Unit VI. Network Security and Public Key Infrastructures

Er. Kobid Karkee
Himalaya College of Engineering

Overview of Network Security

- Network security entails protecting the usability, reliability, integrity, and safety of network and data.
- Effective network security defeats a variety of threats from entering or spreading on a network.
- The primary goal of network security are Confidentiality, Integrity, and Availability.
  - It includes both hardware and software technologies
  - It targets a variety of threats
  - It stops them from entering or spreading on your network
  - Effective network security manages access to the network

Overview of Network Security

- Network security combines multiple layers of defenses at the edge and in the network.
- Each network security layer implements policies and controls. Authorized users gain access to network resources, but malicious actors are blocked from carrying out exploits and threats.
- The major types of network security include:
  - Access Control
  - Antivirus and Antimalware Software
  - Application Security
  - Behavioral analytics to detect abnormal network behavior
  - Email Security, Web Security
  - Data loss prevention
  - Firewalls
  - Intrusion prevention and detection system

Overview of Network Security

- To sum up, Network security consists of the policies and practices adopted to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources.
- Network security involves the authorization of access to data in a network, which is controlled by the network administrator.

Public Key Infrastructure

- A public key infrastructure (PKI) is a set of roles, policies, hardware, software and procedures needed to create, manage, distribute, use, store and revoke digital certificates and manage public-key encryption.
- PKI is based on digital certificates that verify the identity of the machines and/or users that ultimately proves the integrity of the transaction.
- While it is possible for an entity to create its own PKI and issue its own digital certificates, most digital certificates are issued by a certificate authority (CA).
- A certificate authority or certification authority (CA) is an entity that stores, signs, and issues digital certificates.

Public Key Infrastructure

- A digital certificate certifies the ownership of a public key by the named subject of the certificate.
- This allows others (relying parties) to rely upon signatures or on assertions made about the private key that corresponds to the certified public key.
- A CA acts as a trusted third party—trusted both by the subject (owner) of the certificate and by the party relying upon the certificate.
- The format of these certificates is specified by the X.509 or EMV standard.

Digital Certificates

- A digital certificate, also known as a public key certificate, is used to cryptographically link ownership of a public key with the entity that owns it.
- A Digital Certificate is an electronic file that is tied to a cryptographic key pair and authenticates the identity of a website, individual, organization, user, device or server.
- They are issued by Certificate Authorities (CAs) and perform two primary functions:
  - Verifying the identity of the sender/receiver of an electronic message
  - Providing the means to encrypt/decrypt messages between sender and receiver (i.e., binding and entity to their public key)

Digital Certificates

There are three basic types of digital signature certificates:

1. Individual digital signature certificates (signing certificates):

- These certificates are used to identify a person and include personal information.
- They can be used to sign electronic documents (i.e., to provide electronic signatures) and emails, and to implement access control mechanisms for sensitive or valuable information.

Digital Certificates

2. Server certificates:

- These certificates identify a server (computer) and contain the host name or IP address.
- They are used for one- or two-layer SSL to ensure secure communication of data over a network.

3. Encryption certificates:

- These certificates are used to encrypt a message using the public key of the recipient to ensure data confidentiality during transmission.
- Different signatures for encryption and digital signatures are available from different CAs.

Digital Certificate Life Cycle Management

Certificate Life Cycle is divided into the following stages:

1. Certificate Enrollment

- Certificate enrollment is initiated by a user request to the appropriate CA.
- This is a cooperative process between a user (or a user's PKI software, such as an e-mail or Web browser application) and the CA.
- The enrollment request contains the public key and enrollment information.
- Once a user requests a certificate, the CA verifies information based on its established policy rules, creates the certificate, posts the certificate, and then sends an identifying certificate to the user.
- During the certificate distribution, the CA sets policies that affect the use of the certificate.

Digital Certificate Life Cycle Management

2. Certificate Validation

- When a certificate is used, the certificate status is checked to verify that the certificate is still operationally valid.
- During the validation process, the CA checks the status of the certificate and verifies that the certificate is not its Certificate Revocation List (CRL).

Digital Certificate Life Cycle Management

3. Certificate Revocation

- A certificate issued by a CA includes an expiration date that defines how long the certificate is valid.
- If a certificate needs to be revoked before that date, the CA can be instructed to add the certificate to its CRL.
- Reasons a certificate might need to be revoked include the certificate being lost or compromised, or the person the certificate was issued to leaving the company.

Digital Certificate Life Cycle Management

4. Certificate Renewal

- When a certificate reaches its expiration date, and if the certificate policy allows it, it is renewed either automatically, or by user intervention.
- When renewing a certificate, you must choose whether or not to generate new public and private keys.

Digital Certificate Life Cycle Management

5. Certificate Removal

- When a certificate is no longer in use, the certificate and any backup copies or archived copies of the certificate should be destroyed, along with the private key associated with the certificate.
- This helps ensure that the certificate is not compromised and used.

6. Certificate Auditing

- Certificate auditing involves tracking the creation, expiration, and revocation of certificates.
- In certain instances, it can also track each successful use of a certificate.

So, Finally, What is PKI?

- The entire system that is formed by CAs together with the necessary support mechanisms is called a public-key infrastructure (PKI).
- The principal objective for developing a PKI is to enable secure, convenient, and efficient acquisition of public keys.

Issues in PKI: -

1. Cross-certification of CAs

- When a user from one CA communicates with another user from a different CA, cross-certification of CAs is required, and thus a "chain of trust" or "trust model" should be established by "delegating trust".

2. Certificate Revocation Lists (CRLs)

- Certificates must be revoked whenever a user is no longer in the network. For this, CRLs must be sent out periodically which is a burden to the bandwidth of the system.

PKIX Architecture Model

End entity:

- users, devices etc.

Certification authority (CA):

- The issuer of certificates and CRLs.

Registration Authorities:

- CA often delegates different functions to them such as processing certificates.

PKIX Management Functions of:

- Registration
- Initialization
- Certification
- Key pair recovery
- Key pair update
- Revocation request
- Cross certification

