloud specific security threats, together with suggested countermeasures:

Cloud Security Risks, Countermeasures

- **Abuse and nefarious use of cloud computing:**
- For many CPs, it is relatively easy to register and begin using cloud services, some even offering free limited trial periods.

Cloud Security Risks, Countermeasures

- This enables attackers to get inside the cloud to conduct various attacks, such as spamming, malicious code attacks, and denial of service.
- PaaS providers have traditionally suffered most from this kind of attacks.

Cloud Security Risks, Countermeasures

- **Countermeasures:**
- Stricter initial registration and validation processes.
- Enhanced credit card fraud monitoring and coordination.
- Comprehensive introspection of customer network traffic.

Cloud Security Risks, Countermeasures

- **Insecure interfaces and APIs:**
- CPs expose a set of software interfaces or APIs that customers use to manage and interact with cloud services.
- Protect against both accidental and malicious attempts.

Cloud Security Risks, Countermeasures

- **Countermeasures:**
- Analyze the security model of CP interfaces.
- Ensure that strong authentication and access controls are implemented in concert with encrypted transmission.

Cloud Security Risks, Countermeasures

- **Malicious insiders:**
- Under the cloud computing paradigm, an organization relinquishes direct control over many aspects of security.
- The risk of malicious insider activity.
- E.g. CP system administrators.

Cloud Security Risks, Countermeasures

- **Countermeasures:**
- Enforce strict supply chain management and conduct a comprehensive supplier assessment.
- Specify human resource requirements as part of legal contract.

Cloud Security Risks, Countermeasures

- **Shared technology issues:**
- IaaS vendors deliver their services in a scalable way by sharing infrastructure.
- CPs typically approach this risk by the use of isolated virtual machines for individual clients.

Cloud Security Risks, Countermeasures

- This approach is still vulnerable to attack, by both insiders and outsiders.

Cloud Security Risks, Countermeasures

- **Countermeasures:**
- Implement security best practices for installation/configuration.
- Monitor environment for unauthorized changes/activity.
- Promote strong authentication and access control.

Cloud Security Risks, Countermeasures

- **Data loss or leakage:**
- For many clients, the most devastating impact from a security breach is the loss or leakage of data.

Cloud Security Risks, Countermeasures

- **Countermeasures:**
- Implement strong API access control.
- Encrypt and protect integrity of data in transit.
- Analyze data protection at both design and run time.

Cloud Security Risks, Countermeasures

- **Account or service hijacking:**
- With stolen credentials, attackers can access critical areas of services.
- This compromises the confidentiality, integrity, and availability of those services.

Cloud Security Risks, Countermeasures

End

- **Countermeasures:**
- Prohibit the sharing of account credentials between users and services.
- Leverage strong two-factor authentication techniques.
- Employ proactive monitoring to detect unauthorized activity.

Data Protection In The Cloud

Network Security

Data Protection In The Cloud

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain how to protect data in the cloud.

Data Protection In The Cloud

Figures and material in this topic have been adapted from

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

Data Protection In The Cloud

- There are many ways to compromise data.
- Alteration of records without a backup of the original content.
- Unlinking a record from a larger context may render it unrecoverable as can storage on unreliable media.

Data Protection In The Cloud

- Loss of an encoding key may result in effective destruction.
- Unauthorized parties must be prevented from gaining access to sensitive data.

Data Protection In The Cloud

- Database environments used in cloud computing can vary significantly.
- **Multi-instance model**:
- provides a unique DBMS running on a virtual machine for each cloud subscriber.

Data Protection In The Cloud

- This gives the subscriber complete control over role definition, user authorization, and other administrative tasks related to security.

Data Protection In The Cloud

- **Multi-tenant model**:
- provides a predefined environment for the cloud subscriber that is shared with other tenants, typically through tagging data with a subscriber identifier.

Data Protection In The Cloud

- Tagging gives the appearance of exclusive use of the instance, but relies on the CP to establish and maintain a sound secure database environment.

Data Protection In The Cloud

- Data must be secured while at rest, in transit, and in use, and access to the data must be controlled.
- The client can employ encryption to protect data in transit, though this involves key management responsibilities for CP.

Data Protection In The Cloud

- For data at rest, the ideal security measure is for the client to encrypt the database and only store encrypted data in the cloud, with the CP having no access to the encryption key.

Data Protection In The Cloud

- A straightforward solution to the security problem in this context is to encrypt the entire database and not provide the encryption/decryption keys to the service provider .

Data Protection In The Cloud

- The user has little ability to access individual data items based on searches or indexing on key parameters.

Data Protection In The Cloud

- The user would have to download entire tables from the database, decrypt the tables, and work with the results.
- To provide more flexibility it must be possible to work with the database in its encrypted form.

Data Protection In The Cloud

- An example of such an approach can involve four entities.
- **Data owner:** An organization that produces data to be made available for controlled release, either within the organization or to external users.

Data Protection In The Cloud

- **User:** Human entity that presents requests to the system.
- After authentication, the user is granted access.
- **Client:** Frontend that transforms user queries into queries on the encrypted data stored on the server.

Data Protection In The Cloud

- **Server:** An organization that receives the encrypted data from a data owner and makes them available for distribution to clients.

Data Protection In The Cloud

- The server can be owned by the data owner.
- Typically, it is a facility owned and maintained by an external provider.
- For our discussion, the server is a cloud server.

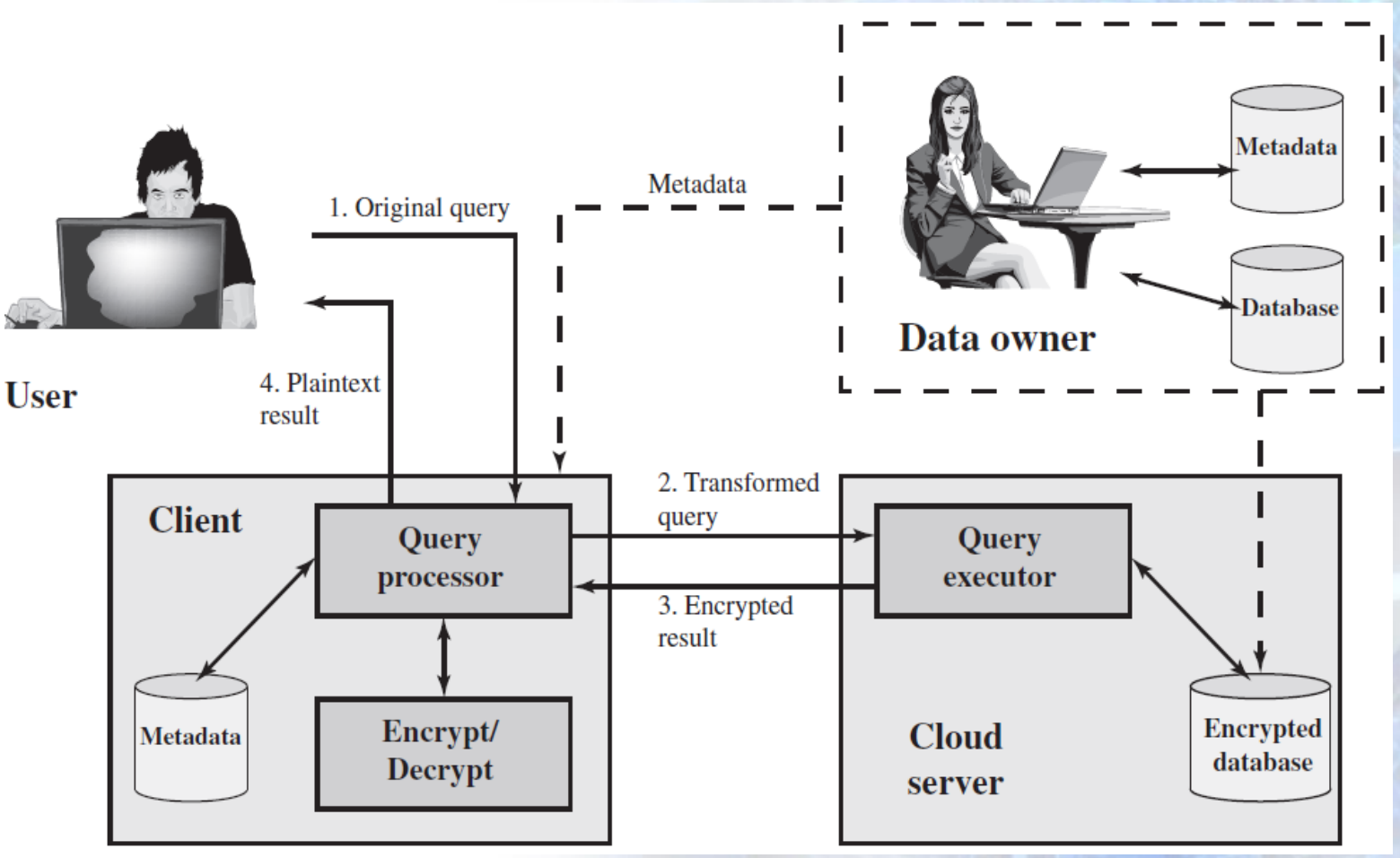
Data Protection In The Cloud

- Now, suppose that each individual item in the database is encrypted separately, all using the same encryption key.
- The encrypted database is stored at the server, but the server does not have the encryption key.

Data Protection In The Cloud

- The data are secure at the server.
- The client system does have a copy of the encryption key.
- A user at the client can retrieve a record from the database with the following sequence:

Data Protection In The Cloud



Cloud Security As A Service

Network Security

Cloud Security As A Service

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe cloud security as a service.

Cloud Security As A Service

Figures and material in this topic have been adapted from

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

Cloud Security As A Service

- The term security as a service (SecaaS) means a package of security services offered by a service provider that offloads much of the security responsibility from an enterprise to the security service provider.

Cloud Security As A Service

- Among the services, authentication, antivirus, antimalware/-spyware, intrusion detection, and security event management are provided.

Cloud Security As A Service

- In the context of cloud computing, cloud security as a service, designated SecaaS, is a segment of the SaaS offering of a CP.

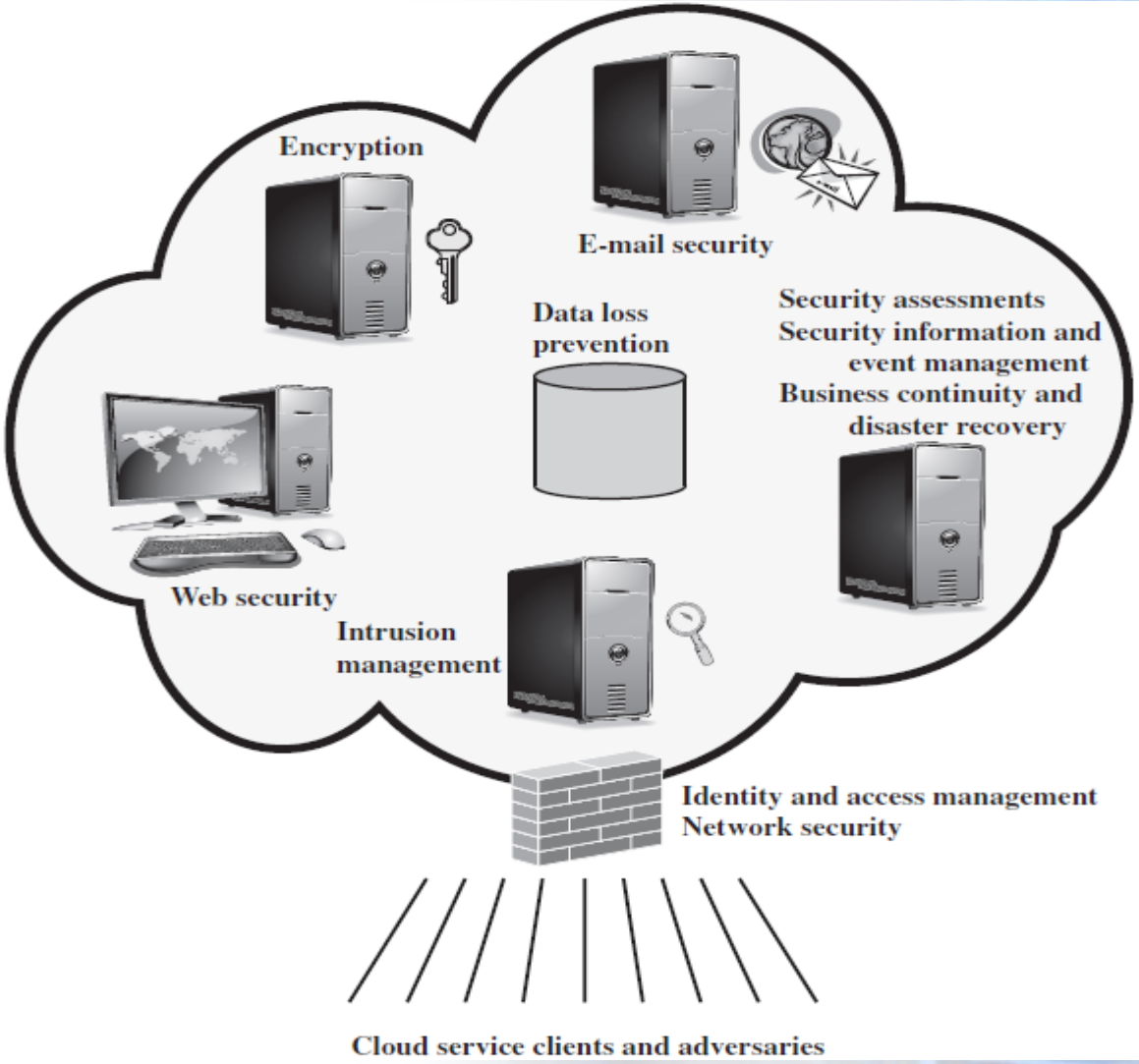
Cloud Security As A Service

- The Cloud Security Alliance defines SecaaS as the provision of security applications and services via the cloud either to cloud-based infrastructure and software or from the cloud to customers' on-premise systems.

Cloud Security As A Service

- The Cloud Security Alliance has identified the following SecaaS categories of service:

Cloud Security As A Service



Cloud Security As A Service

- **Identity and access management (IAM)** includes people, processes that are used to manage access to enterprise resources by assuring that the identity of an entity is verified, and then granting the correct level of access.

Cloud Security As A Service

- **Data loss prevention (DLP)** is the monitoring, protecting, and verifying the security of data at rest, in motion, and in use.
- Much of DLP can be implemented by the cloud client.

Cloud Security As A Service

- The cloud service provider (CSP) can also provide DLP services, such as implementing rules about what functions can be performed on data in various contexts.

Cloud Security As A Service

- **Web security** is real-time protection offered either on premise through software/ appliance installation or via the cloud by proxying or redirecting Web traffic to the CP.

Cloud Security As A Service

- **E-mail security** provides control over inbound and outbound e-mail, protecting the organization from phishing, malicious attachments, enforcing corporate polices.

Cloud Security As A Service

- **Security assessments** are third-part audits of cloud services.
- **Intrusion management** encompasses intrusion detection, prevention, and response to detect unauthorized access to a host system.

Cloud Security As A Service

- **Security information and event management (SIEM)** aggregates (via push or pull mechanisms) log and event data from virtual and real networks, applications, and systems.

Cloud Security As A Service

- **Encryption** is a pervasive service that can be provided for data at rest in the cloud, e-mail traffic, client-specific network management information, and identity information.

Cloud Security As A Service

- **Business continuity and disaster recovery** comprise measures and mechanisms to ensure operational resiliency in the event of any service interruptions.

Cloud Security As A Service

End

- **Network security** consists of security services that allocate access, distribute, monitor, and protect the underlying resource services.
- Services include perimeter and server firewalls and denial-of-service protection.

Web Security Considerations

Network Security

Web Security Considerations

Objectives of the Topic

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

Web Security Considerations

Figures and material in this topic have been adapted from

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

Web Security Considerations

- The World Wide Web is fundamentally a client/server application running over the Internet and TCP/IP intranets.
- The following characteristics of Web usage suggest the need for tailored security tools:

Web Security Considerations

- 1. Web browsers are very easy to use.
- Web servers are relatively easy to configure and manage.
- Web content is increasingly easy to develop.

Web Security Considerations

- The underlying software is extraordinarily complex.
- May hide many potential security flaws.

Web Security Considerations

- 2. A Web server can be exploited as a launching pad into the corporation's or agency's entire computer complex.
- An attacker may be able to gain access to systems not part of Web itself but connected to server.

Web Security Considerations

- 3. Casual and untrained (in security matters) users are common clients for Web based services.
- Such users are not aware of security risks that exist and do not have tools/knowledge to take effective countermeasures.

Web Security Considerations

Web Security Threats:

- A summary of the types of security threats faced when using the Web is given next.

Web Security Considerations

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

Web Security Considerations

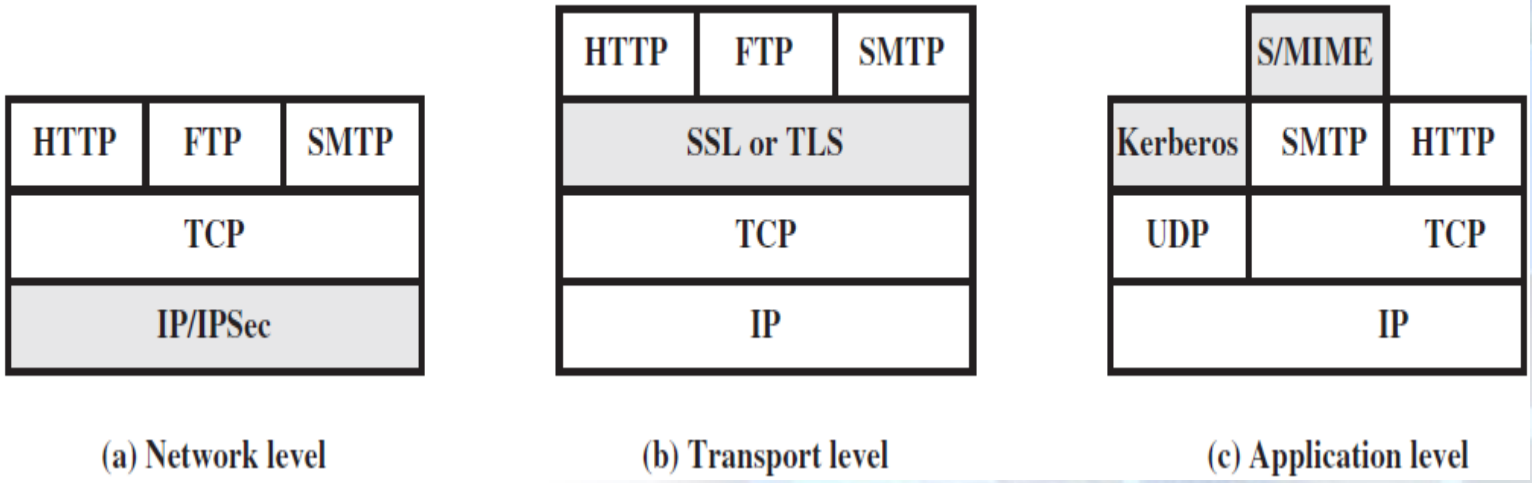
Web Traffic Security Approaches:

- A number of approaches to providing Web security are possible.

Web Security Considerations

- The various approaches differ with respect to their scope of applicability and their relative location within the TCP/IP protocol stack.

Web Security Considerations



(a) Network level

(b) Transport level

(c) Application level

Web Security Considerations

- The advantage of using IPsec is that it is transparent to end users and applications and provides a general-purpose solution.

Web Security Considerations

End

- Implementing security just above TCP provides transparency to applications.
- The advantage of application-specific security services is that the service can be tailored to the specific needs of a given application.

Secure Sockets Layer Architecture

Network Security

Secure Sockets Layer Architecture

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain the SSL architecture.

Secure Sockets Layer Architecture

Figures and material in this topic have been adapted from

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

Secure Sockets Layer Architecture

- Secure Sockets Layer (SSL) is one of the most widely used security services.
- It is a general-purpose service implemented as a set of protocols that rely on TCP.

Secure Sockets Layer Architecture

- There are two implementation choices:
- Could be provided as part of the underlying protocol suite and therefore be transparent to applications.
- Can be embedded in specific packages.

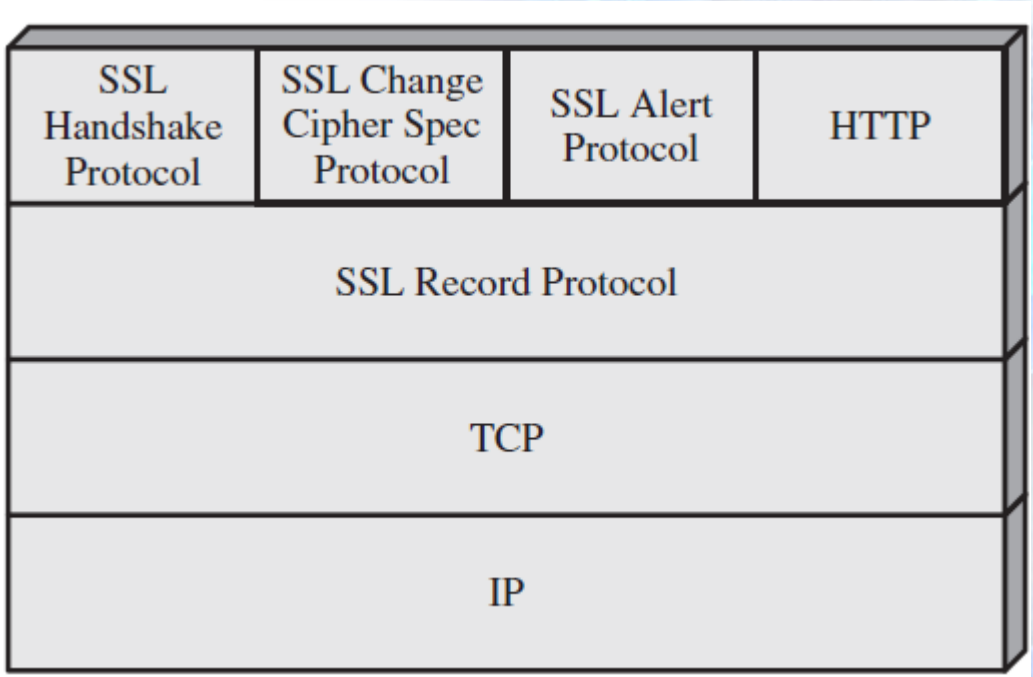
Secure Sockets Layer Architecture

SSL Architecture:

- SSL is designed to make use of TCP to provide a reliable end-to-end secure service.
- SSL is not a single protocol but rather two layers of protocols.

Secure Sockets Layer Architecture

SSL Protocol Stack



Secure Sockets Layer Architecture

- The SSL Record Protocol provides basic security services to various higher layer protocols.
- Hypertext Transfer (HTTP) which provides transfer service for Web client/server interaction, can operate on top of SSL.

Secure Sockets Layer Architecture

- Three higher-layer protocols are defined as part of SSL.
- Handshake Protocol, Change Cipher Spec Protocol, and Alert Protocol.
- These SSL-specific protocols are used in the management of SSL exchanges.

Secure Sockets Layer Architecture

- Two important SSL concepts are:
- the SSL session and,
- the SSL connection.

Secure Sockets Layer Architecture

- **Connection:** is a transport that provides a suitable type of service.
- For SSL, such connections are peer-to-peer relationships.
- Every connection is associated with one session.

Secure Sockets Layer Architecture

- **Session:** is an association between a client and a server.
- Sessions are created by the Handshake Protocol.
- Sessions define a set of cryptographic parameters which can be shared among multiple connections.

Secure Sockets Layer Architecture

- Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

Secure Sockets Layer Architecture

- Between any pair of parties, there may be multiple secure connections but multiple simultaneous sessions are not used in practice.
- There are a number of states associated with each session:

Secure Sockets Layer Architecture

- **Session identifier:** An arbitrary byte sequence chosen by the server to identify an active or resumable session state.
- **Peer certificate:** An X509.v3 certificate of the peer.

Secure Sockets Layer Architecture

- **Compression method:** The algorithm used to compress data prior to encryption.
- **Is resumable:** A flag indicating whether the session can be used to initiate new connections.

Secure Sockets Layer Architecture

- **Cipher spec:** Specifies the bulk data encryption algorithm and a hash algorithm used for MAC calculation.
- **Master secret:** 48-byte secret shared between the client and the server.

Secure Sockets Layer Architecture

- A connection state is defined by the following parameters.
- **Server and client random:** Byte sequences that are chosen by the server and client for each connection.

Secure Sockets Layer Architecture

- **Server write MAC secret:** the secret key used in MAC operations on data sent by the server.
- **Client write MAC secret:** the secret key used in MAC operations on data sent by the client.

Secure Sockets Layer Architecture

- **Server write key:** The secret encryption key for data encrypted by server and decrypted by the client.
- **Client write key:** The symmetric encryption key for data encrypted by the client and decrypted by the server.

Secure Sockets Layer Architecture

- **Initialization vectors:** When a block cipher in CBC mode is used, an initialization vector (IV) is maintained for each key.

Secure Sockets Layer Architecture

End

- **Sequence numbers:** Each party maintains separate sequence numbers for transmitted and received messages for each connection.

SSL Record Protocol

Network Security

SSL Record Protocol

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain the SSL Record Protocol.

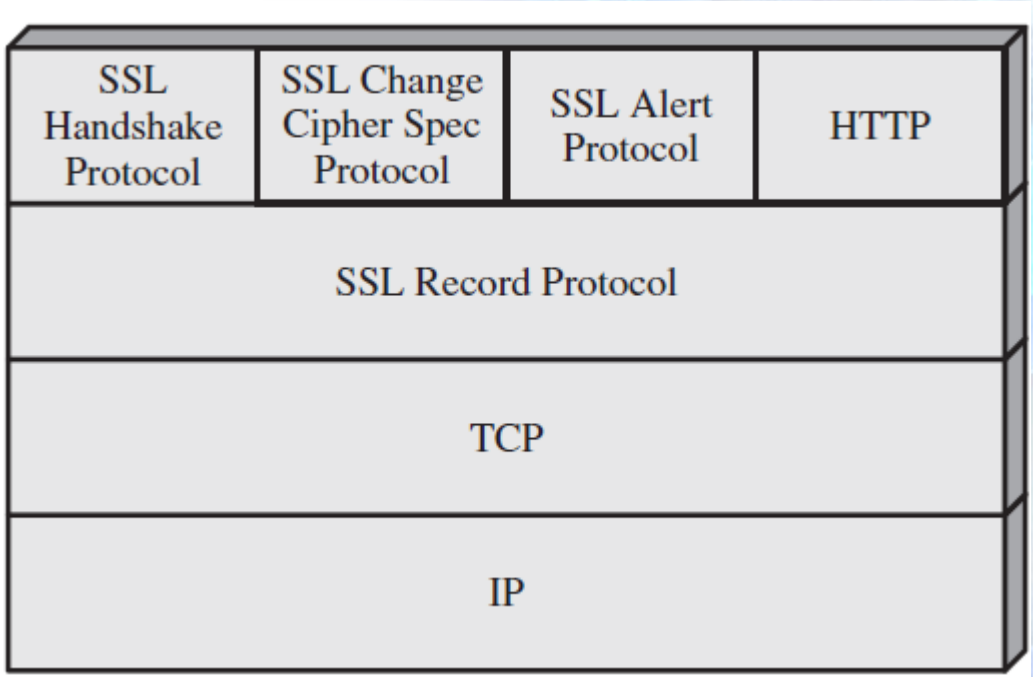
SSL Record Protocol

Figures and material in this topic have been adapted from

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

SSL Record Protocol

SSL Protocol Stack



SSL Record Protocol

- The SSL Record Protocol provides basic security services to various higher layer protocols. E.g.
- Hypertext Transfer (HTTP) which provides the transfer service for Web client/server interaction, can operate on top of SSL.

SSL Record Protocol

- The SSL Record Protocol provides two services for SSL connections:
- **Confidentiality:** The Handshake Protocol defines a shared secret key that is used for conventional encryption of SSL payloads.

SSL Record Protocol

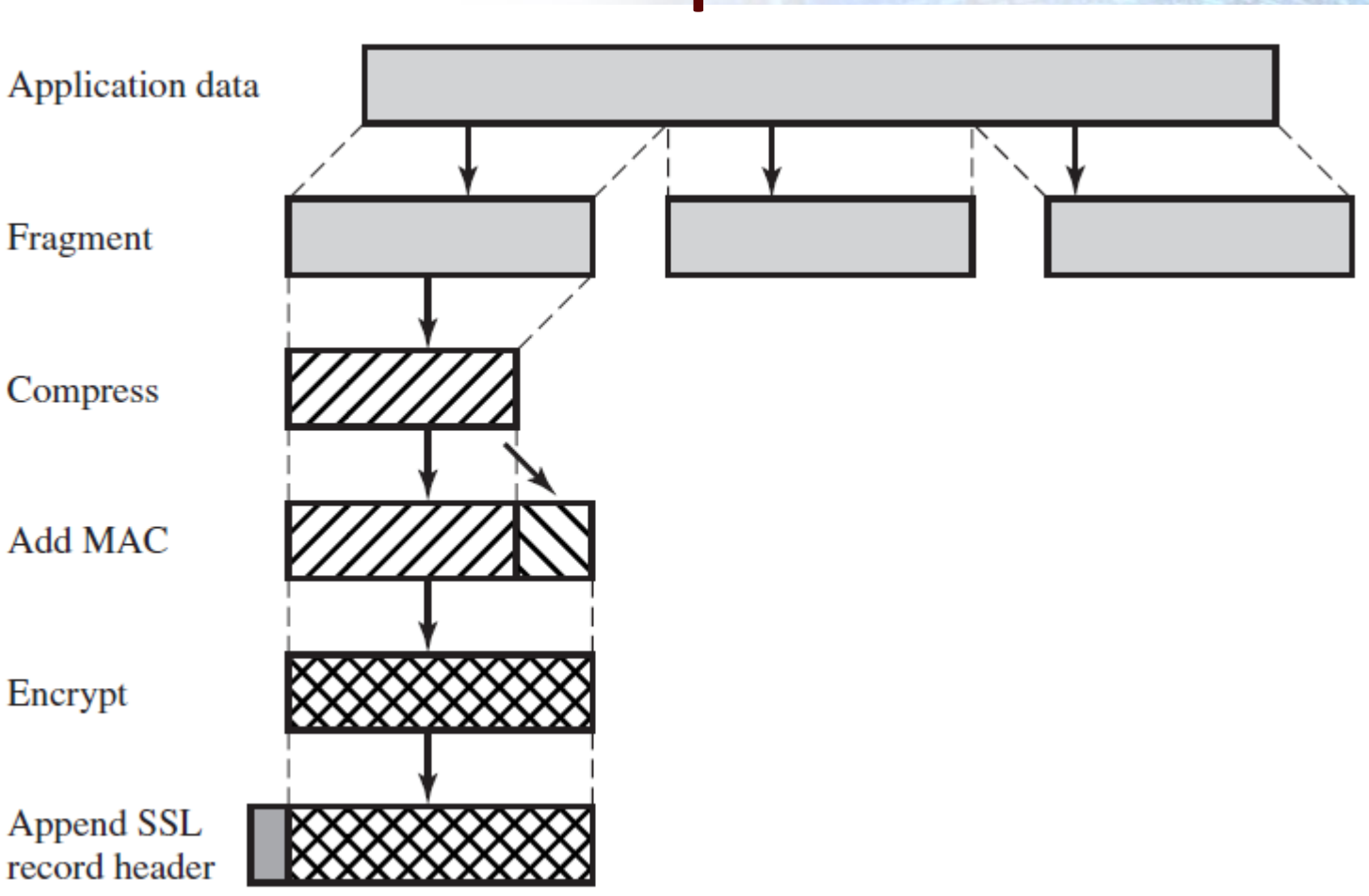
- **Message Integrity:** The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC).
- Now lets look at the SSL Record Protocol Operation:

SSL Record Protocol

- The Record Protocol takes an application message, fragments data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, adds a header, and transmits the resulting unit in a TCP segment.

SSL Record Protocol

SSL Record Protocol Operation



SSL Record Protocol

- Received data are decrypted, verified, decompressed, and reassembled before being delivered to higher-level users.

SSL Record Protocol

- In the **fragmentation** step, each upper-layer message is fragmented into blocks of 2^{14} bytes (16384 bytes) or less.
- Next, compression must be lossless and may not increase the content length by more than 1024 bytes

SSL Record Protocol

- The next step in processing is to compute a message authentication code over the compressed data.
- For this purpose, a shared secret key is used.

SSL Record Protocol

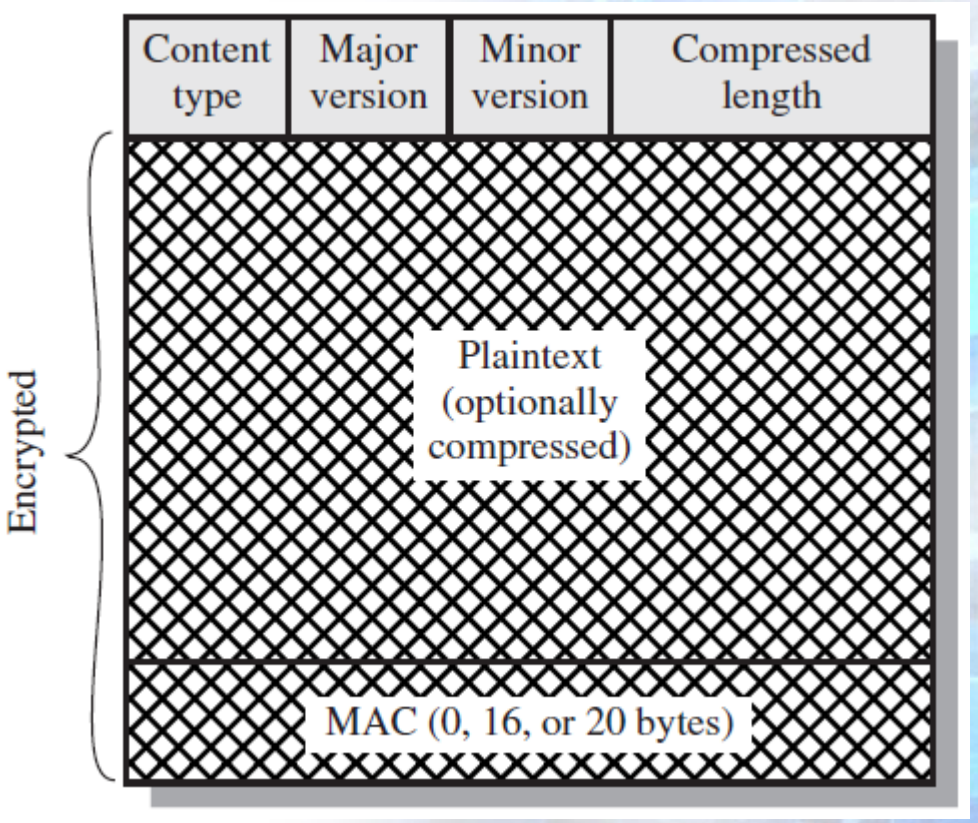
- Next, the compressed message plus the MAC are encrypted using symmetric encryption.
- It may not increase the content length by more than 1024 bytes, so that the total length may not exceed 2^{14} + 2048.

SSL Record Protocol

- The final step of SSL Record Protocol processing is to prepare a header consisting of the following fields:

SSL Record Protocol

SSL Record Format



SSL Record Protocol

- **Content Type (8 bits):** The higher-layer protocol used to process the enclosed fragment.
- **Major Version (8 bits):** Indicates major version of SSL in use. For SSLv3, the value is 3.

SSL Record Protocol

- **Minor Version (8 bits):** Indicates minor version in use. For SSLv3, the value is 0.
- **Compressed Length (16 bits):** The length in bytes of the plaintext fragment (or compressed fragment if compression is used).

SSL Record Protocol

End

- The content types that have been defined are change_cipher_spec, alert, handshake, and application_data.
- The first three are the SSL-specific protocols.

Change Cipher Spec and Alert Protocols

Network Security

Change Cipher Spec and Alert Protocols

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain the SSL's cipher spec and alert Protocols.

Change Cipher Spec and Alert Protocols

Figures and material in this topic have been adapted from

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

Change Cipher Spec and Alert Protocols

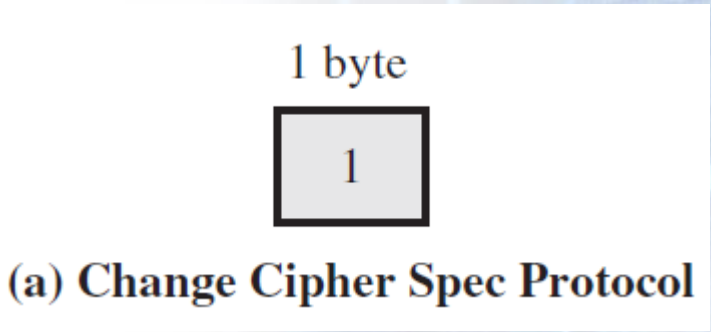
- The **Change Cipher Spec Protocol** is one of the three SSL-specific protocols that use the SSL Record Protocol, and it is the simplest.

Change Cipher Spec and Alert Protocols

- This protocol consists of a single message of a single byte with the value 1.
- Its sole purpose is to cause pending state to be copied into the current state, which updates the cipher suite to be used on this connection.

Change Cipher Spec and Alert Protocols

SSL Record Protocol Payload



(a) Change Cipher Spec Protocol

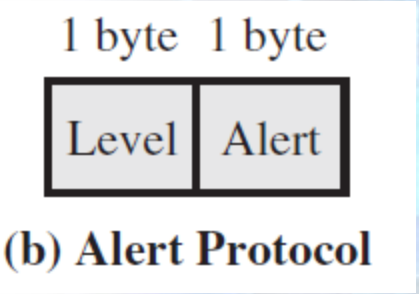
Change Cipher Spec and Alert Protocols

Alert Protocol

- is used to convey SSL-related alerts to the peer entity.
- As with other applications that use SSL, alert messages are compressed and encrypted, as specified by the current state.

Change Cipher Spec and Alert Protocols

SSL Record Protocol Payload



(b) Alert Protocol

Change Cipher Spec and Alert Protocols

- The first byte takes the value warning (1) or fatal (2) to convey the severity of the message.
- If the level is fatal, SSL immediately terminates the connection.

Change Cipher Spec and Alert Protocols

- Other connections on the same session may continue, but no new connections on this session may be established.
- The second byte contains a code that indicates the specific alert.

Change Cipher Spec and Alert Protocols

- First, we list those alerts that are always fatal (definitions from the SSL specification):
- **unexpected_message :** An inappropriate message was received.
- **bad_record_mac:** An incorrect MAC was received.

Change Cipher Spec and Alert Protocols

- **decompression_failure:** The decompression function received improper input (e.g., unable to decompress or decompress to greater than maximum allowable length).

Change Cipher Spec and Alert Protocols

- **handshake_failure:** Sender was unable to negotiate an acceptable set of security parameters given the options available.
- **illegal_parameter:** A field in a handshake message was out of range or inconsistent.

Change Cipher Spec and Alert Protocols

- Other alerts are:
- **close_notify:** Notifies the recipient that the sender will not send any more messages on this connection.
- Each party is required to send a **close_notify** alert before closing the write side of a connection.

Change Cipher Spec and Alert Protocols

- **no_certificate:** May be sent in response to a certificate request if no appropriate certificate is available.
- **bad_certificate:** A received certificate was corrupt (e.g., contained a signature that did not verify).

Change Cipher Spec and Alert Protocols

- **unsupported_certificate:** The type of the received certificate is not supported.
- **certificate_revoked:** A certificate has been revoked by its signer.
- **certificate_expired:** A certificate has expired.

Change Cipher Spec and Alert Protocols

End

- **certificate_unknown:** Some other unspecified issue arose in processing the certificate, rendering it unacceptable.

Phase 1 of SSL Handshake Protocol

Network Security

Phase 1 of SSL Handshake Protocol

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain phase 1 of SSL handshake protocol.

Phase 1 of SSL Handshake Protocol

Figures and material in this topic have been adapted from

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

Phase 1 of SSL Handshake Protocol

- The most complex part of SSL is the Handshake Protocol.
- It is used before any application data is transmitted.

Phase 1 of SSL Handshake Protocol

- This protocol allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in an SSL record.

Phase 1 of SSL Handshake Protocol

- The Handshake Protocol consists of a series of messages exchanged by client and server.
- Each message has three fields:

Phase 1 of SSL Handshake Protocol

SSL Record Protocol Payload



(c) Handshake Protocol

Phase 1 of SSL Handshake Protocol

- **Type (1 byte):** Indicates one of ten messages.
- **Length (3 bytes):** The length of the message in bytes.
- **Content (≥ 0 bytes):** The parameters associated with this message.

Phase 1 of SSL Handshake Protocol

SSL Handshake Protocol Message Types

Message Type	Parameters
hello_request	null
client_hello	version, random, session id, cipher suite, compression method
server_hello	version, random, session id, cipher suite, compression method
certificate	chain of X.509v3 certificates
server_key_exchange	parameters, signature
certificate_request	type, authorities
server_done	null
certificate_verify	signature
client_key_exchange	parameters, signature
finished	hash value

Phase 1 of SSL Handshake Protocol

- **Phase 1. Establish Security Capabilities:**
- initiates a logical connection and establishes the security capabilities that will be associated with it.

Phase 1 of SSL Handshake Protocol

- The exchange is initiated by the client, which sends a **client_hello** message with the following parameters:
- **Version:** The highest SSL version understood by the client.

Phase 1 of SSL Handshake Protocol

- **Random:** A client-generated random structure consisting of a 32-bit timestamp and 28 bytes.
- These values serve as nonces and are used during key exchange to prevent replay attacks.

Phase 1 of SSL Handshake Protocol

- **Session ID:** A nonzero value indicates that client wants to update parameters of an existing connection or to create a new connection on this session.
- A zero value indicates a new connection on a new session.

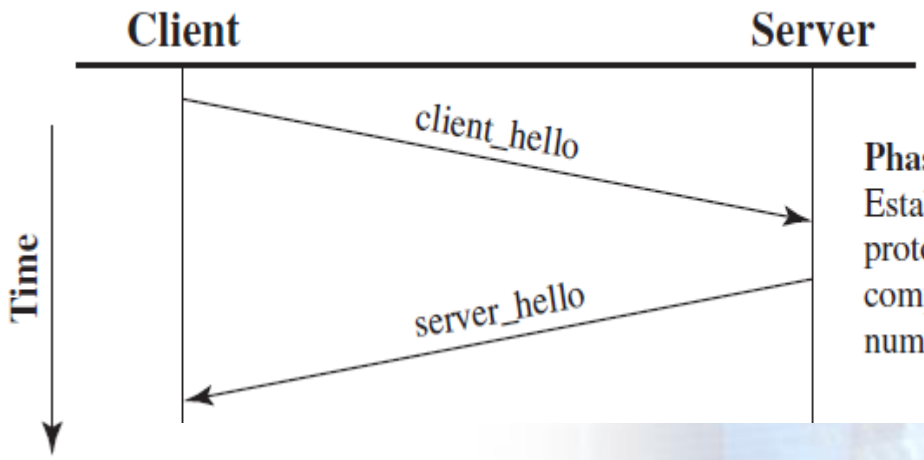
Phase 1 of SSL Handshake Protocol

- **CipherSuite:** This is a list that contains the combinations of cryptographic algorithms supported by the client, in decreasing order of preference.

Phase 1 of SSL Handshake Protocol

- **Compression Method:** This is a list of the compression methods the client supports.

Phase 1 of SSL Handshake Protocol



Phase 1
Establish security capabilities, including protocol version, session ID, cipher suite, compression method, and initial random numbers.

Phase 1 of SSL Handshake Protocol

- The **server_hello** message contains the same parameters as the **client_hello** message.

Phase 1 of SSL Handshake Protocol

- The **Version** field contains the lower of versions suggested by the client and the highest supported by the server.
- The **Random** field is generated by the server and is independent of the client's **Random** field.

Phase 1 of SSL Handshake Protocol

- If the **SessionID** field of the client was nonzero, the same value is used by the server.
- Otherwise the server's **SessionID** field contains the value for a new session.

Phase 1 of SSL Handshake Protocol

End

- The **CipherSuite** field contains the single cipher suite selected by the server from those proposed by the client.
- The **Compression** field contains compression method selected by the server from those proposed by client.

Phase 2 of SSL Handshake Protocol

Network Security

Phase 2 of SSL Handshake Protocol

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain phase 2 of SSL handshake protocol.

Phase 2 of SSL Handshake Protocol

Figures and material in this topic have been adapted from

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

Phase 2 of SSL Handshake Protocol

- Following are key exchange algorithms supported:
- **RSA**
- **Ephemeral Diffie-Hellman:** This technique is used to create ephemeral (temporary, one-time) secret keys.

Phase 2 of SSL Handshake Protocol

- **Fixed Diffie-Hellman:** This is a Diffie-Hellman key exchange in which the server's certificate contains the Diffie-Hellman public parameters signed by the certificate authority (CA).

Phase 2 of SSL Handshake Protocol

- **Anonymous Diffie-Hellman:** The base Diffie-Hellman algorithm is used with no authentication.
- **Fortezza:** The technique defined for the Fortezza scheme.

Phase 2 of SSL Handshake Protocol

- **Phase 2. Server Authentication and Key Exchange:**
- The server begins this phase by sending its certificate if it needs to be authenticated.

Phase 2 of SSL Handshake Protocol

- The message contains one or a chain of X.509 certificates.
- The **certificate** message is required for any agreed-on key exchange method except anonymous Diffie-Hellman.

Phase 2 of SSL Handshake Protocol

- If fixed Diffie-Hellman is used, this certificate message functions as the server's key exchange message because it contains the server's public Diffie-Hellman parameters.

Phase 2 of SSL Handshake Protocol

- Next, a **server_key_exchange** message may be sent if it is required.
- It is not required in two instances: (1) the server has sent a certificate with fixed Diffie- Hellman or (2) a RSA key exchange is to be used.

Phase 2 of SSL Handshake Protocol

- It is needed for:
- **Anonymous Diffie-Hellman:** The message content consists of two global Diffie-Hellman values (a prime number and a primitive root of that number) plus the server's public Diffie-Hellman key.

Phase 2 of SSL Handshake Protocol

- **Ephemeral Diffie-Hellman:** The message content includes the three Diffie-Hellman parameters provided for anonymous Diffie-Hellman plus a signature of those parameters.

Phase 2 of SSL Handshake Protocol

- **RSA key exchange (in which the server is using RSA but has a signature-only RSA key):** The client cannot simply send a secret key encrypted with the server's public key.

Phase 2 of SSL Handshake Protocol

- Instead, the server must create a temporary RSA public/private key pair and use the **server_key_exchange** message to send the public key.
- **Fortezza**

Phase 2 of SSL Handshake Protocol

- Next, a nonanonymous server (server not using anonymous Diffie-Hellman) can request a certificate from the client.
- The **certificate_request** message includes two parameters:

Phase 2 of SSL Handshake Protocol

- certificate_type and certificate_authorities
- The certificate type indicates the public-key algorithm and its use:
- RSA, signature only
- DSS, signature only
- RSA for fixed Diffie-Hellman, used only for authentication.

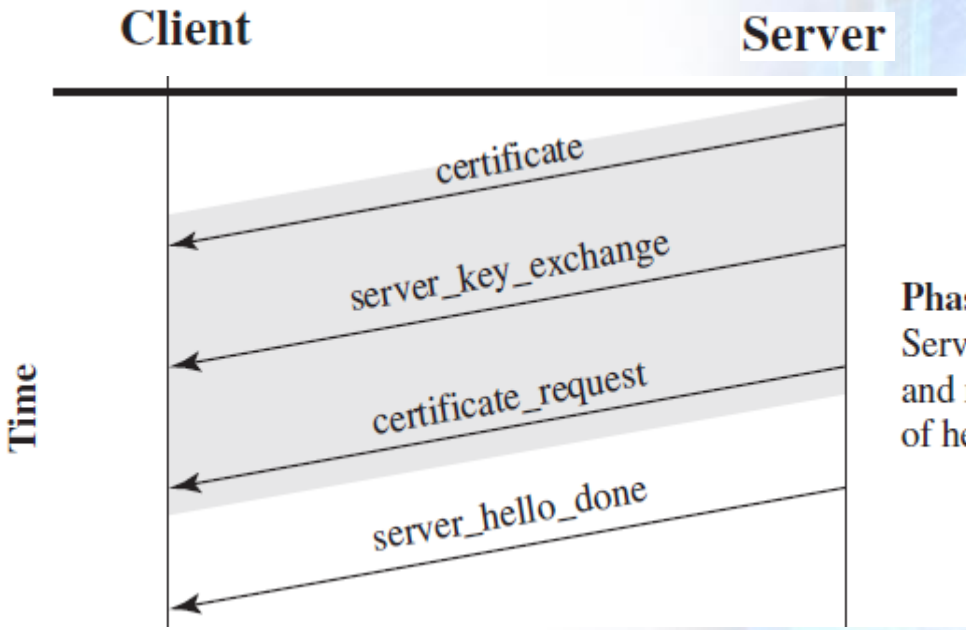
Phase 2 of SSL Handshake Protocol

- DSS for fixed Diffie-Hellman, used only for authentication
- RSA for ephemeral Diffie-Hellman
- DSS for ephemeral Diffie-Hellman
- Fortezza

Phase 2 of SSL Handshake Protocol

- The second parameter in the certificate_request message is a list of the distinguished names of acceptable certificate authorities.

Phase 2 of SSL Handshake Protocol



Phase 2
Server may send certificate, key exchange, and request certificate. Server signals end of hello message phase.

Note: Shaded transfers are optional or situation-dependent messages that are not always sent.

Phase 2 of SSL Handshake Protocol

End

- The final message in phase 2, and one that is always required, is the **server_done** message, which is sent by the server to indicate the end of the server hello and associated messages.
- This message has no parameters.

Phases 3 & 4 of SSL Handshake Protocol

Network Security

Phases 3 & 4 of SSL Handshake Protocol

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain phases 3 & 4 of SSL handshake protocol.

Phases 3 & 4 of SSL Handshake Protocol

Figures and material in this topic have been adapted from

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

Phases 3 & 4 of SSL Handshake Protocol

- **Phase 3. Client Authentication and Key Exchange:**
- Upon receipt of **server_done**, client verifies that the server provided a valid certificate (if required) and **server_hello** parameters are acceptable.

Phases 3 & 4 of SSL Handshake Protocol

- If all is satisfactory, the client sends one or more messages back to the server.

Phases 3 & 4 of SSL Handshake Protocol

- If the server has requested a certificate, the client begins this phase by sending a **certificate message**.
- If no suitable certificate is available, the client sends a **no_certificate** alert instead.

Phases 3 & 4 of SSL Handshake Protocol

- Next is the **client_key_exchange message**, which must be sent in this phase.
- The content of the message depends on the type of key exchange, as follows.

Phases 3 & 4 of SSL Handshake Protocol

- **RSA:** The client generates a 48-byte *pre-master secret* and encrypts with the public key from the server's certificate or temporary RSA key from a **server_key_exchange** message.

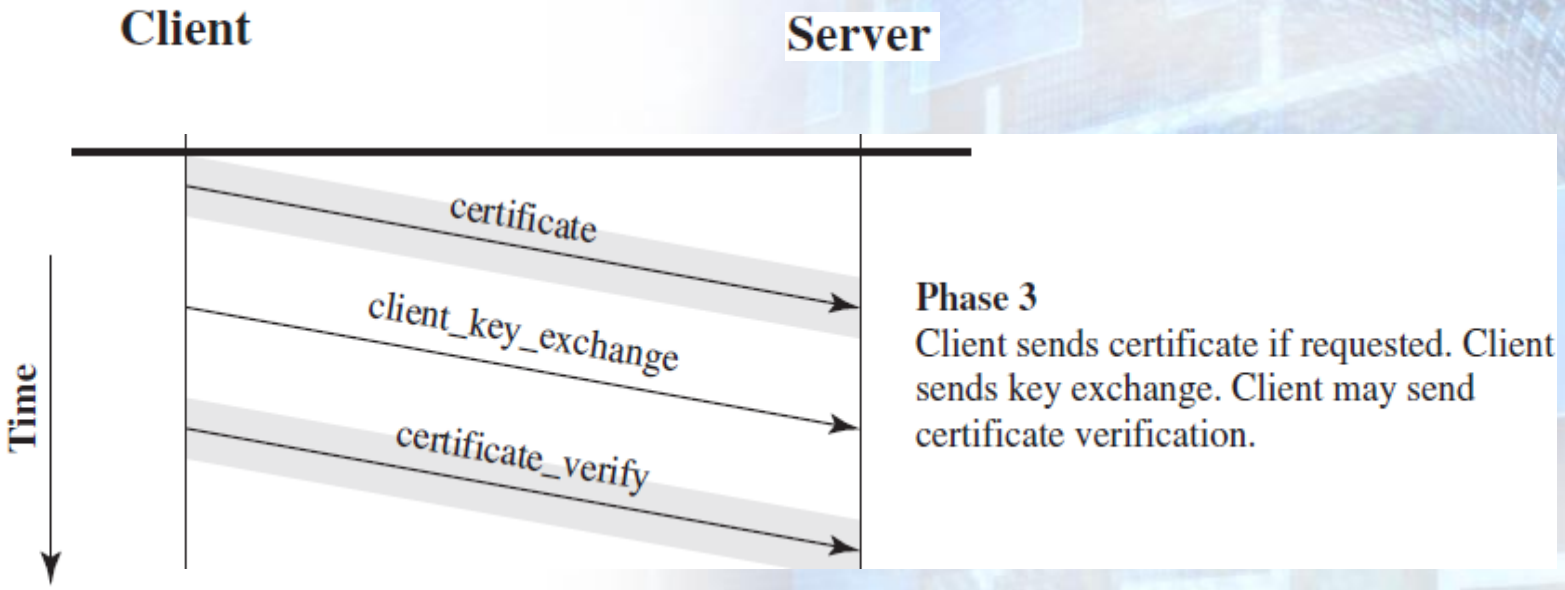
Phases 3 & 4 of SSL Handshake Protocol

- **Ephemeral or Anonymous Diffie-Hellman:** Client's public Diffie-Hellman parameters are sent.
- **Fixed Diffie-Hellman:** Client's public Diffie-Hellman parameters were sent in a certificate, so content of message is null.

Phases 3 & 4 of SSL Handshake Protocol

- **Fortezza:** The client's Fortezza parameters are sent.
- Finally, in this phase, the client may send a **certificate_verify** message to provide explicit verification of a client certificate.

Phases 3 & 4 of SSL Handshake Protocol



Note: Shaded transfers are optional or situation-dependent messages that are not always sent.

Phases 3 & 4 of SSL Handshake Protocol

- **Phase 4. Finish**
- This phase completes the setting up of a secure connection.
- The client sends a **change_cipher_spec** message and copies the pending CipherSpec into the current CipherSpec.

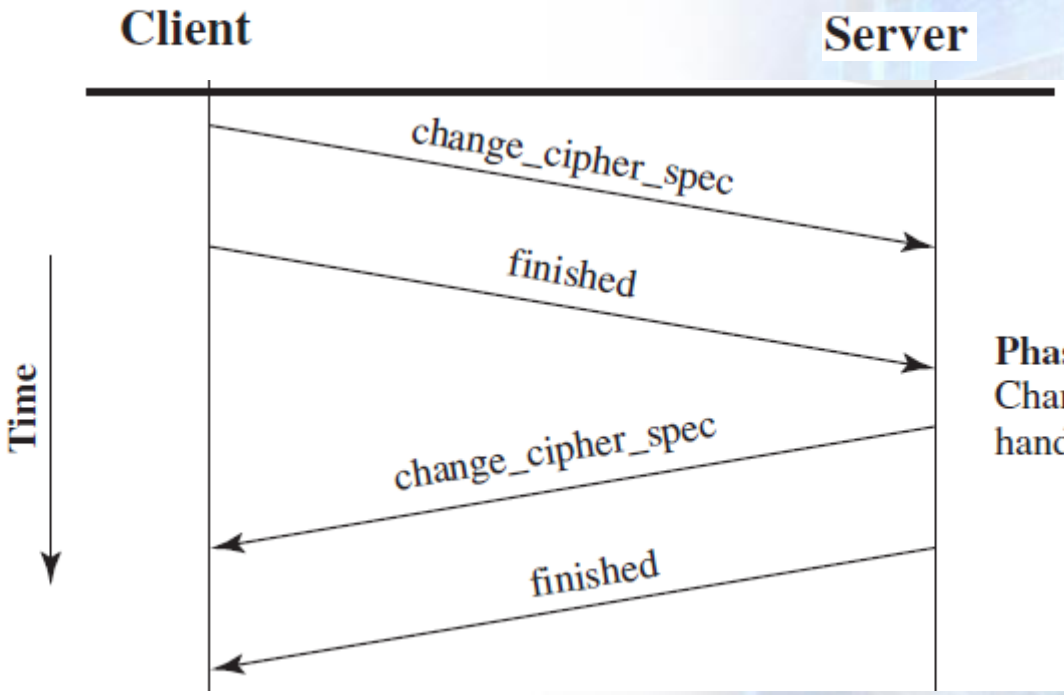
Phases 3 & 4 of SSL Handshake Protocol

- The client then immediately sends the **finished** message under the new algorithms, keys, and secrets.
- The finished message verifies key exchange and authentication processes were successful.

Phases 3 & 4 of SSL Handshake Protocol

- In response to these two messages, the server sends its own **change_ cipher_spec** message, transfers the pending to the current CipherSpec, and sends its finished message.

Phases 3 & 4 of SSL Handshake Protocol



Phase 4
Change cipher suite and finish handshake protocol.

Phases 3 & 4 of SSL Handshake Protocol

- At this point, the handshake is complete and the client and server may begin to exchange application-layer data.

End

TLS Authentication Code



TLS Authentication Code

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe some of the differences between Transport Layer Security and Secure Sockets Layer.

TLS Authentication Code

Figures and material in this topic have been adapted from

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

TLS Authentication Code

- Transport Layer Security (TLS) is an IETF standardization initiative whose goal is to produce an Internet standard version of SSL.
- TLS is defined in RFC 5246.
- RFC 5246 is very similar to SSLv3.

TLS Authentication Code

Differences between SSL and TLS:

- Version number
- Message Authentication Code
- Pseudorandom function
- Alert keys
- Cipher suites
- Client certificate types

TLS Authentication Code

- Certificate_verify and Finished Messages
- Cryptographic computations
- Padding

TLS Authentication Code

- **Version Number:**
- The TLS Record Format is the same as that of the SSL Record Format, and the fields in the header have the same meanings.

TLS Authentication Code

- The one difference is in version values.
- For the current version of TLS, the major version is 3 and the minor version is 3.

TLS Authentication Code

- **Message Authentication Code:**
- There are two differences between the SSLv3 and TLS MAC schemes: the actual algorithm and the scope of the MAC calculation.

TLS Authentication Code

- TLS makes use of the HMAC algorithm defined in RFC 2104.
- HMAC is defined as

TLS Authentication Code

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

where

H = embedded hash function (for TLS, either MD5 or SHA-1)

M = message input to HMAC

K^+ = secret key padded with zeros on the left so that the result is equal to the block length of the hash code (for MD5 and SHA-1, block length = 512 bits)

ipad = 00110110 (36 in hexadecimal) repeated 64 times (512 bits)

opad = 01011100 (5C in hexadecimal) repeated 64 times (512 bits)

TLS Authentication Code

- SSLv3 uses the same algorithm, except that the padding bytes are concatenated with the secret key rather than being XORed with the secret key padded to the block length.
- The level of security should be about the same in both cases.

TLS Authentication Code

- For TLS, the MAC calculation covers all of the fields covered by the SSLv3 calculation, plus the field TLSCompressed.version, which is the version of the protocol being employed.

TLS Authentication Code

- **Pseudorandom Function:**
- TLS makes use of a pseudorandom function referred to as PRF to expand secrets into blocks of data for purposes of key generation or validation.

TLS Authentication Code

- The objective is to make use of a relatively small shared secret value but to generate longer blocks of data in a way that is secure from the kinds of attacks made on hash functions and MACs.

TLS Authentication Code

- The PRF is based on the data expansion function given as:

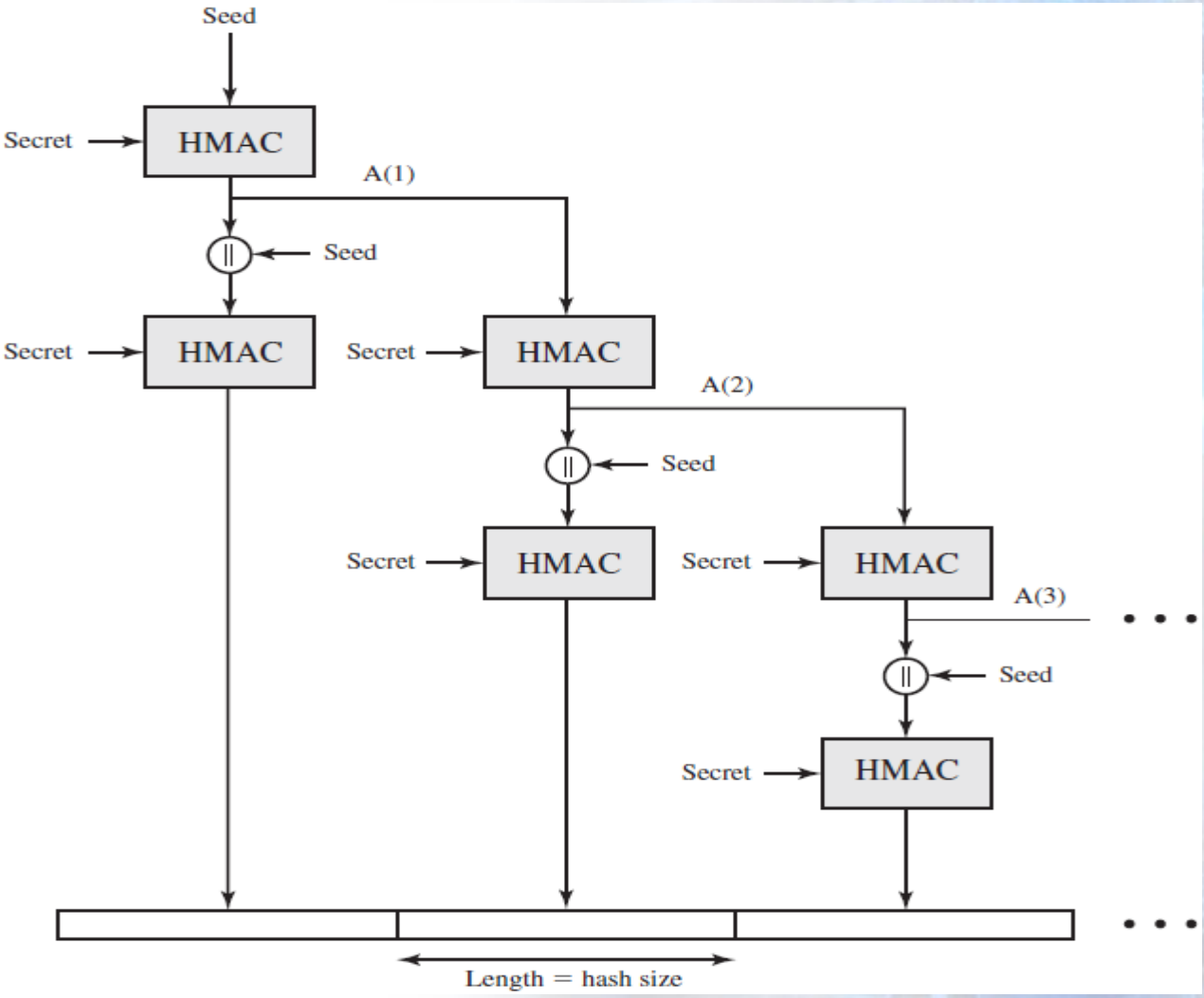
TLS Authentication Code

$$
\begin{aligned}
\texttt{P_hash(secret, seed)} = {} & \texttt{HMAC_hash(secret,A(1)} \parallel \texttt{seed)} \parallel \\
& \texttt{HMAC_hash(secret,A(2)} \parallel \texttt{seed)} \parallel \\
& \texttt{HMAC_hash(secret,A(3)} \parallel \texttt{seed)} \parallel \ldots
\end{aligned}
$$

where A() is defined as

$$
\begin{aligned}
\texttt{A(0)} &= \text{seed} \\
\texttt{A(}i\texttt{)} &= \text{HMAC_hash(secret, A(}i-1\text{))}
\end{aligned}
$$

TLS Authentication Code



TLS Authentication Code

End

- PRF takes as input a secret value, an identifying label, and a seed value and produces an output of arbitrary length.
- The P_hash can be iterated as many times as necessary to produce the required quantity of data.

Alert Codes and Cipher suites of TLS

Network Security

Alert Codes and Cipher suites of TLS

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe some more differences between Transport Layer Security and Secure Sockets Layer.

Alert Codes and Cipher suites of TLS

Figures and material in this topic have been adapted from

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

Alert Codes and Cipher suites of TLS

Differences between SSL and TLS:

- Version number
- Message Authentication Code
- Pseudorandom function
- Alert keys
- Cipher suites
- Client certificate types

Alert Codes and Cipher suites of TLS

- Certificate_verify and Finished Messages
- Cryptographic computations
- Padding

Alert Codes and Cipher suites of TLS

- **Alert Codes:**
- TLS supports all of the alert codes defined in SSLv3 with the exception of no_certificate.
- A number of additional codes are defined in TLS; of these, following are always fatal.

Alert Codes and Cipher suites of TLS

- **record_overflow:** A TLS record was received with a payload whose length exceeds 2^{14} + 2048 bytes.

Alert Codes and Cipher suites of TLS

- **unknown_ca:** A valid certificate chain was received, but the certificate was not accepted because the CA certificate could not be located or could not be matched with a known, trusted CA.

Alert Codes and Cipher suites of TLS

- **access_denied:** A valid certificate was received, but when access control was applied, the sender decided not to proceed with the negotiation.

Alert Codes and Cipher suites of TLS

- **decode_error:** A message could not be decoded. May be a field was out of its specified range.
- **protocol_version:** The protocol version the client attempted to negotiate is recognized but not supported.

Alert Codes and Cipher suites of TLS

- **insufficient_security:** the server requires ciphers more secure than those supported by the client.

Alert Codes and Cipher suites of TLS

- **unsupported_extension:** Sent by clients that receive an extended server hello containing an extension not in the corresponding client hello.

Alert Codes and Cipher suites of TLS

- **internal_error:** An internal error.
- **decrypt_error:** A handshake cryptographic operation failed, including being unable to verify a signature, decrypt a key exchange, or validate a finished message.

Alert Codes and Cipher suites of TLS

- **Cipher Suites**
- There are several small differences between the cipher suites available under SSLv3 and under TLS:
- **Key Exchange:** TLS supports all of the key exchange techniques of SSLv3 with the exception of Fortezza.

Alert Codes and Cipher suites of TLS

- **Symmetric Encryption Algorithms:** TLS includes all of the symmetric encryption algorithms found in SSLv3, with the exception of Fortezza.

Alert Codes and Cipher suites of TLS

- **Client Certificate Types**
- TLS defines the following certificate types to be requested in a certificate_request message: rsa_sign, dss_sign, rsa_fixed_dh, and dss_fixed_dh.

Alert Codes and Cipher suites of TLS

- These are all defined in SSLv3. In addition, SSLv3 includes rsa_ephemeral_dh, dss_ephemeral_dh, and fortezza_kea.
- Ephemeral Diffie-Hellman involves signing the Diffie-Hellman parameters with either RSA/DSS.

Alert Codes and Cipher suites of TLS

End

- For TLS, the rsa_sign and dss_sign types are used for that function; a separate signing type is not needed to sign Diffie-Hellman parameters.
- TLS does not include the Fortezza scheme.

HTTPS

Network Security

HTTPS

Objectives of the Topic

- After completing this topic, a student will be able to
  - Explain working of HTTPS.

HTTPS

Figures and material in this topic have been adapted from

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

HTTPS

- HTTPS (HTTP over SSL) refers to the combination of HTTP and SSL to implement secure communication between a Web browser and a Web server.

HTTPS

- The HTTPS capability is built into all modern Web browsers.
- Its use depends on the Web server supporting HTTPS communication.

HTTPS

- For example, some search engines do not support HTTPS.
- Google provides HTTPS as an option: https://google.com.

HTTPS

- The principal difference seen by a user of a Web browser is that URL (uniform resource locator) addresses begin with https:// rather than http://.

HTTPS

- A normal HTTP connection uses port 80.
- If HTTPS is specified, port 443 is used, which invokes SSL.

HTTPS

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

HTTPS

- When HTTPS is used, the following elements of the communication are encrypted:
- URL of the requested document
- Contents of the document

HTTPS

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

HTTPS

Connection Initiation:

- The agent acting as the HTTP client also acts as the TLS client.
- The client initiates a connection to the server on the appropriate port and then sends the TLS ClientHello to begin the TLS handshake.

HTTPS

- When TLS handshake has finished, the client may then initiate the first HTTP request.
- All HTTP data is to be sent as TLS application data.
- There are three levels of awareness of a connection in HTTPS.

HTTPS

- At the HTTP level, an HTTP client requests a connection to an HTTP server by sending a connection request to the next lowest layer.
- Typically, the next lowest layer is TCP, but it also may be TLS/SSL.

HTTPS

- At the level of TLS, a session is established between a TLS client and a TLS server.
- This session can support one or more connections at any time.

HTTPS

- A TLS request to establish a connection begins with the establishment of a TCP connection between the TCP entity on the client side and the TCP entity on the server side.

HTTPS

- **Connection Closure:**
- An HTTP client or server can indicate the closing of a connection by including the line **`Connection: close`** in an HTTP record

HTTPS

- The closure of an HTTPS connection requires that TLS close the connection with the peer TLS entity on the remote side, which will involve closing the underlying TCP connection.

HTTPS

- TLS implementations must initiate an exchange of closure alerts before closing a connection.

HTTPS

- A TLS implementation may, after sending a closure alert, close the connection without waiting for the peer to send its closure alert, generating an “incomplete close”.

HTTPS

- An unannounced TCP closure could be evidence of some sort of attack so the HTTPS client should issue some sort of security warning when this occurs.

End

Introduction to Secure Shell (SSH)

Network Security

Introduction to Secure Shell (SSH)

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe basics of Secure Shell (SSH).

Introduction to Secure Shell (SSH)

Figures and material in this topic have been adapted from

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

Introduction to Secure Shell (SSH)

- Secure Shell (SSH) is a protocol for secure network communications designed to be relatively simple and inexpensive to implement.

Introduction to Secure Shell (SSH)

- The initial version, SSH1 was focused on providing a secure remote logon facility to replace TELNET and other remote logon schemes that provided no security.

Introduction to Secure Shell (SSH)

- SSH also provides a more general client/server capability and can be used for such network functions as file transfer and e-mail.

Introduction to Secure Shell (SSH)

- A new version, SSH2, fixes a number of security flaws in the original scheme.
- SSH2 is documented as a proposed standard in IETF RFCs 4250 through 4256.

Introduction to Secure Shell (SSH)

- SSH client and server applications are widely available for most operating systems.

Introduction to Secure Shell (SSH)

- It has become the method of choice for remote login and X tunneling and is rapidly becoming one of the most pervasive applications for encryption technology outside of embedded systems.

Introduction to Secure Shell (SSH)

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

Introduction to Secure Shell (SSH)

- **Transport Layer Protocol:** Provides server authentication, data confidentiality, and data integrity with forward secrecy (i.e., if a key is compromised during one session, the knowledge does not affect the security of earlier sessions).

Introduction to Secure Shell (SSH)

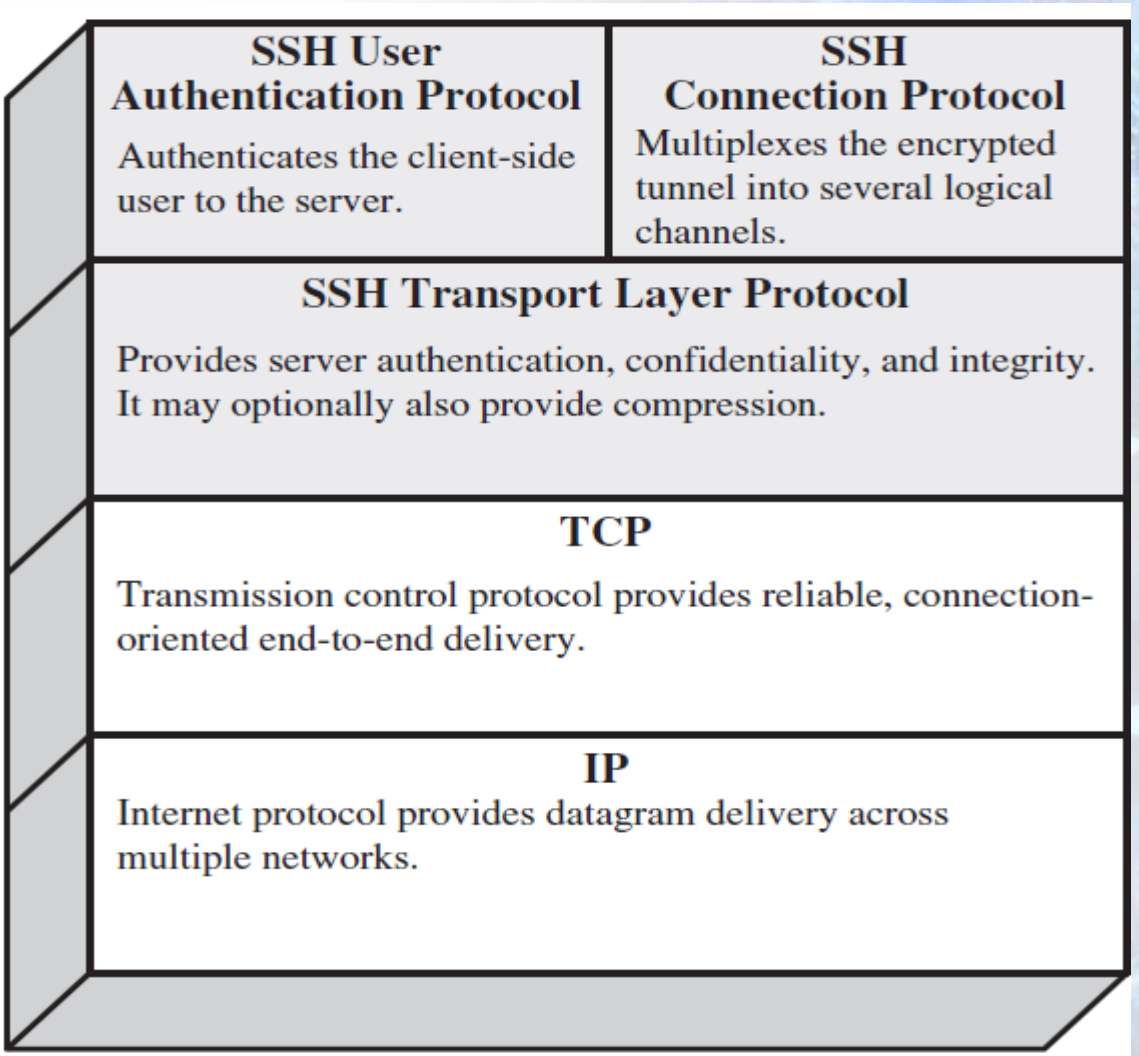
- The transport layer may optionally provide compression.

Introduction to Secure Shell (SSH)

- **User Authentication Protocol:** Authenticates the user to the server.
- **Connection Protocol:** Multiplexes multiple logical communications channels over a single, underlying SSH connection.

Introduction to Secure Shell (SSH)

SSH Protocol Stack



SSH Transport Layer Protocol

Network Security

SSH Transport Layer Protocol

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe working of the SSH Transport Layer Protocol.

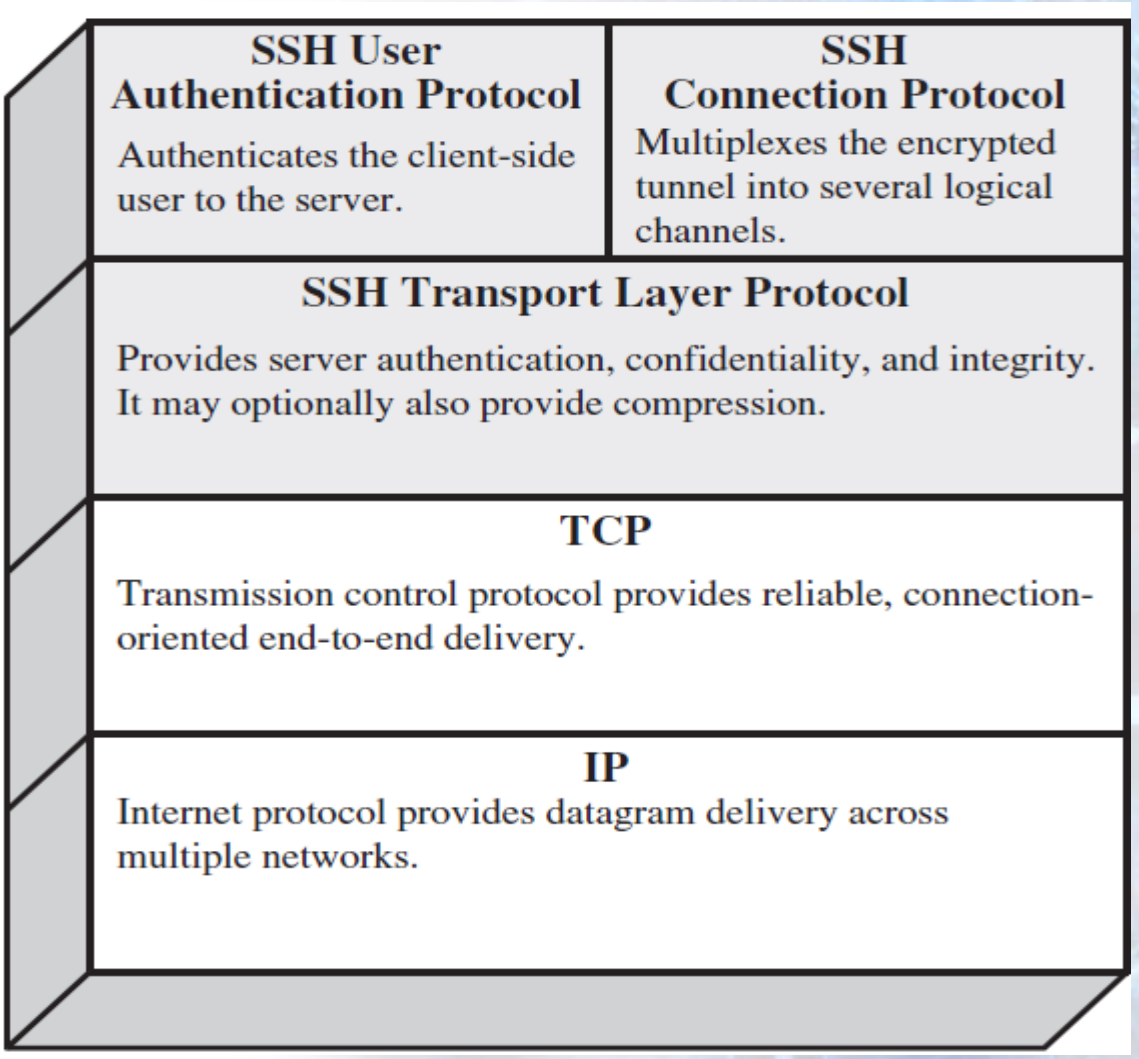
SSH Transport Layer Protocol

Figures and material in this topic have been adapted from

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

SSH Transport Layer Protocol

SSH Protocol Stack



SSH Transport Layer Protocol

Host Keys:

- Server authentication occurs at transport layer, based on the server possessing a public/private pair.
- A server may have multiple host keys using multiple different asymmetric encryption algorithms

SSH Transport Layer Protocol

- Multiple hosts may share the same host key.
- The server host key is used during key exchange to authenticate the identity of the host.
- RFC 4251 dictates two alternative trust models:

SSH Transport Layer Protocol

- The client has a local database that associates each host name with the corresponding public host key.
- Drawback: name-to-key associations may become burdensome to maintain.

SSH Transport Layer Protocol

- The host name-to-key association is certified by a trusted certification authority (CA).
- The client only knows the CA root key and can verify the validity of all host keys certified by accepted CAs.

SSH Transport Layer Protocol

Packet Exchange:

- First, the client establishes a TCP connection to the server.
- This is done via the TCP protocol and is not part of the Transport Layer Protocol.

SSH Transport Layer Protocol

- Once the connection is established, the client and server exchange data, referred to as packets, in the data field of a TCP segment.
- Each packet is in the following format:

SSH Transport Layer Protocol

- **Packet length:** Length of the packet in bytes, not including the packet length and MAC fields.
- **Padding length:** Length of the random padding field.

SSH Transport Layer Protocol

- **Payload:** Useful contents of the packet.
- If compression is negotiated, then in subsequent packets, this field is compressed.

SSH Transport Layer Protocol

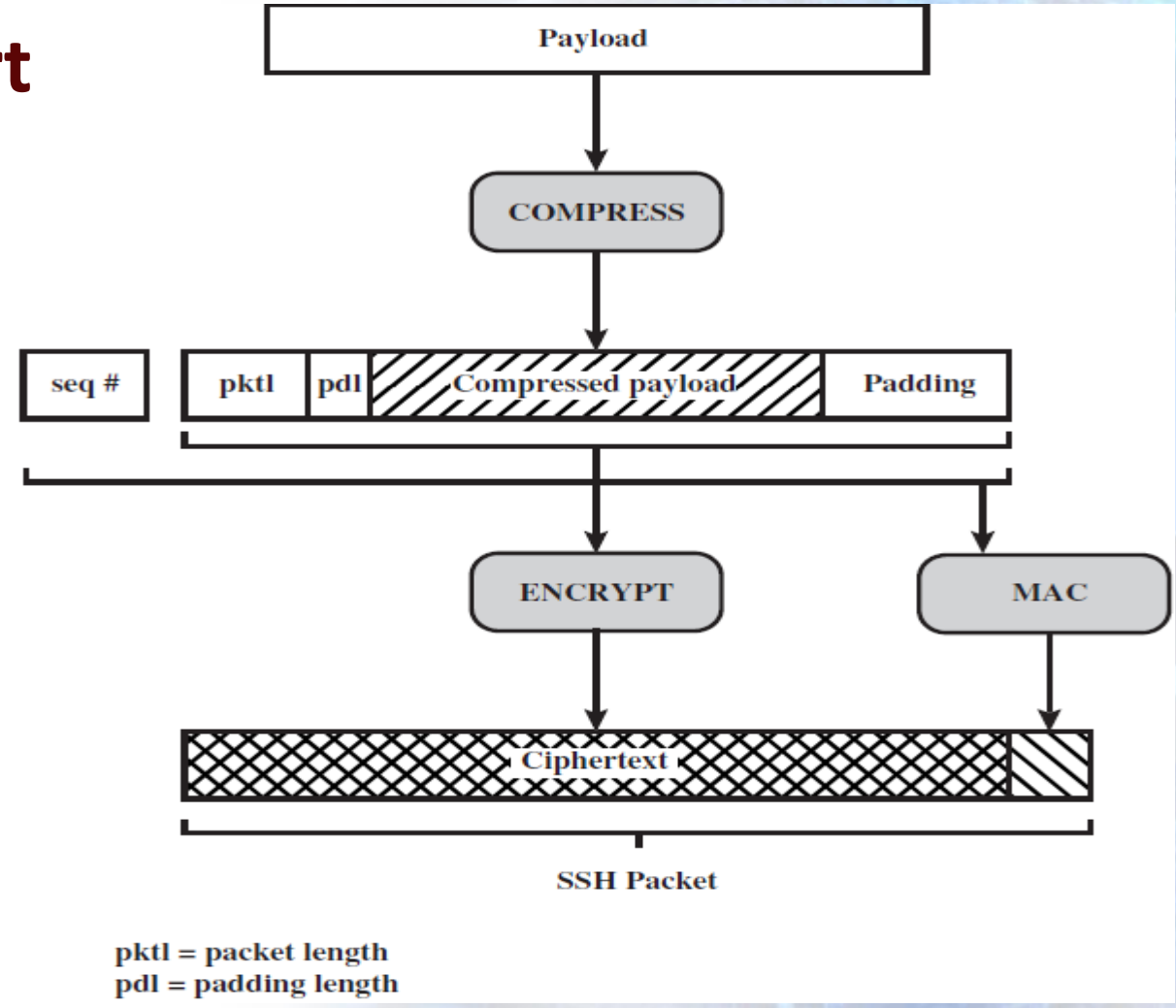
- **Random padding:** Once an encryption algorithm has been negotiated, this field is added. It contains random bytes of padding.

SSH Transport Layer Protocol

- **Message authentication code (MAC):** If message authentication has been negotiated, this field contains the MAC value. The MAC value is computed over the entire packet plus a sequence number, excluding the MAC field.

SSH Transport Layer Protocol

Transport Layer Protocol Packet Forma-tion



SSH Transport Layer Protocol

- The first step of the packet exchange is the **identification string exchange**.
- It begins with the client sending a packet with an identification string.
- The server responds with its own identification string.

SSH Transport Layer Protocol

- Next comes **algorithm negotiation**.
- Each side sends an SSH_MSG_KEXINIT containing lists of supported algorithms in the order of preference to the sender.

SSH Transport Layer Protocol

Cipher	
3des-cbc*	Three-key 3DES in CBC mode
blowfish-cbc	Blowfish in CBC mode
twofish256-cbc	Twofish in CBC mode with a 256-bit key
twofish192-cbc	Twofish with a 192-bit key
twofish128-cbc	Twofish with a 128-bit key
aes256-cbc	AES in CBC mode with a 256-bit key
aes192-cbc	AES with a 192-bit key
aes128-cbc**	AES with a 128-bit key
Serpent256-cbc	Serpent in CBC mode with a 256-bit key
Serpent192-cbc	Serpent with a 192-bit key
Serpent128-cbc	Serpent with a 128-bit key
arcfour	RC4 with a 128-bit key
cast128-cbc	CAST-128 in CBC mode

* = Required
** = Recommended

MAC algorithm	
hmac-sha1*	HMAC-SHA1; digest length = key length = 20
hmac-sha1-96**	First 96 bits of HMAC-SHA1; digest length = 12; key length = 20
hmac-md5	HMAC-MD5; digest length = key length = 16
hmac-md5-96	First 96 bits of HMAC-MD5; digest length = 12; key length = 16

Compression algorithm	
none*	No compression
zlib	Defined in RFC 1950 and RFC 1951

SSH Transport Layer Protocol

- Next is **key exchange**.
- Two versions of Diffie-Hellman (RFC 2409) key exchange are specified and require only one packet in each direction.
- The two sides share a master key and server been authenticated to client.

SSH Transport Layer Protocol

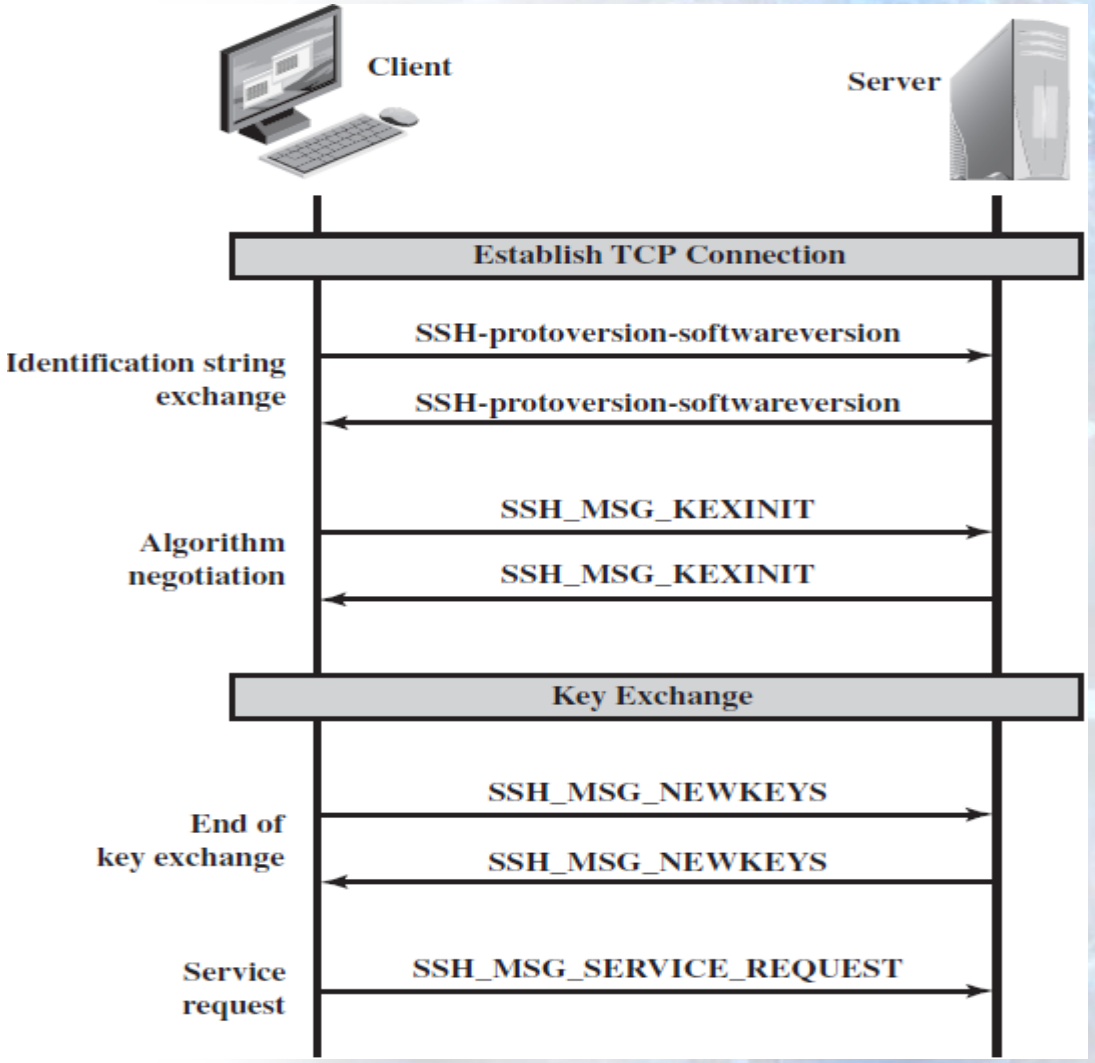
- The **end of key exchange** is signaled by the exchange of SSH_MSG_NEWKEYS packets.
- The final step is **service request**.
- The client sends an SSH_MSG_SERVICE_REQUEST packet.

SSH Transport Layer Protocol

- This is to request either the User Authentication or the Connection Protocol.
- Subsequent to this, all data is exchanged as the payload of an SSH Transport Layer packet, protected by encryption and MAC.

SSH Transport Layer Protocol

SSH Transport Layer Protocol Packet Exchanges



SSH User Authentication Protocol

Network Security

SSH User Authentication Protocol

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe working of the SSH User Authentication Protocol.

