

Key Management

In cryptography it is a very tedious task to distribute the public and private key between sender and receiver. If key is known to the third party (forger/eavesdropper) then the whole security mechanism becomes worthless. So, there comes the need to secure the exchange of keys.

There are 2 aspects for Key Management:

- ❑ Distribution of public keys.
- ❑ Use of public-key encryption to distribute secret

Key Management Techniques

- ❑ Symmetric-key encryption
- ❑ Public-key encryption

Symmetric Key Distribution Using Symmetric Encryption

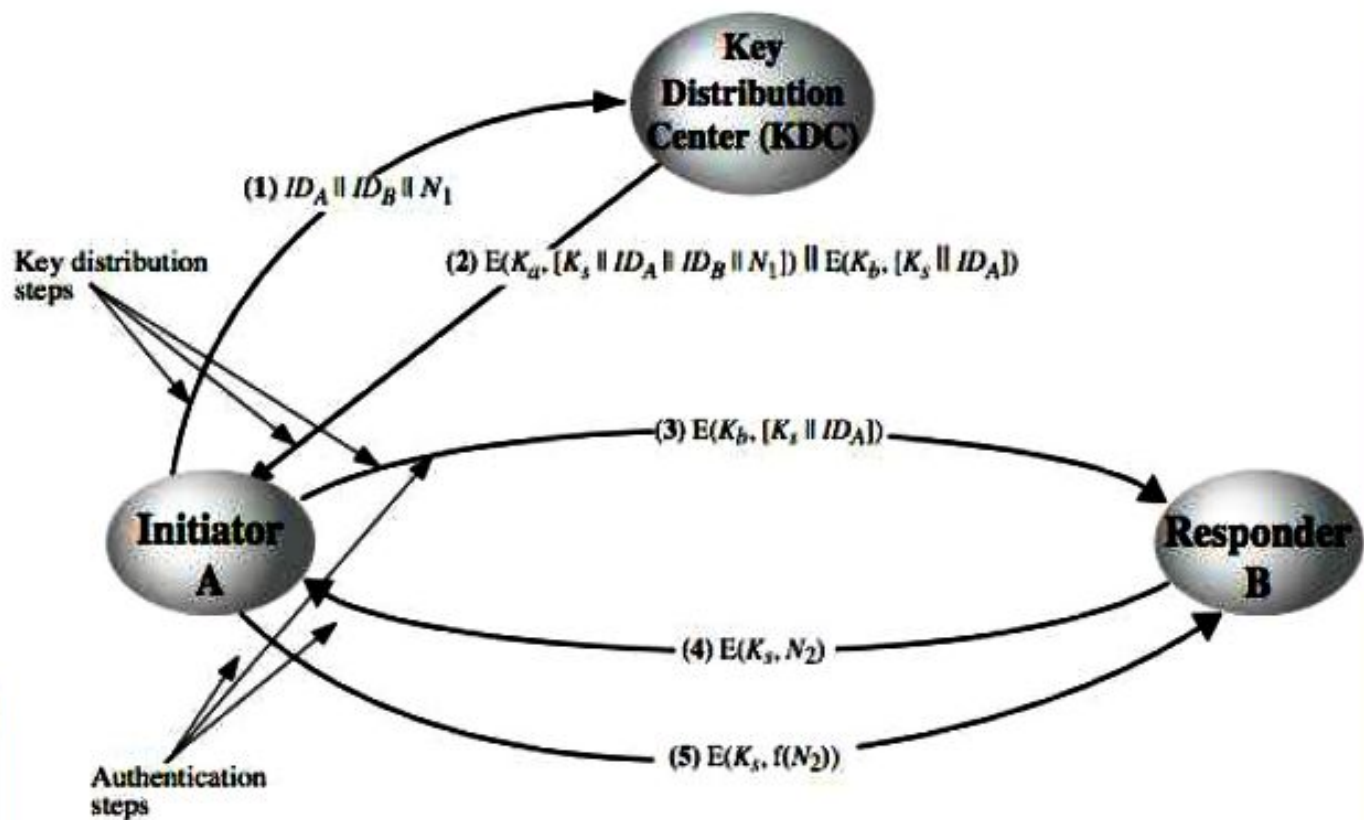
For **symmetric** encryption to work, the two parties to an exchange must share the same key, and that key must be protected from access by others.

Therefore, the term that refers to the means of delivering a key to two parties who wish to exchange data, without allowing others to see the key.

Key Distribution

- ⌘ Given parties A and B have various **key distribution** alternatives:
 - ⌘ A can select key and physically deliver to B
 - ⌘ third party can select & deliver key to A & B
 - ⌘ if A & B have communicated previously can use previous key to encrypt a new key
 - ⌘ if A & B have secure communications with a third party C, C can relay key between A & B

Key Distribution Scenario



6

Major Issues with KDC

Hierarchical Key Control

It is not necessary to limit the key distribution function to a single KDC. Indeed, for very large networks, it may not be practical to do so. As an alternative, a hierarchy of KDCs can be established.

A hierarchical scheme minimizes the effort involved in master key distribution, because most master keys are those shared by a local KDC with its local entities.

Session Key Lifetime

✎ The distribution of session keys delays the start of any exchange and places a burden on network capacity. A security manager must try to balance these competing considerations in determining the lifetime of a particular session key.

A Transparent Key Control Scheme

- ✎ The approach useful for providing end-to-end encryption at a network or transport level in a way that is transparent to the end users.
- ✎ The approach assumes that communication makes use of a connection-oriented end-to-end protocol, such as TCP.

Decentralized Key Control

- ✎ The use of a key distribution center imposes the requirement that the KDC be trusted and be protected from subversion. This requirement can be avoided if key distribution is fully decentralized.
- ✎ Although full decentralization is not practical for larger networks using symmetric encryption only, it may be useful within a local context.
- ✎ A decentralized approach requires that each end system be able to communicate in a secure manner with all potential partner end systems for purposes of session key distribution

Controlling Key Usage

✧ The concept of a key hierarchy and the use of automated key distribution techniques greatly reduce the number of keys that must be manually managed and distributed. It also may be desirable to impose some control on the way in which automatically distributed keys are used. For example, in addition to separating master keys from session keys, we may wish to define different types of session keys on the basis of use, such as

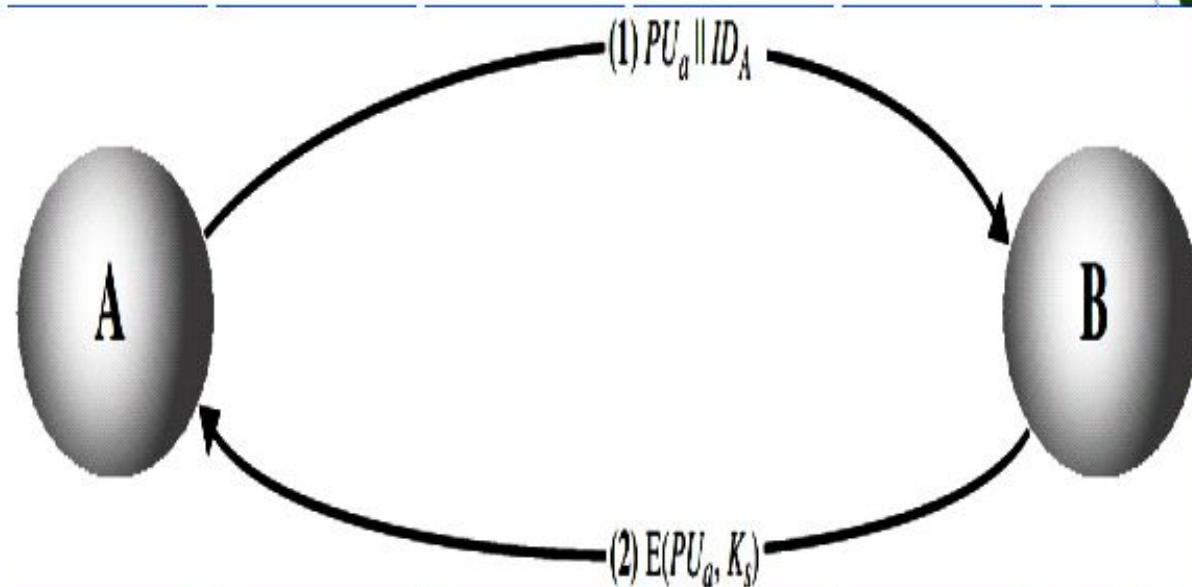
- ✧ Data-encrypting key, for general communication across a network
- ✧ PIN-encrypting key, for personal identification numbers (PINs) used in electronic funds transfer and point-of-sale applications
- ✧ File-encrypting key, for encrypting files stored in publicly accessible locations

10

Symmetric Key Distribution Using Asymmetric Encryption

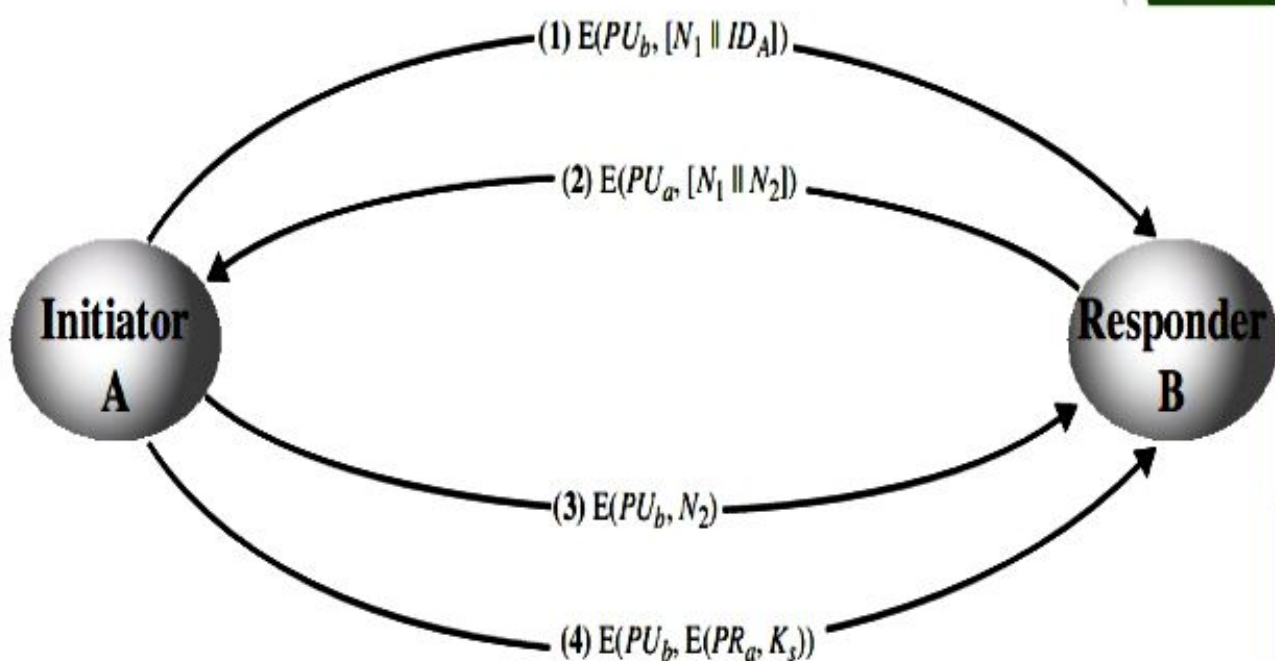
- ✧ Once public keys have been distributed or have become accessible, secure communication that thwarts eavesdropping, tampering, or both, is possible.
- ✧ Public-key encryption provides for the distribution of secret keys to be used for conventional encryption

Simple Secret Key Distribution



12

Secret Key Distribution with Confidentiality and Authentication



Distribution of Public Keys

Several techniques have been proposed for the distribution of public keys, which can mostly be grouped into the categories shown:

- ⊠ Public announcement
- ⊠ Publicly available directory
- ⊠ Public-key authority
- ⊠ Public-key certificates

Public Announcement

Users distribute public keys to recipients or broadcast to community at large

- ⊠ Eg. append PGP keys to email messages or post to news groups or email list

major weakness is forgery

- ⊠ anyone can create a key claiming to be someone else and broadcast it
- ⊠ until forgery is discovered can masquerade as claimed user.