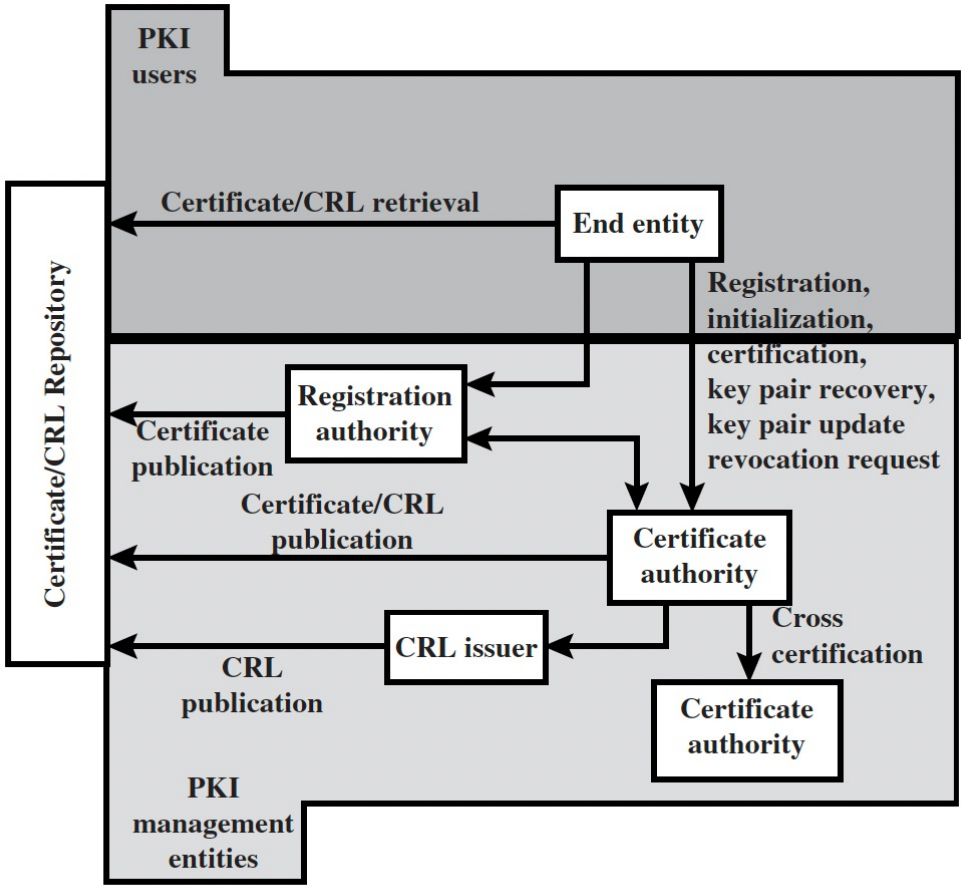


Figure 14.16 PKIX Architectural Model

X.509 Authentication Service

- a digital certificate that uses the widely accepted international X.509 public key infrastructure (PKI) standard to verify that a public key belongs to the user, computer or service identity contained within the certificate.
- part of CCITT X.500 directory service standards
  - distributed servers maintaining some info database
- defines framework for authentication services
  - directory may store public-key certificates
  - with public key of user
  - signed by certification authority
- also defines authentication protocols
- uses public-key crypto & digital signatures
  - algorithms not standardised, but RSA recommended

X.509 Certificates

- The heart of the X.509 scheme is the public-key certificate associated with each user.

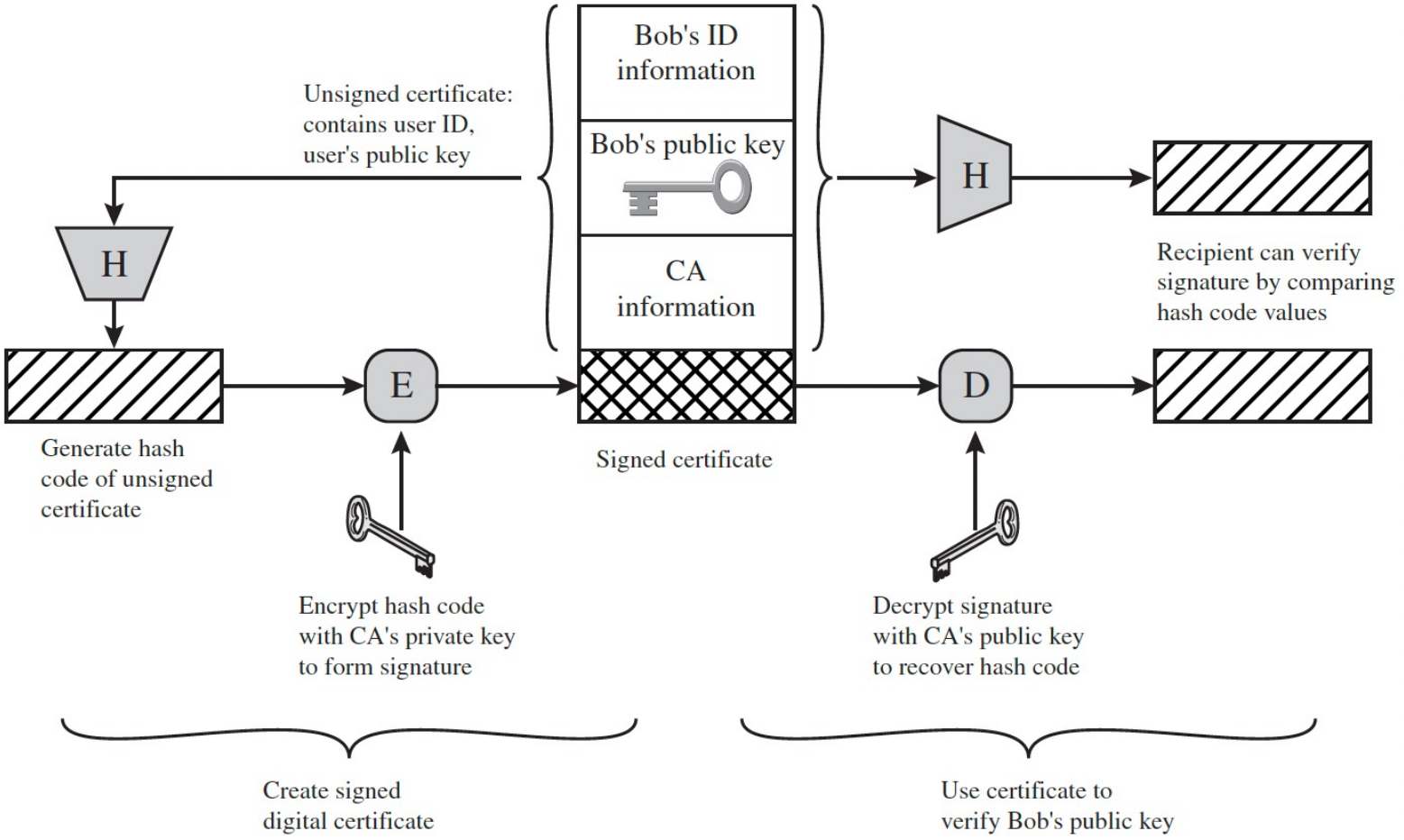
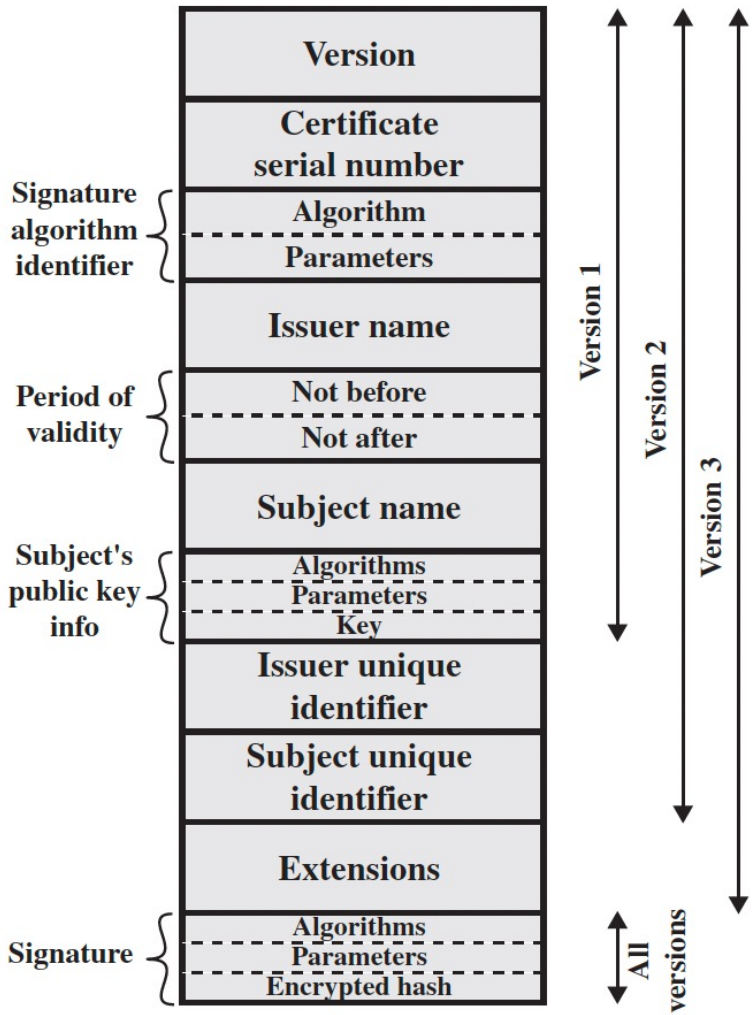