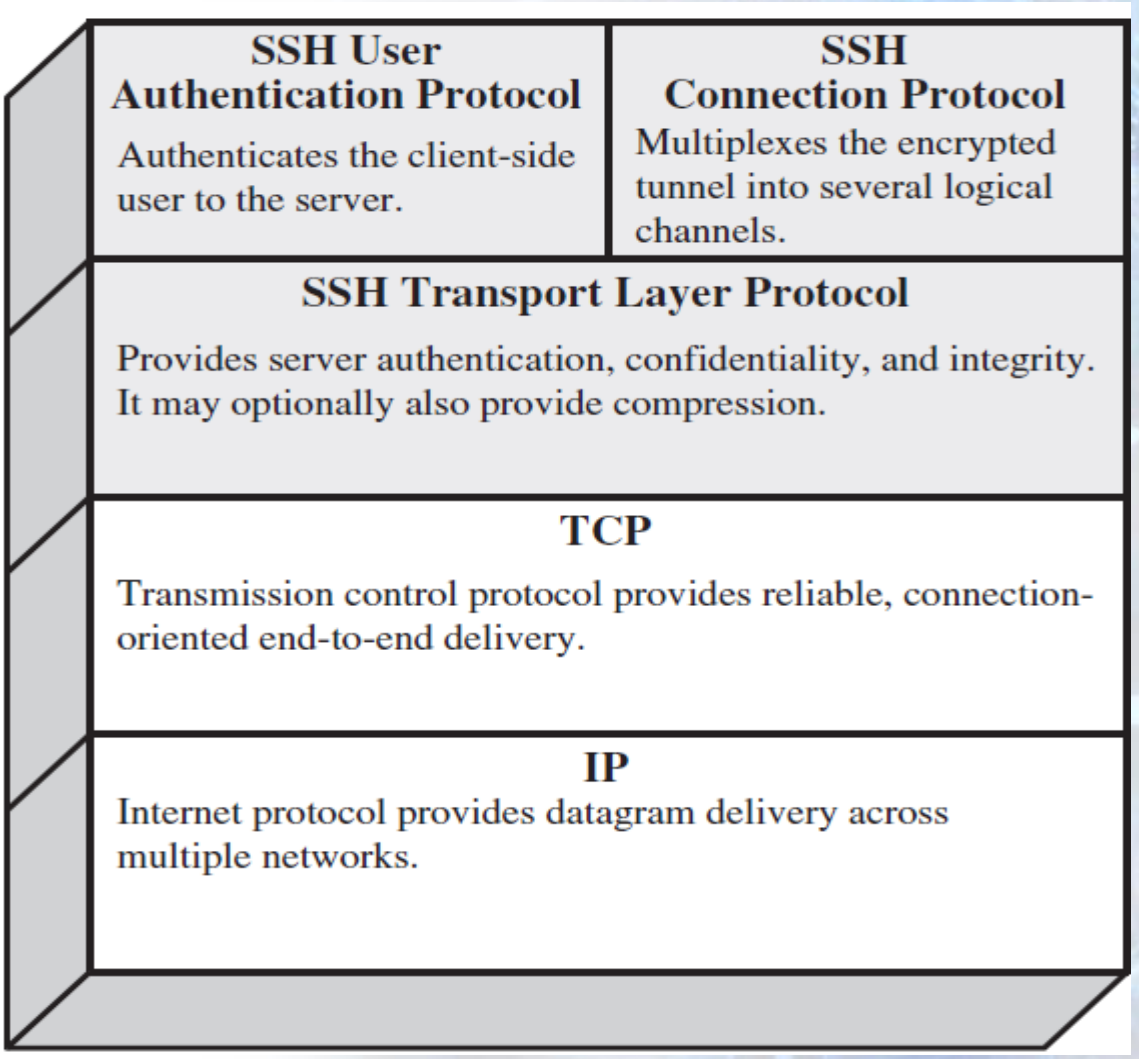
SSH User Authentication Protocol

Figures and material in this topic have been adapted from

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

SSH User Authentication Protocol

SSH Protocol Stack



SSH User Authentication Protocol

- **Message Types and Formats:** Three types of messages are always used in the User Authentication Protocol.
- Authentication requests from the client have the format:

SSH User Authentication Protocol

byte	SSH_MSG_USERAUTH_REQUEST (50)
string	user name
string	service name
string	method name
...	method specific fields

where user name is the authorization identity the client is claiming, service name is the facility to which the client is requesting access (typically the SSH Connection Protocol), and method name is the authentication method being used in this request. The first byte has decimal value 50, which is interpreted as SSH_MSG_USERAUTH_REQUEST.

SSH User Authentication Protocol

- If the server either (1) rejects the authentication request or (2) accepts the request but requires one or more additional authentication methods, the server sends a message with the format:

SSH User Authentication Protocol

byte	`SSH_MSG_USERAUTH_FAILURE (51)`
name-list	authentications that can continue
boolean	partial success

where the name-list is a list of methods that may productively continue the dialog. If the server accepts authentication, it sends a single byte message: `SSH_MSG_USERAUTH_SUCCESS (52)`.

SSH User Authentication Protocol

- **Message Exchange:** The message exchange involves the following steps.
- **1.** The client sends a SSH_MSG_USERAUTH _REQUEST with a requested method of none.

SSH User Authentication Protocol

- **2.** The server checks to determine if the user name is valid. If not, the server returns SSH_MSG_USERAUTH _FAILURE with the partial success value of false.
- If the user name is valid, the server proceeds to step 3.

SSH User Authentication Protocol

- **3.** The server returns SSH_MSG_USERAUTH _FAILURE with a list of one or more authentication methods to be used.

SSH User Authentication Protocol

- **4.** The client selects one of the acceptable authentication methods and sends a SSH_MSG_USERAUTH_REQUEST with that method name and the required method-specific fields.

SSH User Authentication Protocol

- **5.** If authentication succeeds and more authentication methods are required, the server proceeds to step 3, using a partial success value of true.
- If authentication fails, server proceeds to step 3, using a partial success value of false.

SSH User Authentication Protocol

- **6.** When all required authentication methods succeed, the server sends a SSH_MSG_USERAUTH_SUCCESS message, and the Authentication Protocol is over.

SSH User Authentication Protocol

- **Authentication Methods:** The server may require one or more of the following authentication methods.

SSH User Authentication Protocol

- **publickey:** The client sends a message to the server that contains the client's public key, with the message signed by the client's private key.

SSH User Authentication Protocol

- When the server receives this message, it checks whether the supplied key is acceptable for authentication and, if so, it checks whether the signature is correct.

SSH User Authentication Protocol

- **Password:** The client sends a message containing a plaintext password, which is protected by encryption by the Transport Layer Protocol.

SSH User Authentication Protocol

- **hostbased:** Authentication is performed on the client's host rather than the client itself.
- This method works by having the client send a signature created with the private key of the client host.

SSH User Authentication Protocol

- Rather than directly verifying the user's identity, the SSH server verifies the identity of the client host.

SSH Connection Protocol

Network Security

SSH Connection Protocol

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe working of the SSH Connection Protocol.

SSH Connection Protocol

Figures and material in this topic have been adapted from

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

SSH Connection Protocol

- The SSH Connection Protocol runs on top of the SSH Transport Layer Protocol and assumes that a secure authentication connection is in use.
- Connection, referred to as a **tunnel,** is used to multiplex a number of logical channels.

SSH Connection Protocol

Channel Mechanism:

- All types of communication using SSH, such as a terminal session, are supported using separate channels.
- Either side may open a channel.

SSH Connection Protocol

- For each channel, each side associates a unique channel number, which need not be the same on both ends.

SSH Connection Protocol

- Channels are flow controlled using a window mechanism.
- No data may be sent to a channel until a message is received to indicate that window space is available.

SSH Connection Protocol

- The life of a channel progresses through three stages:
- opening a channel,
- data transfer, and
- closing a channel.

SSH Connection Protocol

- When either side wishes to **open a new channel,** it allocates a local number for the channel and then sends a message of the form:

SSH Connection Protocol

byte	SSH_MSG_CHANNEL_OPEN
string	channel type
uint32	sender channel
uint32	initial window size
uint32	maximum packet size
….	channel type specific data follows

- where uint32 means unsigned 32-bit integer.
- The channel type identifies the application for this channel.
- The sender channel is the local channel number.

SSH Connection Protocol

byte	SSH_MSG_CHANNEL_OPEN
string	channel type
uint32	sender channel
uint32	initial window size
uint32	maximum packet size
....	channel type specific data follows

- The initial window size specifies how many bytes of channel data can be sent to the sender of this message without adjusting the window.
- The maximum packet size specifies the maximum size of an individual data packet that can be sent to the sender.

SSH Connection Protocol

- If the remote side is able to open the channel, it returns a SSH_MSG_CHANNEL_OPEN_CONFIRMATION, which includes the sender channel number, the recipient channel number, and window and packet size values for incoming traffic.

SSH Connection Protocol

- Otherwise, the remote side returns a SSH_MSG_CHANNEL_OPEN_FAILURE message with a reason code indicating the reason for failure.

SSH Connection Protocol

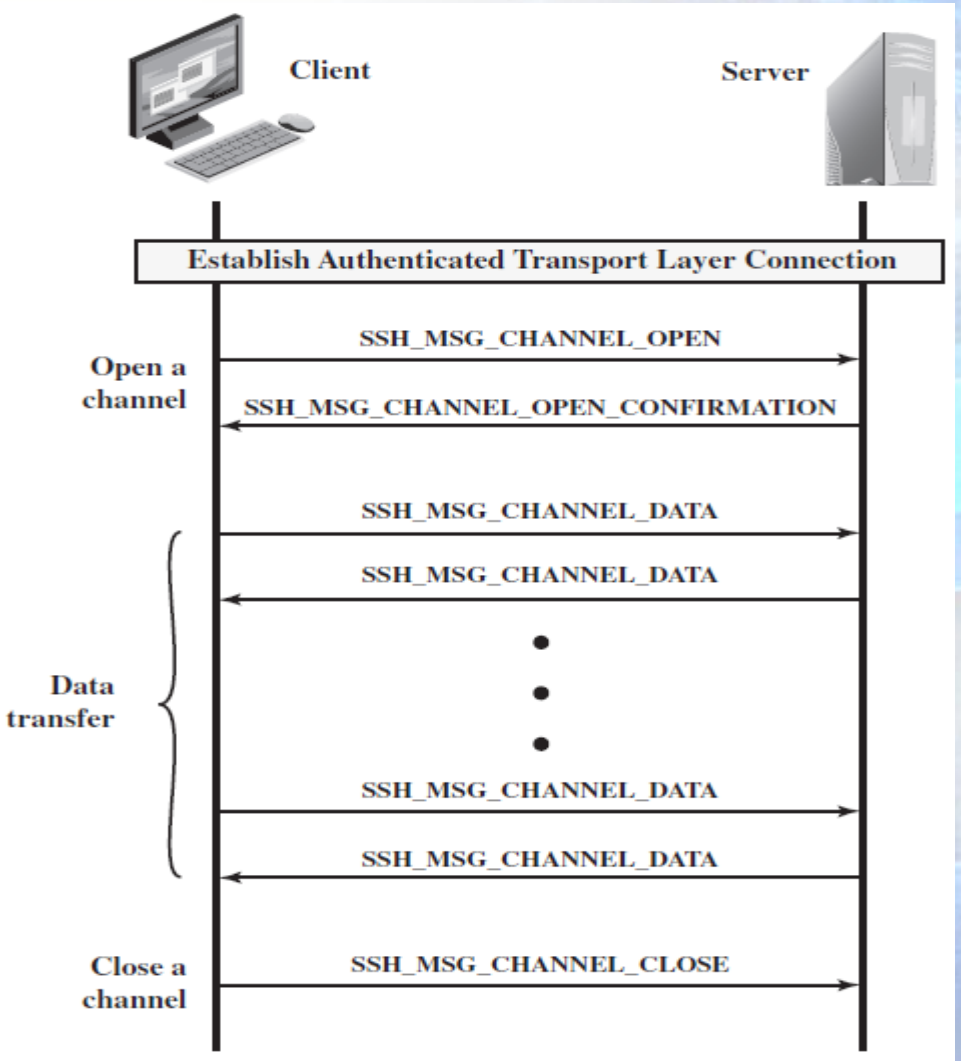
- Once a channel is open, **data transfer** is performed using a SSH_MSG_CHANNEL_DATA message, which includes the recipient channel number and a block of data.

SSH Connection Protocol

- These messages, in both directions, may continue as long as the channel is open.
- When either side wishes to **close a channel,** it sends a SSH_MSG_CHANNEL_CLOSE message (includes recipient channel number).

SSH Connection Protocol

An Example



SSH Connection Protocol

Channel Types:

- Four channel types are recognized in the SSH Connection Protocol specification.
- **session:** The remote execution of a program.

SSH Connection Protocol

- The program may be a shell, an application such as file transfer or e-mail, a system command, or some built-in subsystem.
- Once a session channel is opened, subsequent requests are used to start the remote program.

SSH Connection Protocol

- **x11:** refers to the X Window System, a computer software system and network protocol that provides a graphical user interface (GUI) for networked computers.

SSH Connection Protocol

End

- X allows applications to run on a network server but to be displayed on a desktop machine.
- **forwarded-tcpip:** This is remote port forwarding.
- **direct-tcpip:** This is local port forwarding.

SSH Tunneling

Network Security

SSH Tunneling

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe working of the SSH Tunneling.

SSH Tunneling

Figures and material in this topic have been adapted from

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

SSH Tunneling

- One of the most useful features of SSH is port forwarding.
- It provides the ability to convert any insecure TCP connection into a secure SSH connection.
- It is also referred to as SSH tunneling.

SSH Tunneling

- A **port** is an identifier of a user of TCP.
- Any application that runs on top of TCP has a port number.
- Incoming TCP traffic is delivered to the appropriate application on the basis of the port number.

SSH Tunneling

- To understand the basic concept, lets assume that we have a client application that is identified by port number *x* and a server application identified by port number *y*.

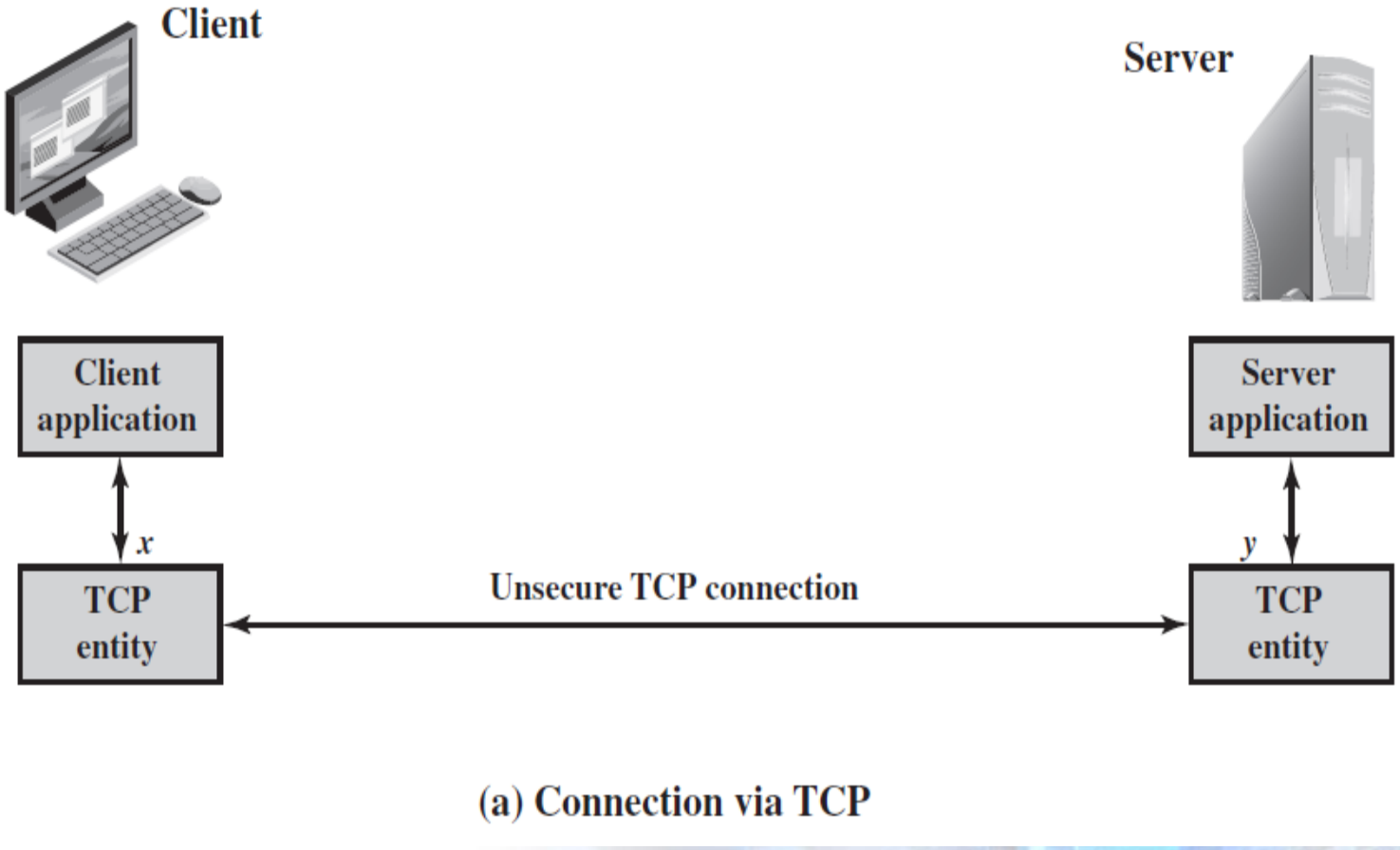
SSH Tunneling

- At some point, the client application invokes the local TCP entity and requests a connection to the remote server on port *y*.

SSH Tunneling

- The local TCP entity negotiates a TCP connection with the remote TCP entity, such that the connection links local port *x* to remote port *y*.

SSH Tunneling



(a) Connection via TCP

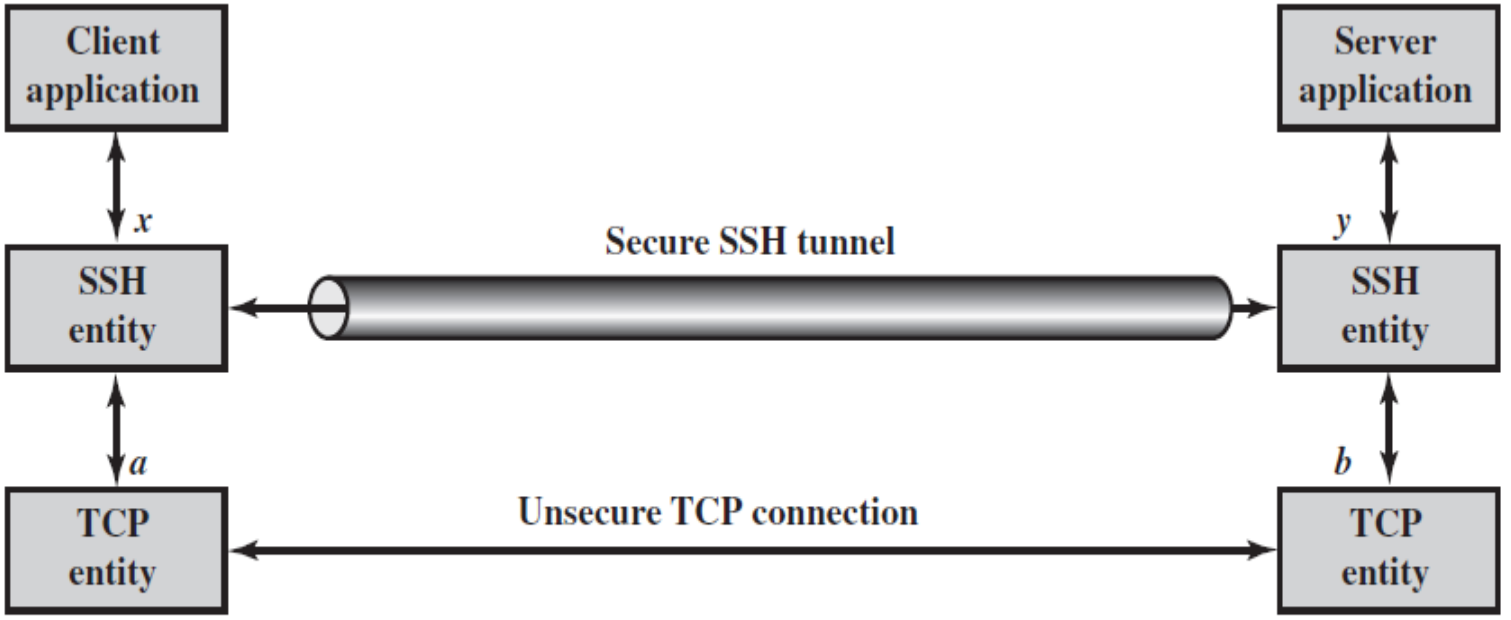
SSH Tunneling

- To secure this connection, SSH is configured so that the SSH Transport Layer Protocol establishes a TCP connection between the SSH client and server entities, with TCP port numbers *a* and *b*, respectively.

SSH Tunneling

- A secure SSH tunnel is established over this TCP connection.
- Traffic from the client at port *x* is redirected to the local SSH entity and travels through tunnel where remote SSH entity delivers the data to the server application on port *y*.

SSH Tunneling



(b) Connection via SSH tunnel

SSH Tunneling

- SSH supports two types of port forwarding:
- local forwarding and remote forwarding.
- **Local forwarding** allows the client to set up a “hijacker” process.

SSH Tunneling

- This will intercept selected application-level traffic and redirect it from an unsecured TCP connection to a secure SSH tunnel.
- SSH is configured to listen on selected ports.

SSH Tunneling

- SSH grabs all traffic using a selected port and sends it through an SSH tunnel.
- On the other end, the SSH server sends the incoming traffic to the destination port dictated by the client application.

SSH Tunneling

An Example

Suppose you have an e-mail client on your desktop and use it to get e-mail from your mail server via the Post Office Protocol (POP). The assigned port number for POP3 is port 110. We can secure this traffic in the following way:

1. The SSH client sets up a connection to the remote server.
2. Select an unused local port number, say 9999, and configure SSH to accept traffic from this port destined for port 110 on the server.
3. The SSH client informs the SSH server to create a connection to the destination, in this case mailserver port 110.
4. The client takes any bits sent to local port 9999 and sends them to the server inside the encrypted SSH session. The SSH server decrypts the incoming bits and sends the plaintext to port 110.
5. In the other direction, the SSH server takes any bits received on port 110 and sends them inside the SSH session back to the client, who decrypts and sends them to the process connected to port 9999.

SSH Tunneling

- With **remote forwarding,** the user's SSH client acts on the server's behalf.
- Client receives traffic with a given dest. port number, places the traffic on the correct port and sends it to the destination the user chooses.

SSH Tunneling

An Example

You wish to access a server at work from your home computer. Because the work server is behind a firewall, it will not accept an SSH request from your home computer. However, from work you can set up an SSH tunnel using remote forwarding. This involves the following steps.

1. From the work computer, set up an SSH connection to your home computer. The firewall will allow this, because it is a protected outgoing connection.
2. Configure the SSH server to listen on a local port, say 22, and to deliver data across the SSH connection addressed to remote port, say 2222.
3. You can now go to your home computer, and configure SSH to accept traffic on port 2222.
4. You now have an SSH tunnel that can be used for remote logon to the work server.

Introduction to Wireless Security

Network Security

Introduction to Wireless Security

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe basics of Wireless Security and threats to Wireless networks.

Introduction to Wireless Security

Figures and material in this topic have been adapted from

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

Introduction to Wireless Security

- Wireless networks, and the wireless devices that use them, introduce a host of security problems over and above those found in wired networks.

Introduction to Wireless Security

- Some of the key factors contributing to the higher security risk of wireless networks compared to wired networks include the following:

Introduction to Wireless Security

- **Channel:** Wireless networking typically involves broadcast communications, which is far more susceptible to eavesdropping and jamming than wired networks.

Introduction to Wireless Security

- **Mobility:** Wireless devices are far more portable and mobile than wired devices.
- This mobility results in a number of risks.

Introduction to Wireless Security

- **Resources:** Some wireless devices, such as smartphones and tablets, have sophisticated OSs but limited memory and processing resources with which to counter threats, including denial of service and malware.

Introduction to Wireless Security

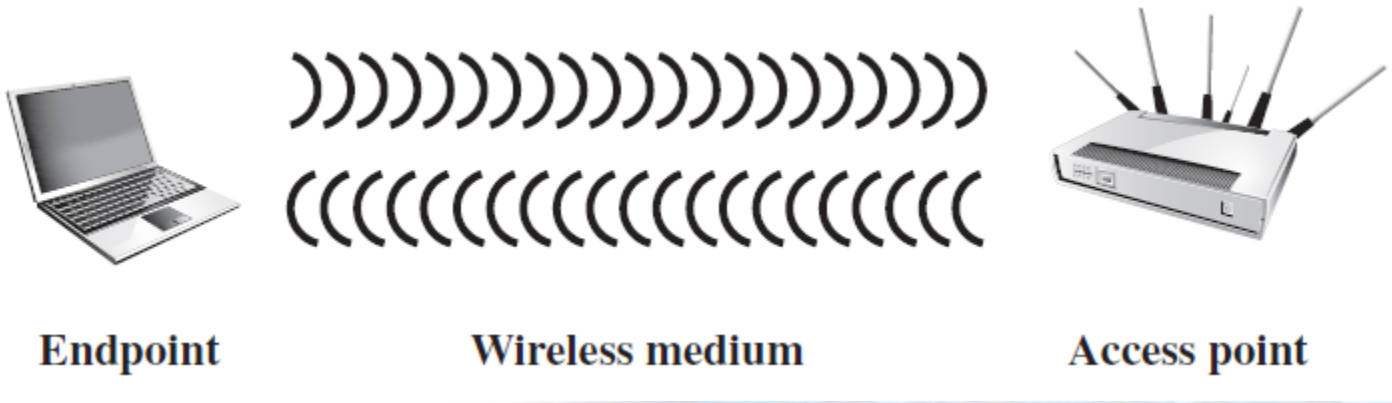
- **Accessibility:** Some wireless devices, such as sensors and robots, may be left unattended in remote and/or hostile locations.
- This greatly increases their vulnerability to physical attacks.

Introduction to Wireless Security

- The wireless environment consists of three components that provide point of attack:

Introduction to Wireless Security

Wireless Networking Components



Introduction to Wireless Security

- 1. The wireless client can be a cell phone, a Wi-Fi–enabled laptop or tablet, a wireless sensor, a Bluetooth device, and so on.

Introduction to Wireless Security

- 2. The wireless access point provides a connection to the network or service.
- Examples of access points are cell towers, Wi-Fi hotspots, and wireless access points to wired local or wide area networks.

Introduction to Wireless Security

- 3. The transmission medium, which carries the radio waves for data transfer, is also a source of vulnerability.

Introduction to Wireless Security

- **Wireless Network Threats:**
- Following are the security threats to wireless networks:

Introduction to Wireless Security

- **Accidental association:** Wireless LANs or wireless access points to wired LANs in close proximity (e.g., in the same or neighboring buildings) may create overlapping transmission ranges.

Introduction to Wireless Security

- A user intending to connect to one LAN may unintentionally lock on to a wireless access point from a neighboring network.

Introduction to Wireless Security

- **Malicious association:** A wireless device is configured to appear to be a legitimate access point, enabling the operator to steal passwords from legitimate users and then penetrate a wired network through a legitimate wireless access point.

Introduction to Wireless Security

- **Ad hoc networks:** These are peer-to-peer networks between wireless computers with no access point between them.
- Such networks can pose a security threat due to a lack of a central control point.

Introduction to Wireless Security

- **Nontraditional networks:** Personal network Bluetooth devices, barcode readers, and handheld PDAs, pose a security risk in terms of both eavesdropping and spoofing.

Introduction to Wireless Security

- **Identity theft (MAC spoofing):** This occurs when an attacker is able to eavesdrop on network traffic and identify the MAC address of a computer with network privileges.

Introduction to Wireless Security

- **Man-in-the middle attacks:** This attack involves persuading a user and an access point to believe that they are talking to each other when in fact communication is going through an intermediate attacking device.

Introduction to Wireless Security

- **Denial of service:** This attack occurs when an attacker continually bombards a wireless access point or some other accessible wireless port with various protocol messages designed to consume system resources.

Introduction to Wireless Security

End

- **Network injection:** This attack targets wireless access points that are exposed to nonfiltered network traffic, such as routing protocol messages or network management messages.

Wireless Network Security Measures

Network Security

Wireless Network Security Measures

Objectives of the Topic

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

Wireless Network Security Measures

Figures and material in this topic have been adapted from

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

Wireless Network Security Measures

- We can group wireless security measures along three dimensions.

Wireless Network Security Measures

1.Securing Wireless Transmissions:

- The principal threats to wireless transmission are eavesdropping, altering or inserting messages, and disruption.

Wireless Network Security Measures

- To deal with eavesdropping, two types of countermeasures are appropriate:
- **A. Signal-hiding technique:**
- turn off service set identifier (SSID) broadcasting by wireless access points

Wireless Network Security Measures

- Assign cryptic names to SSIDs.
- Reduce signal strength to the lowest level that still provides requisite coverage.
- Locate wireless access points in the interior of the building, away from windows and exterior walls.

Wireless Network Security Measures

- **B. Encryption**
- is effective against eavesdropping to the extent that the encryption keys are secured.

Wireless Network Security Measures

- The use of encryption and authentication protocols is the standard method of countering attempts to alter or insert transmissions.

Wireless Network Security Measures

- **2. Securing Wireless Access Points**
- The main threat involving wireless access points is unauthorized access to the network.

Wireless Network Security Measures

- The principal approach for preventing such access is the IEEE 802.1X standard for port-based network access control.

Wireless Network Security Measures

- IEEE 802.1X standard provides an authentication mechanism for devices wishing to attach to a LAN or a wireless network.

Wireless Network Security Measures

- The use of IEEE 802.1X can prevent rogue access points and other unauthorized devices from becoming insecure backdoors.

Wireless Network Security Measures

- **Securing Wireless Networks:**
- Following techniques are recommended for wireless network security:

Wireless Network Security Measures

- **1. Use encryption:**
- Wireless routers are typically equipped with built-in encryption mechanisms for router-to-router traffic.

Wireless Network Security Measures

- **2. Use antivirus and antispyware software, and a firewall:**
- These facilities should be enabled on all wireless network endpoints.

Wireless Network Security Measures

- **3. Turn off identifier broadcasting:**
- Wireless routers are typically configured to broadcast an identifying signal so that any device within range can learn of the router's existence.

Wireless Network Security Measures

- If a network is configured so that authorized devices know the identity of routers, this capability can be disabled, so as to thwart attackers.

Wireless Network Security Measures

- **4. Change the identifier on your router from the default**:
- This measure thwarts attackers who will attempt to gain access to a wireless network using default router identifiers.

Wireless Network Security Measures

- **5. Change your router's pre-set password for administration:**
- This is another prudent step.

Wireless Network Security Measures

End

- **6. Allow only specific computers to access your wireless network:** A router can be configured to only communicate with approved MAC addresses, which can be spoofed, so this is just one element of a security strategy.

Basics of Mobile Device Security

Network Security

Basics of Mobile Device Security

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain basics of Mobile Device Security.

Basics of Mobile Device Security

Figures and material in this topic have been adapted from

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

Basics of Mobile Device Security

- Mobile devices have become an essential element for organizations as part of the overall network infrastructure.

Basics of Mobile Device Security

- Prior to the widespread use of smartphones, network security was based upon clearly defined perimeters that separated trusted internal networks from the untrusted Internet.

Basics of Mobile Device Security

- Corporate IT was tightly controlled.
- User devices were typically limited to Windows PCs.
- Business applications were controlled by IT and either run locally on endpoints or on physical servers in data centers.

Basics of Mobile Device Security

- Today, there have been massive changes in each of these assumptions.
- An organization's networks must accommodate the following:

Basics of Mobile Device Security

- **Growing use of new devices:**
- Organizations are experiencing significant growth in employee use of mobile devices.

Basics of Mobile Device Security

- **Cloud-based applications:**
- Applications no longer run solely on physical servers in corporate data centers.
- Can run anywhere; on traditional physical servers, on mobile virtual servers, or in the cloud.

Basics of Mobile Device Security

- Additionally, end users can now take advantage of a wide variety of cloud-based applications and IT services for personal and professional use.
- Facebook can be used as a component of a corporate marketing campaign.

Basics of Mobile Device Security

- **De-perimeterization:**
- Given new device proliferation, application mobility, and cloud-based consumer and corporate services, the notion of a static network perimeter is all but gone.

Basics of Mobile Device Security

- Now there are a multitude of network perimeters around devices, applications, users, and data.
- These perimeters have also become quite dynamic as they must adapt to various environmental conditions.

Basics of Mobile Device Security

- **External business requirements:**
- The enterprise must also provide guests, third-party contractors, and business partners network access using various devices from a multitude of locations.

Basics of Mobile Device Security

- **Security Threats:**
- Guidelines for Managing and Securing Mobile Devices in the Enterprise, July 2012 lists seven major security concerns for mobile devices:

Basics of Mobile Device Security

- **1. Lack of physical security controls:**
- The security policy for mobile devices must be based on the assumption that any mobile device may be stolen or at least accessed by a malicious party.

Basics of Mobile Device Security

- **2. Use of untrusted mobile devices:**
- The organization must assume that not all devices are trustworthy.

Basics of Mobile Device Security

- **3. Use of untrusted networks:**
- The security policy must be based on the assumption that the networks between the mobile device and the organization are not trustworthy.

Basics of Mobile Device Security

- **4. Use of untrusted content:**
- Mobile devices may access and use content that other computing devices do not encounter.

Basics of Mobile Device Security

- **5. Use of applications created by unknown parties:**
- It is easy to find and install third-party applications on mobile devices and this poses the risk of installing malicious software.

Basics of Mobile Device Security

- **6. Interaction with other systems:** Unless an organization has control of all devices involved in synchronization, there is a high risk of data of organization being stored in unsecured location, plus risk of introduction of malware.

Basics of Mobile Device Security

End

- **7. Use of location services:**
- An attacker can use location information to determine where the device and user are located, which may be of use to the attacker.

Mobile Device Security Strategy

Network Security

Mobile Device Security Strategy

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe Mobile Device Security Strategy.

Mobile Device Security Strategy

Figures and material in this topic have been adapted from

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

Mobile Device Security Strategy

- The principal elements of a mobile device security strategy fall into three categories:
- Device security,
- client/server traffic security, and
- barrier security

Mobile Device Security Strategy

- **1. Device Security:**
- A number of organizations will supply mobile devices for employee use and preconfigure those devices to conform to the enterprise security policy.

Mobile Device Security Strategy

- Bring your-own-device (BYOD) policy allows the personal mobile devices of employees to have access to corporate resources.
- IT managers should be able to inspect each device before allowing network access.

Mobile Device Security Strategy

- Whether a device is owned by the organization or BYOD, the device should be configured with security controls, including the following:

Mobile Device Security Strategy

- Enable auto-lock causes the device to lock if it has not been used for a given amount of time, requiring the user to re-enter a PIN or a password to re-activate the device.

Mobile Device Security Strategy

- Enable password or PIN protection.
- The PIN or password is needed to unlock the device.
- E-mail and other data on the device are encrypted using PIN or password and can only be retrieved with the PIN or password.

Mobile Device Security Strategy

- Avoid using auto-complete features that remember user names or passwords.
- Enable remote wipe.
- Ensure that SSL protection is enabled, if available.
- Install antivirus software as it becomes available.

Mobile Device Security Strategy

- Make sure that software, including operating systems and applications, is up to date.
- Either sensitive data should be prohibited from storage on the mobile device or it should be encrypted.

Mobile Device Security Strategy

- IT staff should also have the ability to remotely access devices, wipe the device of all data, and then disable the device in the event of loss or theft.

Mobile Device Security Strategy

- The organization may prohibit installation of all unapproved applications, or implement a secure sandbox that isolates the organization's data and applications from other data and applications.

Mobile Device Security Strategy

- The organization can implement and enforce restrictions on what devices can synchronize and on the use of cloud-based storage.

Mobile Device Security Strategy

- To counter the threat of malicious use of location services, the security policy can dictate that such service is disabled on all mobile devices.

Mobile Device Security Strategy

- **2. Traffic Security** is based on the usual mechanisms for encryption and authentication.
- All traffic should be encrypted and travel by secure means, such as SSL.

Mobile Device Security Strategy

- Configure all traffic between mobile device and organization's network via a VPN.
- Employ a two-layer authentication mechanism that authenticates device, & then authenticates user of device.

Mobile Device Security Strategy

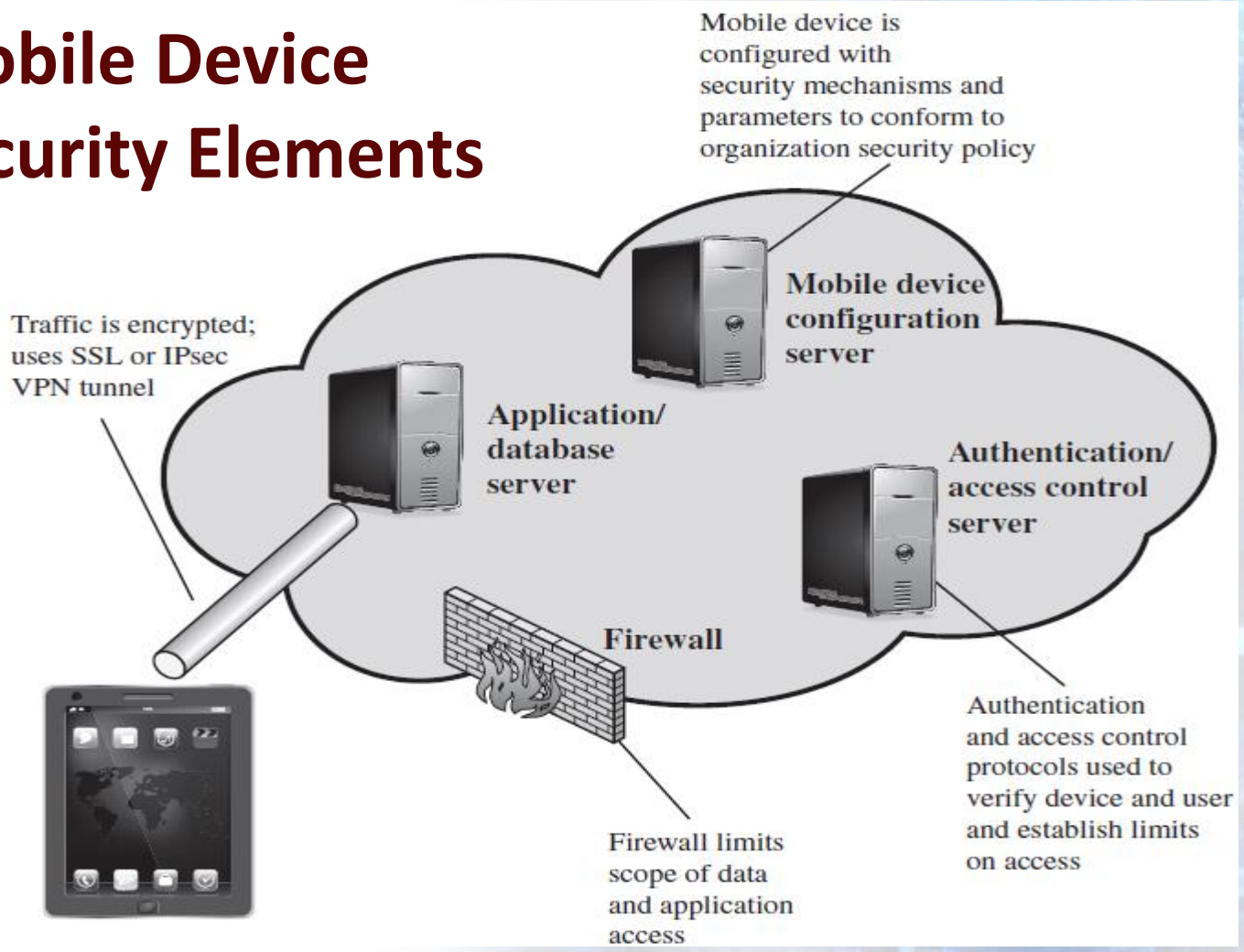
- **3. Barrier Security:** The organization should have security mechanisms to protect network from unauthorized access.
- The security strategy can also include firewall policies specific to mobile device traffic.

Mobile Device Security Strategy

- Firewall policies can limit the scope of data and application access for all mobile devices.
- Similarly, intrusion detection and intrusion prevention systems can be configured to have tighter rules for mobile device traffic.

Mobile Device Security Strategy

Mobile Device Security Elements



Introduction to IEEE 802.11 WLAN

Network Security

Introduction to IEEE 802.11 WLAN

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain basics of IEEE 802.11 Wireless LAN.

Introduction to IEEE 802.11 WLAN

Figures and material in this topic have been adapted from

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

Introduction to IEEE 802.11 WLAN

- IEEE 802 is a committee that has developed standards for a wide range of local area networks (LANs).

Introduction to IEEE 802.11 WLAN

- In 1990, the IEEE 802 Committee formed a new working group, IEEE 802.11, with a charter to develop a protocol and transmission specifications for wireless LANs (WLANs).

Introduction to IEEE 802.11 WLAN

- Since that time, the demand for WLANs at different frequencies and data rates has exploded.
- Next, we briefly define key terms used in the IEEE 802.11 standard.

Introduction to IEEE 802.11 WLAN

Access point (AP)	Any entity that has station functionality and provides access to the distribution system via the wireless medium for associated stations.
Basic service set (BSS)	A set of stations controlled by a single coordination function.
Coordination function	The logical function that determines when a station operating within a BSS is permitted to transmit and may be able to receive PDUs.
Distribution system (DS)	A system used to interconnect a set of BSSs and integrated LANs to create an ESS.
Extended service set (ESS)	A set of one or more interconnected BSSs and integrated LANs that appear as a single BSS to the LLC layer at any station associated with one of these BSSs.
MAC protocol data unit (MPDU)	The unit of data exchanged between two peer MAC entities using the services of the physical layer.
MAC service data unit (MSDU)	Information that is delivered as a unit between MAC users.
Station	Any device that contains an IEEE 802.11 conformant MAC and physical layer.

Introduction to IEEE 802.11 WLAN

- The first 802.11 standard to gain broad industry acceptance was 802.11b.
- Wireless Ethernet Compatibility Alliance (WECA), an industry consortium was formed in 1999.

Introduction to IEEE 802.11 WLAN

- Subsequently, it was renamed the Wi-Fi (Wireless Fidelity) Alliance, created test suite to certify interoperability for 802.11 products.

Introduction to IEEE 802.11 WLAN

- Wi-Fi is the term used for certified 802.11b products.
- Wi-Fi certification has been extended to 802.11g products
- A certification process for 802.11a products that was developed by the Wi-Fi Alliance is termed Wi-Fi5.

Introduction to IEEE 802.11 WLAN

- Recently the Wi-Fi Alliance has developed certification procedures for IEEE 802.11 security standards.
- Referred to as Wi-Fi Protected Access (WPA).

Introduction to IEEE 802.11 WLAN

- The smallest building block of a wireless LAN is a **basic service set (BSS)**, which consists of wireless stations executing the same MAC protocol and competing for access to the same shared wireless medium.

Introduction to IEEE 802.11 WLAN

- A BSS may be isolated, or it may connect to a backbone **distribution system (DS)** through an **access point (AP)**.
- The AP functions as a bridge and a relay point.

Introduction to IEEE 802.11 WLAN

- In a BSS, client stations do not communicate directly with one another.
- A media access control (MAC) frame is first sent from the originating station to the AP and then from the AP to the destination station.

Introduction to IEEE 802.11 WLAN

- Similarly, a MAC frame from a station in the BSS to a remote station is sent from the local station to the AP and then relayed by the AP over the DS on its way to the destination station.

Introduction to IEEE 802.11 WLAN

- The BSS generally corresponds to what is referred to as a cell in the literature.
- The DS can be a switch, a wired network, or a wireless network.

Introduction to IEEE 802.11 WLAN

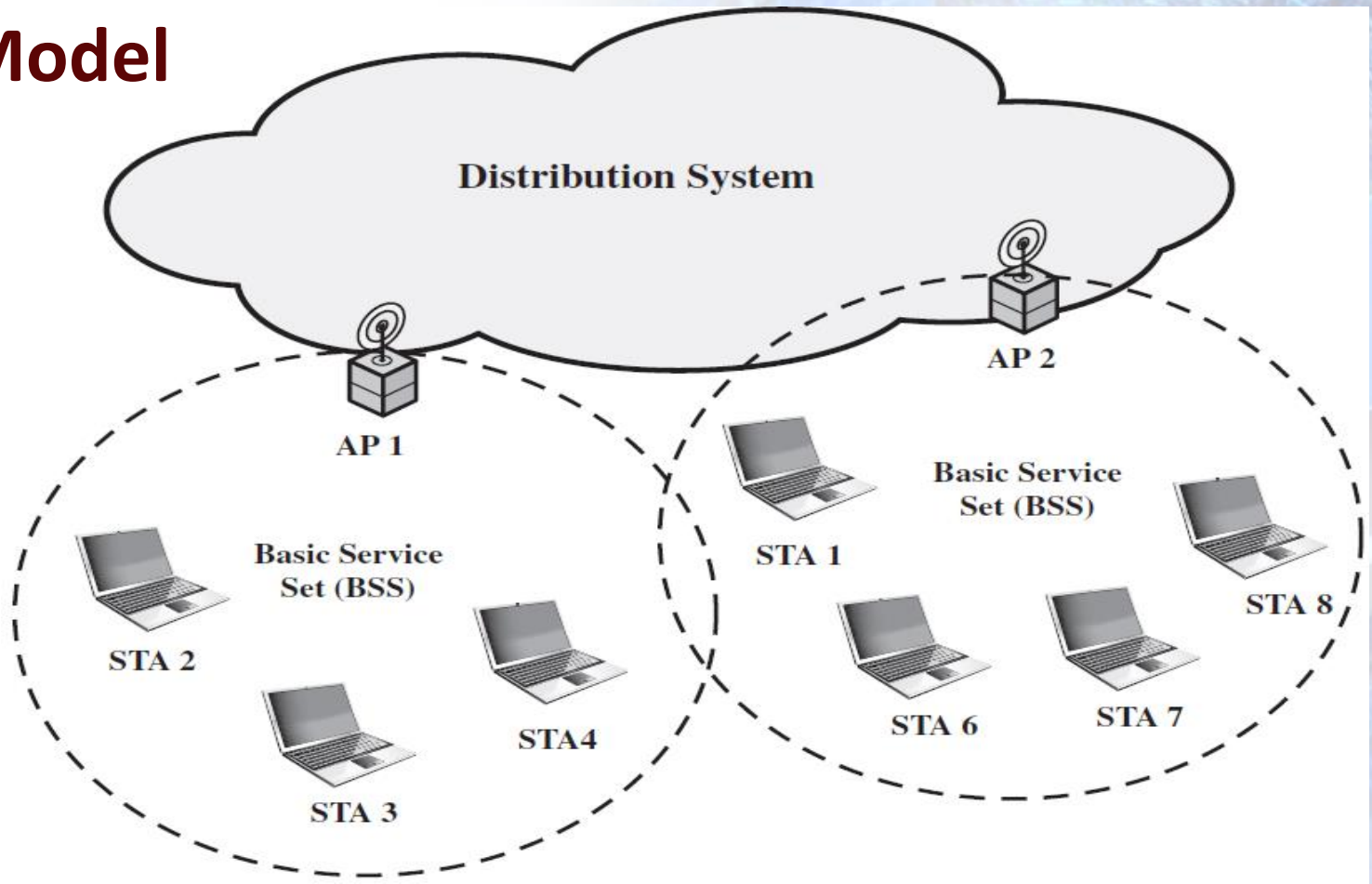
- When all the stations in the BSS are mobile stations that communicate directly with one another (not using an AP), the BSS is called an **independent BSS (IBSS)**.
- An IBSS is typically an ad hoc network.

Introduction to IEEE 802.11 WLAN

- Next, we show a simple configuration in which each station belongs to a single BSS.
- Here, each station is within wireless range only of other stations within the same BSS.

Introduction to IEEE 802.11 WLAN

Network Components And Architectural Model



Introduction to IEEE 802.11 WLAN

End

- It is also possible for two BSSs to overlap geographically, so that a single station could participate in more than one BSS.
- Association between a station and a BSS is dynamic; stations turn off, come within range, go outof range.

IEEE 802 Protocol Architecture

Network Security

IEEE 802 Protocol Architecture

Objectives of the Topic

- After completing this topic, a student will be able to
  - Describe Architecture of IEEE 802 Protocol.

IEEE 802 Protocol Architecture

Figures and material in this topic have been adapted from

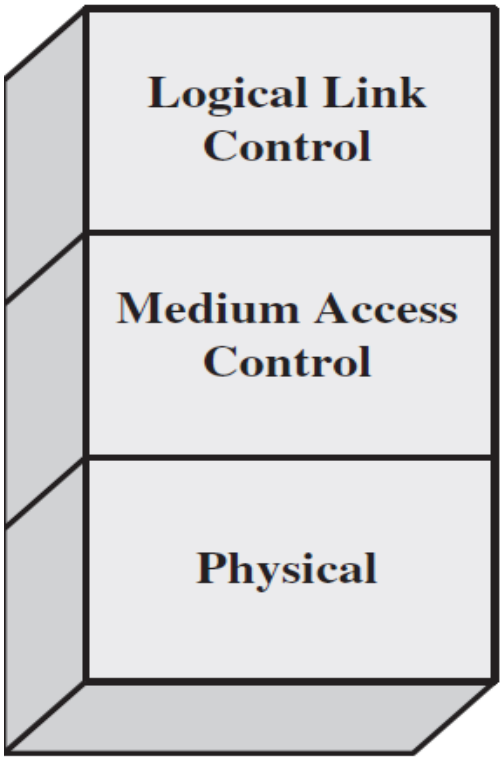
- *"Network Security Essentials: Applications and Standards"*, 2014 by William Stallings.

IEEE 802 Protocol Architecture

- IEEE 802.11 standards are defined within the structure of a layered set of protocols.
- This structure, used for all IEEE 802 standards, is illustrated next.

IEEE 802 Protocol Architecture

IEEE 802.11 Protocol Stack



	General IEEE 802 functions	Specific IEEE 802.11 functions
Logical Link Control	Flow control Error control	
Medium Access Control	Assemble data into frame Addressing Error detection Medium access	Reliable data delivery Wireless access control protocols
Physical	Encoding/decoding of signals Bit transmission/reception Transmission medium	Frequency band definition Wireless signal encoding

IEEE 802 Protocol Architecture

- **Physical Layer**
- The lowest layer of the IEEE 802 reference model is the physical layer, which includes such functions as encoding/decoding of signals and bit transmission/ reception.

IEEE 802 Protocol Architecture

- In addition, the physical layer includes a specification of the transmission medium.
- In the case of IEEE 802.11, the physical layer also defines frequency bands and antenna characteristics.

IEEE 802 Protocol Architecture

- **Media Access Control:**
- All LANs consist of collections of devices that share the network's transmission capacity.

IEEE 802 Protocol Architecture

- Some means of controlling access to the transmission medium is needed to provide an orderly and efficient use of that capacity.
- This is the function of a **media access control (MAC)** layer.

IEEE 802 Protocol Architecture

- The MAC layer receives data from a higher-layer protocol, typically the Logical Link Control (LLC) layer, in the form of a block of data known as the **MAC service data unit (MSDU)**.

IEEE 802 Protocol Architecture

- In general, the MAC layer performs the following functions:
- 1. On transmission, assemble data into a frame, known as a **MAC protocol data unit (MPDU)** with address and error-detection fields.

IEEE 802 Protocol Architecture

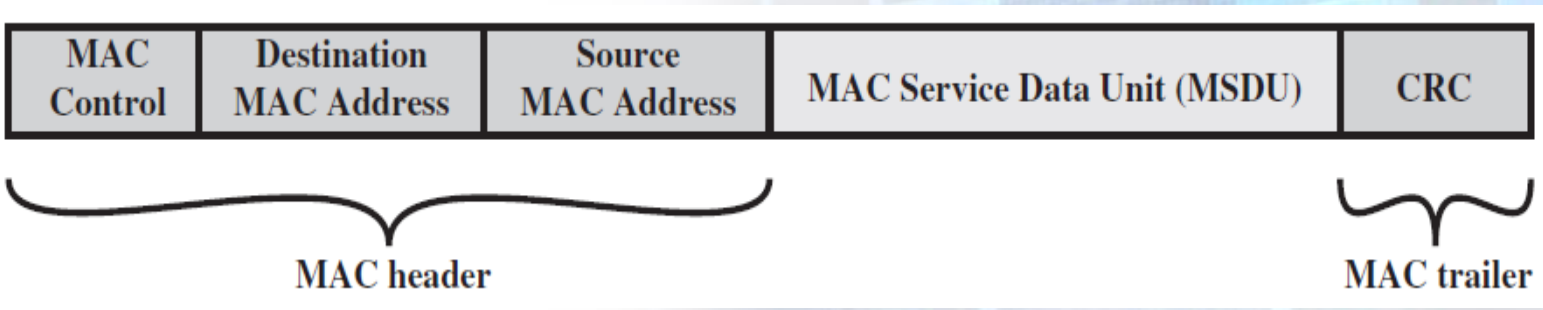
- 2. On reception, disassemble frame, and perform address recognition and error detection.
- 3. Govern access to the LAN transmission medium.

IEEE 802 Protocol Architecture

- The exact format of the MPDU differs somewhat for the various MAC protocols in use.
- All of the MPDUs have a similar format in general.
- The fields of this frame are as follows.

IEEE 802 Protocol Architecture

General IEEE 802 MPDU Format



IEEE 802 Protocol Architecture

- **MAC Control:** This field contains any protocol control information needed for the functioning of the MAC protocol.
- For example, a priority level could be indicated here.

IEEE 802 Protocol Architecture

- **Destination MAC Address:** The destination physical address on the LAN for this MPDU.
- **Source MAC Address:** The source physical address on the LAN for this MPDU.

IEEE 802 Protocol Architecture

- **MAC Service Data Unit:** The data from the next higher layer.
- **CRC:** The cyclic redundancy check field; also known as the Frame Check Sequence (FCS) field.
- This is an error-detecting code.

IEEE 802 Protocol Architecture

- The CRC is calculated based on the bits in the entire MPDU.
- The sender calculates the CRC and adds it to the frame.

IEEE 802 Protocol Architecture

- The receiver performs the same calculation on the incoming MPDU and compares that calculation to the CRC field in that incoming MPDU.
- If the two values don’t match, then one or more bits have been altered in transit.

IEEE 802 Protocol Architecture

- The **MAC header** and **MAC trailer** contain control information that accompany the data field and that are used by the MAC protocol.

IEEE 802 Protocol Architecture

- **Logical Link Control** In most data-link control protocols, the data-link protocol entity is responsible not only for detecting errors using the CRC, but for recovering from those errors by retransmitting damaged frames.

IEEE 802 Protocol Architecture

- In the LAN protocol architecture, these two functions are split between the MAC and LLC layers.
- The MAC layer is responsible for detecting errors and discarding any frames that contain errors.

IEEE 802 Protocol Architecture

- The LLC layer optionally keeps track of which frames have been successfully received and retransmits unsuccessful frames.

End

IEEE 802.11 Services

Network Security

IEEE 802.11 Services

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain IEEE 802.11 Services.

IEEE 802.11 Services

Figures and material in this topic have been adapted from

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

IEEE 802.11 Services

- IEEE 802.11 defines nine services that need to be provided by the wireless LAN to achieve functionality equivalent to that which is inherent to wired LANs.

IEEE 802.11 Services

IEEE 802.11 Services

Service	Provider	Used to support
Association	Distribution system	MSDU delivery
Authentication	Station	LAN access and security
Deauthentication	Station	LAN access and security
Disassociation	Distribution system	MSDU delivery
Distribution	Distribution system	MSDU delivery
Integration	Distribution system	MSDU delivery
MSDU delivery	Station	MSDU delivery
Privacy	Station	LAN access and security
Reassociation	Distribution system	MSDU delivery

IEEE 802.11 Services

- Two ways of categorizing the services:
- **1.** The service provider can be either the station or the DS.
- Station services are implemented in every 802.11 station, including AP stations.

IEEE 802.11 Services

- Distribution services are provided between BSSs.
- These services may be implemented in an AP or in another special-purpose device attached to the distribution system.

IEEE 802.11 Services

- **2.** Three of the services are used to control IEEE 802.11 LAN access and confidentiality.
- Six of the services are used to support delivery of MSDUs between stations.

IEEE 802.11 Services

- If the MSDU is too large to be transmitted in a single MPDU, it may be fragmented and transmitted in a series of MPDUs.
- We next discuss the services:

IEEE 802.11 Services

- **Distribution of Messages Within a DS:** The two services involved with the distribution of messages within a DS are distribution and integration.

IEEE 802.11 Services

- **Distribution** is the primary service used by stations to exchange MPDUs when the MPDUs must traverse the DS to get from a station in one BSS to a station in another BSS.

IEEE 802.11 Services

- If the two stations that are communicating are within the same BSS, then the distribution service logically goes through the single AP of that BSS.

IEEE 802.11 Services

- The **integration** service enables transfer of data between a station on an IEEE 802.11 LAN and a station on an integrated IEEE 802.x LAN.

IEEE 802.11 Services

- The integration service takes care of any address translation and media conversion logic required for the exchange of data.

IEEE 802.11 Services

- **Association-Related Services:**
- To understand the concept of association, lets look at the three transition types, based on mobility:

IEEE 802.11 Services

- **No transition:** A station of this type is either stationary or moves only within the direct communication range of the communicating stations of a single BSS.

IEEE 802.11 Services

- **BSS transition:** This is defined as a station movement from one BSS to another BSS within the same ESS.
- Delivery of data to the station requires that the addressing capability be able to recognize the new location of the station.

IEEE 802.11 Services

- **ESS transition:** is a station movement from a BSS in one ESS to a BSS within another ESS.
- Maintenance of upper-layer connections supported by 802.11 cannot be guaranteed.

IEEE 802.11 Services

- To deliver a message within a DS, the distribution service needs to know the identity of the AP to which the message should be delivered in order for that message to reach the destination station.

IEEE 802.11 Services

- Three services relate to a station maintaining an association with the AP within its current BSS:
- **Association:** Establishes an initial association between a station and an AP.

IEEE 802.11 Services

- **Reassociation:** Enables an established association to be transferred from one AP to another, allowing a mobile station to move from one BSS to another.

IEEE 802.11 Services

- **Disassociation:** A notification from either a station or an AP that an existing association is terminated.

End

IEEE 802.11i WLAN Security

Network Security

IEEE 802.11i WLAN Security

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain basics of IEEE 802.11i WLAN Security.

IEEE 802.11i WLAN Security

Figures and material in this topic have been adapted from

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

IEEE 802.11i WLAN Security

- There are two characteristics of a wired LAN that are not inherent in a wireless LAN.

IEEE 802.11i WLAN Security

- **1.** In order to transmit over a wired LAN, a station must be physically connected to the LAN.
- On the other hand, with a wireless LAN, any station within radio range of the other devices on the LAN can transmit.

IEEE 802.11i WLAN Security

- In a sense, there is a form of authentication with a wired LAN in that it requires some positive and presumably observable action to connect a station to a wired LAN.

IEEE 802.11i WLAN Security

- **2.** To receive a transmission from a station that is part of a wired LAN, the receiving station also must be attached to the wired LAN.
- On the other hand, with a WLAN, any station within radio range can receive.

IEEE 802.11i WLAN Security

- Thus, a wired LAN provides a degree of privacy, limiting reception of data to stations connected to the LAN.

IEEE 802.11i WLAN Security

- There's an increased need for robust security services and mechanisms for wireless LANs.
- For privacy, 802.11 defined the **Wired Equivalent Privacy (WEP)** algorithm.
  - major weaknesses

IEEE 802.11i WLAN Security

- The Wi-Fi Alliance published **Wi-Fi Protected Access (WPA)**, which is a set of security mechanisms that eliminates most 802.11 security issues and was based on the current state of the 802.11i standard.

IEEE 802.11i WLAN Security

- The final form of the 802.11i standard is referred to as **Robust Security Network (RSN)**.
- The RSN specification is quite complex.

IEEE 802.11i WLAN Security

- The 802.11i RSN security specification defines the following services:

IEEE 802.11i WLAN Security

- **Authentication:** A protocol is used to define an exchange between a user and an AS that provides mutual authentication and generates temporary keys to be used between the client and the AP over the wireless link.

IEEE 802.11i WLAN Security

- **Access control:** This function enforces the use of authentication function, routes the messages properly, and facilitates key exchange.
- It can work with a variety of authentication protocols.

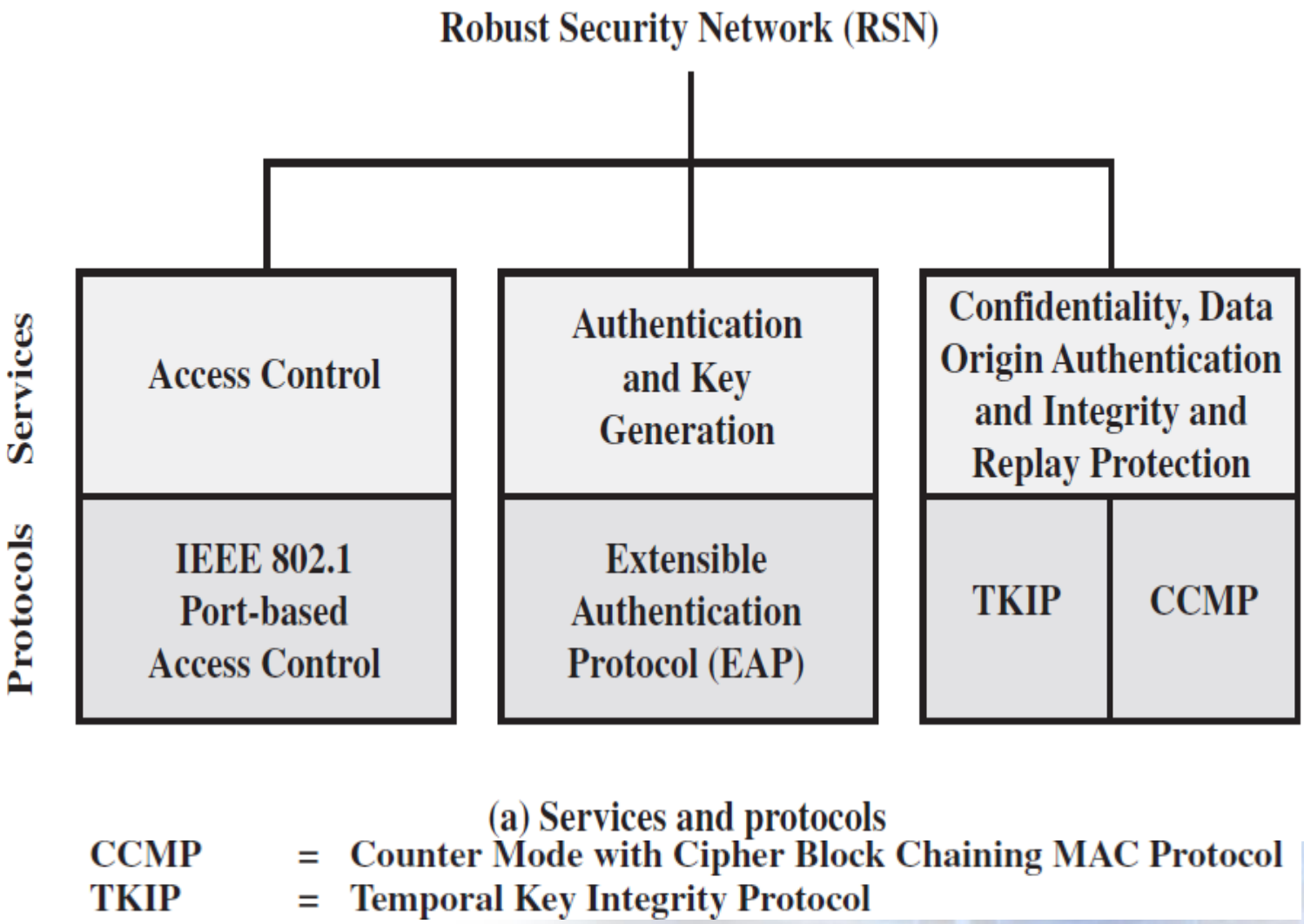
IEEE 802.11i WLAN Security

- **Privacy with message integrity:** MAC-level data (e.g., an LLC PDU) are encrypted along with a message integrity code that ensures that the data have not been altered.

IEEE 802.11i WLAN Security

- The figure on the next slide indicates the security protocols used to support these services.

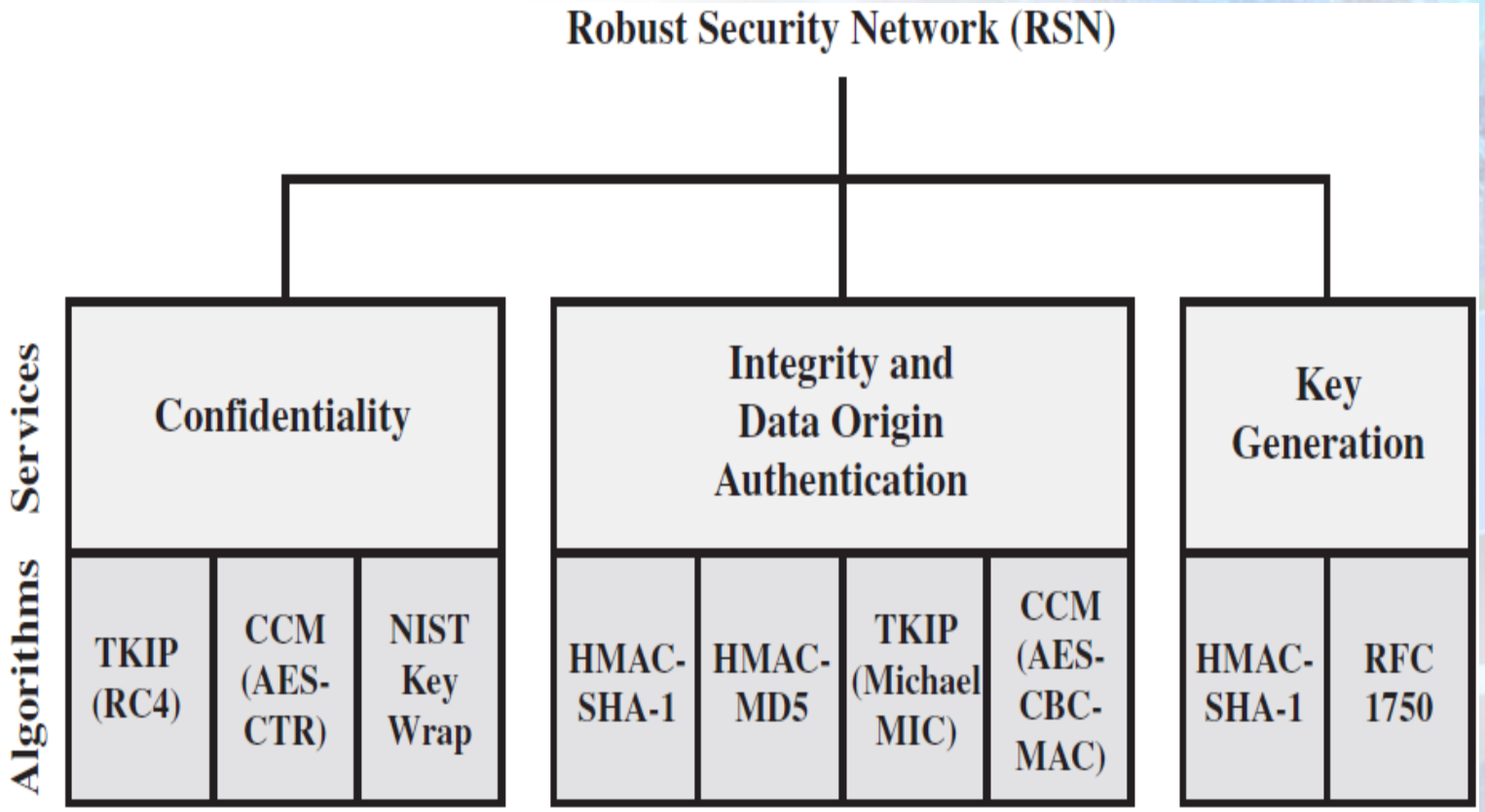
IEEE 802.11i WLAN Security



IEEE 802.11i WLAN Security

- The figure on the next slide lists the cryptographic algorithms used for these services.

IEEE 802.11i WLAN Security



(b) Cryptographic algorithms

CBC-MAC = Cipher Block Chaining Message Authentication Code (MAC)

CCM = Counter Mode with Cipher Block Chaining Message Authentication Code

IEEE 802.11i Phases of Operation



IEEE 802.11i Phases of Operation

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain IEEE 802.11i Phases of Operation.

IEEE 802.11i Phases of Operation

Figures and material in this topic have been adapted from

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

IEEE 802.11i Phases of Operation

- The operation of an IEEE 802.11i RSN can be broken down into five distinct phases of operation.
- The exact nature of the phases will depend on the configuration and the end points of the communication.

IEEE 802.11i Phases of Operation

- Possibilities include:
- **1.** Two wireless stations in the same BSS communicating via the access point (AP) for that BSS.

IEEE 802.11i Phases of Operation

- **2.** Two wireless stations (STAs) in the same ad hoc IBSS communicating directly with each other.

IEEE 802.11i Phases of Operation

- **3.** Two wireless stations in different BSSs communicating via their respective APs across a distribution system.

IEEE 802.11i Phases of Operation

- **4.** A wireless station communicating with an end station on a wired network via its AP and the distribution system.

IEEE 802.11i Phases of Operation

- IEEE 802.11i security is concerned only with secure communication between the STA and its AP.

IEEE 802.11i Phases of Operation

- In case 1 in the preceding list, secure communication is assured if each STA establishes secure communications with the AP.
- Case 2 is similar, with the AP functionality residing in the STA.

IEEE 802.11i Phases of Operation

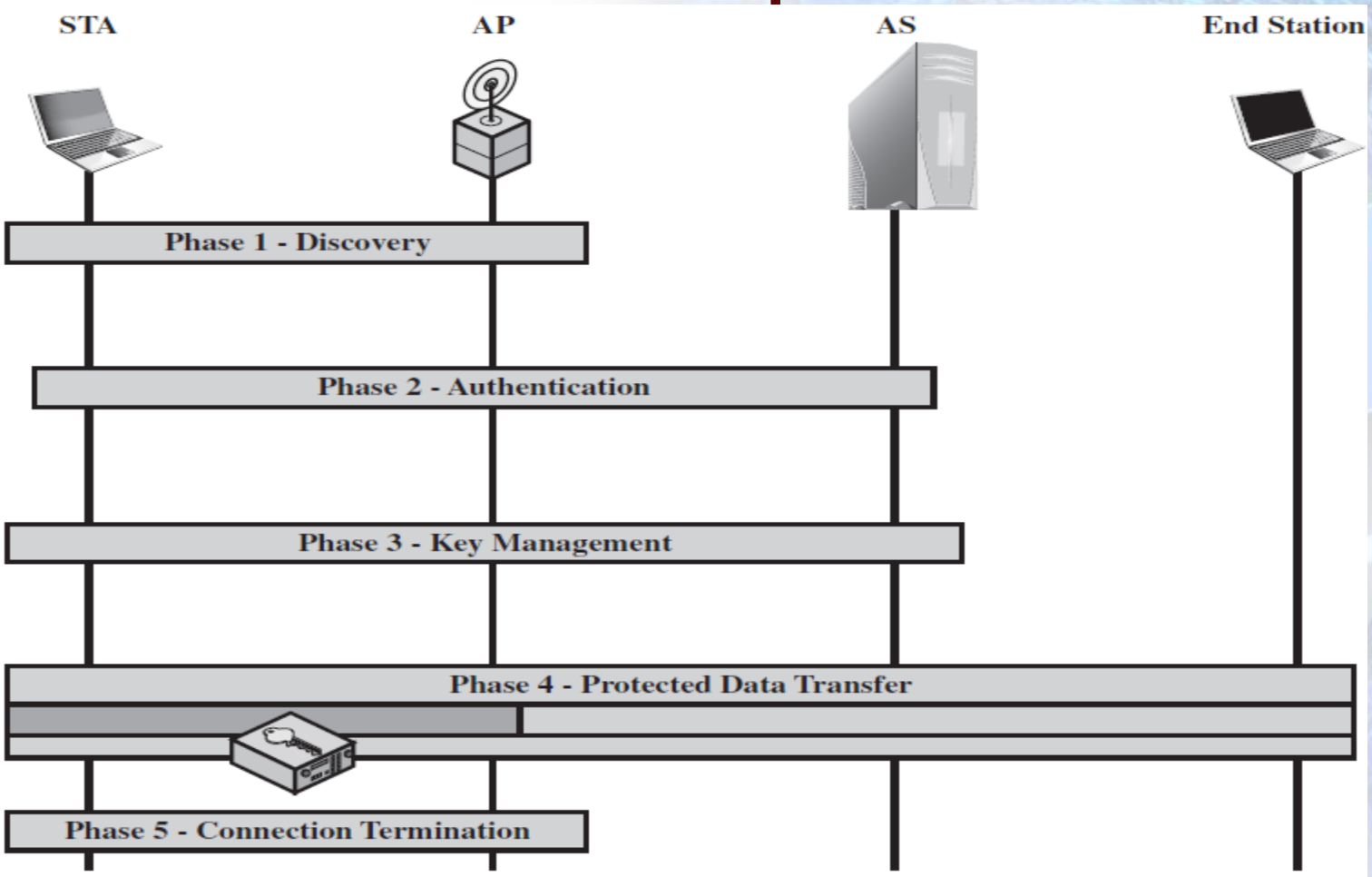
- For case 3, security is not provided across the distribution system at the level of IEEE 802.11, but only within each BSS.
- End- to-end security (if required) must be provided at a higher layer.

IEEE 802.11i Phases of Operation

- Similarly, in case 4, security is only provided between the STA and its AP.
- Next, we depict the five phases of operation for an RSN.

IEEE 802.11i Phases of Operation

IEEE 802.11i Phases of Operation:



IEEE 802.11i Phases of Operation

- **1. Discovery:** An AP uses messages called Beacons and Probe Responses to advertise its IEEE 802.11i security policy.
- STA uses these to identify an AP with which it wishes to communicate.

IEEE 802.11i Phases of Operation

- The STA associates with the AP, which it uses to select the cipher suite and authentication mechanism when the Beacons and Probe Responses present a choice.

IEEE 802.11i Phases of Operation

- **2. Authentication:** The STA and AS prove their identities to each other. The AP blocks non-authentication traffic between the STA and AS until the authentication transaction is successful.

IEEE 802.11i Phases of Operation

- **3. Key generation and distribution:** The AP and the STA perform several operations that cause cryptographic keys to be generated and placed on the AP and the STA.
- Frames are exchanged between AP and STA.

IEEE 802.11i Phases of Operation

- **4. Protected data transfer:** Frames are exchanged between the STA and the end station through the AP.
- Secure data transfer occurs between the STA and the AP only; security is not provided end-to-end.

IEEE 802.11i Phases of Operation

End

- **5. Connection termination:** The AP and STA exchange frames.
- The secure connection is torn down and the connection is restored to the original state.

Discovery Phase of IEEE 802.11i

Network Security

Discovery Phase of IEEE 802.11i

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe discovery phase of IEEE 802.11i.

Discovery Phase of IEEE 802.11i

Figures and material in this topic have been adapted from

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

Discovery Phase of IEEE 802.11i

- The purpose of this phase is for an STA and an AP to recognize each other, agree on a set of security capabilities, and establish an association for future communication using those security capabilities.

Discovery Phase of IEEE 802.11i

- **Security Capabilities** The STA and AP decide on specific techniques:
- 1.Confidentiality and MPDU integrity protocols for protecting unicast traffic (traffic only between this STA and AP).

Discovery Phase of IEEE 802.11i

- 2.Authentication method
- 3.Cryptography key management approach

Discovery Phase of IEEE 802.11i

- Confidentiality and integrity protocols for protecting multicast/broadcast traffic are dictated by the AP, since all STAs in a multicast group must use the same protocols and ciphers.

Discovery Phase of IEEE 802.11i

- The specification of a protocol, along with the chosen key length (if variable) is known as a **cipher suite**.
- The options for the confidentiality and integrity cipher suite are:

Discovery Phase of IEEE 802.11i

- WEP, with either a 40-bit or 104-bit key, which allows backward compatibility with older IEEE 802.11 implementations
- TKIP
- CCMP
- Vendor-specific methods

Discovery Phase of IEEE 802.11i

- The other negotiable suite is authentication and key management suite, which defines (1) means by which AP and STA perform mutual authentication and (2) the means for deriving a root key from which other keys may be generated.

Discovery Phase of IEEE 802.11i

- The possible authentication and key management (AKM) suites are:

Discovery Phase of IEEE 802.11i

- IEEE 802.1X
- Pre-shared key (no explicit authentication takes place and mutual authentication is implied if the STA and AP share a unique secret key)
- Vendor-specific methods

Discovery Phase of IEEE 802.11i

- **MPDU Exchange** The discovery phase consists of three exchanges.
- **1.Network and security capability discovery:** STAs discover the existence of a network with which to communicate.

Discovery Phase of IEEE 802.11i

- The AP either periodically broadcasts its security capabilities in a specific channel through the Beacon frame or responds to a station's Probe Request through a Probe Response frame.

Discovery Phase of IEEE 802.11i

- A wireless station may discover available access points and corresponding security capabilities by either passively monitoring the Beacon frames or actively probing every channel.

Discovery Phase of IEEE 802.11i

- **2.Open system authentication:**
- The two devices (STA and AP) simply exchange identifiers.
- The purpose is simply to maintain backward compatibility with the IEEE 802.11.

Discovery Phase of IEEE 802.11i

- **3.Association:**
- The purpose of this stage is to agree on a set of security capabilities to be used.
- The STA then sends an Association Request frame to the AP.

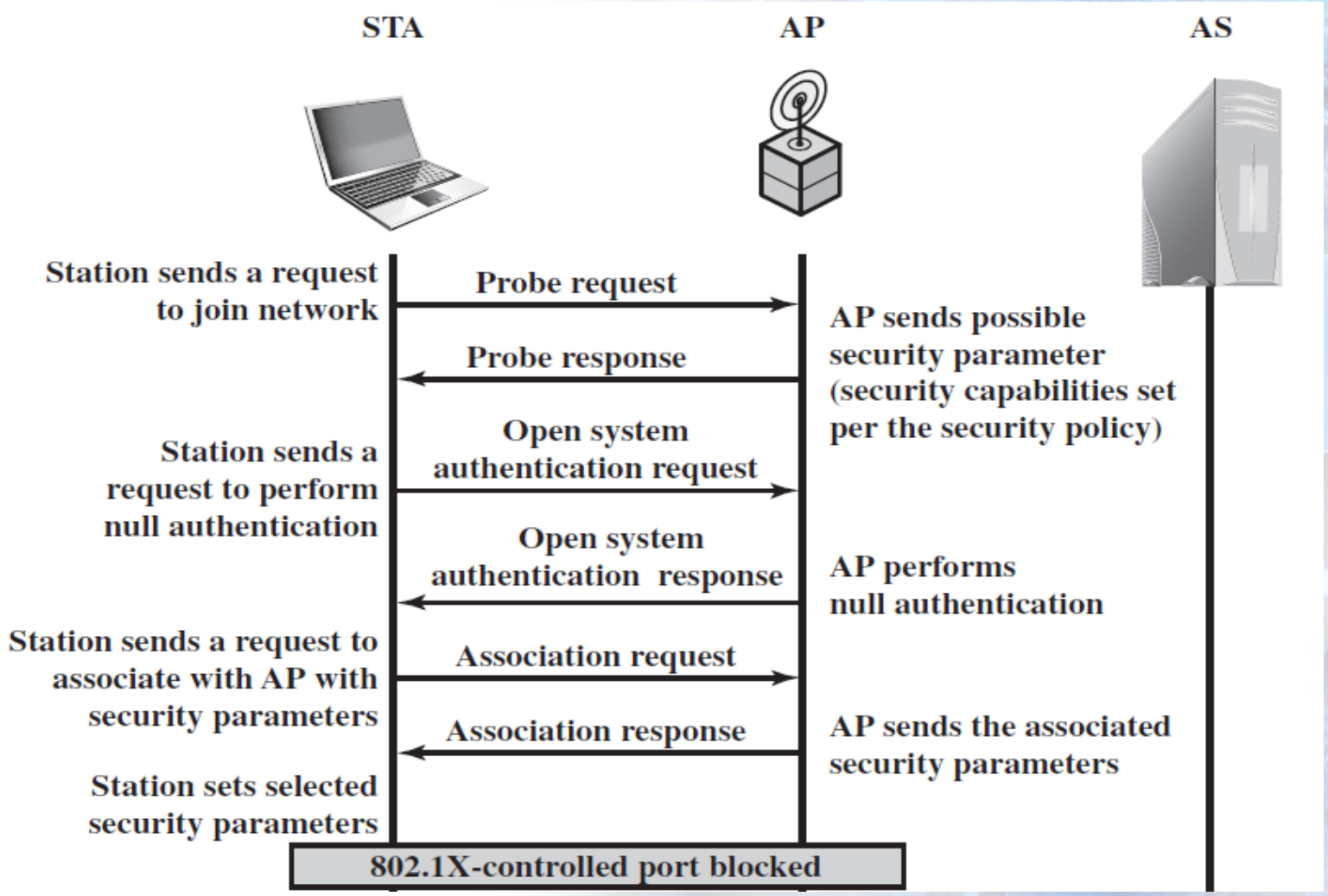
Discovery Phase of IEEE 802.11i

- In this frame, the STA specifies one set of matching capabilities (one authentication and key management suite, one pairwise cipher suite, and one group-key cipher suite) from among those advertised by the AP.

Discovery Phase of IEEE 802.11i

- If there is no match in capabilities between the AP and the STA, the AP refuses the Association Request.
- The STA blocks it too, in case it has associated with a rogue AP or someone is inserting frames illicitly on its channel.

Discovery Phase of IEEE 802.11i



Authentication Phase of IEEE 802.11i

Network Security

Authentication Phase of IEEE 802.11i

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe authentication phase of IEEE 802.11i.

Authentication Phase of IEEE 802.11i

Figures and material in this topic have been adapted from

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

Authentication Phase of IEEE 802.11i

- The authentication phase enables mutual authentication between an STA and an authentication server (AS) located in the DS.

Authentication Phase of IEEE 802.11i

- **IEEE 802.1X Access Control Approach:**
- IEEE 802.11i makes use of IEEE 802.1X, Port-Based Network Access Control to provide access control functions for LANs.

Authentication Phase of IEEE 802.11i

- Extensible Authentication Protocol (EAP) is the authentication protocol that is used and is defined in the IEEE 802.1X standard.

Authentication Phase of IEEE 802.11i

- IEEE 802.1X uses the terms *supplicant*, *authenticator*, and *authentication server* (AS).
- In the context of an 802.11 WLAN, the first two terms correspond to the wireless station and the AP.

Authentication Phase of IEEE 802.11i

- The AS is typically a separate device on the wired side of the network (i.e., accessible over the DS) but could also reside directly on the authenticator.

Authentication Phase of IEEE 802.11i

- For a WLAN, the authenticator (the AP) may have only two physical ports:
- one connecting to the DS and one for wireless communication within its BSS.

Authentication Phase of IEEE 802.11i

- Each logical port (802.1X) is mapped to one of these two physical ports.
- An uncontrolled port allows the exchange of PDUs between the supplicant and the other AS, regardless of authentication state of the supplicant.

Authentication Phase of IEEE 802.11i

- A controlled port allows the exchange of PDUs between a supplicant and other systems on the LAN only if the current state of the supplicant authorizes such an exchange.

Authentication Phase of IEEE 802.11i

- **MPDU Exchange** Following are the three phases:
- **1. Connect to AS:** The STA sends a request to its AP for connection to the AS.
- The AP acknowledges this request and sends an access request to the AS.

Authentication Phase of IEEE 802.11i

- **2.EAP exchange:** This exchange authenticates the STA and AS to each other.
- **3.Secure key delivery:** AS generates a master session key (MSK) or the Authentication, Authorization, and Accounting (AAA) key and sends it to STA.

Authentication Phase of IEEE 802.11i

- All the cryptographic keys needed by STA for secure communication with its AP are generated from this MSK.
- For secure delivery of MSK, an MPDU contains an encrypted MSK from the AS, via the AP, to the STA.

Authentication Phase of IEEE 802.11i

- **EAP Exchange:** Typically, message flow between STA and AP employs EAP over LAN (EAPOL), and the message flow between the AP and AS uses the Remote Authentication Dial In User Service (RADIUS) protocol.

Authentication Phase of IEEE 802.11i

- **1.** The AP issues an EAP-Request/Identity frame to the STA.
- **2.** The STA replies with an EAP-Response/Identity frame, which the AP receives over the uncontrolled port.

Authentication Phase of IEEE 802.11i

- The packet is then encapsulated in RADIUS over EAP and passed on to the RADIUS server as a RADIUS-Access-Request packet.

Authentication Phase of IEEE 802.11i

- **3.** The AAA server replies with a RADIUS-Access-Challenge packet, which is passed on to the STA as an EAP-Request.
- This request is of the appropriate authentication type.

Authentication Phase of IEEE 802.11i

- **4.** The STA formulates an EAP-Response message and sends it to the AS.
- The response is translated by the AP into a Radius-Access-Request with the response to the challenge as a data field.

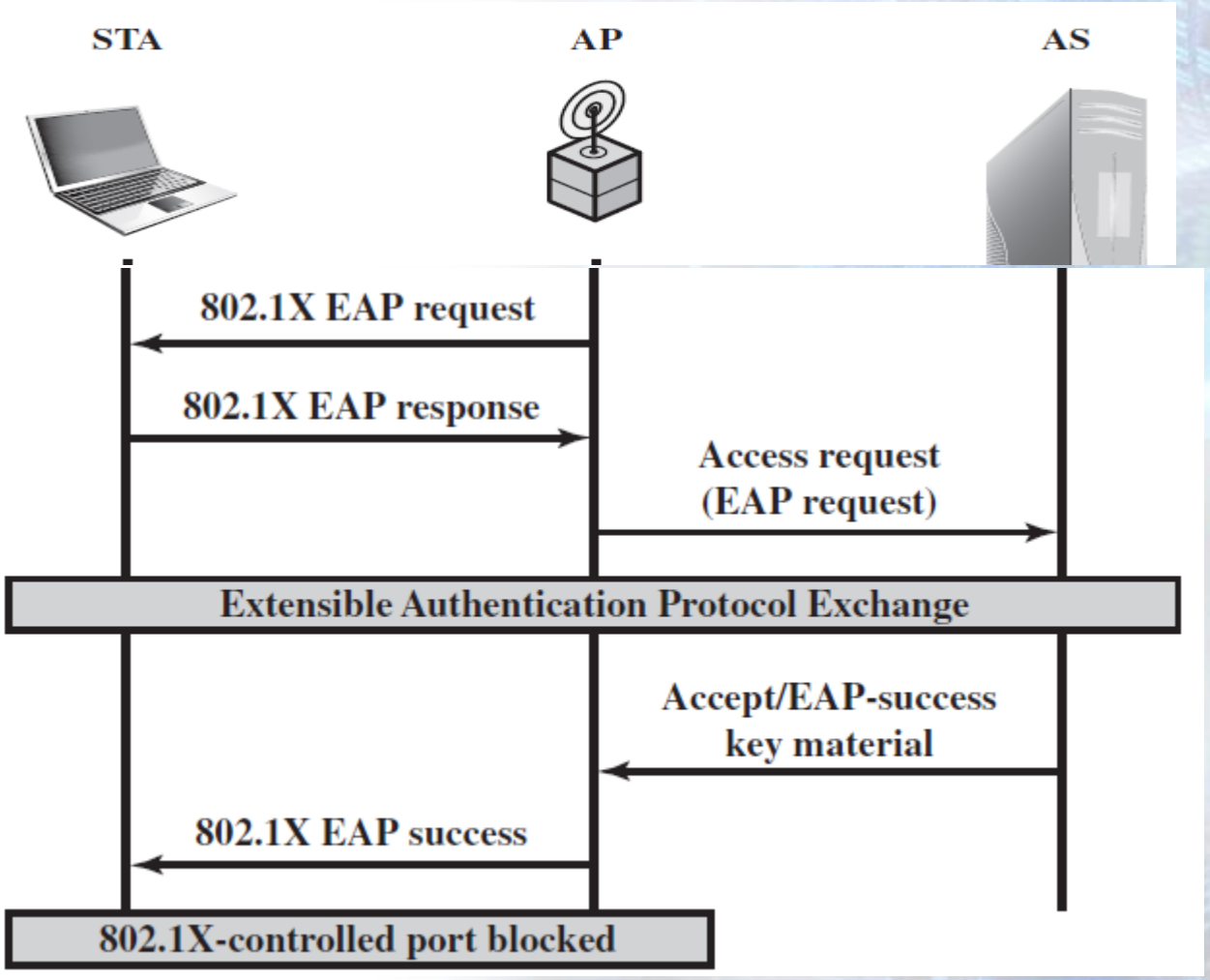
Authentication Phase of IEEE 802.11i

- Steps 3 and 4 may be repeated multiple times, depending on the EAP method in use.
- **5.** The AAA server grants access with a Radius-Access-Accept packet.
- The AP issues an EAP-Success frame.

Authentication Phase of IEEE 802.11i

- The controlled port is authorized, and the user may begin to access the network.

Authentication Phase of IEEE 802.11i



Key Management Phase of IEEE 802.11i

Network Security

Key Management Phase of IEEE 802.11i

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe Key Management phase of IEEE 802.11i.

Key Management Phase of IEEE 802.11i

Figures and material in this topic have been adapted from

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

Key Management Phase of IEEE 802.11i

- During the key management phase, a variety of cryptographic keys are generated and distributed to STAs.
- Two types of keys:
- pairwise keys and
- group keys.

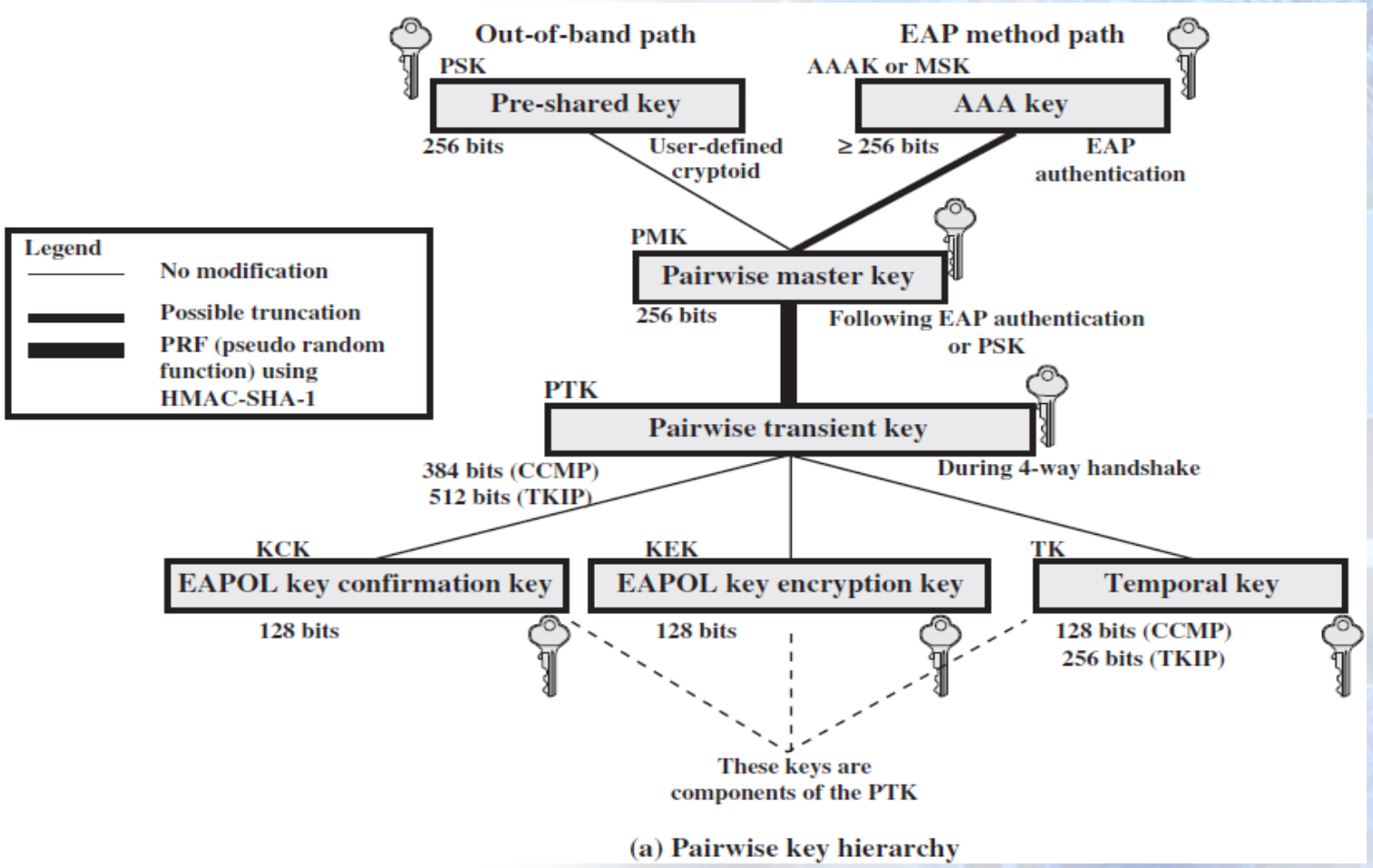
Key Management Phase of IEEE 802.11i

- **Pairwise Keys:** are used for communication between a pair of devices, typically between an STA and an AP.