Publicly Available Directory

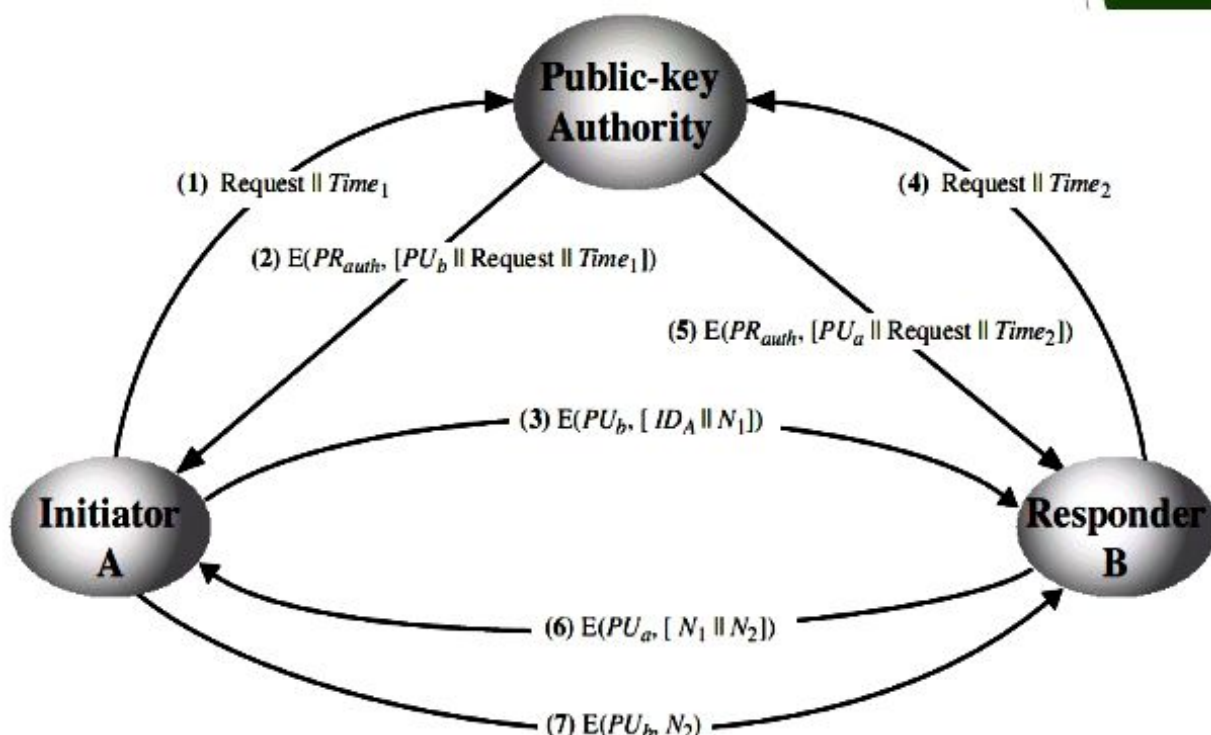
Can obtain greater security by registering keys with a public directory

Directory must be trusted with properties:

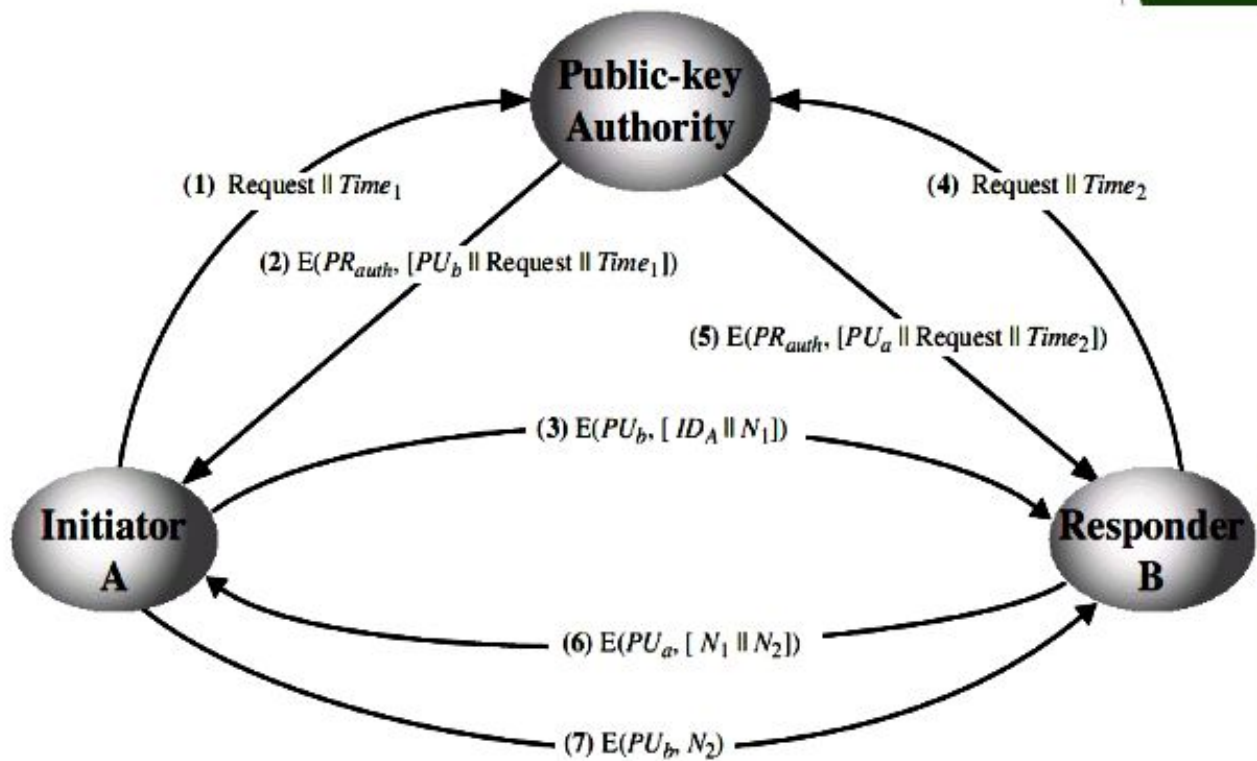
- contains {name,public-key} entries
- participants register securely with directory
- participants can replace key at any time
- directory is periodically published
- directory can be accessed electronically

This scheme is clearly more secure than individual public announcements but still has vulnerabilities.

Public-Key Authority

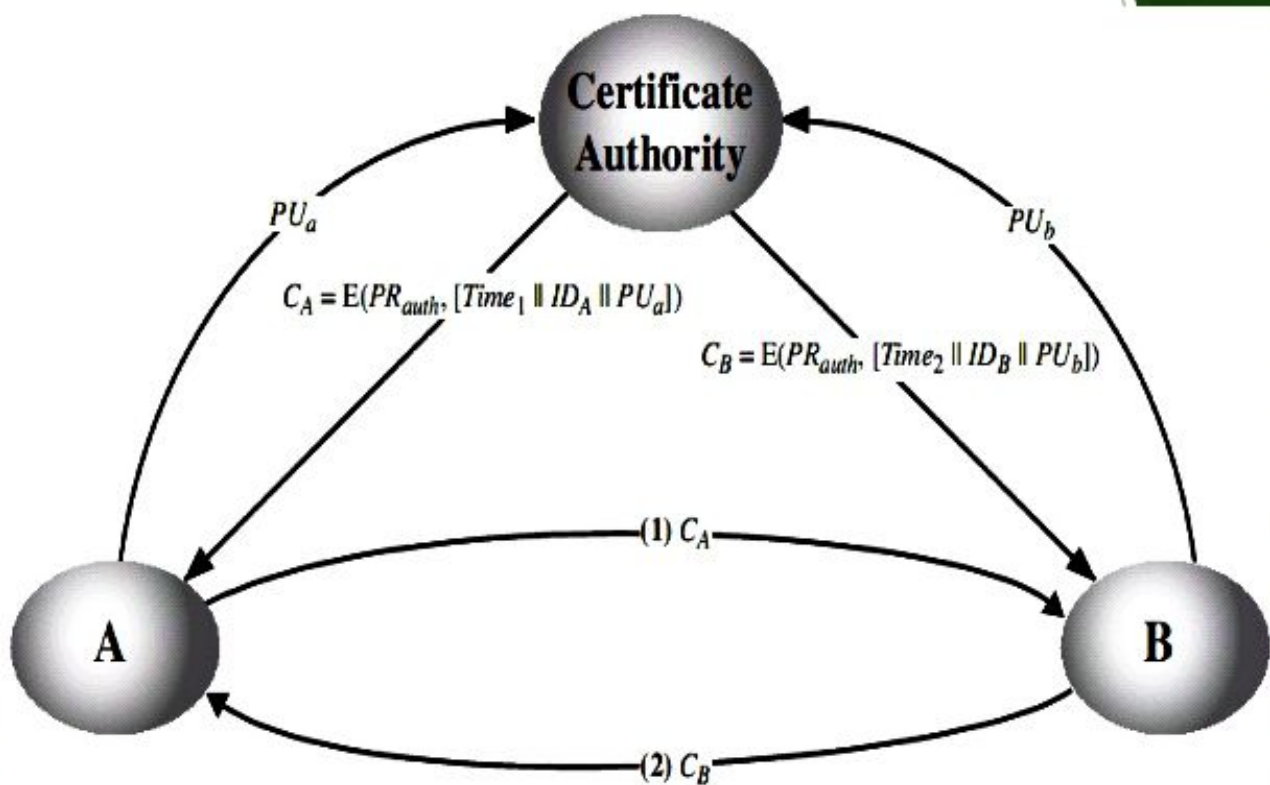


Public-Key Authority



17

Public-Key Certificates



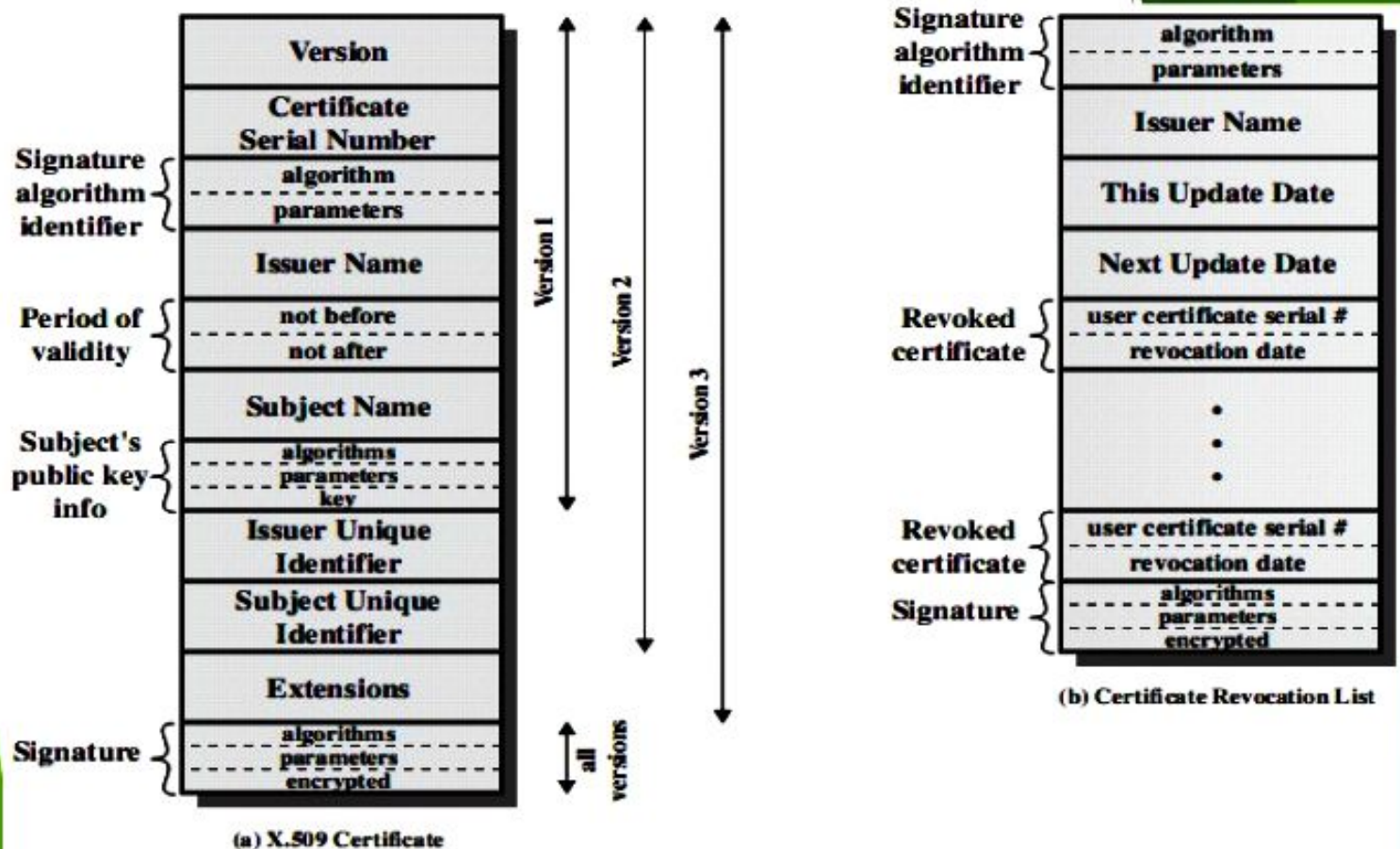
X.509 Certificate

- ❑ In cryptography, a **public key certificate**, also known as a **digital certificate** or **identity certificate**, is an electronic document used to prove the ownership of a public key.
- ❑ The certificate includes information about the key, information about the identity of its owner (called the subject), and the digital signature of an entity that has verified the certificate's contents (called the issuer).
- ❑ If the signature is valid, and the software examining the certificate trusts the issuer, then it can use that key to communicate securely with the certificate's subject.
- ❑ The most common format for public key certificates is defined by X.509 Certificate.