Figure 14.13 Public-Key Certificate Use

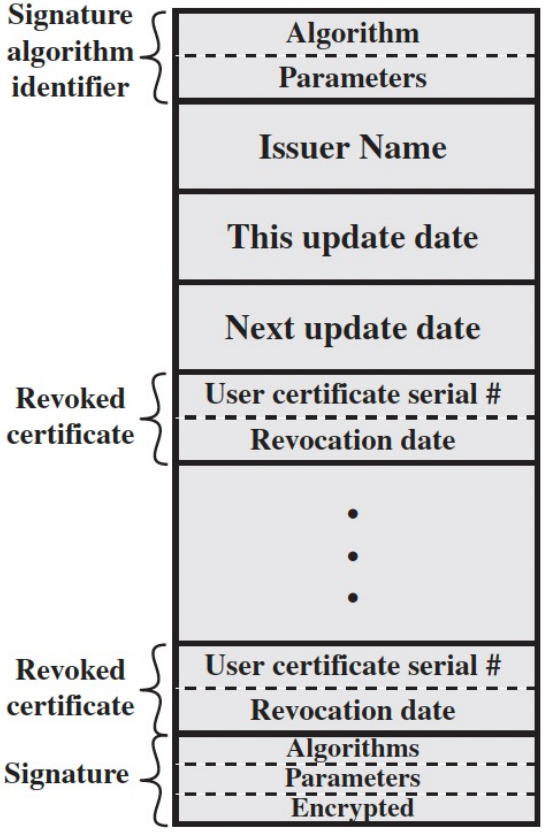
X.509 Certificates

- issued by a Certification Authority (CA), containing:
  - version (1, 2, or 3)
  - serial number (unique within CA) identifying certificate
  - signature algorithm identifier
  - issuer X.500 name (CA)
  - period of validity (from - to dates)
  - subject X.500 name (name of owner)
  - subject public-key info (algorithm, parameters, key)
  - issuer unique identifier (v2+)
  - subject unique identifier (v2+)
  - extension fields (v3)
  - signature (of hash of all fields in certificate)
- notation `CA<<A>>` denotes certificate for A signed by CA

X.509 Certificates



(a) X.509 certificate



(b) Certificate revocation list

Obtaining a Certificate

- any user with access to CA can get any certificate from it
- only the CA can modify a certificate
- because cannot be forged, certificates can be placed in a public directory

CA Hierarchy

- if both users share a common CA then they are assumed to know its public key
- otherwise CA's must form a hierarchy
- use certificates linking members of hierarchy to validate other CA's
  - each CA has certificates for clients (forward) and parent (backward)
- each client trusts parents certificates
- enable verification of any certificate from one CA by users of all other CAs in hierarchy

CA Hierarchy Use

- Track chains of certificates:
- A acquires B certificate using chain:
  X<<W>>W<<V>>V<<Y>>Y<<Z>>Z<<B>>
- B acquires A certificate using chain:
  Z<<Y>>Y<<V>>V<<W>>W<<X>>X<<A>>

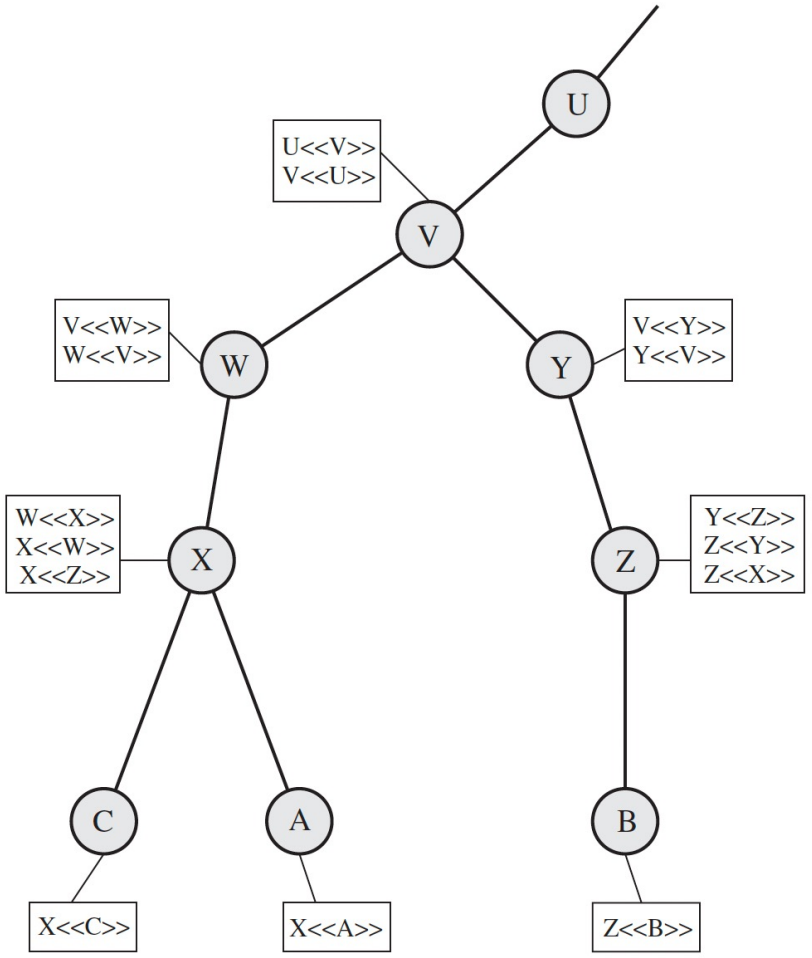


Figure 14.15 X.509 Hierarchy: A Hypothetical Example

Certificate Revocation

- certificates have a period of validity
- Typically, a new certificate is issued just before the expiration of the old one.
- may need to revoke before expiry for one of the following reasons:
  1. user's private key is compromised
  2. user is no longer certified by this CA
  3. CA's certificate is compromised
- CA's maintain list of revoked certificates
  - the Certificate Revocation List (CRL)
- users should check certs with CA's CRL

Authentication Procedures

- X.509 includes three alternative authentication procedures:
  - One-Way Authentication
  - Two-Way Authentication
  - Three-Way Authentication
- all use public-key signatures

One-Way Authentication

- 1 message (A→B) used to establish
  - the identity of A and that message is from A
  - message was intended for B
  - integrity & originality of message
- message must include timestamp, nonce, B's identity and is signed by A

Two-Way Authentication

- 2 messages (A→B, B→A) which also establishes in addition to one-way authentication:
  - the identity of B and that reply is from B
  - that reply is intended for A
  - integrity & originality of reply
- reply includes original nonce from A, also timestamp and nonce from B
- requires synchronised clocks

Three-Way Authentication

- 3 messages ($A \rightarrow B, B \rightarrow A, A \rightarrow B$) which enables above authentication without synchronized clocks
- has reply from A back to B containing signed copy of nonce from B
- means that timestamps need not be checked or relied upon

X.509 Version 3

- has been recognised that additional information is needed in a certificate
  - email/URL, policy details, usage constraints
- rather than explicitly naming new fields defined a general extension method
- extensions consist of:
  - extension identifier
  - criticality indicator
  - extension value

Certificate Extensions

- key and policy information
  - convey info about subject & issuer keys, plus indicators of certificate policy
- certificate subject and issuer attributes
  - support alternative names, in alternative formats for certificate subject and/or issuer
- certificate path constraints
  - allow constraints on use of certificates by other CA's

Email Security

- Email security refers to the collective measures , techniques and procedures used to protect email accounts, contents, and communication against unauthorized access and loss or compromise.
- Email is often used to spread malware, spam and phishing attacks.
- It allows an individual or organization to protect the overall access to one or more email addresses/accounts.
- An email service provider implements email security to secure subscriber email accounts and data from hackers - at rest and in transit.
- Email security is a broad term that encompasses multiple techniques used to secure an email service.