Key Management Phase of IEEE 802.11i

- These keys form a hierarchy beginning with a master key from which other keys are derived dynamically and used for a limited period of time.

Key Management Phase of IEEE 802.11i



(a) Pairwise key hierarchy

Key Management Phase of IEEE 802.11i

- At the top level of the hierarchy, there are two possibilities.
- A **pre-shared key (PSK)** is a secret key shared by the AP and a STA and installed in some fashion outside the scope of IEEE 802.11i.

Key Management Phase of IEEE 802.11i

- The other alternative is the **master session key (MSK)**, also known as the AAAK, which is generated using the IEEE 802.1X protocol during the authentication phase.

Key Management Phase of IEEE 802.11i

- In either case (PSK or MSK), there is a unique key shared by the AP with each STA with which it communicates.
- All the other keys derived from this master key are also unique between an AP and an STA.

Key Management Phase of IEEE 802.11i

- The **pairwise master key (PMK)** is derived from the master key.
- If a PSK is used, then the PSK is used as the PMK.
- If a MSK is used, then the PMK is derived from the MSK by truncation.

Key Management Phase of IEEE 802.11i

- The PMK is used to generate the **pairwise transient key (PTK)**.
- PTK consists of three keys to be used for communication between an STA and AP after they have been mutually authenticated.

Key Management Phase of IEEE 802.11i

- Using the STA and AP addresses in the generation of the PTK provides protection against session hijacking and impersonation.
- Using nonces provides additional random keying material.

Key Management Phase of IEEE 802.11i

- The three parts of the PTK are as follows:

Key Management Phase of IEEE 802.11i

- **EAP Over LAN (EAPOL) Key Confirmation Key (EAPOL-KCK):**
- Supports the integrity and data origin authenticity of STA-to-AP control frames during operational setup of an RSN.

Key Management Phase of IEEE 802.11i

- It also performs an access control function: proof-of-possession of the PMK.
- An entity that possesses the PMK is authorized to use the link.

Key Management Phase of IEEE 802.11i

- **EAPOL Key Encryption Key (EAPOL-KEK)**
- Protects the confidentiality of keys and other data during some RSN association procedures.
- **Temporal Key (TK)**
- Provides the actual protection for user traffic.

Key Management Phase of IEEE 802.11i

- **Group Keys:** Group keys are used for multicast communication in which one STA sends MPDUs to multiple STAs.
- At the top level of the group key hierarchy is the **group master key (GMK)**.

Key Management Phase of IEEE 802.11i

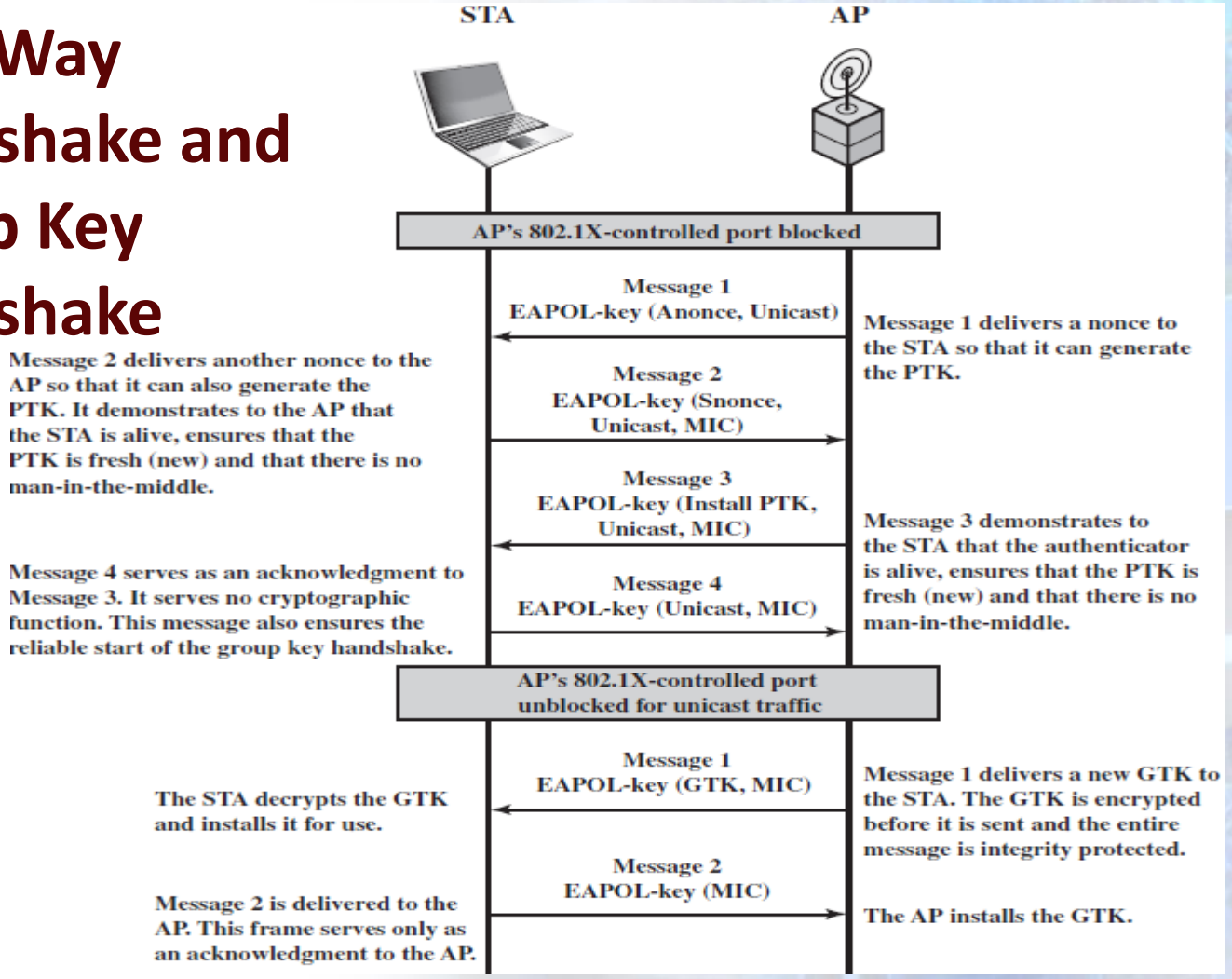
- **Group master key (GMK):** Key-generating key used with other inputs to derive the GTK
- **Group temporal key (GTK):** Generated by the AP and transmitted to its associated STAs.

Key Management Phase of IEEE 802.11i

- IEEE 802.11i requires that GTK's value is computationally indistinguishable from random.
- Distributed securely using pairwise keys already established.
- Is changed every time a device leaves network.

Key Management Phase of IEEE 802.11i

Four-Way Handshake and Group Key Handshake



Protected Data Transfer of IEEE 802.11i

Network Security

Protected Data Transfer of IEEE 802.11i

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe Protected Data Transfer phase of IEEE 802.11i.

Protected Data Transfer of IEEE 802.11i

Figures and material in this topic have been adapted from

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

Protected Data Transfer of IEEE 802.11i

- IEEE 802.11i defines two schemes for protecting data transmitted in 802.11 MPDUs:
- the Temporal Key Integrity Protocol (TKIP), and
- the Counter Mode-CBC MAC Protocol (CCMP).

Protected Data Transfer of IEEE 802.11i

- **TKIP:**
- TKIP is designed to require only software changes to devices that are implemented with the older wireless LAN security approach called Wired Equivalent Privacy (WEP).

Protected Data Transfer of IEEE 802.11i

- TKIP provides two services:
- **Message integrity:** TKIP adds a message integrity code (MIC) to the 802.11 MAC frame after the data field.

Protected Data Transfer of IEEE 802.11i

- The MIC is generated by an algorithm, called Michael.
- It computes a 64-bit value using as input the source and destination MAC address values and the Data field, plus key material.

Protected Data Transfer of IEEE 802.11i

- **Data confidentiality:**
- is provided by encrypting the MPDU plus MIC value using RC4.
- The 256-bit Temporal Key (TK) is employed as follows:

Protected Data Transfer of IEEE 802.11i

- Two 64-bit keys are used with the Michael message digest algorithm to produce a message integrity code.
- One key is used to protect STA-to-AP messages, and the other key is used to protect AP-to-STA.

Protected Data Transfer of IEEE 802.11i

- The remaining 128 bits are truncated to generate the RC4 key used to encrypt the transmitted data.

Protected Data Transfer of IEEE 802.11i

- For additional protection, a monotonically increasing TKIP sequence counter (TSC) is assigned to each frame.
- The TSC serves two purposes.

Protected Data Transfer of IEEE 802.11i

- First, the TSC is included with each MPDU and is protected by the MIC to protect against replay attacks.

Protected Data Transfer of IEEE 802.11i

- Second, the TSC is combined with the session TK to produce a dynamic encryption key that changes with each transmitted MPDU, thus making cryptanalysis more difficult.

Protected Data Transfer of IEEE 802.11i

- **CCMP:** CCMP is intended for newer IEEE 802.11 devices that are equipped with the hardware to support this scheme.
- As with TKIP, CCMP provides two services:

Protected Data Transfer of IEEE 802.11i

- **Message integrity:**
- CCMP uses the cipher block chaining message authentication code (CBC-MAC).

Protected Data Transfer of IEEE 802.11i

- **Data confidentiality:**
- CCMP uses the CTR block cipher mode of operation with AES for encryption.

Protected Data Transfer of IEEE 802.11i

End

- The same 128-bit AES key is used for both integrity and confidentiality.
- The scheme uses a 48-bit packet number to construct a nonce to prevent replay attacks.

IEEE 802.11i Pseudorandom Function

Network Security

IEEE 802.11i Pseudorandom Function

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe pseudorandom function of IEEE 802.11i.

IEEE 802.11i Pseudorandom Function

Figures and material in this topic have been adapted from

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

IEEE 802.11i Pseudorandom Function

- At a number of places in the IEEE 802.11i scheme, a pseudorandom function (PRF) is used.
- For example, it is used to generate nonces, to expand pairwise keys, and to generate the group temporal key (GTK).

IEEE 802.11i Pseudorandom Function

- Best security practice dictates that different pseudorandom number streams be used for these different purposes.

IEEE 802.11i Pseudorandom Function

- However, for implementation efficiency, we would like to rely on a single pseudorandom number generator function.

IEEE 802.11i Pseudorandom Function

- The PRF is built on the use of HMAC-SHA-1 to generate a pseudorandom bit stream.
- HMAC-SHA-1 takes a message (block of data) and a key of length at least 160 bits and produces a 160-bit hash value.

IEEE 802.11i Pseudorandom Function

- SHA-1 has the property that the change of a single bit of the input produces a new hash value with no apparent connection to the preceding hash value.

IEEE 802.11i Pseudorandom Function

- This property is the basis for pseudorandom number generation.

IEEE 802.11i Pseudorandom Function

- The IEEE 802.11i PRF takes four parameters as input and produces the desired number of random bits.
- The function is of the form PRF(*K*, *A*, *B*, *Len*), where
- *K* = a secret key

IEEE 802.11i Pseudorandom Function

- *A* = a text string specific to the application (e.g., nonce generation or pairwise key expansion)
- *B* = some data specific to each case
- *Len* = desired number of pseudorandom bits

IEEE 802.11i Pseudorandom Function

As an example, for the pairwise transient key (PTK) for Counter Mode-CBC MAC Protocol (CCMP):

```
PTK = PRF(PMK,"Pairwise key expansion",min(AP-
      Addr,STA-Addr) || max(AP-Addr,STA-Addr) || min
      (Anonce,Snonce) || max(Anonce,Snonce),384)
```

IEEE 802.11i Pseudorandom Function

- *K* = Pairwise Master Key (PMK)
- *A* = the text string "Pairwise key expansion"
- *B* = a sequence of bytes formed by concatenating the two MAC addresses and the two nonces
- *Len* = 384 bits

IEEE 802.11i Pseudorandom Function

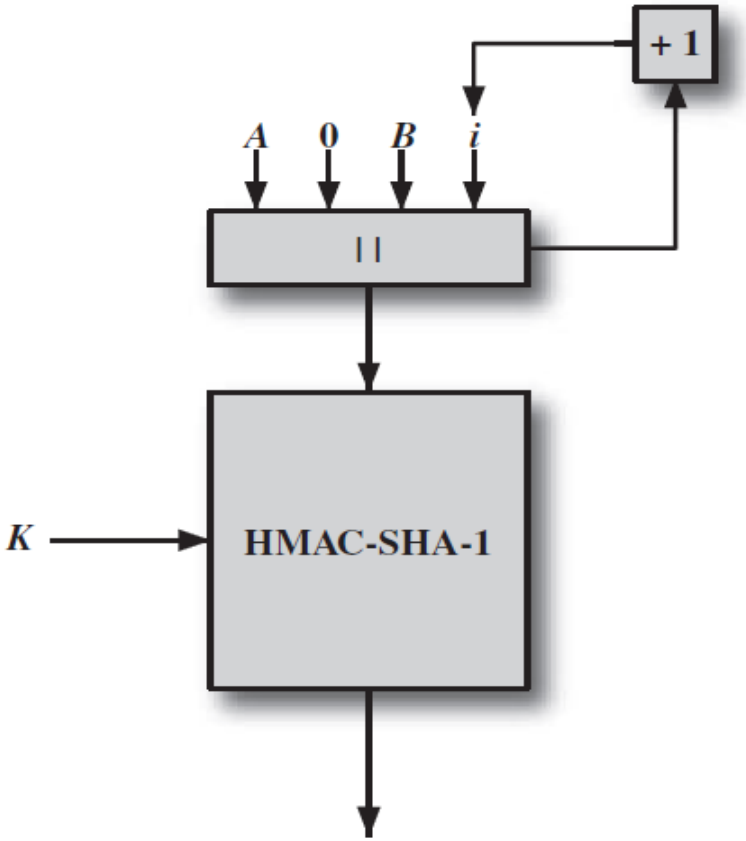
Anther example, a nonce can be generated as:

```
Nonce = PRF(Random Number,"InitCounter",MAC || Time, 256)
```

where **Time** is a measure of the network time known to the nonce generator.

IEEE 802.11i Pseudorandom Function

IEEE 802.11i Pseudorandom Function



$R = \text{HMAC-SHA-1}(K, A \| 0 \| B \| i)$

IEEE 802.11i Pseudorandom Function

- *K* serves as the key input to HMAC.
- The message input consists of four items concatenated together:
- the parameter *A*, a byte with value 0, the parameter *B*, and a counter *i*, which is initialized to 0.

IEEE 802.11i Pseudorandom Function

End

- The HMAC algorithm is run once, producing a 160-bit hash value.
- If more bits are required, HMAC is run again with the same inputs, except that *i* is incremented each time until the necessary number of bits is generated.

Introduction to Pretty Good Privacy

Network Security

Introduction to Pretty Good Privacy

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain basics of Pretty Good Privacy (PGP).

Introduction to Pretty Good Privacy

Figures and material in this topic have been adapted from

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

Introduction to Pretty Good Privacy

- PGP provides a confidentiality and authentication service that can be used for electronic mail and file storage applications.

Introduction to Pretty Good Privacy

- PGP is a remarkable phenomenon.
- Largely it is the effort of a single person, Phil Zimmermann.
- In essence, Zimmermann has done the following:

Introduction to Pretty Good Privacy

- 1. Selected the best available cryptographic algorithms as building blocks.

Introduction to Pretty Good Privacy

- 2. Integrated these algorithms into a general-purpose application that is independent of operating system and processor and that is based on a small set of easy-to-use commands.

Introduction to Pretty Good Privacy

- 3. Made the package and its documentation, including the source code, freely available via the Internet, bulletin boards, and commercial networks such as AOL (America On Line).

Introduction to Pretty Good Privacy

- 4. Entered into an agreement with a company (Viacrypt, now Network Associates) to provide a fully compatible, low-cost commercial version of PGP.

Introduction to Pretty Good Privacy

- PGP has grown explosively and is now widely used.
- A number of reasons can be cited for this growth.

Introduction to Pretty Good Privacy

- **1.** It is available free worldwide in versions that run on a variety of platforms, including Windows, UNIX, Macintosh, etc.
- Also, commercial version satisfies users who want a product that comes with vendor support.

Introduction to Pretty Good Privacy

- **2.** It is based on algorithms that have survived extensive public review and are considered extremely secure.

Introduction to Pretty Good Privacy

- Specifically, the package includes RSA, DSS, and Diffie-Hellman for public-key encryption
- CAST-128, IDEA, and 3DES for symmetric encryption, and
- SHA-1 for hash coding.

Introduction to Pretty Good Privacy

- **3.** It has a wide range of applicability, from corporations that wish to select a standardized scheme for encrypting files to individuals who wish to communicate securely with others worldwide over the Internet/networks.

Introduction to Pretty Good Privacy

- **4.** It was not developed by, nor is it controlled by, any governmental or standards organization.
- This makes PGP attractive.

Introduction to Pretty Good Privacy

- **5.** PGP is now on an Internet standards track (RFC 3156; *MIME Security with OpenPGP*).

Introduction to Pretty Good Privacy

- The actual operation of PGP, as opposed to the management of keys, consists of four services:
- authentication,
- confidentiality,
- compression, and
- e-mail compatibility

Introduction to Pretty Good Privacy

Summary of PGP Services

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

PGP Authentication Service

Network Security

PGP Authentication Service

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe working of PGP authentication service.

PGP Authentication Service

Figures and material in this topic have been adapted from

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

PGP Authentication Service

- PGP provides a confidentiality and authentication service that can be used for electronic mail and file storage applications.

PGP Authentication Service

- **Authentication:**
- **1.** The sender creates a message.
- **2.** SHA-1 is used to generate a 160-bit hash code of the message.

PGP Authentication Service

- **3.** The hash code is encrypted with RSA using the sender's private key, and the result is prepended to the message.
- **4.** The receiver uses RSA with the sender's public key to decrypt and recover the hash code.

PGP Authentication Service

- **5.** The receiver generates a new hash code for the message and compares it with the decrypted hash code.
- If the two match, the message is accepted as authentic.
- Next, we show it.

PGP Authentication Service

- First, lets look at the symbols that are used.
- *Ks* = session key used in symmetric encryption scheme
- *PRa* = private key of user A, used in public-key encryption scheme

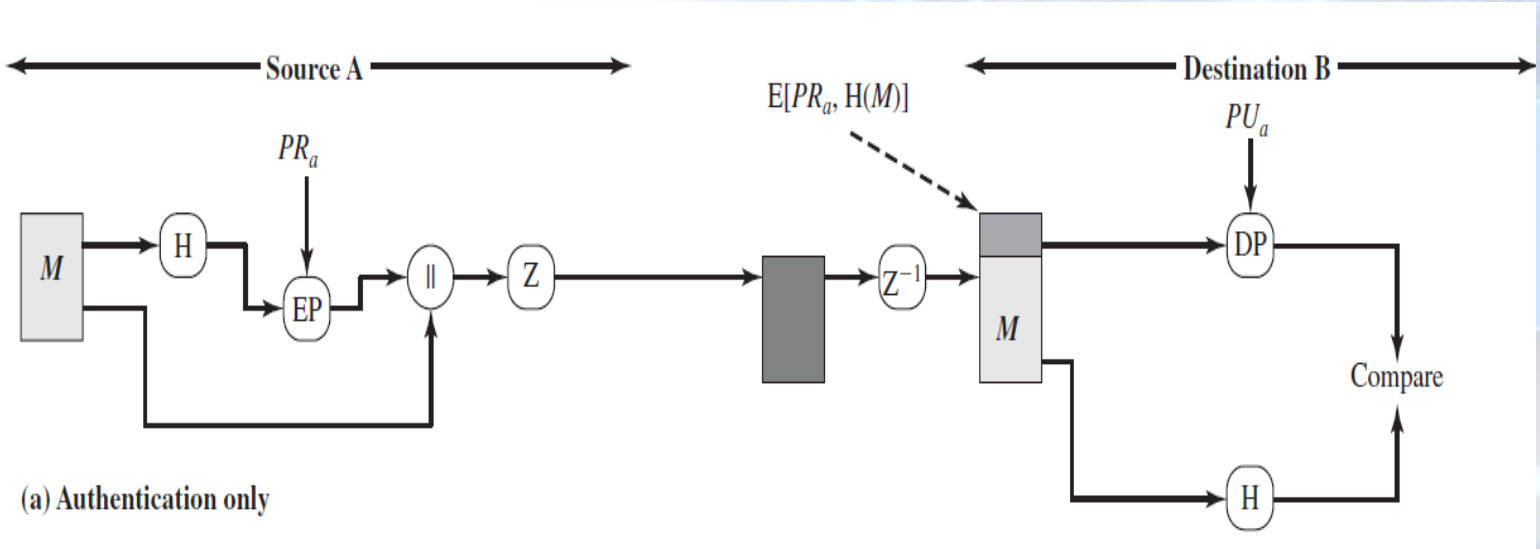
PGP Authentication Service

- *PUa* = public key of user A, used in public-key encryption scheme
- EP = public-key encryption
- DP = public-key decryption
- EC = symmetric encryption

PGP Authentication Service

- DC = symmetric decryption
- H = hash function
- || = concatenation
- Z = compression using ZIP algorithm
- R64 = conversion to radix 64 ASCII format

PGP Authentication Service



(a) Authentication only

PGP Authentication Service

- The combination of SHA-1 and RSA provides an effective digital signature scheme.

PGP Authentication Service

- Because of the strength of RSA, the recipient is assured that only the possessor of the matching private key can generate the signature.

PGP Authentication Service

- Because of the strength of SHA-1, the recipient is assured that no one else could generate a new message that matches the hash code and, hence, the signature of the original message.

PGP Authentication Service

- As an alternative, signatures can be generated using DSS/SHA-1.

PGP Authentication Service

- Although signatures normally are found attached to the message or file that they sign, this is not always the case:
- Detached signatures are supported.

PGP Authentication Service

- A detached signature may be stored and transmitted separately from the message it signs.
- This is useful.
- A user may wish to maintain a separate signature log of all messages sent or received.

PGP Authentication Service

- A detached signature of an executable program can detect subsequent virus infection.
- Finally, detached signatures can be used when more than one party must sign a document, such as a legal contract.

PGP Authentication Service

End

- Each person's signature is independent and is applied only to doc.
- Otherwise, signatures would have to be nested, with the second signer signing both the document and the first signature, and so on.

PGP Confidentiality Service

Network Security

PGP Confidentiality Service

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe working of PGP confidentiality service.

PGP Confidentiality Service

Figures and material in this topic have been adapted from

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

PGP Confidentiality Service

- PGP confidentiality is provided by encrypting messages to be transmitted or to be stored locally as files.
- In both cases, the symmetric encryption algorithm CAST-128 may be used.

PGP Confidentiality Service

- Alternatively, International Data Encryption Algorithm (IDEA) or 3DES may be used.
- The 64-bit cipher feedback (CFB) mode is used.

PGP Confidentiality Service

- In PGP, each symmetric key is used only once.
- A new key is generated as a random 128-bit number for each message.

PGP Confidentiality Service

- This is referred to in the documentation as a session key, it is in reality a one-time key.
- The session key is bound to the message and transmitted with it.
- To protect the key, it is encrypted with the receiver's public key.

PGP Confidentiality Service

- The sequence for confidentiality is:
- **1.** The sender generates a message and a random 128-bit number to be used as a session key for this message only.

PGP Confidentiality Service

- **2.** The message is encrypted using CAST-128 (or IDEA or 3DES) with the session key.
- **3.** The session key is encrypted with RSA using the recipient's public key and is prepended to the message.

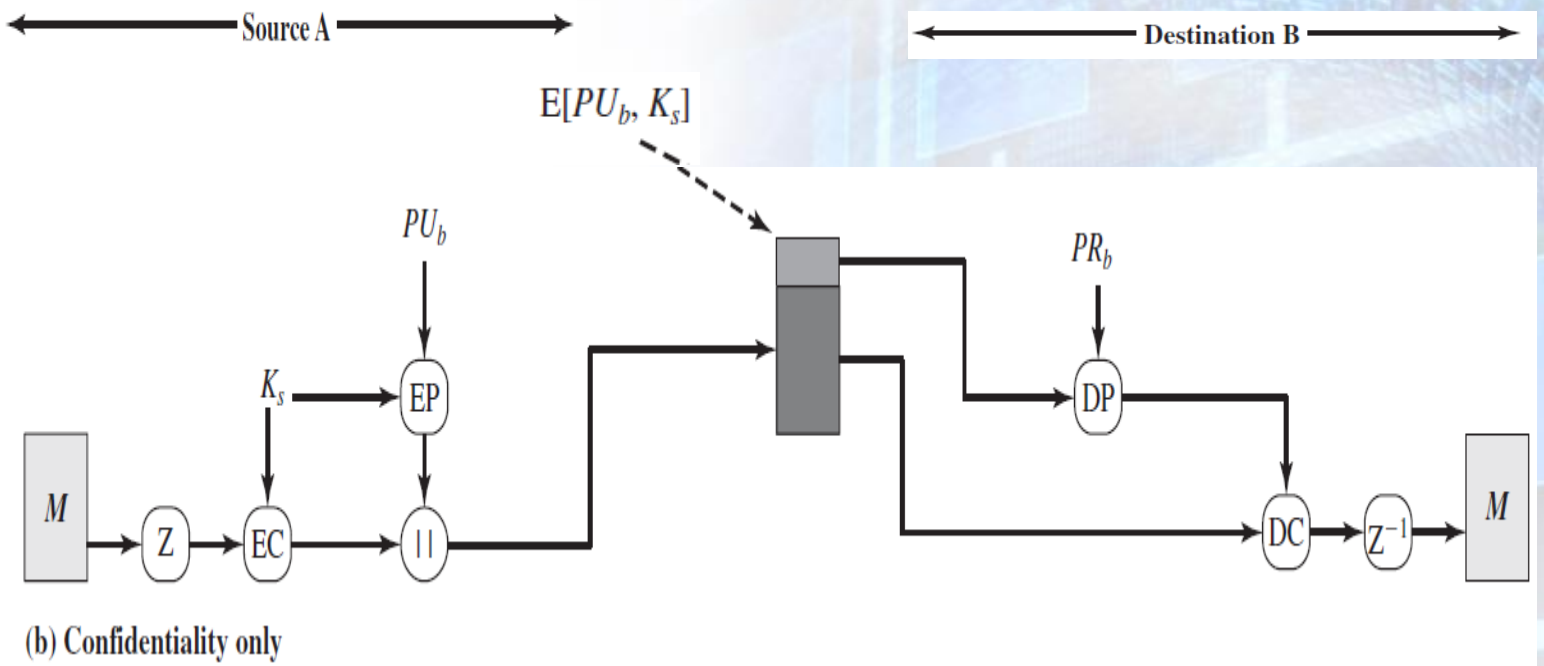
PGP Confidentiality Service

- **4.** The receiver uses RSA with its private key to decrypt and recover the session key.
- **5.** The session key is used to decrypt the message.

PGP Confidentiality Service

- As an alternative to the use of RSA for key encryption, PGP provides an option referred to as *Diffie-Hellman*.
- PGP uses a variant of Diffie-Hellman that does provide encryption/decryptio, known as ElGamal.

PGP Confidentiality Service



(b) Confidentiality only

Ks = session key used in symmetric encryption scheme
EP = public-key encryption, DP = public-key decryption ,
EC = symmetric encryption, DC = symmetric decryption
Z = compression using ZIP algorithm

PGP Confidentiality Service

- Several observations may be made.
- First, to reduce encryption time, the combination of symmetric and public-key encryption is used in preference to simply using RSA or ElGamal to encrypt the message directly.

PGP Confidentiality Service

- CAST-128 and the other symmetric algorithms are substantially faster than RSA or ElGamal.

PGP Confidentiality Service

- Second, the use of the public-key algorithm solves the session-key distribution problem, because only the recipient is able to recover the session key that is bound to the message.

PGP Confidentiality Service

- Finally, the use of one-time symmetric keys strengthens what is already a strong symmetric encryption approach.

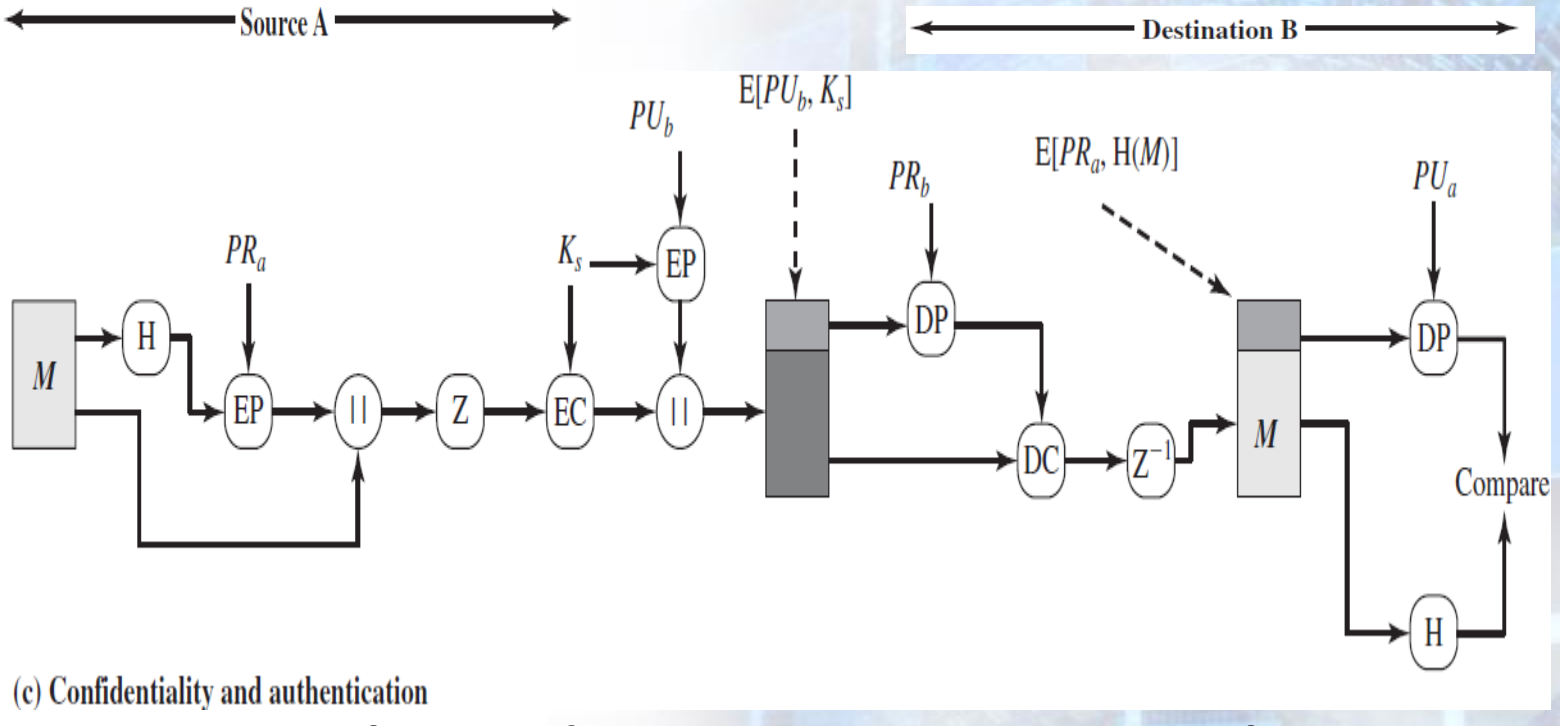
PGP Confidentiality Service

- **Confidentiality and Authentication:**
- Both services may be used for the same message.
- First a signature is generated for the plaintext message and prepended to the message.

PGP Confidentiality Service

- Then the plaintext message plus signature is encrypted using CAST-128 (or IDEA or 3DES) and the session key is encrypted using RSA (or ElGamal).

PGP Confidentiality Service



(c) Confidentiality and authentication

Ks = session key used in symmetric encryption scheme

EP = public-key encryption, DP = public-key decryption ,

EC = symmetric encryption, DC = symmetric decryption

Z = compression using ZIP algorithm

PGP Compression & E-Mail Compatibility

Network Security

PGP Compression & E-Mail Compatibility

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe PGP compression service and E-mail Compatibility.

PGP Compression & E-Mail Compatibility

Figures and material in this topic have been adapted from

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

PGP Compression & E-Mail Compatibility

- As a default, PGP compresses the message after applying the signature but before encryption.
- This has the benefit of saving space both for e-mail transmission and for file storage.

PGP Compression & E-Mail Compatibility

- The placement of the compression algorithm is critical.
- **1.** The signature is generated before compression for two reasons:

PGP Compression & E-Mail Compatibility

- **a.** It is preferable to sign an uncompressed message so that one can store only the uncompressed message together with the signature for future verification.

PGP Compression & E-Mail Compatibility

- If one signed a compressed document, then it would be necessary either to store a compressed version of the message for later verification or to recompress message when verification is required.

PGP Compression & E-Mail Compatibility

- **b.** Even if one were willing to generate dynamically a recompressed message for verification, PGP's compression algorithm presents a difficulty.

PGP Compression & E-Mail Compatibility

- The algorithm is not deterministic.
- Various implementations of the algorithm achieve different tradeoffs in running speed versus compression ratio and, as a result, produce different compressed forms.

PGP Compression & E-Mail Compatibility

- These different compression algorithms are interoperable because any version of the algorithm can correctly decompress the output of any other version.

PGP Compression & E-Mail Compatibility

- Applying the hash function and signature after compression would constrain all PGP implementations to the same version of the compression algorithm.

PGP Compression & E-Mail Compatibility

- **2.** Message encryption is applied after compression to strengthen cryptographic security.
- Compressed message has less redundancy than original plaintext, cryptanalysis is more difficult.

PGP Compression & E-Mail Compatibility

- The compression algorithm used is ZIP.

PGP Compression & E-Mail Compatibility

- **E-mail Compatibility:**
- Many electronic mail systems only permit the use of blocks consisting of ASCII text.
- PGP provides service of converting raw 8-bit binary stream to a stream of printable ASCII characters.

PGP Compression & E-Mail Compatibility

- The scheme used for this purpose is radix-64 conversion.
- Each group of three octets of binary data is mapped into four ASCII characters.
- This format also appends a CRC to detect transmission errors.

PGP Compression & E-Mail Compatibility

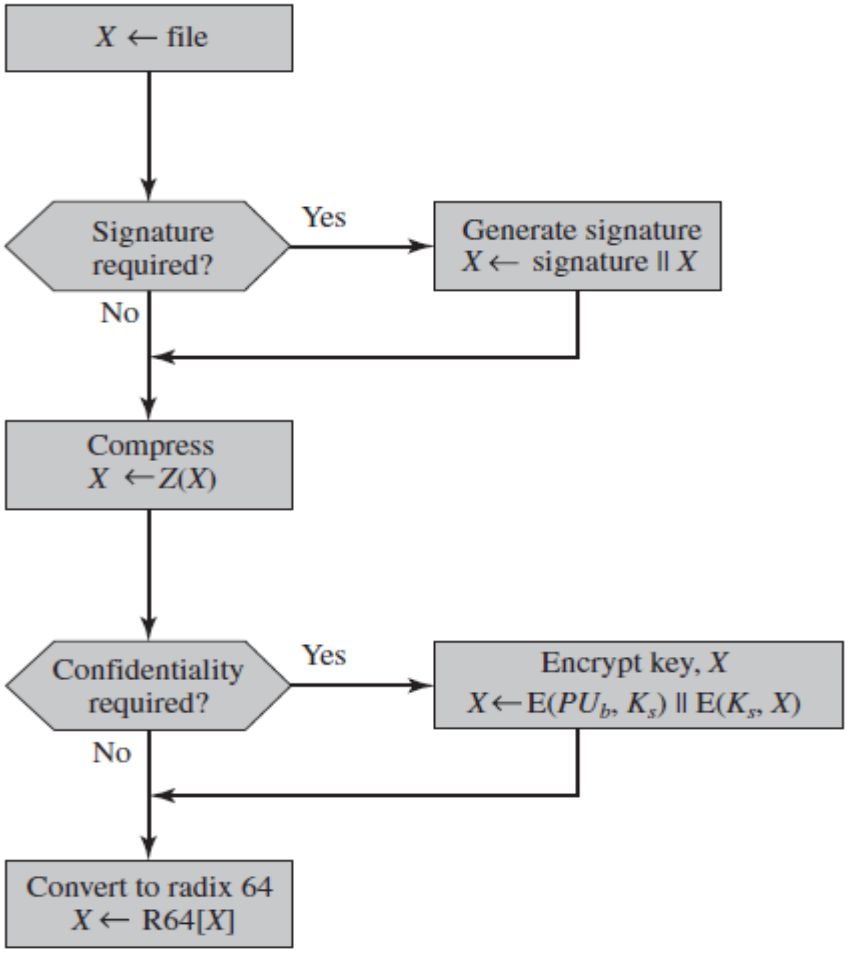
- The use of radix 64 expands a message by 33%.
- The compression should be more than enough to compensate for the radix-64 expansion.

PGP Compression & E-Mail Compatibility

Transmission and Reception of PGP Messages:

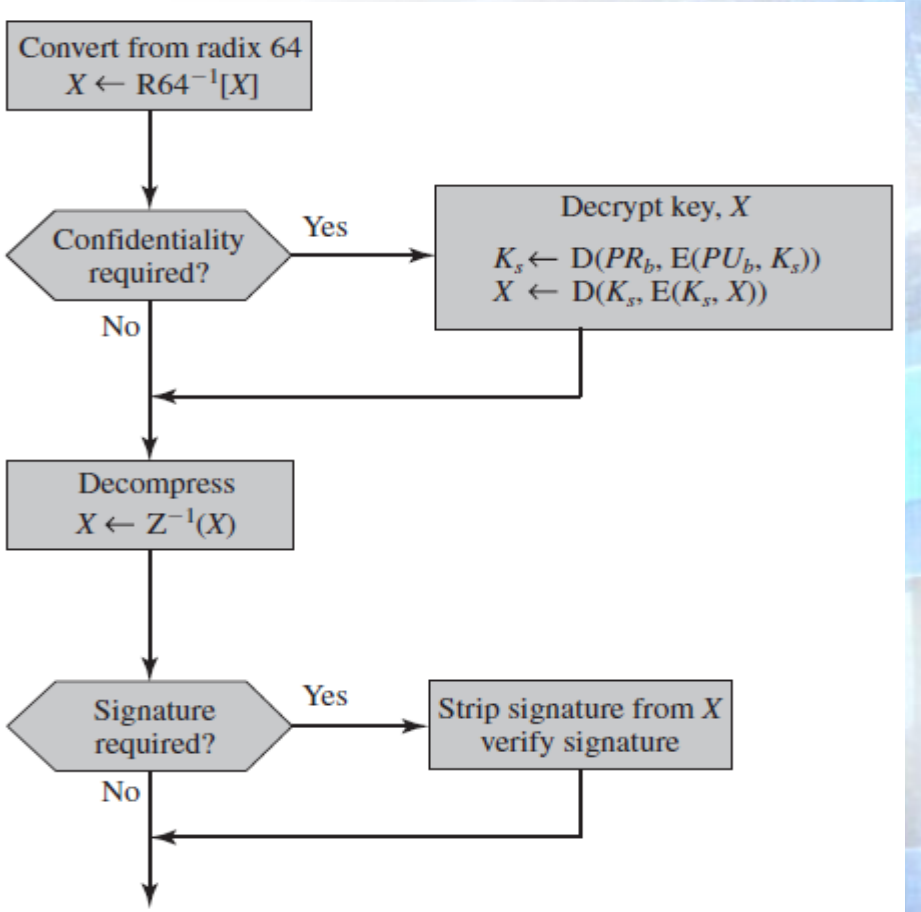
- The relationship among authentication, confidentiality, compression, and e-mail compatibility is shown.

PGP Compression & E-Mail Compatibility



(a) Generic transmission diagram (from A)

PGP Compression & E-Mail Compatibility



(b) Generic reception diagram (to B)

PGP Cryptographic Keys

Network Security

PGP Cryptographic Keys

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe PGP Cryptographic Keys.

PGP Cryptographic Keys

Figures and material in this topic have been adapted from

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

PGP Cryptographic Keys

- PGP makes use of four types of keys:
- one-time session symmetric keys,
- public keys,
- private keys, and
- passphrase-based symmetric keys

PGP Cryptographic Keys

- Three separate requirements can be identified with respect to these keys.
- **1.** A means of generating unpredictable session keys is needed.

PGP Cryptographic Keys

- **2.** We would like to allow a user to have multiple public-key/private-key pairs.
- One reason is that the user may wish to change his or her key pair from time to time.

PGP Cryptographic Keys

- Some means is needed for identifying particular keys.
- **3.** Each PGP entity must maintain a file of its own public/private key pairs as well as a file of public keys of correspondents.

PGP Cryptographic Keys

- **Session key generation:**
- Each session key is associated with a single message and is used only for the purpose of encrypting and decrypting that message.

PGP Cryptographic Keys

- Message encryption/decryption is done with a symmetric encryption algorithm.
- CAST-128 and IDEA use 128-bit keys.
- 3DES uses a 168-bit key.
- Lets use CAST-128 for our discussion:

PGP Cryptographic Keys

- Random 128-bit numbers are generated using CAST-128 itself.
- The input to the random number generator consists of a 128-bit key and two 64-bit blocks that are treated as plaintext to be encrypted.

PGP Cryptographic Keys

- Using cipher feedback mode, the CAST-128 encrypter produces two 64-bit cipher text blocks, which are concatenated to form the 128-bit session key.

PGP Cryptographic Keys

- **Key identifiers:**
- An encrypted message is accompanied by an encrypted form of the session key that was used for message encryption.
- The session key itself is encrypted with the recipient's public key.

PGP Cryptographic Keys

- Only the recipient will be able to recover the session key and therefore recover the message.
- There's a requirement that any given user may have multiple public/private key pairs.

PGP Cryptographic Keys

- How, then, does the recipient know which of its public keys was used to encrypt the session key?
- One simple solution would be to transmit the public key with the message.

PGP Cryptographic Keys

- This works, but wasteful of space.
- An RSA public key may be hundreds of decimal digits in length.
- Another solution is to associate an identifier with each public key that is unique at least within one user.

PGP Cryptographic Keys

- The combination of user ID and key ID would be sufficient to identify a key uniquely.
- In this case, only the much shorter key ID would need to be transmitted.

PGP Cryptographic Keys

- PGP assigns a key ID to each public key unique within a user ID.
- The key ID associated with each public key consists of its least significant 64 bits.
- The key ID of public key PUa is (PUa mod 2^{64}).

PGP Cryptographic Keys

- A key ID is also required for the PGP digital signature.
- Because a sender may use one of a number of private keys to encrypt the message digest, the recipient must know which public key is intended for use.

PGP Cryptographic Keys

End

- Digital signature component of a message includes the 64-bit key ID of the required public key.
- When the message is received, the recipient verifies that key ID is for a public key for that sender and then verifies the signature.

PGP Key Rings

Network Security

PGP Key Rings

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe PGP Key Rings.

PGP Key Rings

Figures and material in this topic have been adapted from

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

PGP Key Rings

- Key IDs are critical to the operation of PGP.
- Two key IDs are included in any PGP message that provides both confidentiality and authentication.

PGP Key Rings

- These keys need to be stored and organized in a systematic way for efficient and effective use by all parties.

PGP Key Rings

- The scheme used in PGP is to provide a pair of data structures at each node, one to store the public/private key pairs owned by that node and one to store the public keys of other users known at this node.

PGP Key Rings

- These data structures are referred to, respectively, as the **private-key ring** and the **public-key ring**.

PGP Key Rings

- The general structure of a **private-key ring** can be viewed as a table in which each row represents one of the public/private key pairs owned by a user.
- Each row contains the entries:

PGP Key Rings

- **Timestamp:** The date/time when this key pair was generated.
- **Key ID:** The least significant 64 bits of the public key for this entry.
- **Public key:** The public-key portion of the pair.

PGP Key Rings

- **Private key:** The private-key portion of the pair; this field is encrypted.
- **User ID:** Typically, this will be the user's e-mail address (e.g., stallings@acm.org).

PGP Key Rings

- However, the user may choose to associate a different name with each pair (e.g., Stallings, WStallings,WilliamStallings, etc.) or to reuse the same User ID more than once.

PGP Key Rings

The general structure of a private-key ring: The ring can be viewed as a table.

Private-Key Ring

Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
• • •	• • •	• • •	• • •	• • •
T_i	$PU_i \bmod 2^{64}$	PU_i	$E(H(P_i), PR_i)$	User i
• • •	• • •	• • •	• • •	• • •

* = field used to index table

PGP Key Rings

- The procedure to encrypt private key using CAST-128 (or IDEA or 3DES) is as follows:
- **1.** The user selects a passphrase to be used for encrypting private keys.

PGP Key Rings

- **2.** When the system generates a new public/private key pair using RSA, it asks the user for passphrase.
- Using SHA-1, a 160-bit hash code is generated from the passphrase, and the passphrase is discarded.

PGP Key Rings

- **3.** The system encrypts the private key using CAST-128 with the 128 bits of the hash code as the key.
- The hash code is then discarded, and the encrypted private key is stored in the private-key ring.

PGP Key Rings

- PGP will retrieve the encrypted private key, generate the hash code of the passphrase, and decrypt the encrypted private key using CAST-128 with the hash code.

PGP Key Rings

- Next, we show the general structure of a **public-key ring.**
- This data structure is used to store public keys of other users that are known to this user.

PGP Key Rings

The general structure of a public-key ring

Public-Key Ring

Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
T_i	$PU_i \bmod 2^{64}$	PU_i	trust_flag_i	User i	trust_flag_i		
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •

* = field used to index table

PGP Key Rings

- **Timestamp:** The date/time when this entry was generated.
- **Key ID:** The least significant 64 bits of the public key for this entry.
- **Public Key:** The public key for this entry.

PGP Key Rings

- **User ID:** Identifies the owner of this key. Multiple user IDs may be associated with a single public key.

End

Public Key Management by PGP

Network Security

Public Key Management by PGP

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe Public Key Management approaches by PGP.

Public Key Management by PGP

Figures and material in this topic have been adapted from

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

Public Key Management by PGP

- PGP contains a clever, efficient, interlocking set of functions and formats to provide an effective confidentiality and authentication service.

Public Key Management by PGP

- The PGP documentation captures the importance of public-key management:

Public Key Management by PGP

This whole business of protecting public keys from tampering is the single most difficult problem in practical public key applications. It is the "Achilles heel" of public key cryptography, and a lot of software complexity is tied up in solving this one problem.

Public Key Management by PGP

- PGP provides a structure for solving this problem with several suggested options that may be used.

Public Key Management by PGP

- Because PGP is intended for use in a variety of formal and informal environments, no rigid public-key management scheme is set up.

Public Key Management by PGP

- **Approaches to public-key management:**
- The problem is:
- User A must build up a public-key ring containing the public keys of other users to interoperate with them using PGP.

Public Key Management by PGP

- Suppose that A's key ring contains a public key attributed to B, but in fact the key is owned by C., E.g.
- If A got the key from a bulletin board system (BBS) that was used by B to post the public key but that has been compromised by C.

Public Key Management by PGP

- The result is that two threats now exist.
- First, C can send messages to A and forge B's signature so that A will accept the message as coming from B.

Public Key Management by PGP

- Second, any encrypted message from A to B can be read by C.

Public Key Management by PGP

- A number of approaches are possible for minimizing the risk that a user's public-key ring contains false public keys.
- Suppose that A wishes to obtain a reliable public key for B.

Public Key Management by PGP

- **1.** Physically get the key from B.
- B could store her public key on a floppy disk and hand it to A.
- A could then load the key into his system.
- This is a very secure method but has obvious practical limitations.

Public Key Management by PGP

- **2.** Verify a key by telephone.
- If A can recognize B on the phone, A could call B and ask her to dictate the key, in radix-64 format, over the phone.

Public Key Management by PGP

- As a more practical alternative, B could transmit her key in an e-mail message to A.
- A could have PGP generate a 160-bit SHA-1 digest of the key and display it in hexadecimal format.
- This is “fingerprint” of the key.

Public Key Management by PGP

- A could then call B and ask her to dictate the fingerprint over the phone.
- If the two fingerprints match, the key is verified.

Public Key Management by PGP

- **3.** Obtain B's public key from a mutual trusted individual D.
- For this purpose, the introducer D creates a signed certificate. The certificate includes B's public key, the time of creation of the key, and a validity period for the key.

Public Key Management by PGP

- D generates an SHA-1 digest of this certificate, encrypts it with her private key, and attaches the signature to the certificate.

Public Key Management by PGP

- Because only D could have created the signature, no one else can create a false public key and pretend that it is signed by D.
- The signed certificate could be sent directly to A by B or D, or post on a bulletin board.

Public Key Management by PGP

- **4.** Obtain B's public key from a trusted certifying authority.
- A public-key certificate is created and signed by the authority.
- A could then access authority, providing a user name, receiving a signed certificate.

Public Key Management by PGP

End

- For cases 3 and 4, A already would have to have a copy of the introducer's public key and trust that this key is valid.
- Ultimately, it is up to A to assign a level of trust to anyone who is to act as an introducer.

PGP Trust Model

Network Security

PGP Trust Model

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe PGP Trust Model.

PGP Trust Model

Figures and material in this topic have been adapted from

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

PGP Trust Model

- PGP does not include any specification for establishing certifying authorities or for establishing trust.
- It does provide a convenient means of using trust, associating trust with public keys.

PGP Trust Model

- The basic structure is as follows:
- Each entry in the public-key ring is a public key certificate.

PGP Trust Model

- Associated with each such entry is a **key legitimacy field** that indicates the extent to which PGP will trust that this is a valid public key for this user.

PGP Trust Model

- Higher the level of trust, the stronger is the binding of this user ID to this key.
- This field is computed by PGP.

PGP Trust Model

- Also associated with the entry are zero or more signatures that the key ring owner has collected that sign this certificate.

PGP Trust Model

- Each signature has associated with it a **signature trust field** that indicates the degree to which this PGP user trusts the signer to certify public keys.
- The key legitimacy field is derived from signature trust fields.

PGP Trust Model

- Finally, each entry defines an **owner trust field** that indicates the degree to which this public key is trusted to sign other public-key certificates.
- This level of trust is assigned by the user.

PGP Trust Model

- These three fields are each contained in a structure referred to as a trust flag byte.

PGP Trust Model

(a) Trust Assigned to Public-Key Owner (appears after key packet; user defined)	(b) Trust Assigned to Public Key/User ID Pair (appears after User ID packet; computed by PGP)	(c) Trust Assigned to Signature (appears after signature packet; cached copy of OWNERTRUST for this signator)
OWNERTRUST Field —undefined trust —unknown user —usually not trusted to sign other keys —usually trusted to sign other keys —always trusted to sign other keys —this key is present in secret key ring (ultimate trust)	KEYLEGIT Field —unknown or undefined trust —key ownership not trusted —marginal trust in key ownership —complete trust in key ownership	SIGTRUST Field —undefined trust —unknown user —usually not trusted to sign other keys —usually trusted to sign other keys —always trusted to sign other keys —this key is present in secret key ring (ultimate trust)
BUCKSTOP bit —set if this key appears in secret key ring	WARNONLY bit —set if user wants only to be warned when key that is not fully validated is used for encryption	CONTIG bit —set if signature leads up a contiguous trusted certification path back to the ultimately trusted key ring owner

PGP Trust Model

- Suppose that we are dealing with public key ring of user A.
- We can describe the operation of the trust processing as follows:

PGP Trust Model

- **1.** When A inserts a new public key on the public-key ring, PGP must assign a value to the trust flag that is associated with the owner of this public key.

PGP Trust Model

- If the owner is A, then a value of *ultimate trust* is automatically assigned to the trust field.
- Otherwise, PGP asks A for his assessment: the owner is unknown, untrusted, marginally trusted, or completely trusted.

PGP Trust Model

- **2.** When the new public key is entered, one or more signatures may be attached to it.
- To do this, PGP searches public-key ring to see if author of this signature is among known public-key owners.

PGP Trust Model

- If so, the OWNERTRUST value for this owner is assigned to the SIGTRUST field for this signature.
- If not, an *unknown user* value is assigned.

PGP Trust Model

- **3.** The value of the key legitimacy field is calculated on the basis of the signature trust fields present in this entry.
- If at least one signature has a trust value of *ultimate*, key legitimacy value is set to complete.

PGP Trust Model

- In the absence of ultimate trust, at least X signatures that are always trusted, Y signatures that are usually trusted, or some combination is needed.

PGP Trust Model

- Periodically, PGP processes the public-key ring to achieve consistency.
- In essence, this is a top-down process

PGP Trust Model

- For each OWNERTRUST field, PGP scans the ring for all signatures authored by that owner and updates the SIGTRUST field to equal the OWNERTRUST field.

PGP Trust Model

- This process starts with keys for which there is ultimate trust.
- Then all KEYLEGIT fields are computed on the basis of the attached signatures.

End

Multipurpose Internet Mail Extension

Network Security

Multipurpose Internet Mail Extension

Objectives of the Topic

- After completing this topic, a student will be able to
  - Explain basics of Multipurpose Internet Mail Extension.

Multipurpose Internet Mail Extension

Figures and material in this topic have been adapted from

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

Multipurpose Internet Mail Extension

- **RFC 5322 (Internet Message Format):**
- defines a format for text messages that are sent using electronic mail.
- Messages are viewed as having an envelope and contents.

Multipurpose Internet Mail Extension

- The envelope contains whatever information is needed to accomplish transmission and delivery.
- The contents compose the object to be delivered to the recipient.

Multipurpose Internet Mail Extension

- The RFC 5322 standard applies only to the contents.
- The content standard includes a set of header fields that may be used by the mail system to create the envelope.

Multipurpose Internet Mail Extension

- The overall structure of a message:
- A message consists of some number of header lines (*the header*) followed by unrestricted text (*the body*).
- The header is separated from the body by a blank line.

Multipurpose Internet Mail Extension

An Example

```
Date: October 8, 2009 2:15:49 PM EDT
From: "William Stallings" <ws@shore.net>
Subject: The Syntax in RFC 5322
To: Smith@Other-host.com
Cc: Jones@Yet-Another-Host.com

Hello. This section begins the actual
message body, which is delimited from the
message heading by a blank line.
```

Multipurpose Internet Mail Extension

- **Multipurpose Internet Mail Extensions (MIME):**
- is an extension to the RFC 5322 framework.
- addresses some of the problems and limitations of the use of Simple Mail Transfer Protocol (SMTP).

Multipurpose Internet Mail Extension

- Following is the list of the limitations of the SMTP/5322 scheme.
- **1.** SMTP cannot transmit executable files or other binary objects.

Multipurpose Internet Mail Extension

- **2.** SMTP cannot transmit text data that includes national language characters, because these are represented by 8-bit codes and SMTP is limited to 7-bit ASCII.

Multipurpose Internet Mail Extension

- **3.** SMTP servers may reject mail message over a certain size.
- **4.** SMTP gateways that translate between ASCII and the character code EBCDIC do not use a consistent set of mappings, resulting in translation problems.

Multipurpose Internet Mail Extension

- **5.** SMTP gateways to X.400 electronic mail networks cannot handle nontextual data included in X.400 messages.

Multipurpose Internet Mail Extension

- **6.** Some SMTP implementations do not adhere completely to the SMTP standard defined in RFC 821.

Multipurpose Internet Mail Extension

- MIME is intended to resolve these problems in a manner that is compatible with existing RFC 5322 implementations.

Multipurpose Internet Mail Extension

- Elements of MIME specification are:
- **1.** Five new message header fields are defined, which may be included in the RFC 5322 header.
- These fields provide information about the body of the message.

Multipurpose Internet Mail Extension

- **2.** A number of content formats are defined, thus standardizing representations that support multimedia electronic mail.

Multipurpose Internet Mail Extension

- **3.** Transfer encodings are defined that enable the conversion of any content format into a form that is protected from alteration by the mail system.

Multipurpose Internet Mail Extension

- The five header fields defined in MIME are
- **MIME-Version:** Must have the parameter value 1.0.
- This field indicates that the message conforms to RFCs 2045 and 2046.

Multipurpose Internet Mail Extension

- **Content-Type:** Describes the data in body with sufficient detail that receiving user agent can pick an appropriate agent to represent the data to the user or otherwise deal with the data in an appropriate manner.

Multipurpose Internet Mail Extension

- **Content-Transfer-Encoding:** Indicates the type of transformation that has been used to represent the body of the message in a way that is acceptable for mail transport.

Multipurpose Internet Mail Extension

End

- **Content-ID:** Used to identify MIME entities uniquely in multiple contexts.
- **Content-Description:** A text description of the object with the body; this is useful when the object is not readable (e.g., audio data).

MIME Content Types

Network Security

MIME Content Types

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe MIME Content Types.

MIME Content Types

Figures and material in this topic have been adapted from

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

MIME Content Types

- There are seven different major types of content and a total of 15 subtypes.
- In general, a content type declares the general type of data, and the subtype specifies a particular format for that type of data.

MIME Content Types

- **1. text type:**
- no special software is required to get the full meaning of the text aside from support of the indicated character set.

MIME Content Types

- The primary subtype is **plain text**, which is simply a string of ASCII characters or ISO 8859 characters.
- The **enriched** subtype allows greater formatting flexibility.

MIME Content Types

- **2. multipart type:**
- indicates that the body contains multiple, independent parts.
- The Content-Type header field includes a parameter (called a boundary) that defines the delimiter between body parts.

MIME Content Types

- Each boundary starts on a new line and consists of two hyphens followed by the boundary value.
- The final boundary, which indicates the end of the last part, also has a suffix of two hyphens.

MIME Content Types

```
From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: Sample message
MIME-Version: 1.0
Content-type: multipart/mixed; boundary="simple
boundary"

This is the preamble. It is to be ignored, though it
is a handy place for mail composers to include an
explanatory note to non-MIME conformant readers.
—simple boundary

This is implicitly typed plain ASCII text. It does NOT
end with a linebreak.
—simple boundary
Content-type: text/plain; charset=us-ascii

This is explicitly typed plain ASCII text. It DOES end
with a linebreak.

—simple boundary—
This is the epilogue. It is also to be ignored.
```

MIME Content Types

- There are four subtypes of the multipart type.
- The **multipart/mixed subtype** is used when there are multiple independent body parts that need to be bundled in a particular order.

MIME Content Types

- For **multipart/parallel subtype**, the order of the parts is not significant.
- If the recipient's system is appropriate, the multiple parts can be presented in parallel.

MIME Content Types

- For example, a picture or text part could be accompanied by a voice commentary that is played while the picture or text is displayed.

MIME Content Types

- For the **multipart/alternative subtype**, the various parts are different representations of the same information.
- In this subtype, the body parts are ordered in terms of increasing preference.

MIME Content Types

```
From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: Formatted text mail
MIME-Version: 1.0
Content-Type: multipart/alternative;
boundary = boundary42

   –boundary42

Content-Type: text/plain; charset = us-ascii

   ...plain text version of message goes here....

   –boundary42
Content-Type: text/enriched

   ...RFC 1896 text/enriched version of same message
goes here...

   –boundary42–
```

MIME Content Types

- If the recipient system is capable of displaying the message in the text/enriched format, this is done.
- Otherwise, the plain text format is used.

MIME Content Types

- The **multipart/digest subtype** is used when each of the body parts is interpreted as an RFC 5322 message with headers.
- This subtype enables the construction of a message whose parts are individual messages.

MIME Content Types

- **3. message type:**
- provides a number of important capabilities in MIME.
- The **message/rfc822 subtype** indicates that the body is an entire message, including header and body.

MIME Content Types

- The **message/partial subtype** enables fragmentation of a large message into a number of parts, which must be reassembled at the destination.

MIME Content Types

- Three parameters are specified in the Content-Type: Message/Partial field: an *id* common to all fragments of the same message, a *sequence number* unique to each fragment, and the *total* number of fragments.

MIME Content Types

- The **message/external-body subtype** indicates that the actual data to be conveyed in this message are not contained in the body.
- The body contains the information needed to access the data.

MIME Content Types

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

MIME Transfer Encodings

Network Security

MIME Transfer Encodings

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain MIME Transfer Encodings.

MIME Transfer Encodings

Figures and material in this topic have been adapted from

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

MIME Transfer Encodings

- The MIME standard defines two methods of encoding data.
- The Content-Transfer-Encoding field can actually take on six values, as listed next.

MIME Transfer Encodings

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

MIME Transfer Encodings

- Three of these values (7bit, 8bit, and binary) indicate that no encoding has been done but provide some information about the nature of the data.

MIME Transfer Encodings

- For SMTP transfer, it is safe to use the 7bit form.
- The 8bit and binary forms may be usable in other mail transport contexts.

MIME Transfer Encodings

- Another Content-Transfer-Encoding value is x-token.
- **X-token** indicates that some other encoding scheme is used for which a name is to be supplied.
- It could be a vendor-specific or application-specific scheme.

MIME Transfer Encodings

- The two actual encoding schemes defined are
- **quoted-printable,** and
- **base64**.

MIME Transfer Encodings

- Two schemes are defined to provide a choice between a transfer technique that is essentially human readable and one that is safe for all types of data in a way that is reasonably compact.

MIME Transfer Encodings

- The **quoted-printable** transfer encoding is useful when the data consists largely of octets that correspond to printable ASCII characters.

MIME Transfer Encodings

- The **base64 transfer encoding** or radix-64 encoding, is a common one for encoding arbitrary binary data in such a way as to be invulnerable to the processing by mail-transport programs.
- It is also used in PGP.

MIME Transfer Encodings

- **A Multipart Example:**
- We next show an example that has been taken from RFC 2045.
- It is the outline of a complex multipart message.

MIME Transfer Encodings

MIME-Version: 1.0
From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: A multipart example
Content-Type: multipart/mixed;
boundary=unique-boundary-1

This is the preamble area of a multipart message. Mail readers that understand multipart format should ignore this preamble. If you are reading this text, you might want to consider changing to a mail reader that understands how to properly display multipart messages.

--unique-boundary-1

...Some text appears here...
[Note that the preceding blank line means no header fields were given and this is text, with charset US ASCII. It could have been done with explicit typing as in the next part.]

--unique-boundary-1
Content-type: text/plain; charset=US-ASCII

This could have been part of the previous part, but illustrates explicit versus implicit typing of body parts.

--unique-boundary-1
Content-Type: multipart/parallel; boundary=unique-boundary-2

MIME Transfer Encodings

```
--unique-boundary-2
Content-Type: audio/basic
Content-Transfer-Encoding: base64

    ... base64-encoded 8000 Hz single-channel mu-law-format audio data goes here....

--unique-boundary-2
Content-Type: image/jpeg
Content-Transfer-Encoding: base64

    ... base64-encoded image data goes here....

--unique-boundary-2--

--unique-boundary-1
Content-type: text/enriched

This is <bold><italic>richtext.</italic></bold> <smaller>as defined in RFC 1896</smaller>

Isn't it <bigger><bigger>cool?</bigger></bigger>

--unique-boundary-1
Content-Type: message/rfc822
```

MIME Transfer Encodings

From: (mailbox in US-ASCII)
To: (address in US-ASCII)
Subject: (subject in US-ASCII)
Content-Type: Text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: Quoted-printable

... Additional text in ISO-8859-1 goes here ...

--unique-boundary-1--

MIME Transfer Encodings

- The message has five parts to be displayed serially:
- two introductory plain text parts, an embedded multipart message, a rich text part, and a closing encapsulated text message in a non-ASCII character set.

MIME Transfer Encodings

- The embedded multipart message has two parts to be displayed in parallel: a picture and an audio fragment.

End

S/MIME Functionality

Network Security

S/MIME Functionality

Objectives of the Topic

- After completing this topic, a student will be able to
  - Describe functionality of S/MIME.

S/MIME Functionality

Figures and material in this topic have been adapted from

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

S/MIME Functionality

- Secure/Multipurpose Internet Mail Extension (S/MIME) is a security enhancement to the MIME Internet e-mail format standard based on technology from RSA Data Security.

S/MIME Functionality

- S/MIME is defined in RFCs 3370, 3850, 3851, and 3852.

S/MIME Functionality

- **S/MIME Functionality:**
- In terms of general functionality, S/MIME is very similar to PGP.
- Both offer the ability to sign and/or encrypt messages.

S/MIME Functionality

- S/MIME provides the following functions:
- **Enveloped data:**
- This consists of encrypted content of any type and encrypted content encryption keys for one or more recipients.

S/MIME Functionality

- **Signed data:**
- A digital signature is formed by taking the message digest of the content to be signed and then encrypting that with the private key of the signer.

S/MIME Functionality

- The content plus signature are then encoded using base64 encoding.
- A signed data message can only be viewed by a recipient with S/MIME capability.

S/MIME Functionality

- **Clear-signed data:**
- As with signed data, a digital signature of the content is formed.
- Only the digital signature is encoded using base64.

S/MIME Functionality

- As a result, recipients without S/MIME capability can view the message content, although they cannot verify the signature.

S/MIME Functionality

- **Signed and enveloped data:**
- Signed-only and encrypted-only entities may be nested:
- encrypted data may be signed and
- signed data or clear-signed data may be encrypted.

S/MIME Functionality

- **Cryptographic Algorithms:**
- S/MIME uses the following terminology taken from RFC 2119 (*Key Words for use in RFCs to Indicate Requirement Levels*) to specify the requirement level:

S/MIME Functionality

- **MUST:** The definition is an absolute requirement of the specification.
- An implementation must include this feature or function to be in conformance with the specification.

S/MIME Functionality

- **SHOULD:** There may exist valid reasons in particular circumstances to ignore this feature or function, but it is recommended that an implementation include the feature or function.

S/MIME Functionality

- S/MIME incorporates three public-key algorithms:
- The Digital Signature Standard (DSS) is the preferred algorithm for digital signature.

S/MIME Functionality

- S/MIME lists Diffie-Hellman as the preferred algorithm for encrypting session keys.
- S/MIME uses a variant of Diffie-Hellman that does provide encryption/decryption known as ElGamal.

S/MIME Functionality

- As an alternative, RSA, can be used for both signatures and session key encryption.
- These are the same algorithms used in PGP and provide a high level of security.

S/MIME Functionality

- For the hash function used to create the digital signature, the specification requires the 160-bit SHA-1 but recommends receiver support for the 128-bit MD5 for backward compatibility with older versions of S/MIME.

S/MIME Functionality

- There is justifiable concern about the security of MD5, so SHA-1 is clearly the preferred alternative.

S/MIME Functionality

- For message encryption, three-key triple DES is recommended, but compliant implementations must support 40-bit RC2.
- The latter is a weak encryption algorithm.

S/MIME Functionality

Function	Requirement
Create a message digest to be used in forming a digital signature.	MUST support SHA-1. Receiver SHOULD support MD5 for backward compatibility.
Encrypt message digest to form a digital signature.	Sending and receiving agents MUST support DSS. Sending agents SHOULD support RSA encryption. Receiving agents SHOULD support verification of RSA signatures with key sizes 512 bits to 1024 bits.
Encrypt session key for transmission with a message.	Sending and receiving agents SHOULD support Diffie-Hellman. Sending and receiving agents MUST support RSA encryption with key sizes 512 bits to 1024 bits.
Encrypt message for transmission with a one-time session key.	Sending and receiving agents MUST support encryption with tripleDES. Sending agents SHOULD support encryption with AES. Sending agents SHOULD support encryption with RC2/40.
Create a message authentication code.	Receiving agents MUST support HMAC with SHA-1. Sending agents SHOULD support HMAC with SHA-1.

S/MIME Messages

Network Security

S/MIME Messages

Objectives of the Topic

- After completing this topic, a student will be able to
  - Explain S/MIME Messages.

S/MIME Messages

Figures and material in this topic have been adapted from

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

S/MIME Messages

- S/MIME makes use of a number of new MIME content types.

S/MIME Messages

- All of the new application types use the designation PKCS.
- This refers to a set of public-key cryptography specifications issued by RSA Laboratories and made available for the S/MIME effort.

S/MIME Messages

Type	Subtype	smime Parameter	Description
Multipart	Signed		A clear-signed message in two parts: one is the message and the other is the signature.
Application	pkcs7-mime	signedData	A signed S/MIME entity.
	pkcs7-mime	envelopedData	An encrypted S/MIME entity.
	pkcs7-mime	degenerate signedData	An entity containing only public-key certificates.
	pkcs7-mime	CompressedData	A compressed S/MIME entity.
	pkcs7-signature	signedData	The content type of the signature subpart of a multipart/signed message.

S/MIME Messages

- **Securing a MIME Entity:**
- S/MIME secures a MIME entity with a signature, encryption, or both.
- A MIME entity may be an entire message, or one or more of the subparts of the message.

S/MIME Messages

- The MIME entity is prepared according to the normal rules for MIME message preparation.

S/MIME Messages

- The MIME entity plus some security-related data, such as algorithm identifiers and certificates, are processed by S/MIME to produce what is known as a PKCS object.

S/MIME Messages

- A PKCS object is then treated as message content and wrapped in MIME.

S/MIME Messages

- **EnvelopedData:**
- The steps for preparing an envelopedData MIME entity are:
- **1.** Generate a pseudorandom session key for a particular symmetric encryption algorithm.

S/MIME Messages

- **2.** For each recipient, encrypt session key with the recipient's public RSA key.

S/MIME Messages

- **3.** For each recipient, prepare a block known as *RecipientInfo* that contains an identifier of recipient's public-key certificate, an identifier of algorithm used to encrypt the session key, and the encrypted session key.

S/MIME Messages

- **4.** Encrypt the message content with the session key.

S/MIME Messages

- **SignedData:** The steps for preparing a signedData MIME entity are
- **1.** Select a message digest algorithm (SHA or MD5).
- **2.** Compute the message digest (hash function) of the content to be signed.

S/MIME Messages

- **3.** Encrypt the message digest with the signer's private key.
- **4.** Prepare a block known as *SignerInfo*.

S/MIME Messages

- It contains the signer's public-key certificate, an identifier of the message digest algorithm, an identifier of the algorithm used to encrypt the message digest, and encrypted message digest.

S/MIME Messages

- **Clear Signing:** is achieved using the multipart content type with a signed subtype.
- This signing process does not involve transforming the message to be signed.

S/MIME Messages

- Recipients with MIME capability but not S/MIME capability are able to read the incoming message.

S/MIME Messages

- **Registration Request:**
- Typically, a user will apply to a certification authority for a public-key certificate.
- The application/pkcs10 S/MIME entity is used to transfer a certification request.

S/MIME Messages

End

- **Certificates-Only Message:**
- A message containing only certificates or certification revocation list (CRL) can be sent in response to a registration request.

S/MIME Certificate Processing

Network Security

S/MIME Certificate Processing

Objectives of the Topic

- After completing this topic, a student will be able to
  - Describe processing of S/MIME certificates.

S/MIME Certificate Processing

Figures and material in this topic have been adapted from

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

S/MIME Certificate Processing

- S/MIME uses public-key certificates that conforms to version 3 of X.509.
- The key-management scheme used by S/MIME is a hybrid between a strict X.509 certification hierarchy and PGP's web of trust.

S/MIME Certificate Processing

- S/MIME managers and/or users must configure each client with a list of trusted keys and with certificate revocation lists.

S/MIME Certificate Processing

- The responsibility is local for maintaining the certificates needed to verify incoming signatures and to encrypt outgoing messages.
- The certificates are signed by certification authorities.

S/MIME Certificate Processing

- **User Agent Role:**
- An S/MIME user has several key-management functions to perform.

S/MIME Certificate Processing

- **Key generation:**
- The user of some related administrative utility must be capable of generating separate Diffie-Hellman and DSS key pairs and should be capable of generating RSA key pairs.

S/MIME Certificate Processing

- A user agent should generate RSA key pairs with a length in the range of 768 to 1024 bits and must not generate a length of less than 512 bits.

S/MIME Certificate Processing

- **Registration:**
- A user's public key must be registered with a certification authority in order to receive an X.509 public-key certificate.

S/MIME Certificate Processing

- **Certificate storage and retrieval:**
- A user requires access to a local list of certificates in order to verify incoming signatures and to encrypt outgoing messages.

S/MIME Certificate Processing

- **VeriSign Certificates:**
- VeriSign provides a certification authority (CA) service that is intended to be compatible with S/MIME and a variety of other applications.
- The most widely used company that provides CA services.

S/MIME Certificate Processing

- It issues X.509 certificates with the product name VeriSign Digital ID.
- At a minimum, each Digital ID contains:
- Owner’s public key
- Owner’s name or alias
- Expiration date of the Digital ID

S/MIME Certificate Processing

- Serial number of the Digital ID
- Name of the certification authority that issued the Digital ID
- Digital signature of the certification authority that issued the Digital ID

S/MIME Certificate Processing

- VeriSign provides three levels, or classes, of security for public-key certificates.

S/MIME Certificate Processing

- A user requests a certificate online at VeriSign's Web site or other participating Web sites.
- Class 1 and Class 2 requests are processed on line, and in most cases take only a few seconds to approve.

S/MIME Certificate Processing

- The following procedures are used.
- For Class 1 Digital IDs, VeriSign confirms the user's e-mail address by sending a PIN and Digital ID pick-up information to the e-mail address provided in the application.

S/MIME Certificate Processing

- For Class 2 Digital IDs, VeriSign verifies the information in the application through an automated comparison with a consumer database in addition to performing all of the checking associated with a Class 1 Digital ID.

S/MIME Certificate Processing

- Finally, confirmation is sent to the specified postal address alerting the user that a Digital ID has been issued in his or her name.

S/MIME Certificate Processing

End

- For Class 3 Digital IDs, VeriSign requires a higher level of identity assurance.
- An individual must prove his or her identity by providing notarized credentials or applying in person.

Internet Mail Architecture

Network Security

Internet Mail Architecture

Objectives of the Topic

- After completing this topic, a student will be able to
  - Describe Internet Mail Architecture.

Internet Mail Architecture

Figures and material in this topic have been adapted from

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

Internet Mail Architecture

- Internet mail architecture is defined in RFC 5598.

Internet Mail Architecture

- The Internet mail architecture consists of a user world in the form of Message User Agents (MUA), and the transfer world, in the form of the Message Handling Service (MHS).

Internet Mail Architecture

- MHS is composed of Message Transfer Agents (MTA).
- The MHS accepts a message from one user and delivers it to one or more other users, creating a virtual MUA-to-MUA exchange environment.

Internet Mail Architecture

- This architecture involves three types of interoperability.

Internet Mail Architecture

- One is directly between users: messages must be formatted by the MUA on behalf of the message author so that the message can be displayed to the message recipient by the destination MUA.

Internet Mail Architecture

- Interoperability requirements between the MUA and the MHS:
- First when a message is posted from an MUA to the MHS and later when it is delivered from the MHS to the destination MUA.

Internet Mail Architecture

- Interoperability is required among the MTA components along the transfer path through the MHS.

Internet Mail Architecture

- **Message User Agent (MUA):**
- Operates on behalf of user actors and user applications.
- Typically, this function is housed in the user's computer and is referred to as a client program or a local network e-mail server.

Internet Mail Architecture

- The author MUA formats a message and performs initial submission into the MHS via a MSA.
- The recipient MUA processes received mail for storage and/or display to the recipient user.

Internet Mail Architecture

- **Mail Submission Agent (MSA):**
- Accepts the message submitted by an MUA and enforces the policies of the hosting domain and the requirements of Internet standards.

Internet Mail Architecture

- This function may be located together with the MUA or as a separate functional model.
- In the latter case, the Simple Mail Transfer Protocol (SMTP) is used between the MUA and the MSA.

Internet Mail Architecture

- **Message Transfer Agent (MTA):**
- Relays mail for one application-level hop.
- It is like a packet switch or IP router in that its job is to make routing assessments and to move the message closer to the recipients.

Internet Mail Architecture

- Relaying is performed by a sequence of MTAs until the message reaches a destination MDA.
- An MTA also adds trace information to the message header.
- SMTP is used between MTAs and bet. an MTA and an MSA or MDA.

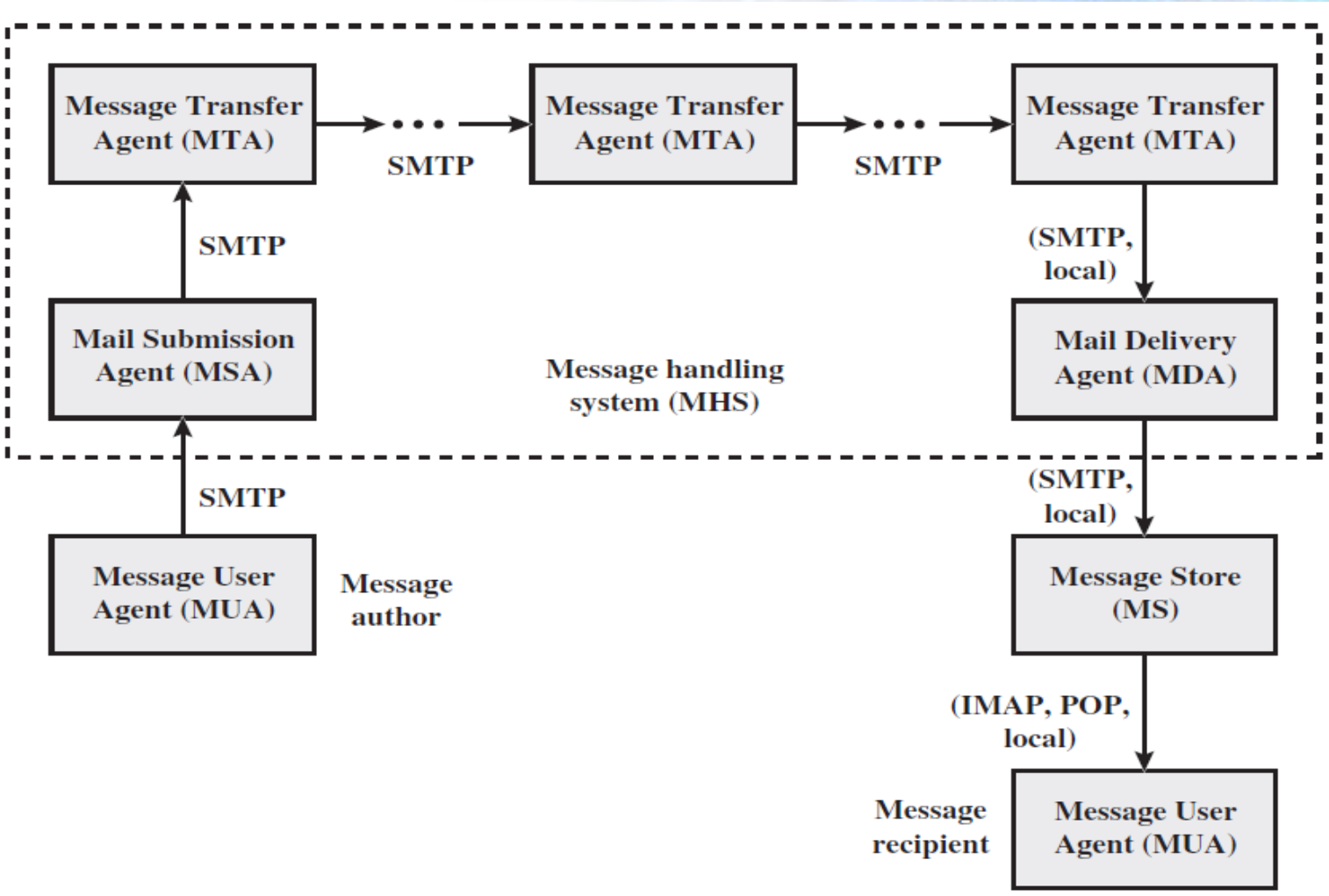
Internet Mail Architecture

- **Mail Delivery Agent (MDA):**
- Responsible for transferring the message from the MHS to the MS.

Internet Mail Architecture

- **Message Store (MS):**
- An MUA can employ a long-term MS.
- An MS can be located on a remote server or on the same machine as the MUA.
- An MUA retrieves messages from a remote server using POP or IMAP.

Internet Mail Architecture



E-mail Threats

Network Security

E-mail Threats

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe E-mail Threats.

E-mail Threats

Figures and material in this topic have been adapted from

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

E-mail Threats

- DomainKeys Identified Mail is a specification for cryptographically signing e-mail messages, permitting a signing domain to claim responsibility for a message in the mail stream.
- DKIM is an Internet Standard (RFC 4871).

E-mail Threats

- DKIM has been widely adopted by a range of e-mail providers, including corporations, government agencies, gmail, yahoo, and many Internet Service Providers (ISPs).

E-mail Threats

- RFC 4686 (*Analysis of Threats Motivating DomainKeys Identified Mail*) describes the threats being addressed by DKIM in terms of the characteristics, capabilities, and location of potential attackers.

E-mail Threats

- **Characteristics:**
- RFC 4686 characterizes the range of attackers on a spectrum of three levels of threat.

E-mail Threats

- The most sophisticated and financially motivated senders of messages are those who stand to receive substantial financial benefit, such as from an e-mail based fraud scheme.

E-mail Threats

- The next level are professional senders of bulk spam mail and often operate as commercial enterprises and send messages on behalf of third parties.

E-mail Threats

- At the low end are attackers who simply want to send e-mail that a recipient does not want to receive.

E-mail Threats

- **Capabilities:**
- RFC 4686 lists the following as capabilities that an attacker might have.
- **1.** Submit messages to MTAs and Message Submission Agents (MSAs) at multiple locations in the Internet.

E-mail Threats

- **2.** Construct arbitrary Message Header fields, including those claiming to be mailing lists, resenders, and other mail agents.
- **3.** Sign messages on behalf of domains under their control.

E-mail Threats

- **4.** Generate substantial numbers of either unsigned or apparently signed messages that might be used to attempt a DOS attack.
- **5.** Resend messages that may have been previously signed by the domain.

E-mail Threats

- **6.** Transmit messages using any envelope information desired.
- **7.** Act as an authorized submitter for messages from a compromised computer.

E-mail Threats

- **8.** Manipulation of IP routing. E.g. cause diversion of messages to a specific domain.
- **9.** Limited influence over portions of DNS using mechanisms such as cache poisoning. E.g. falsify advertisements of DNS-based keys.

E-mail Threats

- **10.** Access to significant computing resources through the worm-infected "zombie" computers.
- **11.** Ability to eavesdrop on existing traffic, perhaps from a wireless network.

E-mail Threats

- **Location:**
- DKIM focuses primarily on attackers located outside of the administrative units of the claimed originator and the recipient.

E-mail Threats

- These administrative units frequently correspond to the protected portions of the network adjacent to the originator and recipient.

E-mail Threats

- It is in this area that the trust relationships required for authenticated message submission do not exist and do not scale adequately to be practical.

E-mail Threats

End

- External “bad actors” are usually attempting to exploit the “any-to-any” nature of e-mail that motivates most recipient MTAs to accept messages from anywhere for delivery to their local domain.

DomainKeys Identified Mail Strategy

Network Security

DomainKeys Identified Mail Strategy

Objectives of the Topic

- After completing this topic, a student will be able to
  - Describe DKIM strategy.

DomainKeys Identified Mail Strategy

Figures and material in this topic have been adapted from

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

DomainKeys Identified Mail Strategy

- DKIM is designed to provide an e-mail authentication technique that is transparent to the end user.

DomainKeys Identified Mail Strategy

- A user's e-mail message is signed by a private key of the administrative domain from which the e-mail originates.
- The signature covers all of the content of the message and some of the RFC 5322 message headers.

DomainKeys Identified Mail Strategy

- At the receiving end, the MDA can access the corresponding public key via a DNS and verify the signature.
- Thus, authenticating that message comes from the claimed administrative domain.

DomainKeys Identified Mail Strategy

- Mail that originates from somewhere else but claims to come from a given domain will not pass the authentication test and can be rejected.

DomainKeys Identified Mail Strategy

- This approach differs from that of S/MIME and PGP, which use the originator's private key to sign the content of the message.

DomainKeys Identified Mail Strategy

- The motivation for DKIM is:
- **1.** S/MIME depends on both the sending and receiving users employing S/MIME.
- For almost all users, the bulk of incoming mail does not use S/MIME.

DomainKeys Identified Mail Strategy

- **2.** S/MIME signs only the message content.
- RFC 5322 header information concerning origin can be compromised.
- **3.** DKIM is not implemented in client programs (MUAs) and is transparent to the user.

DomainKeys Identified Mail Strategy

- **4.** DKIM applies to all mail from cooperating domains.
- **5.** DKIM allows good senders to prove that they did send a particular message and to prevent forgers from masquerading as good senders.

DomainKeys Identified Mail Strategy

- Operation of DKIM begins with a message generated by a user and transmitted into the MHS to an MSA that is within the user's administrative domain.

DomainKeys Identified Mail Strategy

- An e-mail message is generated by an e-mail client program.
- The content of the message, plus selected RFC 5322 headers, is signed by the e-mail provider using the provider's private key.

DomainKeys Identified Mail Strategy

- The signer is associated with a domain, which could be a corporate local network, an ISP, or a public e-mail facility such as gmail.
- The signed message then passes through the Internet via a sequence of MTAs.

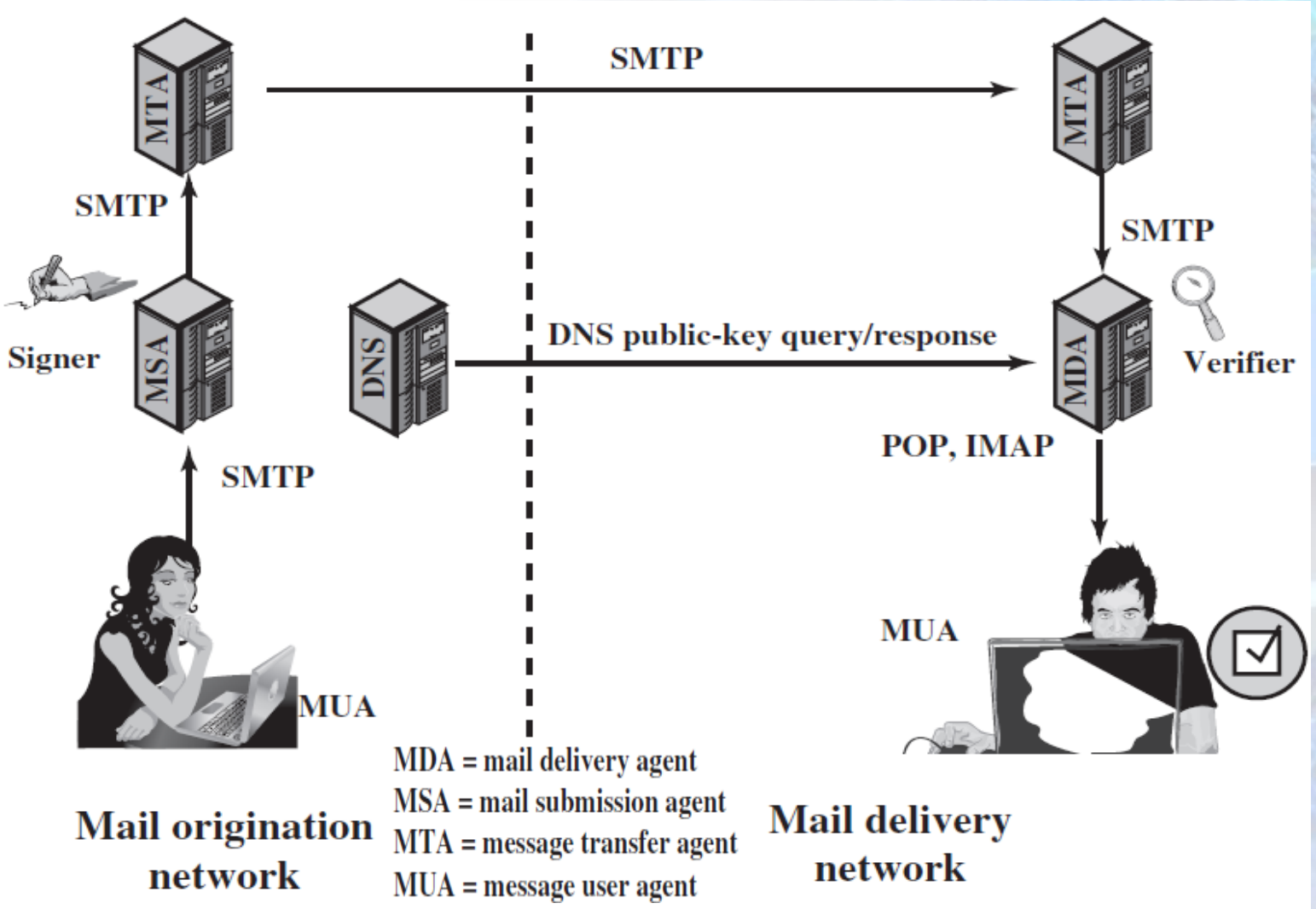
DomainKeys Identified Mail Strategy

- At the destination, the MDA retrieves the public key for the incoming signature and verifies the signature before passing the message on to the destination e-mail client.

DomainKeys Identified Mail Strategy

- The default signing algorithm is RSA with SHA-256.
- RSA with SHA-1 also may be used.
- We next look at a simple example of the operation of DKIM.

DomainKeys Identified Mail Strategy



Introduction to IPsec

Network Security

Introduction to IPsec

Objectives of the Topic

- After completing this topic, a student will be able to
  - Describe the basics of IPsec.

Introduction to IPsec

Figures and material in this topic have been adapted from

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

Introduction to IPsec

- In 1994, the Internet Architecture Board (IAB) issued a report titled “Security in the Internet Architecture” (RFC 1636).
- The report identified key areas for security mechanisms.

Introduction to IPsec

- the need to secure the network infrastructure from unauthorized monitoring, and control of network traffic and,

Introduction to IPsec

- the need to secure end user-to-end-user traffic using authentication and encryption mechanisms.

Introduction to IPsec

- IP-level security encompasses three functional areas:
- authentication,
- confidentiality,
- and key management.

Introduction to IPsec

- The authentication mechanism assures that a received packet was transmitted by party identified as source in the packet header.
- Also, this mechanism assures that packet has not been altered in transit.

Introduction to IPsec

- The confidentiality facility enables communicating nodes to encrypt messages to prevent eavesdropping by third parties.
- The key management facility is concerned with the secure exchange of keys.