X.509

- ❑ X.509 is part of the X.500 series of recommendations that define a directory service, being a server or distributed set of servers that maintains a database of information about users.
- ❑ X.509 defines a framework for the provision of authentication services by the X.500 directory to its users.
- ❑ The directory may serve as a repository of public-key certificates.
- ❑ Each certificate contains the public key of a user and is signed with the private key of a trusted certification authority.
- ❑ X.509 defines alternative authentication protocols based on the use of public-key certificates.
- ❑ X.509 is based on the use of public-key cryptography and digital signatures.

X.509 Format



- ❑ **Version:** Default is version 1. If the issuer unique identifier or subject unique identifier are present, the value must be version 2. If one or more extensions are present, the version must be version 3.
- ❑ **Serial number:** An integer value unique within the issuing CA that is unambiguously associated with this certificate.
- ❑ **Signature algorithm identifier:** The algorithm used to sign the certificate together with any associated parameters.
- ❑ **Issuer name:** X.500 is the name of the CA that created and signed this certificate.
- ❑ **Period of validity:** Consists of two dates: the first and last on which the certificate is valid.

- ❏ **Subject name:** The name of the user to whom this certificate refers. That is, this certificate certifies the public key of the subject who holds the corresponding private key.
- ❏ **Subject's public-key information:** The public key of the subject, plus an identifier of the algorithm for which this key is to be used, together with any associated parameters.
- ❏ **Issuer unique identifier:** An optional-bit string field used to identify uniquely the issuing CA in the event the X.500 name has been reused for different entities.
- ❏ **Subject unique identifier:** An optional-bit string field used to identify uniquely the subject in the event the X.500 name has been reused for different entities.
- ❏ **Extensions:** A set of one or more extension fields. Extensions were added in version 3 and are discussed later in this section.
- ❏ **Signature:** Covers all of the other fields of the certificate; it contains the hash code of the other fields encrypted with the CA's private key. This field includes the signature algorithm identifier.

Notation to Define A Certificate

$$\text{CA} \langle \langle A \rangle \rangle = \text{CA} \{V, SN, AI, CA, UCA, A, UA, Ap, T^A\}$$

Where:

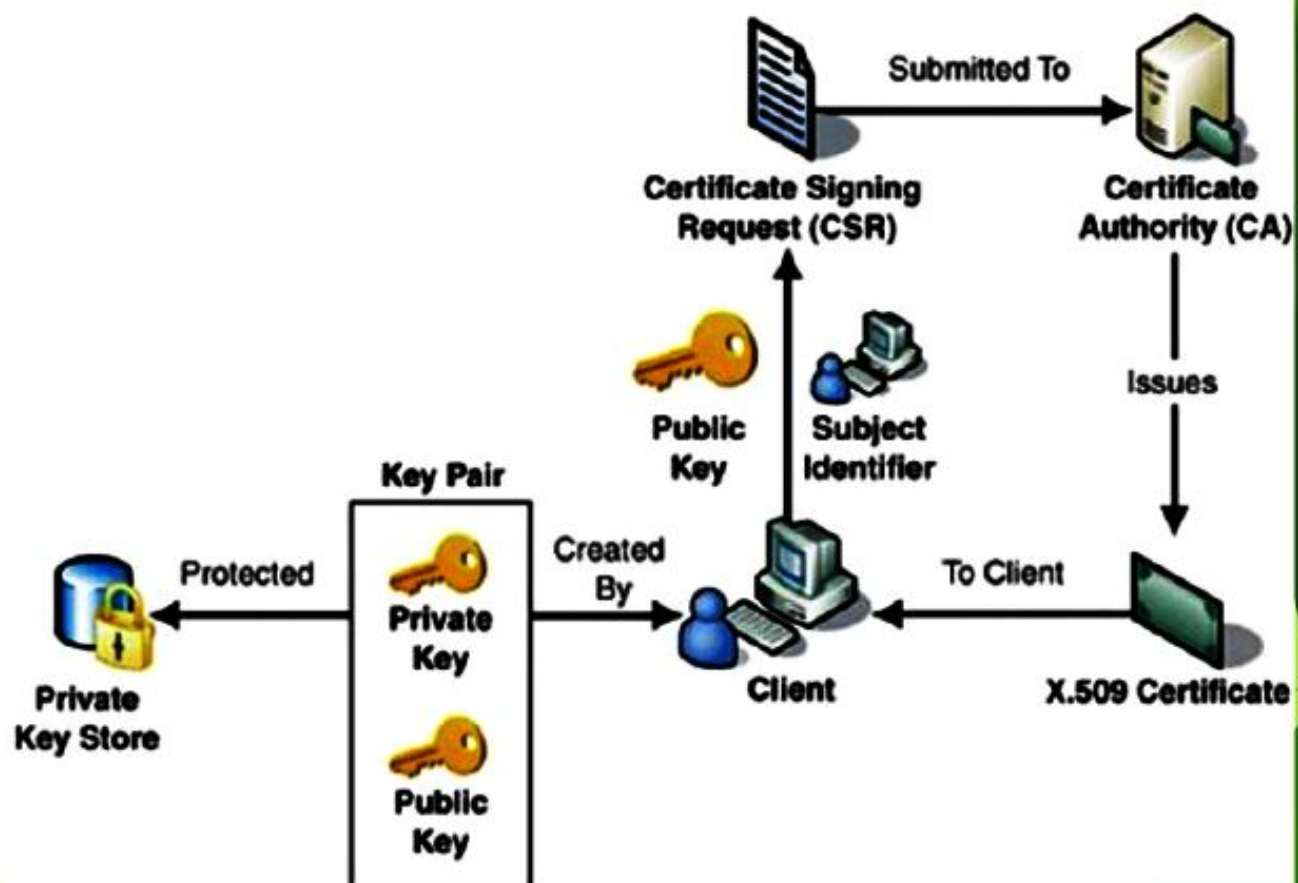
- ❏ $Y \langle \langle X \rangle \rangle$ = the certificate of user X issued by certification authority Y $Y(I)$ = the signing of I by Y. It consists of I with an encrypted hash code appended
- ❏ V = version of the certificate
- ❏ SN = serial number of the certificate
- ❏ AI = identifier of the algorithm used to sign the certificate
- ❏ CA = name of certificate authority
- ❏ UCA = optional unique identifier of the CA
- ❏ A = name of user A
- ❏ UA = optional unique identifier of the user A
- ❏ Ap = public key of user A
- ❏ T^A = period of validity of the certificate

Obtaining a 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.



Public Key Infrastructure (PKI)

- ✘ The most distinct feature of Public Key Infrastructure (PKI) is that it uses a pair of keys to achieve the underlying security service.
- ✘ The key pair comprises of private key and public key.

There are two specific requirements of key management for public key cryptography.

- ✘ **Secrecy of private keys.** Throughout the key lifecycle, secret keys must remain secret from all parties except those who are owner and are authorized to use them.
- ✘ **Assurance of public keys.** In public key cryptography, the public keys are in open domain and seen as public pieces of data. By default there are no assurances of whether a public key is correct, with whom it can be associated, or what it can be used for. Thus key management of public keys needs to focus much more explicitly on assurance of purpose of public keys

PKI

PKI provides assurance of public key. It provides the identification of public keys and their distribution. An anatomy of PKI comprises of the following components.

- ✘ Public Key Certificate, commonly referred to as 'digital certificate'.
- ✘ Private Key tokens.
- ✘ Certification Authority.
- ✘ Registration Authority.
- ✘ Certificate Management System.

Digital Certificate/Public Key Certificate

- ❏ Digital certificates are based on the ITU standard X.509 which defines a standard certificate format for public key certificates and certification validation. Hence digital certificates are sometimes also referred to as X.509 certificates.
- ❏ Public key pertaining to the user client is stored in digital certificates by The Certification Authority (CA) along with other relevant information such as client information, expiration date, usage, issuer etc.
- ❏ CA digitally signs this entire information and includes digital signature in the certificate.
- ❏ Anyone who needs the assurance about the public key and associated information of client, he carries out the signature validation process using CA's public key. Successful validation assures that the public key given in the certificate belongs to the person whose details are given in the certificate.

Certifying Authority (CA)

- ❏ **Generating key pairs** – The CA may generate a key pair independently or jointly with the client.
- ❏ **Issuing digital certificates** – CA issues a certificate after client provides the credentials to confirm his identity. The CA then signs the certificate to prevent modification of the details contained in the certificate.
- ❏ **Publishing Certificates** – The CA need to publish certificates so that users can find them. There are two ways of achieving this. One is to publish certificates in the equivalent of an electronic telephone directory. The other is to send your certificate out to those people you think might need it by one means or another.
- ❏ **Verifying Certificates** – The CA makes its public key available in environment to assist verification of his signature on clients' digital certificate.
- ❏ **Revocation of Certificates** – At times, CA revokes the certificate issued due to some reason such as compromise of private key by user or loss of trust in the client. After revocation, CA maintains the list of all revoked certificate that is available to the environment.

Classes of Certificates

- ✘ **Class 1** – These certificates can be easily acquired by supplying an email address.
- ✘ **Class 2** – These certificates require additional personal information to be supplied.
- ✘ **Class 3** – These certificates can only be purchased after checks have been made about the requestor's identity.
- ✘ **Class 4** – They may be used by governments and financial organizations needing very high levels of trust.

Registration Authority (RA)

- ✘ CA may use a third-party Registration Authority (RA) to perform the necessary checks on the person or company requesting the certificate to confirm their identity.
- ✘ The RA may appear to the client as a CA, but they do not actually sign the certificate that is issued.

Certificate Management System (CMS)

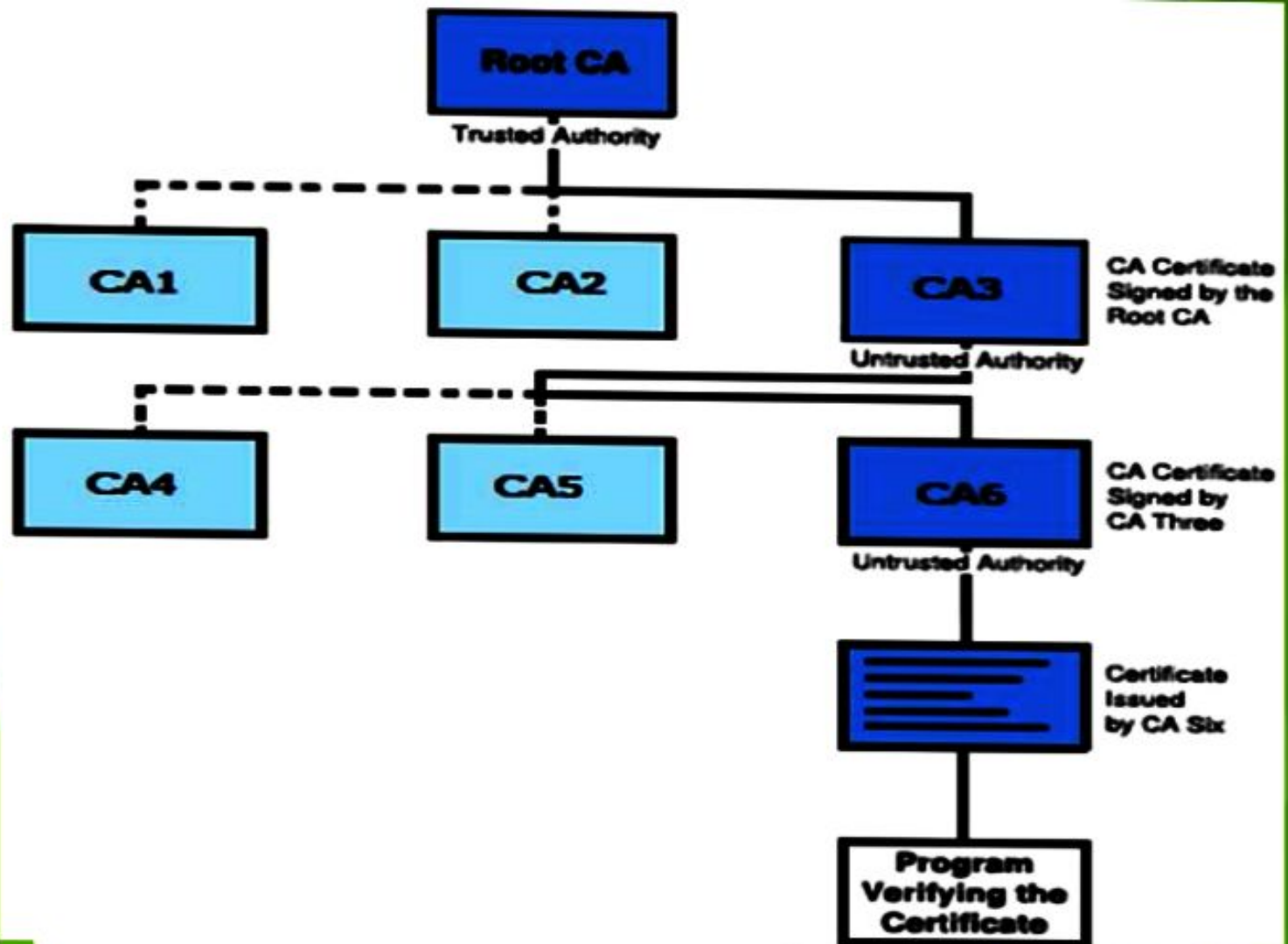
- ✘ It is the management system through which certificates are published, temporarily or permanently suspended, renewed, or revoked.
- ✘ Certificate management systems do not normally delete certificates because it may be necessary to prove their status at a point in time, perhaps for legal reasons.
- ✘ A CA along with associated RA runs certificate management systems to be able to track their responsibilities and liabilities.