Email Security

- From an individual/end user standpoint, proactive email security measures include:
  - Strong passwords
  - Regular password update
  - Spam filters
  - Desktop-based anti-virus/anti-spam applications
- Similarly, a service provider ensures email security by using strong password and access control mechanisms on an email server; encrypting and digitally signing email messages when in the inbox or in transit to or from a subscriber email address.
- Two standard schemes used for email security are PGP and S/MIME

Pretty Good Privacy (PGP)

- developed by Phil Zimmermann, who
  - selected best available cryptographic algorithms :
    - RSA, DSS, and Diffie-Hellman for public-key encryption;
    - CAST-128, IDEA, and 3DES for symmetric encryption; and
    - SHA-1 for hash coding
  - integrated into a general-purpose application
    - using simple and small commands (independent of OS and processors)
    - that can be implemented on Unix, PC, Macintosh and other systems
  - made the package and its documentation, including the source code, freely available.
  - did agreement with a company (PGP Inc., Network Associates, PGP Corp. and now Symantec) to provide a fully compatible, low-cost commercial version of PGP.
- Some PGP compliant free products are OpenPGP, Gnu Privacy Guard (GPG) etc.

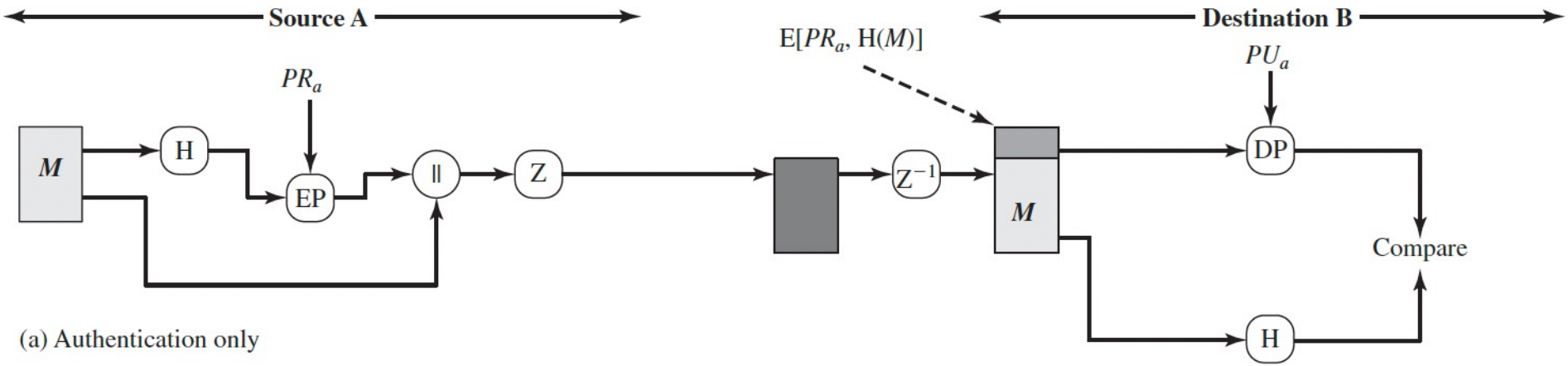
PGP Services

- The entire PGP operation provides four services - authentication, confidentiality, compression, and e-mail compatibility.

Function	Algorithms Used	Description
Digital signature	DSS/SHA or RSA/SHA	A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key and included with the message.
Message encryption	CAST or IDEA or Three-key Triple DES with Diffie-Hellman or RSA	A message is encrypted using CAST-128 or IDEA or 3DES with a one-time session key generated by the sender. The session key is encrypted using Diffie-Hellman or RSA with the recipient's public key and included with the message.
Compression	ZIP	A message may be compressed for storage or transmission using ZIP.
E-mail compatibility	Radix-64 conversion	To provide transparency for e-mail applications, an encrypted message may be converted to an ASCII string using radix-64 conversion.

PGP Operation – Authentication

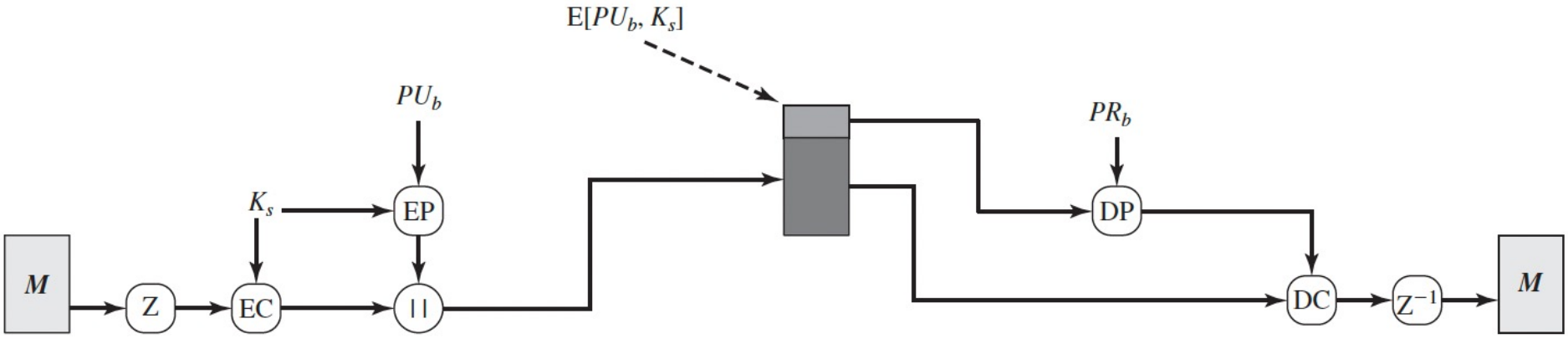
1. sender creates a message
2. SHA-1 used to generate 160-bit hash code of message
3. hash code is encrypted with RSA using the sender's private key, and result is attached to message
4. receiver uses RSA or DSS with sender's public key to decrypt and recover hash code
5. receiver generates new hash code for message and compares with decrypted hash code, if match, message is accepted as authentic



(a) Authentication only

PGP Operation – Confidentiality

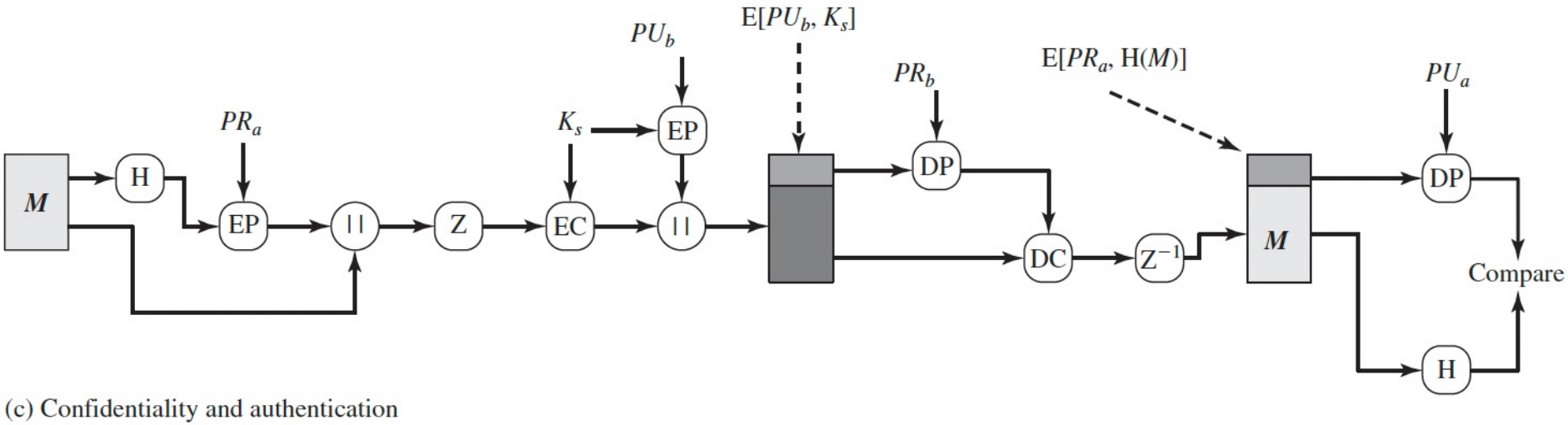
1. sender generates message and random 128-bit number to be used as session key for this message only
2. message is encrypted, using CAST-128 / IDEA/3DES with session key
3. session key is encrypted using RSA with recipient's public key, then attached to message
4. receiver uses RSA with its private key to decrypt and recover session key
5. session key is used to decrypt message