Introduction to IPsec

- The principal feature of IPsec that is that it can encrypt and/or authenticate *all* traffic at the IP level.
- secure networking not only for applications that have security mechanisms but also for the many security-ignorant applications.

Introduction to IPsec

- Lets look at a typical scenario of IPsec usage.
- An organization maintains LANs at dispersed locations.
- Nonsecure IP traffic is conducted on each LAN.

Introduction to IPsec

- For traffic offsite, through some sort of private or public WAN, IPsec protocols are used.
- These protocols operate in networking devices, such as a router or firewall, that connect each LAN to the outside world.

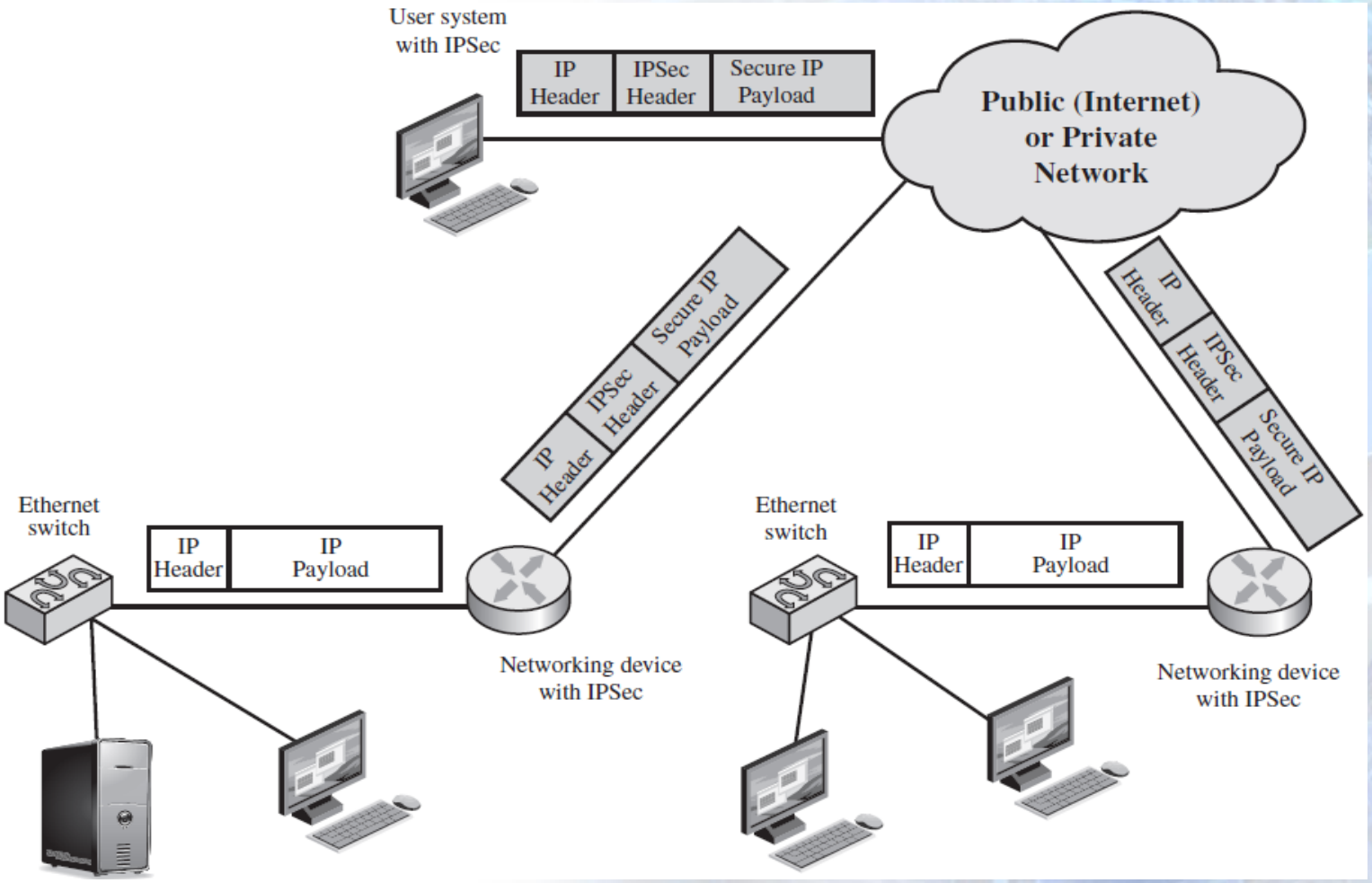
Introduction to IPsec

- The IPsec networking device will typically encrypt and compress all traffic going into the WAN and decrypt and decompress traffic coming from the WAN.

Introduction to IPsec

- These operations are transparent to workstations and servers on the LAN.
- Secure transmission is also possible with individual users who dial into the WAN.
- Such users must implement IPsec to provide security.

Introduction to IPsec



Introduction to IPsec

- **IPsec Documents:**
- **Architecture:**
- Covers the general concepts, security requirements, definitions, and mechanisms defining IPsec technology.
- RFC 4301, *Security Architecture for the Internet Protocol*.

Introduction to IPsec

- **Authentication Header (AH):**
- AH is an extension header to provide message authentication.
- RFC 4302, *IP Authentication Header*.

Introduction to IPsec

- Because message authentication is provided by ESP, the use of AH is deprecated.
- It is included in IPsecv3 for backward compatibility but should not be used in new applications.

Introduction to IPsec

- **Encapsulating Security Payload (ESP):**
- ESP consists of an encapsulating header and trailer used to provide encryption or combined encryption/authentication.

Introduction to IPsec

- RFC 4303, *IP Encapsulating Security Payload (ESP)*.
- **Internet Key Exchange (IKE):** describes the key management schemes for use with IPsec.
- RFC 5996, *Internet Key Exchange (IKEv2) Protocol*.

Introduction to IPsec

- **Cryptographic algorithms:**
- describes cryptographic algorithms for encryption, message authentication, pseudorandom functions (PRFs), and cryptographic key exchange.

Introduction to IPsec

End

- **Other:**
- There are a variety of other IPsec-related RFCs, including those dealing with security policy and management information base (MIB) content.

Applications and Benefits of IPsec

Network Security

Applications and Benefits of IPsec

Objectives of the Topic

- After completing this topic, a student will be able to
  - Describe the Applications and Benefits of IPsec.

Applications and Benefits of IPsec

Figures and material in this topic have been adapted from

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

Applications and Benefits of IPsec

- **IPsec Services:**
- IPsec provides security services at the IP layer by enabling a system to:
- select required security protocols,
- determine the algorithm(s) to use for the service(s), and

Applications and Benefits of IPsec

- put in place any cryptographic keys required to provide the requested services.

Applications and Benefits of IPsec

- Two protocols are used to provide security:
- an authentication protocol designated by the header of the protocol, Authentication Header (AH); and

Applications and Benefits of IPsec

- a combined encryption/authentication protocol designated by the format of the packet for that protocol, Encapsulating Security Payload (ESP).

Applications and Benefits of IPsec

- RFC 4301 lists the following services:
- Access control,
- Connectionless integrity,
- Data origin authentication ,
- Rejection of replayed packets,

Applications and Benefits of IPsec

- Confidentiality, and
- Limited traffic flow confidentiality.

Applications and Benefits of IPsec

- Examples of IPsec use include:
- **Secure branch office connectivity over the Internet:**
- A company can build a secure virtual private network over the Internet.

Applications and Benefits of IPsec

- This enables a business to rely heavily on the Internet and reduce its need for private networks, saving costs and network management overhead.

Applications and Benefits of IPsec

- **Secure remote access over the Internet:**
- An end user whose system is equipped with IP security protocols can make a local call to an Internet Service Provider (ISP) and gain secure access to a company network.

Applications and Benefits of IPsec

- This reduces the cost of toll charges for traveling employees and telecommuters.

Applications and Benefits of IPsec

- **Establishing extranet and intranet connectivity with partners:** IPsec can be used to secure communication with other organizations, ensuring authentication and confidentiality and providing a key exchange mechanism.

Applications and Benefits of IPsec

- **Enhancing electronic commerce security:**
- Some Web and electronic commerce applications have built-in security protocols.
- IPsec adds an additional layer of security to the data of application layer.

Applications and Benefits of IPsec

- **Benefits of IPsec:**
- 1. When IPsec is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter.

Applications and Benefits of IPsec

- Traffic within a company or workgroup does not incur the overhead of security-related processing.

Applications and Benefits of IPsec

- 2. IPsec in a firewall is resistant to bypass if all traffic from the outside must use IP and the firewall is the only means of entrance from the Internet into the organization.

Applications and Benefits of IPsec

- 3. IPsec is below the transport layer and is transparent to applications.
- There is no need to change software on a user or server system when IPsec is implemented in the firewall or router.

Applications and Benefits of IPsec

- 4. IPsec can be transparent to end users.
- no need to train users on security mechanisms, issue keying material on a per-user basis, or revoke keying material when users leave the organization.

Applications and Benefits of IPsec

- 5. IPsec can provide security for individual users if needed.
- This is useful for offsite workers and for setting up a secure virtual subnetwork within an organization for sensitive applications.

Applications and Benefits of IPsec

- **Routing Applications:**
- IPsec can play a vital role in the routing architecture required for internetworking.
- IPsec can assure that:
- A redirect message comes from the router to which the initial IP packet was sent.

Applications and Benefits of IPsec

End

- A router advertisement (a new router advertises its presence) comes from an authorized router.
- A neighbor advertisement comes from an authorized router.
- A routing update is not forged.

IPSec Security Association Database

Network Security

IPSec Security Association Database

Objectives of the Topic

- After completing this topic, a student will be able to
  - Describe security association database of IPSec.

IPSec Security Association Database

Figures and material in this topic have been adapted from

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

IPSec Security Association Database

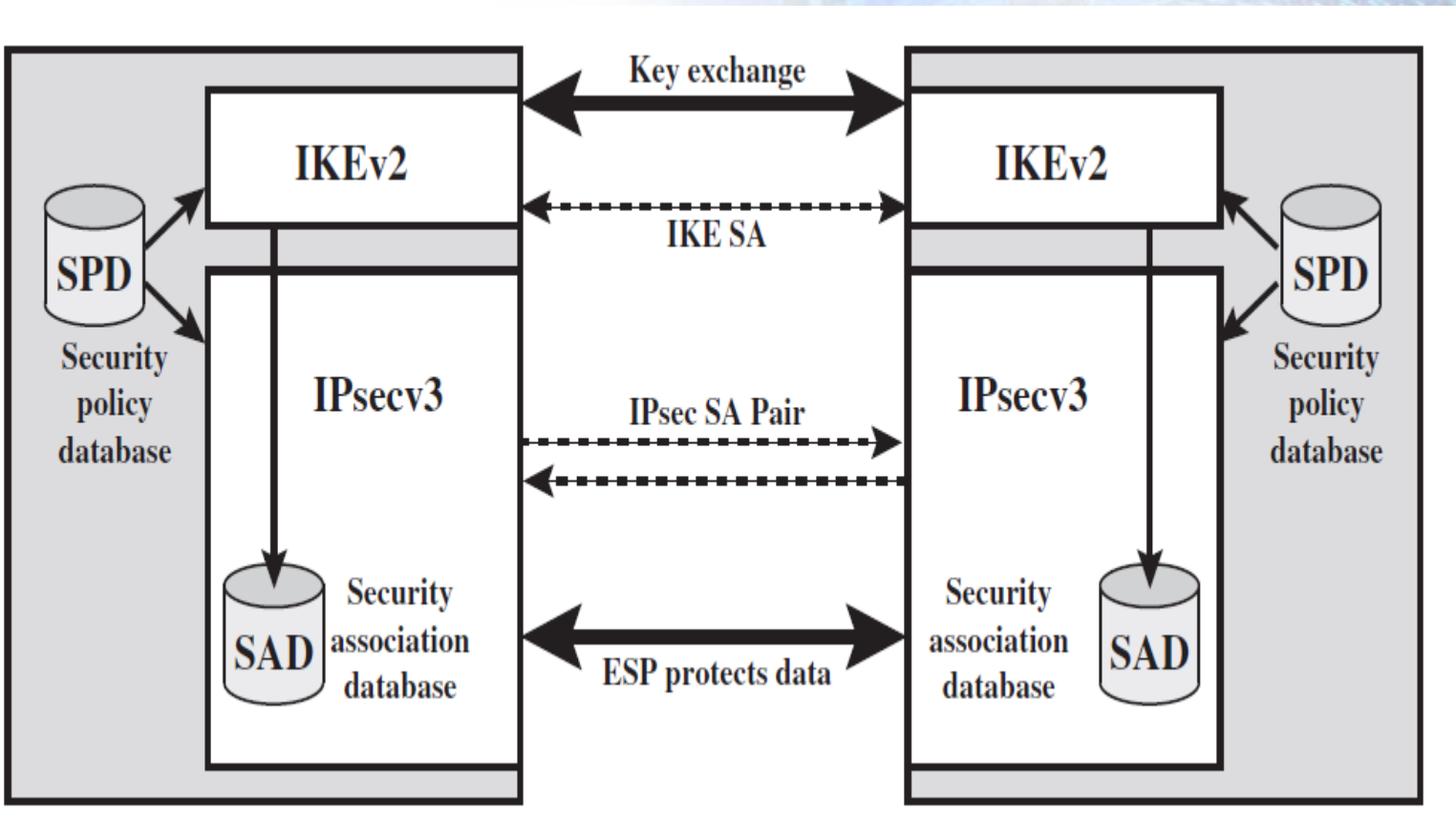
- Fundamental to the operation of IPsec is the concept of a security policy applied to each IP packet that transits from a source to a destination.

IPSec Security Association Database

- IPsec policy is determined by the interaction of two databases,
- **security association database (SAD)** and
- **security policy database (SPD)**.

IPSec Security Association Database

IPsec Architecture



IPSec Security Association Database

- **Security Associations**
- An association is a one-way logical connection between a sender and a receiver that affords security services to the traffic carried on it.

IPSec Security Association Database

- A security association is uniquely identified by three parameters.
- **Security Parameters Index (SPI):**
- A 32-bit unsigned integer assigned to this SA and having local significance only.

IPSec Security Association Database

- The SPI is carried in Authentication header (AH) and Encapsulating Security Payload (ESP) headers to enable the receiving system to select the SA under which a received packet will be processed.

IPSec Security Association Database

- **IP Destination Address:**
- This is the address of the destination endpoint of the SA, which may be an end-user system or a network system such as a firewall or router.

IPSec Security Association Database

- **Security Protocol Identifier:**
- This field from the outer IP header indicates whether the association is an AH or ESP security association.

IPSec Security Association Database

- In any IP packet, the security association is uniquely identified by the Destination Address in the IPv4 or IPv6 header and the SPI in the enclosed extension header (AH or ESP).

IPSec Security Association Database

- **Security Association Database (SAD):**
- A security association is normally defined by the following parameters in an SAD entry.

IPSec Security Association Database

- **Security Parameter Index:**
- A 32-bit value selected by receiving end of an SA to uniquely identify SA.
- In an SAD entry for an outbound SA, SPI is used to construct the packet's AH or ESP header.

IPSec Security Association Database

- In an SAD entry for an inbound SA, the SPI is used to map traffic to the appropriate SA.
- **Sequence Number Counter:**
- A 32-bit value used to generate the Sequence Number field in AH or ESP headers.

IPSec Security Association Database

- **Sequence Counter Overflow:**
- A flag indicating whether overflow of the Sequence Number Counter should generate an auditable event and prevent further transmission of packets on this SA.

IPSec Security Association Database

- **Anti-Replay Window:**
- Used to determine whether an inbound AH or ESP packet is a replay
- **AH Information:**
- Authentication algorithm, keys, key lifetimes, and related parameters being used with AH.

IPSec Security Association Database

- **ESP Information:**
- Encryption and authentication algorithm, keys, initialization values, key lifetimes, and related parameters being used with ESP.

IPSec Security Association Database

- **Lifetime of this Security Association:**
- A time interval or byte count after which an SA must be replaced with a new SA or terminated, plus an indication of which of these actions should occur.

IPSec Security Association Database

End

- **IPsec Protocol Mode:**
- Tunnel, transport, or wildcard.
- **Path MTU:**
- Any observed path MTU (maximum size of a packet that can be transmitted without fragmentation) and aging variables.

IPSec Security Policy Database



IPSec Security Policy Database

Objectives of the Topic

- After completing this topic, a student will be able to
  - Describe security policy database of IPSec.

IPSec Security Policy Database

Figures and material in this topic have been adapted from

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

IPSec Security Policy Database

- IPsec policy is determined by the interaction of two databases,
- **security association database (SAD)** and
- **security policy database (SPD)**.

IPSec Security Policy Database

- The means by which IP traffic is related to specific SAs is the Security Policy Database (SPD).
- An SPD contains entries, each of which defines a subset of IP traffic and points to an SA for that traffic.

IPSec Security Policy Database

- In more complex environments, there may be multiple entries that potentially relate to a single SA or multiple SAs associated with a single SPD entry.

IPSec Security Policy Database

- Each SPD entry is defined by a set of IP and upper-layer protocol field values called selectors.
- These selectors are used to filter outgoing traffic in order to map it into a particular SA.

IPSec Security Policy Database

- The following selectors determine an SPD entry:
- **Remote IP Address:**
- This may be a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address.

IPSec Security Policy Database

- The latter two are required to support more than one destination system sharing the same SA (e.g., behind a firewall).

IPSec Security Policy Database

- **Local IP Address:**
- This may be a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address.

IPSec Security Policy Database

- The latter two are required to support more than one source system sharing the same SA (e.g., behind a firewall).

IPSec Security Policy Database

- **Next Layer Protocol:**
- The IP protocol header includes a field that designates the protocol operating over IP.

IPSec Security Policy Database

- **Name:**
- A user identifier from the operating system.
- This is not a field in the IP or upper-layer headers but is available if IPsec is running on the same operating system as the user.

IPSec Security Policy Database

- **Local and Remote Ports:**
- These may be individual TCP or UDP port values, an enumerated list of ports, or a wildcard port.

IPSec Security Policy Database

An Example:

- Assume a local network configuration consists of two networks.
- The basic corporate network configuration has the IP network number 1.2.3.0/24.

IPSec Security Policy Database

- The local configuration also includes a secure LAN, often known as a DMZ, that is identified as 1.2.4.0/24.
- The DMZ is protected from both the outside world and the rest of the corporate LAN by firewalls.

IPSec Security Policy Database

- Lets assume that the host has the IP address 1.2.3.10, and it is authorized to connect to the server 1.2.4.10 in the DMZ.

IPSec Security Policy Database

An example of an SPD on a host system

Protocol	Local IP	Port	Remote IP	Port	Action	Comment
UDP	1.2.3.101	500	*	500	BYPASS	IKE
ICMP	1.2.3.101	*	*	*	BYPASS	Error messages
*	1.2.3.101	*	1.2.3.0/24	*	PROTECT: ESP intransport-mode	Encrypt intranet traffic
TCP	1.2.3.101	*	1.2.4.10	80	PROTECT: ESP intransport-mode	Encrypt to server
TCP	1.2.3.101	*	1.2.4.10	443	BYPASS	TLS: avoid double encryption
*	1.2.3.101	*	1.2.4.0/24	*	DISCARD	Others in DMZ
*	1.2.3.101	*	*	*	BYPASS	Internet

IPSec Security Policy Database

End

- The entries in the SPD are self-explanatory:
- E.g. UDP port 500 is the designated port for IKE.
- Any traffic from the local host to a remote host for purposes of an IKE exchange bypasses the IPsec processing.

IP Traffic Processing

Network Security

IP Traffic Processing

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe IP traffic processing.

IP Traffic Processing

Figures and material in this topic have been adapted from

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

IP Traffic Processing

- IPsec is executed on a packet-by-packet basis.
- When IPsec is implemented, each outbound IP packet is processed by the IPsec logic before transmission.

IP Traffic Processing

- Similarly, each inbound packet is processed by the IPsec logic after reception and before passing the packet contents on to the next higher layer (e.g., TCP or UDP).

IP Traffic Processing

- We look at the logic of these two situations in turn.

IP Traffic Processing

Outbound Packets:

- For outbound traffic, a block of data from a higher layer, such as TCP, is passed down to the IP layer and an IP packet is formed, consisting of an IP header and an IP body.

IP Traffic Processing

- Then the following steps occur:
- **1.** IPsec searches the Security Policy Database (SPD) for a match to this packet.
- **2.** If no match is found, then the packet is discarded and an error message is generated.

IP Traffic Processing

- **3.** If a match is found, further processing is determined by the first matching entry in the SPD.
- If the policy for this packet is DISCARD, then the packet is discarded.

IP Traffic Processing

- If the policy is BYPASS, then there is no further IPsec processing and the packet is forwarded to the network for transmission.

IP Traffic Processing

- **4.** If policy is PROTECT, then a search is made of the SAD for a matching entry.
- If no entry is found, then Internet Key Exchange (IKE) is invoked to create an SA with appropriate keys and an entry is made in the SA.

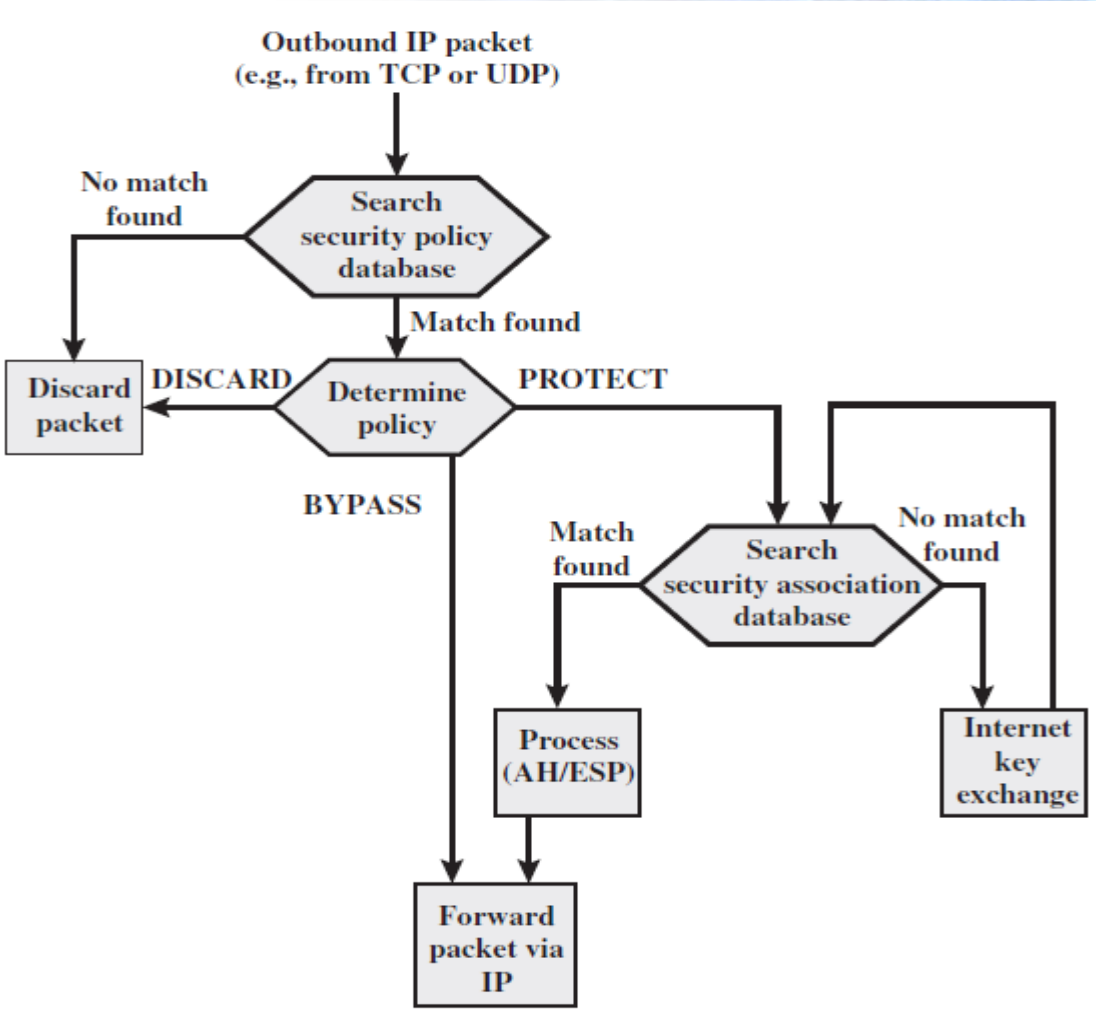
IP Traffic Processing

- **5.** The matching entry in Security association database (SAD) determines processing for this packet.
- Either encryption, authentication, or both can be performed, and either transport or tunnel mode can be used.

IP Traffic Processing

- The packet is then forwarded to the network for transmission.

IP Traffic Processing



IP Traffic Processing

Inbound Packets:

- An incoming IP packet triggers the IPsec processing.
- The following steps occur:

IP Traffic Processing

- **1.** IPsec determines whether this is an unsecured IP packet or one that has ESP or AH headers/trailers, by examining the IP Protocol field (IPv4) or Next Header field (IPv6).

IP Traffic Processing

- **2.** If the packet is unsecured, IPsec searches the SPD for a match to this packet.
- If first matching entry has a policy of BYPASS, IP header is processed, stripped off and packet body is delivered to next higher layer (e.g. TCP).

IP Traffic Processing

- If the first matching entry has a policy of PROTECT or DISCARD, or if there is no matching entry, the packet is discarded.

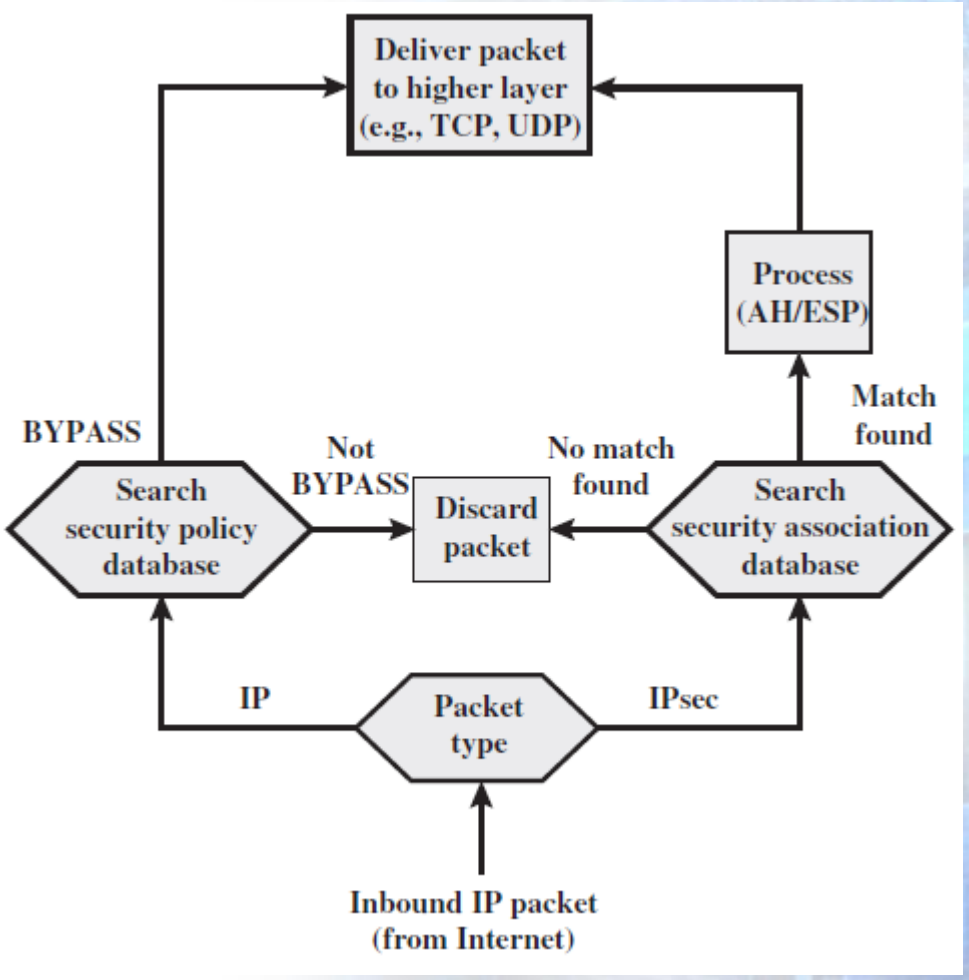
IP Traffic Processing

- **3.** For a secured packet, IPsec searches the SAD.
- If no match is found, the packet is discarded.
- Otherwise, IPsec applies the appropriate ESP or AH processing.

IP Traffic Processing

- Then, the IP header is processed and stripped off and the packet body is delivered to the next higher layer, such as TCP.

IP Traffic Processing



ESP and its Packet Format

Network Security

ESP and its Packet Format

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe basics of Encapsulating Security Payload (ESP) and its Packet Format.

ESP and its Packet Format

Figures and material in this topic have been adapted from

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

ESP and its Packet Format

ESP can provide:

- confidentiality,
- data origin authentication,
- connectionless integrity,
- an anti-replay service, and
- (limited) traffic flow confidentiality.

ESP and its Packet Format

- The set of services provided depends on options selected at the time of Security Association (SA) establishment and on the location of the implementation in a network topology.

ESP and its Packet Format

- ESP can work with a variety of encryption and authentication algorithms, including authenticated encryption algorithms such as Galois/Counter Mode (GCM).

ESP and its Packet Format

ESP Format:

- An ESP packet contains the following fields:
- **Security Parameters Index (32 bits):** Identifies a security association.

ESP and its Packet Format

- **Sequence Number (32 bits):**
- A monotonically increasing counter value.
- This provides an anti-replay function.

ESP and its Packet Format

- **Payload Data (variable):**
- This is a transport-level segment (transport mode) or IP packet (tunnel mode) that is protected by encryption.
- **Padding (0–255 bytes):**

ESP and its Packet Format

- **Pad Length (8 bits):** Indicates the number of pad bytes immediately preceding this field.

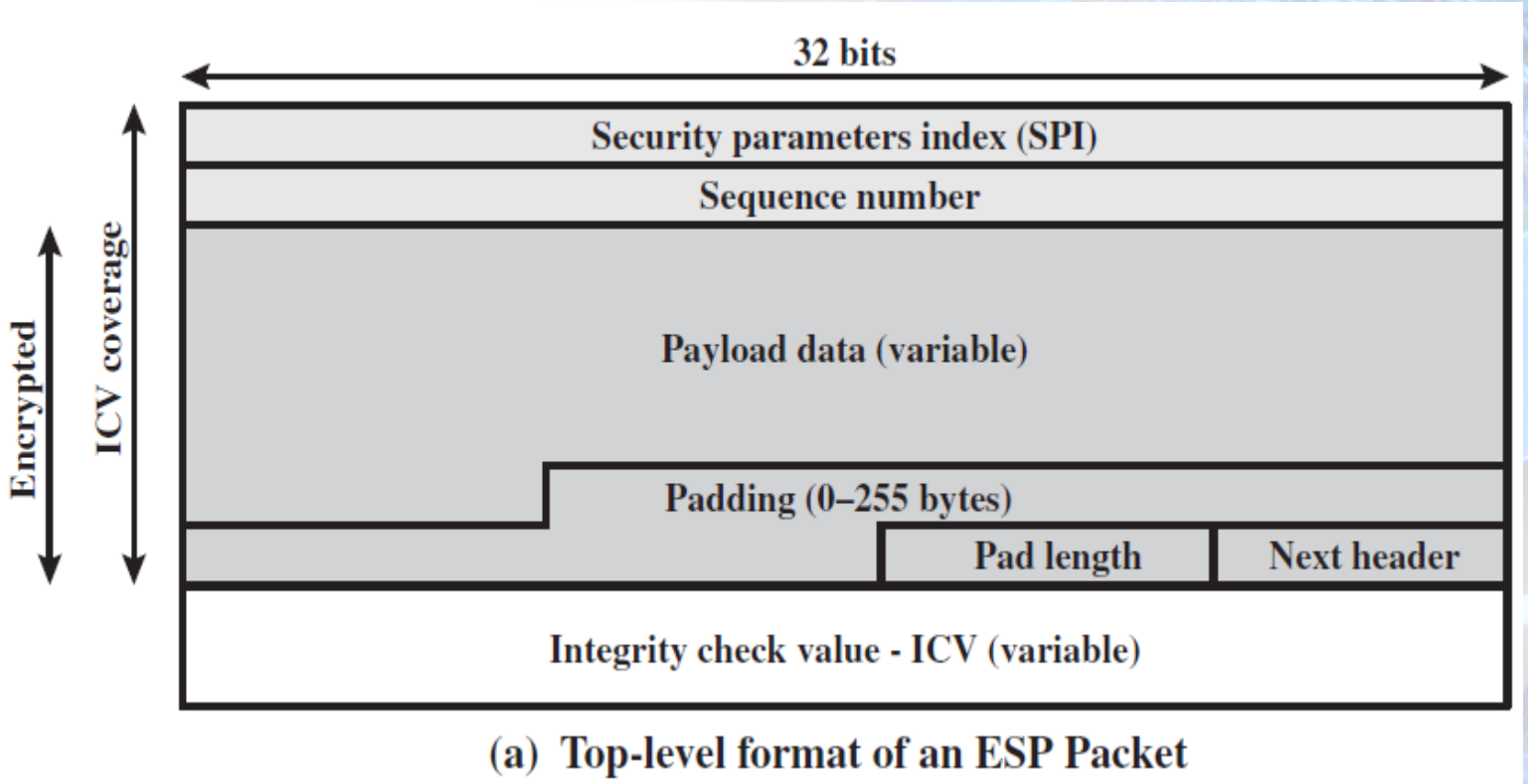
ESP and its Packet Format

- **Next Header (8 bits):**
- Identifies the type of data contained in the payload data field by identifying the first header in that payload (e.g., an extension header in IPv6, or an upper-layer protocol such as TCP).

ESP and its Packet Format

- **Integrity Check Value (variable):**
- A variable-length field (must be an integral number of 32-bit words) that contains the Integrity Check Value computed over the ESP packet minus the Authentication Data field.

ESP and its Packet Format



(a) Top-level format of an ESP Packet

ESP and its Packet Format

- When any combined mode algorithm is employed, the algorithm itself is expected to return both decrypted plaintext and a pass/fail indication for the integrity check.

ESP and its Packet Format

- For combined mode algorithms, the ICV that would normally appear at the end of the ESP packet may be omitted.

ESP and its Packet Format

- When the ICV is omitted and integrity is selected, it is the responsibility of the combined mode algorithm to encode within the Payload Data an ICV-equivalent means of verifying the integrity of the packet.

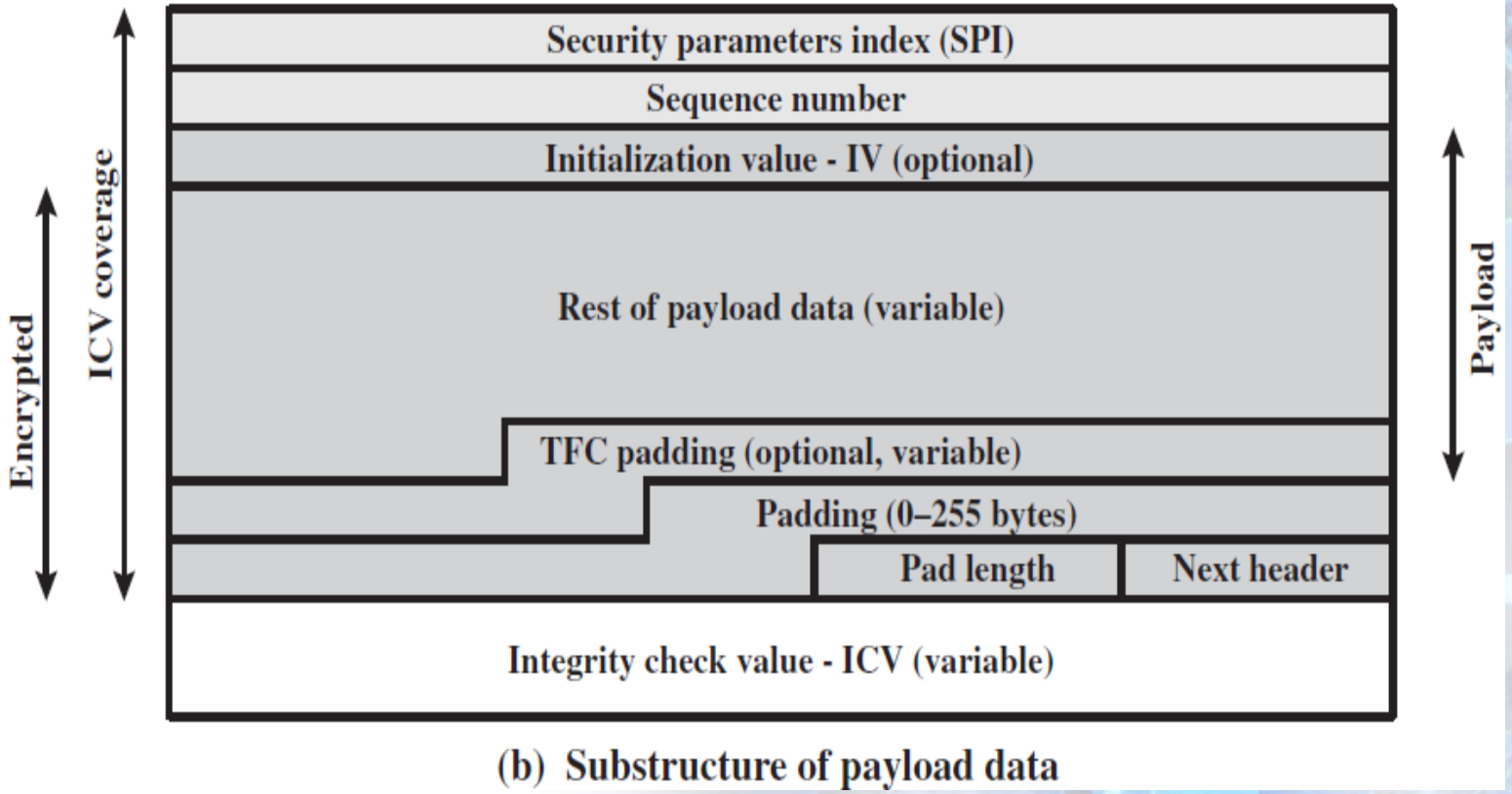
ESP and its Packet Format

- Two additional fields may be present in the payload.
- An **initialization value (IV)**, or nonce, is present if this is required by the encryption or authenticated encryption algorithm used for ESP.

ESP and its Packet Format

- If tunnel mode is being used, then the IPsec implementation may add **traffic flow confidentiality (TFC)** padding after the Payload Data and before the Padding field.

ESP and its Packet Format



(b) Substructure of payload data

Encryption & Authentication Algorithms

Network Security

Encryption & Authentication Algorithms

Objectives of the Topic

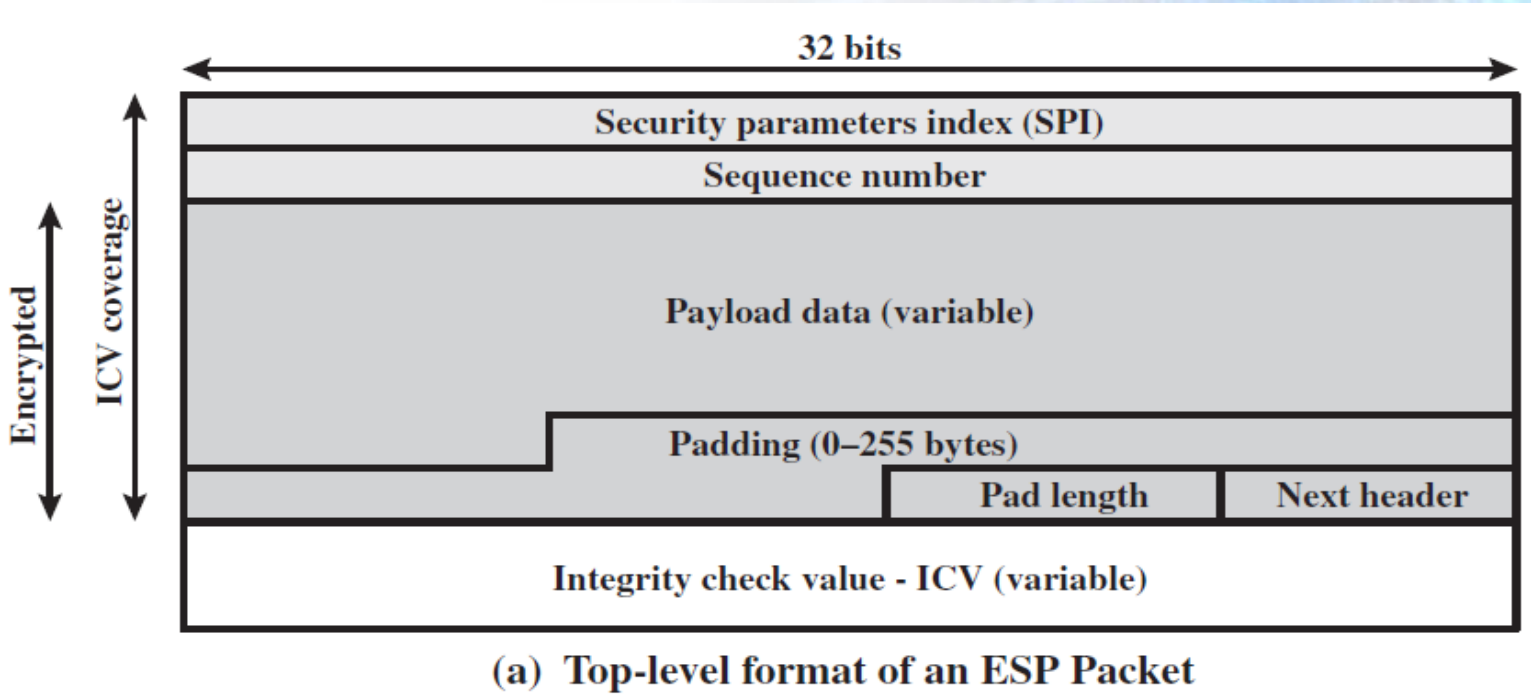
- After completing this topic, a student will be able to
  - explain encryption and authentication algorithms and anti-replay mechanism.

Encryption & Authentication Algorithms

Figures and material in this topic have been adapted from

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

Encryption & Authentication Algorithms



(a) Top-level format of an ESP Packet

- The Payload Data, Padding, Pad Length, and Next Header fields are encrypted by the ESP service.

Encryption & Authentication Algorithms

- If the encryption algorithm requires cryptographic synchronization data, such as an initialization vector (IV), then these data may be carried explicitly at beginning of Payload Data field.

Encryption & Authentication Algorithms

- If included, an IV is usually not encrypted.
- The ICV field is optional and is present only if the integrity service is selected.
- The ICV is computed after the encryption is performed.

Encryption & Authentication Algorithms

- This order of processing facilitates rapid detection and rejection of replayed or bogus packets by the receiver prior to decrypting the packet.
- Thus, reducing the impact of denial of service (DoS) attacks.

Encryption & Authentication Algorithms

- It also allows for the possibility of parallel processing of packets at the receiver
- i.e. decryption can take place in parallel with integrity checking.

Encryption & Authentication Algorithms

- **Padding:** serves several purposes:
- 1. ESP format requires that Pad Length and Next Header fields be right aligned within a 32-bit word, i.e. Ciphertext be integer multiple of 32 bits.
- Padding field is used for alignment.

Encryption & Authentication Algorithms

- 2. If an encryption algorithm requires plaintext to be a multiple of number of bytes, the Padding field is used to expand the plaintext (Payload Data, Padding, Pad Length, and Next Header fields) to required length.

Encryption & Authentication Algorithms

- 3. Additional padding may be added to provide partial traffic-flow confidentiality by concealing the actual length of the payload.

Encryption & Authentication Algorithms

Anti-Replay Service:

- A **replay attack** is one in which an attacker obtains a copy of an authenticated packet and later transmits it to the intended destination.

Encryption & Authentication Algorithms

- Receipt of duplicate, authenticated IP packets may disrupt service in some way or may have some other undesired consequence.
- Sequence Number field is designed to thwart such attacks.

Encryption & Authentication Algorithms

- When a new SA is established, **sender** initializes a sequence number counter to 0.
- Each time a packet is sent on this SA, the sender increments the counter and places the value in the Sequence Number field.

Encryption & Authentication Algorithms

- The first value to be used is 1.
- If anti-replay is enabled, the sender must not allow the sequence number to cycle past $2^{32} - 1$ back to zero.

Encryption & Authentication Algorithms

- Otherwise, there would be multiple valid packets with the same sequence number.
- If the limit of $2^{32} - 1$ is reached, the sender should terminate this SA and negotiate a new SA with a new key.

Encryption & Authentication Algorithms

- Because IP is a connectionless, unreliable service, the protocol does not guarantee that packets will be delivered in order and does not guarantee that all packets will be delivered.

Encryption & Authentication Algorithms

- IPsec authentication document dictates that the **receiver** should implement a window of size *W*, with a default of *W* = 64.

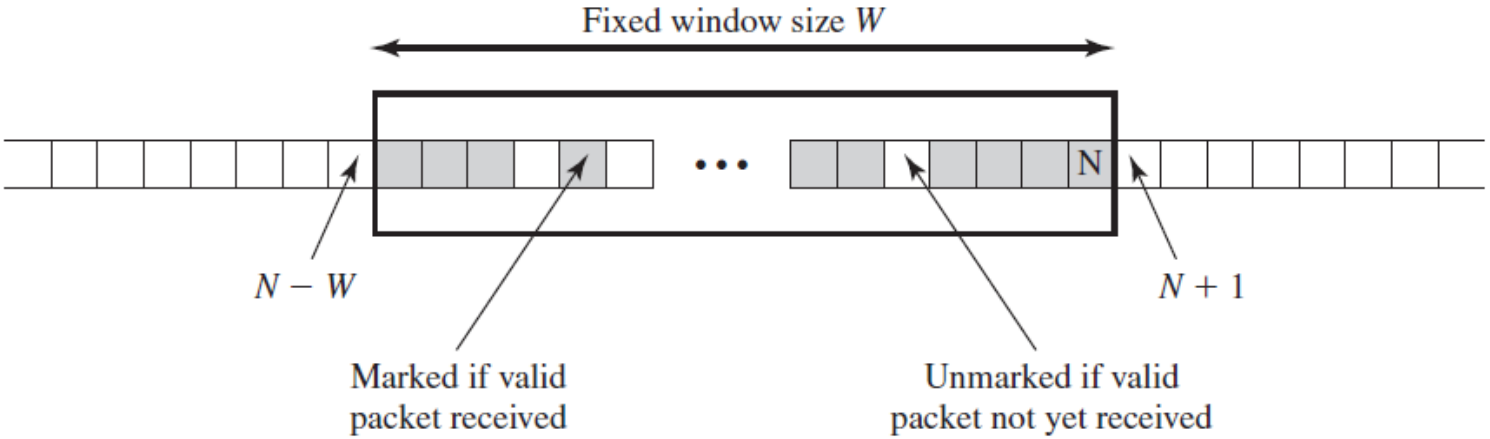
Encryption & Authentication Algorithms

- The right edge of the window represents the highest sequence number, *N*, so far received for a valid packet.

Encryption & Authentication Algorithms

- For any packet with a sequence number in the range from *N* - *W* + 1 to *N* that has been correctly received (i.e., properly authenticated), the corresponding slot in the window is marked.

Encryption & Authentication Algorithms



1. If the received packet falls within the window and is new, the MAC is checked. If the packet is authenticated, the corresponding slot in the window is marked.
2. If the received packet is to the right of the window and is new, the MAC is checked. If the packet is authenticated, the window is advanced so that this sequence number is the right edge of the window, and the corresponding slot in the window is marked.
3. If the received packet is to the left of the window or if authentication fails, the packet is discarded; this is an auditable event.

Transport and Tunnel ESP Modes

Network Security

Transport and Tunnel ESP Modes

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain transport and tunnel modes of ESP.

Transport and Tunnel ESP Modes

Figures and material in this topic have been adapted from

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

Transport and Tunnel ESP Modes

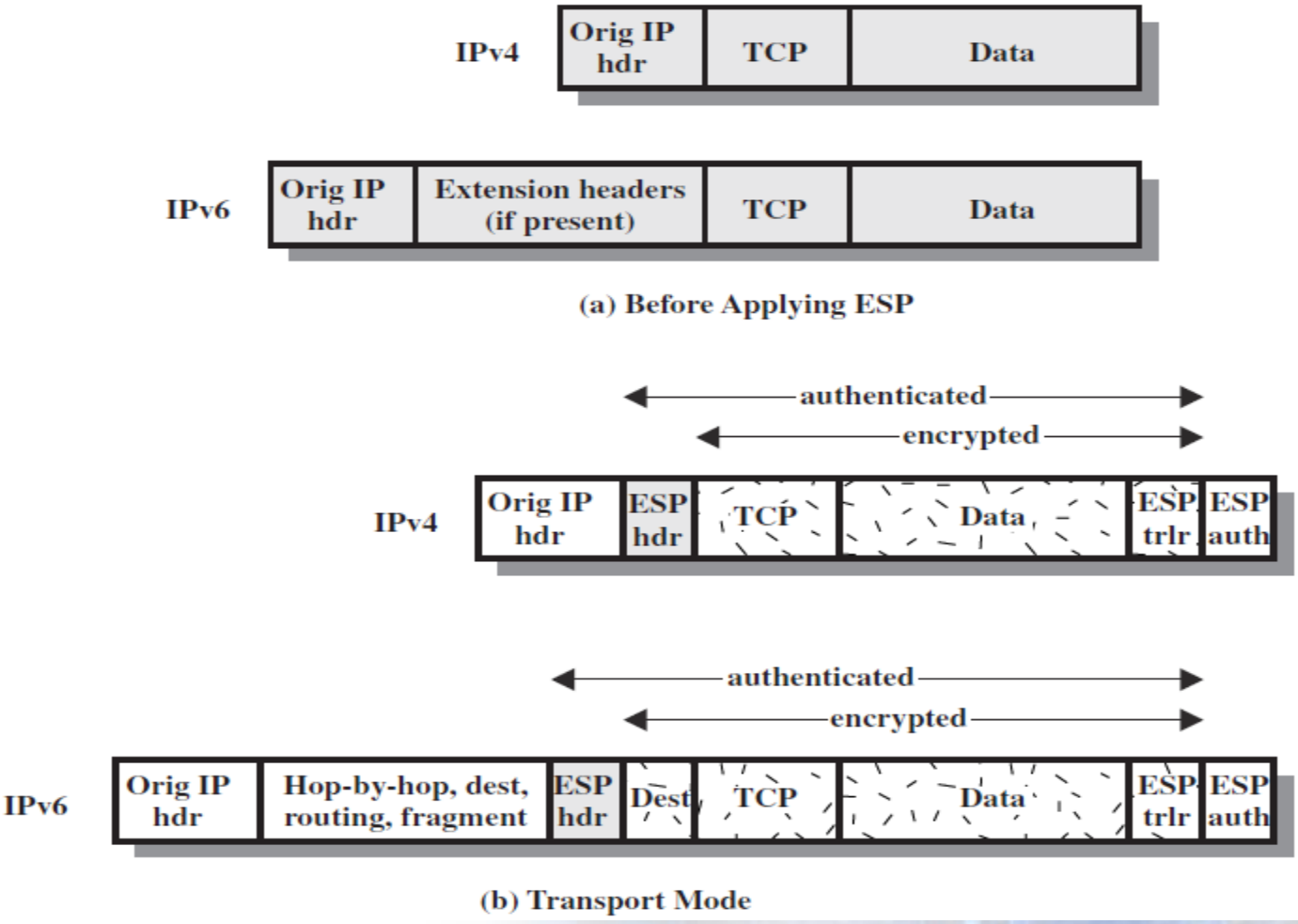
- The IPsec ESP service can be used in two ways:
- Transport mode ESP, and
- Tunnel mode ESP.

Transport and Tunnel ESP Modes

Transport Mode ESP:

- is used to encrypt and optionally authenticate the data carried by IP (e.g., a TCP segment).
- If authentication is selected, the ESP Authentication Data field is added after the ESP trailer.

Transport and Tunnel ESP Modes



Transport and Tunnel ESP Modes

- Summary of transport mode operation is:
- **1.** At the source, the block of data consisting of the ESP trailer plus the entire transport-layer segment is encrypted to form the IP packet for transmission.

Transport and Tunnel ESP Modes

- Authentication is added if this option is selected.
- **2.** The packet is then routed to destination.
- Each intermediate router needs to examine and process the IP header plus any plaintext IP extension headers.

Transport and Tunnel ESP Modes

- **3.** On the basis of the Security Parameters Index (SPI) in the ESP header, the destination node decrypts the packet to recover the plaintext transport-layer segment.

Transport and Tunnel ESP Modes

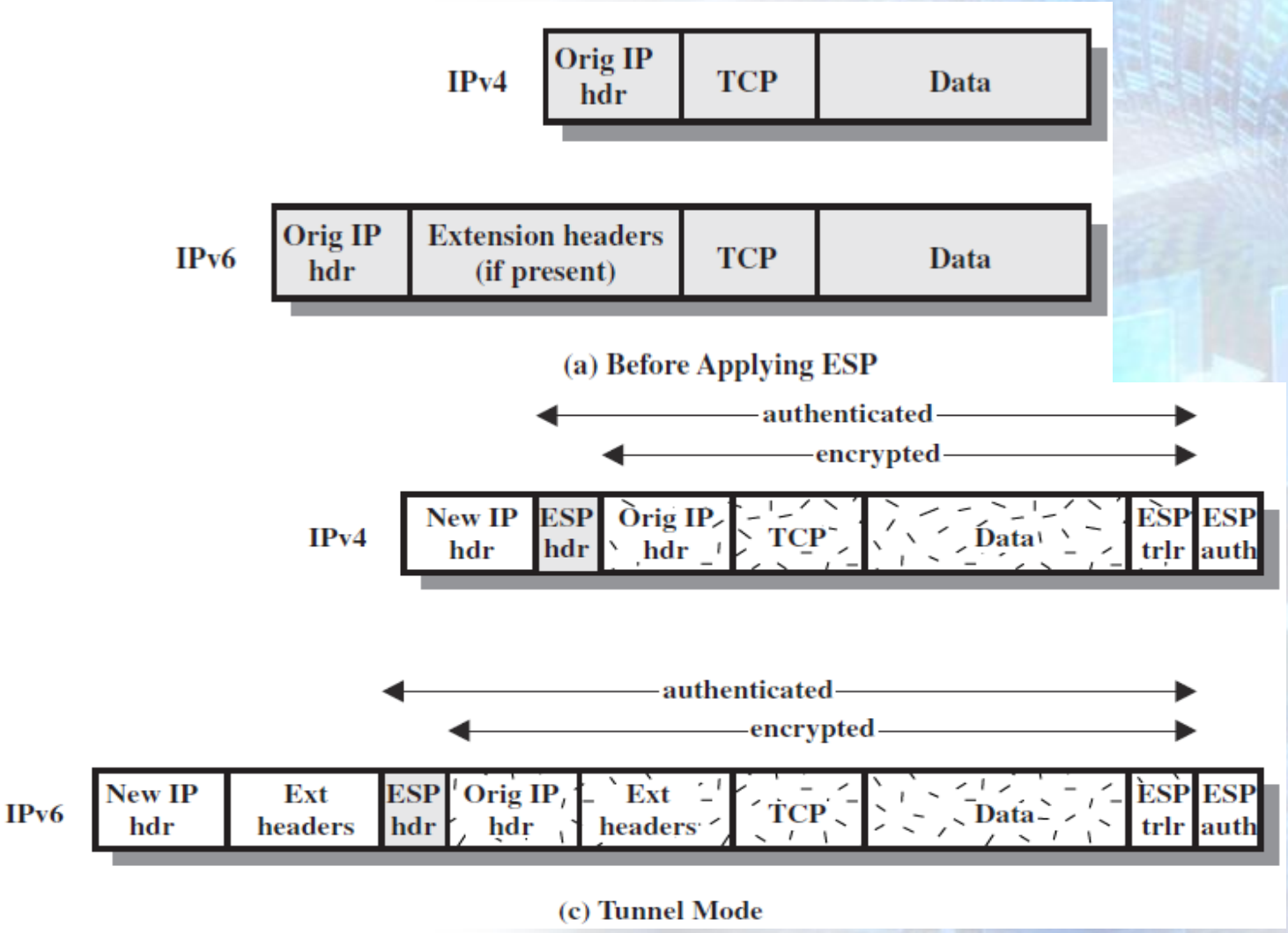
Tunnel Mode ESP:

- is used to encrypt an entire IP packet.
- For this mode, the ESP header is prefixed to the packet and then the packet plus the ESP trailer is encrypted.

Transport and Tunnel ESP Modes

- Intermediate routers would be unable to process such a packet.
- Encapsulate the entire block (ESP header plus ciphertext plus Authentication Data, if present) with a new IP header that will contain sufficient routing information.

Transport and Tunnel ESP Modes



Transport and Tunnel ESP Modes

- Lets assume there is an external host that wishes to communicate with a host on an internal network protected by a firewall.
- ESP is implemented in the external host and the firewalls.

Transport and Tunnel ESP Modes

- The following steps occur for transfer of a transport-layer segment from the external host to the internal host:

Transport and Tunnel ESP Modes

- **1.** The source prepares an inner IP packet with a destination address of target internal host.
- This packet is prefixed by an ESP header.
- The packet and ESP trailer are encrypted and Authentication Data may be added.

Transport and Tunnel ESP Modes

- The resulting block is encapsulated with a new IP header whose destination address is the firewall.
- This forms the outer IP packet.

Transport and Tunnel ESP Modes

- **2.** The outer packet is routed to destination firewall.
- Each intermediate router needs to examine and process the outer IP header plus any outer IP extension headers.

Transport and Tunnel ESP Modes

- **3.** The destination firewall examines and processes the outer IP header plus any outer IP extension headers.
- On basis of SPI in ESP header, destination node decrypts remainder of packet to recover plaintext inner IP packet.

Transport and Tunnel ESP Modes

- This packet is then transmitted in the internal network.
- **4.** The inner packet is routed through zero or more routers in the internal network to the destination host.

Transport and Tunnel ESP Modes

End

- Transport mode is suitable for protecting connections between hosts that support the ESP feature
- Tunnel mode is useful in a configuration that includes a firewall that protects a trusted network from external networks.

Combining Security Associations

Network Security

Combining Security Associations

Objectives of the Topic

- After completing this topic, a student will be able to
  - Explain how to combine security associations of IPsec.

Combining Security Associations

Figures and material in this topic have been adapted from

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

Combining Security Associations

- An individual SA can implement either the AH or ESP protocol but not both.
- A particular traffic flow may require IPsec services between hosts and, for that same flow, separate services between security gateways.

Combining Security Associations

- *Security association bundle* is a sequence of SAs through which traffic must be processed to provide a desired set of IPsec services.
- SAs in a bundle may terminate at different endpoints or at the same endpoints.

Combining Security Associations

- SAs may be combined in two ways:
- **Transport adjacency:**
- Apply more than one security protocol to the same IP packet without invoking tunneling.
- This approach allows for only one level of combination.

Combining Security Associations

- **Iterated tunneling:**
- Refers to application of multiple layers of security protocols effected through IP tunneling
- This approach allows for multiple levels of nesting.

Combining Security Associations

- **Issue Considering SA Bundles:**
- the order in which authentication and encryption may be applied between a given pair of endpoints and the ways of doing so.
- There are several approaches:

Combining Security Associations

- **1. ESP with Authentication Option:**
- The user first applies ESP to the data to be protected and then appends the authentication data field.
- There are two subcases:

Combining Security Associations

- Transport mode ESP, and
- Tunnel mode ESP
- For both methods, authentication applies to the ciphertext rather than the plaintext.

Combining Security Associations

- **2.Transport Adjacency**
- Applies authentication after encryption.
- Uses two bundled transport SAs, with inner being an ESP SA and outer being an AH SA.
- Here, ESP is used without its authentication option.

Combining Security Associations

- Encryption is applied to the IP payload.
- AH is then applied in transport mode.
- **Advantage:** is authentication covers more fields.
- **Disadvantage:** is the overhead of two SAs versus one SA.

Combining Security Associations

- The use of **authentication prior to encryption** might be preferable for several reasons:
- It is impossible for anyone to intercept message and alter the authentication data without detection.

Combining Security Associations

- It may be desirable to store authentication information with the message at the destination for later reference.

Combining Security Associations

- **3. Transport-Tunnel Bundle:**
- Uses a bundle consisting of an inner AH transport SA and an outer ESP tunnel SA.
- Authentication is applied to the IP payload plus the IP header.

Combining Security Associations

- The resulting IP packet is then processed in tunnel mode by ESP.
- The result is that the entire authenticated inner packet is encrypted and a new outer IP header is added.

Combining Security Associations

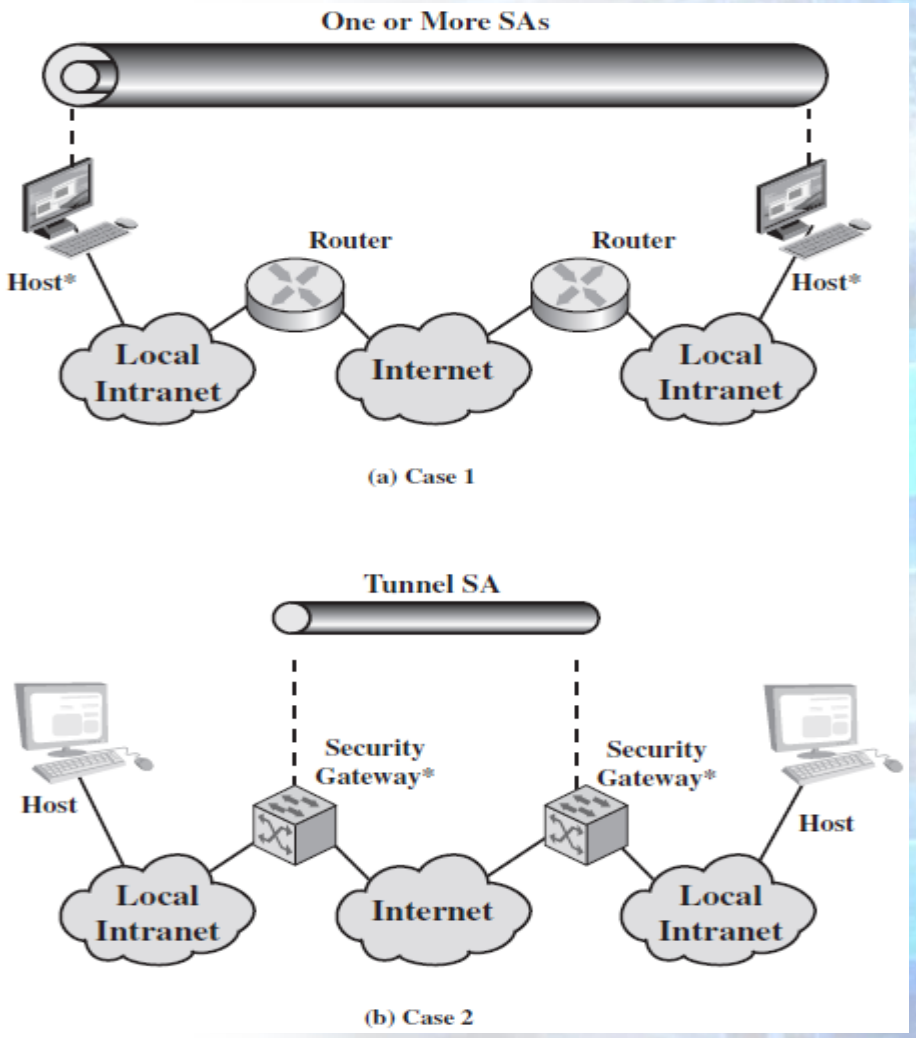
Basic Combinations of Security Associations:

- The IPsec Architecture document lists four combinations of SAs.
- **Case 1**. All security is provided between end systems that implement IPsec.

Combining Security Associations

- **Case 2.** Security is provided only between gateways (routers, firewalls, etc.) and no hosts implement IPsec.
- This case illustrates simple virtual private network support.

Combining Security Associations



* = implements IPsec

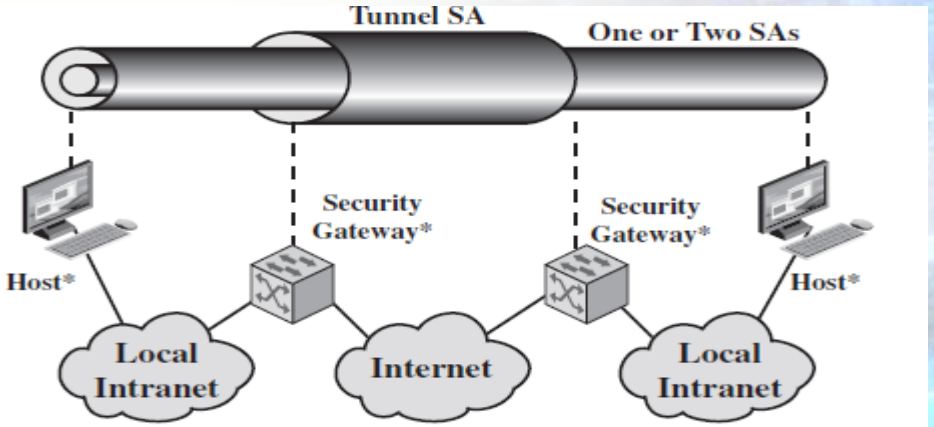
Combining Security Associations

- **Case 3**. This builds on case 2 by adding end-to-end security.
- The gateway-to-gateway tunnel provides either authentication, confidentiality, or both for all traffic between end systems.

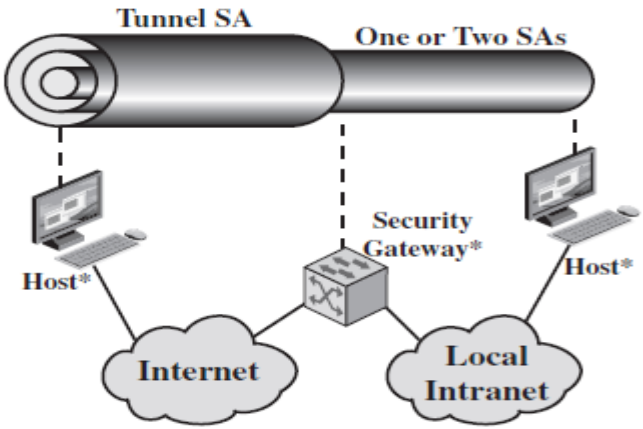
Combining Security Associations

- **Case 4.** This provides support for a remote host that uses the Internet to reach an organization's firewall and then to gain access to some server or workstation behind the firewall.

Combining Security Associations



(c) Case 3



(d) Case 4

* = implements IPsec

Internet Key Exchange (IKE)

Network Security

Internet Key Exchange (IKE)

Objectives of the Topic

- After completing this topic, a student will be able to
  - Explain the basics of Internet Key Exchange (IKE).

Internet Key Exchange (IKE)

Figures and material in this topic have been adapted from

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

Internet Key Exchange (IKE)

- The key management portion of IPsec involves the determination and distribution of secret keys.

Internet Key Exchange (IKE)

- A typical requirement is four keys for communication between two applications:
- transmit and receive pairs for both integrity and confidentiality.

Internet Key Exchange (IKE)

- The IPsec Architecture document mandates support for two types of key management:

Internet Key Exchange (IKE)

- **Manual:** A system administrator manually configures each system with its own keys and with the keys of other communicating systems.
- This is practical for small, relatively static environments.

Internet Key Exchange (IKE)

- **Automated:** An automated system enables the on-demand creation of keys for SAs and facilitates the use of keys in a large distributed system with an evolving configuration.

Internet Key Exchange (IKE)

- The default automated key management protocol for IPsec is **ISAKMP/Oakley**.
- It consists of the following elements:

Internet Key Exchange (IKE)

- **Oakley Key Determination Protocol:** Oakley is a key exchange protocol based on the Diffie-Hellman algorithm but providing added security.
- Oakley is generic as it does not dictate specific formats.

Internet Key Exchange (IKE)

- **Internet Security Association and Key Management Protocol (ISAKMP):**
- provides a framework for Internet key management, and

Internet Key Exchange (IKE)

- provides the specific protocol support, including formats, for negotiation of security attributes.
- ISAKMP consists of a set of message types that enable the use of a variety of key exchange algorithms.

Internet Key Exchange (IKE)

- In IKEv2, the terms Oakley and ISAKMP are no longer used, and there are significant differences from the use of Oakley and ISAKMP in IKEv1.
- Nevertheless, the basic functionality is the same.

Internet Key Exchange (IKE)

Features of IKE Key Determination:

- It is characterized by five important features:
- **1.** It employs a mechanism known as cookies to thwart clogging attacks.

Internet Key Exchange (IKE)

- **2.** It enables the two parties to negotiate a *group*.
- This specifies the global parameters of the Diffie-Hellman key exchange.
- **3.** It uses nonces to ensure against replay attacks.

Internet Key Exchange (IKE)

- **4.** It enables the exchange of Diffie-Hellman public key values.
- **5.** It authenticates the Diffie-Hellman exchange to thwart man-in-the-middle attacks.

Internet Key Exchange (IKE)

- In a **clogging attack**, an opponent forges the source address of a legitimate user and sends a public Diffie-Hellman key to the victim.
- The victim then performs a modular exponentiation to compute secret key.

Internet Key Exchange (IKE)

- Repeated messages of this type can clog the victim's system with useless work.

Internet Key Exchange (IKE)

- The **cookie exchange** requires that each side send a pseudorandom number, the **cookie**, in the initial message, which the other side acknowledges.

Internet Key Exchange (IKE)

- This acknowledgment must be repeated in the first message of the Diffie-Hellman key exchange.
- If the source address was forged, opponent gets no answer.

Internet Key Exchange (IKE)

- An opponent can only force a user to generate acknowledgments and not to perform Diffie-Hellman calculation.

Internet Key Exchange (IKE)

End

- The recommended method for creating the cookie is to perform a fast hash (e.g., MD5) over IP Source, Destination addresses, the UDP Source and Destination ports, and a locally generated secret value.

IKEv2 Exchanges

Network Security

IKEv2 Exchanges

Objectives of the Topic

- After completing this topic, a student will be able to
  - Describe exchanges of IKEv2.

IKEv2 Exchanges

Figures and material in this topic have been adapted from

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

IKEv2 Exchanges

- Internet Key Exchange (IKE) defines procedures and packet formats to establish, negotiate, modify, and delete security associations.

IKEv2 Exchanges

- The IKEv2 protocol involves the exchange of messages in pairs.
- The first two pairs of exchanges are referred to as the **initial exchanges.**

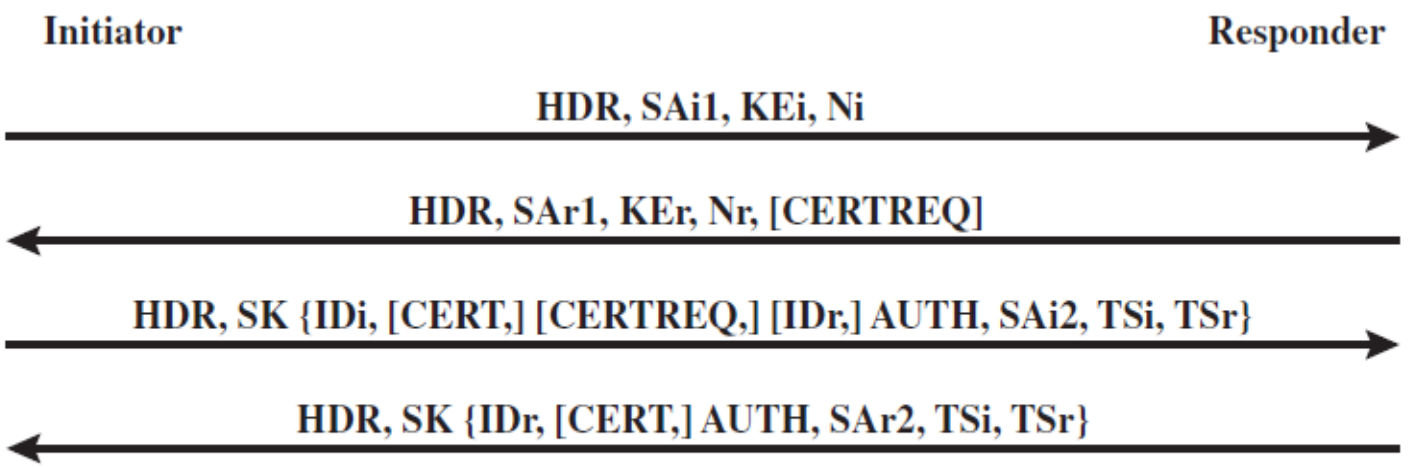
IKEv2 Exchanges

- In the first exchange, the two peers exchange information concerning cryptographic algorithms and other security parameters they are willing to use along with nonces and Diffie-Hellman (DH) values.

IKEv2 Exchanges

- Each nonce is a locally generated pseudorandom number.
- Nonces appear in responses and are encrypted during certain portions of the exchange to secure their use.

IKEv2 Exchanges



(a) Initial exchanges

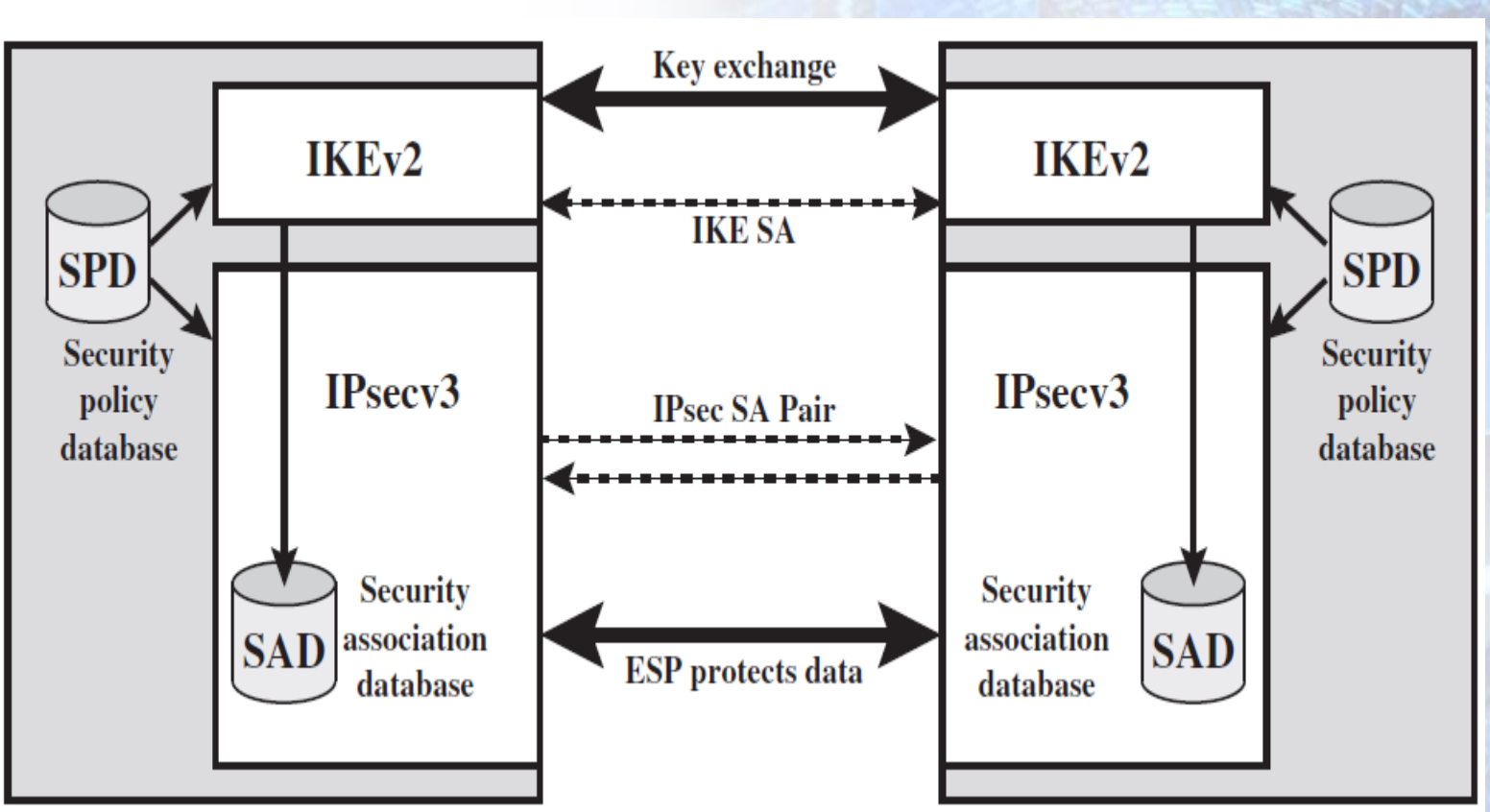
HDR = IKE header
SAx1 = offered and chosen algorithms, DH group
KEx = Diffie-Hellman public key
Nx= nonces
CERTREQ = Certificate request
IDx = identity
CERT = certificate

SK {...} = MAC and encrypt
AUTH = Authentication
SAx2 = algorithms, parameters for IPsec SA
TSx = traffic selectors for IPsec SA
N = Notify
D = Delete
CP = Configuration

IKEv2 Exchanges

- The result of this exchange is to set up a special SA called the IKE SA.
- This SA defines parameters for a secure channel between the peers over which subsequent message exchanges take place.

IKEv2 Exchanges



IKEv2 Exchanges

- All subsequent IKE message exchanges are protected by encryption and message authentication.

IKEv2 Exchanges

- In the second exchange, the two parties authenticate one another and set up a first IPsec SA to be placed in the SADB and used for protecting ordinary (i.e. non-IKE) communications between the peers.

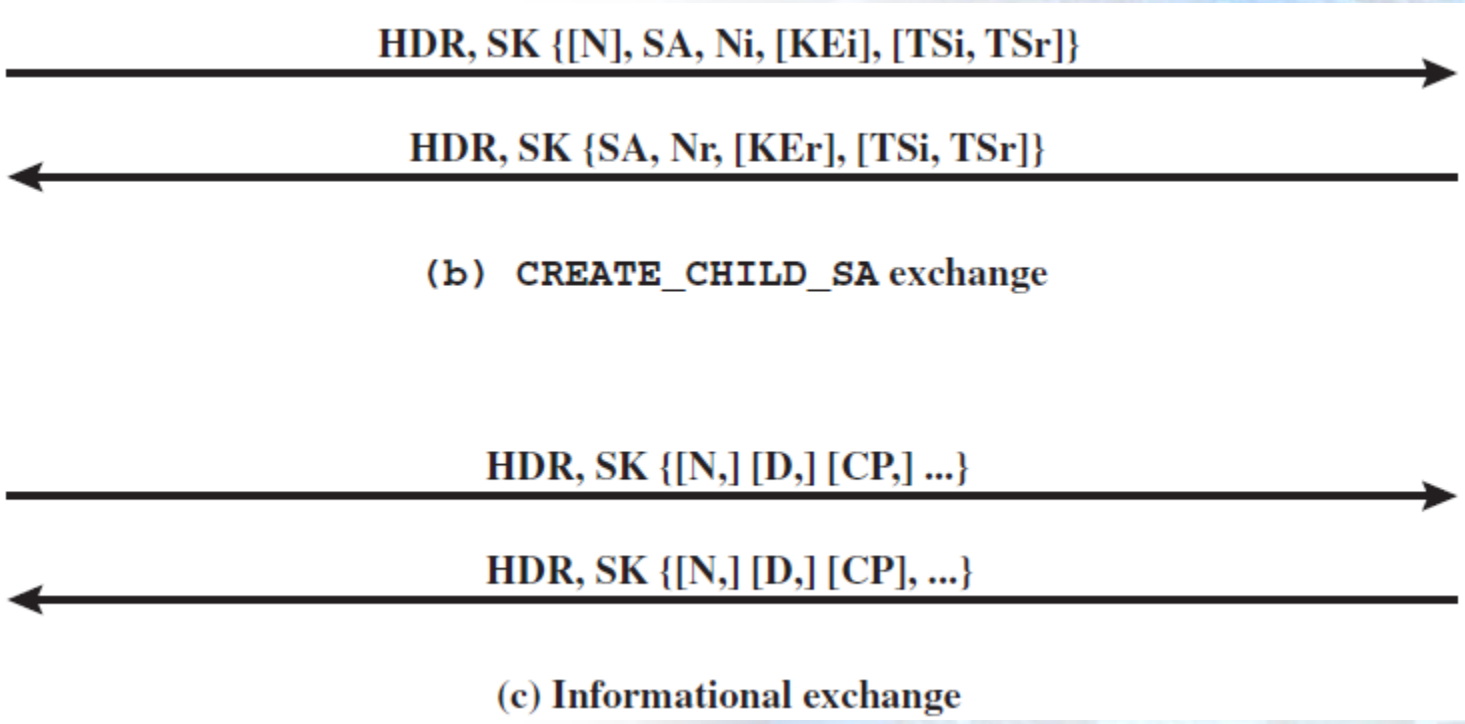
IKEv2 Exchanges

- Thus, four messages are needed to establish the first SA for general use.
- The **CREATE_CHILD_SA exchange** can be used to establish further SAs for protecting traffic.

IKEv2 Exchanges

- The **informational exchange** is used to exchange management information, IKEv2 error messages, and other notifications.

IKEv2 Exchanges



HDR = IKE header
SAx1 = offered and chosen algorithms, DH group
KEx = Diffie-Hellman public key
Nx= nonces
CERTREQ = Certificate request
IDx = identity
CERT = certificate

SK {...} = MAC and encrypt
AUTH = Authentication
SAx2 = algorithms, parameters for IPsec SA
TSx = traffic selectors for IPsec SA
N = Notify
D = Delete
CP = Configuration

IKEv2 Exchanges

Authentication methods used with IKE key determination:

- There are Three different methods:

IKEv2 Exchanges

- **1. Digital signatures:**
- The exchange is authenticated by signing a mutually obtainable hash.
- Each party encrypts the hash with its private key.

IKEv2 Exchanges

- The hash is generated over important parameters, such as user IDs and nonces.

IKEv2 Exchanges

- **2. Public-key encryption:** The exchange is authenticated by encrypting parameters such as IDs and nonces with the sender's private key.

IKEv2 Exchanges

End

- **3. Symmetric-key encryption:** A key derived by some out-of-band mechanism can be used to authenticate the exchange by symmetric encryption of exchange parameters.

IKE Header and Payload Formats

Network Security

IKE Header and Payload Formats

Objectives of the Topic

- After completing this topic, a student will be able to
  - Describe IKE header and payload formats.

IKE Header and Payload Formats

Figures and material in this topic have been adapted from

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

IKE Header and Payload Formats

IKE Header Format:

- An IKE message consists of an IKE header followed by one or more payloads.
- All of this is carried in a transport protocol.
- Implementations must support the use of UDP for the transport protocol.

IKE Header and Payload Formats

- The header format for an IKE message consists of the following fields:
- **Initiator SPI (64 bits):** A value chosen by the initiator to identify a unique IKE security association (SA).

IKE Header and Payload Formats

- **Responder SPI (64 bits):** A value chosen by the responder to identify the unique IKE SA.
- **Next Payload (8 bits):** Indicates type of first payload in message.
- **Major Version (4 bits):** Indicates major version of IKE in use.

IKE Header and Payload Formats

- **Minor Version (4 bits):** Indicates minor version in use.
- **Exchange Type (8 bits):** Indicates the type of exchange.
- **Length (32 bits):** Length of total message (header plus all payloads) in octets.

IKE Header and Payload Formats

- **Message ID (32 bits):** Used to control retransmission of lost packets and matching of requests and responses.
- **Flags (8 bits):** Three bits are defined so far.

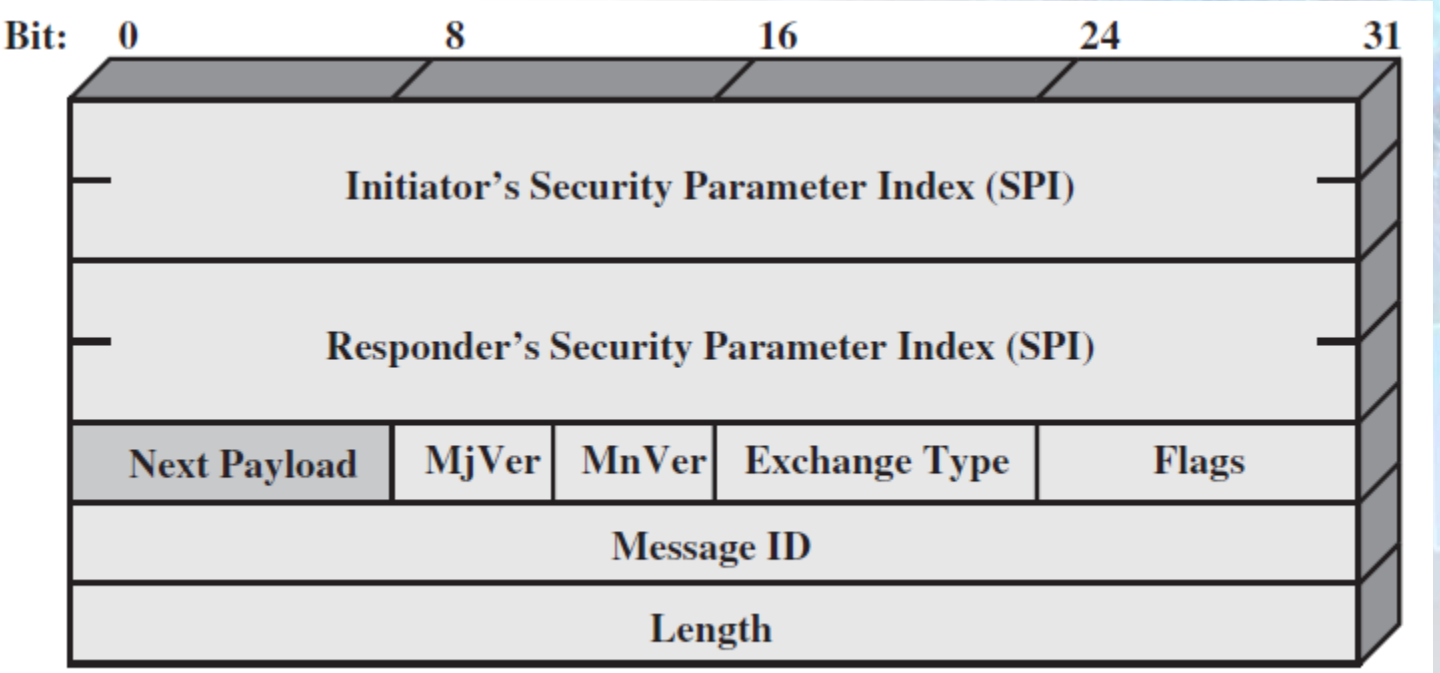
IKE Header and Payload Formats

- a) The initiator bit indicates whether this packet is sent by the SA initiator.
- b) The version bit indicates whether the transmitter is capable of using a higher major version number than the one currently indicated.

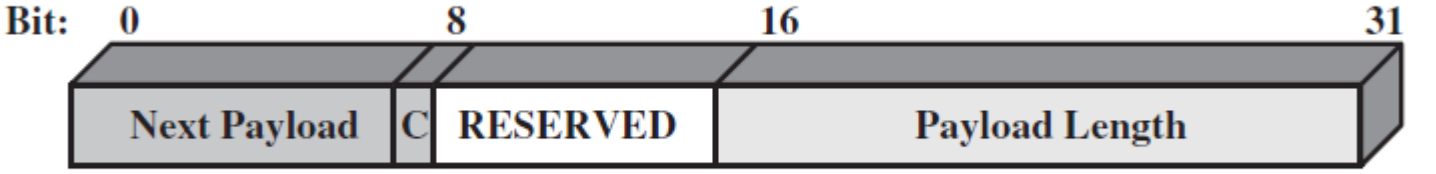
IKE Header and Payload Formats

- c) The response bit indicates whether this is a response to a message containing the same message ID.

IKE Header and Payload Formats



(a) IKE header



(b) Generic Payload header

IKE Header and Payload Formats

IKE Payload Types:

- All IKE payloads begin with the same generic payload header.
- The **Next Payload** field has a value of 0 if this is the last payload in the message; otherwise its value is the type of the next payload.

IKE Header and Payload Formats

- The **Payload Length** field indicates the length in octets of this payload, including the generic payload header.

IKE Header and Payload Formats

- The critical bit is 0 if the sender wants the recipient to skip this payload if it does not understand the payload type code in the Next Payload field of the previous payload.

IKE Header and Payload Formats

- It is set to 1 if the sender wants the recipient to reject this entire message if it does not understand the payload type.

IKE Header and Payload Formats

- The **SA payload** is used to establish SA.
- The **Key Exchange payload** can be used for a variety of key exchange techniques.
- The **Identification payload** is used to determine the identity of communicating peers.

IKE Header and Payload Formats

- The **Certificate payload** transfers a public-key certificate.
- At any point in an IKE exchange, the sender may include a **Certificate Request payload** to request the certificate of the other communicating entity.

IKE Header and Payload Formats

- The **Authentication payload** contains data used for message authentication purposes.
- The **Nonce payload** contains random data used to guarantee liveness during an exchange to protect against replay attacks.

IKE Header and Payload Formats

- The **Notify payload** contains either error or status information associated with this SA or this SA negotiation.
- The **Vendor ID payload** contains a vendor-defined constant.

IKE Header and Payload Formats

- The **Delete payload** indicates one or more SAs that the sender has deleted from its database and that therefore are no longer valid.

IKE Header and Payload Formats

- The **Traffic Selector payload** allows peers to identify packet flows for processing by IPsec services.
- The **Encrypted payload** contains other payloads in encrypted form.

IKE Header and Payload Formats

End

- The **Configuration payload** exchanges configuration information between IKE peers.
- The **Extensible Authentication Protocol (EAP)** payload allows IKE SAs to be authenticated using EAP.

Types Of Malicious Software

Network Security

Types Of Malicious Software

Objectives of the Topic

- After completing this topic, a student will be able to
  - Describe types of malicious software.

Types Of Malicious Software

Figures and material in this topic have been adapted from

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

Types Of Malicious Software

- Malicious software, or malware, arguably constitutes one of the most significant categories of threats to computer systems.

Types Of Malicious Software

What's Malware?

- a program that is covertly inserted into another program with the intent to destroy data, run destructive or intrusive programs, or otherwise compromise the confidentiality, integrity, or availability of the victim's data, applications, or operating system.

Types Of Malicious Software

- We are concerned with the threat malware poses to application programs, to utility programs, such as editors and compilers, and to kernel-level programs.

Types Of Malicious Software

- We are also concerned with its use on compromised or malicious Web sites and servers, or in especially crafted spam e-mails or other messages, which aim to trick users into revealing sensitive personal information.

Types Of Malicious Software

- The terminology in this area presents problems because of a lack of universal agreement on all of the terms and because some of the categories overlap.
- Next, we present some of the terms in use.

Types Of Malicious Software

Name	Description
Virus	Malware that, when executed, tries to replicate itself into other executable code; when it succeeds the code is said to be infected. When the infected code is executed, the virus also executes.
Worm	A computer program that can run independently and can propagate a complete working version of itself onto other hosts on a network.
Logic bomb	A program inserted into software by an intruder. A logic bomb lies dormant until a predefined condition is met; the program then triggers an unauthorized act.
Trojan horse	A computer program that appears to have a useful function, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entity that invokes the Trojan horse program.
Backdoor (trapdoor)	Any mechanism that bypasses a normal security check; it may allow unauthorized access to functionality.
Mobile code	Software (e.g., script, macro, or other portable instruction) that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics.
Exploits	Code specific to a single vulnerability or set of vulnerabilities.