Private Key Tokens

- While the public key of a client is stored on the certificate, the associated secret private key can be stored on the key owner's computer.
- This method is generally not adopted. If an attacker gains access to the computer, he can easily gain access to private key.
- For this reason, a private key is stored on secure removable storage token access to which is protected through a password.
- Different vendors often use different and sometimes proprietary storage formats for storing keys.
- For example, Entrust uses the proprietary .epf format, while Verisign, GlobalSign, and Baltimore use the standard .p12 format.

Hierarchy of CA

- The root CA is at the top of the CA hierarchy and the root CA's certificate is a self-signed certificate.
- The CAs, which are directly subordinate to the root CA (For example, CA1 and CA2) have CA certificates that are signed by the root CA.
- The CAs under the subordinate CAs in the hierarchy (For example, CA5 and CA6) have their CA certificates signed by the higher-level subordinate CAs.



Verifying a certificate chain is the process of ensuring that a specific certificate chain is valid, correctly signed, and trustworthy. The following procedure verifies a certificate chain, beginning with the certificate that is presented for authentication –

- A client whose authenticity is being verified supplies his certificate, generally along with the chain of certificates up to Root CA.
- Verifier takes the certificate and validates by using public key of issuer. The issuer's public key is found in the issuer's certificate which is in the chain next to client's certificate.
- Now if the higher CA who has signed the issuer's certificate, is trusted by the verifier, verification is successful and stops here.
- Else, the issuer's certificate is verified in a similar manner as done for client in above steps. This process continues till either trusted CA is found in between or else it continues till Root CA.

Remote User-Authentication Principles

- ✘ Fundamental security building block
 - ✘ basis of access control & user accountability
- ✘ Is the process of verifying an identity claimed by or for a system entity
- ✘ Has two steps:
 - ✘ identification - specify identifier
 - ✘ verification - bind entity (person) and identifier
- ✘ Distinct from message authentication

Means of User Authentication

There are four general means of authenticating a user's identity, which can be used alone or in combination

- ✘ **Something the individual knows**
 - ✘ Examples include a password, a personal identification number (PIN), or answers to a prearranged set of questions
- ✘ **Something the individual possesses**
 - ✘ Examples include cryptographic keys, electronic keycards, smart cards, and physical keys
 - ✘ This is referred to as a token
- ✘ **Something the individual is (static biometrics)**
 - ✘ Examples include recognition by fingerprint, retina, and face
- ✘ **Something the individual does (dynamic biometrics)**
 - ✘ Examples include recognition by voice pattern, handwriting characteristics, and typing rhythm

Mutual Authentication

- ✘ Used to convince parties of each others identity and to exchange session keys.
- ✘ may be one-way or mutual

key issues are:

- ✘ **Confidentiality – to protect session keys**
- ✘ **Timeliness – to prevent replay attacks**

Replay Attacks

- ✘ **The simplest replay attack is one in which the opponent simply copies a message and replays it later**
- ✘ An opponent can replay a timestamped message within the valid time window
- ✘ An opponent can replay a timestamped message within the valid time window, but in addition, the opponent suppresses the original message; thus, the repetition cannot be detected
- ✘ Another attack involves a backward replay without modification and is possible if symmetric encryption is used and the sender cannot easily recognize the difference between messages sent and messages received on the basis of content

Approaches to Coping With Replay Attacks

Attach a sequence number to each message used in an authentication exchange

- ▣ A new message is accepted only if its sequence number is in the proper order
- ▣ Difficulty with this approach is that it requires each party to keep track of the last sequence number for each claimant it has dealt with
- ▣ Generally not used for authentication and key exchange because of overhead

Timestamps

- ▣ Requires that clocks among the various participants be synchronized
- ▣ Party A accepts a message as fresh only if the message contains a timestamp that, in A's judgment, is close enough to A's knowledge of current time

Challenge/response

Party A, expecting a fresh message from B, first sends B a nonce (challenge) and requires that the subsequent message (response) received from B contain the correct nonce value

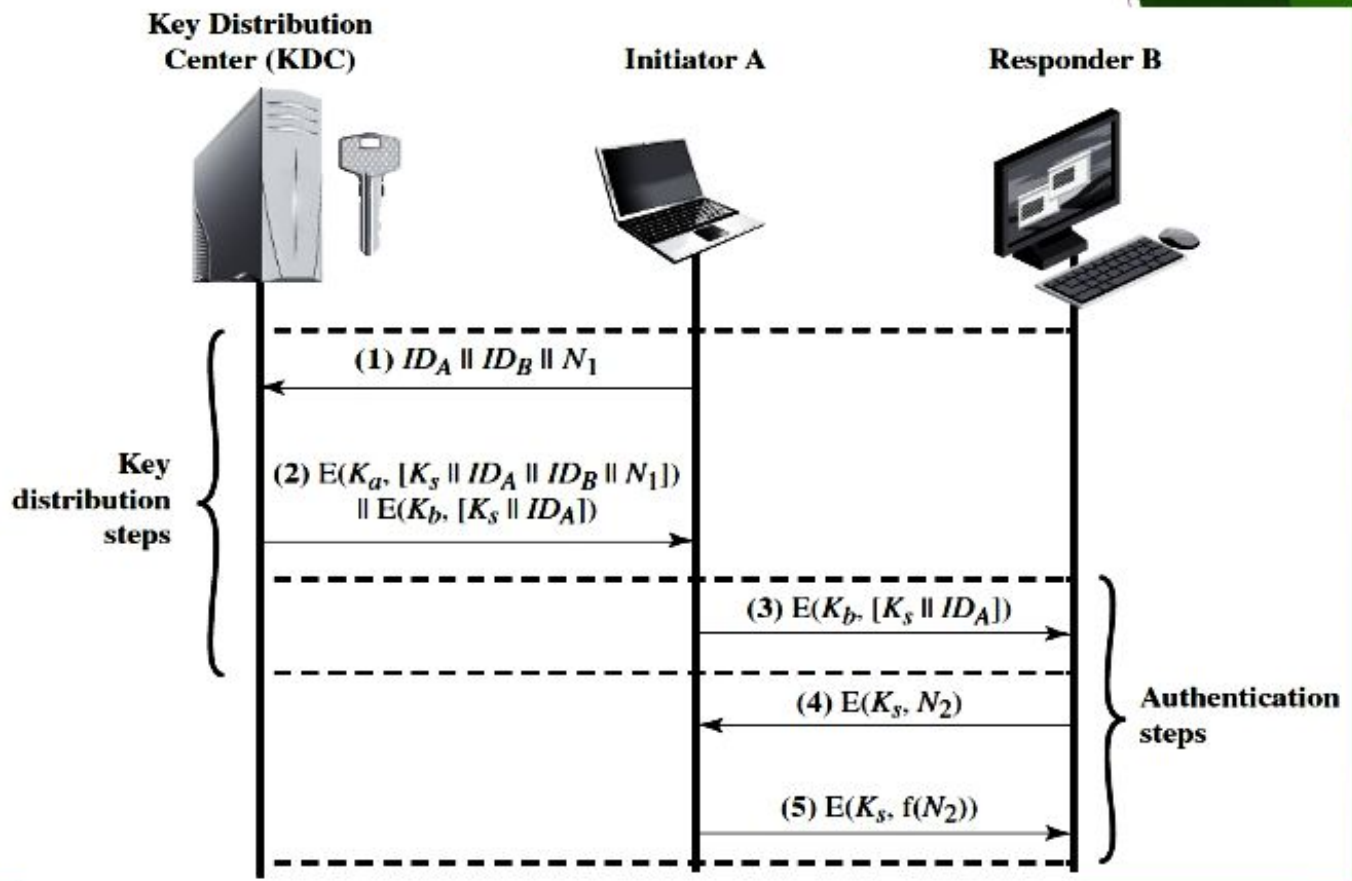
Remote User-Authentication Using Symmetric Encryption

Mutual Authentication:

A two-level hierarchy of symmetric keys can be used to provide confidentiality for communication in a distributed environment

- ▣ Strategy involves the use of a trusted key distribution center (KDC)
- ▣ Each party shares a secret key, known as a master key, with the KDC
- ▣ KDC is responsible for generating keys to be used for a short time over a connection between two parties and for distributing those keys using the master keys to protect the distribution

Mutual Authentication



1. $A \rightarrow KDC: ID_A \parallel ID_B \parallel N_1$

2. $KDC \rightarrow A: E(K_a, [K_s \parallel ID_B \parallel N_1 \parallel E(K_b, [K_s \parallel ID_A])])$

3. $A \rightarrow B: E(K_b, [K_s \parallel ID_A])$

4. $B \rightarrow A: E(K_s, N_2)$

5. $A \rightarrow B: E(K_s, f(N_2))$

1. $A \rightarrow \text{KDC}: ID_A \| ID_B$
2. $\text{KDC} \rightarrow A: E(K_a, [K_s \| ID_B \| T \| E(K_b, [K_s \| ID_A \| T])])$
3. $A \rightarrow B: E(K_b, [K_s \| ID_A \| T])$
4. $B \rightarrow A: E(K_s, N_1)$
5. $A \rightarrow B: E(K_s, f(N_1))$

$$| \text{Clock} - T | < \Delta t_1 + \Delta t_2$$

1. $A \rightarrow B: ID_A \| N_a$
2. $B \rightarrow \text{KDC}: ID_B \| N_b \| E(K_b, [ID_A \| N_a \| T_b])$
3. $\text{KDC} \rightarrow A: E(K_a, [ID_B \| N_a \| K_s \| T_b]) \| E(K_b, [ID_A \| K_s \| T_b]) \| N_b$
4. $A \rightarrow B: E(K_b, [ID_A \| K_s \| T_b]) \| E(K_s, N_b)$

1. $A \rightarrow B: E(K_b, [ID_A \| K_s \| T_b]) \| N'_a$
2. $B \rightarrow A: N'_b \| E(K_s, N'_a)$
3. $A \rightarrow B: E(K_s, N'_b)$

One-Way Authentication

1. $A \rightarrow KDC: ID_A \parallel ID_B \parallel N_1$
2. $KDC \rightarrow A: E(K_a, [K_s \parallel ID_B \parallel N_1 \parallel E(K_b, [K_s \parallel ID_A])])$
3. $A \rightarrow B: E(K_b, [K_s \parallel ID_A]) \parallel E(K_s, M)$

Remote User Authentication Using Asymmetric Encryption

Mutual Authentication:

Public-key encryption for session key distribution

- Assumes each of the two parties is in possession of the current public key of the other
- May not be practical to require this assumption

Denning protocol using timestamps

- Uses an authentication server (AS) to provide public-key certificates
- Requires the synchronization of clocks

Woo and Lam makes use of nonces

- Care needed to ensure no protocol flaws

1. $A \rightarrow AS: ID_A \parallel ID_B$

2. $AS \rightarrow A: E(PR_{as}, [ID_A \parallel PU_a \parallel T]) \parallel E(PR_{as}, [ID_B \parallel PU_b \parallel T])$

3. $A \rightarrow B: E(PR_{as}, [ID_A \parallel PU_a \parallel T]) \parallel E(PR_{as}, [ID_B \parallel PU_b \parallel T]) \parallel E(PU_b, E(PR_a, [K_s \parallel T]))$

1. $A \rightarrow KDC: ID_A \parallel ID_B$

2. $KDC \rightarrow A: E(PR_{auth}, [ID_B \parallel PU_b])$

3. $A \rightarrow B: E(PU_b, [N_a \parallel ID_A])$

4. $B \rightarrow KDC: ID_A \parallel ID_B \parallel E(PU_{auth}, N_a)$

5. $KDC \rightarrow B: E(PR_{auth}, [ID_A \parallel PU_a]) \parallel E(PU_b, E(PR_{auth}, [N_a \parallel K_s \parallel ID_B]))$

6. $B \rightarrow A: E(PU_a, [E(PR_{auth}, [(N_a \parallel K_s \parallel ID_B))] \parallel N_b])$

7. $A \rightarrow B: E(K_s, N_b)$

1. $A \rightarrow KDC: ID_A \parallel ID_B$
2. $KDC \rightarrow A: E(PR_{auth}, [ID_B \parallel PU_b])$
3. $A \rightarrow B: E(PU_b, [N_a \parallel ID_A])$
4. $B \rightarrow KDC: ID_A \parallel ID_B \parallel E(PU_{auth}, N_a)$
5. $KDC \rightarrow B: E(PR_{auth}, [ID_A \parallel PU_a]) \parallel E(PU_b, E(PR_{auth}, [N_a \parallel K_s \parallel ID_B]))$
6. $B \rightarrow A: E(PU_a, [E(PR_{auth}, [(N_a \parallel K_s \parallel ID_B))] \parallel N_b])$
7. $A \rightarrow B: E(K_s, N_b)$