(b) Confidentiality only

PGP Operation – Confidentiality & Authentication

- can use both services on the same message
  - create signature & attach to message
  - encrypt both message & signature
  - attach RSA/ElGamal encrypted session key



(c) Confidentiality and authentication

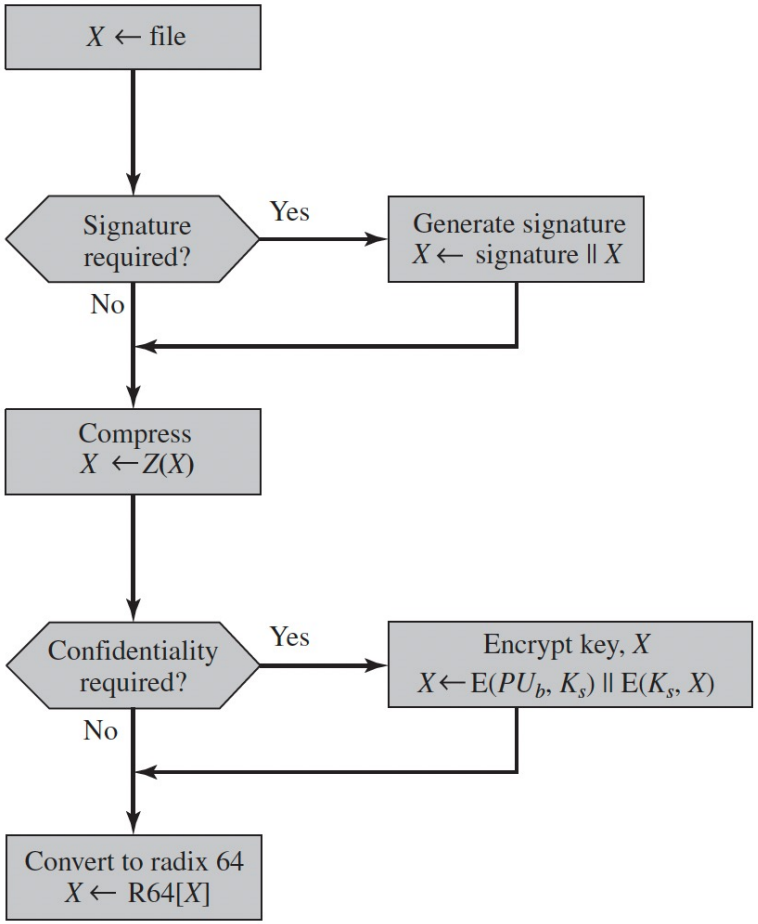
PGP Operation – Compression

- 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

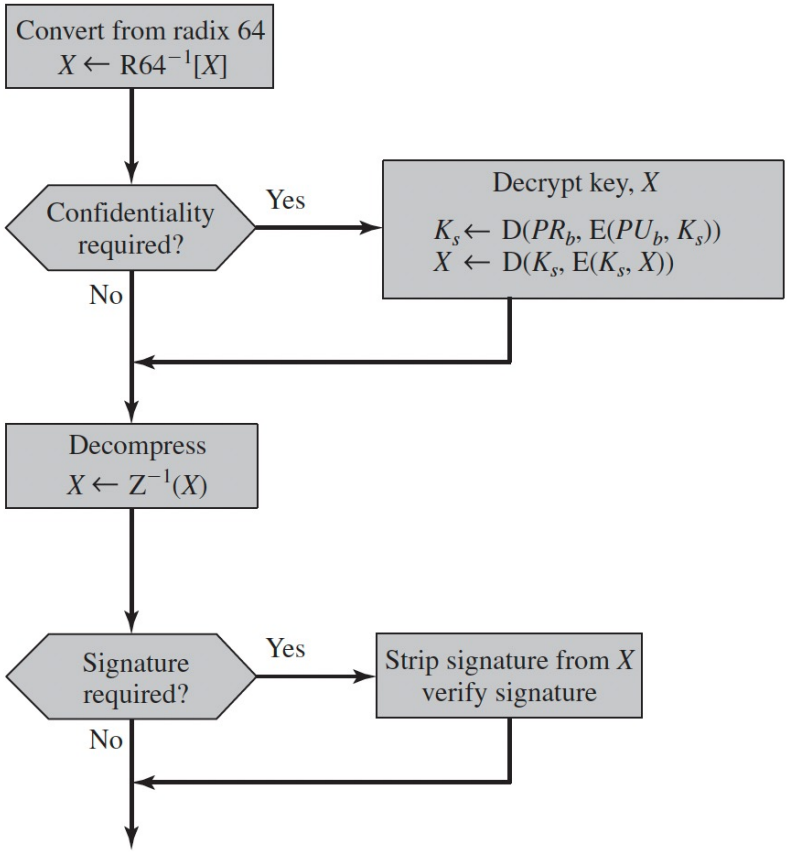
PGP Operation – Email Compatibility

- 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

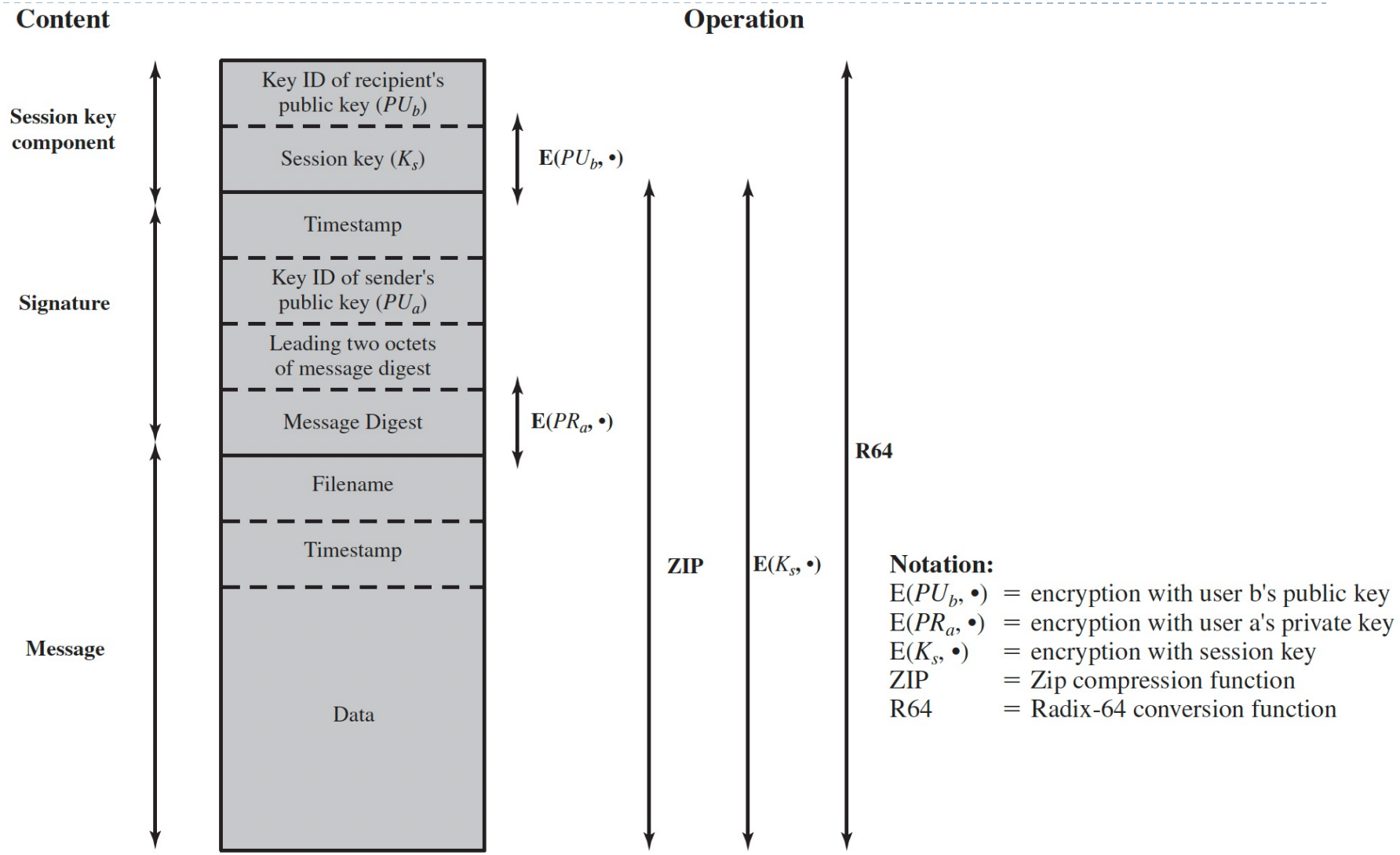


(a) Generic transmission diagram (from A)