Types Of Malicious Software

Downloaders	Program that installs other items on a machine that is under attack. Usually, a downloader is sent in an e-mail.
Auto-rooter	Malicious hacker tools used to break into new machines remotely.
Kit (virus generator)	Set of tools for generating new viruses automatically.
Spammer programs	Used to send large volumes of unwanted e-mail.
Flooders	Used to attack networked computer systems with a large volume of traffic to carry out a denial-of-service (DoS) attack.
Keyloggers	Captures keystrokes on a compromised system.
Rootkit	Set of hacker tools used after attacker has broken into a computer system and gained root-level access.
Zombie, bot	Program activated on an infected machine that is activated to launch attacks on other machines.
Spyware	Software that collects information from a computer and transmits it to another system.
Adware	Advertising that is integrated into software. It can result in pop-up ads or redirection of a browser to a commercial site.

Types Of Malicious Software

A Broad classification of malware:

- Malware can be classified into two broad categories:
- 1. Based first on how it spreads or propagates to reach the desired targets, and

Types Of Malicious Software

- 2. Then, on the actions or payloads it performs once a target is reached.

Types Of Malicious Software

- **Propagation mechanisms:**
- infection of existing executable or interpreted content by viruses that is subsequently spread to other systems

Types Of Malicious Software

- Exploit of software vulnerabilities either locally or over a network by worms or drive-by-downloads to allow the malware to replicate, and

Types Of Malicious Software

- Social engineering attacks that convince users to bypass security mechanisms to install trojans or to respond to phishing attacks.

Types Of Malicious Software

- **Payload actions performed by malware once it reaches a target system can include:**
- Corruption of system or data files
- Theft of service to make system a zombie agent of attack as part of a botnet

Types Of Malicious Software

End

- Theft of information from the system, e.g. logins, passwords, or other personal details by keylogging or spyware programs
- Stealthing where the malware hides its presence on system from attempts to detect and block it

Nature of Viruses

Network Security

Nature of Viruses

Objectives of the Topic

- After completing this topic, a student will be able to
  - Explain nature of viruses.

Nature of Viruses

Figures and material in this topic have been adapted from

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

Nature of Viruses

- A computer virus is a piece of software that can “infect” other programs, or any type of executable content, by modifying them.

Nature of Viruses

- The modification includes injecting the original code with a routine to make copies of the virus code, which can then go on to infect other content.

Nature of Viruses

- A computer virus carries in its instructional code the recipe for making perfect copies of itself.
- The typical virus becomes embedded in a program, or carrier of executable content on computer.

Nature of Viruses

- Then, whenever the infected computer comes into contact with an uninfected piece of code, a fresh copy of the virus passes into the new location.

Nature of Viruses

- Thus, the infection can spread from computer to computer, aided by unsuspecting users, who exchange these programs or carrier files on disk or USB stick, or who send them to one another over a network.

Nature of Viruses

- In a network environment, the ability to access documents, applications, and system services on other computers provides a perfect culture for the spread of such viral code.

Nature of Viruses

- A virus that attaches to an executable program can do anything that the program is permitted to do.
- It executes secretly when the host program is run.

Nature of Viruses

- Once the virus code is executing, it can perform any function, such as erasing files and programs, that is allowed by the privileges of the current user.

Nature of Viruses

- One reason viruses dominated the malware scene in earlier years was the lack of user authentication and access controls on personal computer systems at that time.

Nature of Viruses

- This enabled a virus to infect any executable content on the system.
- The significant quantity of programs shared on floppy disk also enabled its easy, if somewhat slow, spread.

Nature of Viruses

- The inclusion of tighter access controls on modern operating systems significantly hinders the ease of infection of such traditional, machine-executable code, viruses.

Nature of Viruses

- This resulted in the development of **macro viruses** that exploit the active content supported by some document types, such as Microsoft Word or Excel files, or Adobe PDF documents.

Nature of Viruses

- Such documents are easily modified and shared by users as part of their normal system use and are not protected by the same access controls as programs.

Nature of Viruses

End

- Currently, a viral mode of infection is typically one of several propagation mechanisms used by contemporary malware, which may also include worm and Trojan capabilities.

Virus Structure

Network Security

Virus Structure

Objectives of the Topic

- After completing this topic, a student will be able to
  - Explain structure of viruses.

Virus Structure

Figures and material in this topic have been adapted from

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

Virus Structure

- A computer virus, and more generally many contemporary types of malware, includes one or more variants of each of these components:

Virus Structure

- **Infection mechanism:**
- The means by which a virus spreads or propagates, enabling it to replicate.
- The mechanism is also referred to as the infection vector.

Virus Structure

- **Trigger:**
- The event or condition that determines when the payload is activated or delivered, sometimes known as a **logic bomb**.

Virus Structure

- **Payload:**
- What the virus does, besides spreading.
- The payload may involve damage or benign but noticeable activity.

Virus Structure

- During its lifetime, a typical virus goes through the following four phases:
- **1. Dormant phase:**
- The virus is idle.

Virus Structure

- The virus will eventually be activated by some event, such as a date, the presence of another program or file, or capacity of the disk exceeding some limit.
- Not all viruses have this stage.

Virus Structure

- **2. Propagation phase:**
- The virus places a copy of itself into other programs or into certain system areas on the disk.
- The copy may not be identical to the propagating version.
- Viruses often morph to evade detection.

Virus Structure

- Each infected program will now contain a clone of the virus, which will itself enter a propagation phase.

Virus Structure

- **3. Triggering phase:**
- The virus is activated to perform the function for which it was intended.

Virus Structure

- The triggering phase can be caused by a variety of system events, including a count of the number of times that this copy of the virus has made copies of itself.

Virus Structure

- **4. Execution phase:**
- The function is performed.
- The function may be harmless, such as a message on the screen, or damaging, such as the destruction of programs and data files.

Virus Structure

- A machine-executable virus code can be prepended or postpended to some executable program, or it can be embedded into the program in some other fashion.

Virus Structure

- The key to its operation is that the infected program, when invoked, will first execute the virus code and then execute the original code of the program.

Virus Structure

- Lets assume that a virus code, V, is prepended to infected programs, and it is assumed that the entry point to the program, when invoked, is the first line of the program.
- A general depiction of virus structure is:

Virus Structure

A simple Virus

```
        program V :=

{goto main;
        1234567;

        subroutine infect-executable :=
                {loop:
                file := get-random-executable-file;
                if (first-line-of-file = 1234567)
                   then goto loop
                   else prepend V to file; }

        subroutine do-damage :=
                {whatever damage is to be done}

        subroutine trigger-pulled :=
                {return true if some condition holds}

main:   main-program :=
                {infect-executable;
                if trigger-pulled then do-damage;
                goto next;}
next:

}
```

Virus Structure

- When the program is invoked, control is immediately transferred to the main virus program.
- The virus program may first seek out uninfected executable files and infect them.

Virus Structure

- Next, the virus may execute its payload if the required trigger conditions, if any, are met.
- Finally, the virus transfers control to the original program.

Virus Structure

- Here, an infected version of a program is longer than the corresponding uninfected one.
- Compress the executable file so that both the infected and uninfected versions are of identical length