15

One-Way Authentication

Have Public –key approach for E-mail

- ✉ Encryption of message for confidentiality, authentication, or both
- ✉ The public-key algorithm must be applied once or twice to what may be a long message

$A \rightarrow B: E(PU_b, K_s) \parallel E(K_s, M)$: Confidentiality

$A \rightarrow B: M \parallel E(PR_a, H(M))$: Authentication

$A \rightarrow B: M \parallel E(PR_a, H(M)) \parallel E(PR_{as}, [T \parallel ID_A \parallel PU_a])$: Digital Certificate

Authentication Applications

- ✘ Authentication refers to the process of confirming identity. While often used interchangeably with authorization, authentication represents a fundamentally different function.
- ✘ In authentication, a user or application proves they are who they say they are by providing valid credentials for verification.
- ✘ Authentication is often proved through a username and password, sometimes combined with other elements called *factors*, which fall into three categories:
 - (a) what you know
 - (b) what you have
 - (c) what you are
- ✘ Authentication functions
- ✘ Developed to support application-level authentication & digital signatures
- ✘ Will consider Kerberos – a private-key authentication service

Kerberos

- ✘ Kerberos is a network authentication protocol. It is designed to provide strong authentication for client/server applications by using secret-key cryptography.
- ✘ A free implementation of this protocol is available from the Massachusetts Institute of Technology (MIT).
- ✘ Kerberos is available in many commercial products as well.
- ✘ Trusted key server system from MIT
- ✘ Provides centralised private-key third-party authentication in a distributed network
 - ✘ allows users access to services distributed through network
 - ✘ without needing to trust all workstations
 - ✘ rather all trust a central authentication server
- ✘ Two versions in use: 4 & 5

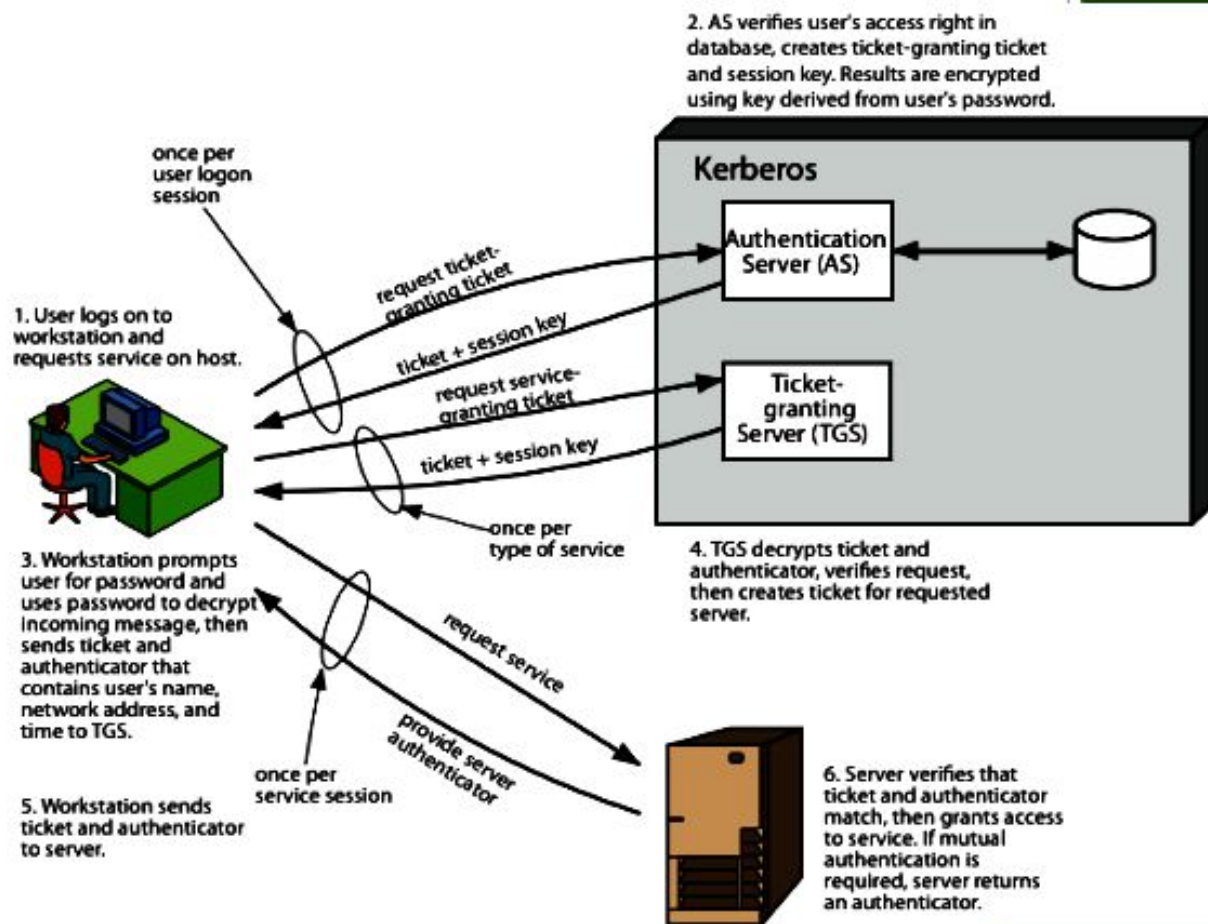
Kerberos v4 Overview

- ⌘ A basic third-party authentication scheme
- ⌘ Have an Authentication Server (AS)
 - ⌘ users initially negotiate with AS to identify self
 - ⌘ AS provides a non-corruptible authentication credential (ticket granting ticket TGT)
- ⌘ Have a Ticket Granting server (TGS)
 - ⌘ users subsequently request access to other services from TGS on basis of users TGT

Kerberos v4 Dialogue

1. Obtain ticket granting ticket from AS
 - once per session
2. Obtain service granting ticket from TGT
 - for each distinct service required
3. Client/Server exchange to obtain service
 - on every service request

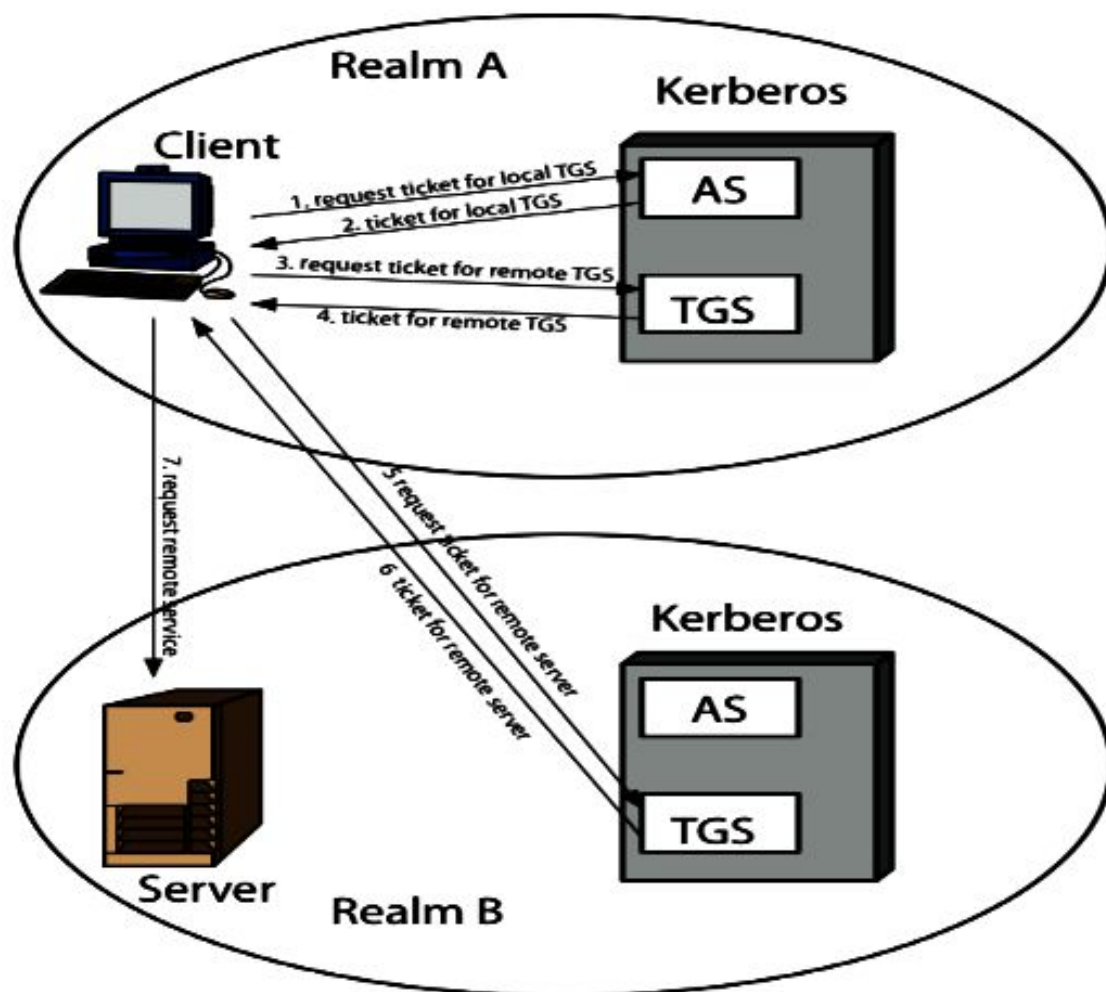
Kerberos 4 Overview



Kerberos Realms

- ❑ A Kerberos environment consists of:
 - ❑ a Kerberos server
 - ❑ a number of clients, all registered with server
 - ❑ application servers, sharing keys with server
- ❑ This is termed a realm
 - ❑ typically a single administrative domain
- ❑ If have multiple realms, their Kerberos servers must share keys and trust

Kerberos Realms



Kerberos Version 5

- ❑ Developed in mid 1990's
- ❑ Specified as Internet standard RFC 1510
- ❑ Provides improvements over v4
 - ❑ addresses environmental shortcomings
 - ❑ encryption algorithm, network protocol, byte order, ticket lifetime, authentication forwarding, inter-realm authentication
 - ❑ and technical deficiencies
 - ❑ double encryption, non-std mode of use, session keys, password attacks

Difference Between Kerberos V4 & V5

BASIS OF COMPARISON	KERBEROS VERSION 4	KERBEROS VERSION 5
Year of Release	Kerberos Version 4 was released in 1980's way before version 5 was released.	The Kerberos version 5 was published in 1993, 13 year after the release of Kerberos Version 4.
Encryption Techniques	Kerberos version 4 uses DES encryption techniques.	In Kerberos version 5, the cipher text is tagged with an encryption type identifier and therefore any type of encryption can be used.
Encoding	Kerberos version 4 uses the "receiver-makes-right" encoding system.	The Kerberos version 5 uses the ASN.1 coding system.
Ticket Lifetime	In Kerberos version 4, the ticket lifetime has to be specified in units of 5 minutes.	In Kerberos version 5, ticket one lifetime can specify an explicit start and finish times allowing arbitrary lifetimes.
Ticket Support	Ticket support is satisfactory in this version.	Ticket support is well extended. Facilitates forwarding, renewing and postdating tickets.
IP Addresses	Kerberos version 4 contains only a few IP addresses and other addresses for types of network protocol.	Kerberos version 5 contains multiple IP addresses for types of network protocols.
Key	Given that the same key is used repeatedly to gain a service from particular server, there is a risk that an attacker can replay messages from an old session to the client or sever.	In Kerberos version 5, this is avoided by requiring a sub session key which is used only for one connection.

Technical Deficiencies in Kerberos

- ❑ Double Encryption
- ❑ PCBC Encryption
- ❑ Session Key
- ❑ Password Attack

Kerberos V5 Advantages

- ❑ V5 is more resistant to determine attack over the network.
- ❑ More flexible, can work with different kind of network.
- ❑ Support delegation of authentication.
- ❑ Longer TKT expiration time.
- ❑ Renewable TKTs.

Disadvantages

- ❑ Single Point failure: It needs contains availability of a central server. When Kerberos server is down, no one can login.
- ❑ Every network service must be individually modified for use with Kerberos.
- ❑ Require a secure Kerberos server.
- ❑ Doesn't work well in time sharing environment.
- ❑ Kerberos requires the clock of the involved hosts to be synchronized.

Web Security

- ❑ Measures to protect data during their transmission over a collection of interconnected networks.
- ❑ The World Wide Web is fundamentally a client/server application running over the internet and TCP/IP intranets.
- ❑ Web now widely used by business, government, individuals
- ❑ But Internet & Web are vulnerable
- ❑ have a variety of threats
 - ❑ integrity
 - ❑ confidentiality
 - ❑ denial of service
 - ❑ authentication
- ❑ need added security mechanisms

Web Security

- ❑ The Web is increasingly serving as a highly visible outlet for corporate and product information and as the platform for business transactions. Reputations can be damaged and money can be lost if the Web servers are subverted.
- ❑ Although Web browsers are very easy to use, Web servers are relatively easy to configure and manage, and Web content is increasingly easy to develop, the underlying software is extraordinarily complex. This complex software may hide many potential security flaws.
The short history of the Web is filled with examples of new and upgraded systems, properly installed, that are vulnerable to a variety of security attacks.