(b) Generic reception diagram (to B)

PGP Message Format



General Format PGP Message (from A 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 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

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

Web Security

- Web security is mainly concerned with three standardized schemes: -
  - SSL/TLS,
  - HTTPS, and
  - SSH.
- Web security challenges: -
  - Web is vulnerable to attacks on the Web servers over the Internet.
  - Reputations and money can be damaged / lost if the Web servers are subverted.
  - The complexity of underlying software may hide many potential security flaws.
  - A Web server can be exploited as a launching pad for an attack and thus may gain access to data and systems not part of the Web itself but connected to the server at the local site.
  - Casual and untrained users of the web are not aware of the security risks that exist and do not have the tools or knowledge to take effective countermeasures.

Threats to Web Security

Two categories: -

- Active or Passive attacks.
- Locations where attacks occur viz.
  1. Web server,
  2. Web browser, and
  3. Network traffic

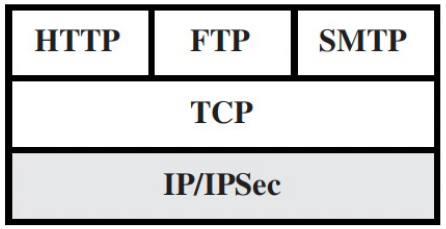
  between browser and server.

Table 16.1 A Comparison of Threats on the Web

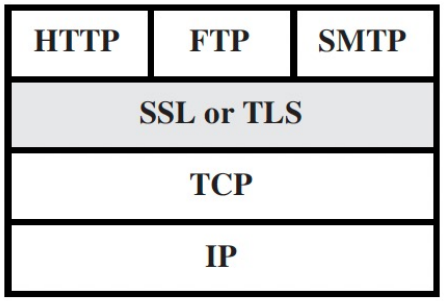
	Threats	Consequences	Countermeasures
Integrity	• Modification of user data • Trojan horse browser • Modification of memory • Modification of message traffic in transit	• Loss of information • Compromise of machine • Vulnerabilty to all other threats	Cryptographic checksums
Confidentiality	• Eavesdropping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

Approaches to Web Scurity

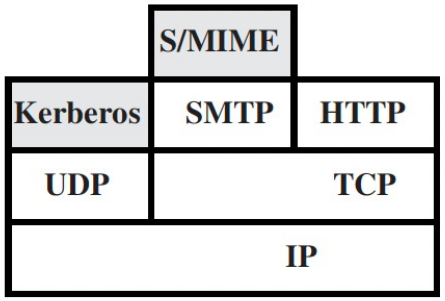
- IPsec: -
  - is transparent to end users and applications and provides a general-purpose solution.
  - includes a filtering capability so that only selected traffic need incur the overhead of IPsec processing.
- Security just above TCP (SSL/TLS) : -
  - Either could be provided as part of the underlying protocol suite (transparent to applications)
  - Or, could be embedded in specific packages like Netscape and IE.
- Application-specific security : -
  - services are embedded within the particular application.



(a) Network level



(b) Transport level



(c) Application level

Secure Socket Layer

- SSL was originally developed by Netscape
- SSL v3, designed with public input, became Internet standard known as Transport Layer Security (TLS)
- It uses underlying protocol layer (i.e. TCP) to provide a reliable end-to-end service
- It is used for the secure transmission of documents over a network.
- The objectives of SSL are:
  - Data integrity: Data is protected from tampering.
  - Data privacy: Data privacy is ensured through a series of protocols, including the SSL Record Protocol, SSL Handshake Protocol, SSL Change Cipher Spec Protocol and SSL Alert Protocol.

SSL Architecture

- SSL has two layers of protocols which are implemented above the TCP layer
- The SSL Record Protocol provides basic security services to various higher layer protocols e.g. Hypertext Transfer Protocol (HTTP).
- Three higher-layer protocols are defined as part of SSL:
  - Handshake Protocol,
  - Change Cipher Spec Protocol, and
  - Alert Protocol.

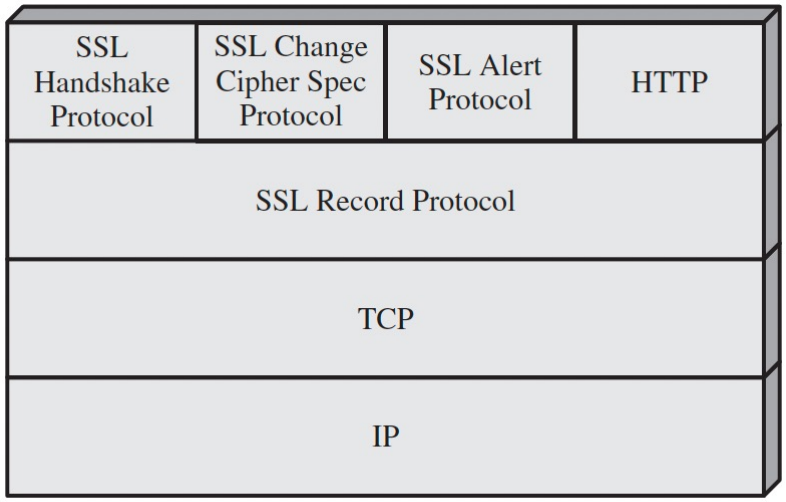


Figure 16.2 SSL Protocol Stack

SSL Architecture

Two SSL concepts: -

- SSL connection
  - a transient, peer-to-peer, communications link (in the OSI layering model definition)
  - Every SSL connection is associated with one SSL session
- SSL session
  - an association between client & server
  - created by the Handshake Protocol
  - defines a set of cryptographic parameters which can be shared by multiple SSL connections
- A connection and session are defined by their states.

SSL Architecture

Parameters of a connection state : -

- Server and client random
- Server write MAC secret
- Client write MAC secret
- Server write key
- Client write key
- Initialization vectors
- Sequence numbers

Parameters of a session state: -

- Session identifier
- Peer certificate
- Compression method
- Cipher spec
- Master secret
- Is resumable

- Once a session is established, there is a current operating state for both read and write (i.e., receive and send).
- During the Handshake Protocol, pending read and write states are created.
- Upon successful conclusion of the Handshake Protocol, the pending states become the current states.