Virus Structure

Logic for a Compression Virus

```
    program CV :=

{goto main;
    01234567;

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

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

Virus Structure

- Assume that a program P1 is infected with the virus CV.
- When this program is invoked, control passes to its virus, which performs the following steps:

Virus Structure

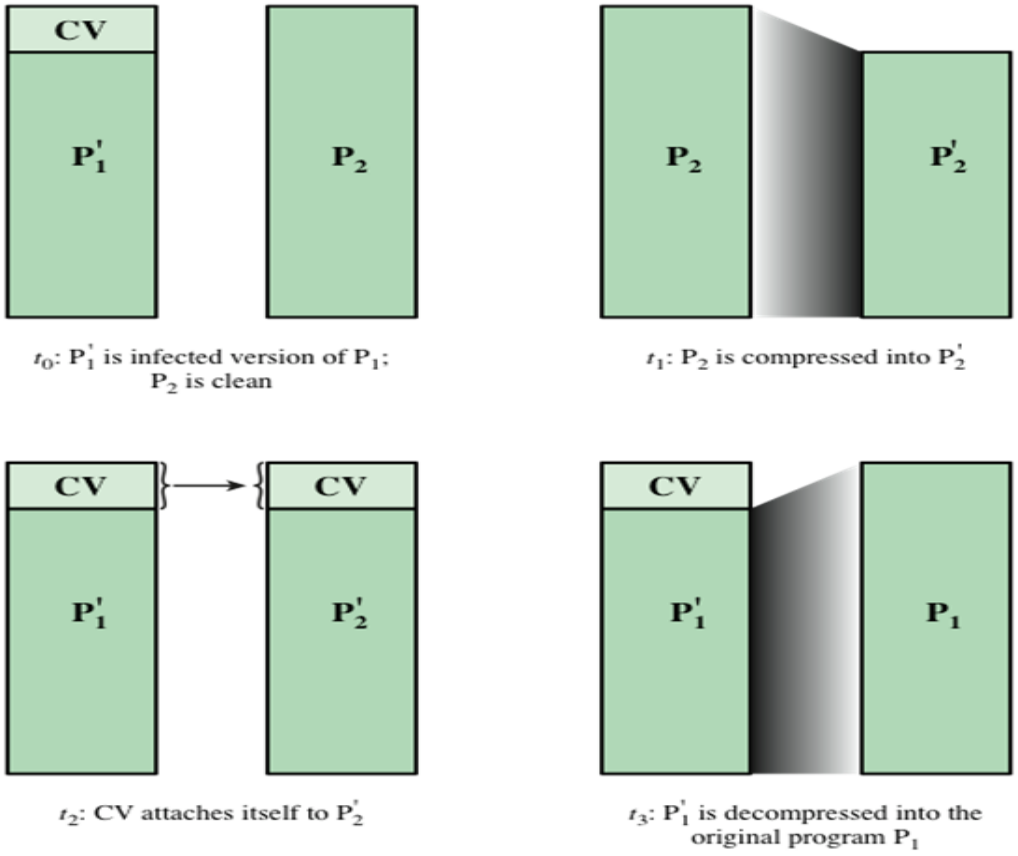


Figure 10.2 A Compression Virus

Viruses Classification

Network Security

Viruses Classification

Objectives of the Topic

- After completing this topic, a student will be able to
  - Describe classification of viruses.

Viruses Classification

Figures and material in this topic have been adapted from

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

Viruses Classification

- There has been a continuous arms race between virus writers and writers of antivirus software since viruses first appeared.

Viruses Classification

- As effective countermeasures are developed for existing types of viruses, newer types are developed.
- There is no simple or universally agreed-upon classification scheme for viruses.

Viruses Classification

- We classify viruses along two orthogonal axes:
- the type of target the virus tries to infect, and
- the method the virus uses to conceal itself from detection by users and antivirus software.

Viruses Classification

- A virus classification by target includes the following categories:
- **Boot sector infector:**
- Infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus.

Viruses Classification

- **File infector:**
- Infects files that the operating system or shell consider to be executable.
- **Macro virus:**
- Infects files with macro or scripting code that is interpreted by an application.

Viruses Classification

- **Multipartite virus:**
- Infects files in multiple ways.
- Typically, multipartite virus is capable of infecting multiple types of files, so that virus eradication must deal with all of the possible sites of infection.

Viruses Classification

- A virus classification by concealment strategy includes the following categories:
- **Encrypted virus:**
- A portion of the virus creates a random encryption key and encrypts the remainder of the virus.

Viruses Classification

- The key is stored with the virus.
- When an infected program is invoked, the virus uses the stored random key to decrypt the virus.
- When the virus replicates, a different random key is selected.

Viruses Classification

- Because the bulk of the virus is encrypted with a different key for each instance, there is no constant bit pattern to observe.
- **Stealth virus:** A form of virus explicitly designed to hide itself from detection by antivirus software.

Viruses Classification

- The entire virus, not just a payload, is hidden.
- It may use both code mutation (e.g. compression), and rootkit techniques to achieve this.

Viruses Classification

- **Polymorphic virus:**
- A virus that mutates with every infection, making detection by the “signature” of the virus impossible.
- **Metamorphic virus:**
- As with a polymorphic virus, it mutates with every infection.

Viruses Classification

- The difference is that a metamorphic virus rewrites itself completely at each iteration, increasing the difficulty of detection.
- Metamorphic viruses may change their behavior as well as their appearance.

Viruses Classification

- **Macro and scripting viruses:**
- Macro viruses infect scripting code used to support active content in a variety of user document types.
- Threatening for a number of reasons:
- 1. A macro virus is platform independent

Viruses Classification

- 2. Macro viruses infect documents, not executable portions of code
- 3. Macro viruses are easily spread, as the documents they exploit are shared in normal use

Viruses Classification

- 4. Because macro viruses infect user documents rather than system programs, traditional file system access controls are of limited use in preventing their spread.

End

Antivirus Approaches

Network Security

Antivirus Approaches

Objectives of the Topic

- After completing this topic, a student will be able to
  - Describe approaches for antiviruses.

Antivirus Approaches

Figures and material in this topic have been adapted from

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

Antivirus Approaches

- The ideal solution to the threat of viruses is prevention:
- Do not allow a virus to get into the system in the first place, or block the ability of a virus to modify any files containing executable code or macros.

Antivirus Approaches

- The next best approach is to be able to do the following:
- **Detection:**
- Once the infection has occurred, determine that it has occurred and locate the virus.

Antivirus Approaches

- **Identification:**
- Once detection has been achieved, identify the specific virus that has infected a program.

Antivirus Approaches

- **Removal:** Once the specific virus has been identified, remove all traces of the virus from the infected program and restore it to its original state.
- Remove virus from all infected systems so that the virus cannot spread further.

Antivirus Approaches

- Four generations of antivirus software can be identified:
- A **First-generation** scanner requires a virus signature to identify a virus.

Antivirus Approaches

- The virus may contain “wildcards” but has essentially the same structure and bit pattern in all copies.
- Such signature-specific scanners are limited to the detection of known viruses.

Antivirus Approaches

- A **second-generation** scanner uses heuristic rules to search for probable virus infection and does not rely on a specific signature.

Antivirus Approaches

- E.g. a scanner may look for the beginning of an encryption loop used in a polymorphic virus and discover the encryption key.
- Once the key is discovered, scanner can decrypt virus to identify it, then remove infection.

Antivirus Approaches

- Another second-generation approach is integrity checking.
- A checksum can be appended to each program. If a virus infects the program without changing the checksum, then an integrity check will catch the change.

Antivirus Approaches

- **Third-generation** programs are memory-resident programs that identify a virus by its actions rather than its structure in an infected program.

Antivirus Approaches

- Such programs have the advantage that it is not necessary to develop signatures and heuristics for a wide array of viruses.
- Identify the small set of actions that indicate an infection is being attempted and then to intervene.

Antivirus Approaches

- **Fourth-generation** products are packages consisting of a variety of antivirus techniques used in conjunction.
- These include scanning and activity trap components.

Antivirus Approaches

- In addition, such a package includes access control capability, which limits the ability of viruses to penetrate a system and then limits the ability of a virus to update files in order to pass on the infection.

Antivirus Approaches

- **Advanced Antivirus Techniques:**
- More sophisticated antivirus approaches and products continue to appear.

Antivirus Approaches

- **Generic decryption (GD):**
- technology enables the antivirus program to easily detect even the most complex polymorphic viruses while maintaining fast scanning speeds.
- GD scanner contains following elements:

Antivirus Approaches

- **CPU emulator:**
- A software-based virtual computer.
- Instructions in an executable file are interpreted by the emulator rather than executed on the underlying processor.

Antivirus Approaches

- The emulator includes software versions of all registers and other processor hardware, so that the underlying processor is unaffected by programs interpreted on the emulator.

Antivirus Approaches

- **Virus signature scanner:**
- A module that scans the target code looking for known virus signatures.
- **Emulation control module:**
- Controls the execution of the target code.

Antivirus Approaches

- At the start of each simulation, the emulator begins interpreting instructions in the target code, one at a time.

Antivirus Approaches

End

- If the code includes a decryption routine that decrypts and hence exposes the virus, that code is interpreted.
- In effect, the virus does the work for the antivirus program by exposing the virus.

Digital Immune System

Network Security

Digital Immune System

Objectives of the Topic

- After completing this topic, a student will be able to
  - Explain working of the Digital Immune System.

Digital Immune System

Figures and material in this topic have been adapted from

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

Digital Immune System

- Advances in virus and antivirus technology go hand in hand.
- As the virus arms race has evolved, both viruses and, necessarily, antivirus software have grown more complex and sophisticated.

Digital Immune System

- **Digital Immune System:**
- The digital immune system is a comprehensive approach to virus protection developed by IBM and subsequently refined by Symantec.

Digital Immune System

- Motivation for this development has been the rising threat of Internet-based virus propagation.
- We first say a few words about this threat and then summarize IBM's approach.

Digital Immune System

- Traditionally, the virus threat was characterized by the relatively slow spread of new viruses and new mutations.
- Antivirus software was typically updated on a monthly basis, and this was sufficient to control problem.

Digital Immune System

- Also traditionally, the Internet played a comparatively small role in spread of viruses.
- But two major trends in Internet technology had an increasing impact on rate of virus propagation in recent years:

Digital Immune System

- **Integrated mail systems:**
- Systems such as Lotus Notes and Microsoft Outlook make it very simple to send anything to anyone and to work with objects that are received.

Digital Immune System

- **Mobile-program systems:**
- Capabilities such as Java and ActiveX allow programs to move on their own from one system to another.

Digital Immune System

- IBM has developed a prototype digital immune system.
- This system expands on the use of program emulation and provides a general purpose emulation and virus-detection system.

Digital Immune System

- The objective of this system is to provide rapid response time so that viruses can be stamped out almost as soon as they are introduced.
- The typical steps in digital immune system operation:

Digital Immune System

- **1.** A monitoring program on each PC uses a variety of heuristics based on system behavior, suspicious changes to programs, or family signature to infer that a virus may be present.

Digital Immune System

- The monitoring program forwards a copy of any program thought to be infected to an administrative machine within the organization.

Digital Immune System

- **2.** The administrative machine encrypts the sample and sends it to a central virus analysis machine.

Digital Immune System

- **3.** This machine creates an environment in which the infected program can be safely run for analysis.

Digital Immune System

- Techniques used for this purpose include emulation, or the creation of a protected environment within which the suspect program can be executed and monitored.

Digital Immune System

- The virus analysis machine then produces a prescription for identifying and removing the virus.

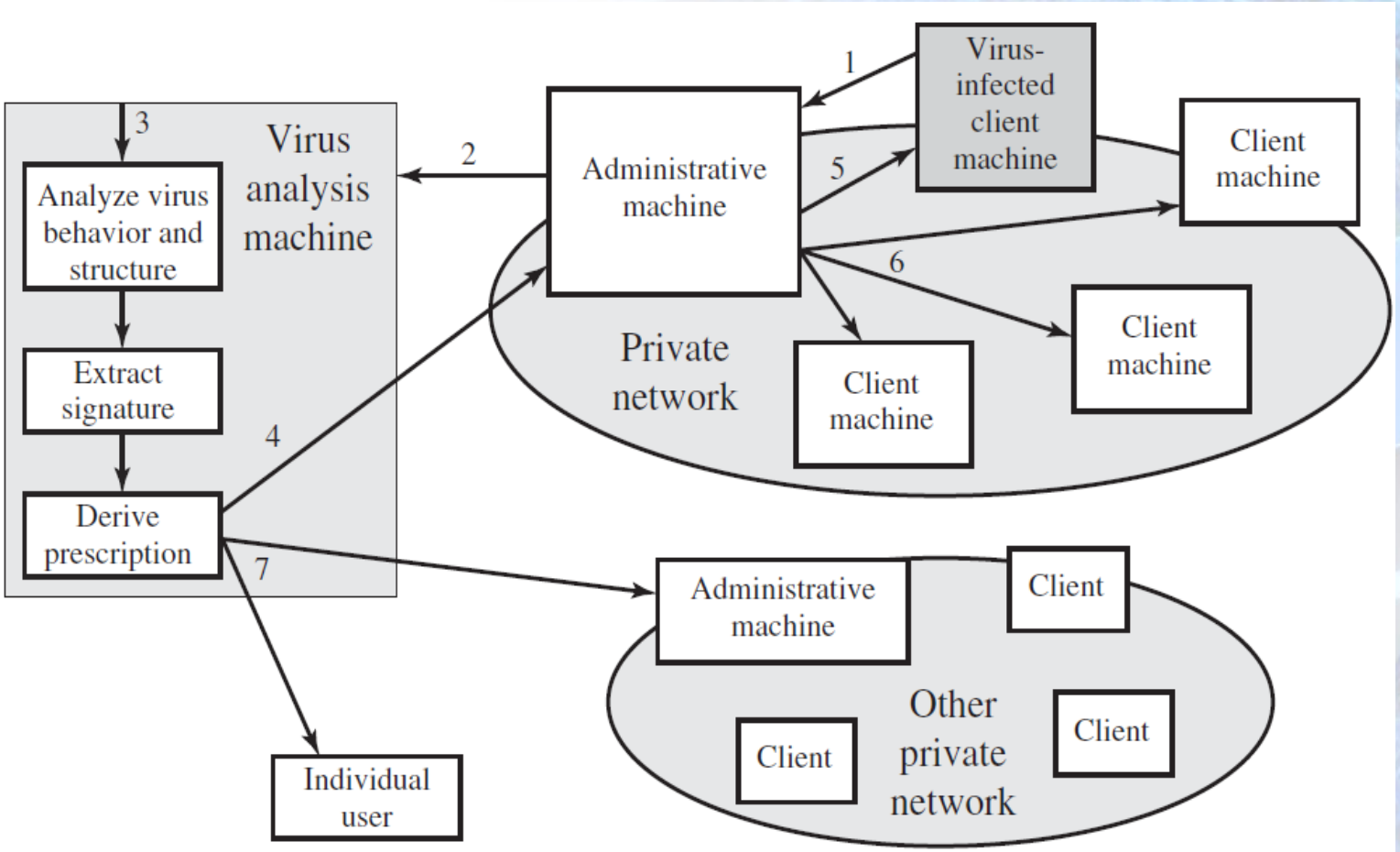
Digital Immune System

- **4.** The resulting prescription is sent back to the administrative machine.
- **5.** The administrative machine forwards the prescription to the infected client.

Digital Immune System

- **6.** The prescription is also forwarded to other clients in the organization.
- **7.** Subscribers around the world receive regular antivirus updates that protect them from the new virus.

Digital Immune System



Digital Immune System

End

- The success of the digital immune system depends on the ability of the virus analysis machine to detect new and innovative virus strains.
- continually update the digital immune software to keep up with the threat.

Behavior-Blocking Software

Network Security

Behavior-Blocking Software

Objectives of the Topic

- After completing this topic, a student will be able to
  - Explain operation of behavior-blocking software.

Behavior-Blocking Software

Figures and material in this topic have been adapted from

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

Behavior-Blocking Software

- Advances in virus and antivirus technology go hand in hand.
- As the virus arms race has evolved, both viruses and, necessarily, antivirus software have grown more complex and sophisticated.

Behavior-Blocking Software

- **Behavior-blocking Software:**
- Behavior-blocking software integrates with the operating system of a host computer and monitors program behavior in real-time for malicious actions.

Behavior-Blocking Software

- The behavior blocking software then blocks potentially malicious actions before they have a chance to affect the system.
- Monitored behaviors can include:

Behavior-Blocking Software

- Attempts to open, view, delete, and/or modify files;
- Attempts to format disk drives and other unrecoverable disk operations;
- Modifications to the logic of executable files or macros;

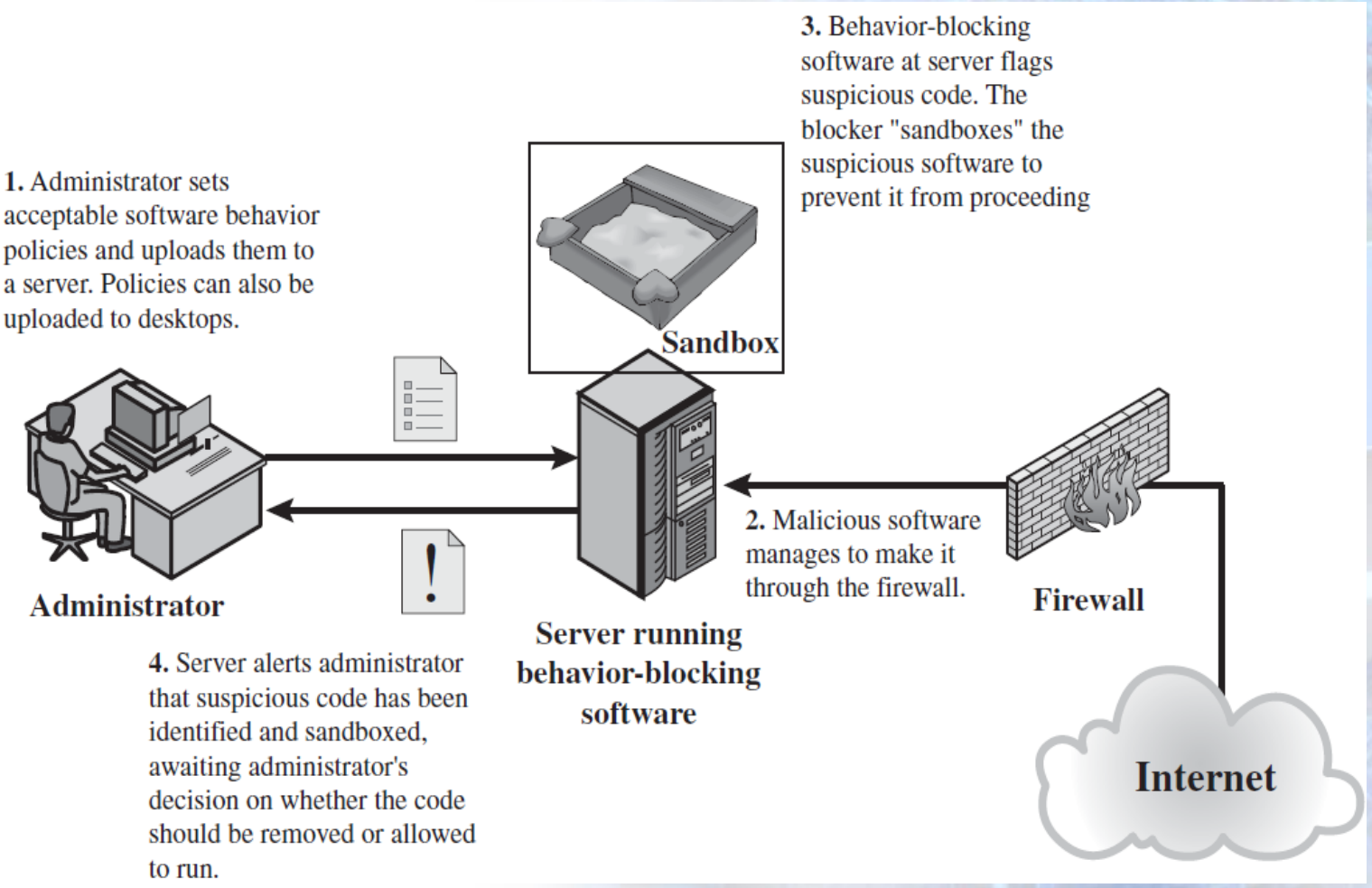
Behavior-Blocking Software

- Modification of critical system settings, such as start-up settings;
- Scripting of e-mail and instant messaging clients to send executable content; and
- Initiation of network communications.

Behavior-Blocking Software

- **Operation:**
- Next figure illustrates the operation of a behavior blocker.

Behavior-Blocking Software



Behavior-Blocking Software

- Behavior-blocking software runs on server and desktop computers.
- It is instructed through policies set by network administrator.

Behavior-Blocking Software

- This lets benign actions take place but to intercede when unauthorized or suspicious actions occur.
- The module blocks any suspicious software from executing.

Behavior-Blocking Software

- A blocker isolates the code in a sandbox, which restricts the code's access to various OS resources and applications.
- The blocker then sends an alert.

Behavior-Blocking Software

- Because a behavior blocker can block suspicious software in real-time, it has an advantage over such established antivirus detection techniques as fingerprinting or heuristics.

Behavior-Blocking Software

- There are literally trillions of different ways to obfuscate and rearrange the instructions of a virus or worm, many of which will evade detection by a fingerprint scanner or heuristic.

Behavior-Blocking Software

- Eventually malicious code makes a well-defined request to the operating system.
- A behavior blocker can intercept all such requests, it can identify and block malicious actions.

Behavior-Blocking Software

- Behavior blocking alone has **limitations**.
- Because the malicious code must run on the target machine before all its behaviors can be identified, it can cause harm before it has been detected and blocked.

Behavior-Blocking Software

- For example, a new virus might shuffle a number of seemingly unimportant files around the hard drive before infecting a single file and being blocked.

Behavior-Blocking Software

End

- Even though the actual infection was blocked, the user may be unable to locate his or her files, causing a loss to productivity or possibly worse.

Introduction to Worms

Network Security

Introduction to Worms

Objectives of the Topic

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

Introduction to Worms

Figures and material in this topic have been adapted from

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

Introduction to Worms

- A worm is a program that actively seeks out more machines to infect, and then each infected machine serves as an automated launching pad for attacks on other machines.

Introduction to Worms

- Worm programs exploit software vulnerabilities in client or server programs to gain access to each new system.
- They can use network connections to spread from system to system.

Introduction to Worms

- They can also spread through shared media, such as USB drives or CD and DVD data disks.

Introduction to Worms

- E-mail worms spread in macro or script code included in documents attached to e-mail or to instant messenger file transfers.
- Upon activation, the worm may replicate and propagate again.

Introduction to Worms

- Also, the worm usually carries some form of payload.
- To replicate itself, a worm uses some means to access remote systems.
- These include the following:

Introduction to Worms

- **Electronic mail or instant messenger facility:** A worm e-mails a copy of itself to other systems or sends itself as an attachment via an of instant message service, its code runs as e-mail or attachment is received or viewed.

Introduction to Worms

- **File sharing:** via removable media (e.g. a USB drive).
- A worm then executes when the drive is connected to another system using the autorun mechanism or when a user opens the infected file on the target system.

Introduction to Worms

- **Remote execution capability:** A worm executes a copy of itself on another system, either by using an explicit remote execution facility or by exploiting a program flaw in a network service to subvert its operations.

Introduction to Worms

- **Remote file access or transfer capability:** A worm uses a remote file access or transfer service to another system to copy itself from one system to the other, where users on that system may then execute it.

Introduction to Worms

- **Remote login capability:** A worm logs onto a remote system as a user and then uses commands to copy itself from one system to the other, where it then executes.

Introduction to Worms

Worm Phases:

- A worm typically uses the same phases as a computer virus: dormant, propagation, triggering, and execution.
- **Propagation phase** generally performs following functions:

Introduction to Worms

- 1. Search for appropriate access mechanisms to other systems to infect by examining host tables, address books, buddy lists, trusted peers, and other similar repositories of remote system access details.

Introduction to Worms

- 2. Use the access mechanisms found to transfer a copy of itself to the remote system and cause the copy to be run.

Introduction to Worms

- **Target discovery:**
- The first function in the propagation phase for a network worm is to search for other systems to infect, a process known as scanning or fingerprinting.
- **Worm network scanning strategies:**

Introduction to Worms

- **1. Random:** Each compromised host probes random addresses in the IP address space.
- Produces a high volume of Internet traffic, which causes generalized disruption even before the actual attack is lunched.

Introduction to Worms

- **2. Hit list:** The attacker first compiles a long list of potential vulnerable machines.
- Attacker begins infecting machines on the list.
- Each infected machine is provided with a portion of the list to scan.

Introduction to Worms

- This results in a very short scanning period, which may make it difficult to detect that infection is taking place.
- **3. Topological:** Uses information contained on an infected victim machine to find more hosts to scan.

Introduction to Worms

End

- **4. Local subnet:** If a host is infected behind a firewall, that host then looks for targets in its own local network.
- Uses subnet address structure to find other hosts that would otherwise be protected by firewall.

Worm Propagation Model

Network Security

Worm Propagation Model

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain worm propagation model, describe Morris worm, and state-of-art technology.

Worm Propagation Model

Figures and material in this topic have been adapted from

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

Worm Propagation Model

- A well-designed worm can spread rapidly and infect massive numbers of hosts.
- It is useful to have a general model for the rate of worm propagation.

Worm Propagation Model

- Computer viruses and worms exhibit similar self-replication and propagation behavior to biological viruses.
- We can look to classic epidemic models for understanding computer virus and worm propagation behavior.

Worm Propagation Model

- Propagation proceeds through three phases.
- In the **initial phase**, the number of hosts increases exponentially.
- Consider a simplified case in which a worm is launched from a single host and infects two nearby hosts.

Worm Propagation Model

- Each of these hosts infects two more hosts and so on.
- This results in exponential growth.
- After a time, infecting hosts waste some time attacking already-infected hosts, which reduces the rate of infection.

Worm Propagation Model

- During this **middle phase**, growth is approximately linear, but the rate of infection is rapid.

Worm Propagation Model

- When most vulnerable computers have been infected, the attack enters a **slow finish phase** as the worm seeks out those remaining hosts that are difficult to identify.

Worm Propagation Model

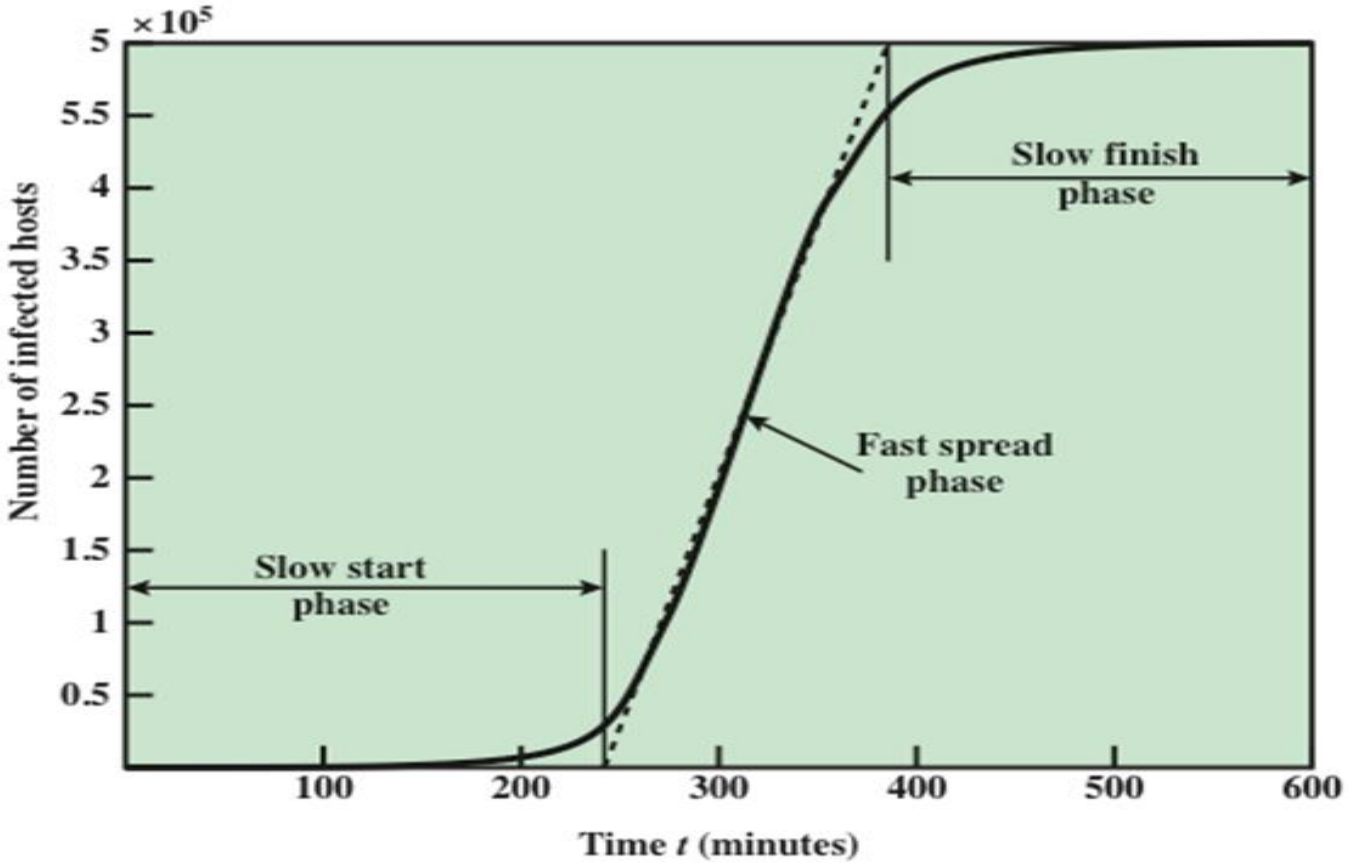


Figure 10.3 Worm Propagation Model

Worm Propagation Model

- Clearly, the objective in countering a worm is to catch the worm in its slow start phase, at a time when few hosts have been infected.

Worm Propagation Model

The Morris Worm:

- Released onto the Internet by Robert Morris in 1988.
- Was designed to spread on UNIX systems and used a number of different techniques for propagation.

Worm Propagation Model

- When a copy began execution its first task was to discover other hosts known to this host that would allow entry from this host.
- For each discovered host, the worm tried a number of methods for gaining access:

Worm Propagation Model

- 1. It attempted to log on to a remote host as a legitimate user.
- 2. It exploited a bug in the UNIX finger protocol, which reports the whereabouts of a remote user.

Worm Propagation Model

- 3. It exploited a trapdoor in the debug option of the remote process that receives and sends mail.

Worm Propagation Model

State of the art in Worm Technology:

- The technology includes the following:
- **1. Multiplatform:**
- Newer worms can attack a variety of platforms and are not limited to Windows.

Worm Propagation Model

- **2. Multi-exploit:**
- New worms penetrate systems in a variety of ways, using exploits against Web servers, browsers, e-mail, file sharing, and other network-based applications, or via shared media.

Worm Propagation Model

- **3. Ultrafast spreading:**
- Exploit various techniques to optimize the rate of spread of a worm to maximize its likelihood of locating as many vulnerable machines as possible in a short time period.

Worm Propagation Model

- **4. Polymorphic:**
- To evade detection, skip past filters, and foil real-time analysis, each copy of the worm has new code generated on the fly using functionally equivalent instructions and encryption techniques

Worm Propagation Model

- **5. Metamorphic:**
- In addition to changing their appearance, metamorphic worms have a repertoire of behavior patterns that are unleashed at different stages of propagation.

Worm Propagation Model

- **6. Transport vehicles:**
- Because worms can rapidly compromise a large number of systems, they are ideal for spreading a wide variety of malicious payloads.

Worm Propagation Model

End

- **7. Zero-day exploit:**
- To achieve maximum surprise and distribution, a worm should exploit an unknown vulnerability that is only discovered by the general network community when the worm is launched.

Proactive Worm Containment

Network Security

Proactive Worm Containment

Objectives of the Topic

- After completing this topic, a student will be able to
  - Describe malware countermeasures and working of Proactive Worm Containment method.

Proactive Worm Containment

Figures and material in this topic have been adapted from

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

Proactive Worm Containment

Countermeasures:

- We can list four main elements of prevention:
- policy,
- awareness,
- vulnerability mitigation, and
- threat mitigation.

Proactive Worm Containment

- One of the first countermeasures is to ensure all systems are as current as possible, with all patches applied, in order to reduce the number of vulnerabilities that might be exploited on the system.

Proactive Worm Containment

- The next is to set appropriate access controls on the applications and data stored on the system, to reduce the number of files that any user can access, and hence infect as a result of them executing some malware code.

Proactive Worm Containment

- Appropriate user awareness and training is another countermeasure.

Proactive Worm Containment

Malware countermeasure approaches:

- If prevention fails, then technical mechanisms can be used to support the following threat mitigation options:

Proactive Worm Containment

- **Detection:** Determine that infection has occurred.
- **Identification:** Identify the specific malware.
- **Removal:** remove all traces of malware virus from all infected systems so that it cannot spread further.

Proactive Worm Containment

- Let us now look at some **requirements** for effective malware countermeasures:
- **Generality:**
- The approach taken should be able to handle a wide variety of attacks.

Proactive Worm Containment

- **Timeliness:**
- The approach should respond quickly so as to limit the number of infected programs or systems and the consequent activity.

Proactive Worm Containment

- **Resiliency:**
- The approach should be resistant to evasion techniques employed by attackers to hide the presence of their malware.

Proactive Worm Containment

- **Minimal denial-of-service costs:**
- The approach should result in minimal reduction in service due to actions of countermeasure software, and should not significantly disrupt normal operation.

Proactive Worm Containment

- **Transparency:**
- The countermeasure software and devices should not require modification to existing (legacy) OSs, application software, and hardware.

Proactive Worm Containment

- **Global and local coverage:**
- The approach should be able to deal with attack sources both from outside and inside the enterprise network.

Proactive Worm Containment

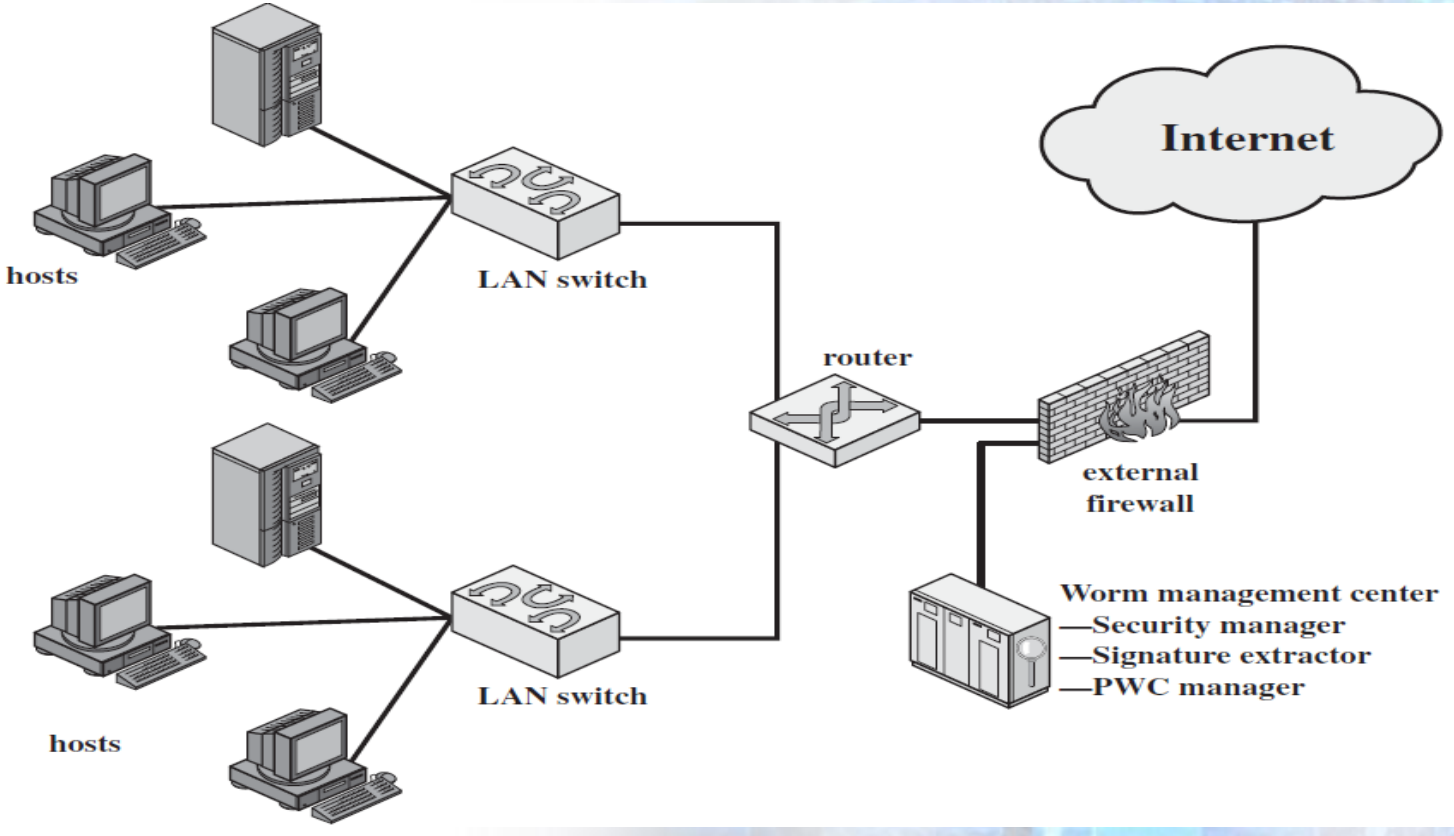
Proactive Worm Containment (PWC):

- is a host based approach.
- PWC is designed to address the threat of worms that spread rapidly.

Proactive Worm Containment

- The software on a host looks for surges in rate of frequency of outgoing connection attempts and diversity of connections to remote hosts.
- A deployed PWC system consists of a PWC manager and PWC agents in hosts.

Proactive Worm Containment



- security manager, signature extractor, and PWC manager can be implemented as in separate devices.

Proactive Worm Containment

- **The operation is :**
- **A.** A PWC agent monitors outgoing traffic for scan activity, determined by a surge in UDP or TCP connection attempts to remote hosts.
- If a surge is detected, the agent performs the following actions:

Proactive Worm Containment

- (1) issues an alert to local system; (2) blocks all outgoing connection attempts; (3) transmits the alert to the PWC manager; and (4) starts a relaxation analysis, described in D.

Proactive Worm Containment

- **B.** A PWC manager receives an alert. The PWC propagates the alert to all other agents (beside the originating agent).
- **C.** The host receives an alert. The agent must decide whether to ignore the alert, in the following way.

Proactive Worm Containment

- If the time since the last incoming packet has been sufficiently long so that the agent would have detected a worm if infected, then the alert is ignored.
- Otherwise, the agent performs the following actions:

Proactive Worm Containment

- (1) blocks all outgoing connection attempts from the specific alerting port; and (2) starts a relaxation analysis.
- **D.** Relaxation analysis is performed as follows:

Proactive Worm Containment

- An agent monitors outgoing activity for a fixed window of time to see if outgoing connections exceed a threshold.
- If so, blockage is continued, relaxation analysis is performed for another window of time.

Proactive Worm Containment

End

- This process continues until the outgoing connection rate drops below the threshold, at which time the agent removes the block.
- Otherwise, the agent isolates the host and reports to the PWC manager.

Network-Based Worm Defense

Network Security

Network-Based Worm Defense

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain working of network based worm defense.

Network-Based Worm Defense

Figures and material in this topic have been adapted from

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

Network-Based Worm Defense

- Detection of the presence of malware can occur in a number of locations.
- **host-based:** may occur on the infected system.
- may take place as part of the **perimeter security** mechanisms e.g. firewall, IDS

Network-Based Worm Defense

Perimeter Scanning Approaches:

- In this case, antivirus software is located on an organization's firewall and IDS.
- It is typically included in e-mail and Web proxy services running on these systems.

Network-Based Worm Defense

- The key element of a network-based worm defense is worm monitoring software.
- Assume an enterprise network at a site, consisting of one or an interconnected set of LANs.

Network-Based Worm Defense

- Two types of monitoring software may be used:
- **Ingress monitors:**
- Located at the border between the enterprise network and the Internet.

Network-Based Worm Defense

- They can be part of the ingress-filtering software of a border router or external firewall or a separate passive monitor.

Network-Based Worm Defense

- An example of a detection technique for an ingress monitor is to look for incoming traffic to unused local IP addresses.

Network-Based Worm Defense

- **Egress monitors:**
- These can be located at the egress point of individual LANs on the enterprise network as well as at the border between the enterprise network and the Internet.

Network-Based Worm Defense

- The egress monitor is designed to catch the source of a malware attack by monitoring outgoing traffic for signs of scanning or other suspicious behavior.

Network-Based Worm Defense

- **Worm monitors** can act in the manner of intrusion detection systems and generate alerts to a central administrative system.

Network-Based Worm Defense

- It is also possible to implement a system that attempts to react in real time to a worm attack, so as to counter zero-day exploits effectively.

Network-Based Worm Defense

An Example:

- An example of worm countermeasure architecture works as:
- **1.** Sensors deployed at various network locations detect a potential worm. The sensor logic can also be incorporated in IDS sensors.

Network-Based Worm Defense

- **2.** The sensors send alerts to a central server that correlates and analyzes the incoming alerts.
- The correlation server determines likelihood that a worm attack is being observed and the key characteristics of the attack.

Network-Based Worm Defense

- **3.** The server forwards its information to a protected environment, where the potential worm may be sandboxed for analysis and testing.

Network-Based Worm Defense

- **4.** The protected system tests the suspicious software against an appropriately instrumented version of the targeted application to identify the vulnerability.

Network-Based Worm Defense

- **5.** The protected system generates one or more software patches and tests these.

Network-Based Worm Defense

- **6.** If the patch is not susceptible to the infection and does not compromise the application's functionality, the system sends the patch to the application host to update the targeted application.

Network-Based Worm Defense

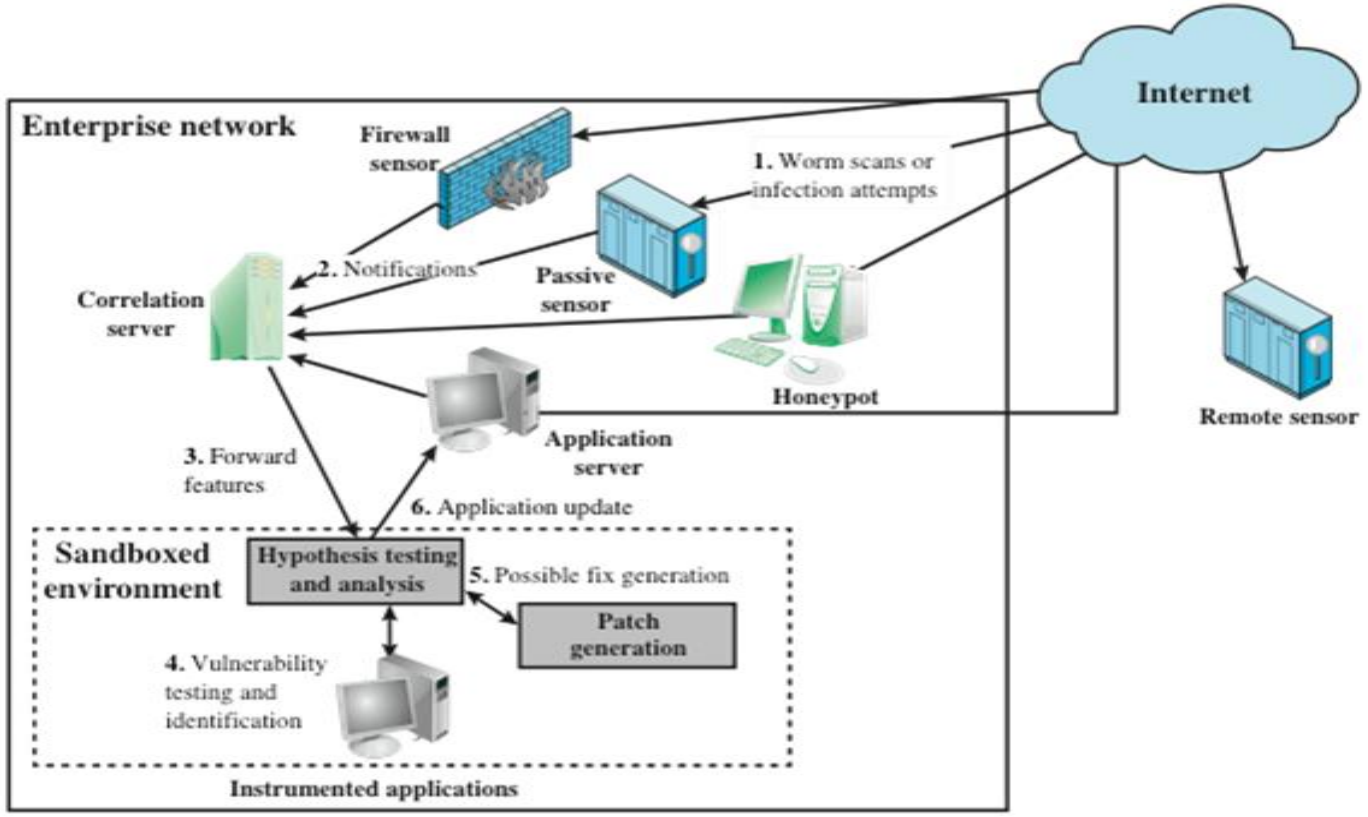


Figure 10.4 Placement of Worm Monitors

Network-Based Worm Defense

End

- The success of such an automated patching system depends on maintaining a current list of potential attacks and developing general tools for patching software to counter such attacks.

Distributed Denial of Service Attacks

Network Security

Distributed Denial of Service Attacks

Objectives of the Topic

- After completing this topic, a student will be able to
  - Describe attacks related to distributed denial of service.

Distributed Denial of Service Attacks

Figures and material in this topic have been adapted from

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

Distributed Denial of Service Attacks

- A denial-of-service (DoS) attack is an attempt to prevent legitimate users of a service from using that service.
- When this attack comes from a single host or network node, then it is referred to as a DoS attack.

Distributed Denial of Service Attacks

- A more serious threat is posed by a Distributed Denial of Service (DDoS) attack.

Distributed Denial of Service Attacks

- DDoS attacks make computer systems inaccessible by flooding servers, networks, or even end-user systems with useless traffic so that legitimate users can no longer gain access to those resources.

Distributed Denial of Service Attacks

- In a typical DDoS attack, a large number of compromised hosts are amassed to send useless packets.

Distributed Denial of Service Attacks

DDoS Attack Description:

- A DDoS attack attempts to consume the target's resources so it cannot provide its service.

Distributed Denial of Service Attacks

- One way to classify DDoS attacks is in terms of the type of resource that is consumed.

Distributed Denial of Service Attacks

- Broadly speaking, the resource consumed is either:
- an internal host resource on the target system, or
- data transmission capacity in the local network to which the target is attacked.

Distributed Denial of Service Attacks

- A simple example of an **internal resource attack** is the SYN flood attack.
- 1. The attacker takes control of multiple hosts over the Internet, instructing them to contact the target Web server.

Distributed Denial of Service Attacks

- 2. The slave hosts begin sending TCP/IP SYN (synchronize/initialization) packets, with erroneous return IP address information, to the target.

Distributed Denial of Service Attacks

- 3. Each SYN packet is a request to open a TCP connection.
- For each such packet, the Web server responds with a SYN/ACK packet, trying to establish a TCP connection with a TCP entity at a spurious IP address.

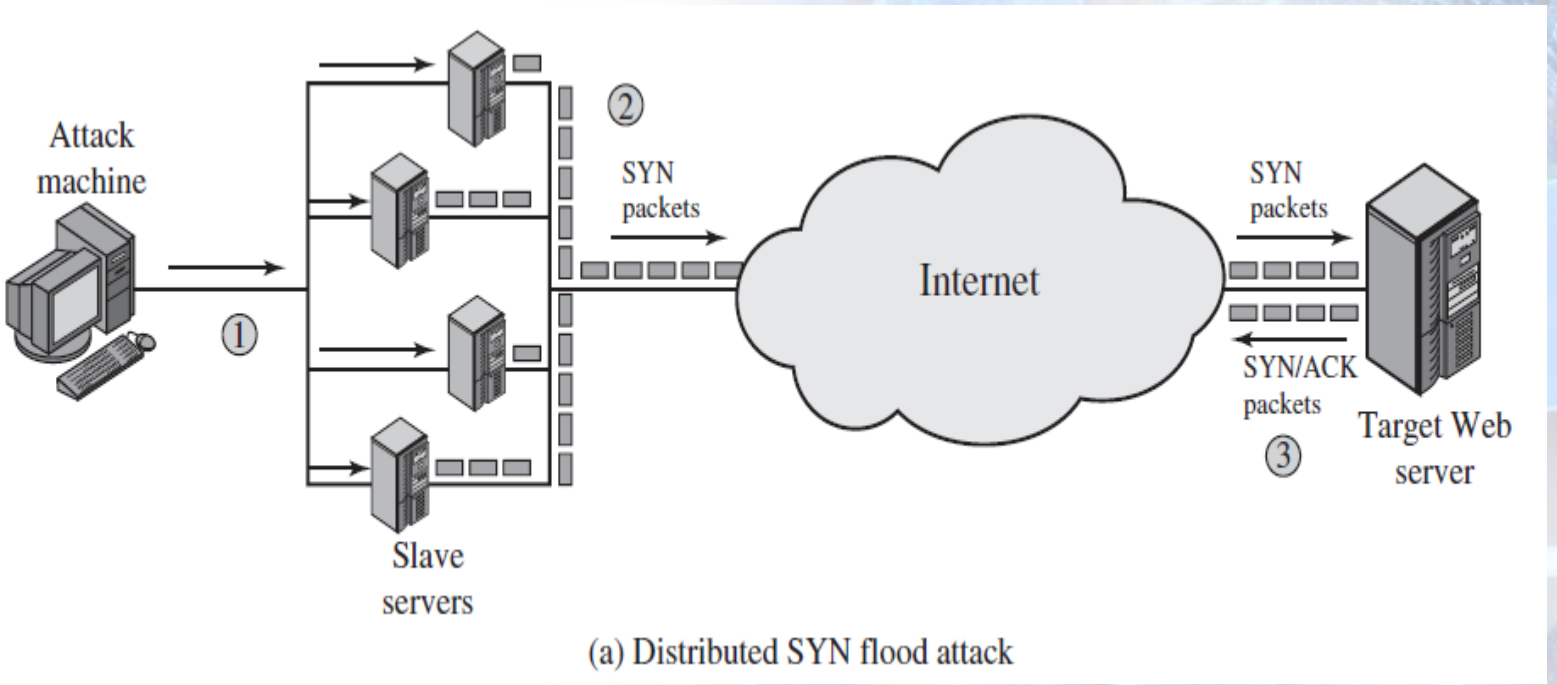
Distributed Denial of Service Attacks

- The Web server maintains a data structure for each SYN request waiting for a response back and becomes bogged down as more traffic floods in.

Distributed Denial of Service Attacks

- The result is that legitimate connections are denied while the victim machine is waiting to complete bogus “half-open” connections.

Distributed Denial of Service Attacks



(a) Distributed SYN flood attack

Distributed Denial of Service Attacks

- Our next example illustrates an attack that consumes data transmission resources.
- The following steps are involved:

Distributed Denial of Service Attacks

- 1. The attacker takes control of multiple hosts over the Internet, instructing them to send ICMP ECHO packets with the target's spoofed IP address to a group of hosts that act as reflectors.

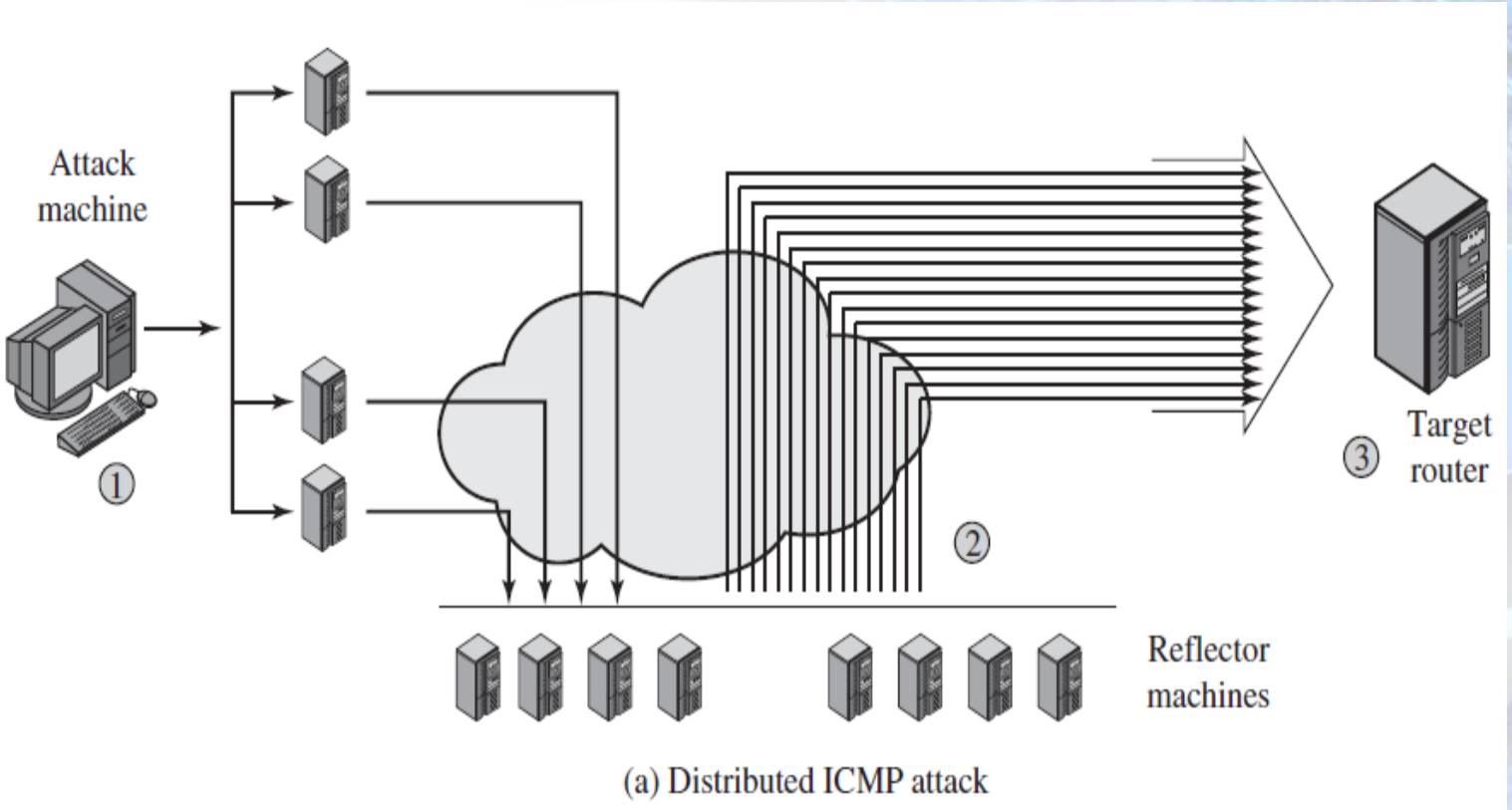
Distributed Denial of Service Attacks

- 2. Nodes at the bounce site receive multiple spoofed requests and respond by sending echo reply packets to the target site.

Distributed Denial of Service Attacks

- 3. The target's router is flooded with packets from the bounce site, leaving no data transmission capacity for legitimate traffic.

Distributed Denial of Service Attacks



(a) Distributed ICMP attack

Types of Flooding-Based DDoS Attacks

Network Security

Types of Flooding-Based DDoS Attacks

Objectives of the Topic

- After completing this topic, a student will be able to
  - explain types of flooding based DDoS attacks..

Types of Flooding-Based DDoS Attacks

Figures and material in this topic have been adapted from

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

Types of Flooding-Based DDoS Attacks

- One way to classify Distributed denial of service (DDoS) attacks is as:
- Direct DDoS attacks and
- reflector DDoS attacks.

Types of Flooding-Based DDoS Attacks

- In a direct DDoS attack, the attacker is able to implant zombie software (Program activated on an infected machine to launch attacks on other machines) on a number of sites distributed throughout Internet.

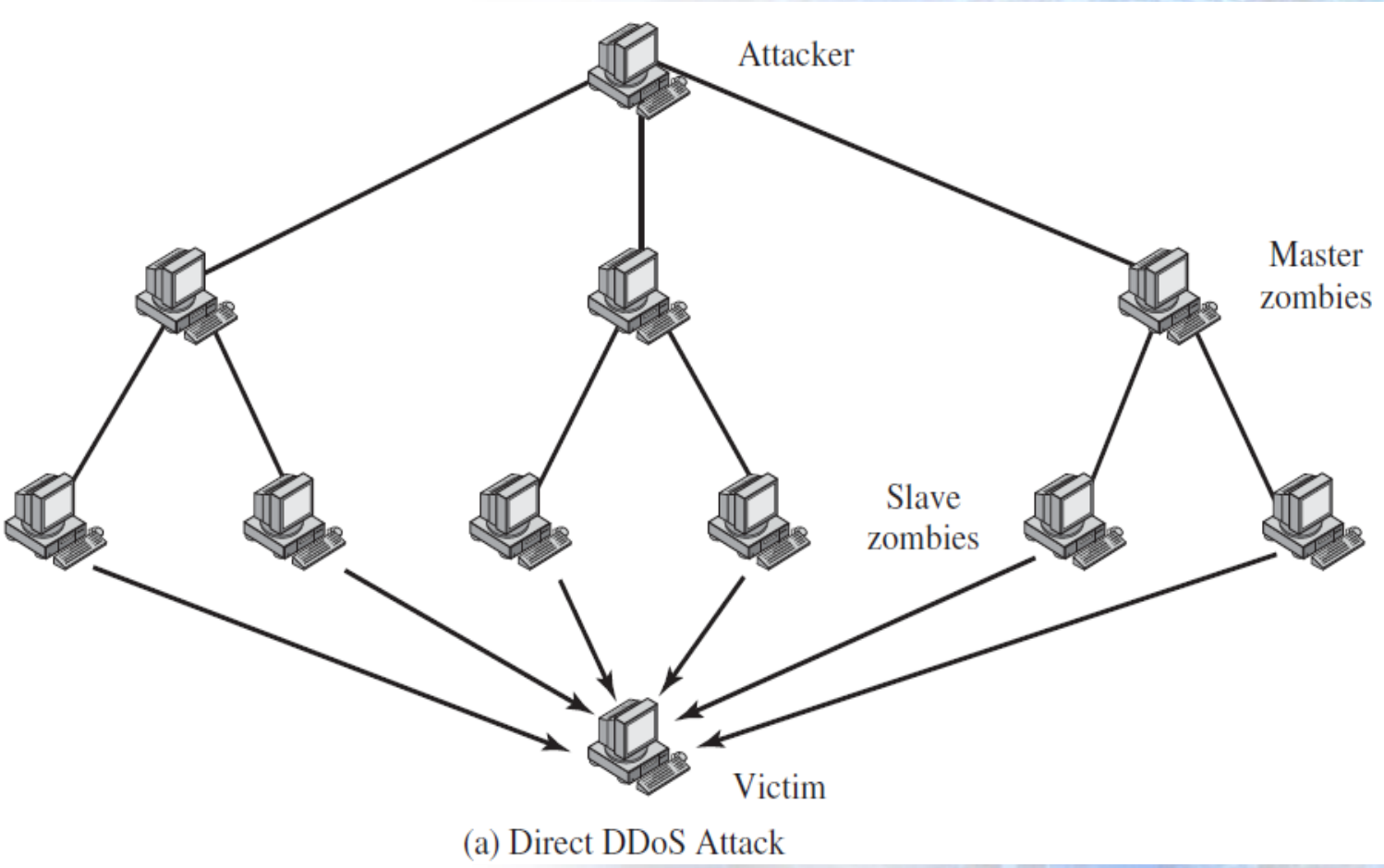
Types of Flooding-Based DDoS Attacks

- The DDoS attack involves two levels of zombie machines:
- master zombies and slave zombies.
- The hosts of both machines have been infected with malicious code.

Types of Flooding-Based DDoS Attacks

- The attacker coordinates and triggers master zombies, which in turn coordinate and trigger slave zombies.
- The use of two levels of zombies makes it more difficult to trace the attack back to its source.

Types of Flooding-Based DDoS Attacks



(a) Direct DDoS Attack

Types of Flooding-Based DDoS Attacks

- A reflector DDoS attack adds another layer of machines.
- In this type of attack, the slave zombies construct packets requiring a response that contain the target's IP address as the source IP address in packet's IP header.

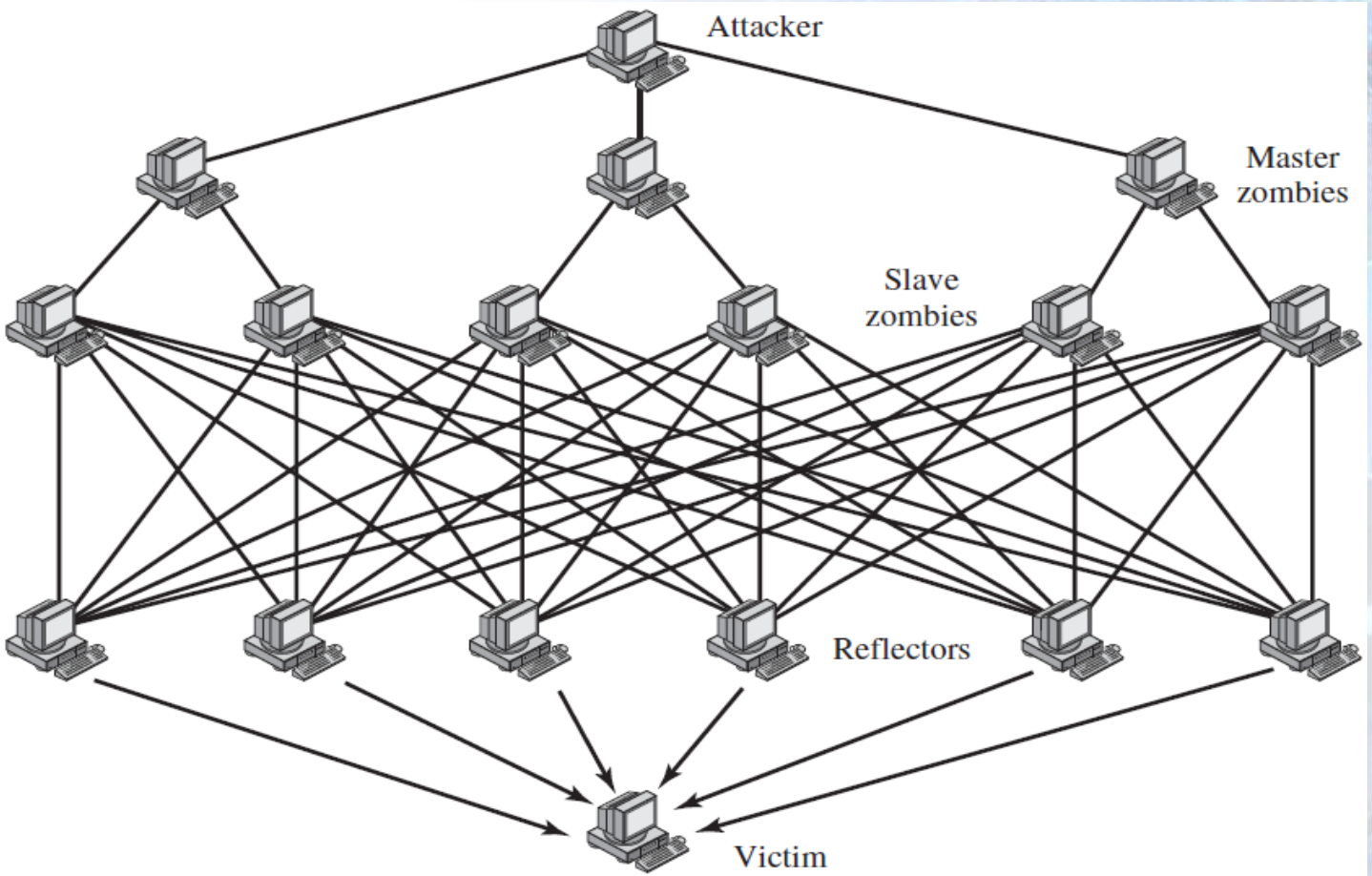
Types of Flooding-Based DDoS Attacks

- These packets are sent to uninfected machines known as reflectors.
- The uninfected machines respond with packets directed at the target machine.

Types of Flooding-Based DDoS Attacks

- A reflector DDoS attack can easily involve more machines and more traffic than a direct DDoS attack and is more damaging.
- Also, tracing back the attack or filtering out the attack packets is more difficult.

Types of Flooding-Based DDoS Attacks



(b) Reflector DDoS Attack

Types of Flooding-Based DDoS Attacks

Constructing the Attack Network:

- The first step in a DDoS attack is for the attacker to infect a number of machines with zombie software that will ultimately be used to carry out the attack.

Types of Flooding-Based DDoS Attacks

- **Essential ingredients in this phase:**
- 1. Software that can carry out the DDoS attack.
- 2. A vulnerability in a large number of systems.
- 3. A strategy for locating vulnerable machines (scanning)

Types of Flooding-Based DDoS Attacks

- **Scanning strategies:**
- **Random:** Each compromised host probes random addresses in the IP address space, using a different seed.
- **Hit list:** The attacker first compiles a long list of potential vulnerable machines

Types of Flooding-Based DDoS Attacks

- **Topological:** This method uses information contained on an infected victim machine to find more hosts to scan.
- **Local subnet:** If a host is infected behind a firewall, that host looks for targets in its own local network.

Types of Flooding-Based DDoS Attacks

DDoS Countermeasures:

- In general, there are three lines of defense against DDoS attacks:

Types of Flooding-Based DDoS Attacks

- **Attack prevention and preemption (before the attack):**
- These mechanisms enable the victim to endure attack attempts without denying service to legitimate clients.

Types of Flooding-Based DDoS Attacks

- **Attack detection and filtering (during the attack):**
- These mechanisms attempt to detect the attack as it begins and respond immediately.

Types of Flooding-Based DDoS Attacks

End

- **Attack source traceback and identification (during and after the attack):**
- This is an attempt to identify the source of the attack as a first step in preventing future attacks.

Recent Forms of Malware Attacks

Network Security

Recent Forms of Malware Attacks

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe Recent Malware Attacks.

Recent Forms of Malware Attacks

Figures and material in this topic have been adapted from

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

Recent Forms of Malware Attacks

Mobile code:

- Refers to programs (e.g., script, macro, or other portable instruction) that is transmitted from a remote system to a local system and then executed on local system without user's explicit instruction.

Recent Forms of Malware Attacks

- Mobile code often acts as a mechanism for a virus, worm, or Trojan horse.
- Popular vehicles for mobile code include Java applets, ActiveX, JavaScript, and VBScript.

Recent Forms of Malware Attacks

Drive-by-downloads:

- Exploit bugs in user applications to install malware.

Recent Forms of Malware Attacks

- When a user views a Web page controlled by the attacker, it contains code that exploits the browser bug to download and install malware on the system without the user's knowledge or consent.

Recent Forms of Malware Attacks

- Drive-by-download does not actively propagate as a worm does, but rather waits for unsuspecting users to visit the malicious Web page in order to spread to their systems.

Recent Forms of Malware Attacks

Spam:

- Unsolicited bulk e-mail.
- Imposes significant costs on both the network infrastructure needed to relay this traffic and on users who need to filter their legitimate e-mails.

Recent Forms of Malware Attacks

- Most recent spam is sent by botnets using compromised user systems.
- Is a significant carrier of malware.
- May be used in a phishing attack.

Recent Forms of Malware Attacks

- Although a significant security concern, in many cases it requires the user's active choice to view the e-mail and any attached document or to permit the installation of some program, in order for the compromise to occur.

Recent Forms of Malware Attacks

Trojan horses:

- Is a useful, or apparently useful, program or utility containing hidden code that, when invoked, performs some unwanted or harmful function.

Recent Forms of Malware Attacks

Payload –
System Corruption:

- Once malware is active on the target system, the next concern is what actions it will take on this system i.e. what payload does it carry.

Recent Forms of Malware Attacks

- **Examples:**
- Data destruction on the infected system
- Display unwanted messages or content on the user's system
- Encrypt the user's data
- Attempt to rewrite the BIOS code

Recent Forms of Malware Attacks

Payload –
Attack Agent:

- Malware subverts the computational and network resources of the infected system for use by the attacker.

Recent Forms of Malware Attacks

- Such a system is known as a Bot (robot), zombie or drone and secretly takes over another Internet-attached computer and then uses that computer to launch or manage attacks.

Recent Forms of Malware Attacks

- A *botnet* is a collection of bots often capable of acting in a coordinated manner.
- **Uses of bots:**
- Distributed denial-of-service (DDoS) attacks.

Recent Forms of Malware Attacks

Remote Control Facility:

- is what distinguishes a bot from a worm.
- A worm propagates itself and activates itself, whereas a bot is controlled from some central facility, at least initially.

Recent Forms of Malware Attacks

- Typical means of implementing is on an IRC server.
- Once a communications path is established between a control module and the bots, the control module can activate the bots.

Recent Forms of Malware Attacks

Payload – Information Theft:

- **Keylogger:** Captures keystrokes on the infected machine to allow an attacker to monitor user login and password credentials.

Recent Forms of Malware Attacks

- **Spyware:** Subvert the compromised machine to allow monitoring of a wide range of activity on the system which can result in significantly compromising the user's personal information

Recent Forms of Malware Attacks

- **Spear-Phishing:** An e-mail claiming to be from a trusted source, however, the recipients are carefully researched by the attacker, and each e-mail is carefully crafted to suit its recipient specifically.

Recent Forms of Malware Attacks

Payload – stealthing:

- Backdoor (trapdoor) is a secret entry point into a program that allows someone who is aware of backdoor to gain access without going through usual security access procedures.

Recent Forms of Malware Attacks

End

Rootkit:

- is a set of programs installed on a system to maintain covert access to that system with administrator (or root) privileges, while hiding evidence of its presence to greatest extent possible.

Intruders

Network Security

Intruders

Objectives of the Topic

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

Intruders

Figures and material in this topic have been adapted from

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

Intruders

- The two most publicized threats to security are:
- the intruders, often referred to as a hackers or crackers.
- viruses.

Intruders

- We can identify three classes of intruders:
- **Masquerader:**
- An individual who is not authorized to use the computer and who penetrates a system's access controls to exploit a legitimate user's account.

Intruders

- **Misfeasor:**
- A legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuses his or her privileges.

Intruders

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

Intruders

- The masquerader is likely to be an outsider, the misfeasor generally is an insider, and the clandestine user can be either an outsider or an insider.

Intruders

- Intruder attacks range from the benign to the serious.
- At the benign end of the scale, there are many people who simply wish to explore internets and see what is out there.

Intruders

- At the serious end are individuals who are attempting to read privileged data, perform unauthorized modifications to data, or disrupt the system.

Intruders

- **Examples of Intrusion**
- We list the following examples of intrusion:
- Performing a remote root compromise of an e-mail server.
- Defacing a Web server
- Guessing and cracking passwords

Intruders

- Viewing sensitive data, including payroll records and medical information, without authorization
- Running a packet sniffer on a workstation to capture usernames and passwords

Intruders

- Copying a database containing credit card numbers
- Using a permission error on an anonymous FTP server to distribute pirated software and music files

Intruders

- Dialing into an unsecured modem and gaining internal network access
- Posing as an executive, calling the help desk, resetting the executive's e-mail password, and learning the new password

Intruders

Hackers:

- Traditionally, those who hack into computers do so for the thrill of it or for status.

Intruders

- Intrusion detection systems (IDSs) and intrusion prevention systems (IPSs) are designed to counter hacker threats.

Intruders

- In addition to using such systems, organizations can consider restricting remote logons to specific IP addresses and/or use virtual private network technology

Intruders

- **Establishment of Computer Emergency Response Teams (CERTs)**
- These cooperative ventures collect information about system vulnerabilities and disseminate it to systems managers

Intruders

- Hackers also routinely read CERT reports
- It is important for system administrators to quickly insert all software patches to discovered vulnerabilities

Intruders

Criminal hackers:

- Organized groups of hackers.
- Usually have specific targets, or at least classes of targets in mind.

Intruders

- Once a site is penetrated, the attacker acts quickly, scooping up as much valuable information as possible and exiting.
- IDSs and IPSs may be less effective because of the quick in-and-out nature of attack.

Intruders

Insider Attacks:

- Among the most difficult to detect and prevent
- Can be motivated by revenge or simply a feeling of entitlement

Intruders

Countermeasures for Insider Attacks:

- Enforce least privilege, only allowing access to the resources employees need to do their job.
- Set logs to see what users access and what commands they are entering.
- Protect sensitive resources with strong authentication.
- Upon termination, delete employee's computer and network access.
- Upon termination, make a mirror image of employee's hard drive before reissuing it. That evidence might be needed if your company information turns up at a competitor.

Intrusion Techniques

Network Security

Intrusion Techniques

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe various intrusion techniques.

Intrusion Techniques

Figures and material in this topic have been adapted from

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

Intrusion Techniques

- The objective of the intruder is to gain access to a system or to increase the range of privileges accessible on a system.

Intrusion Techniques

- Most initial attacks use system or software vulnerabilities that allow a user to execute code that opens a backdoor into the system.

Intrusion Techniques

- Alternatively, the intruder attempts to acquire information that should have been protected.

Intrusion Techniques

- In some cases, this information is in the form of a user password.
- With knowledge of some other user's password, an intruder can log into a system and exercise all the privileges accorded to the legitimate user.

Intrusion Techniques

- Typically, a system must maintain a file that associates a password with each authorized user.
- If such a file is stored with no protection, then it is an easy matter to gain access to it and learn passwords.

Intrusion Techniques

- The password file can be protected in one of two ways:
- **1. One-way function:**
- The system stores only the value of a function based on the user's password.

Intrusion Techniques

- When the user presents a password, the system transforms that password and compares it with the stored value.

Intrusion Techniques

- In practice, the system usually performs a one-way transformation (not reversible), in which the password is used to generate a key for the one-way function and in which a fixed length output is produced.

Intrusion Techniques

- **2. Access control:**
- Access to the password file is limited to one or a very few accounts.
- If one or both of these countermeasures are in place, some effort is needed for a potential intruder to learn passwords.

Intrusion Techniques

Password Guessing:

- 1. Try default passwords used with standard accounts that are shipped with the system.
- Many administrators do not bother to change these defaults.

Intrusion Techniques

- 2. Exhaustively try all short passwords (those of one to three characters).

Intrusion Techniques

- 3. Try words in the system's online dictionary or a list of likely passwords.
- Examples of the latter are readily available on hacker bulletin boards.

Intrusion Techniques

- 4. Collect information about users, such as their full names, the names of their spouse and children, pictures in their office, and books in their office that are related to hobbies.

Intrusion Techniques

- 5. Try users' phone numbers, Social Security numbers, and room numbers.
- 6. Try all legitimate license plate numbers for this state.
- 7. Use a Trojan horse to bypass restrictions on access.

Intrusion Techniques

- 8. Tap the line between a remote user and the host system.

Intrusion Techniques

- The first six methods are various ways of guessing a password.
- If an intruder has to verify the guess by attempting to log in, it is a tedious and easily countered means of attack.

Intrusion Techniques

- For example, a system can simply reject any login after three password attempts, thus requiring the intruder to reconnect to the host to try again.

Intrusion Techniques

- The seventh method of attack listed earlier, the Trojan horse, can be particularly difficult to counter.
- An **example** of a program that bypasses access controls :

Intrusion Techniques

- A low-privilege user produced a game program and invited the system operator to use it in his or her spare time.
- The program did indeed play a game, but in background it also contained code to copy password file.

Intrusion Techniques

- The eighth attack listed, line tapping, is a matter of physical security.

Intrusion Techniques

End

- Other intrusion techniques do not require learning a password.
- Intruders can get access to a system by exploiting attacks such as buffer overflows on a program that runs with certain privileges.

Basics of Intrusion Detection

Network Security

Basics of Intrusion Detection

Objectives of the Topic

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

Basics of Intrusion Detection

Figures and material in this topic have been adapted from

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

Basics of Intrusion Detection

- A system's second line of defense is intrusion detection.
- It is based on the assumption that the behavior of the intruder differs from that of a legitimate user in ways that can be quantified.

Basics of Intrusion Detection

- We cannot expect that there will be a crisp, exact distinction between an attack by an intruder and the normal use of resources by an authorized user.
- We must expect that there will be some overlap.

Basics of Intrusion Detection

- **Considerations:** Interest in IDS research has been motivated by following considerations:
- 1. An effective intrusion detection system can serve as a deterrent, so acting to prevent intrusions.

Basics of Intrusion Detection

- 2. If an intrusion is detected quickly enough, the intruder can be identified and ejected from the system before any damage is done or any data are compromised.

Basics of Intrusion Detection

- 3. Intrusion detection enables the collection of information about intrusion techniques that can be used to strengthen the intrusion prevention facility.

Basics of Intrusion Detection

- The figure on the next slide suggests, in very abstract terms, the nature of the task confronting the designer of an intrusion detection system.

Basics of Intrusion Detection

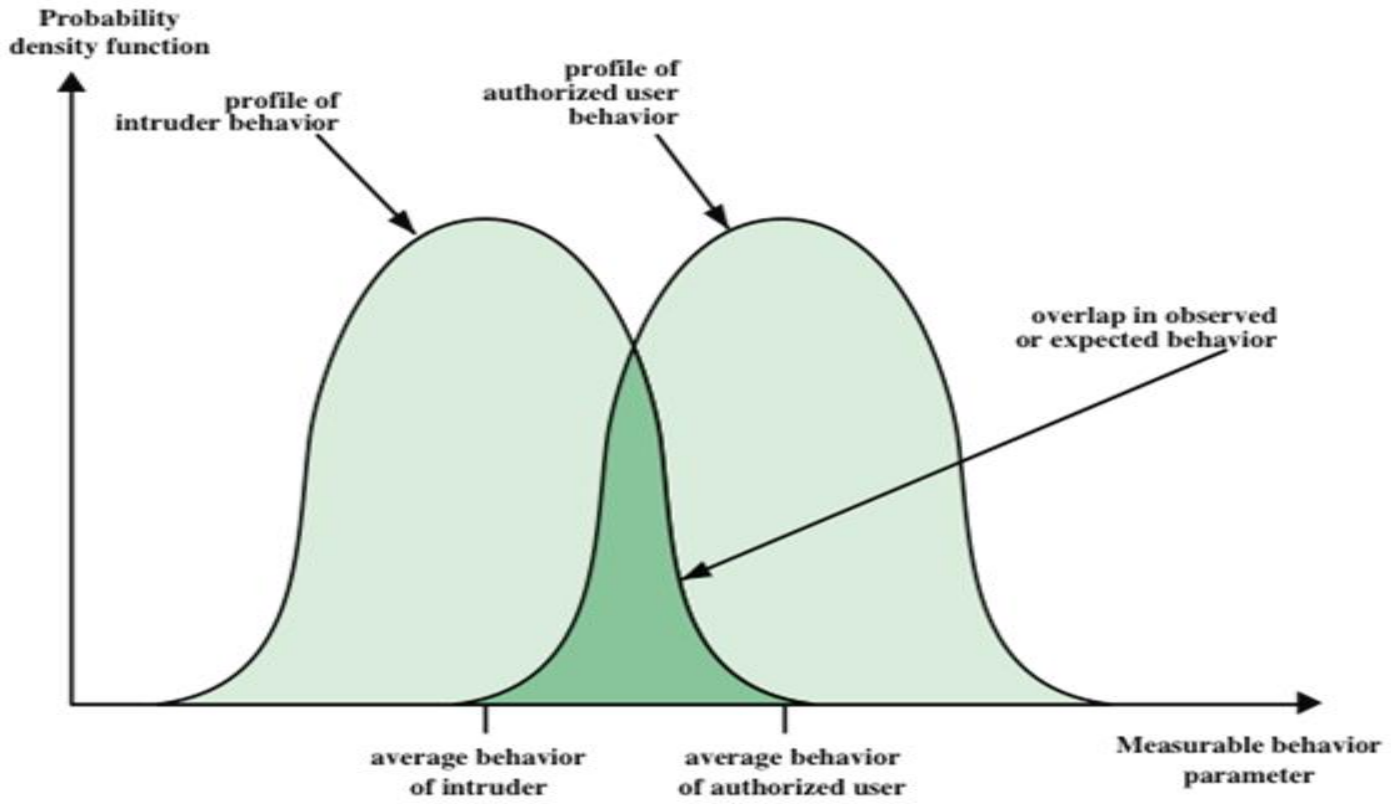


Figure 11.1 Profiles of Behavior of Intruders and Authorized Users

Basics of Intrusion Detection

- A loose interpretation of intruder behavior, which will catch more intruders, will also lead to a number of authorized users identified as intruders (false positives).

Basics of Intrusion Detection

- An attempt to limit false positives by a tight interpretation of intruder behavior will lead to an increase in intruders not identified as intruders (false negatives).

Basics of Intrusion Detection

Approaches to Intrusion Detection:

- Following approaches to intrusion detection can be identified:

Basics of Intrusion Detection

- **1. Statistical anomaly detection:**
- Involves the collection of data relating to the behavior of legitimate users over a period of time.

Basics of Intrusion Detection

- Then, statistical tests are applied to observed behavior to determine with a high level of confidence whether that behavior is not legitimate user behavior.

Basics of Intrusion Detection

- **a. Threshold detection:**
- This approach involves defining thresholds, independent of user, for the frequency of occurrence of various events.

Basics of Intrusion Detection

- **b. Profile based:**
- A profile of the activity of each user is developed and used to detect changes in the behavior of individual accounts.

Basics of Intrusion Detection

- **2. Rule-based detection:**
- Involves an attempt to define a set of rules or attack patterns that can be used to decide that a given behavior is that of an intruder.
- Often referred to as signature detection

Basics of Intrusion Detection

- In terms of the types of attackers, statistical anomaly detection is effective against masqueraders.
- For misfeasors attacks, rule-based approaches may be able to recognize the events and sequences.

Basics of Intrusion Detection

- In practice, a system may exhibit a combination of both approaches to be effective against a broad range of attacks.

Basics of Intrusion Detection

Audit Records:

- Fundamental tool for intrusion detection.
- Two plans are used.
- **Native audit records:** Virtually all multiuser operating systems include accounting software that collects information on user activity.

Basics of Intrusion Detection

- **Advantage:** no additional collection software is needed.
- **Disadvantage:** native audit records may not contain the needed information or may not contain it in a convenient form.

Basics of Intrusion Detection

- **Detection-specific audit records:**
- A collection facility can be implemented that generates audit records containing only that information required by the intrusion detection system.

Basics of Intrusion Detection

End

- One **advantage** of such an approach is that it could be made vendor independent and ported to a variety of systems.
- The **disadvantage** is the extra overhead involved in having two accounting packages running on a machine.

Statistical Anomaly Detection

Network Security

Statistical Anomaly Detection

Objectives of the Topic

- After completing this topic, a student will be able to
  - Explain working of statistical anomaly detection.

Statistical Anomaly Detection

Figures and material in this topic have been adapted from

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

Statistical Anomaly Detection

- Statistical anomaly detection involves the collection of data relating to the behavior of legitimate users over a period of time.

Statistical Anomaly Detection

- Then, statistical tests are applied to observed behavior to determine with a high level of confidence whether that behavior is not legitimate user behavior.

Statistical Anomaly Detection

- Statistical anomaly detection techniques fall into two broad categories:
- threshold detection, and
- profile-based systems.

Statistical Anomaly Detection

- **Threshold detection:**
- Involves counting the number of occurrences of a specific event type over an interval of time.

Statistical Anomaly Detection

- If the count surpasses what is considered a reasonable number that one might expect to occur, then intrusion is assumed.

Statistical Anomaly Detection

- Threshold analysis, by itself, is a crude and ineffective detector of even moderately sophisticated attacks.
- Both the threshold and the time interval must be determined.

Statistical Anomaly Detection

- Because of variability across users, such thresholds are likely to generate either a lot of false positives or false negatives.
- Threshold detectors may be useful in conjunction with more sophisticated techniques.

Statistical Anomaly Detection

- **Profile-based anomaly detection:** focuses on characterizing past behavior of individual users or related groups of users and then detecting significant deviations.

Statistical Anomaly Detection

- A profile may consist of a set of parameters, so that deviation on just a single parameter may not be sufficient in itself to signal an alert.
- The foundation of this approach is an analysis of audit records.

Statistical Anomaly Detection

- Examples of metrics useful for profile-based intrusion detection are:
- **Counter:** A count of certain event types is kept over a particular period of time.
- **Gauge:** is used to measure the current value of some entity.

Statistical Anomaly Detection

- **Interval timer:** Length of time between two related events.
- **Resource utilization:** Quantity of resources consumed during a specified period.

Statistical Anomaly Detection

- Given these general metrics, various tests can be performed to determine whether current activity fits within acceptable limits.

Statistical Anomaly Detection

- **Mean and standard deviation** of a parameter over some historical period gives a reflection of the average behavior and its variability.
- A **multivariate** model is based on correlations between two or more variables.

Statistical Anomaly Detection

- A **Markov process** model is used to establish transition probabilities among various states.
- A **time series** model focuses on time intervals, looking for sequences of events that happen too rapidly or too slowly.

Statistical Anomaly Detection

- An **operational model** is based on a judgment of what is considered abnormal, rather than an automated analysis of past audit records.

Statistical Anomaly Detection

An Example: Measures That May Be Used for Intrusion Detection

Measure	Model	Type of Intrusion Detected
Command or Program Execution Activity		
Execution frequency	Mean and standard deviation	May detect intruders, who are likely to use different commands, or a successful penetration by a legitimate user, who has gained access to privileged commands.
Program resource utilization	Mean and standard deviation	An abnormal value might suggest injection of a virus or Trojan horse, which performs side-effects that increase I/O or processor utilization.
Execution denials	Operational model	May detect penetration attempt by individual user who seeks higher privileges.

Statistical Anomaly Detection

End

- The main advantage of the use of statistical profiles is that a prior knowledge of security flaws is not required.
- The detector program learns what is “normal” behavior and then looks for deviations.

Rule-Based Intrusion Detection

Network Security

Rule-Based Intrusion Detection

Objectives of the Topic

- After completing this topic, a student will be able to
  - Explain working of rule-based intrusion detection.

Rule-Based Intrusion Detection

Figures and material in this topic have been adapted from

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

Rule-Based Intrusion Detection

- Rule-based techniques detect intrusion by observing events in the system and applying a set of rules that lead to a decision regarding whether a given pattern of activity is or is not suspicious.

Rule-Based Intrusion Detection

- In very general terms, all approaches can be characterized as focusing on either anomaly detection or penetration identification.

Rule-Based Intrusion Detection

- **Rule-based anomaly detection:**
- Is similar in terms of its approach and strengths to statistical anomaly detection.

Rule-Based Intrusion Detection

- Historical audit records are analyzed to identify usage patterns and to automatically generate rules that describe those patterns.

Rule-Based Intrusion Detection

- Rules may represent past behavior patterns of users, programs, privileges, time slots, terminals, and so on.

Rule-Based Intrusion Detection

- Current behavior is then observed, and each transaction is matched against the set of rules to determine if it conforms to any historically observed pattern of behavior.

Rule-Based Intrusion Detection

- In order for this approach to be effective, a rather large database of rules will be needed.

Rule-Based Intrusion Detection

- **Rule-based penetration identification:**
- The key feature of such systems is the use of rules for identifying known penetrations or penetrations that would exploit known weaknesses.

Rule-Based Intrusion Detection

- Rules can also be defined that identify suspicious behavior.
- Typically, the rules used in these systems are specific to the machine and operating system.

Rule-Based Intrusion Detection

- The most fruitful approach to developing such rules is to analyze attack tools and scripts collected on the Internet.

Rule-Based Intrusion Detection

- These rules can be supplemented with rules generated by knowledgeable security personnel.
- For this purpose, normal procedure is to interview system administrators and security analysts.

Rule-Based Intrusion Detection

- **USTAT (a State Transition Analysis Tool for UNIX):**
- Deals in general actions rather than the detailed specific actions recorded by the UNIX auditing mechanism.

Rule-Based Intrusion Detection

- Implemented on a SunOS system that provides audit records on 239 events.
- Of these, only 28 are used by a preprocessor, which maps these onto 10 general actions.

Rule-Based Intrusion Detection

USTAT Actions versus SunOS Event Types

USTAT Action	SunOS Event Type
Read	open_r, open_rc, open_rtc, open_rwc, open_rwtc, open_rt, open_rw, open_rwt
Write	truncate, ftruncate, creat, open_rtc, open_rwc, open_rwtc, open_rt, open_rw, open_rwt, open_w, open_wt, open_wc, open_wct
Create	mkdir, creat, open_rc, open_rtc, open_rwc, open_rwtc, open_wc, open_wtc, mknod
Delete	rmdir, unlink
Execute	exec, execve
Exit	exit
Modify_Owner	chown, fchown
Modify_Perm	chmod, fchmod
Rename	rename
Hardlink	link

Rule-Based Intrusion Detection

- Using just these actions and the parameters that are invoked with each action, a state transition diagram is developed that characterizes suspicious activity.

Rule-Based Intrusion Detection

End

- Because a number of different auditable events map into a smaller number of actions, the rule-creation process is simpler.
- The model is easily modified to accommodate newly learned intrusions.

Distributed Intrusion Detection



Distributed Intrusion Detection

Objectives of the Topic

- After completing this topic, a student will be able to
  - Describe working of distributed intrusion detection.

Distributed Intrusion Detection

Figures and material in this topic have been adapted from

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

Distributed Intrusion Detection

- Traditionally, work on intrusion detection systems focused on single-system standalone facilities.
- A typical organization needs to defend a distributed collection of hosts supported by a LAN or internetwork.

Distributed Intrusion Detection

- It is possible to mount a defense by using stand-alone intrusion detection systems on each host.
- A more effective defense can be achieved by coordination and cooperation among IDSs across network.

Distributed Intrusion Detection

- Following are the **major issues** in the design of a distributed intrusion detection system:
- 1. A distributed intrusion detection system may need to deal with different audit record formats.

Distributed Intrusion Detection

- 2. One or more nodes in network will serve as collection and analysis points for the data from the systems on the network.
- Either raw audit data or summary data must be transmitted across the network.

Distributed Intrusion Detection

- There is a requirement to assure the integrity and confidentiality of these data.
- Integrity is required to prevent an intruder from masking his or her activities by altering transmitted audit information.

Distributed Intrusion Detection

- Confidentiality is required because the transmitted audit information could be valuable.
- 3. Either a centralized or decentralized architecture can be used.

Distributed Intrusion Detection

- With a centralized architecture, there is a single central point of collection and analysis of all audit data.
- This eases the task of correlating incoming reports but creates a potential bottleneck and single point of failure.

Distributed Intrusion Detection

- With a decentralized architecture, there are more than one analysis centers, but these must coordinate their activities and exchange information.

Distributed Intrusion Detection

- Overall architecture of a distributed intrusion detection system consists of three main components:
- **Host agent module:**
- An audit collection module operating as a background process on a monitored system.

Distributed Intrusion Detection

- Its purpose is to collect data on security related events on the host and transmit these to the central manager.

Distributed Intrusion Detection

- **LAN monitor agent module:**
- Operates in the same fashion as a host agent module except that it analyzes LAN traffic and reports the results to the central manager.

Distributed Intrusion Detection

- **Central manager module:**
- Receives reports from LAN monitor and host agents, and processes and correlates these reports to detect intrusion.

Distributed Intrusion Detection

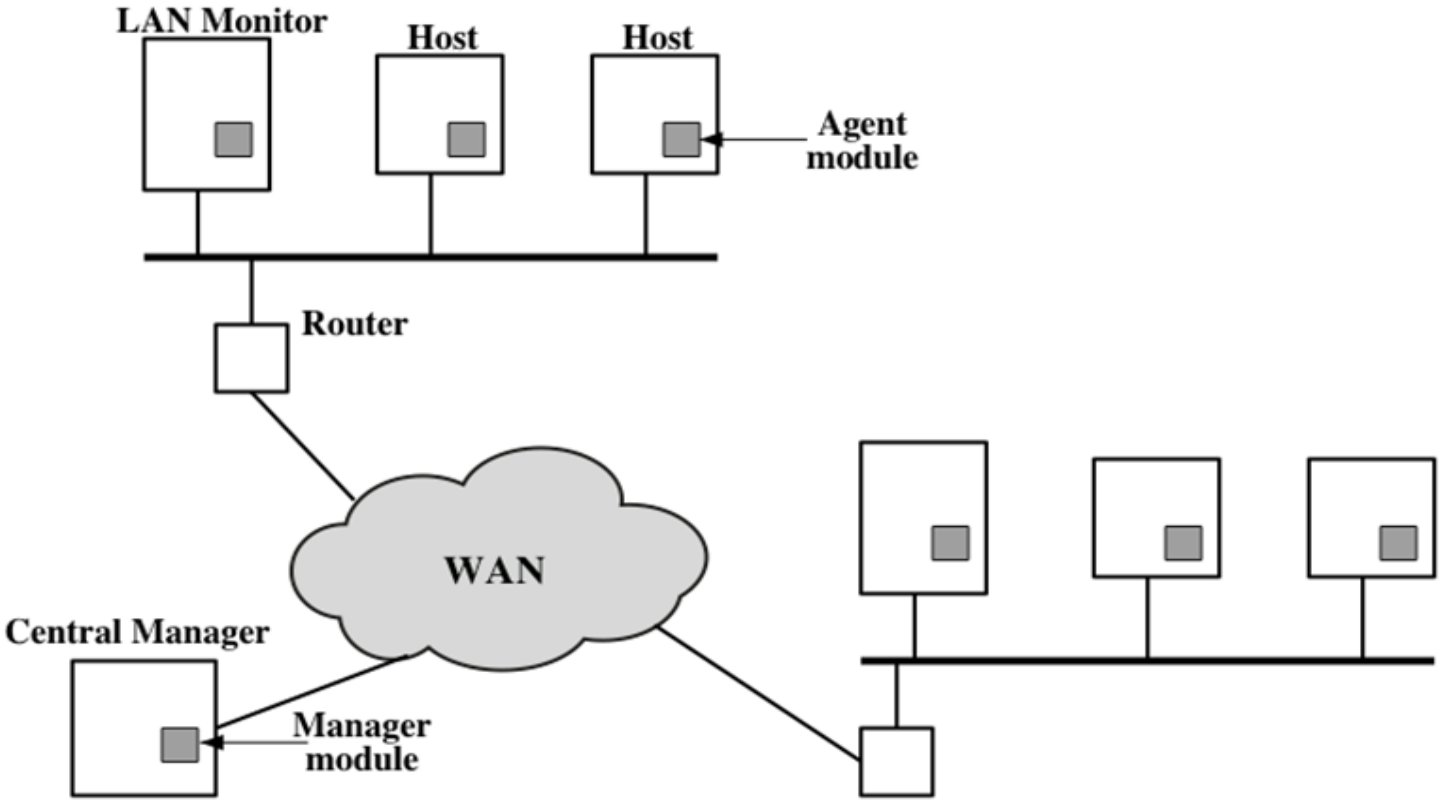


Figure 11.2 Architecture for Distributed Intrusion Detection

Distributed Intrusion Detection

- The scheme is designed to be independent of any operating system or system auditing implementation.
- The **general approach** that is taken is:

Distributed Intrusion Detection

- The agent captures each audit record produced by the native audit collection system.
- A filter is applied that retains only those records that are of security interest.

Distributed Intrusion Detection

- These records are then reformatted into a standardized format referred to as the host audit record (HAR).
- Next, a template-driven logic module analyzes the records for suspicious activity.

Distributed Intrusion Detection

- At the lowest level, the agent scans for notable events that are of interest independent of any past events.
- Examples include failed file accesses, accessing system files, and changing a file's access control.

Distributed Intrusion Detection

- At the next higher level, agent looks for sequences of events, such as known attack patterns (signatures).
- Finally, the agent looks for anomalous behavior of an individual user based on a historical profile of that user.

Distributed Intrusion Detection

Agent Architecture

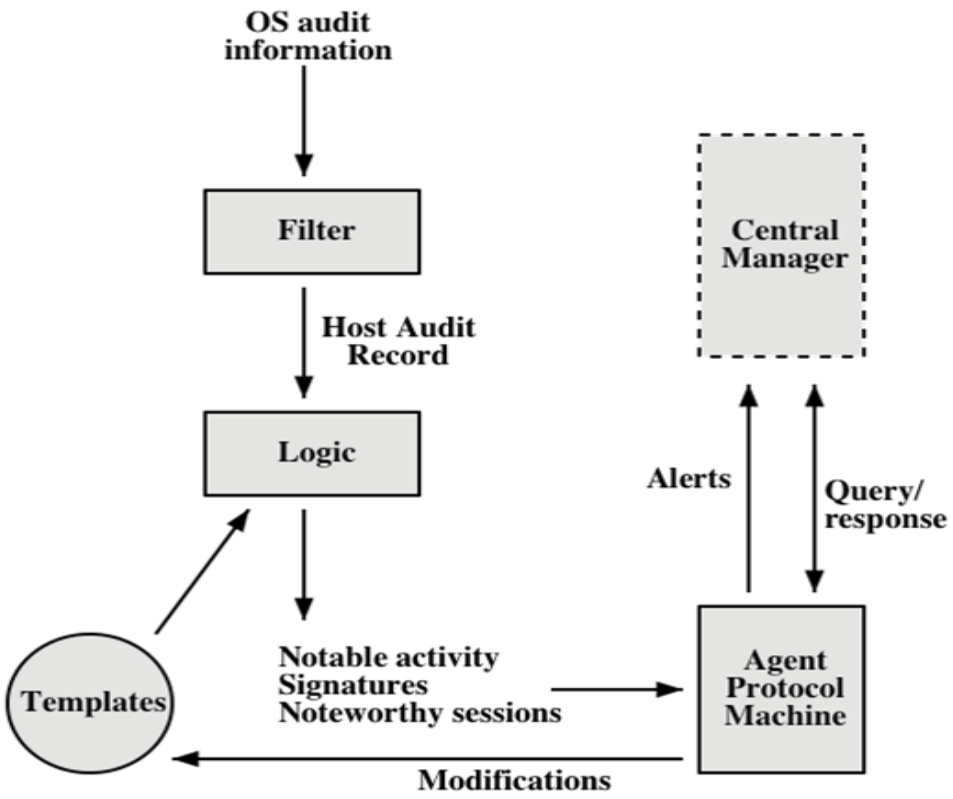


Figure 11.3 Agent Architecture

Distributed Intrusion Detection

End

- When suspicious activity is detected, an alert is sent to the central manager.
- The LAN monitor agent audits host-host connections, services used, and volume of traffic and supplies information to the central manager.

Honeypots



Honeypots

Objectives of the Topic

- After completing this topic, a student will be able to
  - Explain concept of Honeypots.

Honeypots

Figures and material in this topic have been adapted from

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

Honeypots

- Honeypots are decoy systems that are designed to lure a potential attacker away from critical systems.
- Honeypots are **designed to**:
- 1. divert an attacker from accessing critical systems.

Honeypots

- 2. collect information about the attacker's activity.
- 3. encourage the attacker to stay on the system long enough for administrators to respond.

Honeypots

- Honeypots have no **production value**:
- These systems are filled with fabricated information designed to appear valuable but that a legitimate user of the system wouldn’t access.

Honeypots

- Thus, any attempt to communicate with the system is most likely a probe, scan, or attack.
- Conversely, if a honeypot initiates outbound communication, the system has probably been compromised.

Honeypots

- Initial efforts involved a single honeypot computer with IP addresses designed to attract hackers.

Honeypots

- More recent research has focused on building entire honeypot networks that emulate an enterprise, possibly with actual or simulated traffic and data.

Honeypots

- Once hackers are within the network, administrators can observe their behavior in detail and figure out defenses.
- Honeypots can be deployed in a variety of locations.

Honeypots

- The location depends on a number of factors, such as type of information the organization is interested in gathering and level of risk that organizations can tolerate to obtain the maximum amount of data.

Honeypots

- Figure on next slide illustrates some possibilities.

Honeypots

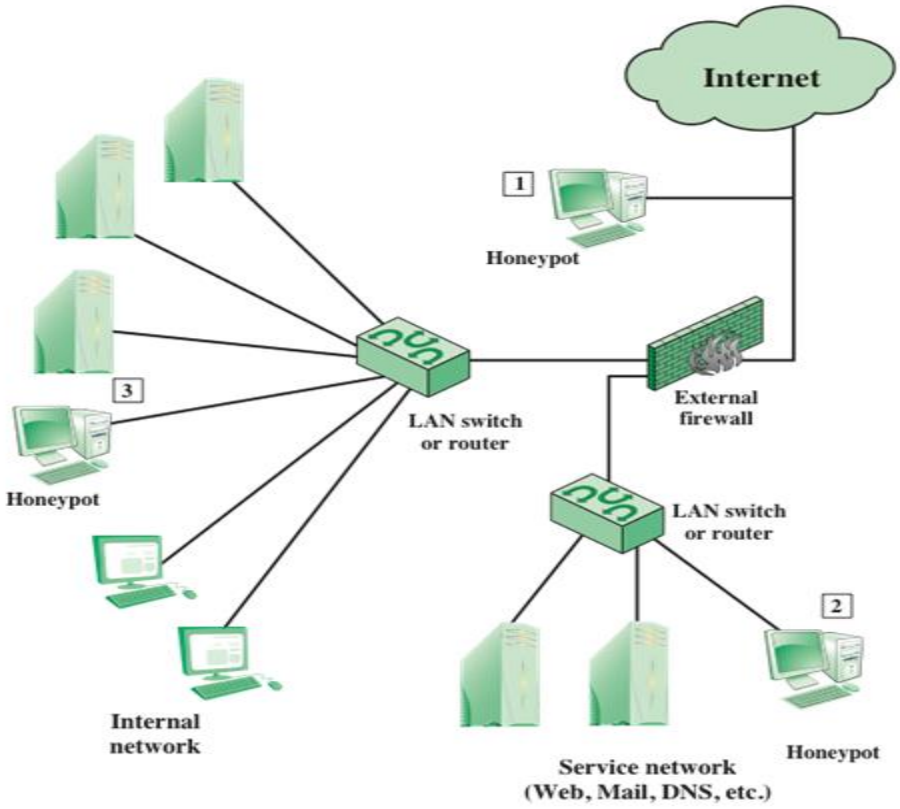


Figure 11.4 Example of Honeypot Deployment

Honeypots

- **Location 1:**
- A honeypot outside the external firewall.
- **Advantages:**
- useful for tracking attempts to connect to unused IP addresses within the scope of the network.

Honeypots

- It does not increase the risk for the internal network.
- It reduces the alerts issued by the firewall and by internal IDS sensors, easing the management burden.

Honeypots

- **Disadvantage:**
- The disadvantage of an external honeypot is that it has little or no ability to trap internal attackers.

Honeypots

- **Location 2:**
- The network of externally available services, such as Web and mail, often called the DMZ (demilitarized zone), is another candidate for locating a honeypot (location 2).

Honeypots

- The security administrator must assure that the other systems in the DMZ are secure against any activity generated by the honeypot.

Honeypots

- **Disadvantage**
- a typical DMZ is not fully accessible, and the firewall typically blocks traffic to the DMZ that attempts to access unneeded services.

Honeypots

- **Location 3:**
- A fully internal honeypot.
- **Advantages:**
- It can catch internal attacks.

Honeypots

- It can detect a misconfigured firewall that forwards impermissible traffic from Internet to the internal network.

Honeypots

End

- **Disadvantages:**
- If the honeypot gets compromised, it can attack other internal systems.
- Any further traffic from Internet to attacker is not blocked by firewall because it is regarded as traffic to the honeypot only.

Password Protection



Password Protection

Objectives of the Topic

- After completing this topic, a student will be able to
  - Explain password protection.

Password Protection

Figures and material in this topic have been adapted from

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

Password Protection

- The front line of defense against intruders is the password system.
- Virtually, all multiuser systems require that a user provide not only a name or identifier (ID) but also a password.

Password Protection

- The password serves to authenticate the ID of the individual logging on to the system.
- The ID provides security in the following ways:

Password Protection

- 1. The ID determines whether the user is authorized to gain access to a system.
- 2. ID determines the privileges accorded to the user.
- 3. The ID is used in what is referred to as discretionary access control.

Password Protection

Attack strategies and Countermeasures:

- Typically, a system that uses password-based authentication maintains a password file indexed by user ID.

Password Protection

- One technique that is typically used is to store not the user's password but a one-way hash function of the password.
- We can identify the following attack strategies and countermeasures:

Password Protection

- **Workstation hijacking:**
- Attacker waits until a logged-in workstation is unattended.
- The standard countermeasure is automatically logging the workstation out after a period of inactivity.

Password Protection

- **Offline dictionary attack:**
- Determined hackers can frequently bypass access controls and gain access to the system's password file.

Password Protection

- Countermeasures include controls to prevent unauthorized access to password file, intrusion detection measures to identify a compromise, and rapid reissuance of passwords should password file be compromised

Password Protection

- **Exploiting user mistakes:**
- The attackers obtain passwords by using social engineering tactics that trick the user into revealing a password.
- Countermeasures include user training, intrusion detection.

Password Protection

- **Specific account attack:** The attacker targets a specific account and submits password guesses until correct password is discovered.
- Countermeasure is to lockout account after a number of failed login attempts.

Password Protection

- A widely used password security technique is the use of hashed passwords and a salt value.
- This scheme is found on virtually all UNIX variants as well as on a number of other operating systems.

Password Protection

- The following procedure is employed :
- To load a new password into the system, the user selects or is assigned a password.
- This password is combined with a fixed-length salt value.

Password Protection

- The salt value is generated by using a pseudorandom or random number.
- The password and salt serve as inputs to a hashing algorithm to produce a fixed-length hash code.

Password Protection

- The hash algorithm is designed to be slow to execute to thwart attacks.
- The hashed password is then stored, together with a plaintext copy of the salt, in the password file for the corresponding user ID.

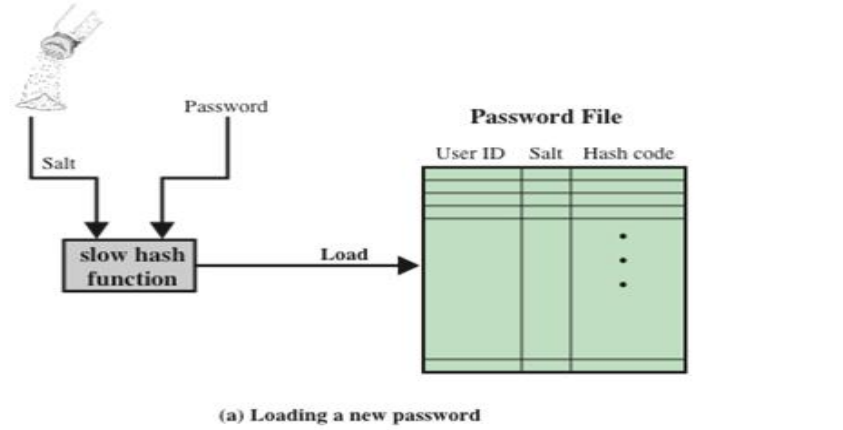
Password Protection

- When a user attempts to log onto a UNIX system, the user provides an ID and a password.
- The operating system uses the ID to index into the password file and retrieve the plaintext salt and the encrypted password.

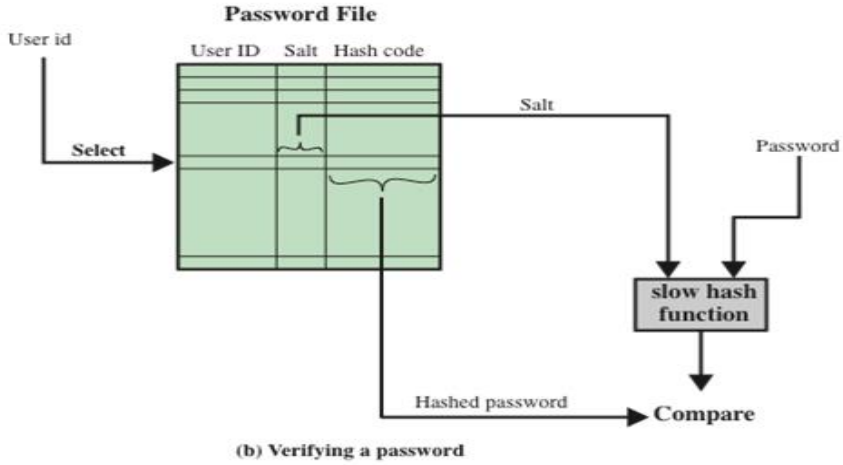
Password Protection

- The salt and user-supplied password are used as input to the encryption routine.
- If the result matches the stored value, the password is accepted.

Password Protection



(a) Loading a new password



(b) Verifying a password

Figure 11.6 UNIX Password Scheme

Password Protection

- The salt serves three purposes:
- 1. It prevents duplicate passwords from being visible in the password file.
- The hashed passwords of the two users will differ.

Password Protection

End

- 2. It greatly increases the difficulty of offline dictionary attacks.
- 3. It becomes nearly impossible to find out whether a person with passwords on two or more systems has used the same password on all of them.

Password Selection Strategies



Password Selection Strategies

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe strategies for password selection.

Password Selection Strategies

Figures and material in this topic have been adapted from

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

Password Selection Strategies

- Many users choose a password that is too short or too easy to guess.
- At other extreme, if passwords consisting of eight randomly selected printable characters are used, password cracking is effectively impossible.

Password Selection Strategies

- But it would be almost as impossible for most users to remember their passwords.
- Our goal is to eliminate guessable passwords while allowing the user to select a password that is memorable.

Password Selection Strategies

- Four basic techniques are in use:
- User education
- Computer-generated passwords
- Reactive password checking
- Proactive password checking

Password Selection Strategies

- **User education:**
- Users can be told the importance of using hard-to-guess passwords and can be provided with guidelines for selecting strong passwords.

Password Selection Strategies

- This user education strategy is unlikely to succeed at most installations.
- Many users will simply ignore the guidelines.
- Others may not be good judges of what is a strong password.

Password Selection Strategies

- **Computer-generated passwords:** also have problems.
- If the passwords are quite random in nature, users will not be able to remember them.

Password Selection Strategies

- FIPS PUB 181 defines one of the best-designed automated password generators.
- The standard includes not only a description of the approach but also a complete listing of the C source code of the algorithm.

Password Selection Strategies

- It generates words by forming pronounceable syllables and concatenating them to form a word.
- A random number generator produces a random stream of characters used to construct syllables.

Password Selection Strategies

- A **reactive password** checking strategy is one in which the system periodically runs its own password cracker to find guessable passwords.
- The system cancels any passwords that are guessed and notifies the user.

Password Selection Strategies

- This tactic has a number of drawbacks.
- It is resource intensive if the job is done right.

Password Selection Strategies

- The most promising approach to improved password security is a **proactive password checker.**
- A user is allowed to select his or her own password.

Password Selection Strategies

- At the time of selection, the system checks to see if the password is allowable and, if not, rejects it.

Password Selection Strategies

- With sufficient guidance from the system, users can select memorable passwords from a fairly large password space that are not likely to be guessed in a dictionary attack.

Password Selection Strategies

Approaches to Proactive Password Checking:

- The following rules could be enforced:
- All passwords must be at least eight characters long.

Password Selection Strategies

- In the first eight characters, the passwords must include at least one each of uppercase, lowercase, numeric digits, and punctuation marks.

Password Selection Strategies

- Another possible procedure is simply to compile a large dictionary of possible “bad” passwords.
- When a user selects a password, the system checks to make sure that it is not on the disapproved list.

Password Selection Strategies

End

- There are two problems:
- **Space:** The dictionary must be very large to be effective.
- **Time:** The time required to search a large dictionary may itself be large.

Basics of Firewalls



Basics of Firewalls

Objectives of the Topic

- After completing this topic, a student will be able to
  - Explain basics of Firewalls.

Basics of Firewalls

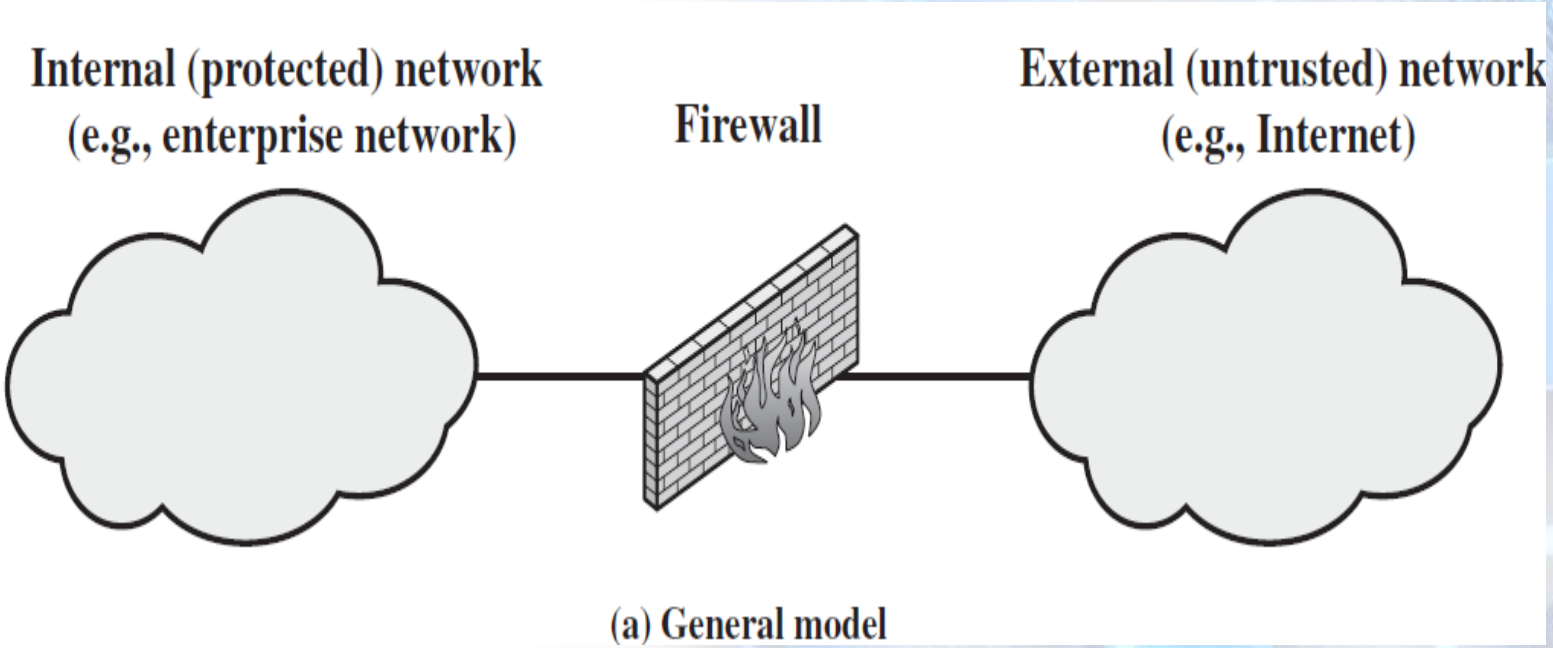
Figures and material in this topic have been adapted from

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

Basics of Firewalls

- Firewalls can be an effective means of protecting a local system from network-based security threats while at the same time affording access to the outside world via wide area networks and the Internet.

Basics of Firewalls



(a) General model

Basics of Firewalls

The Need for Firewalls:

- 1. Internet connectivity is no longer optional for organizations.
- Individual users within the organization want and need Internet access.

Basics of Firewalls

- 2. While Internet access provides benefits to the organization, it enables the outside world to reach and interact with local network assets.
- This creates a threat to the organization.

Basics of Firewalls

- It is possible to equip each workstation and server on the premises network with strong security features, this may not be sufficient and in some cases is not cost-effective.

Basics of Firewalls

- **Firewall:** An alternative, or at least complement, to host-based security services.
- Is inserted between premises network and Internet to establish a controlled link and to erect outer security wall (perimeter).

Basics of Firewalls

- The aim of this perimeter is to protect the premises network from Internet-based attacks and to provide a single choke point where security and auditing can be imposed.

Basics of Firewalls

- May be a single computer system or a set of two or more systems that cooperate to perform the firewall function.

Basics of Firewalls

Firewall Characteristics:

- **Design goals for a firewall:**
- 1. All traffic from inside to outside, and vice versa, must pass through the firewall.

Basics of Firewalls

- 2. Only authorized traffic, as defined by the local security policy, will be allowed to pass.
- 3. The firewall itself is immune to penetration.

Basics of Firewalls

- Four techniques that firewalls use to control access and enforce the site's security policy:
- **1. Service control:** Determines the types of Internet services that can be accessed, inbound or outbound.

Basics of Firewalls

- The firewall may filter traffic on the basis of IP address, protocol, or port number.
- **2. Direction control:** Determines direction in which particular service requests may be initiated and allowed to flow through firewall.

Basics of Firewalls

- **3. User control:** Controls access to a service according to which user is attempting to access it.
- This feature is typically applied to users inside the firewall perimeter (local users).

Basics of Firewalls

- **4. Behavior control:** Controls how particular services are used.
- E.g., firewall may filter e-mail to eliminate spam, or it may enable external access to only a portion of the information on a local Web server.

Basics of Firewalls

Firewall Expectations:

- **1.** A firewall defines a single choke point that keeps unauthorized users out of the protected network, prohibits potentially vulnerable services from entering or leaving network,

Basics of Firewalls

- provides protection from various kinds of IP spoofing and routing attacks.
- **2.** A firewall provides a location for monitoring security-related events.
- Audits and alarms can be implemented on the firewall system.

Basics of Firewalls

- **3.** A firewall is a convenient platform for several Internet functions that are not security related.
- These include a network address translator, and a network management function that audits or logs Internet usage.

Basics of Firewalls

- **4.** A firewall can serve as the platform for IPsec.
- Using the tunnel mode capability, the firewall can be used to implement virtual private networks.

Basics of Firewalls

Firewall Limitations:

- **1.** The firewall cannot protect against attacks that bypass the firewall.

Basics of Firewalls

- **2.** The firewall may not protect fully against internal threats, such as a dissatisfied employee or an employee who unknowingly cooperates with an external attacker.

Basics of Firewalls

- **3.** An internal firewall that separates portions of an enterprise network cannot guard against wireless communications between local systems on different sides of the internal firewall.

Basics of Firewalls

- **4.** A laptop, PDA, or portable storage device may be used and infected outside the corporate network, and then attached and used internally.

End

Packet Filtering Firewall



Packet Filtering Firewall

Objectives of the Topic

- After completing this topic, a student will be able to
  - Explain basics of Packet filtering firewall.

Packet Filtering Firewall

Figures and material in this topic have been adapted from

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

Packet Filtering Firewall

- A firewall may act as a packet filter.
- Depending on the type of firewall, it may examine one or more protocol headers in each packet, the payload of each packet, or the pattern generated by a sequence of packets.

Packet Filtering Firewall

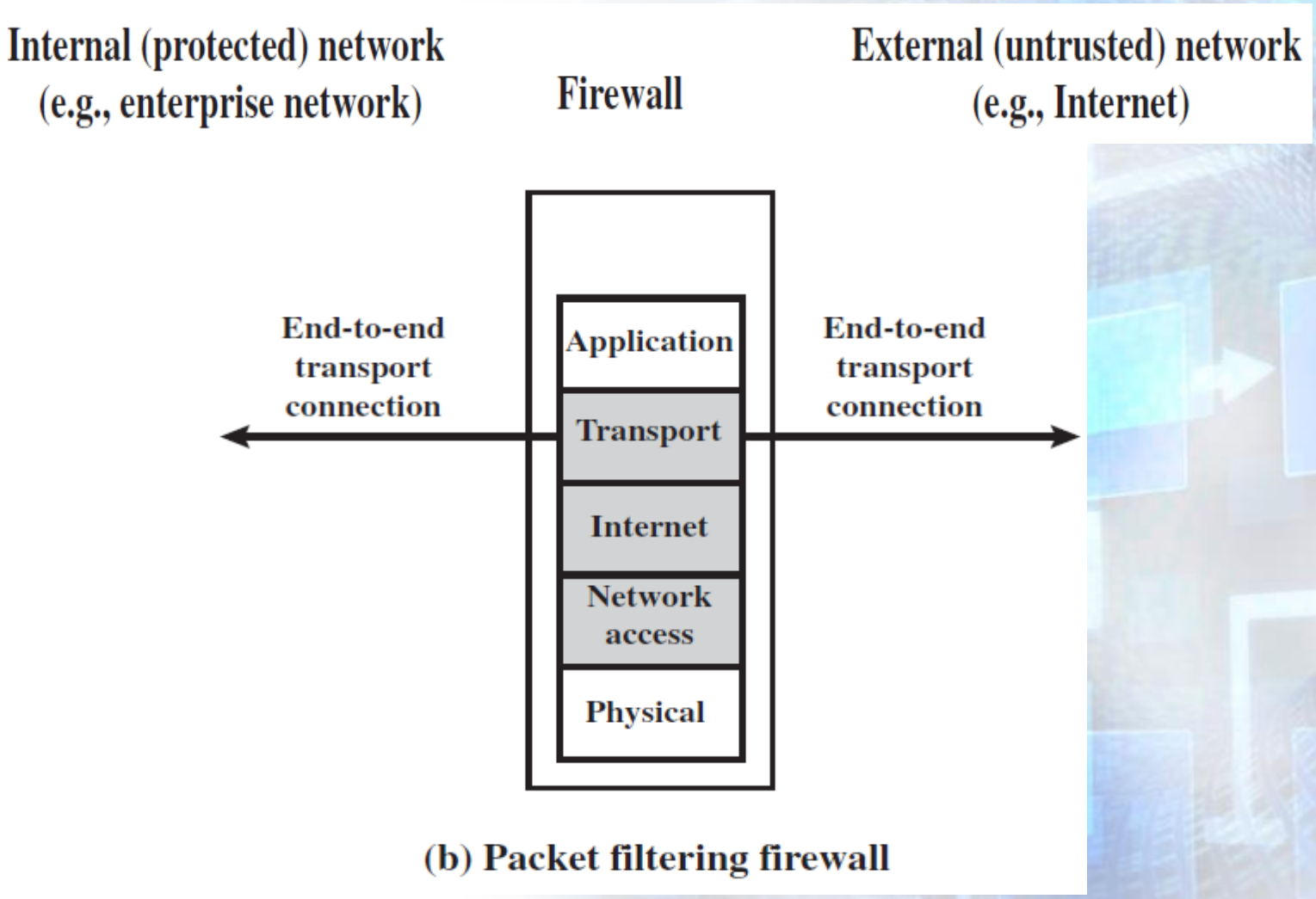
Packet Filtering Firewall:

- A packet filtering firewall applies a set of rules to each incoming and outgoing IP packet and then forwards or discards the packet.

Packet Filtering Firewall

- The firewall is typically configured to filter packets going in both directions (from and to the internal network).

Packet Filtering Firewall



(b) Packet filtering firewall

Packet Filtering Firewall

- Filtering rules are based on information contained in a network packet:
- **Source IP address:**
- The IP address of the system that originated the IP packet (e.g., 192.178.1.1)

Packet Filtering Firewall

- **Destination IP address:**
- The IP address of the system the IP packet is trying to reach (e.g., 192.168.1.2)

Packet Filtering Firewall

- **Source and destination transport-level address:**
- The transport-level (e.g., TCP or UDP) port number, which defines applications such as SNMP or TELNET.

Packet Filtering Firewall

- **IP protocol field:** Defines the transport protocol.
- **Interface:** For a firewall with three or more ports, which interface of firewall the packet came from or which interface of the firewall the packet is destined for.

Packet Filtering Firewall

- The packet filter is typically set up as a list of rules based on matches to fields in the IP or TCP header.
- If there is a match to one of the rules, that rule is invoked to determine whether to forward or discard the packet.

Packet Filtering Firewall

- If there is no match to any rule, then a default action is taken.
- Two default policies are possible:
- **1. Default = discard:** That which is not expressly permitted is prohibited.

Packet Filtering Firewall

- This policy is more conservative.
- Initially, everything is blocked, and services must be added on a case-by-case basis.
- This policy is more visible to users, who are more likely to see the firewall as a hindrance.

Packet Filtering Firewall

- This is the policy likely to be preferred by businesses and government organizations.
- Visibility to users diminishes as rules are created.

Packet Filtering Firewall

- **2. Default = forward:** That which is not expressly prohibited is permitted.

Packet Filtering Firewall

- This policy increases ease of use for end users but provides reduced security.
- The security administrator must react to each new security threat as it becomes known.

Packet Filtering Firewall

- This policy may be used by generally more open organizations, such as universities.

Packet Filtering Firewall

- Next we show some rules.
- In each set, the rules are applied top to bottom.
- The “*” in a field is a wildcard designator that matches everything.
- Assume the **default = discard** policy.

Packet Filtering Firewall

Examples of packet filtering rulesets:

Rule Set A

action	ourhost	port	theirhost	port	comment
block	*	*	SPIGOT	*	we don't trust these people
allow	OUR-GW	25	*	*	connection to our SMTP port

- Inbound mail is allowed (port 25 is for SMTP incoming), but only to a gateway host.
- Packets from a particular external host, SPIGOT, are blocked because that host has a history of sending massive files in e-mail messages.

Packet Filtering Firewall

Examples of packet filtering rulesets:

Rule Set B

action	ourhost	port	theirhost	port	comment
block	*	*	*	*	default

- This is an explicit statement of the default policy. All rulesets include this rule implicitly as the last rule.

Packet Filtering Firewall

Examples of packet filtering rulesets:

Rule Set C

action	ourhost	port	theirhost	port	comment
allow	*	*	*	25	connection to their SMTP port

- This ruleset is intended to specify that any inside host can send mail to the outside.
- The problem with this rule is that the use of port 25 for SMTP receipt is only a default. An attacker could gain access to internal machines by sending packets with a TCP source port number of 25.

Packet Filtering Firewall

Examples of packet filtering rulesets:

Rule Set D

action	src	port	dest	port	flags	comment
allow	{our hosts}	*	*	25		our packets to their SMTP port
allow	*	25	*	*	ACK	their replies

- This ruleset states that it allows IP packets where the source IP address is one of a list of designated internal hosts and the destination TCP port number is 25. It also allows incoming packets with a source port number of 25 that include the ACK flag in the TCP segment.

Packet Filtering Firewall

Rule Set E

action	src	port	dest	port	flags	comment
allow	{our hosts}	*	*	*		our outgoing calls
allow	*	*	*	*	ACK	replies to our calls
allow	*	*	*	>1024		traffic to nonservers

- This ruleset allows
- Packets that originate internally
- Reply packets to a connection initiated by an internal machine
- Packets destined for a high-numbered port on an internal machine

Weaknesses of Packet Filter Firewalls



Weaknesses of Packet Filter Firewalls

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe weaknesses of Packet filtering firewall.

Weaknesses of Packet Filter Firewalls

Figures and material in this topic have been adapted from

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

Weaknesses of Packet Filter Firewalls

- One advantage of a packet filtering firewall is its simplicity.
- Also, packet filters typically are transparent to users and are very fast.

Weaknesses of Packet Filter Firewalls

- Following are the weaknesses of packet filter firewalls:
- 1. Because packet filter firewalls do not examine upper-layer data, they cannot prevent attacks that employ application-specific vulnerabilities or functions.

Weaknesses of Packet Filter Firewalls

- 2. Because of the limited information available to the firewall, the logging functionality present in packet filter firewalls is limited.

Weaknesses of Packet Filter Firewalls

- 3. Most packet filter firewalls do not support advanced user authentication schemes.
- This limitation is mostly due to the lack of upper-layer functionality by the firewall.

Weaknesses of Packet Filter Firewalls

- 4. Packet filter firewalls are generally vulnerable to attacks and exploits that take advantage of problems within the TCP/IP specification and protocol stack, such as *network layer address spoofing*.

Weaknesses of Packet Filter Firewalls

- Many packet filter firewalls cannot detect a network packet in which the OSI Layer 3 addressing information has been altered.
- Intruders can bypass the security controls implemented in a firewall platform.

Weaknesses of Packet Filter Firewalls

- 5. Due to the small number of variables used in access control decisions, packet filter firewalls are susceptible to security breaches caused by improper configurations.

Weaknesses of Packet Filter Firewalls

Attacks and Countermeasures:

- Some of the attacks that can be made on packet filtering firewalls and the appropriate countermeasures are the following:

Weaknesses of Packet Filter Firewalls

- **1. IP address spoofing:**
- The intruder transmits packets from the outside with a source IP address field containing an address of an internal host.

Weaknesses of Packet Filter Firewalls

- The attacker hopes that the use of a spoofed address will allow penetration of systems that employ simple source address security, in which packets from specific trusted internal hosts are accepted.

Weaknesses of Packet Filter Firewalls

- The **countermeasure** is to discard packets with an inside source address if the packet arrives on an external interface.
- This countermeasure is often implemented at the router external to the firewall.

Weaknesses of Packet Filter Firewalls

- **2. Source routing attacks:** The source station specifies the route that a packet should take as it crosses the Internet, in the hopes that this will bypass security measures that do not analyze the source routing information.

Weaknesses of Packet Filter Firewalls

- The **countermeasure** is to discard all packets that use this option.

Weaknesses of Packet Filter Firewalls

- **3. Tiny fragment attacks:**
- The intruder uses the IP fragmentation option to create extremely small fragments and forces the TCP header information into a separate packet fragment.

Weaknesses of Packet Filter Firewalls

- This attack is designed to circumvent filtering rules that depend on TCP header information.
- Typically, a packet filter will make a filtering decision on the first fragment of a packet.

Weaknesses of Packet Filter Firewalls

- All subsequent fragments of that packet are filtered out solely on the basis that they are part of the packet whose first fragment was rejected.

Weaknesses of Packet Filter Firewalls

- The attacker hopes that the filtering firewall examines only the first fragment and that the remaining fragments are passed through.

Weaknesses of Packet Filter Firewalls

- A tiny fragment attack can be defeated by enforcing a rule that the first fragment of a packet must contain a predefined minimum amount of the transport header.

Weaknesses of Packet Filter Firewalls

- If the first fragment is rejected, the filter can remember the packet and discard all subsequent fragments.

End

Stateful Inspection Firewalls



Stateful Inspection Firewalls

Objectives of the Topic

- After completing this topic, a student will be able to
  - Explain working of Stateful Inspection firewall.

Stateful Inspection Firewalls

Figures and material in this topic have been adapted from

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

Stateful Inspection Firewalls

- A traditional packet filter makes filtering decisions on an individual packet basis and does not take into consideration any higher layer context.

Stateful Inspection Firewalls

- Let's try to understand what is meant by *context* and why a traditional packet filter is limited with regard to context.

Stateful Inspection Firewalls

- Most standardized applications that run on top of TCP follow a client/server model.
- For example, for the Simple Mail Transfer Protocol (SMTP), e-mail is transmitted from a client system to a server system.

Stateful Inspection Firewalls

- The client system generates new e-mail messages, typically from user input.
- The server system accepts incoming e-mail messages and places them in the appropriate user mailboxes.

Stateful Inspection Firewalls

- SMTP operates by setting up a TCP connection between client and server, in which the TCP server port number, which identifies the SMTP server application, is 25.

Stateful Inspection Firewalls

- The TCP port number for the SMTP client is a number between 1024 and 65535 that is generated by the SMTP client.

Stateful Inspection Firewalls

- In general, when an application that uses TCP creates a session with a remote host, server side port number is less than 1024 and port number for client is a number between 1024 and 65535.

Stateful Inspection Firewalls

- The numbers less than 1024 are the “well-known” port numbers and are assigned permanently to particular applications (e.g., 25 for server SMTP, 80 for the webserver).

Stateful Inspection Firewalls

- The numbers between 1024 and 65535 are generated dynamically and have temporary significance only for the lifetime of a TCP connection.

Stateful Inspection Firewalls

- A simple packet filtering firewall must permit inbound network traffic on all these high-numbered ports for TCP-based traffic to occur.
- This creates a vulnerability that can be exploited by unauthorized users.

Stateful Inspection Firewalls

- A stateful inspection packet firewall tightens up the rules for TCP traffic by creating a directory of outbound TCP connections.
- There is an entry for each currently established connection.

Stateful Inspection Firewalls

Example Stateful Firewall Connection State Table

Source Address	Source Port	Destination Address	Destination Port	Connection State
192.168.1.100	1030	210.22.88.29	80	Established
192.168.1.102	1031	216.32.42.123	80	Established
192.168.1.101	1033	173.66.32.122	25	Established
192.168.1.106	1035	177.231.32.12	79	Established
223.43.21.231	1990	192.168.1.6	80	Established
2122.22.123.32	2112	192.168.1.6	80	Established
210.922.212.18	3321	192.168.1.6	80	Established
24.102.32.23	1025	192.168.1.6	80	Established
223.21.22.12	1046	192.168.1.6	80	Established

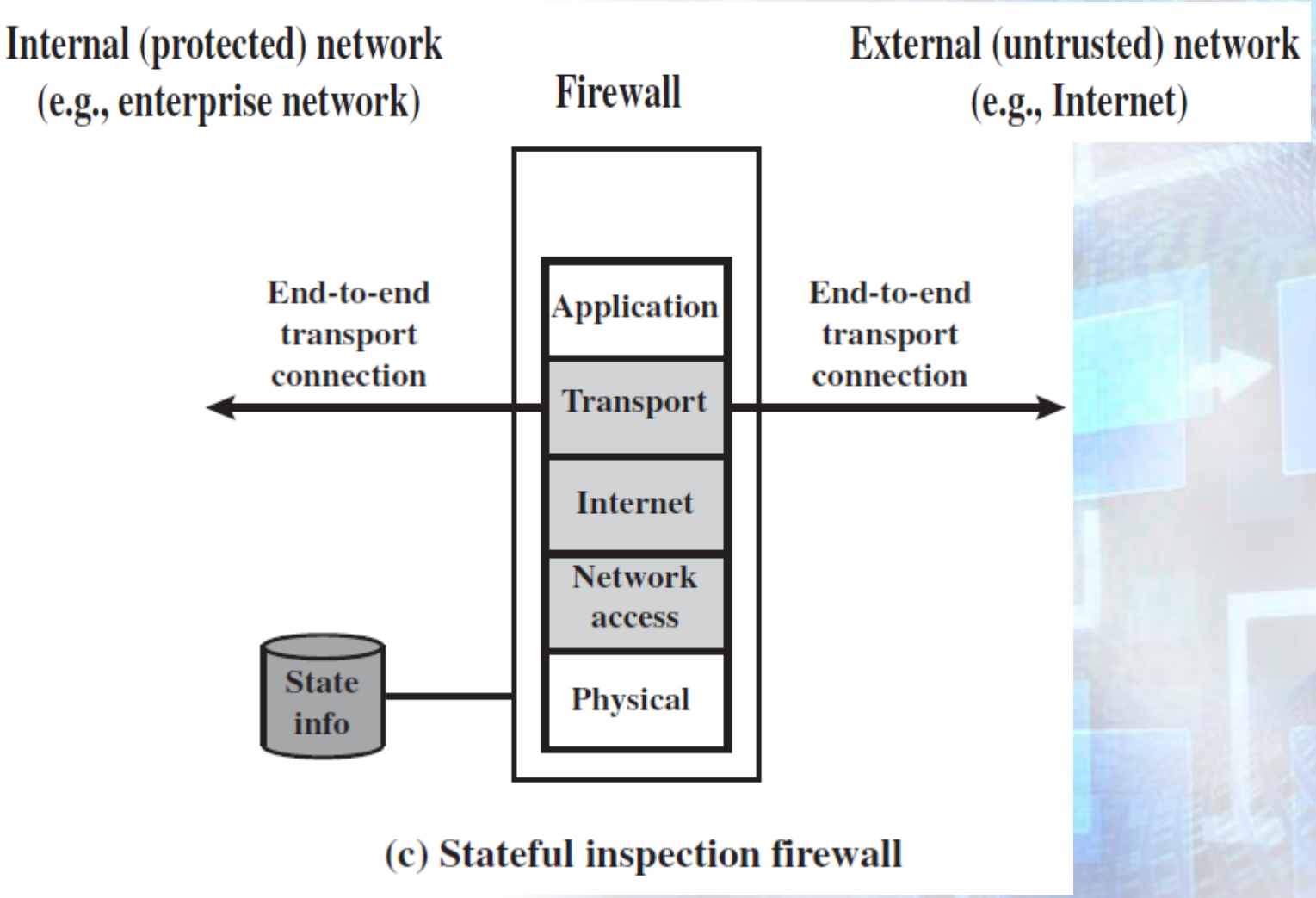
Stateful Inspection Firewalls

- The packet filter will now allow incoming traffic to high-numbered ports only for those packets that fit the profile of one of the entries in this directory.

Stateful Inspection Firewalls

- A stateful packet inspection firewall reviews the same packet information as a packet filtering firewall, but also records information about TCP connections.

Stateful Inspection Firewalls



(c) Stateful inspection firewall

Stateful Inspection Firewalls

- Some stateful firewalls also keep track of TCP sequence numbers to prevent attacks that depend on the sequence number, such as session hijacking.

Stateful Inspection Firewalls

End

- Some even inspect limited amounts of application data for some well-known protocols like FTP, IM and SIPS commands, in order to identify and track related connections.

Application and Circuit-Level Gateways



Application and Circuit-Level Gateways

Objectives of the Topic

- After completing this topic, a student will be able to
  - Explain working of application and circuit-level firewalls.

Application and Circuit-Level Gateways

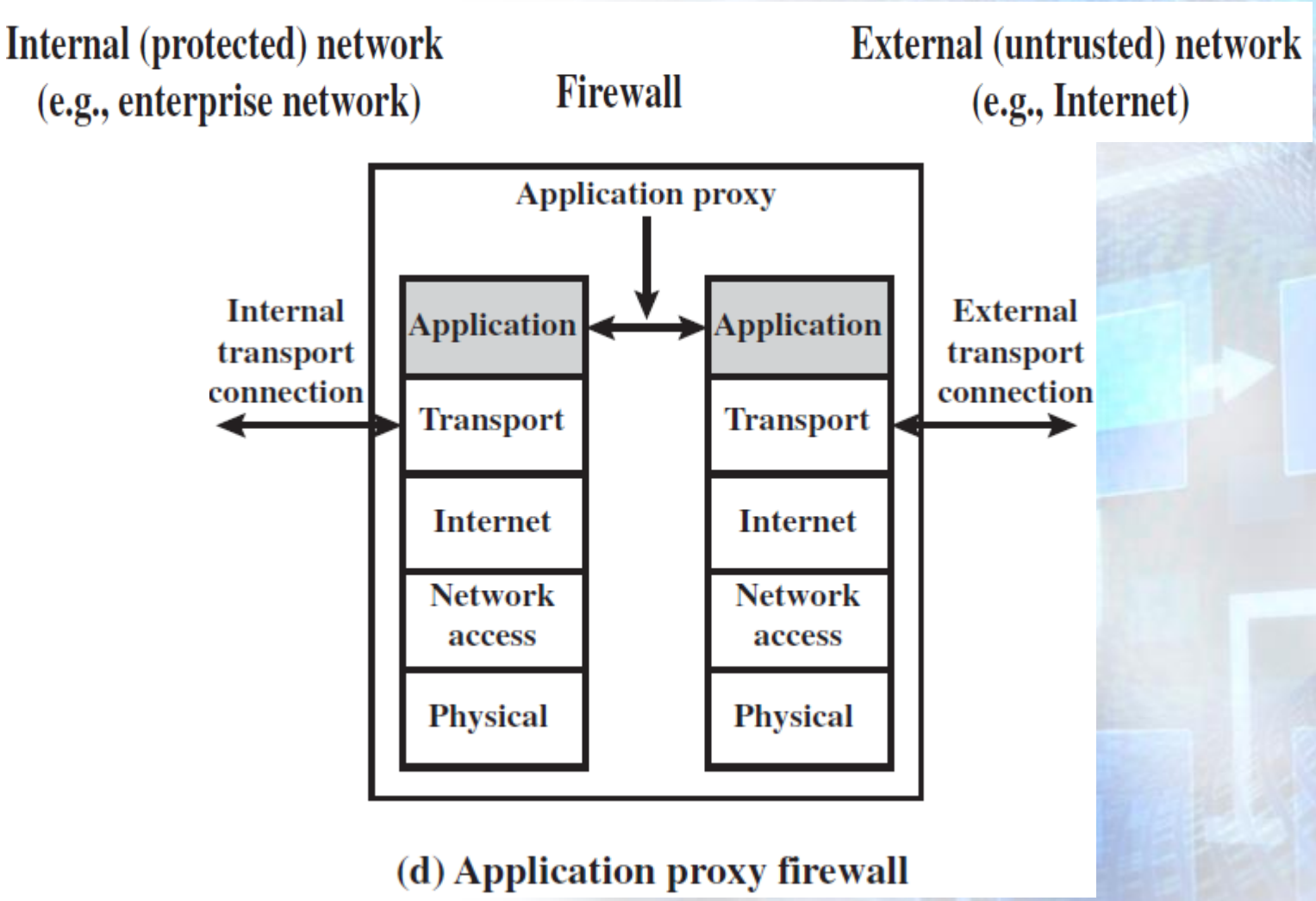
Figures and material in this topic have been adapted from

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

Application and Circuit-Level Gateways

- An application-level gateway, also called an **application proxy**, acts as a relay of application-level traffic.

Application and Circuit-Level Gateways



(d) Application proxy firewall

Application and Circuit-Level Gateways

- The user contacts the gateway using a TCP/IP application, such as Telnet or FTP, and the gateway asks the user for the name of the remote host to be accessed.

Application and Circuit-Level Gateways

- When user responds and provides a valid user ID and authentication information, gateway contacts application on remote host and relays TCP segments containing application data between two endpoints.

Application and Circuit-Level Gateways

- If the gateway does not implement the proxy code for a specific application, the service is not supported and cannot be forwarded across the firewall.

Application and Circuit-Level Gateways

- The gateway can be configured to support only specific features of an application that the network administrator considers acceptable while denying all other features.

Application and Circuit-Level Gateways

- Application-level gateways tend to be more secure than packet filters.
- Only scrutinize a few allowable applications.
- Also, it is easy to log and audit all incoming traffic at the application level.

Application and Circuit-Level Gateways

- **Disadvantage**: Processing overhead on each connection.
- There are two spliced connections between end users with gateway at splice point, and gateway must examine and forward all traffic in both directions.

Application and Circuit-Level Gateways

Circuit-Level Gateway:

- or **circuit-level proxy** can be a stand-alone system or it can be a specialized function performed by an application-level gateway for certain applications.

Application and Circuit-Level Gateways

- Like application gateway, it does not permit an end-to-end TCP connection.
- It sets up two TCP connections, one between itself and a TCP user on an inner host and one between itself and a TCP user on an outside host.

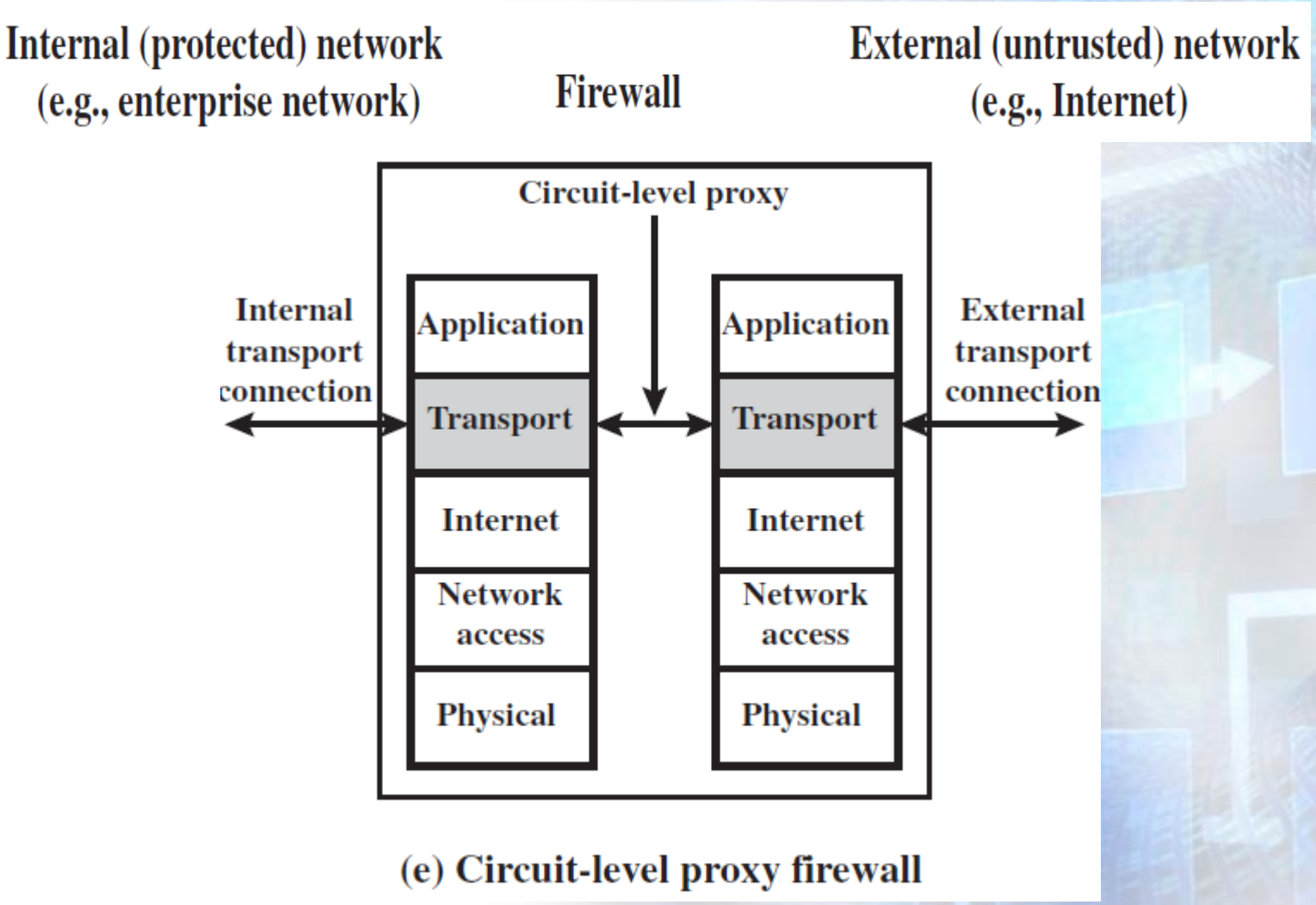
Application and Circuit-Level Gateways

- Once the two connections are established, gateway typically relays TCP segments from one connection to the other without examining contents.

Application and Circuit-Level Gateways

- The security function consists of determining which connections will be allowed.

Application and Circuit-Level Gateways



(e) Circuit-level proxy firewall

Application and Circuit-Level Gateways

- A typical use of circuit-level gateways is a situation in which the system administrator trusts the internal users.

Application and Circuit-Level Gateways

- An example of a circuit-level gateway implementation is the SOCKS package.
- It consists of the following components:

Application and Circuit-Level Gateways

- 1. The SOCKS server, which often runs on a UNIX-based firewall.
- SOCKS is also implemented on Windows systems.
- 2. The SOCKS client library, which runs on internal hosts protected by the firewall.

Application and Circuit-Level Gateways

- 3. SOCKS-ified versions of several standard client programs such as FTP and TELNET.

End

Firewall Basing



Firewall Basing

Objectives of the Topic

- After completing this topic, a student will be able to
  - describe Firewall basing.

Firewall Basing

Figures and material in this topic have been adapted from

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

Firewall Basing

- It is common to base a firewall on a stand-alone machine running a common operating system, such as UNIX or Linux.
- Firewall functionality can also be implemented as a software module in a router or LAN switch.

Firewall Basing

- An additional firewall basing consideration is:

Firewall Basing

Bastion Host:

- is a system identified by the firewall administrator as a critical strong point in network's security.
- Typically, the bastion host serves as a platform for an application-level or circuit-level gateway.

Firewall Basing

- **Common characteristics** of a bastion host are as follows:
- 1. The bastion host hardware platform executes a secure version of its operating system, making it a hardened system.

Firewall Basing

- 2. Only the services that the network administrator considers essential are installed on the bastion host.
- These could include proxy applications for DNS, FTP, HTTP, and SMTP.

Firewall Basing

- 3. Bastion host may require additional authentication before a user is allowed access to the proxy services.
- Also, each proxy service may require its own authentication before granting user access.

Firewall Basing

- 4. Each proxy is configured to support only a subset of the standard application's command set.

Firewall Basing

- 5. Each proxy is configured to allow access only to specific host systems.
- This means that the limited command/feature set may be applied only to a subset of systems on the protected network.

Firewall Basing

- 6. Each proxy maintains detailed audit information by logging all traffic, each connection, and the duration of each connection.
- The audit log is an essential tool for discovering and terminating attacks.

Firewall Basing

- 7. Each proxy module is a very small software package specifically designed for network security.
- It is easier to check such modules for security flaws.

Firewall Basing

- 8. Each proxy is independent of other proxies on the bastion host.
- If there is a problem with the operation of any proxy, it can be uninstalled without affecting operation of the other proxy applications.

Firewall Basing

- 9. A proxy generally performs no disk access other than to read its initial configuration file.

Firewall Basing

- The portions of the file system containing executable code can be made read only.
- This makes it difficult for an intruder to install Trojan horse sniffers or other dangerous files on the bastion host.

Firewall Basing

- 10. Each proxy runs as a nonprivileged user in a private and secured directory on the bastion host.

End

Demilitarized Zone (DMZ) Networks



Demilitarized Zone (DMZ) Networks

Objectives of the Topic

- After completing this topic, a student will be able to
  - Explain working of Demilitarized Zone (DMZ) Networks.

Demilitarized Zone (DMZ) Networks

Figures and material in this topic have been adapted from

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

Demilitarized Zone (DMZ) Networks

- A firewall is positioned to provide a protective barrier between an external, potentially untrusted source of traffic and an internal network.

Demilitarized Zone (DMZ) Networks

- With that general principle in mind, a security administrator must decide on the location and on the number of firewalls needed.
- In this topic, we look at one such common option.

Demilitarized Zone (DMZ) Networks

DMZ Networks:

- Next figure depicts the most common distinction, that between an internal and an external firewall.

Demilitarized Zone (DMZ) Networks

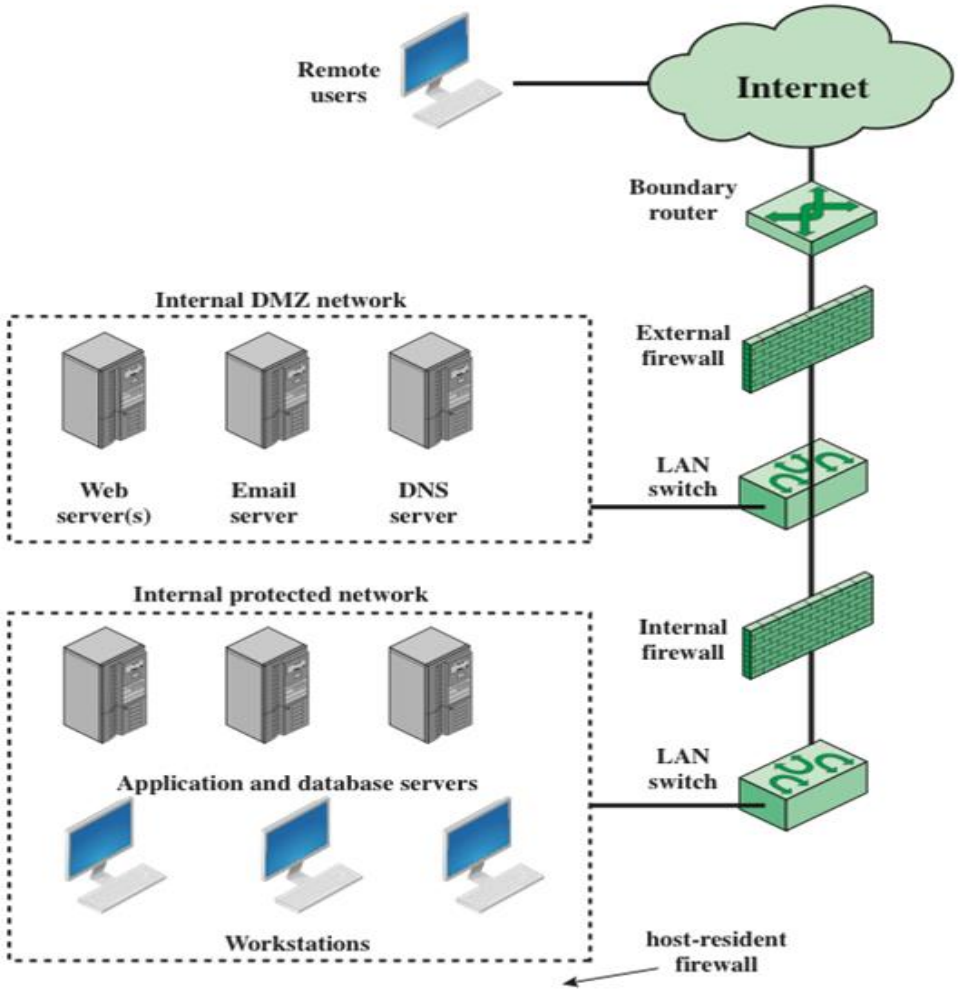


Figure 12.3 Example Firewall Configuration

Demilitarized Zone (DMZ) Networks

- An external firewall is placed at the edge of a local or enterprise network, just inside the boundary router that connects to the Internet or some wide area network (WAN).

Demilitarized Zone (DMZ) Networks

- One or more internal firewalls protect the bulk of the enterprise network.
- Between these two types of firewalls are one or more networked devices in a region referred to as a DMZ (demilitarized zone) network.

Demilitarized Zone (DMZ) Networks

- Systems that are externally accessible but need some protections are usually located on DMZ networks.

Demilitarized Zone (DMZ) Networks

- Typically, the systems in the DMZ require or foster external connectivity, such as a corporate Web site, an e-mail server, or a DNS (domain name system) server.

Demilitarized Zone (DMZ) Networks

- The external firewall provides a measure of access control and protection for the DMZ systems consistent with their need for external connectivity.

Demilitarized Zone (DMZ) Networks

- The external firewall also provides a basic level of protection for the remainder of the enterprise network.
- In this type of configuration, internal firewalls serve three purposes:

Demilitarized Zone (DMZ) Networks

- **1.** The internal firewall adds more stringent filtering capability, compared to the external firewall, in order to protect enterprise servers and workstations from external attack.

Demilitarized Zone (DMZ) Networks

- **2.** The internal firewall provides two-way protection with respect to the DMZ.
- First, the internal firewall protects the remainder of the network from attacks launched from DMZ systems.

Demilitarized Zone (DMZ) Networks

- Such attacks might originate from worms, rootkits, bots, or other malware lodged in a DMZ system.
- Second, an internal firewall can protect the DMZ systems from attack from the internal protected network.

Demilitarized Zone (DMZ) Networks

- **3.** Multiple internal firewalls can be used to protect portions of the internal network from each other.
- For example, firewalls can be configured so that internal servers are protected from internal workstations and vice versa.

Demilitarized Zone (DMZ) Networks

- A common practice is to place the DMZ on a different network interface on the external firewall from that used to access the internal networks.

End

VPNs and Distributed Firewalls



VPNs and Distributed Firewalls

Objectives of the Topic

- After completing this topic, a student will be able to
  - Describe working of VPNs and Distributed Firewalls .

VPNs and Distributed Firewalls

Figures and material in this topic have been adapted from

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

VPNs and Distributed Firewalls

- At each corporate site, workstations, servers, and databases are linked by one or more local area networks (LANs).

VPNs and Distributed Firewalls

- The Internet or some other public network can be used to interconnect sites, providing a cost savings over use of a private network and offloading wide area network management task to the public network provider.

VPNs and Distributed Firewalls

- That same public network provides an access path for telecommuters and other mobile employees to log on to corporate systems from remote sites.

VPNs and Distributed Firewalls

- But the manager faces a fundamental requirement: security
- Use of a public network exposes corporate traffic to eavesdropping and provides an entry point for unauthorized users.
- Solution: VPN

VPNs and Distributed Firewalls

- A Virtual Private Network (VPN) uses encryption and authentication in the lower protocol layers to provide a secure connection through an otherwise insecure network, typically the Internet.

VPNs and Distributed Firewalls

- VPNs are generally cheaper than real private networks using private lines but rely on having the same encryption and authentication system at both ends.

VPNs and Distributed Firewalls

- The encryption may be performed by firewall software or possibly by routers.
- The most common protocol mechanism used for this purpose is at the IP level and is known as IPsec.

VPNs and Distributed Firewalls

- An organization maintains LANs at dispersed locations.
- Nonsecure IP traffic is conducted on each LAN.
- For traffic off site, through some sort of private or public WAN, IPsec protocols are used.

VPNs and Distributed Firewalls

- These protocols operate in networking devices, such as a router or firewall, that connect each LAN to the outside world.

VPNs and Distributed Firewalls

- IPsec networking device encrypts and compresses all traffic going into WAN and decrypts and uncompresses traffic coming from WAN.
- Authentication may also be provided.
- These operations are transparent to LAN.

VPNs and Distributed Firewalls

- Secure transmission is also possible with individual users who dial into the WAN.
- Such user workstations must implement the IPsec protocols to provide security.

VPNs and Distributed Firewalls

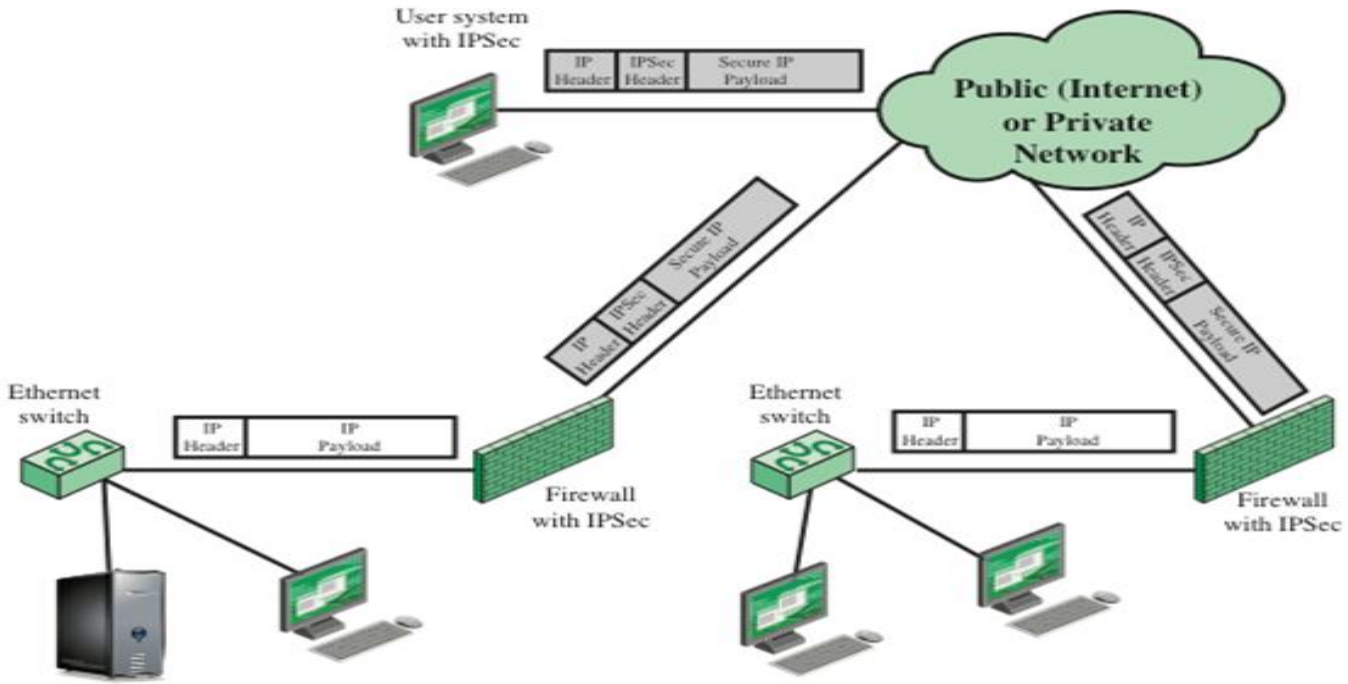


Figure 12.4 A VPN Security Scenario

VPNs and Distributed Firewalls

Distributed Firewalls:

- A distributed firewall configuration involves stand-alone firewall devices plus host-based firewalls working together under a central administrative control.

VPNs and Distributed Firewalls

- Administrators can configure host-resident firewalls on hundreds of servers and workstations as well as configure personal firewalls on local and remote user systems.

VPNs and Distributed Firewalls

- Tools let the network administrator set policies and monitor security across the entire network.
- These firewalls protect against internal attacks and provide protection to specific machines and applications.

VPNs and Distributed Firewalls

- Stand-alone firewalls provide global protection, including internal firewalls and an external firewall.

VPNs and Distributed Firewalls

- With distributed firewalls, both internal and external DMZ may be established.
- Web servers contain less critical information, need less protection and could be placed in an external DMZ, outside the external firewall.

VPNs and Distributed Firewalls

- The protection that is needed is provided by host-based firewalls on these servers.
- An important aspect of a distributed firewall configuration is security monitoring.

VPNs and Distributed Firewalls

- Such monitoring typically includes log aggregation and analysis, firewall statistics, and fine-grained remote monitoring of individual hosts if needed.

VPNs and Distributed Firewalls

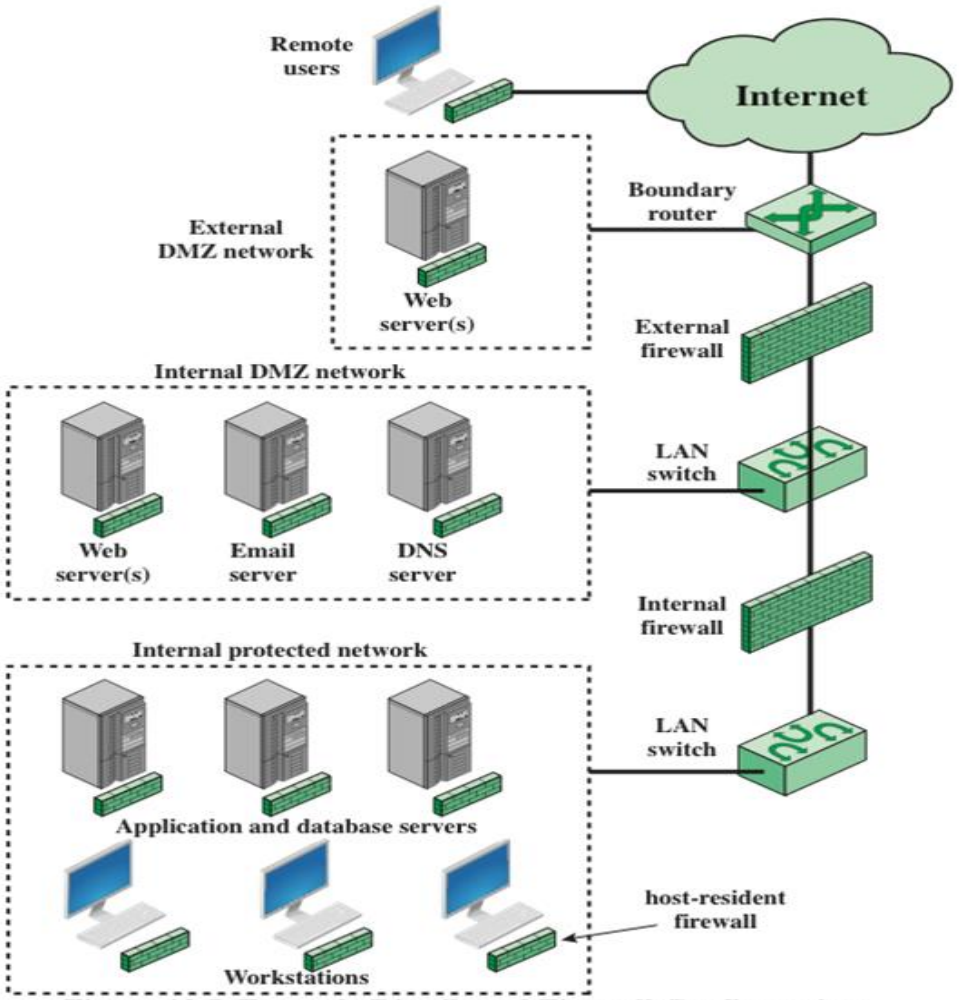


Figure 12.5 Example Distributed Firewall Configuration

Host-Based and Personal Firewalls



Host-Based and Personal Firewalls

Objectives of the Topic

- After completing this topic, a student will be able to
  - Explain basics of Host-Based and Personal Firewalls.

Host-Based and Personal Firewalls

Figures and material in this topic have been adapted from

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

Host-Based and Personal Firewalls

- It is common to base a firewall on a stand-alone machine running a common operating system, such as UNIX or Linux.
- Firewall functionality can also be implemented as a software module in a router or LAN switch.

Host-Based and Personal Firewalls

- We look at two additional firewall basing considerations.

Host-Based and Personal Firewalls

Host-Based Firewalls:

- A host-based firewall is a software module used to secure an individual host.
- Such modules are available in many operating systems or can be provided as an add-on package.

Host-Based and Personal Firewalls

- Like conventional stand-alone firewalls, host-resident firewalls filter and restrict the flow of packets.
- A common location for such firewalls is a server.

Host-Based and Personal Firewalls

- There are several advantages to the use of a server-based or workstation based firewall:

Host-Based and Personal Firewalls

- 1. Filtering rules can be tailored to the host environment.
- Specific corporate security policies for servers can be implemented, with different filters for servers used for different application.

Host-Based and Personal Firewalls

- 2. Protection is provided independent of topology.
- Thus both internal and external attacks must pass through the firewall.

Host-Based and Personal Firewalls

- 3. Used in conjunction with stand-alone firewalls, the host-based firewall provides an additional layer of protection.

Host-Based and Personal Firewalls

- A new type of server can be added to the network, with its own firewall, without the necessity of altering the network firewall configuration.

Host-Based and Personal Firewalls

Personal Firewall:

- A personal firewall controls the traffic between a personal computer or workstation on one side and the Internet or enterprise network on the other side.

Host-Based and Personal Firewalls

- Personal firewall functionality can be used in the home environment and on corporate intranets.
- Typically, the personal firewall is a software module on the personal computer.

Host-Based and Personal Firewalls

- In a home environment with multiple computers connected to the Internet, firewall functionality can also be housed in a router that connects all of home computers to a DSL, or other Internet interface.

Host-Based and Personal Firewalls

- Personal firewalls are typically much less complex than either server-based firewalls or stand-alone firewalls.
- The primary role of the personal firewall is to deny unauthorized remote access to the computer.

Host-Based and Personal Firewalls

- The firewall can also monitor outgoing activity in an attempt to detect and block worms and other malware.

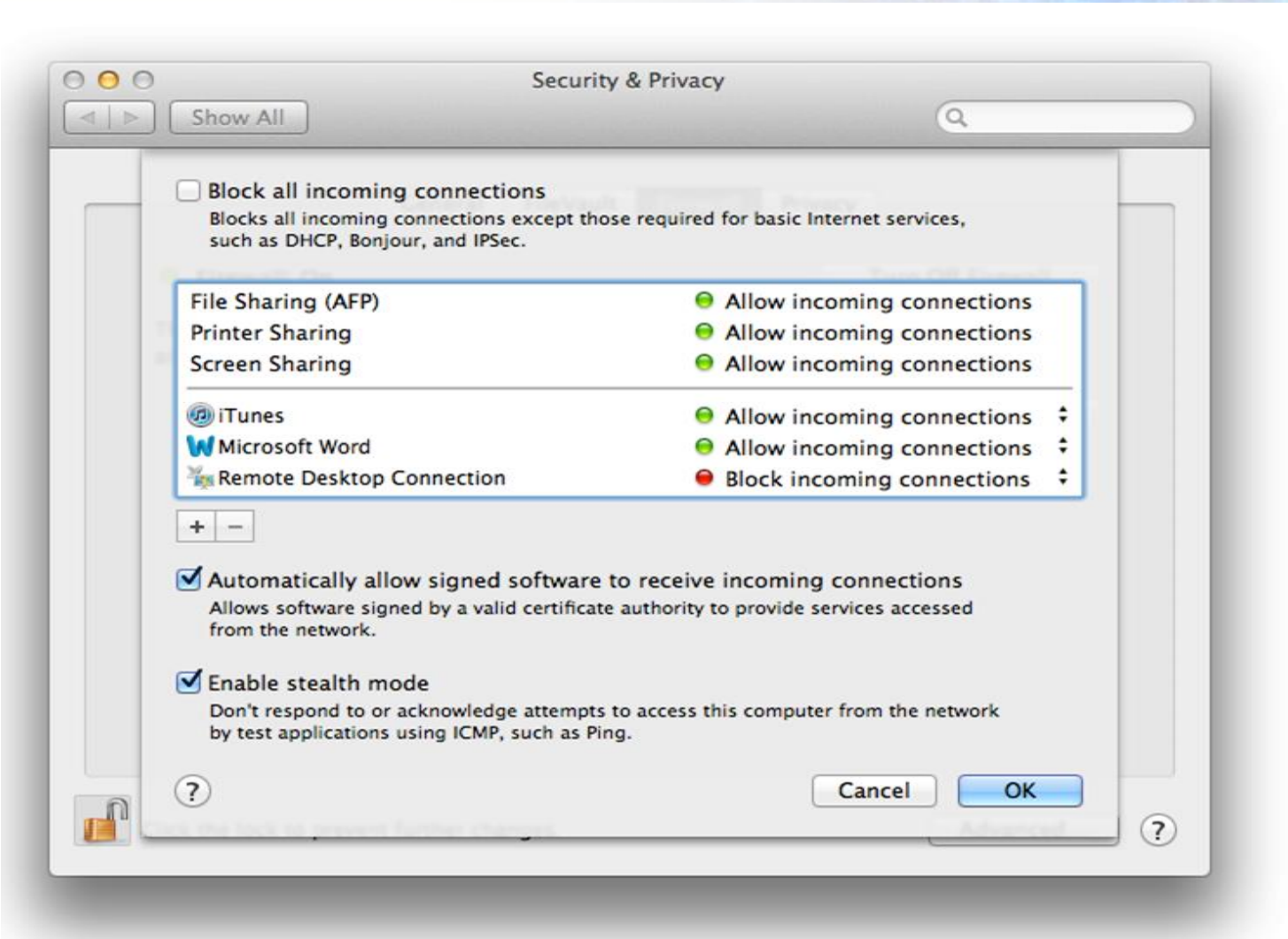
Host-Based and Personal Firewalls

- An example of a personal firewall is the capability built in to the Mac OS X.

Host-Based and Personal Firewalls

- This relatively simple utility can be turned on or off.
- If it is turned on, any applications and services unauthorized by the system won't be allowed to automatically accept incoming network traffic.

Host-Based and Personal Firewalls



Host-Based and Personal Firewalls

- “Block all incoming connections” blocks all sharing services while allowing basic Internet connections.
- If this option is unchecked, then the window below that option lists individual programs and services.

Host-Based and Personal Firewalls

End

- Above horizontal line in window, features of OS X itself are listed.
- These are features selected by the user in another window.
- The user can add applications to a list displayed below line and designate each as blocked or unblocked.

Configure Routers for Syslog, NTP, SSH

Network Security

Configure Routers for Syslog, NTP, SSH

Objectives of the Topic

- After completing this topic, a student will be able to
  - Configure Cisco Routers for Syslog, NTP, and SSH in Packet tracer.

Configure Routers for Syslog, NTP, SSH

Figures and material in this topic have been adapted from

- *Chapter 2: PT Activity: Configure Cisco Routers for Syslog, NTP, and SSH Operations – "CCNA Security 1.1 Student Packet Tracer Manual", Cisco 2012.*

Configure Routers for Syslog, NTP, SSH

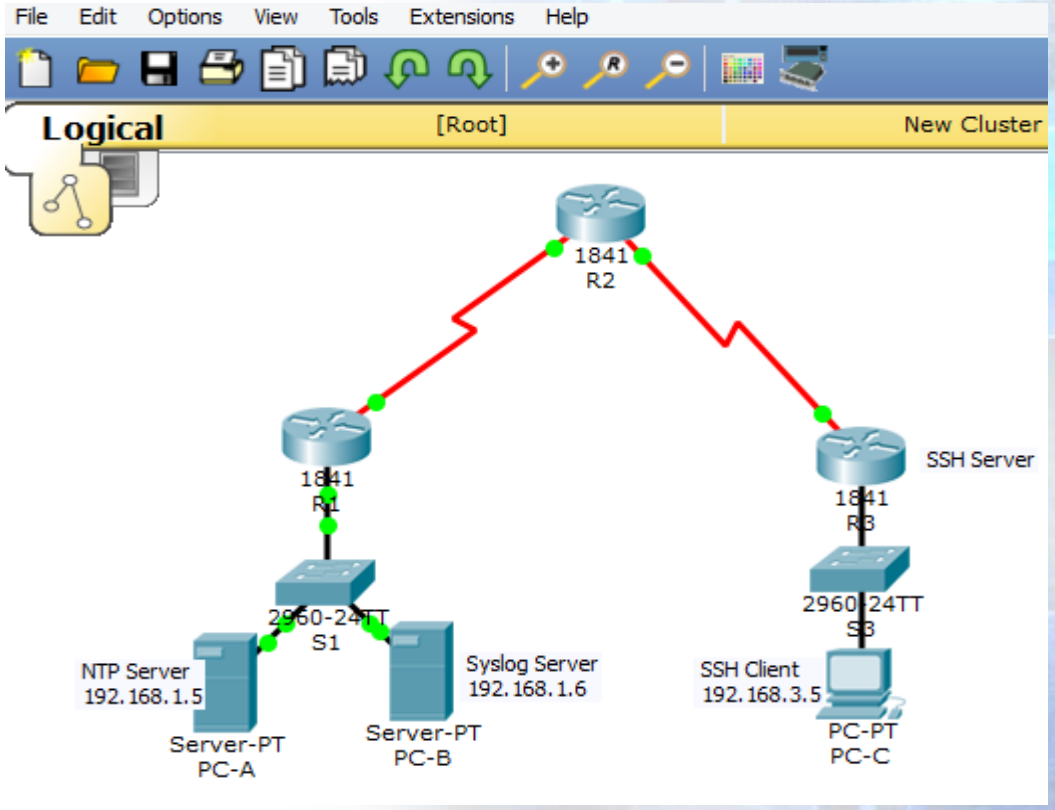
- Routers on a network can get their time settings synchronized with an NTP server via the **Network Time Protocol (NTP)**.
- Obtaining time and date information from a single source will allow a group of NTP clients to have more consistent time settings.
- Routers can update their hardware clock using NTP, timestamp the log messages and send them to the **Syslog server**.

Configure Routers for Syslog, NTP, SSH

- Displaying the correct time and date in **Syslog** messages makes their analysis easier.
- This helps in troubleshooting the network problems and attacks.
- **Secure Shell (SSH)** is used to encrypt information over a network link and is used to authenticate a remote computer.

Configure Routers for Syslog, NTP, SSH

Experimental Setup: Lets assume a network topology consists of three routers R1, R2 and R3.



Configure Routers for Syslog, NTP, SSH

Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway	Switch Port
R1	FA0/1	192.168.1.1	255.255.255.0	N/A	S1 FA0/5
	S0/0/0 (DCE)	10.1.1.1	255.255.255.252	N/A	N/A
R2	S0/0/0	10.1.1.2	255.255.255.252	N/A	N/A
	S0/0/1 (DCE)	10.2.2.2	255.255.255.252	N/A	N/A
R3	FA0/1	192.168.3.1	255.255.255.0	N/A	S3 FA0/5
	S0/0/1	10.2.2.1	255.255.255.252	N/A	N/A
PC-A	NIC	192.168.1.5	255.255.255.0	192.168.1.1	S1 FA0/6
PC-B	NIC	192.168.1.6	255.255.255.0	192.168.1.1	S2 FA0/18
PC-C	NIC	192.168.3.5	255.255.255.0	192.168.3.1	S3 FA0/6

Configure Routers for Syslog, NTP, SSH

- R2 is an ISP and is connected to two remote networks: R1 and R3.
- Since R3 is a managed router, the ISP needs access to R3 for occasional troubleshooting or updates. This access is provided in a secure fashion by using Secure Shell (SSH).
- In this activity, we will configure NTP and Syslog on all routers while SSH will be configured on R3.
- The servers have been pre-configured for NTP and Syslog services.

Configure Routers for Syslog, NTP, SSH

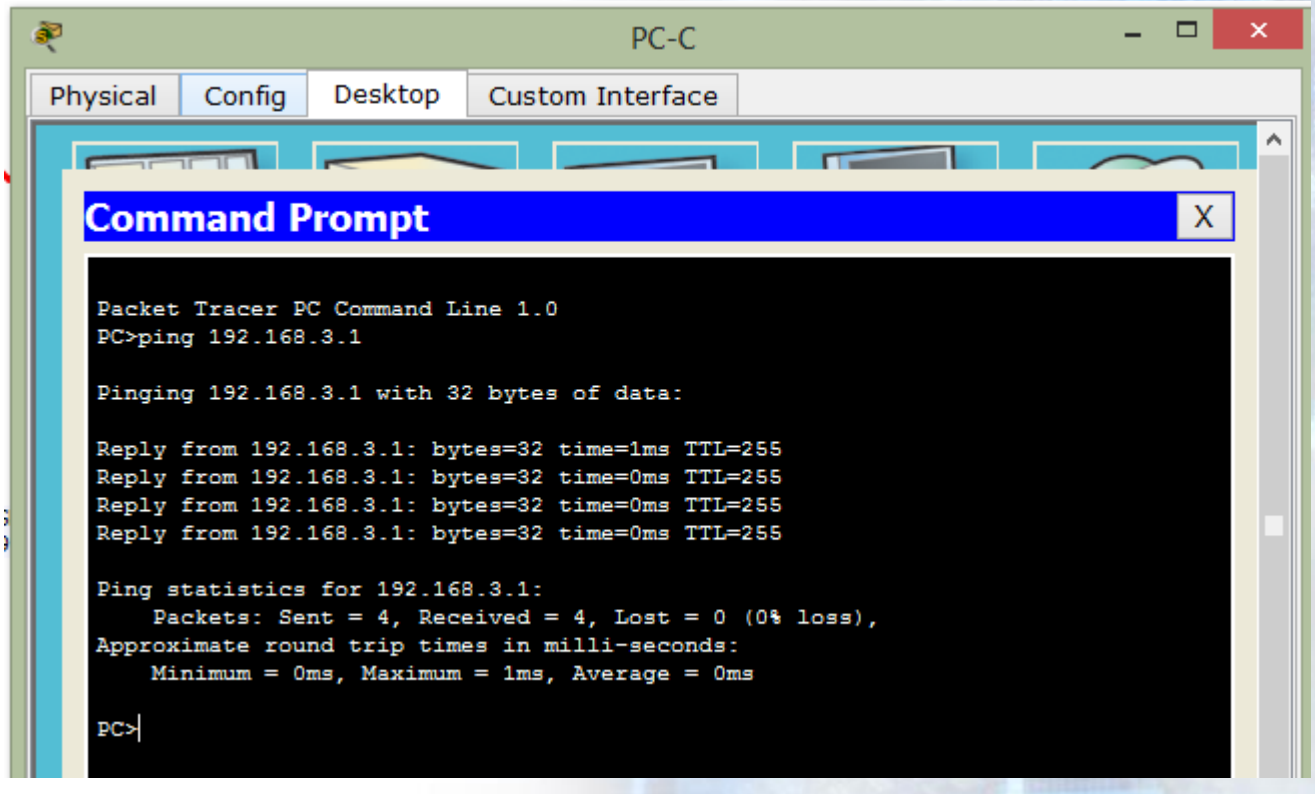
- Furthermore, the routers are pre-configured with the following:
- Enable password: **ciscoenpa55**
- Password for vty lines: **ciscovtypa55**
- Static routing

Configure Routers for Syslog, NTP, SSH

Task 1: Configure routers as NTP Clients.

- **Step 1. Test Connectivity**
- Ping from PC-C to R3.

Configure Routers for Syslog, NTP, SSH

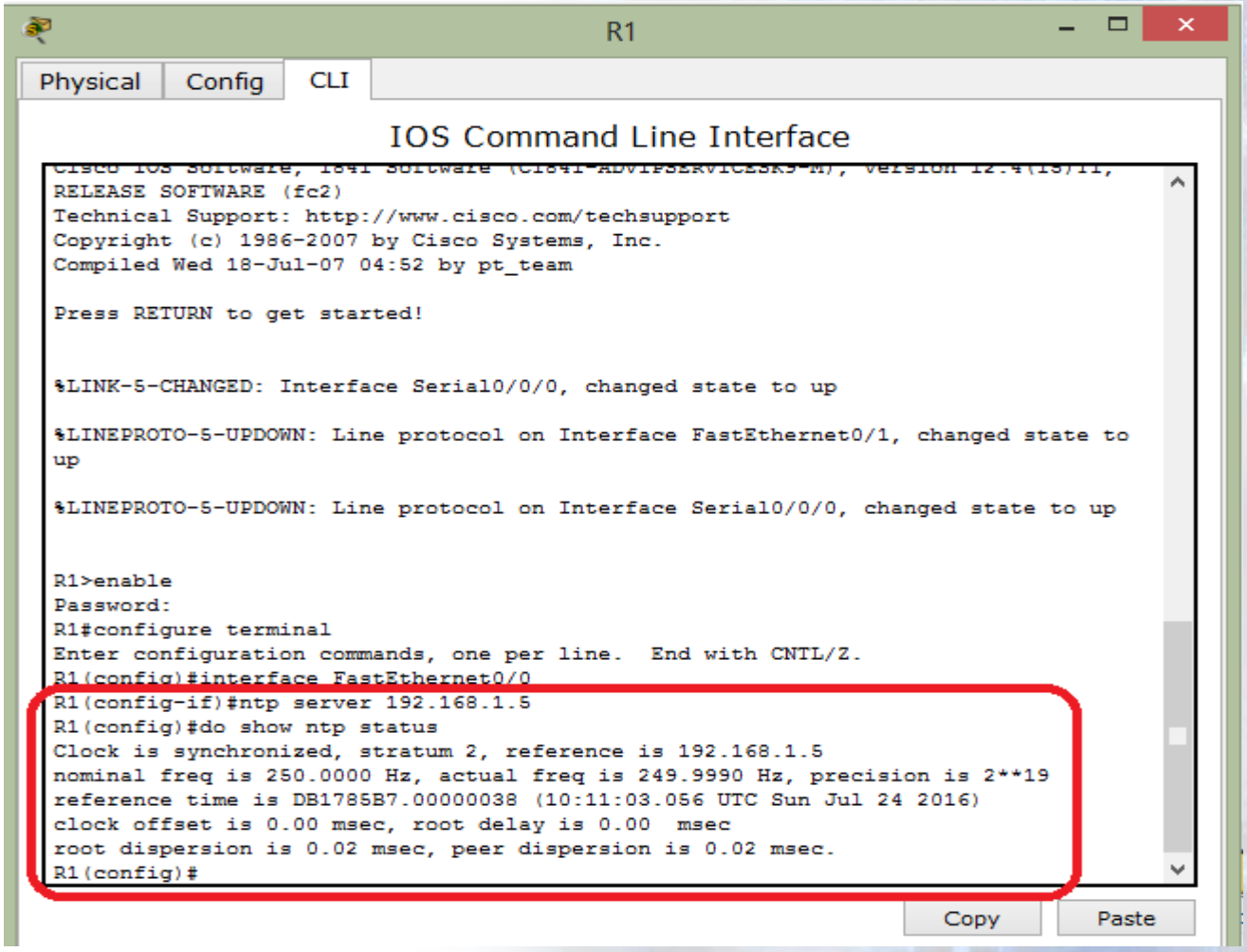


- Ping from R2 to R3. Telnet from PC-C to R3. Exit the Telnet session. Telnet from R2 to R3. Exit the Telnet session.

Configure Routers for Syslog, NTP, SSH

- **Step 2. Configure R1, R2 and R3 as NTP clients.**
- R1(config)# **ntp server 192.168.1.5**
- Use the command ***show ntp status*** to verify client configuration.
- Repeat it for R2 and R3.

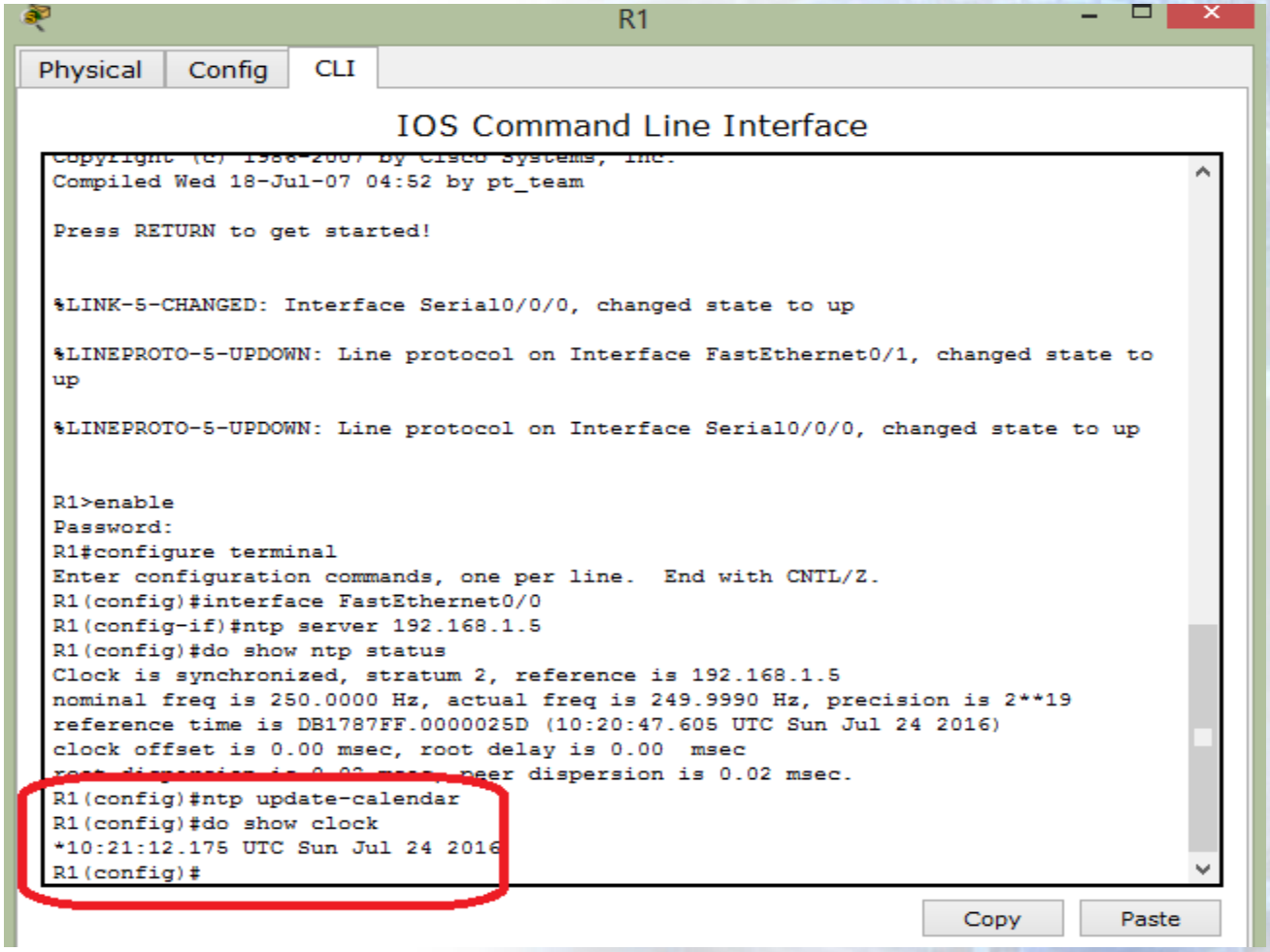
Configure Routers for Syslog, NTP, SSH



Configure Routers for Syslog, NTP, SSH

- **Step 3. Configure routers to update hardware clock.**
- R1(config)# **ntp update-calendar**
- Use the command ***show clock*** to verify that the hardware clock was updated.
- Do the same for R2 and R3.

Configure Routers for Syslog, NTP, SSH



Configure Routers for Syslog, NTP, SSH

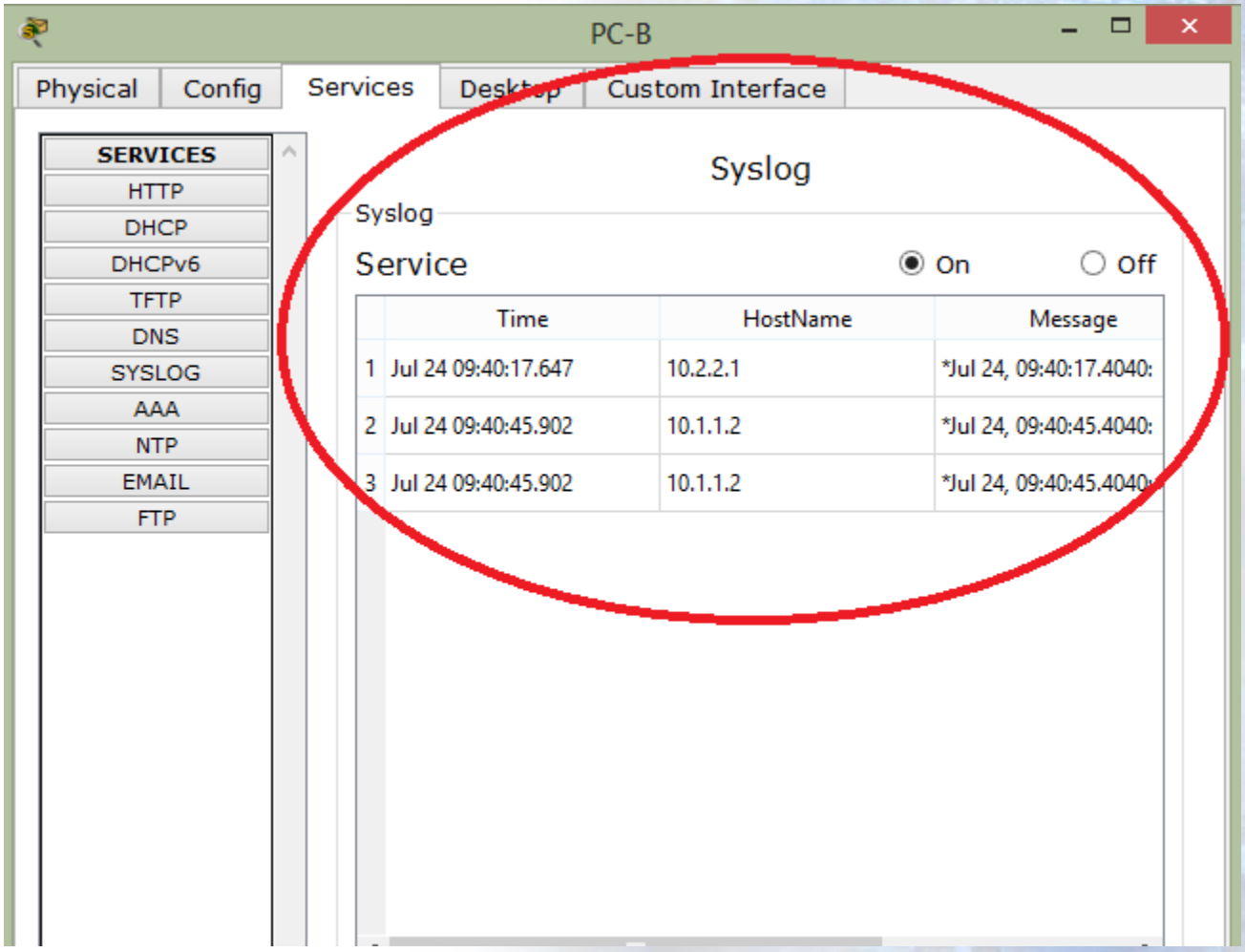
- **Step 4. Configure routers to timestamp log messages.**
- R1(config)# **service timestamps log datetime msec**
- Do the same for R2 and R3

Configure Routers for Syslog, NTP, SSH

Task 2: Configure routers to log messages to the Syslog Server.

- **Step 1. Configure the routers to identify the Syslog server.** Do same for R2, R3.
- R1(config)# **logging host 192.168.1.6**
- **Step 2. Verify logging configuration using the command *show logging*.**
- **Step 3. Examine logs of the Syslog server.**
- From the **Config** tab of the Syslog server's dialogue box, select **Syslog services** button.

Configure Routers for Syslog, NTP, SSH



Configure Routers for Syslog, NTP, SSH

Task 3: Configure R3 to support SSH connections.

- **Step 1. Configure a domain name of ccnasecurity.com on R3.**
- R3(config)# **ip domain-name ccnasecurity.com**
- **Step 2. Configure users for login from the SSH client on R3. user ID = SSHadmin, Password = ciscosshpa55. Highest possible privilege level .**
- R3(config)# **username SSHadmin privilege 15 secret ciscosshpa55**

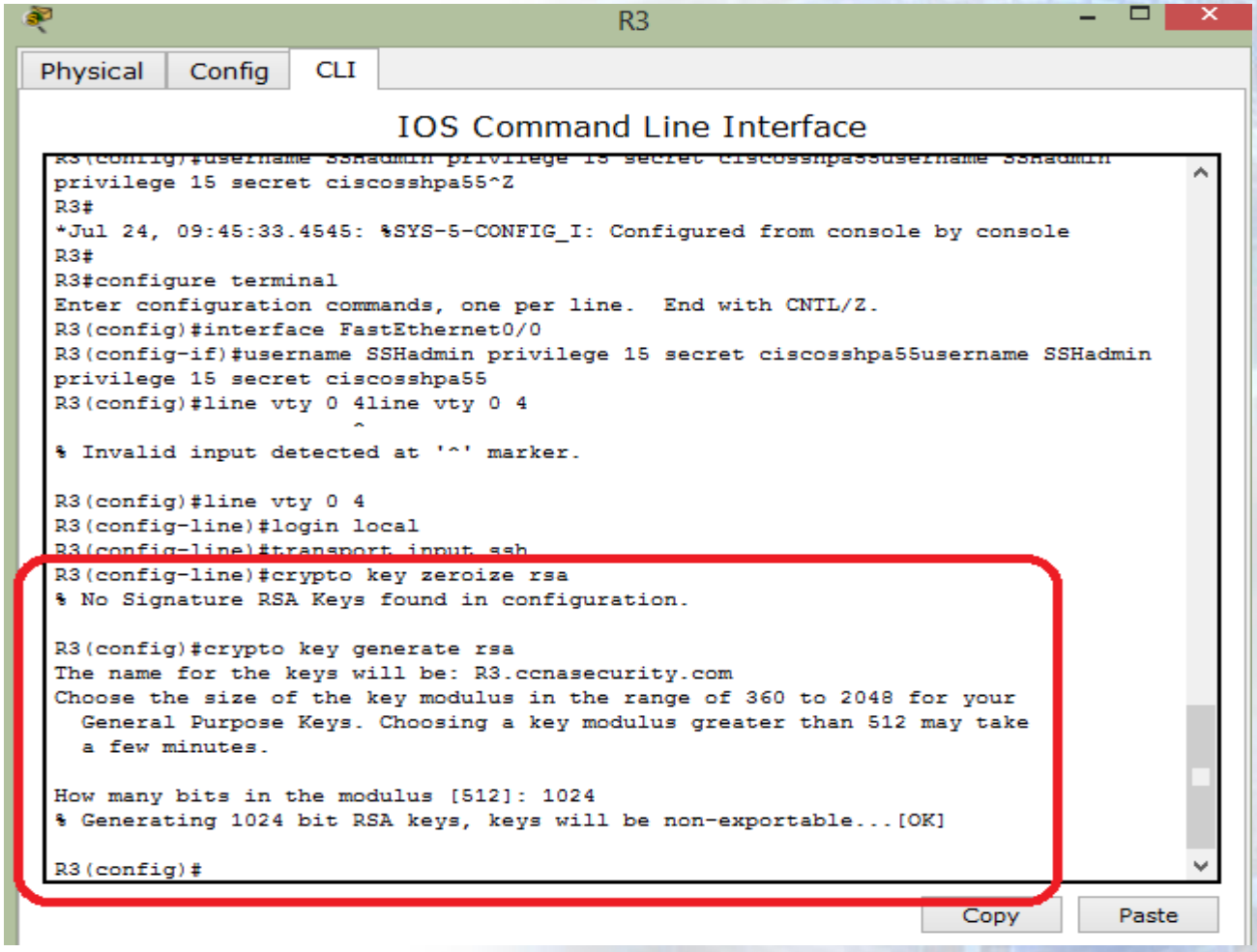
Configure Routers for Syslog, NTP, SSH

- **Step 3. Configure the incoming VTY lines on R3.**
- Use the local user accounts for mandatory login and validation. Accept only SSH connections.
- R3(config)# **line vty 0 4**
- R3(config-line)# **login local**
- R3(config-line)# **transport input ssh**

Configure Routers for Syslog, NTP, SSH

- **Step 4. Erase existing key pairs on R3.**
- R3(config)#**crypto key zeroize rsa**
- % No Signature RSA Keys found in configuration.
- **Step 5. Generate the RSA encryption key pair for R3.**
- The router uses the RSA key pair for authentication and encryption of transmitted SSH data. Configure the RSA keys with a modulus of **1024**. The default is 512, and the range is from 360 to 2048.

Configure Routers for Syslog, NTP, SSH



Configure Routers for Syslog, NTP, SSH

- **Step 6. Verify the SSH configuration.**
- Use the ***show ip ssh*** command to see the current settings.
- Verify that the default values of authentication timeout and retries are 120 and 3.

Configure Routers for Syslog, NTP, SSH

- **Step 7. Configure SSH timeouts and authentication parameters.**
- Set the timeout to **90** seconds, the number of authentication retries to **2**, and the version to **2**.
- R3(config)# **ip ssh time-out 90**
- R3(config)# **ip ssh authentication-retries 2**
- R3(config)# **ip ssh version 2**
- To confirm that the values have been updated, issue the ***show ip ssh*** command.

Configure Routers for Syslog, NTP, SSH

- **Step 8. Attempt to connect to R3 via Telnet from PC-C.**
- From the Command Prompt icon of PC-C, enter
- PC> **telnet 192.168.3.1**
- As R3 has been configured to accept only SSH connections, this connection should fail.
- **Step 9. Connect to R3 using SSH on PC-C.**
- PC> **ssh –l SSHadmin 192.168.3.1**
- When prompted for the password, enter **ciscosshpa55**.

Configure Routers for Syslog, NTP, SSH

- **Step 10. Connect to R3 using SSH on R2.**
- In order to troubleshoot and maintain the R3 router, the administrator at the ISP must use SSH to access the router CLI.
- R2# **ssh –v 2 –l SSHadmin 10.2.2.1**
- When prompted for the password, enter the password configured for the administrator: **ciscosshpa55**.

Configure Routers for Syslog, NTP, SSH

End

- **Step 11. Check results.**
- Your completion percentage should be 100%.
- Click **Check Results** to see feedback and verification of which required components have been completed.

Configure AAA Authentication on Routers

Network Security

Configure AAA Authentication on Routers

Objectives of the Topic

- After completing this topic, a student will be able to
  - Configure AAA Authentication on Cisco Routers in Packet tracer.

Configure AAA Authentication on Routers

Figures and material in this topic have been adapted from

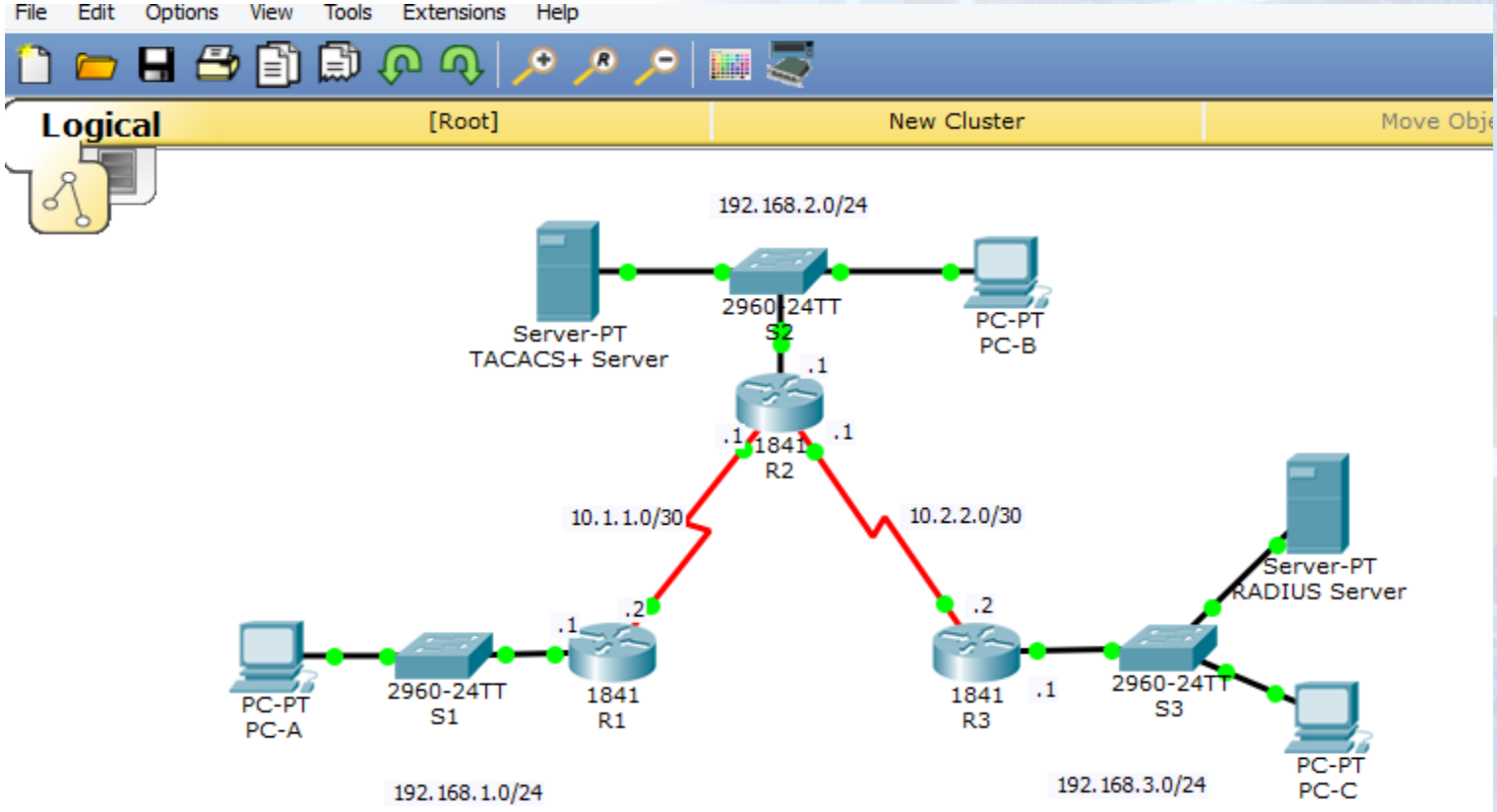
- *Chapter 3: PT Activity: Configure AAA Authentication on Cisco Routers – "CCNA Security 1.1 Student Packet Tracer Manual", Cisco 2012.*

Configure AAA Authentication on Routers

- "**Authentication** asks the question, "Who or what are you?" **Authorization** asks, "What are you allowed to do?" And finally, **accounting** wants to know, "What did you do?""
- Remote Authentication Dial-In User Service (RADIUS) and Terminal Access Controller Access Control System (TACACS)– most widely accepted AAA protocol. RADIUS is an Internet standard.
- ***Sean Convery, "Network Authentication, Authorization, and Accounting: Part One"*** **- The Internet Protocol Journal Volume 10, No. 1, 2007**

Configure AAA Authentication on Routers

Experimental Setup: Lets assume a network topology consists of three routers R1, R2 and R3.



Configure AAA Authentication on Routers

Addressing Table

Device	Interface	IP Address	Subnet Mask
R1	Fa0/0	192.168.1.1	255.255.255.0
	S0/0/0	10.1.1.2	255.255.255.252
R2	S0/0/0	10.1.1.1	255.255.255.252
	Fa0/0	192.168.2.1	255.255.255.0
	S0/0/1	10.2.2.1	255.255.255.252
R3	S0/0/1	10.2.2.2	255.255.255.252
	Fa0/0	192.168.3.1	255.255.255.0
TACACS+ Server	NIC	192.168.2.2	255.255.255.0
RADIUS Server	NIC	192.168.3.2	255.255.255.0
PC-A	NIC	192.168.1.3	255.255.255.0
PC-B	NIC	192.168.2.3	255.255.255.0
PC-C	NIC	192.168.3.3	255.255.255.0

Configure AAA Authentication on Routers

- The task is to configure and test local and server-based AAA solutions.
- We will create a local user account and configure local AAA on router R1 to test the console and VTY logins.
- User account: **Admin1** and password **admin1pa55**
- Then, we will configure router R2 to support server-based authentication using the TACACS+ protocol.

Configure AAA Authentication on Routers

- The TACACS+ server has been pre-configured with the following:
- Client: **R2** using the keyword **tacacspa55**
- User account: **Admin2** and password **admin2pa55**
- Finally, we will configure router R3 to support server-based authentication using the RADIUS protocol.
- The RADIUS server has been pre-configured with the following:

Configure AAA Authentication on Routers

- Client: **R3** using the keyword **radiuspa55**
- User account: **Admin3** and password **admin3pa55**
- The routers have also been pre-configured with the following:
- Enable secret password: **ciscoenpa55**
- RIP version 2

Configure AAA Authentication on Routers

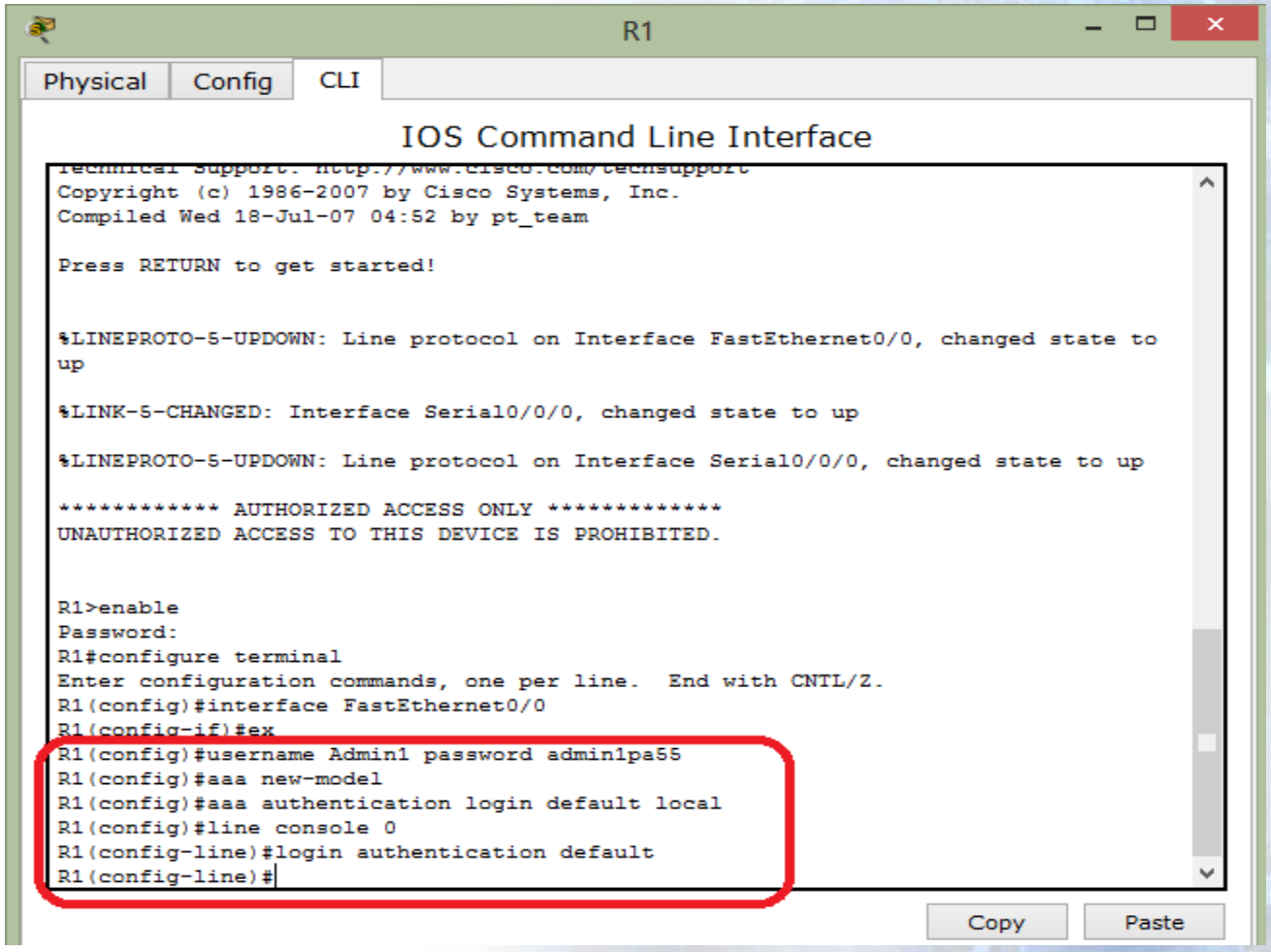
Task 1: Configure Local AAA Authentication for Console Access on R1

- **Step 1. Test connectivity.**
- **Ping** from PC-A to PC-B. **Ping** from PC-A to PC-C. **Ping** from PC-B to PC-C.
- **Step 2. Configure a local username on R1.**
- R1(config)# **username Admin1 password admin1pa55**
- **Step 3. Configure local AAA authentication for console access on R1.**

Configure AAA Authentication on Routers

- Enable AAA on R1 and configure AAA authentication for console login to use the local database.
- R1(config)# **aaa new-model**
- R1(config)# **aaa authentication login default local**
- **Step 4. Configure the line console to use the defined AAA authentication method.**
- R1(config)# **line console 0**
- R1(config-line)# **login authentication default**

Configure AAA Authentication on Routers



Configure AAA Authentication on Routers

- **Step 5. Verify the AAA authentication method.**