Web Security Threats

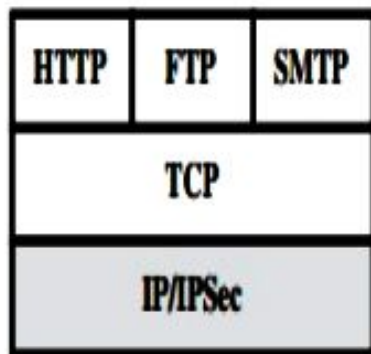
- ⌘ These can be described as passive attacks including eavesdropping on network traffic between browser and gaining access to information on a website that is supposed to be restricted.
- ⌘ Active attacks including impersonating another user, altering information on a website.

	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

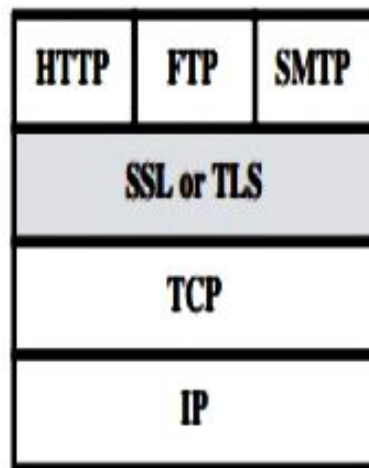
Web Traffic Security Approaches

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

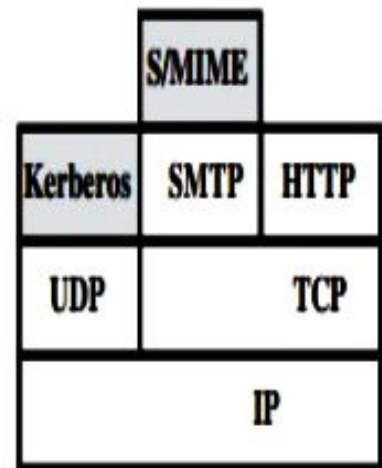
Web Traffic Security Approaches



(a) Network level



(b) Transport level



(c) Application level

Web Traffic Security Approaches

- ⊠ One way to provide Web security is to use IP security (IPsec).
- ⊠ The advantage of using IPsec is that it is transparent to end users and applications and provides a general-purpose solution.
- ⊠ Furthermore, IPsec includes a filtering capability so that only selected traffic need incur the overhead of IPsec processing.

Web Traffic Security Approaches

- ❏ Sockets Layer (SSL) and the follow-on Internet standard known as Transport Layer Security (TLS).
- ❏ At this level, there are two implementation choices. For full generality, SSL (or TLS) could be provided as part of the underlying protocol suite and therefore be transparent to applications.
- ❏ Alternatively, SSL can be embedded in specific packages.
- ❏ For example, Netscape and Microsoft Explorer browsers come equipped with SSL, and most Web servers have implemented the protocol.

Web Traffic Security Approaches

- ❏ Application specific security services are embedded within the particular application.
- ❏ The advantage of this approach is that the service can be tailored to the specific needs of a given application.

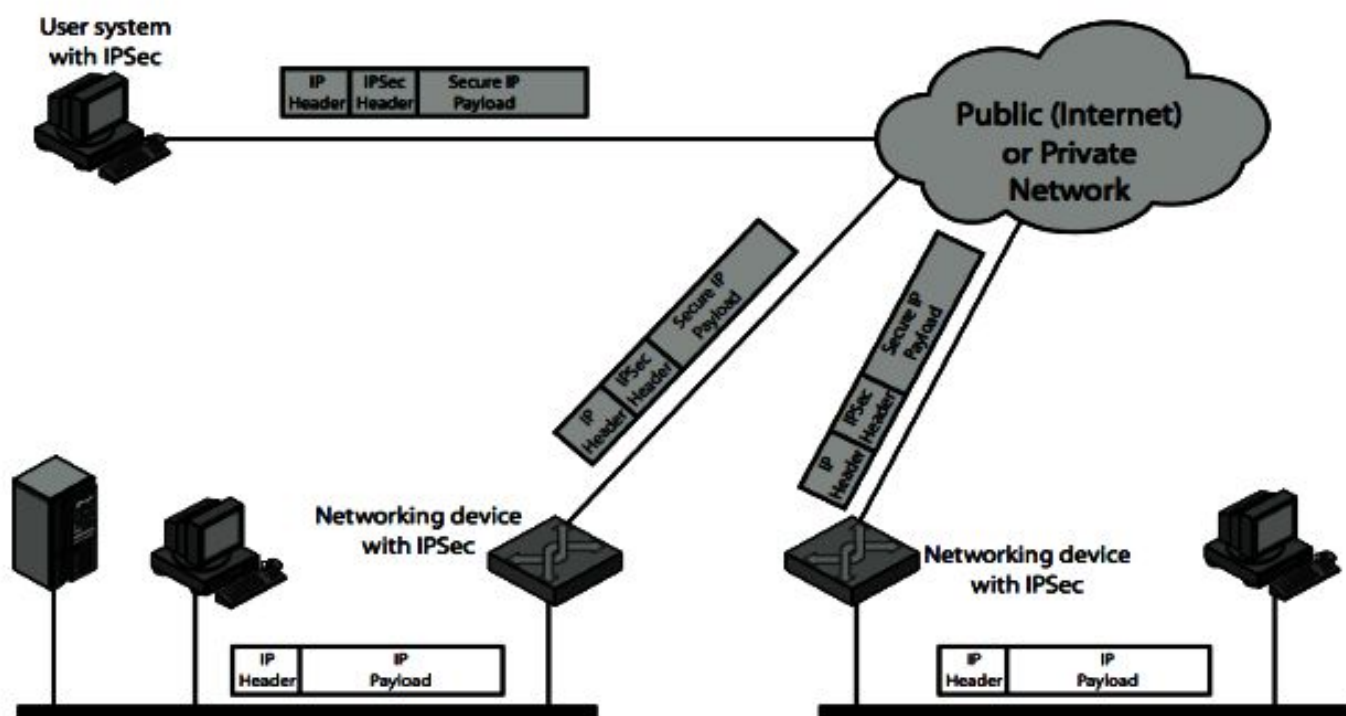
IP Security

- ❑ Have a range of application specific security mechanisms
 - ❑ 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

IPSec

- ❑ General IP Security mechanisms
- ❑ Provides
 - ❑ authentication
 - ❑ confidentiality
 - ❑ key management
- ❑ Applicable to use over LANs, across public & private WANs, & for the Internet

IPSec 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
- ❏ Defined in numerous RFC's
 - ❏ incl. RFC 2401/2402/2406/2408
 - ❏ many others, grouped by category
- ❏ Mandatory in IPv6, optional in IPv4
- ❏ Have two security header extensions:
 - ❏ Authentication Header (AH)
 - ❏ Encapsulating Security Payload (ESP)

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

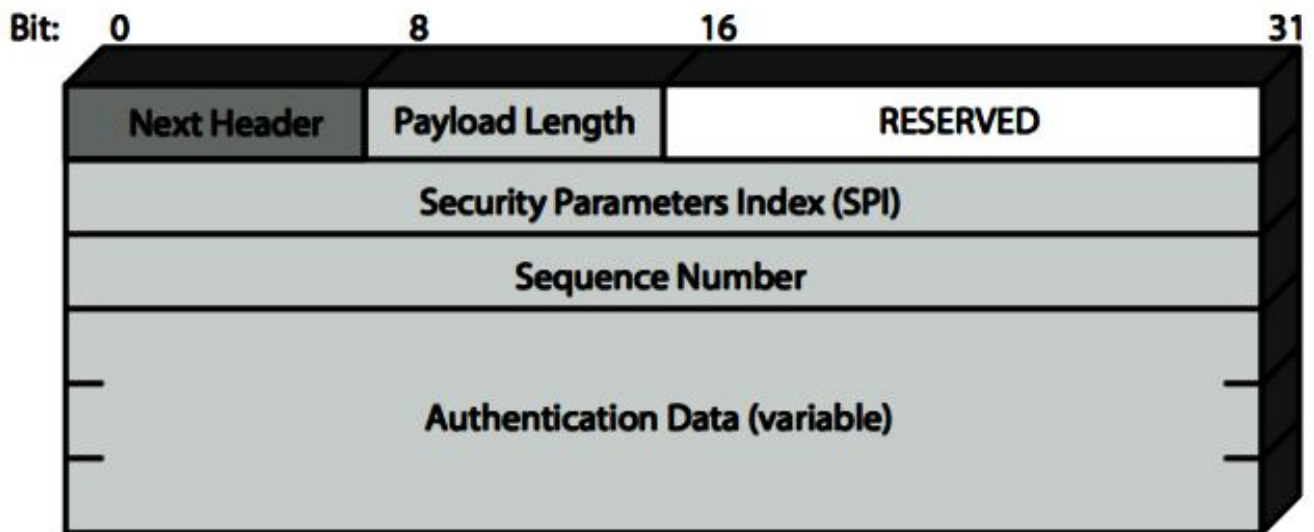
Security Associations

- ⊠ A one-way relationship between sender & receiver that affords security for traffic flow
- ⊠ Defined by 3 parameters:
 - ⊠ Security Parameters Index (SPI)
 - ⊠ IP Destination Address
 - ⊠ Security Protocol Identifier
- ⊠ Has a number of other parameters
 - ⊠ seq no, AH & EH info, lifetime etc
- ⊠ Have a database of Security Associations

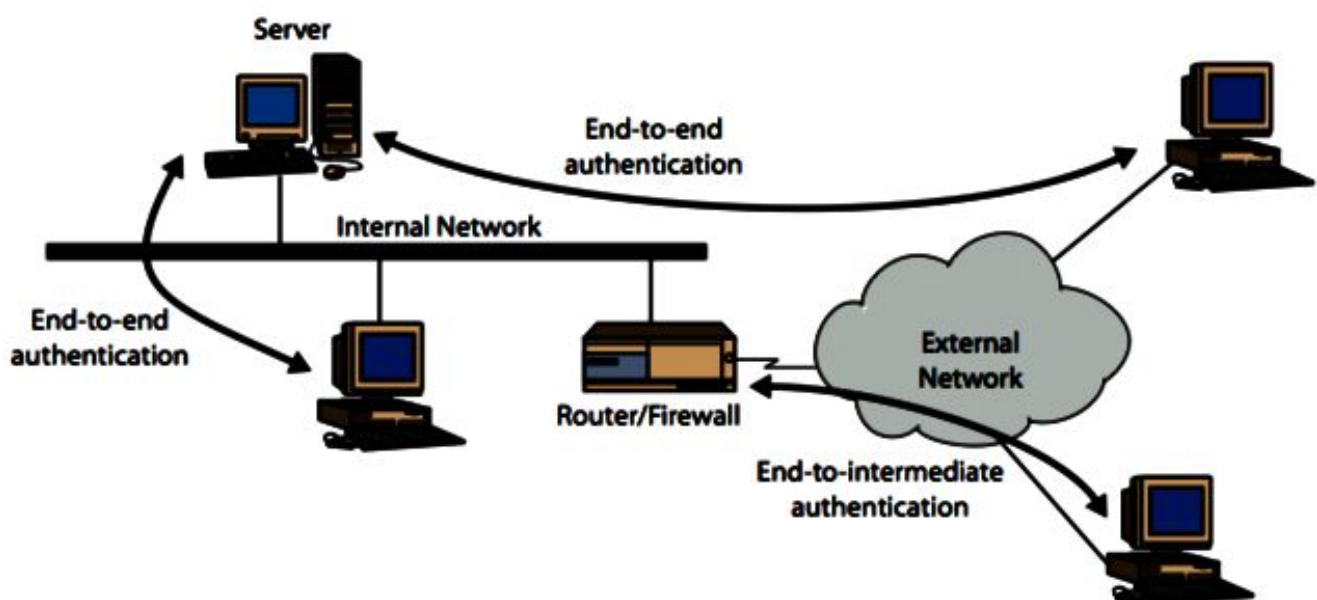
Authentication Header (AH)

- ⊠ Provides support for data integrity & authentication of IP packets
 - ⊠ end system/router can authenticate user/app
 - ⊠ prevents address spoofing attacks by tracking sequence numbers
- ⊠ Based on use of a MAC
 - ⊠ HMAC-MD5-96 or HMAC-SHA-1-96
- ⊠ Parties must share a secret key