SSL Record Protocol Services

- Confidentiality
  - uses 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
- Message Integrity
  - uses a MAC with shared secret key
  - similar to HMAC but with different padding

SSL Record Protocol Operation

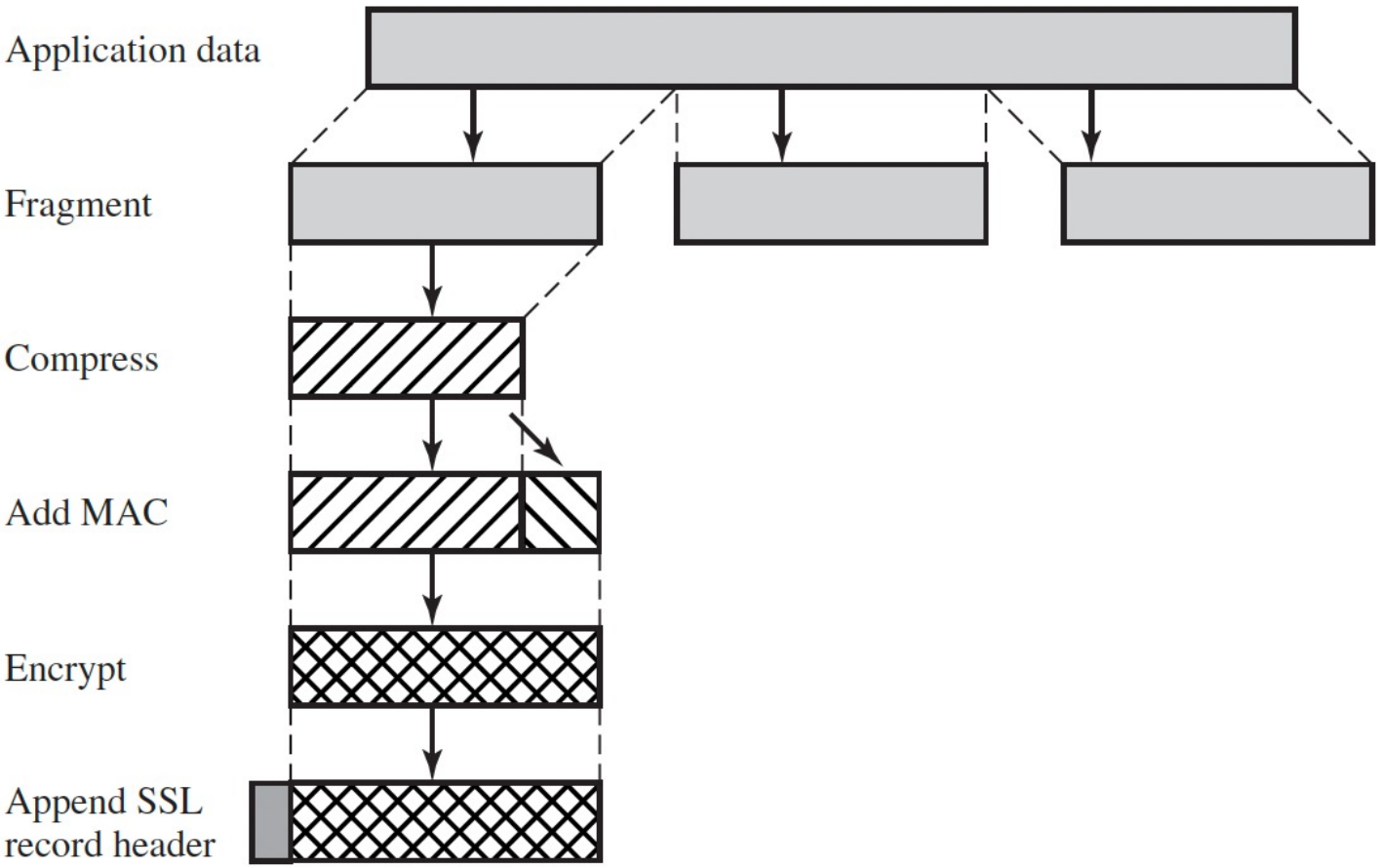


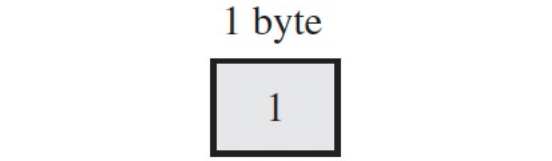
Figure 16.3 SSL Record Protocol Operation

SSL Record Protocol Operation

- In SSL Record Protocol application data is divided into fragments(2^{14} bytes).
- The fragment is compressed (Lossless compression therefore kept optional) and then encrypted MAC (Message Authentication Code) generated by algorithms like SHA (Secure Hash Protocol) and MD5 (Message Digest) is appended.
- After that encryption (Symmetric) of the data is done and in last SSL header(40 bits) is appended to the data.

SSL Change Cipher Spec Protocol

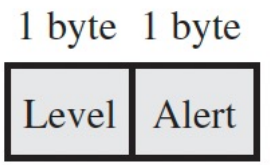
- This protocol uses SSL record protocol.
- Unless Handshake Protocol is completed, the SSL record Output will be in pending state.
- After handshake protocol the Pending state is converted into Current state.
- Change-cipher protocol consists of single message which is 1 byte in length and can have only one value (1).
- This protocol purpose is to cause the pending state to be copied into current state.



(a) Change Cipher Spec Protocol

SSL Alert Protocol

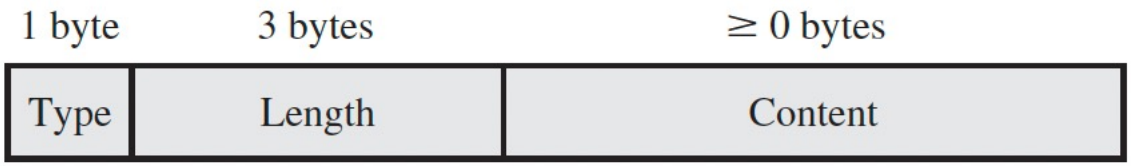
- conveys SSL-related alerts to peer entity
- alert messages are compressed and encrypted like all SSL data
- Severity:
  - The first byte takes the value warning (1) or fatal (2)
  - The second byte contains a code that indicates the specific alert.
- Specific alerts
  - fatal: unexpected message, bad record mac, decompression failure, handshake failure, illegal parameter
  - warning: close notify, no certificate, bad certificate, unsupported certificate, certificate revoked, certificate expired, certificate unknown



(b) Alert Protocol

SSL Handshake Protocol

- Allows server & client to:
  - authenticate each other
  - to negotiate encryption & MAC algorithms
  - to negotiate cryptographic keys to be used
- SSL Handshake protocol is used before transmitting of any application data.