```
R1(config-line)# end
%SYS-5-CONFIG_I: Configured from console by console
R1# exit

R1 con0 is now available
Press RETURN to get started.

************ AUTHORIZED ACCESS ONLY ************
UNAUTHORIZED ACCESS TO THIS DEVICE IS PROHIBITED.


User Access Verification

Username: Admin1
Password: admin1pa55
R1>
```

Configure AAA Authentication on Routers

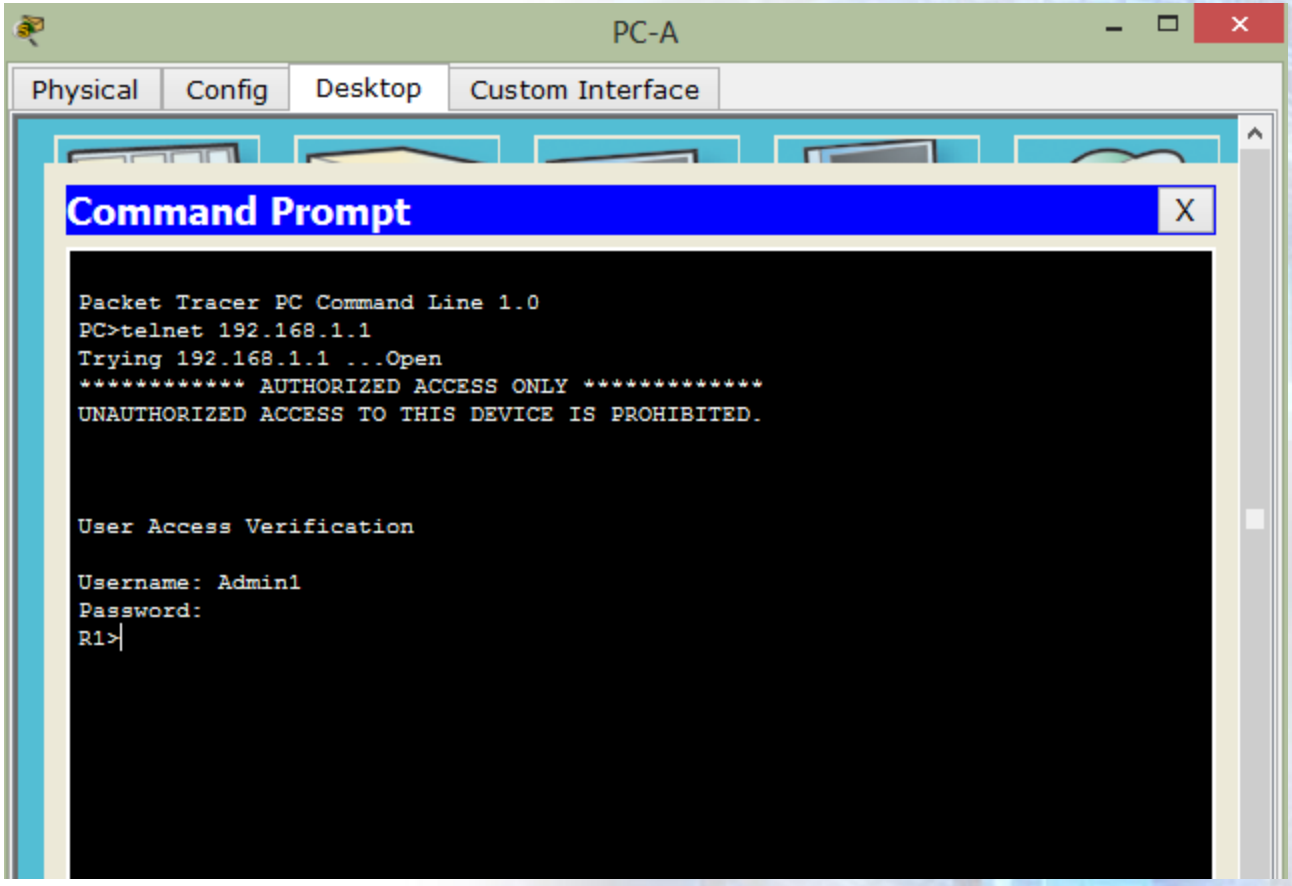
Task 2: Configure Local AAA Authentication for VTY Lines on R1

- **Step 1. Configure a named list AAA authentication method for VTY lines on R1.**
- Configure a named list called **TELNET-LOGIN** to authenticate logins using local AAA.
- R1(config)# **aaa authentication login TELNET-LOGIN local**

Configure AAA Authentication on Routers

- **Step 2. Configure the VTY lines to use the defined AAA authentication method.**
- R1(config)# **line vty 0 4**
- R1(config-line)# **login authentication TELNET-LOGIN**
- R1(config-line)# **end**
- **Step 3. Verify the AAA authentication method.**
- Verify the Telnet configuration.
- From command prompt of PC-A, Telnet to R1.
  PC> **telnet 192.168.1.1**

Configure AAA Authentication on Routers



Configure AAA Authentication on Routers

Task 3: Configure Server-Based AAA Authentication Using TACACS+ on R2

- **Step 1. Configure a backup local database entry called Admin.**
- For backup purposes, configure a local username of Admin and secret password of adminpa55.
- R2(config)# **username Admin password adminpa55**

Configure AAA Authentication on Routers

- **Step 2. Verify the TACACS+ Server configuration.**
- Select the TACACS+ Server. From the **Config** tab, click on **AAA** and notice that there is a Network configuration entry for R2 and a User Setup entry for Admin2.
- **Step 3. Configure the TACACS+ server specifics on R2.**
- R2(config)# **tacacs-server host 192.168.2.2**
- R2(config)# **tacacs-server key tacacspa55**

Configure AAA Authentication on Routers

- **Step 4. Configure AAA login authentication for console access on R2.**
- Enable AAA on R2 and configure all logins to authenticate using the AAA TACACS+ server and if not available, then use the local database.
- R2(config)# **aaa new-model**
- R2(config)# **aaa authentication login default group tacacs+ local**

Configure AAA Authentication on Routers

- **Step 5. Configure the line console to use the defined AAA authentication method.**
- R2(config)# **line console 0**
- R2(config-line)# **login authentication default**

Configure AAA Authentication on Routers

- **Step 6. Verify the AAA authentication method.**
- Verify the user EXEC login using the AAA TACACS+ server.

```
R2(config-line)# end
%SYS-5-CONFIG_I: Configured from console by console
R2# exit

R2 con0 is now available
Press RETURN to get started.

************ AUTHORIZED ACCESS ONLY *************
UNAUTHORIZED ACCESS TO THIS DEVICE IS PROHIBITED.


User Access Verification

Username: Admin2
Password: admin2pa55
R2>
```

Configure AAA Authentication on Routers

Task 4: Configure Server-Based AAA Authentication Using RADIUS on R3

- **Step 1. Configure a backup local database entry called Admin.**
- R3(config)# **username Admin password adminpa55**
- **Step 2. Verify RADIUS configuration.**
- Select the RADIUS Server.

Configure AAA Authentication on Routers

- From the Config tab, click on **AAA** and notice that there is a Network configuration entry for R3 and a User Setup entry for Admin3.
- **Step 3. Configure the RADIUS server specifics on R3.**
- Configure the AAA RADIUS server IP address and secret key on R3.
- R3(config)# **radius-server host 192.168.3.2**
- R3(config)# **radius-server key radiuspa55**

Configure AAA Authentication on Routers

- **Step 4. Configure AAA login authentication for console access on R3.**
- R3(config)# **aaa new-model**
- R3(config)# **aaa authentication login default group radius local**
- **Step 5. Configure the line console to use defined AAA authentication method.**
- R3(config)# **line console 0**
- R3(config-line)# **login authentication default**

Configure AAA Authentication on Routers

- **Step 6. Verify the AAA authentication method.**
- Verify the user EXEC login using the AAA TACACS+ server.

```
R3(config-line)# end
%SYS-5-CONFIG_I: Configured from console by console
R3# exit

R3 con0 is now available
Press RETURN to get started.

************ AUTHORIZED ACCESS ONLY *************
UNAUTHORIZED ACCESS TO THIS DEVICE IS PROHIBITED.


User Access Verification

Username: Admin3
Password: admin3pa55
R3>
```

Configure AAA Authentication on Routers

End

- **Step 7. Check results.**
- Your completion percentage should be 100%.
- Click **Check Results** to see feedback and verification of which required components have been completed.

Configure IP ACLs to Mitigate Attacks

Network Security

Configure IP ACLs to Mitigate Attacks

Objectives of the Topic

- After completing this topic, a student will be able to
  - Configure IP ACLs to Mitigate Attacks in Packet tracer.

Configure IP ACLs to Mitigate Attacks

Figures and material in this topic have been adapted from

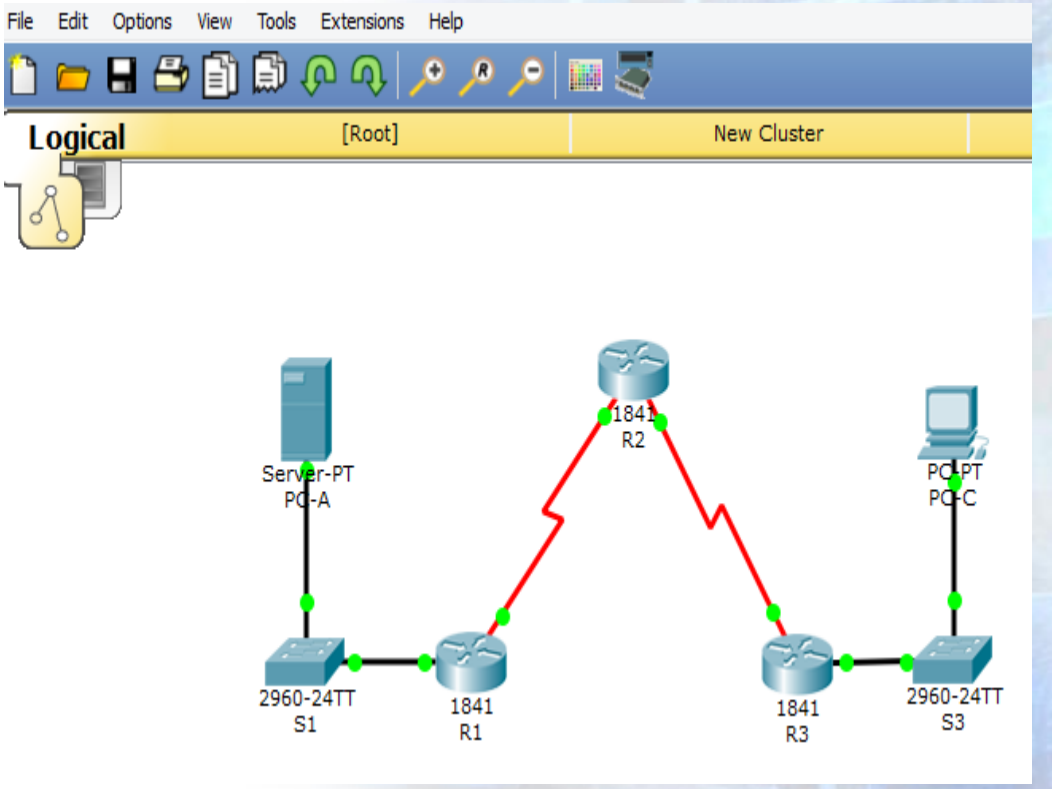
- *Chapter 4: PT Activity: Configure IP ACLs to Mitigate Attacks – "CCNA Security 1.1 Student Packet Tracer Manual", Cisco 2012.*

Configure IP ACLs to Mitigate Attacks

- "Cisco provides basic traffic filtering capabilities with access control lists (ACL). Access lists filter network traffic by controlling whether routed packets are forwarded or blocked at the router's interfaces. Your router examines each packet to determine whether to forward or drop the packet, on the basis of the criteria you specified within the access lists: source address, destination address of the traffic, the upper-layer protocol, or other information".
- ***"Cisco IOS Security Configuration Guide"*** **– Cisco Systems, Inc, 2006**

Configure IP ACLs to Mitigate Attacks

Experimental Setup: Lets assume a network topology consists of three routers R1, R2 and R3.



Configure IP ACLs to Mitigate Attacks

Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	Fa0/1	192.168.1.1	255.255.255.0	N/A
	S0/0/0 (DCE)	10.1.1.1	255.255.255.252	N/A
R2	S0/0/0	10.1.1.2	255.255.255.252	N/A
	S0/0/1(DCE)	10.2.2.2	255.255.255.252	N/A
	Lo0	192.168.2.1	255.255.255.0	N/A
R3	Fa0/1	192.168.3.1	255.255.255.0	N/A
	S0/0/1	10.2.2.1	255.255.255.252	N/A
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1
PC-C	NIC	192.168.3.3	255.255.255.0	192.168.3.1

Configure IP ACLs to Mitigate Attacks

- Access to routers R1, R2, and R3 should only be permitted from PC-C, the management station.
- PC-C is also used for connectivity testing to PC-A, a server providing DNS, SMTP, FTP, and HTTPS services.
- The standard procedure is to apply ACLs on edge routers to mitigate common threats based on source and/or destination IP address.
- We will create ACLs on edge routers R1 and R3 to achieve this goal and then verify ACL functionality from internal and external hosts.

Configure IP ACLs to Mitigate Attacks

- The routers have been pre-configured with:
- Enable password: **ciscoenpa55**
- Password for console: **ciscoconpa55**
- Username for VTY lines: **SSHadmin**
- Password for VTY lines: **ciscosshpa55**
- IP addressing
- Static routing

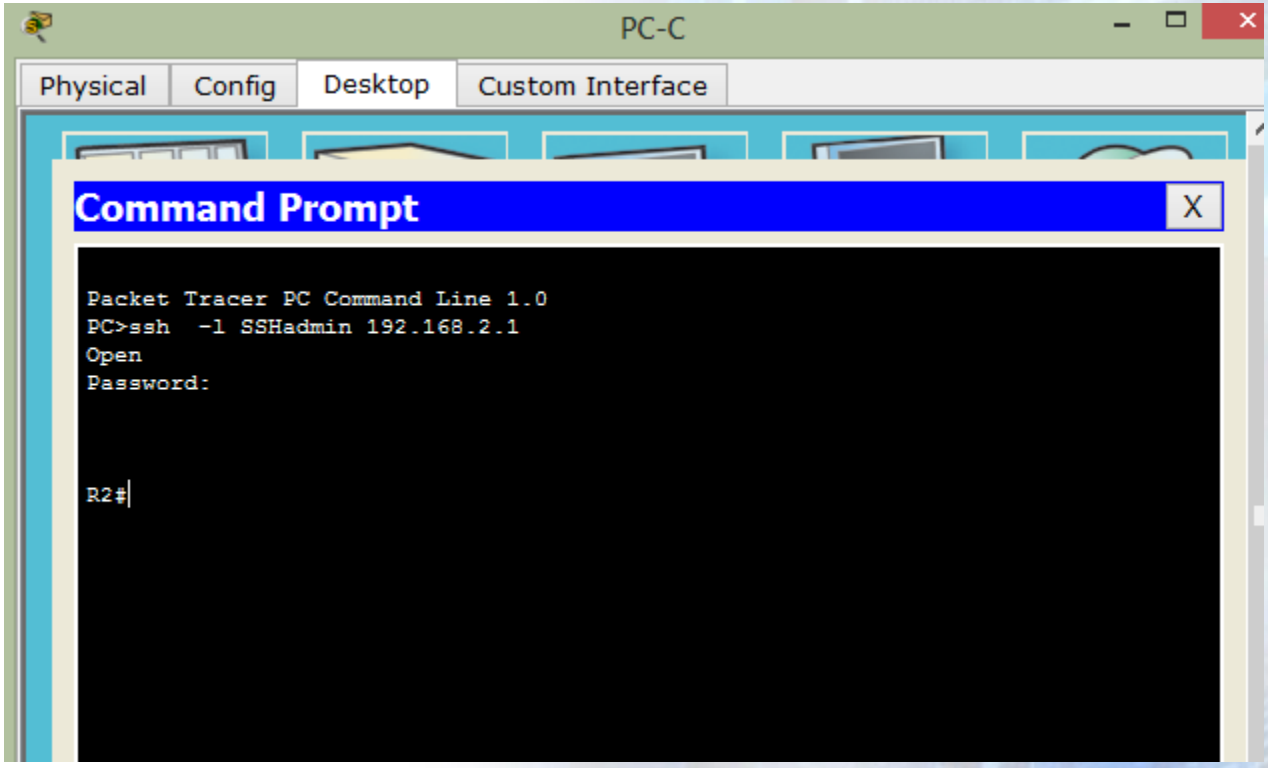
Configure IP ACLs to Mitigate Attacks

Task 1: Verify Basic Network Connectivity

- Step 1. From the PC-C command prompt, ping the PC-A server.
- Step 2. From the PC-C command prompt, SSH to the router R2 Lo0 interface. Exit the SSH session.

Configure IP ACLs to Mitigate Attacks

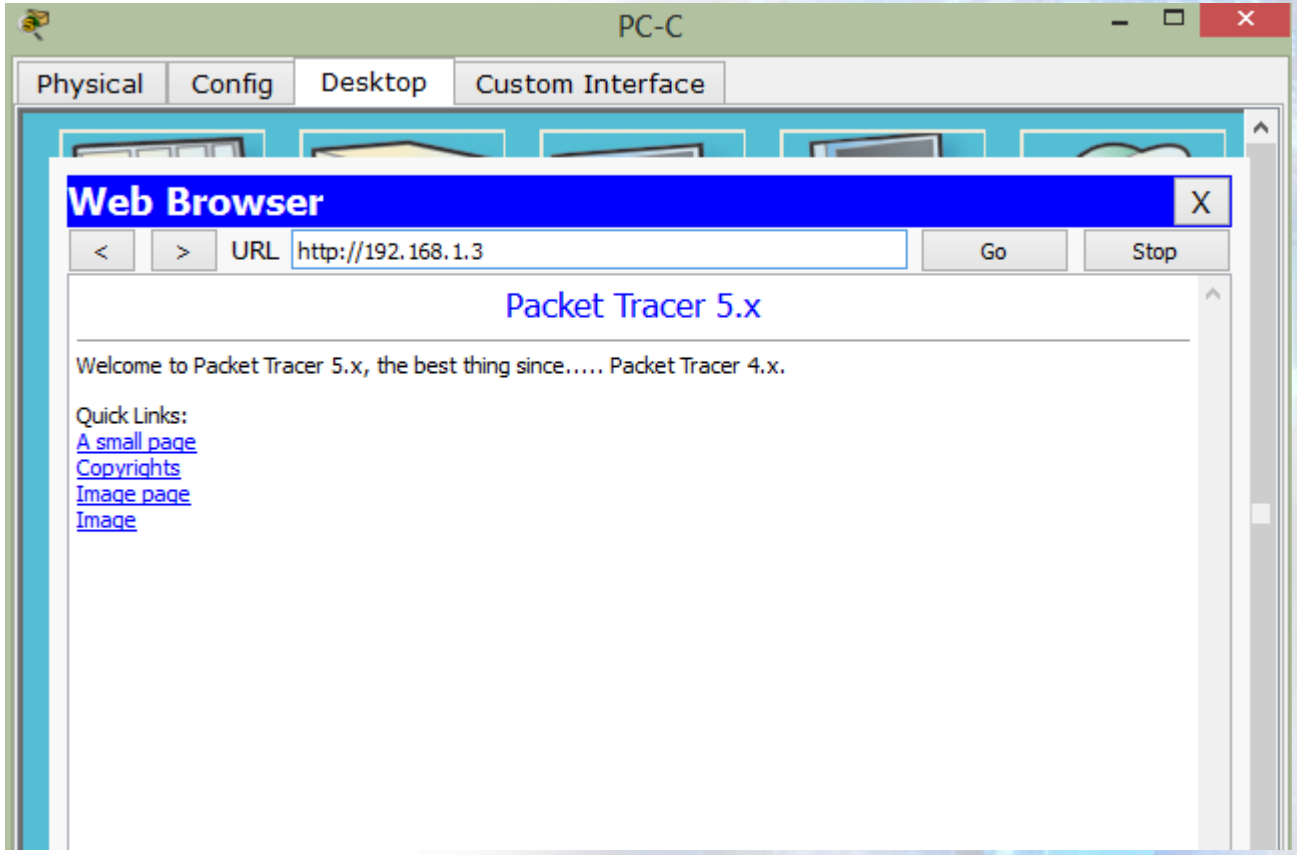
- PC> **ssh –l SSHadmin 192.168.2.1**
- **Password: ciscosshpa55**



Configure IP ACLs to Mitigate Attacks

- **Step 3. From PC-C, open a web browser to the PC-A server (using the IP address) to display the web page. Close the browser on PC-C .**
- **Step 4. From the PC-A server command prompt, ping PC-C .**

Configure IP ACLs to Mitigate Attacks



Configure IP ACLs to Mitigate Attacks

Task 2: Secure Access to Routers

- **Step 1. Configure ACL 10 to block all remote access to the routers except from PC-C.**
- Use the **access-list** command to create a numbered IP ACL on R1, R2, and R3.
- R1(config)# **access-list 10 permit 192.168.3.3 0.0.0.0**
- Do the same for R2 and R3.
- **Step 2. Apply ACL 10 to ingress traffic on the VTY lines .**

Configure IP ACLs to Mitigate Attacks

- Use the **access-class** command to apply the access list to incoming traffic on the VTY lines.
- R1(config-line)# **access-class 10 in**
- Repeat for R2 and R3.
- **Step 3. Verify exclusive access from management station PC -C.**
- SSH to 192.168.2.1 from PC-C (should be successful). SSH to 192.168.2.1 from PC-A (should fail).
- PC> **ssh –l SSHadmin 192.168.2.1**

Configure IP ACLs to Mitigate Attacks

Task 3: Create a Numbered IP ACL 100 :

- On R3, block all packets containing the source IP address from pool of addresses: 127.0.0.0/8, any RFC 1918 private addresses, and any IP multicast address.
- **Step 1. Configure ACL 100 to block all specified traffic from the outside network.**
- You should also block traffic sourced from your own internal address space if it is not an RFC 1918 address.

Configure IP ACLs to Mitigate Attacks

- R3(config)# **access-list 100 deny ip 10.0.0.0 0.255.255.255 any**
- R3(config)# **access-list 100 deny ip 172.16.0.0 0.15.255.255 any**
- R3(config)# **access-list 100 deny ip 192.168.0.0 0.0.255.255 any**
- R3(config)# **access-list 100 deny ip 127.0.0.0 0.255.255.255 any**
- R3(config)# **access-list 100 deny ip 224.0.0.0 15.255.255.255 any**
- R3(config)# **access-list 100 permit ip any any**

Configure IP ACLs to Mitigate Attacks

- **Step 2. Apply the ACL to interface Serial 0/0/1.**
- Use **ip access-group** command to apply access list to incoming traffic on interface Serial 0/0/1.
- R3(config)# **interface s0/0/1**
- R3(config-if)# **ip access-group 100 in**
- **Step 3. Confirm that the specified traffic entering interface Serial 0/0/1 is dropped.**
- From the PC-C command prompt, ping the PC-A server. The ICMP echo *replies* are blocked by the ACL.

Configure IP ACLs to Mitigate Attacks

- **Step 4. Remove the ACL from interface Serial 0/0/1.**
- Remove ACL to allow traffic from the outside network.
- Use the **no ip access-group** command to remove the access list from interface Serial 0/0/1.
- R3(config)# **interface s0/0/1**
- R3(config-if)# **no ip access-group 100 in**

Configure IP ACLs to Mitigate Attacks

Task 4: Create a Numbered IP ACL 110

- Deny all outbound packets with source address outside the range of internal IP addresses.
- **Step 1. Configure ACL 110 to permit only traffic from the inside network.**
- R3(config)# **access-list 110 permit ip 192.168.3.0 0.0.0.255 any**
- **Step 2. Apply the ACL to interface F0/1.**
- R3(config)# **interface fa0/1**
- R3(config-if)# **ip access-group 110 in**

Configure IP ACLs to Mitigate Attacks

Task 5: Create a Numbered IP ACL 120:

- Permit any outside host to access DNS, SMTP, and FTP services on server PC-A, deny any outside host access to HTTPS services on PC-A, and permit PC-C to access R1 via SSH.
- **Step 1. Verify that PC-C can access the PC-A via HTTPS using the web browser.**
- Disable HTTP and enable HTTPS on server PC-A.
- **Step 2. Configure ACL 120 to specifically permit and deny the specified traffic.**

Configure IP ACLs to Mitigate Attacks

- R1(config)# **access-list 120 permit udp any host 192.168.1.3 eq domain**
- R1(config)# **access-list 120 permit tcp any host 192.168.1.3 eq smtp**
- R1(config)# **access-list 120 permit tcp any host 192.168.1.3 eq ftp**
- R1(config)# **access-list 120 deny tcp any host 192.168.1.3 eq 443**
- R1(config)# **access-list 120 permit tcp host 192.168.3.3 host 10.1.1.1 eq 22**

Configure IP ACLs to Mitigate Attacks

- **Step 3. Apply the ACL to interface S0/0/0.**
- R1(config)# **interface s0/0/0**
- R1(config-if)# **ip access-group 120 in**
- **Step 4. Verify that PC-C cannot access PC-A via HTTP using the web browser.**

Task 6: Modify An Existing ACL:

- Permit ICMP echo replies and destination unreachable messages from the outside network (relative to R1); deny all other incoming ICMP packets.

Configure IP ACLs to Mitigate Attacks

- **Step 1. Verify that PC-A cannot success fully ping the loopback interface on R2.**
- **Step 2. Make any necessary changes to ACL 120 to permit and deny the specified traffic.**
- R1(config)# **access-list 120 permit icmp any any echo-reply**
- R1(config)# **access-list 120 permit icmp any any unreachable**
- R1(config)# **access-list 120 deny icmp any any**
- R1(config)# **access-list 120 permit ip any any**

Configure IP ACLs to Mitigate Attacks

End

- **Step 3. Verify that PC-A can successfully ping the loopback interface on R2.**
- **Step 4. Check results .**
- Your completion percentage should be 100%.
- Click **Check Results** to see feedback.

Configure Context-Based Access Control

Network Security

Configure Context-Based Access Control

Objectives of the Topic

- After completing this topic, a student will be able to
  - Configure Context-Based Access Control in Packet tracer.

Configure Context-Based Access Control

Figures and material in this topic have been adapted from

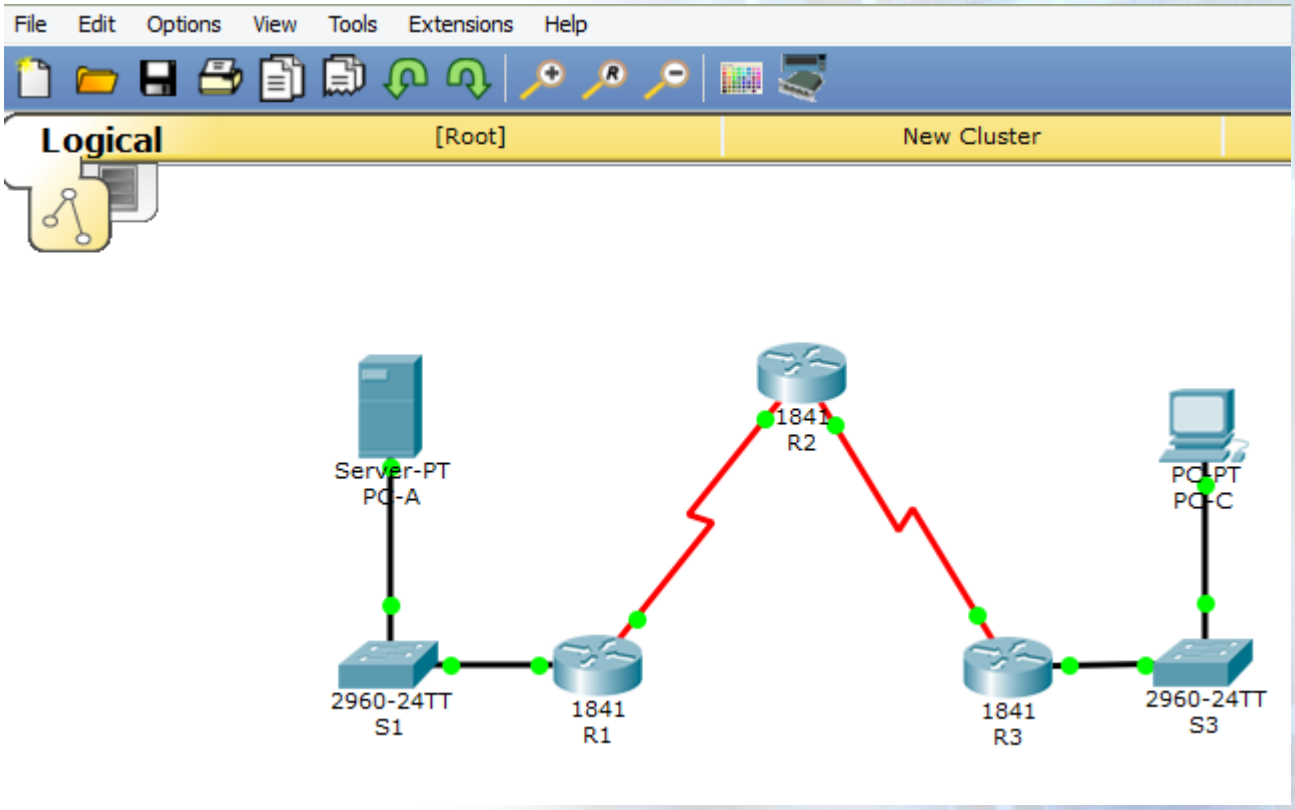
- *Chapter 4: PT Activity: Configuring Context-Based Access Control (CBAC) – "CCNA Security 1.1 Student Packet Tracer Manual", Cisco 2012.*

Configure Context-Based Access Control

- "Context-Based Access Control (CBAC) provides advanced traffic filtering functionality and can be used as an integral part of your network's firewall. CBAC works to provide network protection on multiple levels using the following functions: Traffic Filtering, Traffic Inspection, Alerts and Audit Trails, and Intrusion Detection"
- ***"Cisco IOS Security Configuration Guide"*** **– Cisco Systems, Inc, 2006**

Configure Context-Based Access Control

Experimental Setup: Lets assume a network topology consists of three routers R1, R2 and R3.



Configure Context-Based Access Control

Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	Fa0/1	192.168.1.1	255.255.255.0	N/A
	S0/0/0	10.1.1.1	255.255.255.252	N/A
R2	S0/0/0	10.1.1.2	255.255.255.252	N/A
	S0/0/1	10.2.2.2	255.255.255.252	N/A
R3	Fa0/1	192.168.3.1	255.255.255.0	N/A
	S0/0/1	10.2.2.1	255.255.255.252	N/A
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1
PC-C	NIC	192.168.3.3	255.255.255.0	192.168.3.1

Configure Context-Based Access Control

- Context-Based Access Control (CBAC) is used to create an IOS firewall.
- In this activity, we will create a basic CBAC configuration on edge router R3 that provides access to resources outside of the network for hosts on the inside network.
- R3 blocks external hosts from accessing internal resources.

Configure Context-Based Access Control

- The routers have been pre-configured:
- Enable password: **ciscoenpa55**
- Password for console: **ciscoconpa55**
- Password for vty lines: **ciscovtypa55**
- IP addressing
- Static routing
- All switch ports are in VLAN 1 for switches S1 and S3.

Configure Context-Based Access Control

Task 1: Block Traffic From Outside

- **Step 1. Verify Basic Network Connectivity.**
- From the PC-C command prompt, ping the PC-A server. From the PC-C command prompt, Telnet to the Router R2 S0/0/1 interface: IP address 10.2.2.2. Exit the Telnet session.
- From PC-C, open a web browser to the PC-A server to display the web page. Close the browser on PC-C.
- From PC-A server command prompt, ping PC-C.

Configure Context-Based Access Control

- **Step 2. Configure a named IP ACL on R3 to block all traffic originating from the outside network.**
- Use the **ip access-list extended** command to create a named IP ACL.
- R3(config)# **ip access-list extended OUT-IN**
- R3(config-ext-nacl)# **deny ip any any**
- R3(config-ext-nacl)# **exit**
- **Step 3. Apply the ACL to interface Serial 0/0/1.**
- R3(config)# **interface s0/0/1**

Configure Context-Based Access Control

- R3(config-if)# **ip access-group OUT-IN in**
- **Step 4. Confirm that traffic entering interface Serial 0/0/1 is dropped.**
- From the PC-C command prompt, ping the PC-A server.
- The ICMP echo replies are blocked by the ACL.

Task 2: Create a CBAC Inspection Rule

- **Step 1. Create an inspection rule to inspect ICMP, Telnet, and HTTP traffic.**

Configure Context-Based Access Control

- R3(config)# **ip inspect name IN-OUT-IN icmp**
- R3(config)# **ip inspect name IN-OUT-IN telnet**
- R3(config)# **ip inspect name IN-OUT-IN http**
- **Step 2. Turn on time-stamped logging and CBAC audit trail messages.**
- Use the **ip inspect audit-trail** command to turn on CBAC audit messages to provide a record of network access through the firewall, including illegitimate access attempts.

Configure Context-Based Access Control

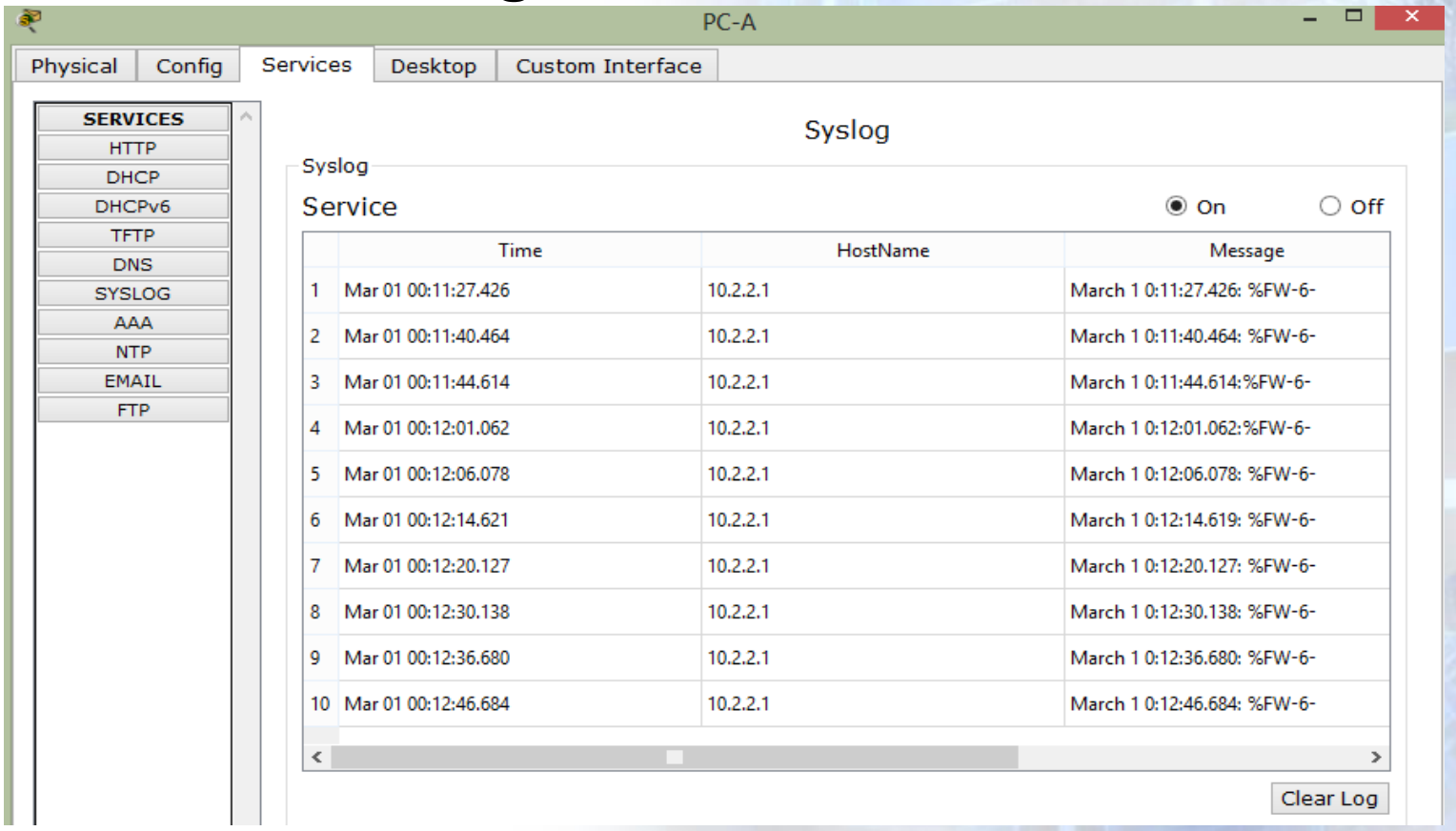
- Enable logging to the syslog server, 192.168.1.3, with the **logging host** command. Make sure that logged messages are time stamped.
- R3(config)# **ip inspect audit-trail**
- R3(config)# **service timestamps debug datetime msec**
- R3(config)# **logging host 192.168.1.3**
- **Step 3. Apply the inspection rule to egress traffic on interface S 0/0/1.**
- R3(config-if)# **ip inspect IN-OUT-IN out**

Configure Context-Based Access Control

- **Step 4. Verify that audit trail messages are being logged on the syslog server.**
- From PC-C, test connectivity to PC-A with ping, Telnet, and HTTP. Ping and HTTP should be successful. PC-A will reject the Telnet session.
- From PC-A, test connectivity to PC-C with ping and Telnet. All should be blocked.

Configure Context-Based Access Control

- Review the syslog messages on server PC-A: click the **Config** tab and then click the **SYSLOG**.



Configure Context-Based Access Control

Task 3: Verify Firewall Functionality

- **Step 1. Open a Telnet session from PC-C to R2.**
- The Telnet should succeed. While the Telnet session is active, issue the command **show ip inspect sessions** on R3. This command displays the existing sessions that are currently being tracked and inspected by CBAC.
- **R3# show ip inspect sessions**
- Session 145976424 (192.168.3.3:1029)=> (10.1.1.2:23) telnet SIS_OPEN

Configure Context-Based Access Control

- What is the source IP address and port number? 192.168.3.3:1029 (port 1029 is random)
- What is the destination IP address and port number? 10.1.1.2:23 (Telnet = port 23)
- **Exit** the Telnet session
- **Step 2. From PC-C, open a web browser to the PC-A server web page using the server IP address.**

Configure Context-Based Access Control

- The HTTP session should succeed. While the HTTP session is active, issue the command **show ip inspect sessions** on R3.
- R3# **show ip inspect sessions**
- Established Sessions: Session 104637440
- (192.168.3.3:1033)=>(192.168.1.3:80) http SIS_OPEN
- Source IP/port number=> destination IP/port number.
- **Close** the browser on PC-C.

Configure Context-Based Access Control

- **Step 3. View the interface configuration and inspection rule timers**.
- Enter the ***show ip inspect interfaces*** command on R3. The output shows existing sessions that are currently being tracked and inspected by CBAC.

```
R3# show ip inspect interfaces
Interface Configuration
 Interface Serial0/0/1
  Inbound inspection rule is not set
  Outgoing inspection rule is IN-OUT-IN
    icmp alert is on audit-trail is off timeout 10
    telnet alert is on audit-trail is off timeout 3600
    http alert is on audit-trail is off timeout 3600
  Inbound access list is OUT-IN
  Outgoing access list is not set
```

Configure Context-Based Access Control

Task 4: Review CBAC Configuration

- **Step 1. Display CBAC configuration.**
- Enter the ***show ip inspect config*** command on R3 to display the complete CBAC inspection configuration.

Configure Context-Based Access Control

```
R3# show ip inspect config
Session audit trail is enabled
Session alert is enabled
one-minute (sampling period) thresholds are [unlimited : unlimited]
connections
max-incomplete sessions thresholds are [unlimited : unlimited]
max-incomplete tcp connections per host is unlimited. Block-time 0 minute.
tcp synwait-time is 30 sec -- tcp finwait-time is 5 sec
tcp idle-time is 3600 sec -- udp idle-time is 30 sec
tcp reassembly queue length 16; timeout 5 sec; memory-limit 1024 kilo
bytes
dns-timeout is 5 sec
Inspection Rule Configuration
 Inspection name IN-OUT-IN
    icmp alert is on audit-trail is off timeout 10
    telnet alert is on audit-trail is off timeout 3600
    http alert is on audit-trail is off timeout 3600
```

Configure Context-Based Access Control

- **Step 2. Display real-time output that can be used for troubleshooting.**
- Enter the ***debug ip inspect detailed*** command on R3 to display detailed messages about CBAC software events, including information about CBAC packet processing.
- From PC-C, open a web browser; enter the PC-A (server) IP address: 192.168.1.3.

Configure Context-Based Access Control

```
R3# debug ip inspect detailed
INSPECT Detailed Debug debugging is on
*Mar 01, 02:37:28.3737:  %FW-6-SESS AUDIT TRAIL START: Start http session:
initiator (192.168.3.3:1039) -- responder (192.168.1.3:80)
*Mar 01, 02:37:28.3737: CBAC: Finding pregen session for src_tableid:0,
src addr:192.168.3.3, src_port:1039, dst_tableid:0, dst_addr:192.168.1.3,
dst port:80
*Mar 01, 02:37:38.3737:  %FW-6-SESS_AUDIT_TRAIL_STOP: Stop http session:
initiator (192.168.3.3:1041) -- responder (192.168.1.3:80)
```

Configure Context-Based Access Control

End

- **Step 3. Check Results.**
- Your completion percentage should be 100%.
- Click **Check Results** to see feedback and verification of which required components have been completed.

Configure a Zone-Based Policy Firewall

Network Security

Configure a Zone-Based Policy Firewall

Objectives of the Topic

- After completing this topic, a student will be able to
  - Configure a Zone-Based Policy Firewall in Packet tracer.

Configure a Zone-Based Policy Firewall

Figures and material in this topic have been adapted from

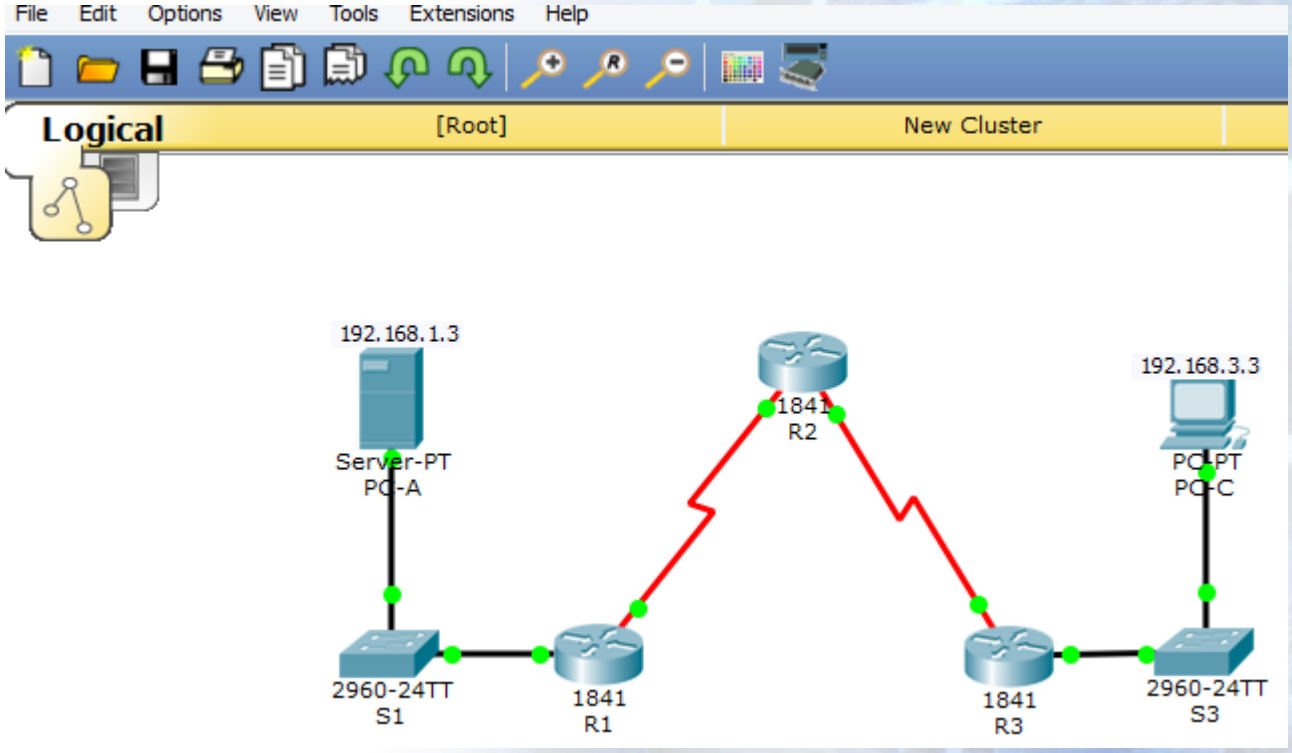
- *Chapter 4: PT Activity: Configuring a Zone-Based Policy Firewall (ZPF) – "CCNA Security 1.1 Student Packet Tracer Manual", Cisco 2012.*

Configure a Zone-Based Policy Firewall

- "Zone-based policy (ZPF) firewalls are the latest development in the evolution of Cisco firewall technologies. Interfaces are assigned to zones, and inspection policy is applied to traffic moving between the zones. Inter-zone policies offer considerable flexibility and granularity, so different inspection policies can be applied to multiple host groups connected to the same router interface."
- ***"Zone–Based Policy Firewall Design and Application Guide"*** **– Document ID: 98628, Cisco Systems, Inc, 2010**

Configure a Zone-Based Policy Firewall

Experimental Setup: Lets assume a network topology consists of three routers R1, R2 and R3.



Configure a Zone-Based Policy Firewall

Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	Fa0/1	192.168.1.1	255.255.255.0	N/A
	S0/0/0	10.1.1.1	255.255.255.252	N/A
R2	S0/0/0	10.1.1.2	255.255.255.252	N/A
	S0/0/1	10.2.2.2	255.255.255.252	N/A
R3	Fa0/1	192.168.3.1	255.255.255.0	N/A
	S0/0/1	10.2.2.1	255.255.255.252	N/A
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1
PC-C	NIC	192.168.3.3	255.255.255.0	192.168.3.1

Configure a Zone-Based Policy Firewall

- In this activity, we will configure a basic ZPF on an edge router R3 that allows internal hosts access to external resources and blocks external hosts from accessing internal resources.
- The routers have been pre-configured:
- Console password: **ciscoconpa55**
- Password for vty lines: **ciscovtypa55**
- Enable password: **ciscoenpa55**
- Host names,
- IP addressing, and Static routing

Configure a Zone-Based Policy Firewall

Task 1: Verify Basic Network Connectivity

- **Step 1. From the PC-A command prompt, ping PC-C at 192.168.3.3.**
- **Step 2. From the PC-C command prompt, Telnet to the Router R2 S0/0/1 interface at 10.2.2.2. Exit the Telnet session.**
- **Step 3. From PC-C, open a web browser to the PC-A server.** Click the **Desktop** tab and click the **Web Browser** application. Enter the PC-A IP address 192.168.1.3 as the URL. Close the browser on PC-C.

Configure a Zone-Based Policy Firewall

Task 2: Create the Firewall Zones on R3

- **Step 1. Create an internal zone.**
- Use the ***zone security*** command to create a zone named **IN-ZONE**.
- R3(config)# **zone security IN-ZONE**
- **Step 2. Create an external zone.**
- R3(config-sec-zone)# **zone security OUT-ZONE**
- R3(config-sec-zone)# **exit**

Configure a Zone-Based Policy Firewall

Task 3: Define a Traffic Class and Access List

- **Step 1. Create an ACL that defines internal traffic.**
- Use the ***access-list*** command to create extended ACL 101 to permit all IP protocols from the 192.168.3.0/24 source network to any destination.
- R3(config)# **access-list 101 permit ip 192.168.3.0 0.0.0.255 any**

Configure a Zone-Based Policy Firewall

- **Step 2. Create a class map referencing the internal traffic ACL.**
- Use the ***class map type inspect*** command with the match-all option to create a class map named **IN-NETCLASS-MAP**. Use the ***match access-group*** command to match ACL 101.
- R3(config)# **class-map type inspect match-all IN-NET-CLASS-MAP**
- R3(config-cmap)# **match access-group 101**
- R3(config-cmap)# **exit**

Configure a Zone-Based Policy Firewall

Task 4: Specify Firewall Policies

- **Step 1. Create a policy map to determine what to do with matched traffic.**
- Use the ***policy-map type inspect*** command and create a policy map named **IN-2-OUT-PMAP**.
- R3(config)# **policy-map type inspect IN-2-OUT-PMAP**
- **Step 2. Specify a class type of inspect and reference class map IN-NET-CLASS -MAP.**

Configure a Zone-Based Policy Firewall

- R3(config-pmap)# **class type inspect IN-NET-CLASS-MAP**
- **Step 3. Specify the action of inspect for this policy map**
- The use of the ***inspect*** command invokes context-based access control.
- R3(config-pmap-c)# **inspect**
- %No specific protocol configured in class IN-NET-CLASS-MAP for inspection. All protocols will be inspected.

Configure a Zone-Based Policy Firewall

- Issue the **exit** command twice to leave **config-pmap-c** mode and return to **config** mode.
- R3(config-pmap-c)# **exit**
- R3(config-pmap)# **exit**

Configure a Zone-Based Policy Firewall

Task 5: Apply Firewall Policies

- **Step 1. Create a pair of zones.**
- Using the ***zone-pair security*** command, create a zone pair named **IN-2-OUT-ZPAIR**. Specify the source and destination zones that were created in Task 1.
- R3(config)# **zone-pair security IN-2-OUT-ZPAIR source IN-ZONE destination OUT-ZONE**
- **Step 2. Specify the policy map for handling the traffic between the two zones.**

Configure a Zone-Based Policy Firewall

- Attach a policy-map and its associated actions to the zone pair using the ***service-policy type inspect*** command and reference the policy map previously created, **IN-2-OUT-PMAP**.
- R3(config-sec-zone-pair)# **service-policy type inspect IN-2-OUT-PMAP**
- R3(config-sec-zone-pair)# **exit**
- **Step 3. Assign interfaces to the appropriate security zones.**

Configure a Zone-Based Policy Firewall

- Use the ***zone-member security*** command in interface config mode to assign Fa0/1 to **IN-ZONE** and S0/0/1 to **OUT-ZONE**.
- R3(config)# **interface fa0/1**
- R3(config-if)# **zone-member security IN-ZONE**
- R3(config-if)# **exit**
- R3(config)# **interface s0/0/1**
- R3(config-if)# **zone-member security OUT-ZONE**
- **Step 4. Copy the running config to the startup config.**

Configure a Zone-Based Policy Firewall

Task 6: Test Firewall Functionality from IN-ZONE to OUT-ZONE

- Verify that internal hosts can still access external resources after configuring the zone-based policy firewall.
- **Step 1. From internal PC-C, ping the external PC-A server.**
- From the PC-C Command Prompt, ping PC-A at 192.168.1.3. The ping should succeed.

Configure a Zone-Based Policy Firewall

- **Step 2. From internal PC-C, Telnet to the router R2 S0/0/1 interface.**
- From the PC-C Command Prompt, telnet to R2 at 10.2.2.2 and provide the vty password **ciscovtypa55**.
- The telnet should succeed. While the Telnet session is active, issue the command ***show policy-map type inspect zone-pair sessions*** on R3 to view established sessions.

Configure a Zone-Based Policy Firewall

```
R3# show policy-map type inspect zone-pair sessions

Zone-pair: IN-ZONE-OUT-ZONE

  Service-policy inspect : IN-2-OUT-PMAP

    Class-map: IN-NET-CLASS-MAP (match-all)
      Match: access-group 101
      Inspect
        Established Sessions
         Session 139644744 (192.168.3.3:1025)=>(10.2.2.2:23) telnet:tcp
SIS_OPEN
          Created 00:00:02, Last heard 00:00:00
          Bytes sent (initiator:responder) [0:0]
```

- source IP/port number = 192.168.3.3:1025
- destination IP/port number = 10.2.2.2:23

Configure a Zone-Based Policy Firewall

- **Step 3. From PC-C, exit the Telnet session on R2 and close the Command Prompt window.**
- **Step 4. From internal PC-C, open a web browser to the PC-A server web page.**
- Enter the server IP address 192.168.1.3 in the browser URL field and click **Go**.
- The HTTP session should succeed. While the HTTP session is active, issue the command ***show policy-map type inspect zone-pair sessions*** on R3 to view established sessions.

Configure a Zone-Based Policy Firewall

```
R3# show policy-map type inspect zone-pair sessions
 Zone-pair: IN-ZONE-OUT-ZONE