Authentication Header



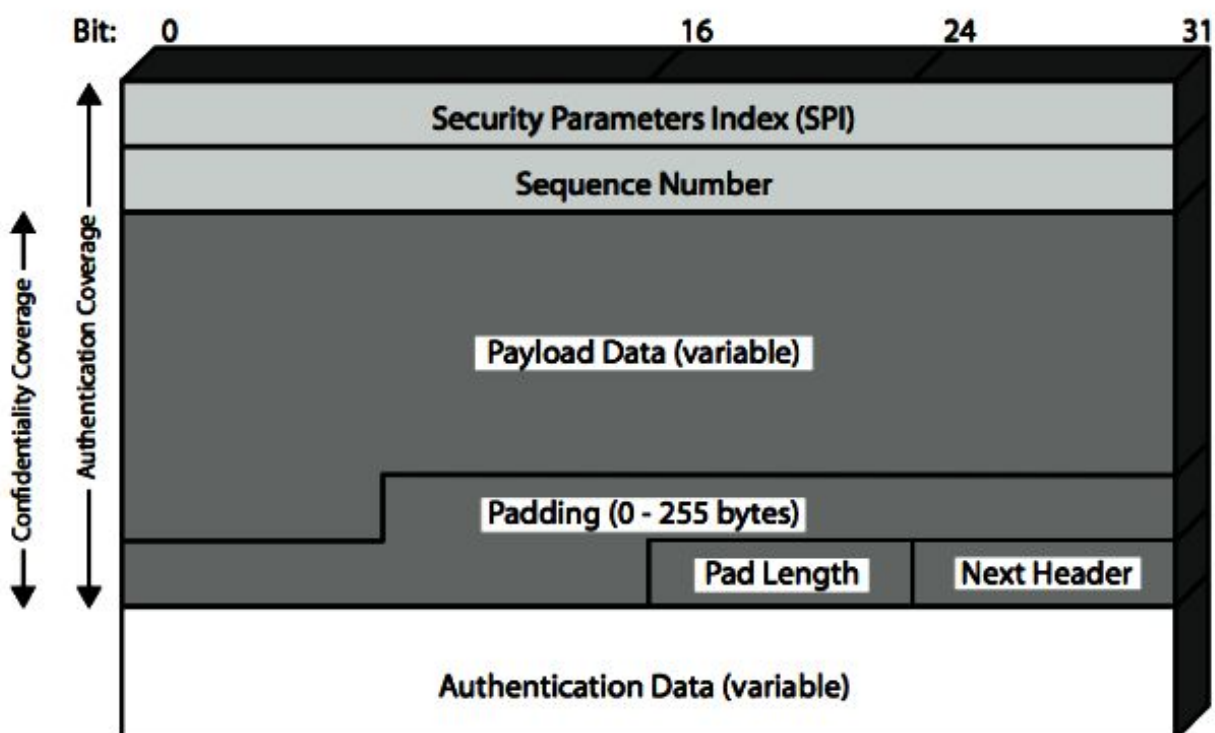
Transport & Tunnel Modes



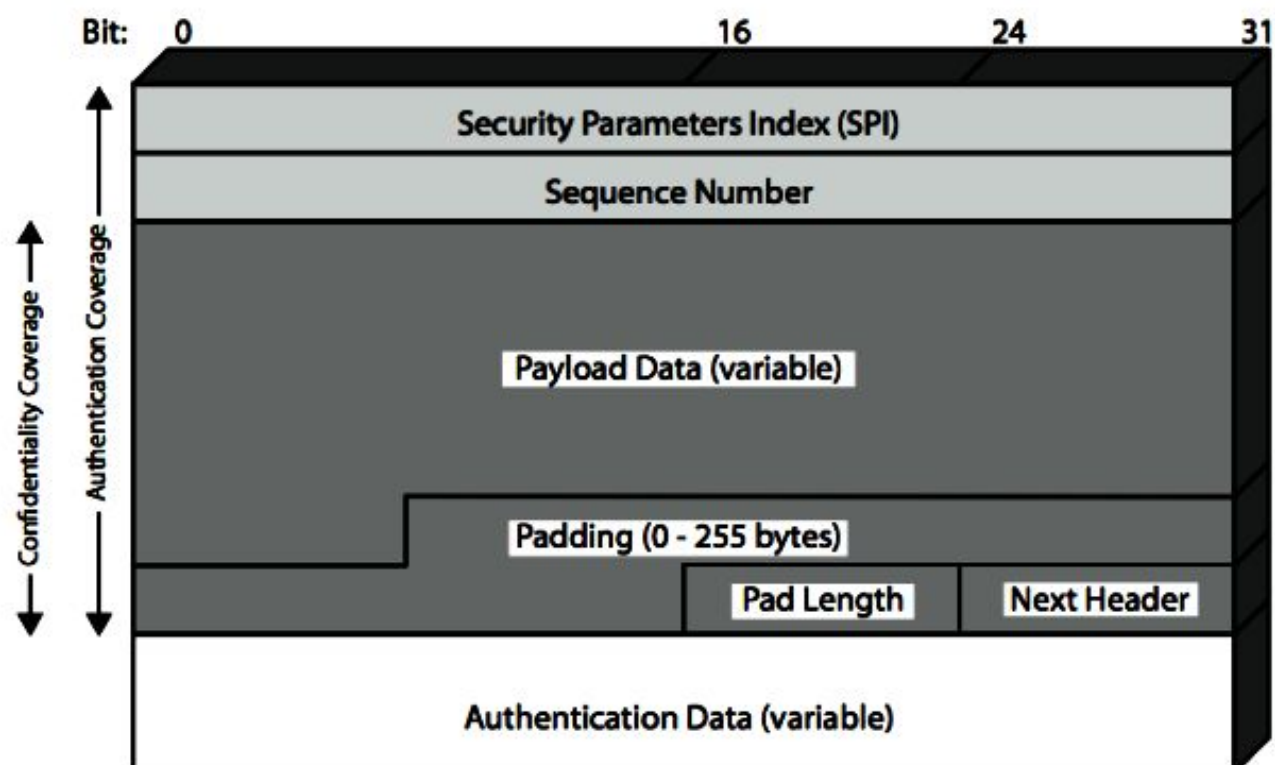
Encapsulating Security Payload (ESP)

- ✘ Provides message content confidentiality & limited traffic flow confidentiality
- ✘ Can optionally provide the same authentication services as AH
- ✘ Supports range of ciphers, modes, padding
 - ✘ incl. DES, Triple-DES, RC5, IDEA, CAST etc
 - ✘ CBC & other modes
 - ✘ padding needed to fill blocksize, fields, for traffic flow

Encapsulating Security Payload



Encapsulating Security Payload



13

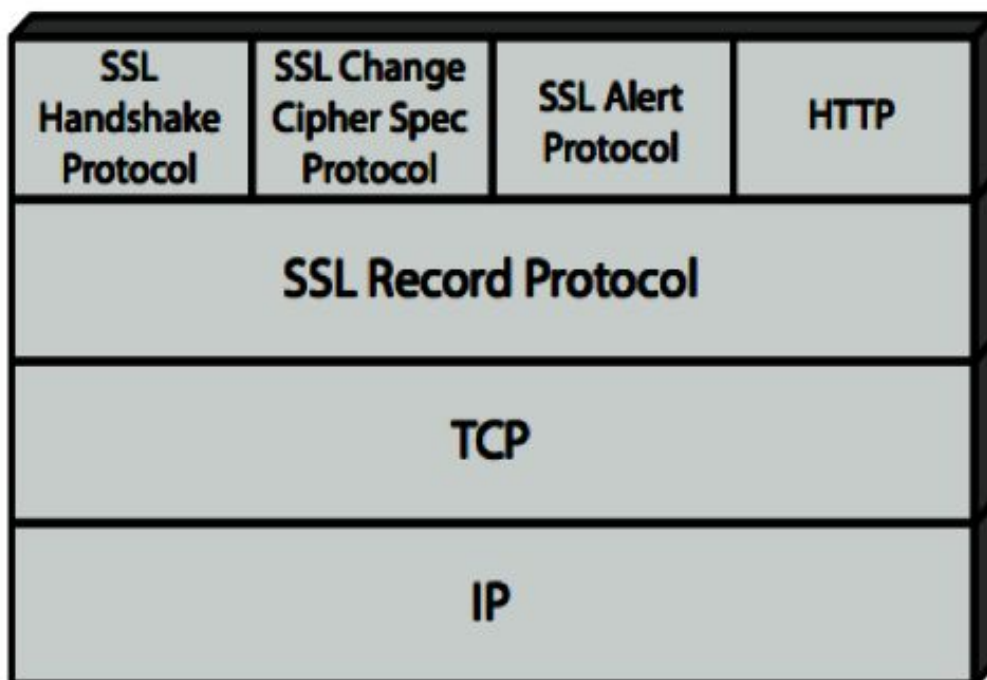
Transport vs Tunnel Mode ESP

- ❑ Transport mode is used to encrypt & optionally authenticate IP data
 - ❑ data protected but header left in clear
 - ❑ 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
 - ❑ good for VPNs, gateway to gateway security

SSL (Secure Socket Layer)

- ✘ Transport layer security service
- ✘ Originally developed by Netscape
- ✘ Version 3 designed with public input
- ✘ Subsequently became Internet standard known as TLS (Transport Layer Security)
- ✘ Uses TCP to provide a reliable end-to-end service
- ✘ SSL has two layers of protocols

SSL Architecture



SSL Architecture

✘ SSL connection

- ✘ a transient, peer-to-peer, communications link
- ✘ associated with 1 SSL session

✘ SSL session

- ✘ an association between client & server
- ✘ created by the Handshake Protocol
- ✘ define a set of cryptographic parameters
- ✘ may be shared by multiple SSL connections

SSL Record Protocol Services

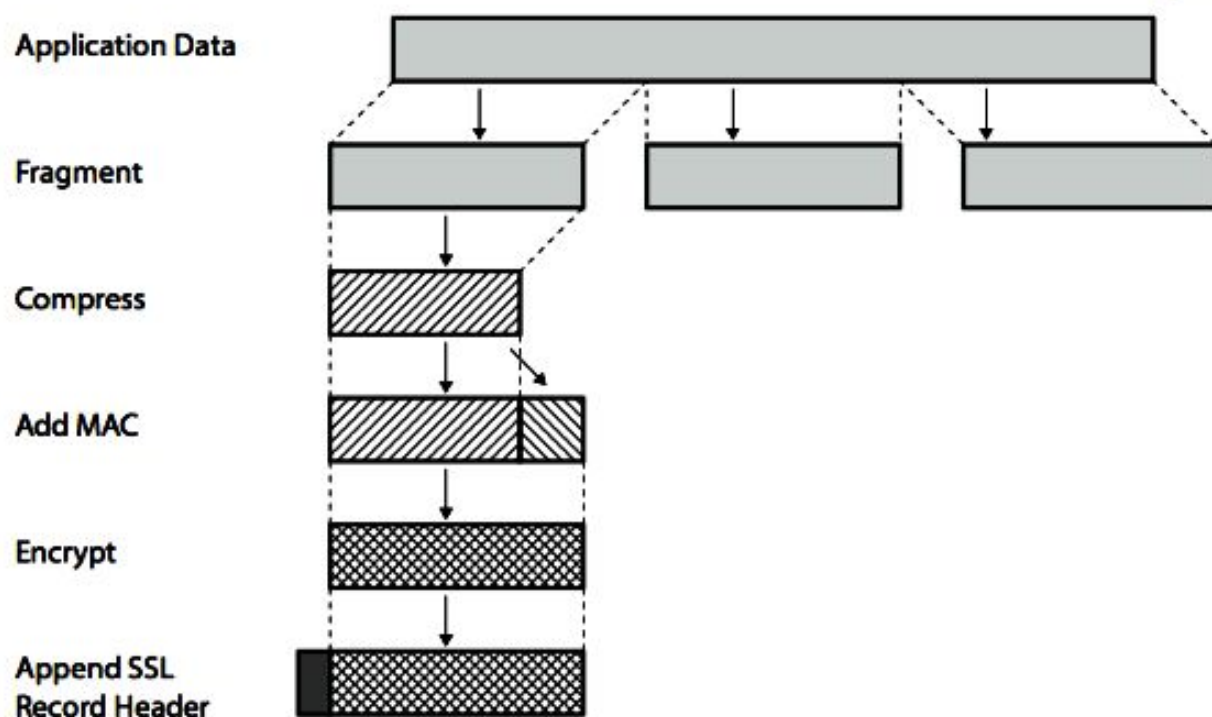
✘ Message integrity

- ✘ using a MAC with shared secret key
- ✘ similar to HMAC but with different padding

✘ Confidentiality

- ✘ using symmetric encryption with a shared secret key defined by Handshake Protocol
- ✘ AES, IDEA, RC2-40, DES-40, DES, 3DES, Fortezza, RC4-40, RC4-128
- ✘ message is compressed before encryption

SSL Record Protocol Operation



SSL Change Cipher Spec Protocol

- ☒ One of 3 SSL specific protocols which use the SSL Record protocol
- ☒ A single message
- ☒ Causes pending state to become current
- ☒ Hence updating the cipher suite in use