  Service-policy inspect : IN-2-OUT-PMAP

    Class-map: IN-NET-CLASS-MAP (match-all)
      Match: access-group 101
      Inspect
        Established Sessions
                Session 139142400 (192.168.3.3:1027)=>(192.168.1.3:80)
                http:tcp SIS_OPEN
            Created 00:00:02, Last heard 00:00:00
            Bytes sent (initiator:responder) [0:0]
```

- source IP/port number = 192.168.3.3:1027
- destination IP/port number = 192.168.1.3:80

Configure a Zone-Based Policy Firewall

Task 7: Test Firewall Functionality from OUT-ZONE to IN-ZONE

- Verify that external hosts CANNOT access internal resources after configuring the zone-based policy firewall.
- **Step 1. From the PC-A server command prompt, ping PC-C at 192.168.3.3.** The ping should fail.
- **Step 2. From router R2, ping PC-C at 192.168.3.3.** The ping should fail.

Configure a Zone-Based Policy Firewall

End

- **Step 3. Check results.**
- Your completion percentage should be 100%.
- Click **Check Results** to see feedback and verification of which required components have been completed.

Configure IOS Intrusion Prevention System

Network Security

Configure IOS Intrusion Prevention System

Objectives of the Topic

- After completing this topic, a student will be able to
  - Configure IOS Intrusion Prevention System using CLI in Packet tracer.

Configure IOS Intrusion Prevention System

Figures and material in this topic have been adapted from

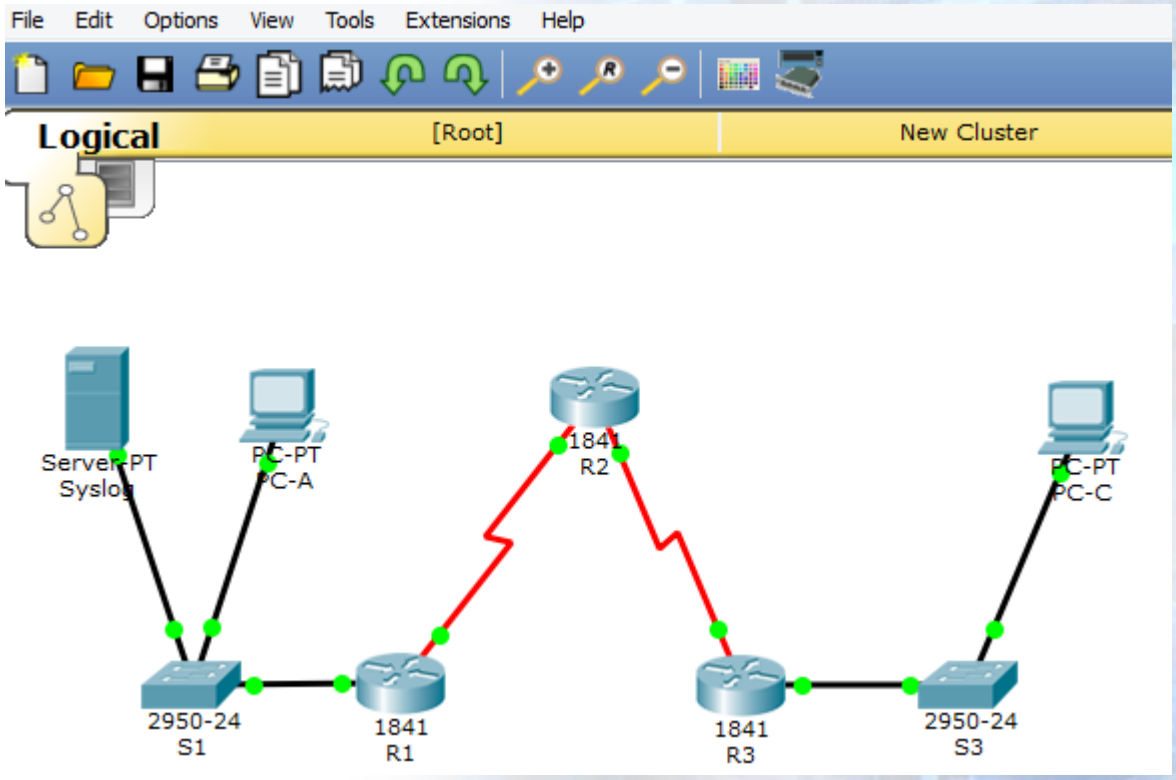
- *Chapter 5: PT Activity: Configure IOS Intrusion Prevention System (IPS) using CLI – "CCNA Security 1.1 Student Packet Tracer Manual", Cisco 2012.*

Configure IOS Intrusion Prevention System

- "Cisco IOS Intrusion Prevention System (IOS IPS) is used to manage intrusion prevention on Cisco routers. It acts as an in-line intrusion prevention sensor, watching packets and sessions as they flow through the router and scanning each packet to match any of the Cisco IOS IPS signatures. When it detects suspicious activity, it responds before network security can be compromised and logs the event through Cisco IOS syslog messages."
- ***"User Guide for Cisco Security Manager 4.1"*** **– Cisco Systems, Inc, 2011**

Configure IOS Intrusion Prevention System

Experimental Setup: Lets assume a network topology consists of three routers R1, R2, and R3.



Configure IOS Intrusion Prevention System

Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	FA0/0	192.168.1.1	255.255.255.0	N/A
	S0/0/0	10.1.1.1	255.255.255.0	N/A
R2	S0/0/0 (DCE)	10.1.1.2	255.255.255.0	N/A
	S0/0/1 (DCE)	10.2.2.1	255.255.255.0	N/A
R3	FA0/0	192.168.3.1	255.255.255.0	N/A
	S0/0/0	10.2.2.2	255.255.255.0	N/A
Syslog Server	NIC	192.168.1.50	255.255.255.0	192.168.1.1
PC-A	NIC	192.168.1.2	255.255.255.0	192.168.1.1
PC-C	NIC	192.168.3.2	255.255.255.0	192.168.3.1

Configure IOS Intrusion Prevention System

- We want to configure router R1 for IPS in order to scan traffic entering the 192.168.1.0 network.
- The server labeled ‘Syslog Server’ is used to log IPS messages. You must configure the router to identify the syslog server in order to receive logging messages. Displaying the correct time and date in syslog messages is vital when using syslog to monitor the network.
- Set the clock and configure timestamp service for logging on the routers.

Configure IOS Intrusion Prevention System

- Finally, enable IPS to produce an alert and drop ICMP echo reply packets inline.
- The server and PCs have been preconfigured.
- The routers have also been preconfigured with the following:
- Enable password: **ciscoenpa55**
- Console password: **ciscoconpa55**
- VTY line password: **ciscovtypa55**
- Enhanced Interior Gateway Routing Protocol (EIGRP 101)

Configure IOS Intrusion Prevention System

Task 1: Enable IOS IPS

- Within Packet Tracer, the routers already have the signature files imported and in place. For this reason, it is not necessary to configure the public crypto key and complete a manual import of the signature files.
- **Step 1. Verify network connectivity.**
- **Ping** from PC-C to PC-A. The ping should be successful. **Ping** from PC-A to PC-C. The ping should be successful.

Configure IOS Intrusion Prevention System

- **Step 2. Create an IOS IPS configuration directory in flash.**
- R1#**mkdir ipsdir**
- Create directory filename [ipsdir]? <**Enter**>
- Created dir flash:ipsdir
- **Step 3. Configure the IPS signature storage location.**
- On R1, configure the IPS signature storage location to be the directory you just created.
- R1(config)#**ip ips config location flash:ipsdir**

Configure IOS Intrusion Prevention System

- **Step 4. Create an IPS rule.**
- On R1, create an IPS rule name using the **ip ips name** *name* command in global configuration mode. Name the IPS rule **iosips**.
- R1(config)# **ip ips name iosips**
- **Step 5. Enable logging.**
- IOS IPS supports the use of syslog to send event notification. Syslog notification is enabled by default. If logging console is enabled, you see IPS syslog messages.

Configure IOS Intrusion Prevention System

- Enable syslog if it is not enabled.
- R1(config)# **ip ips notify log**
- Use the ***clock set*** command from privileged EXEC mode to reset the clock if necessary.
- R1# **clock set 01:20:00 6 january 2009**
- Verify that the timestamp service for logging is enabled on the router using the ***show run*** command.
- Enable the timestamp service if it is not enabled.

Configure IOS Intrusion Prevention System

- R1(config)# **service timestamps log datetime msec**
- Send log messages to the Syslog server at IP address 192.168.1.50.
- R1(config)# **logging host 192.168.1.50**
- **Step 6. Configure IOS IPS to use the signature categories.**
- Retire the ***all*** signature category with the ***retired true*** command (all signatures within the signature release).

Configure IOS Intrusion Prevention System

- Unretire the **IOS_IPS Basic** category with the ***retired false*** command.
- R1(config)# **ip ips signature-category**
- R1(config-ips-category)# **category all**
- R1(config-ips-category-action)# **retired true**
- R1(config-ips-category-action)# **exit**
- R1(config-ips-category)# **category ios_ips basic**
- R1(config-ips-category-action)# **retired false**
- R1(config-ips-category-action)# **exit**
- R1(config-ips-cateogry)# **exit**

Configure IOS Intrusion Prevention System

- Do you want to accept these changes? **<Enter>**
- **Step 7. Apply the IPS rule to an interface.**
- Apply the IPS rule to an interface with the **ip ips** *name direction* command in interface configuration mode. Apply the rule outbound on the Fa0/0 interface of R1. After you enable IPS, some log messages will be sent to the console line indicating that the IPS engines are being initialized.

Configure IOS Intrusion Prevention System

- The direction **in** means that IPS inspects only traffic going into the interface. Similarly, **out** means only traffic going out the interface.
- R1(config)# **interface fa0/0**
- R1(config-if)# **ip ips iosips out**

Configure IOS Intrusion Prevention System

Task 2: Modify the Signature

- **Step 1. Change the event-action of a signature.**
- Un-retire the echo request signature (signature 2004, subsig ID 0), enable it and change the signature action to alert, and drop.
- R1(config)# **ip ips signature-definition**
- R1(config-sigdef)# **signature 2004 0**
- R1(config-sigdef-sig)# **status**
- R1(config-sigdef-sig-status)# **retired false**
- R1(config-sigdef-sig-status)# **enabled true**

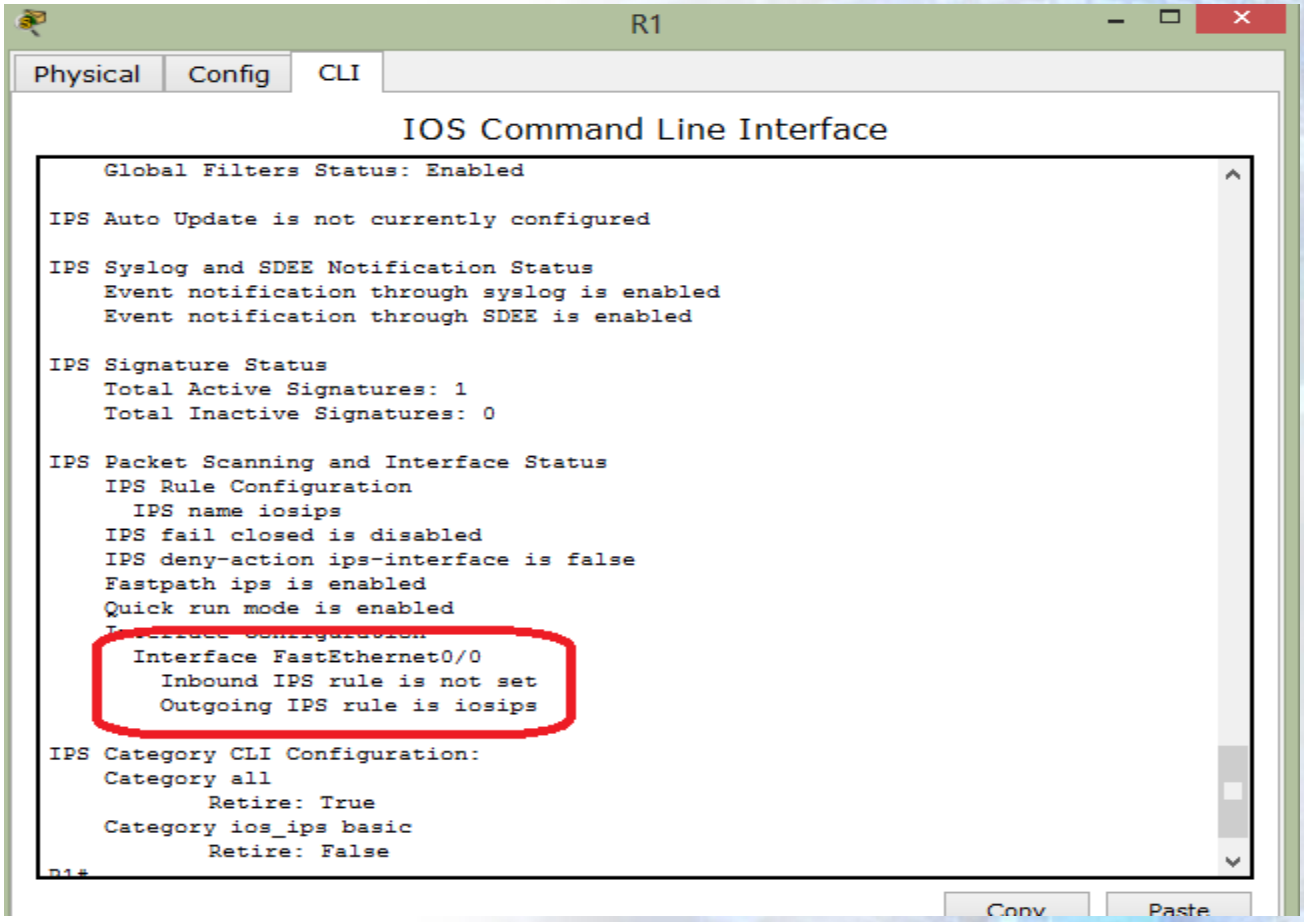
Configure IOS Intrusion Prevention System

- R1(config-sigdef-sig-status)# **exit**
- R1(config-sigdef-sig)# **engine**
- R1(config-sigdef-sig-engine)# **event-action produce-alert**
- R1(config-sigdef-sig-engine)# **event-action deny-packet-inline**
- R1(config-sigdef-sig-engine)# **exit**
- R1(config-sigdef-sig)# **exit**
- R1(config-sigdef)# **exit**
- Do you want to accept these changes? **<Enter>**

Configure IOS Intrusion Prevention System

- **Step 2. Use show commands to verify IPS .**
- Use the ***show ip ips all*** command to see an IPS configuration status summary. To which interfaces and in which direction is the iosips rule applied?
- Fa0/0 outbound.

Configure IOS Intrusion Prevention System



- Fa0/0 outbound.

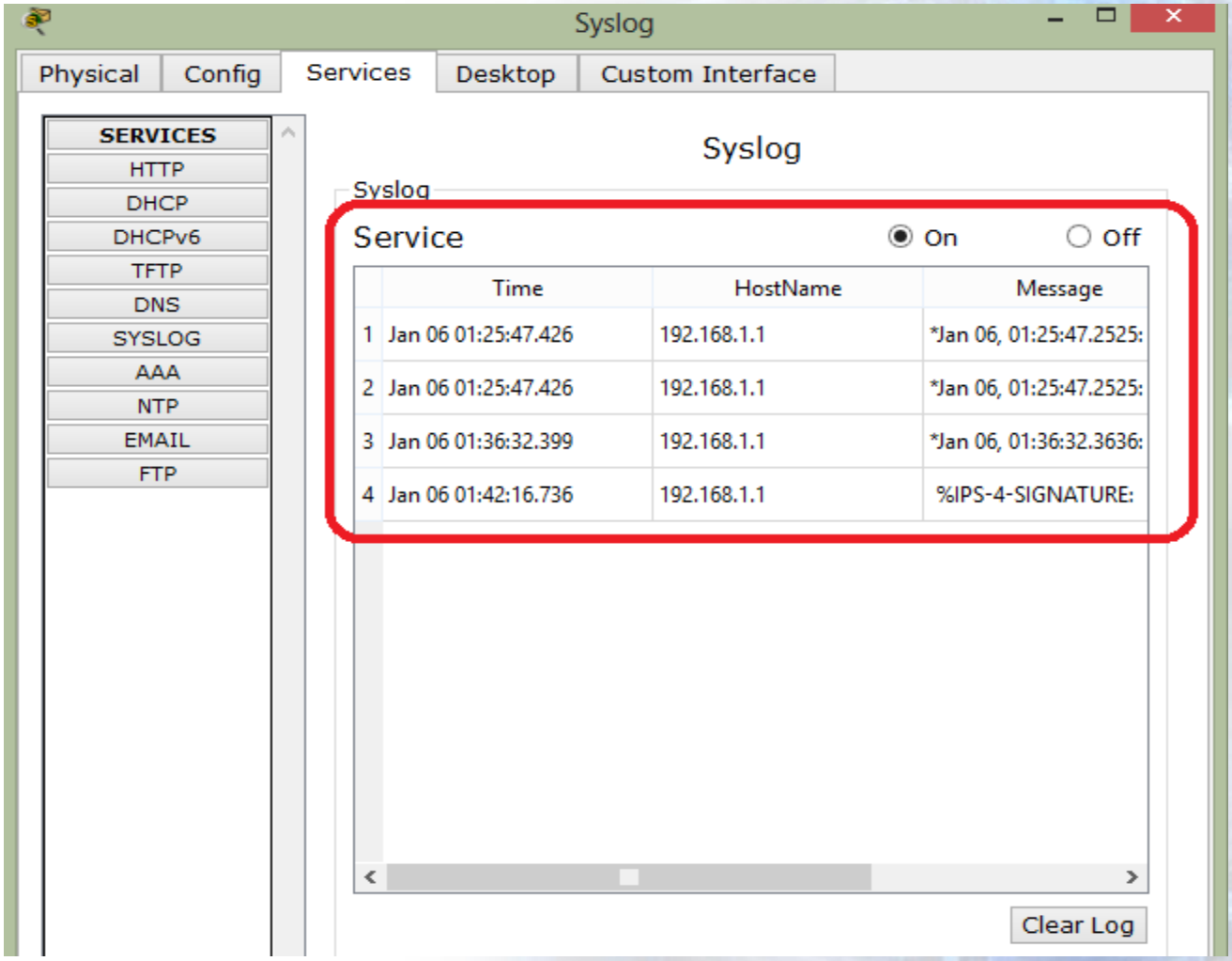
Configure IOS Intrusion Prevention System

- **Step 3. Verify that IPS is working properly.**
- From PC-C, attempt to **ping** PC-A. Were the pings successful? Why or why not?
- The pings should fail. This is because the IPS rule for event-action of an echo request was set to “deny-packet-inline”.

Configure IOS Intrusion Prevention System

- From PC-A, attempt to **ping** PC-C. Were the pings successful? Why or why not?
- The ping should be successful. This is because the IPS rule does not cover echo reply. When PC-A pings PC-C, PC-C responds with an echo reply.
- **Step 4. View the Syslog messages.**
- Click on the **Syslog** server. Select the **Config** tab. In the left navigation menu, select **SYSLOG** to view the log file.

Configure IOS Intrusion Prevention System



Configure IOS Intrusion Prevention System

End

- **Step 5. Check results.**
- Your completion percentage should be 100%.
- Click **Check Results** to see feedback and verification of which required components have been completed.

Configuring Layer 2 Security

Network Security

Configuring Layer 2 Security

Objectives of the Topic

- After completing this topic, a student will be able to
  - Configure Layer 2 security in Packet tracer.

Configuring Layer 2 Security

Figures and material in this topic have been adapted from

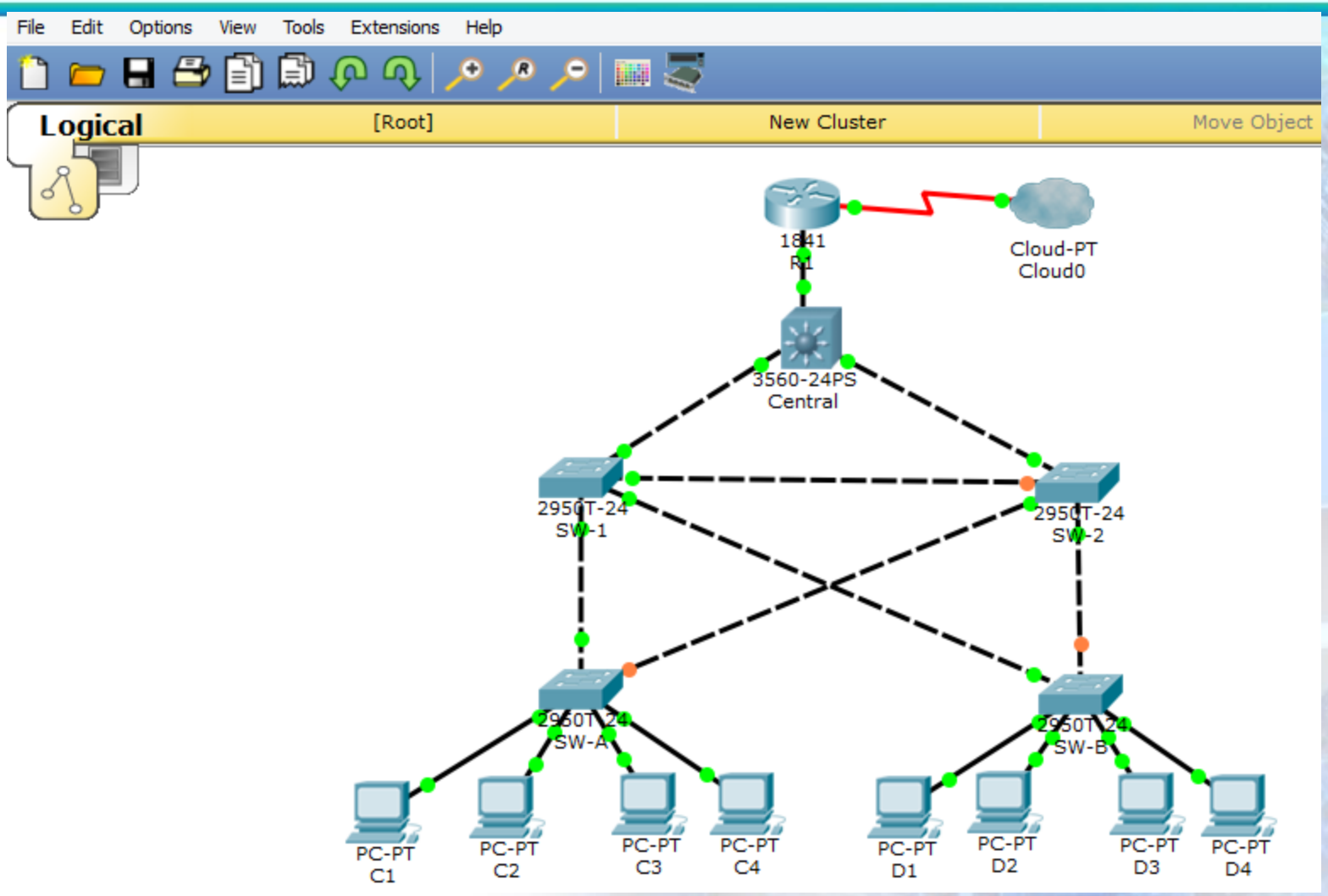
- *Chapter 6: PT Activity: Layer 2 Security – "CCNA Security 1.1 Student Packet Tracer Manual", Cisco 2012.*

Configuring Layer 2 Security

Experimental Setup:

- Assume that your company's network administrator has assigned you the task of configuring Layer 2 security.
- For optimum performance and security, he has recommended that the root bridge is the 3560 Central switch and wants that the spanning tree protocol (STP- a Layer 2 link management protocol) parameters are secure against spanning-tree manipulation attacks.

Configuring Layer 2 Security



Configuring Layer 2 Security

- To prevent broadcast storms, the network administrator wants you to enable storm control.
- Finally, he has decided to configure port security to limit the number of MAC addresses that can be learned per switch port to prevent against MAC address table overflow attacks,.
- In case, the number of MAC addresses exceeds the set limit, the administrator wants the port to be shutdown.

Configuring Layer 2 Security

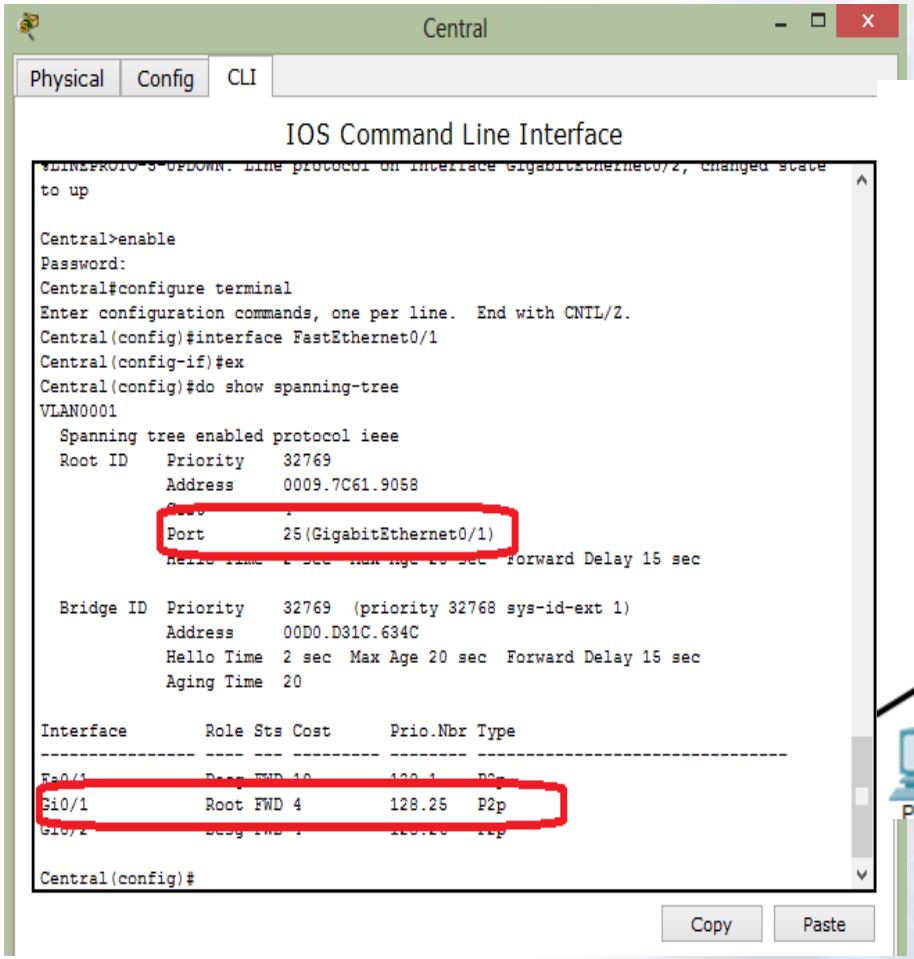
- All switch devices have been preconfigured with the following:
- Enable password: **ciscoenpa55**
- Console password: **ciscoconpa55**
- VTY line password: **ciscovtypa55**

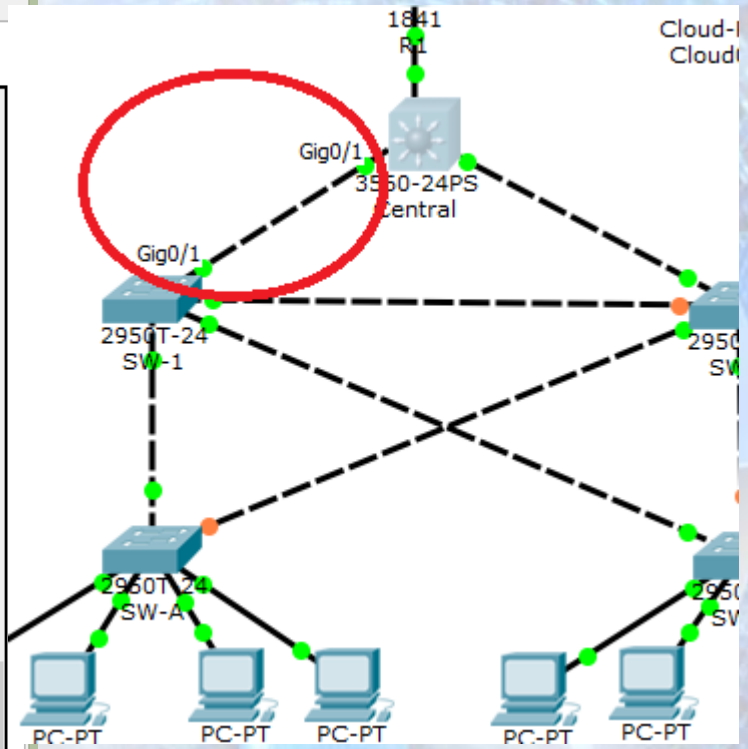
Configuring Layer 2 Security

Task 1: Configure Root Bridge

- **Step 1. Determine the current root bridge.**
- From Central, issue the ***show spanning-tree*** command to determine the current root bridge and to see the ports in use and their status.
- Which switch is the current root-bridge?

Configuring Layer 2 Security





Current root is SW-1

Configuring Layer 2 Security

- **Step 2. Assign Central as the primary root bridge.**
- Use the ***spanning-tree vlan 1 root primary*** command to make the 3560 Central switch as the root bridge.
- Central(config)# **spanning-tree vlan 1 root primary**
- **Step 3. Assign SW-1 as a secondary root bridge.**
- Use the ***spanning-tree vlan 1 root secondary*** command.

Configuring Layer 2 Security

- SW-1(config)# **spanning-tree vlan 1 root secondary**
- **Step 4. Verify the spanning-tree configuration.**
- Issue the ***show spanning-tree*** command to verify that 3560 Central switch is the root bridge.
- Which switch is the current root-bridge?
  Current root is Central

Configuring Layer 2 Security

Task 2: Protect Against STP Attacks

- Secure the STP parameters to prevent STP manipulation attacks.
- **Step 1. Enable PortFast on all access ports.**
- PortFast is configured on access ports that connect to a single workstation or server to enable them to become active more quickly.
- On the connected access ports of the SW-A and SW-B switches, use the ***spanning-tree portfast*** command.

Configuring Layer 2 Security

- SW-A(config)# **interface range fastethernet 0/1 - 4**
- SW-A(config-if-range)# **spanning-tree portfast**
- SW-B(config)# **interface range fastethernet 0/1 - 4**
- SW-B(config-if-range)# **spanning-tree portfast**
- **Step 2. Enable BPDU guard on all access ports.**
- Bridge Protocol Data Unit (BPDU) guard is a feature that can help prevent rogue switches and spoofing on access ports.

Configuring Layer 2 Security

- Enable BPDU guard on SW-A and SW-B.
- SW-A(config)# **interface range fastethernet 0/1 - 4**
- SW-A(config-if-range)# **spanning-tree bpduguard enable**
- SW-B(config)# **interface range fastethernet 0/1 - 4**
- SW-B(config-if-range)# **spanning-tree bpduguard enable**

Configuring Layer 2 Security

- **Step 3. Enable root guard.**
- Root guard can be enabled on all ports on a switch that are not root ports.
- It is best deployed on ports that connect to other non-root switches.
- Use the ***show spanning-tree*** command to determine the location of the root port on each switch.
- On switch SW-1, enable root guard on ports Fa0/23 and Fa0/24. On switch SW-2, enable root guard on ports Fa0/23 and Fa0/24.

Configuring Layer 2 Security

- SW-1(config)# **interface fa0/23**
- SW-1(config-if)# **spanning-tree guard root**
- SW-1(config-if)# **interface fa0/24**
- SW-1(config-if)# **spanning-tree guard root**
- SW-2(config)# **interface fa0/23**
- SW-2(config-if)# **spanning-tree guard root**
- SW-2(config-if)# **interface fa0/24**
- SW-2(config-if)# **spanning-tree guard root**

Configuring Layer 2 Security

Task 3: Enable Storm Control

- **Step 1. Enable storm control for broadcasts.**
- Enable storm control for broadcasts on all ports connecting switches (trunk ports).
- Set a **50** percent rising suppression level using the ***storm-control broadcast*** command.
- Enable storm-control on interfaces connecting Central, SW-1, and SW-2.

Configuring Layer 2 Security

- SW-1(config)# **interface gi1/1**
- SW-1(config-if)# **storm-control broadcast level 50**
- SW-1(config-if)# **interface fa0/1**
- SW-1(config-if)# **storm-control broadcast level 50**
- SW-1(config-if)# **interface fa0/23**
- SW-1(config-if)# **storm-control broadcast level 50**

Configuring Layer 2 Security

- SW-1(config-if)# **interface fa0/24**
- SW-1(config-if)# **storm-control broadcast level 50**
- Repeat on SW-2 (gig1/1, fa0/1, fa0/23, and fa0/24) and Central (gig0/1, gig0/2, fa0/1) connection to other switches

Configuring Layer 2 Security

- **Step 2. Verify storm control configuration.**
- Verify your configuration with the ***show storm-control broadcast*** command and the ***show run*** command.

Task 4: Configure Port Security and Disable Unused Ports

- **Step 1. Configure basic port security on all ports connected to host devices.**
- This procedure should be performed on all access ports on SW-A and SW-B.

Configuring Layer 2 Security

- Set the maximum number of learned MAC address to **2**, allow the MAC address to be learned dynamically, and set the violation to **shutdown**.
- A switch port must be configured as an access port to enable port security.

Configuring Layer 2 Security

- SW-A(config)# **interface FastEthernet 0/1**
- SW-A(config-if)# **switchport mode access**
- SW-A(config-if)# **switchport port-security**
- SW-A(config-if)# **switchport port-security maximum 2**
- SW-A(config-if)# **switchport port-security violation shutdown**
- SW-A(config-if)# **switchport port-security mac-address sticky**
- Repeat on other ports in SW-A and SW-B

Configuring Layer 2 Security

- Why would you not want to enable port-security on ports connected to other switches or routers?
- Ports connected to other switch devices and routers can, and should, have a multitude of MAC addresses learned for that single port.
- Limiting the number of MAC addresses that can be learned on these ports can significantly impact network functionality.

Configuring Layer 2 Security

- **Step 2. Verify port security.**
- On SW-A, issue the ***show port-security interface fa0/1*** command to verify that port security has been configured.
- **Step 3. Disable unused ports.**
- Disable all ports that are currently unused.
- SW-A(config)# **interface FastEthernet 0/5**
- SW-A(config-if)# **shutdown**
- Repeat on other ports on SW-A and SW-B

Configuring Layer 2 Security

End

- **Step 4. Check results.**
- Your completion percentage should be 100%.
- Click **Check Results** to see feedback and verification of which required components have been completed.

Configuring Layer 2 VLAN Security

Network Security

Configuring Layer 2 VLAN Security

Objectives of the Topic

- After completing this topic, a student will be able to
  - Configure Layer 2 VLAN security in Packet tracer.

Configuring Layer 2 VLAN Security

Figures and material in this topic have been adapted from

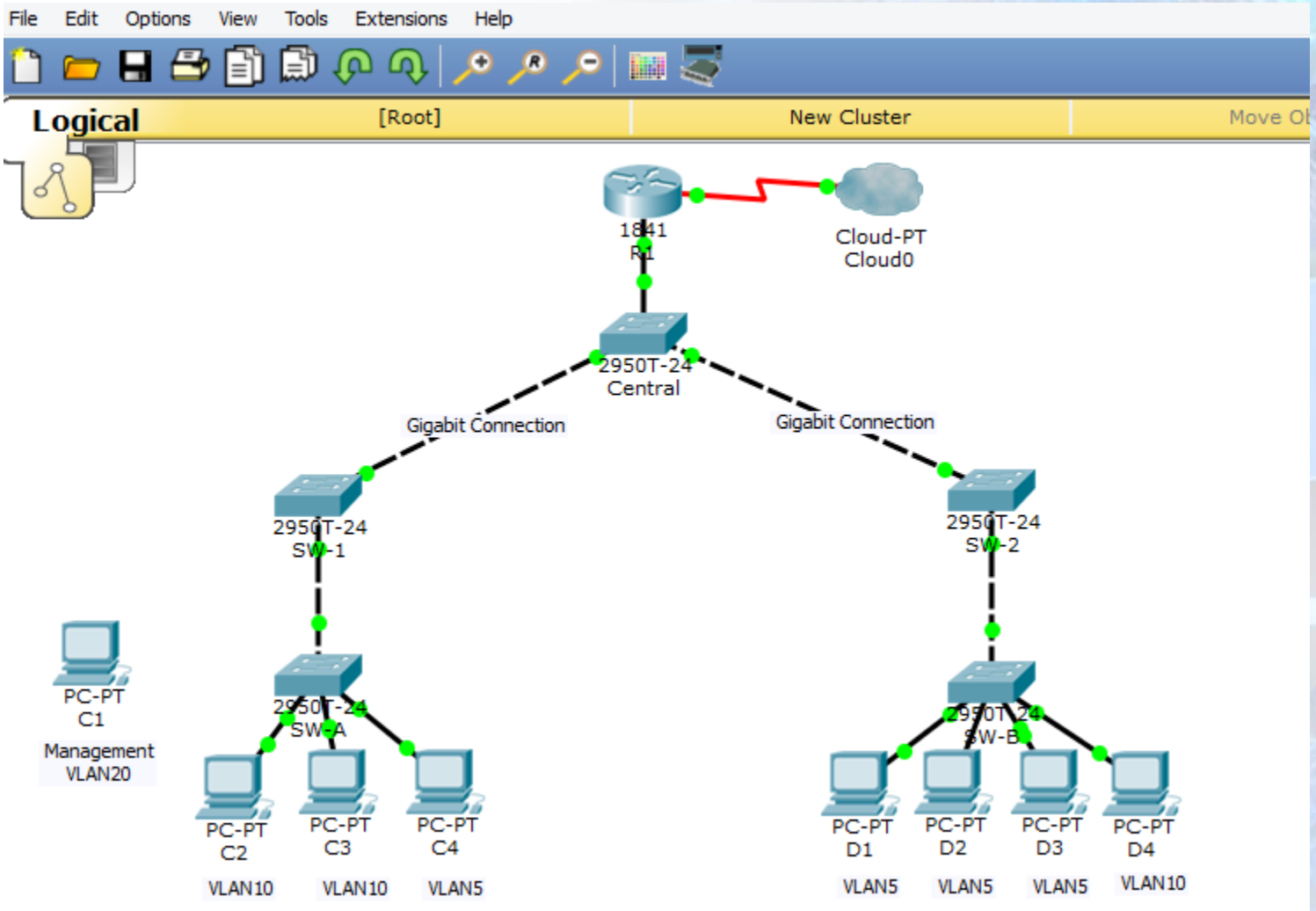
- *Chapter 6: PT Activity: Layer 2 VLAN Security – "CCNA Security 1.1 Student Packet Tracer Manual", Cisco 2012.*

Configuring Layer 2 VLAN Security

Experimental Setup:

- Lets assume that the network of a company is set up using two separate VLANs: VLAN 5 and VLAN 10.
- All trunk ports are configured with VLAN 15.
- The network administrator wants to add a redundant link between switch SW-1 and SW-2.
- The link must have trunking enabled and all security requirements should be in place.

Configuring Layer 2 VLAN Security



Configuring Layer 2 VLAN Security

- Also, the network administrator wants to connect a management PC to switch SW-A.
- The administrator would like to allow the management PC to be able to connect to all switches and the router, but does not want any other devices to be able to connect to the management PC or the switches.
- For management purposes, the administrator would like to create a new VLAN 20.

Configuring Layer 2 VLAN Security

- All devices have been preconfigured with:
- Enable secret password: **ciscoenpa55**
- Console password: **ciscoconpa55**
- VTY line password: **ciscovtypa55**

Configuring Layer 2 VLAN Security

Task 1: Verify Connectivity

- **Step 1. Verify connectivity between C2 (VLAN 10) and C3 (VLAN 10).**
- **Step 2. Verify connectivity between C2 (VLAN 10) and D1 (VLAN 5).**
- If you are using the simple PDU GUI packet, **ping** twice to allow for ARP.

Configuring Layer 2 VLAN Security

Task 2: Create a Redundant Link Between SW-1 and SW-2

- **Step 1. Connect SW-1 and SW-2.**
- Use a crossover cable to connect port Fa0/23 on SW-1 to port Fa0/23 on SW-2.
- **Step 2. Enable trunking, including all trunk security mechanisms on the link between SW-1 and SW-2.**
- Trunking has already been configured on all pre-existing trunk interfaces.

Configuring Layer 2 VLAN Security

- The new link must be configured for trunking, including all trunk security mechanisms. On both SW-1 and SW-2, set the port to trunk, assign native VLAN 15 to the trunk port, and disable auto-negotiation.
- SW-1(config)# **interface fa0/23**
- SW-1(config-if)# **no shutdown**
- SW-1(config-if)# **switchport mode trunk**
- SW-1(config-if)# **switchport trunk native vlan 15**
- SW-1(config-if)# **switchport nonegotiate**

Configuring Layer 2 VLAN Security

- SW-1(config-if)# **switchport nonegotiate**
- SW-2(config)# **interface fa0/23**
- SW-2(config-if)# **no shutdown**
- SW-2(config-if)# **switchport mode trunk**
- SW-2(config-if)# **switchport trunk native vlan 15**
- SW-2(config-if)# **switchport nonegotiate**

Configuring Layer 2 VLAN Security

- **Task 3: Enable VLAN 20 as a Management VLAN**
- The network administrator would like to access all switch and routing devices using a management PC.
- For security, the administrator wants to ensure that all managed devices are on a separate VLAN.

Configuring Layer 2 VLAN Security

- **Step 1. Enable a management VLAN (VLAN 20) on SW-A.**
- Let's use the default name of VLAN0020.
- SW-A(config)# **vlan 20**
- SW-A(config-vlan)# **exit**
- Create an interface VLAN 20 and assign an IP address within the 192.168.20.0/24 network
- SW-A(config)# **interface vlan 20**
- SW-A(config-if)# **ip address 192.168.20.1 255.255.255.0**

Configuring Layer 2 VLAN Security

- **Step 2. Enable the same management VLAN on all other switches.**
- Create the VLAN on all switches: SW-B, SW-1, SW-2 and Central.
- SW-B(config)# **vlan 20**
- SW-B(config-vlan)# **exit**
- SW-B(config)# **interface vlan 20**
- SW-B(config-if)# **ip address 192.168.20.2 255.255.255.0**

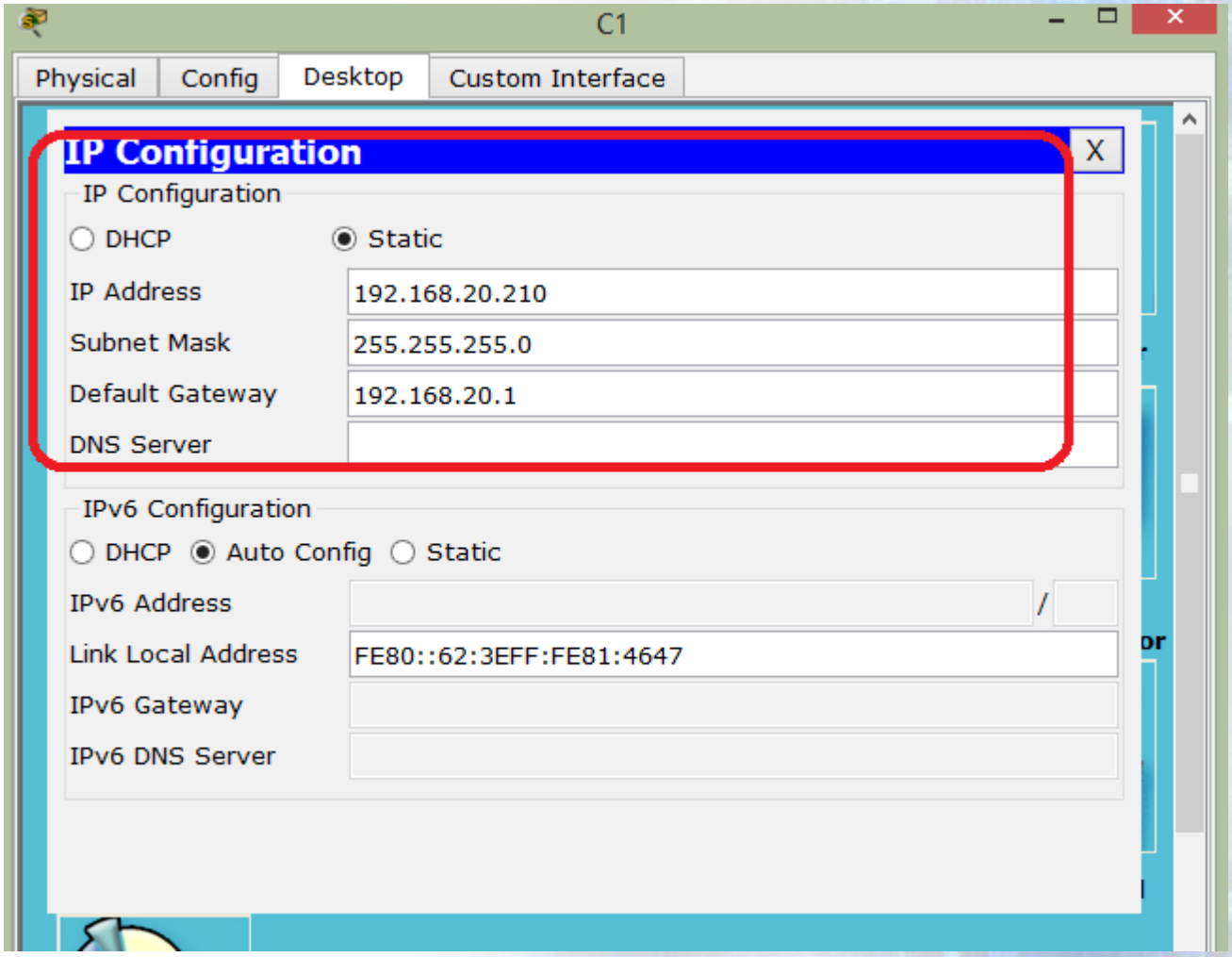
Configuring Layer 2 VLAN Security

- SW-1(config)# **vlan 20**
- SW-1(config-vlan)# **exit**
- SW-1(config)# **interface vlan 20**
- SW-1(config-if)# **ip address 192.168.20.3 255.255.255.0**
- SW-2(config)# **vlan 20**
- SW-2(config-vlan)# **exit**
- SW-2(config)# **interface vlan 20**
- SW-2(config-if)# **ip address 192.168.20.4 255.255.255.0**

Configuring Layer 2 VLAN Security

- Central(config)# **vlan 20**
- Central(config-vlan)# **exit**
- Central(config)# **interface vlan 20**
- Central(config-if)# **ip address 192.168.20.5 255.255.255.0**
- **Step 3. Configure the management PC and connect it to SW-A port Fa0/1.**
- Assign management PC an IP address within the 192.168.20.0/24 network and connect it to SW-A port Fa0/1.

Configuring Layer 2 VLAN Security



Configuring Layer 2 VLAN Security

- **Step 4. On SW-A, ensure the management PC is part of VLAN 20**
- Interface Fa0/1 must be part of VLAN 20.
- SW-A(config)# **interface fa0/1**
- SW-A(config-if)# **switchport access vlan 20**
- SW-A(config-if)# **no shutdown**
- **Step 5. Verify connectivity of the management PC to all switches.**
- The management PC should be able to **ping** SW-A, SW-B, SW-1, SW-2 and Central.

Configuring Layer 2 VLAN Security

Task 4: Enable the Management PC to Access Router R1

- **Step 1. Enable a new subinterface on router R1.**
- Create subinterface Fa0/0.3 and assign an IP address within the 192.168.20.0/24 network.
- Set encapsulation to dot1q 20 to account for VLAN 20.

Configuring Layer 2 VLAN Security

- R1(config)# **interface fa0/0.3**
- R1(config-subif)# **encapsulation dot1q 20**
- R1(config-subif)# **ip address 192.168.20.100 255.255.255.0**
- **Step 2. Verify connectivity between the management PC and R1.**
- Configure the default gateway on the management PC to allow for connectivity.

Configuring Layer 2 VLAN Security

- **Step 3. Enable security.**
- While the management PC must be able to access the router, no other PC should be able to access the management VLAN. Create an ACL(s) that denies any network from accessing the 192.168.20.0/24 network, but permits all other networks to access one another.
- R1(config)# **access-list 101 deny ip any 192.168.20.0 0.0.0.255**
- R1(config)# **access-list 101 permit ip any any**

Configuring Layer 2 VLAN Security

- Apply the ACL to the proper interface(s).
- R1(config)# **int fa0/0.1**
- R1(config-subif)# **ip access-group 101 in**
- R1(config-subif)# **int fa0/0.2**
- R1(config-subif)# **ip access-group 101 in**
- The management PC must be able to connect to all switches and the router. All other PCs should not be able to connect to any devices within the management VLAN.

Configuring Layer 2 VLAN Security

- **Step 4. Verify Security.**
- From the management PC, **ping** SW-A, SW-B, and R1. Was the **ping** successful?
- The ping should have been successful because all devices within the 192.168.20.0 network should be able to ping one another.
- Devices within VLAN20 are not required to route through the router.

Configuring Layer 2 VLAN Security

- From D1, **ping** the management PC. Was the **ping** successful?
- The ping should have failed. This is because in order for a device within a different VLAN to successfully ping a device within VLAN20, it must be routed.
- The router has an ACL that prevents all packets from accessing the 192.168.20.0 network.

Configuring Layer 2 VLAN Security

End

- **Step 5. Check results.**
- Your completion percentage should be 100%.
- Click **Check Results** to see feedback and verification of which required components have been completed

Configure & Verify a Site-to-Site IPsec VPN

Network Security

Configure & Verify a Site-to-Site IPsec VPN

Objectives of the Topic

- After completing this topic, a student will be able to
  - Configure and Verify a Site-to-Site IPsec VPN in Packet tracer.

Configure & Verify a Site-to-Site IPsec VPN

Figures and material in this topic have been adapted from

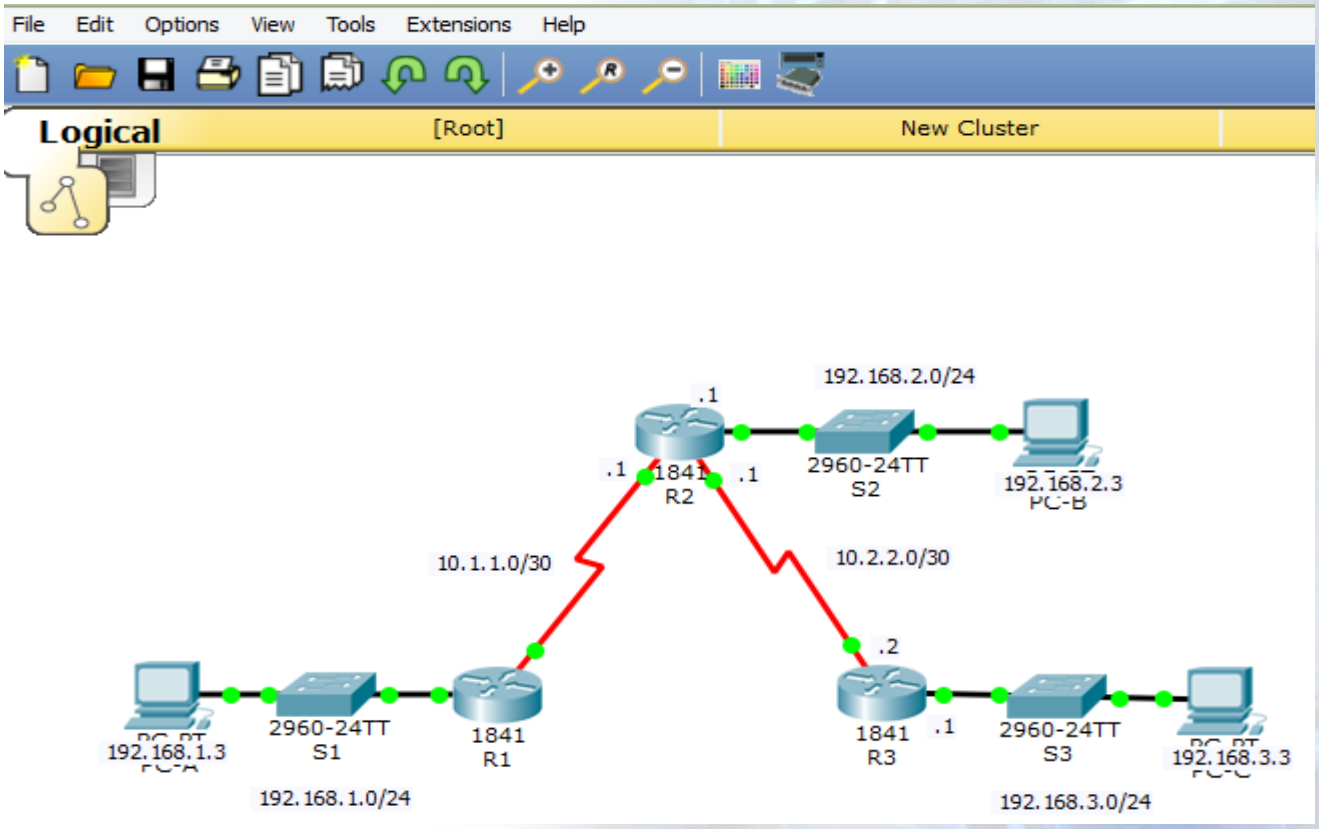
- *Chapter 8: PT Activity: Configure and Verify a Site-to-Site IPsec VPN using CLI – "CCNA Security 1.1 Student Packet Tracer Manual", Cisco 2012.*

Configure & Verify a Site-to-Site IPsec VPN

- IPsec provides secure transmission of sensitive information over unprotected networks such as the Internet.
- IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec devices (peers), such as Cisco routers.

Configure & Verify a Site-to-Site IPsec VPN

Experimental Setup: Lets assume a network topology consists of three routers R1, R2 and R3.



Configure & Verify a Site-to-Site IPsec VPN

Addressing Table

Device	Interface	IP Address	Subnet Mask
R1	Fa0/0	192.168.1.1	255.255.255.0
	S0/0/0	10.1.1.2	255.255.255.252
R2	S0/0/0	10.1.1.1	255.255.255.252
	Fa0/0	192.168.2.1	255.255.255.0
	S0/0/1	10.2.2.1	255.255.255.252
R3	S0/0/1	10.2.2.2	255.255.255.252
	Fa0/0	192.168.3.1	255.255.255.0
PC-A	NIC	192.168.1.3	255.255.255.0
PC-B	NIC	192.168.2.3	255.255.255.0
PC-C	NIC	192.168.3.3	255.255.255.0

Configure & Verify a Site-to-Site IPsec VPN

- We want to configure routers R1 and R3 to support a site-to-site IPsec VPN when traffic flows from their respective LANs.
- The IPsec VPN tunnel is from router R1 to router R3 via R2.
- R2 acts as a pass-through and has no knowledge of the VPN.

Configure & Verify a Site-to-Site IPsec VPN

- The routers are already configured with the following settings:
- Password for console line: **ciscoconpa55**
- Password for vty lines: **ciscovtypa55**
- Enable password: **ciscoenpa55**
- RIP version 2

Configure & Verify a Site-to-Site IPsec VPN

Task 1: Configure IPsec parameters on R1

- **Step 1. Test connectivity.**
- **Ping** from PC-A to PC-C.
- **Step 2. Identify interesting traffic on R1.**
- Configure ACL **110** to identify the traffic from the LAN on R1 to the LAN on R3 as interesting.
- This interesting traffic will trigger the IPsec VPN to be implemented whenever there is traffic between R1 to R3 LANs.

Configure & Verify a Site-to-Site IPsec VPN

- All other traffic sourced from the LANs will not be encrypted.
- R1(config)# **access-list 110 permit ip 192.168.1.0 0.0.0.255 192.168.3.0 0.0.0.255**
- **Step 3. Configure the ISAKMP Phase1 properties on R1.**
- Configure the crypto ISAKMP policy **10** properties on R1 along with the shared crypto key **vpnpa55**.
- Refer to the ISAKMP Phase 1 table for the specific parameters to configure.

Configure & Verify a Site-to-Site IPsec VPN

ISAKMP Phase 1 Policy Parameters

Parameters		R1	R3
Key distribution method	Manual or **ISAKMP**	**ISAKMP**	**ISAKMP**
Encryption algorithm	**DES**, 3DES, or AES	AES	AES
Hash algorithm	MD5 or **SHA-1**	**SHA-1**	**SHA-1**
Authentication method	Pre-shared keys or **RSA**	pre-share	pre-share
Key exchange	DH Group **1**, 2, or 5	DH 2	DH 2
IKE SA Lifetime	86400 seconds or less	**86400**	**86400**
ISAKMP Key		vpnpa55	vpnpa55

Note: Bolded parameters are defaults. Only unbolded parameters have to be explicitly configured.

Configure & Verify a Site-to-Site IPsec VPN

- We do not have to configure the default values.
- The encryption, key exchange method, and DH method must be configured.
- R1(config)# **crypto isakmp policy 10**
- R1(config-isakmp)# **encryption aes**
- R1(config-isakmp)# **authentication pre-share**
- R1(config-isakmp)# **group 2**
- R1(config-isakmp)# **exit**
- R1(config)# **crypto isakmp key vpnpa55 address 10.2.2.2**

Configure & Verify a Site-to-Site IPsec VPN

- **Step 4. Configure the ISAKMP Phase 2 properties on R1.**
- Create the transform-set **VPN-SET** to use **esp-3des** and **esp-sha-hmac**.
- Then create the crypto map **VPN-MAP** that binds all of the Phase 2 parameters together.
- Use sequence number **10** and identify it as an **ipsec-isakmp** map.
- R1(config)# **crypto ipsec transform-set VPN-SET esp-3des esp-sha-hmac**

Configure & Verify a Site-to-Site IPsec VPN

IPsec Phase 2 Policy Parameters

Parameters	R1	R3
Transform Set	VPN-SET	VPN-SET
Peer Hostname	R3	R1
Peer IP Address	10.2.2.2	10.1.1.2
Network to be encrypted	192.168.1.0/24	192.168.3.0/24
Crypto Map name	VPN-MAP	VPN-MAP
SA Establishment	ipsec-isakmp	ipsec-isakmp

Configure & Verify a Site-to-Site IPsec VPN

- R1(config)# **crypto map VPN-MAP 10 ipsec-isakmp**
- R1(config-crypto-map)# **description VPN connection to R3**
- R1(config-crypto-map)# **set peer 10.2.2.2**
- R1(config-crypto-map)# **set transform-set VPN-SET**
- R1(config-crypto-map)# **match address 110**
- R1(config-crypto-map)# **exit**

Configure & Verify a Site-to-Site IPsec VPN

- **Step 5. Configure the crypto map on the outgoing interface.**
- Finally, bind the **VPN-MAP** crypto map to the outgoing Serial 0/0/0 interface.
- R1(config)# **interface S0/0/0**
- R1(config-if)# **crypto map VPN-MAP**

Configure & Verify a Site-to-Site IPsec VPN

Task 2: Configure IPsec Parameters on R3

- **Step 1. Configure router R3 to support a site-to-site VPN with R1.**
- Now configure reciprocating parameters on R3. Configure ACL **110** identifying the traffic from the LAN on R3 to the LAN on R1 as interesting.
- R3(config)# **access-list 110 permit ip 192.168.3.0 0.0.0.255 192.168.1.0 0.0.0.255**
- **Step 2. Configure the ISAKMP Phase 1 properties on R3.**

Configure & Verify a Site-to-Site IPsec VPN

- Configure the crypto ISAKMP policy **10** properties on R3 along with the shared crypto key **vpnpa55**.
- R3(config)# **crypto isakmp policy 10**
- R3(config-isakmp)# **encryption aes**
- R3(config-isakmp)# **authentication pre-share**
- R3(config-isakmp)# **group 2**
- R3(config-isakmp)# **exit**
- R3(config)# **crypto isakmp key vpnpa55 address 10.1.1.2**

Configure & Verify a Site-to-Site IPsec VPN

- **Step 3. Configure the ISAKMP Phase 2 properties on R1.**
- Like we did on R1, create the transform-set **VPN-SET** to use **esp-3des** and **esp-sha-hmac**.
- Then create the crypto map **VPN-MAP** that binds all of the Phase 2 parameters together.
- Use sequence number **10** and identify it as an **ipsec-isakmp** map.
- R3(config)# **crypto ipsec transform-set VPN-SET esp-3des esp-sha-hmac**

Configure & Verify a Site-to-Site IPsec VPN

- R3(config)# **crypto map VPN-MAP 10 ipsec-isakmp**
- R3(config-crypto-map)# **description VPN connection to R1**
- R3(config-crypto-map)# **set peer 10.1.1.2**
- R3(config-crypto-map)# **set transform-set VPN-SET**
- R3(config-crypto-map)# **match address 110**
- R3(config-crypto-map)# **exit**

Configure & Verify a Site-to-Site IPsec VPN

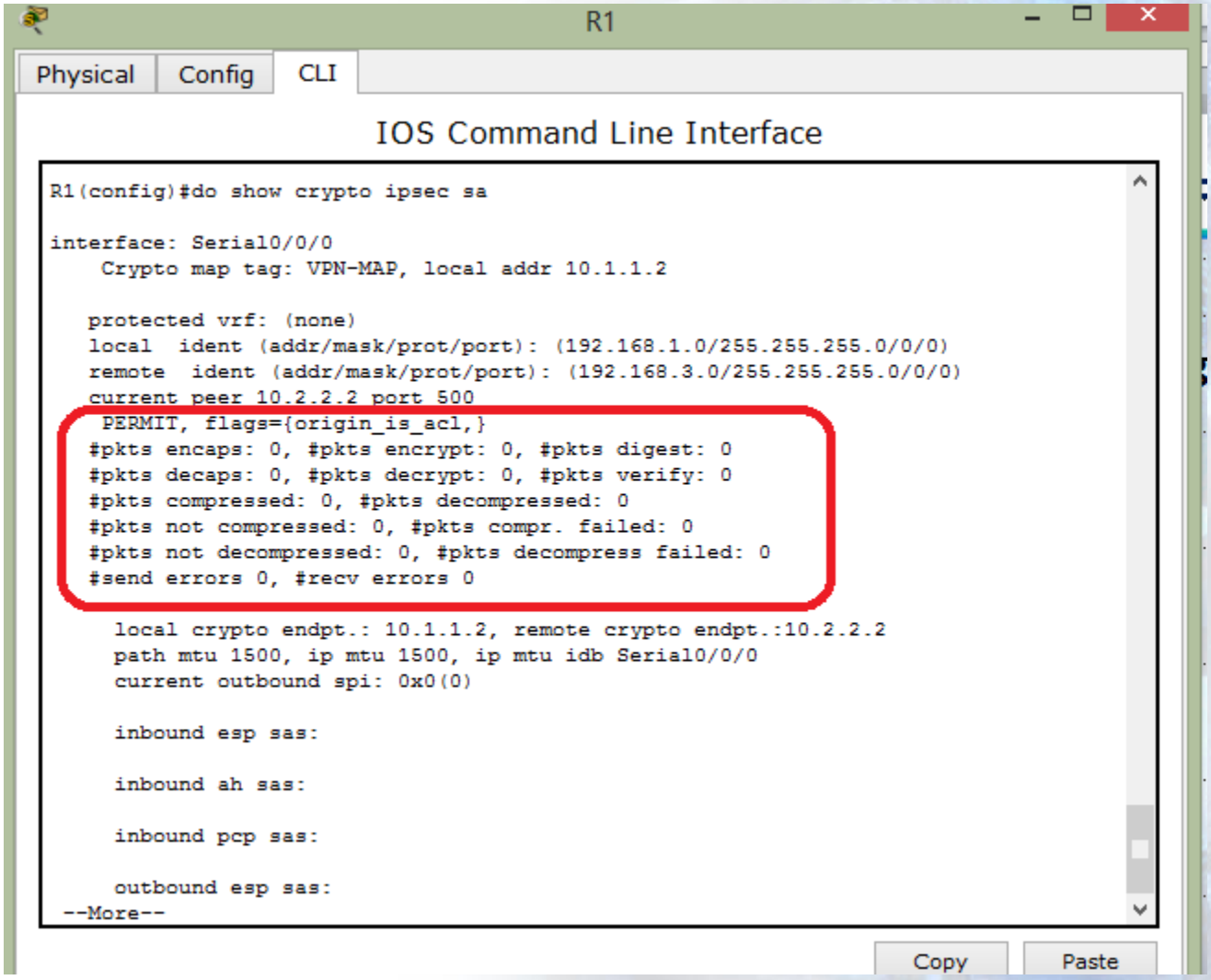
- **Step 4. Configure the crypto map on the outgoing interface.**
- Finally, bind the **VPN-MAP** crypto map to the outgoing Serial 0/0/1 interface.
- R3(config)# **interface S0/0/1**
- R3(config-if)# **crypto map VPN-MAP**

Configure & Verify a Site-to-Site IPsec VPN

Task 3: Verify the IPsec VPN

- **Step 1. Verify the tunnel prior to interesting traffic.**
- Issue the ***show crypto ipsec sa*** command on R1.
- Notice that the number of packets encapsulated, encrypted, decapsulated and decrypted are all set to 0.

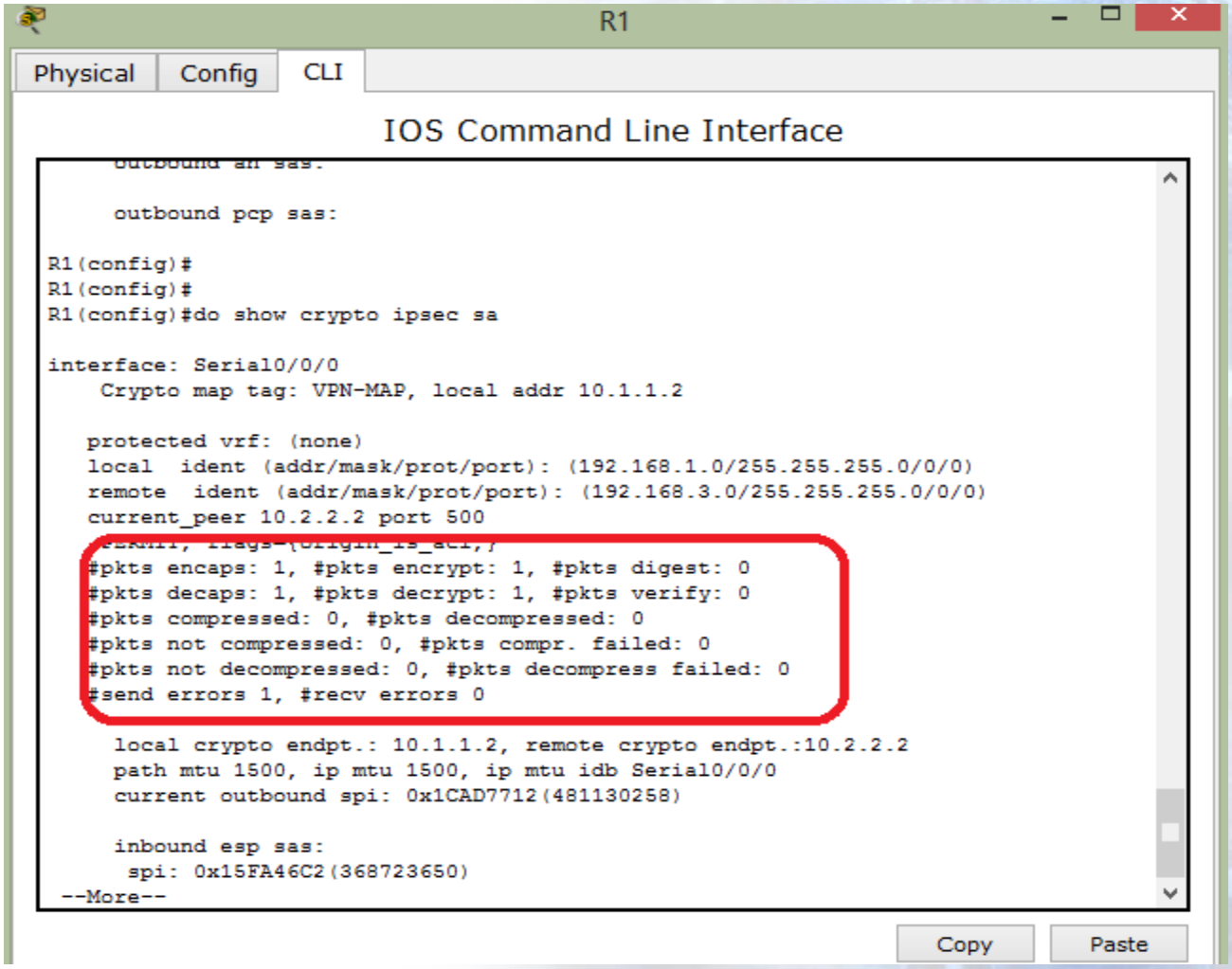
Configure & Verify a Site-to-Site IPsec VPN



Configure & Verify a Site-to-Site IPsec VPN

- **Step 2. Create interesting traffic.**
- From PC-A, **ping** PC-C.
- **Step 3. Verify the tunnel after interesting traffic.**
- On R1, re-issue the ***show crypto ipsec sa*** command. Now notice that the number of packets is more than 0 indicating that the IPsec VPN tunnel is working.

Configure & Verify a Site-to-Site IPsec VPN



Configure & Verify a Site-to-Site IPsec VPN

- **Step 4. Create uninteresting traffic .**
- From PC-A, **ping** PC-B.
- **Step 5. Verify the tunnel.**
- On R1, re-issue the ***show crypto ipsec sa***. We can find that number of packets has not changed.
- **Step 6. Check results.**
- Your completion percentage should be 100%.
- Click **Check Results** to see feedback and to see which components have been completed.

Configure a Network for Secure Operation-1

Network Security

Configure a Network for Secure Operation-1

Objectives of the Topic

- After completing this topic, a student will be able to
  - Configure a Network for Secure Operation in Packet tracer.

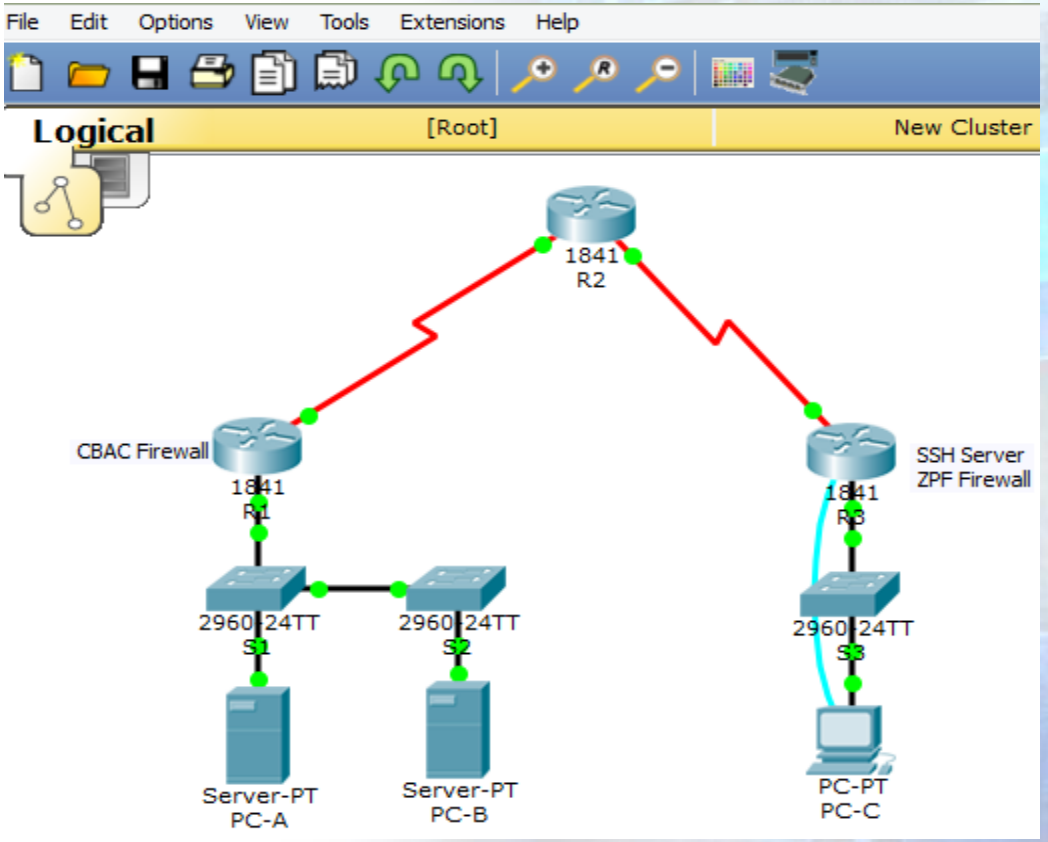
Configure a Network for Secure Operation-1

Figures and material in this topic have been adapted from

- *Chapter 9: PT Activity: Configure a Network for Secure Operation – "CCNA Security 1.1 Student Packet Tracer Manual", Cisco 2012.*

Configure a Network for Secure Operation-1

Experimental Setup: Lets assume a network topology consists of three routers R1, R2 and R3.



Configure a Network for Secure Operation-1

Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway	Switch Port
R1	FA0/1	192.168.1.1	255.255.255.0	N/A	S1 FA0/5
	S0/0/0 (DCE)	10.1.1.1	255.255.255.252	N/A	N/A
R2	S0/0/0	10.1.1.2	255.255.255.252	N/A	N/A
	S0/0/1 (DCE)	10.2.2.2	255.255.255.252	N/A	N/A
R3	FA0/1	192.168.3.1	255.255.255.0	N/A	S3 FA0/5
	S0/0/1	10.2.2.1	255.255.255.252	N/A	N/A
PC-A	NIC	192.168.1.5	255.255.255.0	192.168.1.1	S1 FA0/6
PC-B	NIC	192.168.1.6	255.255.255.0	192.168.1.1	S2 FA0/18
PC-C	NIC	192.168.3.5	255.255.255.0	192.168.3.1	S3 FA0/6

Configure a Network for Secure Operation-1

- In the topology, R1 is the edge router for the Company A while R3 is the edge router for Company B.
- These networks are interconnected via the R2 router which represents the ISP.
- The following preconfigurations have been made:
- Hostnames on all devices
- IP addresses on all devices
- R2 console password: **ciscoconpa55**

Configure a Network for Secure Operation-1

- R2 password on VTY lines: **ciscovtypa55**
- R2 enable password: **ciscoenpa55**
- Static routing
- Syslog services on PC-B
- DNS lookup has been disabled
- IP default gateways for all switches

Configure a Network for Secure Operation-1

Task 1: Test Connectivity and Verify Configurations

- **Step 1. Verify IP addresses.**
- R1# **show ip interface brief**
- R1# **show run**
- **Step 2. Verify routing tables.**
- R1# **show ip route**
- **Step 3. Test connectivity.**
- From PC-A, **ping** PC-C at IP address 192.168.3.5.

Configure a Network for Secure Operation-1

Task 2: Secure the Routers

- **Step 1. Set minimum a password length of 10 characters on router R1 and R3.**
- R1(config)# **security passwords min-length 10**
- R3(config)# **security passwords min-length 10**
- **Step 2. Configure an enable secret password on router R1 and R3.**
- Use an enable secret password of **ciscoenpa55.**
- R1(config)# **enable secret ciscoenpa55**
- R3(config)# **enable secret ciscoenpa55**

Configure a Network for Secure Operation-1

- **Step 3. Encrypt plaintext passwords .**
- R1(config)# **service password-encryption**
- R3(config)# **service password-encryption**
- **Step 4. Configure the console lines on R1 and R3.**
- Configure a console password of **ciscoconpa55** and enable login. Set the **exec-timeout** to log out after **5** minutes of inactivity. Prevent console messages from interrupting command entry.

Configure a Network for Secure Operation-1

- R1(config)# **line console 0**
- R1(config-line)# **password ciscoconpa55**
- R1(config-line)# **exec-timeout 5 0**
- R1(config-line)# **login**
- R1(config-line)# **logging synchronous**
- Repeat the above on R3.
- **Step 5. Configure vty lines on R1.**
- Configure a vty line password of **ciscovtypa55** and enable login. Set the **exec-timeout** to log out after **5** minutes of inactivity.

Configure a Network for Secure Operation-1

- Set the login authentication to use the default AAA list to be defined later.
- R1(config)# **line vty 0 4**
- R1(config-line)# **password ciscovtypa55**
- R1(config-line)# **exec-timeout 5 0**
- R1(config-line)# **login authentication default**
- **Step 6. Configure login banner on R1 and R3.**
- Configure a warning to unauthorized users with a message-of-the-day (MOTD) banner that says: “No Unauthorized Access!”.

Configure a Network for Secure Operation-1

- R1(config)# **banner motd $No Unauthorized Access!$**
- R3(config)# **banner motd $No Unauthorized Access!$**

Task 3: Configure Local Authentication on R1 and R3

- **Step 1. Configure the local user database.**
- Create a local user account of **Admin01** with a secret password of **Admin01pa55**.

Configure a Network for Secure Operation-1

- R1(config)# **username Admin01 privilege 15 secret Admin01pa55**
- R3(config)# **username Admin01 privilege 15 secret Admin01pa55**
- **Step 2. Enable AAA services.**
- R1(config)# **aaa new-model**
- R3(config)# **aaa new-model**
- **Step 3. Implement AAA services using the local database.**

Configure a Network for Secure Operation-1

- Create the default login authentication method list using local authentication with no backup method.
- R1(config)# **aaa authentication login default local none**
- R3(config)# **aaa authentication login default local none**

Task 4: Configure NTP

- **Step 1. Enable NTP authentication on PC-A.**

Configure a Network for Secure Operation-1

- On PC-A, choose the **Config** tab, and then the **NTP** button. Select **On** for NTP service. **Enable** authentication and enter a Key of **1** and a password of **ciscontppa55**.
- **Step 2. Configure R1 as an NTP Client.**
- Configure NTP authentication Key **1** with a password of **ciscontppa55**. Configure R1 to synchronize with the NTP server and authenticate using Key **1**.
- R1(config)# **ntp authenticate**

Configure a Network for Secure Operation-1

- R1(config)# **ntp authentication-key 1 md5 ciscontppa55**
- R1(config)# **ntp trusted-key 1**
- R1(config)# **ntp server 192.168.1.5 key 1**
- **Step 3. Configure routers to update hardware clock.**
- R1(config)# **ntp update-calendar**

Task 5: Configure R1 as Syslog Client

- **Step 1. Configure R1 to timestamp log messages .**

Configure a Network for Secure Operation-1

- R1(config)# **service timestamps log datetime msec**
- **Step 2. Configure R1 to log messages to the syslog server.**
- R1(config)# **logging 192.168.1.6**
- **Step 3. Check for syslog messages on PC-B.**
- On R1, exit config mode to generate a syslog message. Open the syslog server on PC-B to view the message sent from R1.

Configure a Network for Secure Operation-1

Task 6: Secure Router Against Login Attacks

- **Step 1. Log unsuccessful login attempts to R1.**
- R1(config)# **login on-failure log**
- **Step 2. Telnet to R1 from PC-A.**
- Provide the username **Admin01** and password **Admin01pa55**. The Telnet should be successful.
- **Step 3. Telnet to R1 from PC-A and check syslog messages on the syslog server.**
- Exit from the current Telnet session and Telnet again to R1.

Configure a Network for Secure Operation-1

- This time, use the username of **baduser** and any password. Check the syslog server on PC-B. You should see an error message.

Task 7: Configure SSH on R3

- **Step 1. Configure a domain name.**
- R3(config)# **ip domain-name ccnasecurity.com**
- **Step 2. Configure the incoming vty lines on R3.**
- Use the local user accounts for mandatory login and validation and accept only SSH connections.

Configure a Network for Secure Operation-1

- R3(config)# **line vty 0 4**
- R3(config-line)# **exec-timeout 5 0**
- R3(config-line)# **login local**
- R3(config-line)# **transport input ssh**
- **Step 3. Configure RSA encryption key pair for R3.**
- R3(config)# **crypto key zeroize rsa**
- % No Signature RSA Keys found in configuration.
- R3(config)# **crypto key generate rsa [Enter]**

Configure a Network for Secure Operation-1

- The name for the keys will be: R3.ccnasecurity.com
- Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose Keys. Choosing a key modulus greater than 512 may take a few minutes.
- How many bits in the modulus [512]:**1024**
- **% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]**
- **Step 4. Configure SSH timeouts and authentication parameters.**

Configure a Network for Secure Operation-1

- Set SSH timeout to **90** seconds, the number of authentication retries to **2**, and the version to **2**.
- R3(config)# **ip ssh time-out 90**
- R3(config)# **ip ssh authentication-retries 2**
- R3(config)# **ip ssh version 2**

Task 8: Configure CBAC on R1

- **Step 1. Configure a named IP ACL .**
- R1(config)# **ip access-list extended OUT-IN**
- R1(config-ext-nacl)# **deny ip any any**
- R1(config-ext-nacl)# **exit**

Configure a Network for Secure Operation-1

- R1(config)# **interface s0/0/0**
- R1(config-if)# **ip access-group OUT-IN in**
- **Step 2. Confirm that traffic entering interface Serial 0/0/0 is dropped.**
- From the PC-A command prompt, **ping** PC-C. The ICMP echo replies are blocked by the ACL.
- **Step 3. Create an inspection rule to inspect ICMP, Telnet and HTTP traffic.**

Configure a Network for Secure Operation-1

- R1(config)# **ip inspect name IN-OUT-IN icmp**
- R1(config)# **ip inspect name IN-OUT-IN telnet**
- R1(config)# **ip inspect name IN-OUT-IN http**
- **Step 4. Apply the inspect rule to the outside interface.**
- R1(config)# **interface s0/0/0**
- R1(config-if)# **ip inspect IN-OUT-IN out**
- **Step 5. Test operation of the inspection rule.**
- From the PC-A command prompt, **ping** PC-C.

Configure a Network for Secure Operation-1

- The ICMP echo replies should be inspected and allowed through.

Task 9: Configure ZPF on R3

- **Step 1. Test connectivity.**
- Verify that the internal host can access external resources.
- From PC-C, test connectivity with **ping** and Telnet to R2; all should be successful.
- From R2 **ping** to PC-C. The pings should be allowed.

Configure a Network for Secure Operation-1

- **Step 2. Create the firewall zones.**
- Create an internal zone named **IN-ZONE**.
- R3(config)# **zone security IN-ZONE**
- Create an external zone named **OUT-ZONE**.
- R3(config)# **zone security OUT-ZONE**
- **Step 3. Create an ACL that defines internal traffic.** Create an extended, numbered ACL that permits all IP protocols from 192.168.3.0/24 source network to any destination. Use **101** for the ACL number.

Configure a Network for Secure Operation-1

- R3(config)# **access-list 101 permit ip 192.168.3.0 0.0.0.255 any**
- **Step 4. Create a class map referencing the internal traffic ACL.**
- Create a class map named **IN-NET-CLASS-MAP** to match ACL 101.
- R3(config)# **class-map type inspect match-all IN-NET-CLASS-MAP**
- R3(config-cmap)# **match access-group 101**
- R3(config-cmap)# **exit**

Configure a Network for Secure Operation-1

- **Step 5. Specify firewall policies.**
- Create a policy map named **IN-2-OUT-PMAP** to determine what to do with matched traffic.
- R3(config)# **policy-map type inspect IN-2-OUT-PMAP**
- Specify a class type of **inspect** and reference class map **IN-NET-CLASS-MAP**.
- R3(config-pmap)# **class type inspect IN-NET-CLASS-MAP**
- Specify the action of **inspect** for this policy map

Configure a Network for Secure Operation-1

- R3(config-pmap-c)# **inspect**
- You should see the following console message:
- %No specific protocol configured in class IN-NET-CLASS-MAP for inspection. All protocols will be inspected."
- Exit to the global config prompt.
- R3(config-pmap-c)# **exit**
- R3(config-pmap)# **exit**
- **Step 6. Apply firewall policies.**

Configure a Network for Secure Operation-1

- Create a zone pair named **IN-2-OUT-ZPAIR**. Specify the source and destination zones that were created earlier.
- R3(config)# **zone-pair security IN-2-OUT-ZPAIR source IN-ZONE destination OUT-ZONE**
- Attach a policy map and actions to the zone pair referencing the policy map previously created, **IN-2-OUTPMAP**.
- R3(config-sec-zone-pair)# **service-policy type inspect IN-2-OUT-PMAP**

Configure a Network for Secure Operation-1

- Exit to the global config prompt and assign the internal and external interfaces to the security zones.
- R3(config)# **interface fa0/1**
- R3(config-if)# **zone-member security IN-ZONE**
- R3(config-if)# **interface s0/0/1**
- R3(config-if)# **zone-member security OUT-ZONE**
- **Step 7. Test firewall functionality.**

Configure a Network for Secure Operation-1

- Verify that the internal host can still access external resources.
- From PC-C, test connectivity with **ping** and Telnet to R2; all should be successful.
- From R2 **ping** to PC-C. The pings should now be blocked.

Configure a Network for Secure Operation-2

Network Security

Configure a Network for Secure Operation-2

Objectives of the Topic

- After completing this topic, a student will be able to
  - Configure a Network for Secure Operation in Packet tracer.

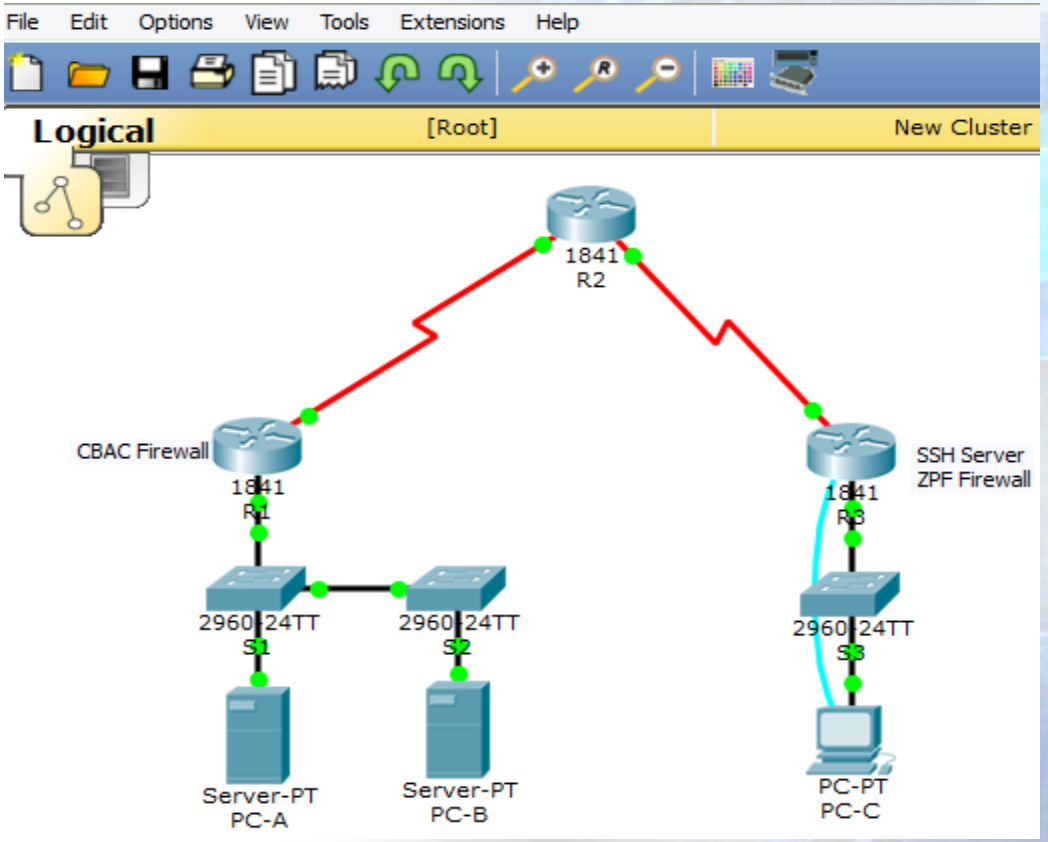
Configure a Network for Secure Operation-2

Figures and material in this topic have been adapted from

- *Chapter 9: PT Activity: Configure a Network for Secure Operation – "CCNA Security 1.1 Student Packet Tracer Manual", Cisco 2012.*

Configure a Network for Secure Operation-2

Experimental Setup: Lets assume a network topology consists of three routers R1, R2 and R3.



Configure a Network for Secure Operation-2

Task 10: Secure the Switches

- **Step 1. Configure an enable secret password on all switches.**
- Use an enable secret password of **ciscoenpa55.**
- S1(config)# **enable secret ciscoenpa55**
- **Step 2. Encrypt plaintext passwords .**
- S1(config)# **service password-encryption**
- **Step 3. Configure the console lines on all switches.**

Configure a Network for Secure Operation-2

- Configure a console password of **ciscoconpa55** and enable login. Set the **exec-timeout** to log out after **5** minutes of inactivity. Prevent console messages from interrupting command entry.
- S1(config)# **line console 0**
- S1(config-line)# **password ciscoconpa55**
- S1(config-line)# **exec-timeout 5 0**
- S1(config-line)# **login**
- S1(config-line)# **logging synchronous**

Configure a Network for Secure Operation-2

- **Step 4. Configure vty lines on all switches .**
- Configure a vty line password of **ciscovtypa55** and enable login. Set the **exec-timeout** to log out after **5** minutes of inactivity. Set the basic login parameter.
- S1(config)# **line vty 0 4**
- S1(config-line)# **password ciscovtypa55**
- S1(config-line)# **exec-timeout 5 0**
- S1(config-line)# **login**

Configure a Network for Secure Operation-2

- **Step 5. Secure trunk ports on S1 and S2.**
- Configure port Fa0/1 on S1 as a trunk port.
- S1(config)# **interface FastEthernet 0/1**
- S1(config-if)# **switchport mode trunk**
- Repeat for S2.
- Verify that S1 port Fa0/1 is in trunking mode.
- S1# **show interfaces trunk**
- Set the native VLAN on S1 and S2 trunk ports to an unused VLAN 99.

Configure a Network for Secure Operation-2

- S1(config)# **interface Fa0/1**
- S1(config-if)# **switchport trunk native vlan 99**
- S1(config-if)# **end**
- Repeat for S2.
- Set the trunk ports on S1 and S2 so that they do not negotiate by turning off the generation of DTP frames.
- S1(config)# **interface Fa0/1**
- S1(config-if)# **switchport nonegotiate**
- Repeat for S2.

Configure a Network for Secure Operation-2

- Enable storm control for broadcasts on the S1 and S2 trunk ports with a 50 percent rising suppression level.
- S1(config)# **interface FastEthernet 0/1**
- S1(config-if)# **storm-control broadcast level 50**
- Repeat for S2.
- **Step 6. Secure access ports.**
- S1(config)# **interface FastEthernet 0/5**
- S1(config-if)# **switchport mode access**

Configure a Network for Secure Operation-2

- Repeat for S1 **FastEthernet 0/6,** S2 **FastEthernet 0/18,** S3 **FastEthernet 0/5**, and S3 **FastEthernet 0/6.**
- Enable PortFast on S1, S2, and S3 access ports.
- S1(config)# **interface FastEthernet 0/5**
- S1(config-if)# **spanning-tree portfast**
- Repeat for S1 **FastEthernet 0/6,** S2 **FastEthernet 0/18,** S3 **FastEthernet 0/5**, and S3 **FastEthernet 0/6.**

Configure a Network for Secure Operation-2

- Enable BPDU guard on the switch ports previously configured as access only.
- S1(config)# **interface FastEthernet 0/5**
- S1(config-if)# **spanning-tree bpduguard enable**
- Repeat for S1 **FastEthernet 0/6,** S2 **FastEthernet 0/18,** S3 **FastEthernet 0/5**, and S3 **FastEthernet 0/6.**
- Enable basic default port security on all end-user access ports that are in use. Use the **sticky** option. Reenable each access port to which port security was applied.

Configure a Network for Secure Operation-2

- S1(config)# **interface FastEthernet 0/5**
- S1(config-if)# **shutdown**
- S1(config-if)# **switchport port-security**
- S1(config-if)# **switchport port-security mac-address sticky**
- S1(config-if)# **no shutdown**
- Repeat for S1 **FastEthernet 0/6,** S2 **FastEthernet 0/18,** S3 **FastEthernet 0/5**, and S3 **FastEthernet 0/6.**

Configure a Network for Secure Operation-2

- Disable any ports not being used on each switch.

```
S1(config)# interface range Fa0/2 - 4
S1(config-if-range)# shutdown
S1(config-if-range)# interface range Fa0/7 - 24
S1(config-if-range)# shutdown
S1(config-if-range)# interface range gigabitethernet1/1 - 2
S1(config-if-range)# shutdown

S2(config)# interface range Fa0/2 - 17
S2(config-if-range)# shutdown
S2(config-if-range)# interface range Fa0/19 - 24
S2(config-if-range)# shutdown
S3(config-if-range)# interface range gigabitethernet1/1 - 2
S2(config-if-range)# shutdown

S3(config)# interface range Fa0/1 - 4
S3(config-if-range)# shutdown
S3(config-if-range)# interface range Fa0/7 - 24
S3(config-if-range)# shutdown
S3(config-if-range)# interface range gigabitethernet1/1 - 2
S3(config-if-range)# shutdown
```

Configure a Network for Secure Operation-2

Task 11: Verification

- **Step 1. Test SSH configuration.**
- Attempt to connect to R3 via Telnet from PC-C.
- From PC-C, enter the command to connect to R3 via Telnet at IP address 192.168.3.1. This connection should fail, since R3 has been configured to accept only SSH connections on the virtual terminal lines.

Configure a Network for Secure Operation-2

- From PC-C, enter the **ssh –l Admin01 192.168.3.1** command to connect to R3 via SSH.
- When prompted for the password, enter the password **Admin01pa55** configured for the local administrator.
- Use the **show ip ssh** command to see the configured settings.
- R3# **show ip ssh**
- SSH Enabled - version 2.0, Authentication timeout: 90 secs; Authentication retries: 2

Configure a Network for Secure Operation-2

- **Step 2. Verify timestamps , NTP status for R1 and PC-A.**
- R1# **show clock**
- R1# **show ntp status**
- **Step 3. Test CBAC firewall on R1.**
- **Ping** from PC-A to R2 at 10.2.2.2 (should succeed). Telnet from PC-A to R2 10.2.2.2 (should succeed). **Ping** from R2 to PC-A at 192.168.1.3 (should fail)

Configure a Network for Secure Operation-2

- **Step 4. Test ZPF firewall on R3.**
- **Ping** from PC-C to R2 at 10.2.2.2 (should succeed)
- Telnet from PC-C to R2 at 10.2.2.2 (should succeed)
- **Ping** from R2 to PC-C at 192.168.3.5 (should fail)
- Telnet from R2 to R3 at 10.2.2.1 (should fail – only SSH is allowed)

Configure a Network for Secure Operation-2

- **Step 5. Verify port security.**
- On S2, use the **show run** command to confirm that S2 has added a sticky MAC address for Fa0/18. This should be the MAC address of PC-B. Record the MAC address for later use.
- S2#**show run**
- Select PC-B. Go to the **Config** tab. Select **FastEthernet** under the Interface section. Edit the MAC address field. For example, change it from 0001.435D.3057 to 0001.435D.AAAA.

Configure a Network for Secure Operation-2

- This should cause a port security violation and S2 should shut down port Fa0/18.
- Use the **show interface Fa0/18** command to view the status of the port. The port should be in the err-disabled state.
- S2#**show int fa0/18**
- S2#**show port-security**
- On PC-B, go to the **Config** tab. Select **FastEthernet** under the Interface section. Change the MAC address to another address.

Configure a Network for Secure Operation-2

- For example, change it from 0001.435D.AAAA to 0001.435D.BBBB.
- From interface configuration mode on switch S2 for Fa0/18, use the **no switchport port-security macaddress sticky *address*** command to remove the original PC-B learned address.
- S2(config)# **int fa0/18**
- S2(config-if)# **no switchport port-security mac-address sticky 0001.435D.3057**

Configure a Network for Secure Operation-2

- Shutdown and then re-enable the Fa0/18 interface.
- S2(config)# **int fa0/18**
- S2(config-if)# **shutdown**
- S2(config-if)# **no shutdown**
- On S2, use the **show run** command to confirm that the port comes up and that the new MAC address has been learned.
- S2#**show run**

Configure a Network for Secure Operation-2

- **Step 6. Check results.**
- Your completion percentage should be 100%. Click **Check Results** to see feedback and verification of which required components have been completed.