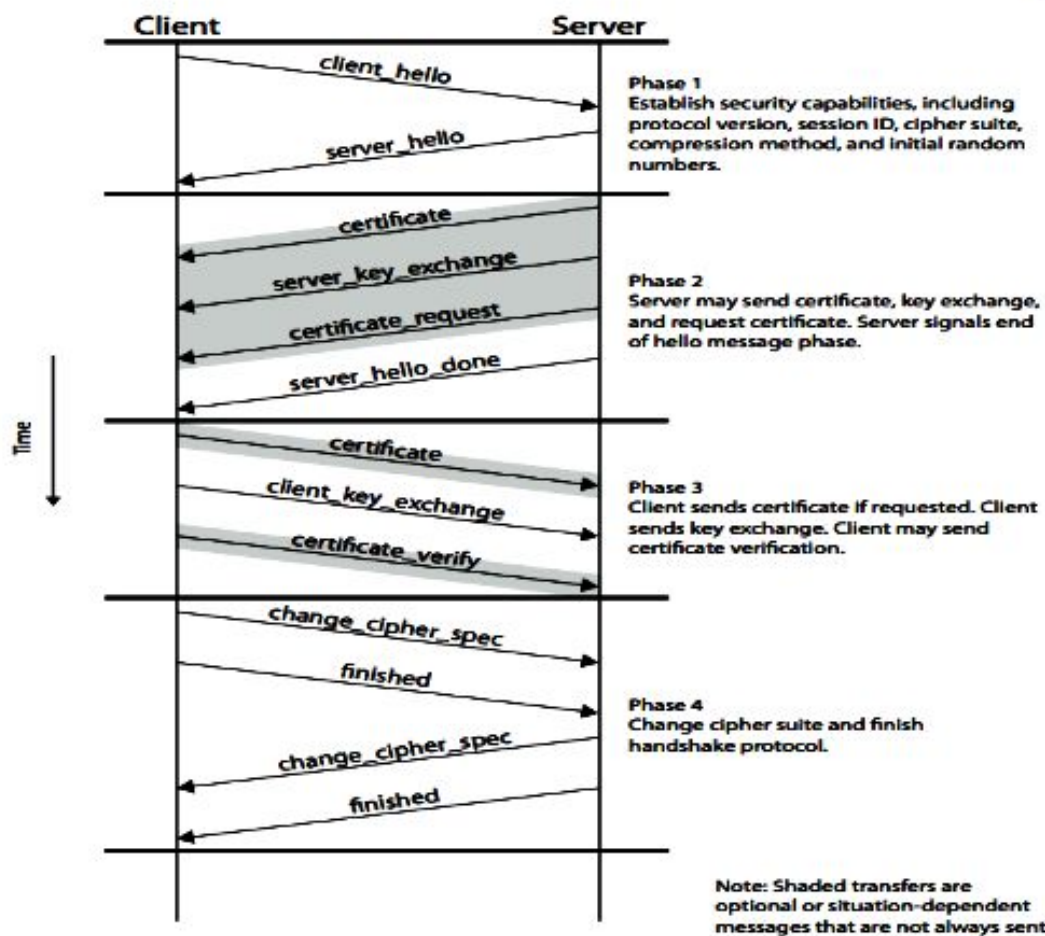
SSL Alert Protocol

- ✘ conveys SSL-related alerts to peer entity
- ✘ severity
 - ⚠ warning or fatal
- ✘ specific alert
 - ⚠ fatal: unexpected message, bad record mac, decompression failure, handshake failure, illegal parameter
 - ⚠ warning: close notify, no certificate, bad certificate, unsupported certificate, certificate revoked, certificate expired, certificate unknown
- ✘ compressed & encrypted like all SSL data

SSL Handshake Protocol

- ✘ Allows server & client to:
 - ✘ authenticate each other
 - ✘ to negotiate encryption & MAC algorithms
 - ✘ to negotiate cryptographic keys to be used
- ✘ Comprises a series of messages in phases
 1. Establish Security Capabilities
 2. Server Authentication and Key Exchange
 3. Client Authentication and Key Exchange
 4. Finish

SSL Handshake Protocol



TLS (Transport Layer Security)

- ❑ Transport Layer Securities (TLS) are designed to provide security at the transport layer.
- ❑ TLS was derived from a security protocol called Secure Service Layer (SSL).
- ❑ TLS ensures that no third party may eavdrops or tamper with any message
- ❑ IETF standard RFC 2246 similar to SSLv3
- ❑ Some minor differences b/w TLS & SSL

Transport Layer Security (TLS)

There are several benefits of TLS

✘ **Encryption:**

TLS/SSL can help to secure transmitted data using encryption.

✘ **Interoperability:**

TLS/SSL works with most web browsers, including Microsoft Internet Explorer and on most operating systems and web servers.

✘ **Algorithm flexibility:**

TLS/SSL provides operations for authentication mechanism, encryption algorithms and hashing algorithm that are used during the secure session.

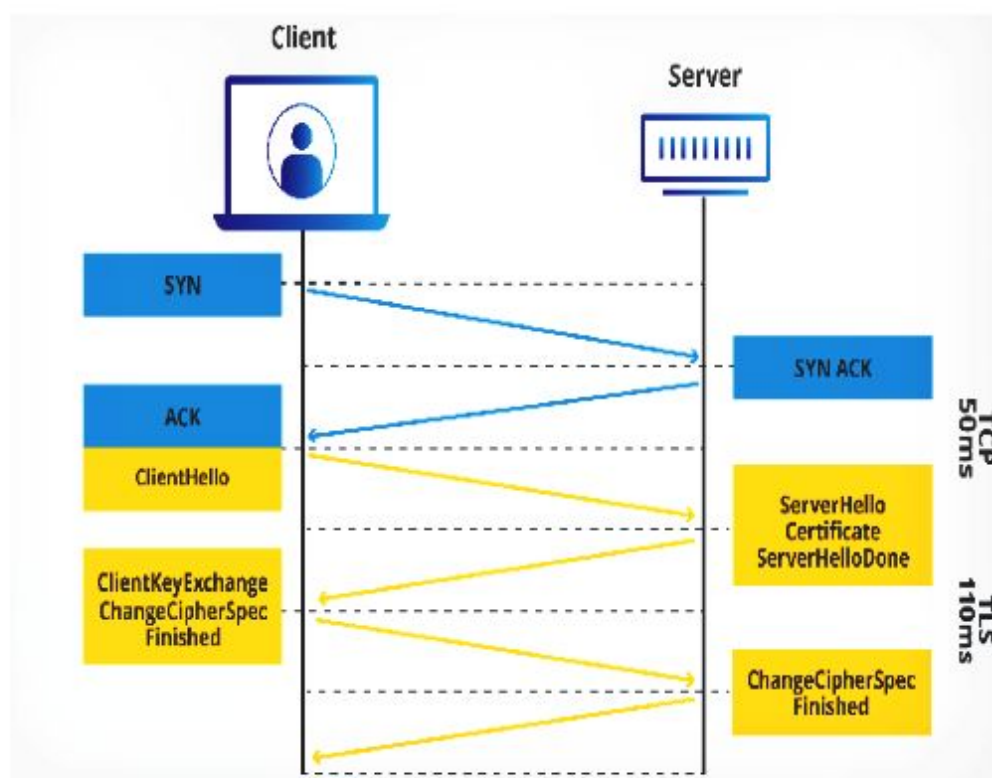
✘ **Ease of Deployment:**

Many applications TLS/SSL temporarily on a windows server 2003 operating systems.

✘ **Ease of Use:**

Because we implement TLS/SSL beneath the application layer, most of its operations are completely invisible to client.

TLS Working



Difference Between SSL and TLS

BASIS FOR COMPARISON	SSL	TLS
Version	3.0	1.0
Cipher Suite	Supports Fortezza (algorithm)	Does not support Fortezza
Cryptography secret	Uses message digest of the pre-master secret for creating master secret.	Uses a pseudorandom function to create master secret.
Record protocol	Uses MAC (Message Authentication Code)	Uses HMAC (Hashed MAC)
Alert protocol	The "No certificate" alert message is included.	It eliminates alert description (No certificate) and adds a dozen other values.
Message authentication	Ad hoc	Standard
key material authentication	Ad hoc	Pseudorandom function
Certificate verify	Complex	Simple
Finished	Ad hoc	Pseudorandom function

Hypertext Transfer Protocol Secure (HTTPS)

- ✘ It is an extension of the Hypertext Transfer Protocol (HTTP).
- ✘ It is used for secure communication over a computer network, and is widely used on the Internet.
- ✘ In HTTPS, the communication protocol is encrypted using Transport Layer Security (TLS) or, formerly, its predecessor, Secure Sockets Layer (SSL).
- ✘ The protocol is therefore also often referred to as **HTTP over TLS**, or **HTTP over SSL**.
- ✘ HTTPS URLs begin with "https://" and use port 443 by default, whereas, HTTP URLs begin with "http://" and use port 80 by default.

Hypertext Transfer Protocol Secure (HTTPS)

- ✘ The principal motivations for HTTPS are authentication of the accessed website, protection of the privacy and integrity of the exchanged data while in transit.
- ✘ It protects against man-in-the-middle attacks.
- ✘ The bidirectional encryption of communications between a client and server protects against eavesdropping and tampering of the communication.
- ✘ In practice, this provides a reasonable assurance that one is communicating without interference by attackers with the website that one intended to communicate with, as opposed to an impostor.

Secure Shell (SSH)

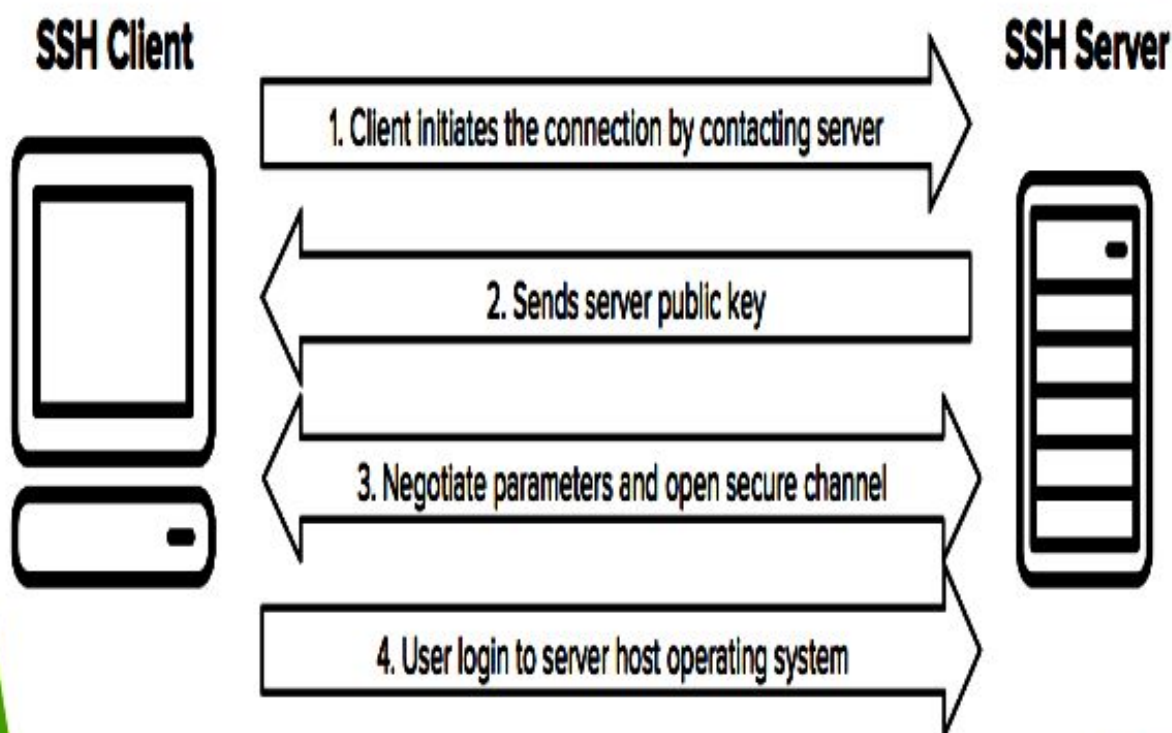
- ✘ The SSH protocol (also referred to as Secure Shell) is a method for secure remote login from one computer to another.
- ✘ It provides several alternative options for strong authentication, and it protects the communications security and integrity with strong encryption.
- ✘ It is a secure alternative to the non-protected login protocols (such as **telnet**, **rlogin**) and insecure file transfer methods (such as **FTP**).

Typical uses of the SSH Protocol

The protocol is used in corporate networks for:

- ✘ providing secure access for users and automated processes
- ✘ interactive and automated file transfers
- ✘ issuing remote commands
- ✘ managing network infrastructure and other mission-critical system components.

SSH Protocol Work



Difference B/W SSH & SSL/TLS

SSH

SSH runs on port 22

SSH is for securely executing commands on a server.

SSH uses a username/password authentication system to establish a secure connection.

SSH is working based on network tunnels.

SSH is a remote protocol

It is used to reduce security threats for remote server login

SSH follows authentication process by server's verification done by client, session key generation, and client's authentication

Data integrity is measured with algorithms like SHA, SHA-2, SHA-256

SSL/TLS

SSL runs on port 443

SSL is used for securely communicating personal information.

SSL normally uses X.509 digital certificates for server and client authentication.

SSL is working based on digital certificates.

SSL is a security protocol

It allows secure transition of data between a server and the browser thus, keeps information intact.

SSL follows authentication process by exchange of digital certificate

Data integrity is measured with the message digest and added to encrypted data before the data is sent.