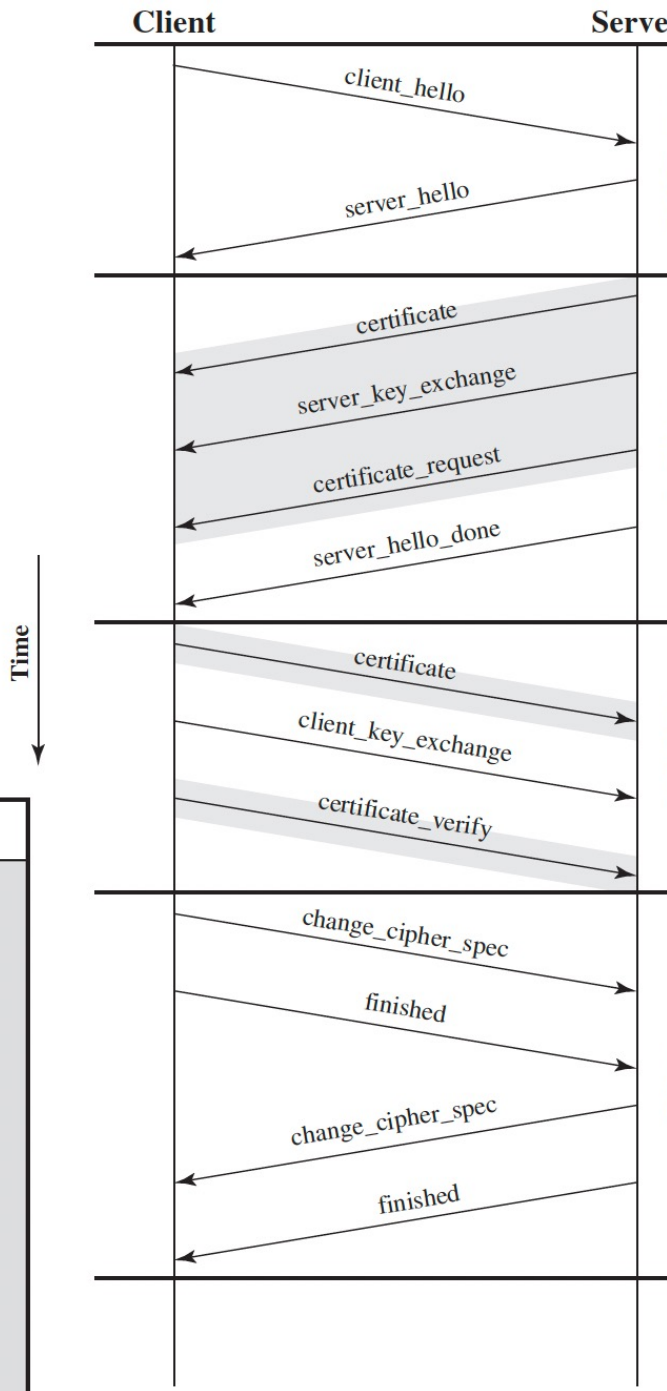


(c) Handshake Protocol

SSL Handshake Protocol

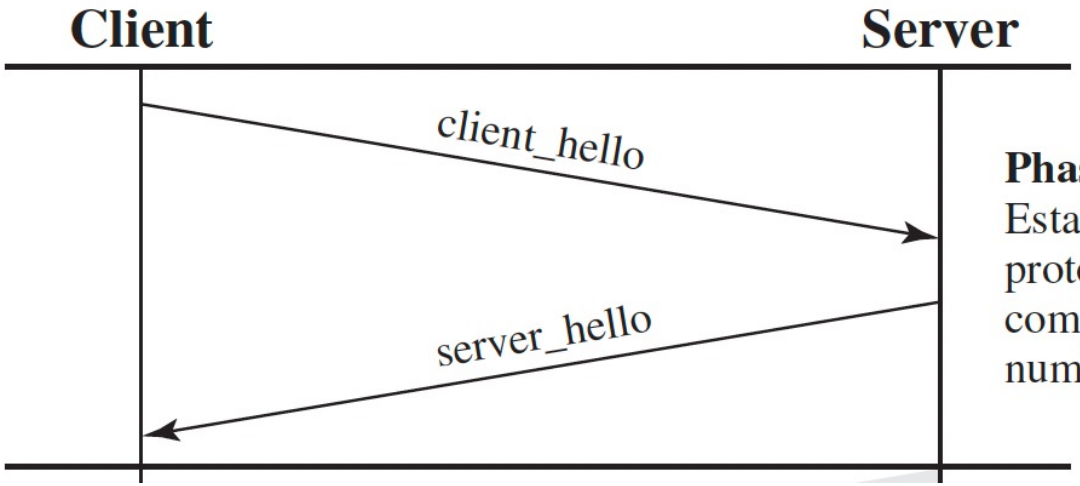
- Has a series of message exchanges in phases
  - Establish Security Capabilities
  - Server Authentication and Key Exchange
  - Client Authentication and Key Exchange
  - Finish

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



ure 16.6 Handshake Protocol Action

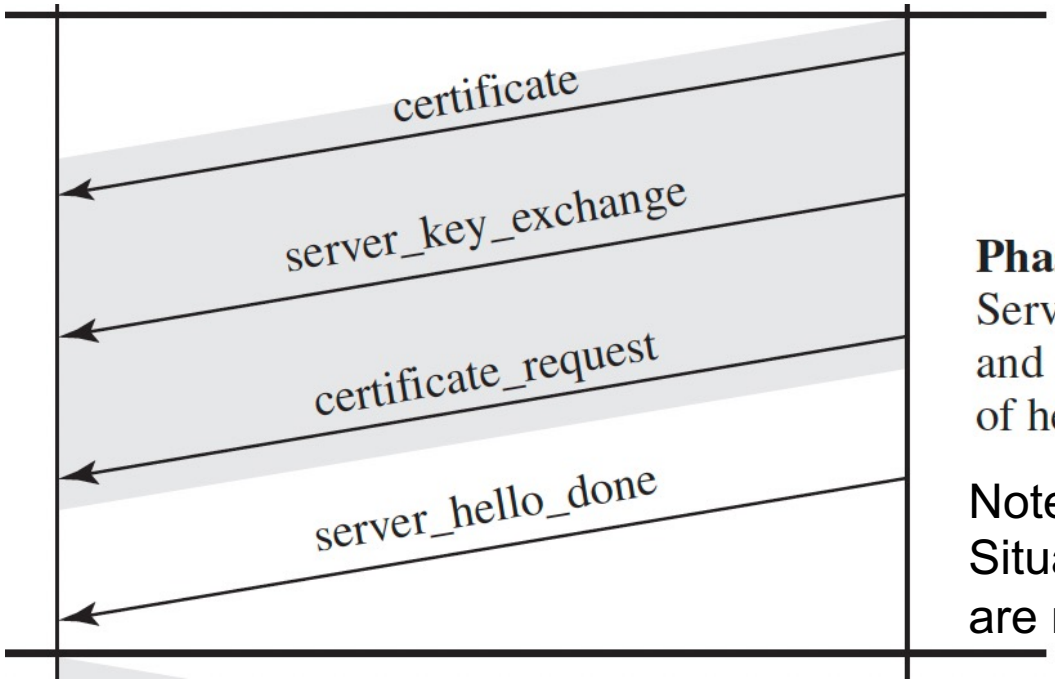
SSL Handshake Protocol



Phase 1
Establish security capabilities, including protocol version, session ID, cipher suite, compression method, and initial random numbers.

`client_hello`	version, random, session id, cipher suite, compression method
`server_hello`	version, random, session id, cipher suite, compression method

SSL Handshake Protocol

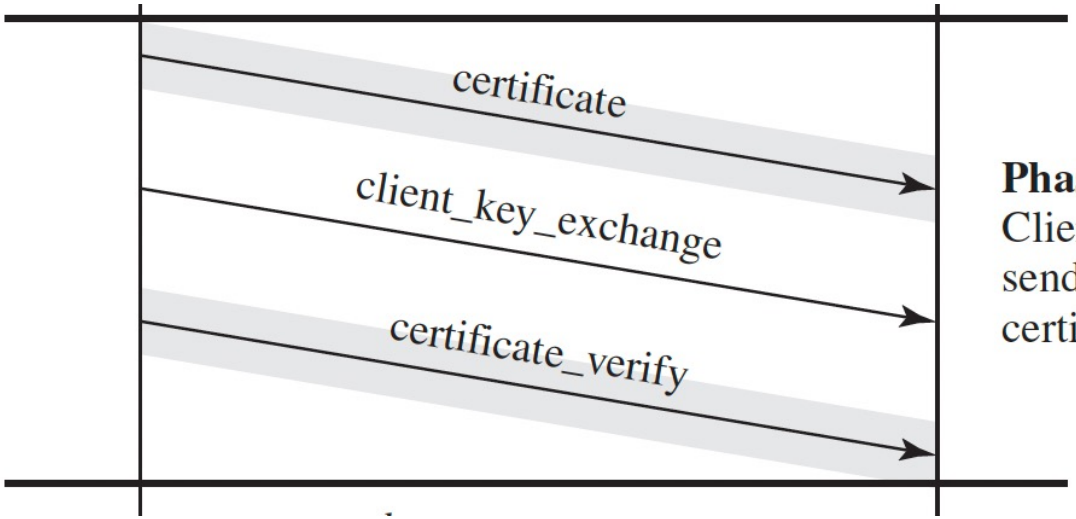


Phase 2
Server may send certificate, key exchange, and request certificate. Server signals end of hello message phase.

Note: Shaded transfers are optional or Situation dependent messages that are not always sent.

certificate	chain of X.509v3 certificates
server_key_exchange	parameters, signature
certificate_request	type, authorities
server_done	null

SSL Handshake Protocol

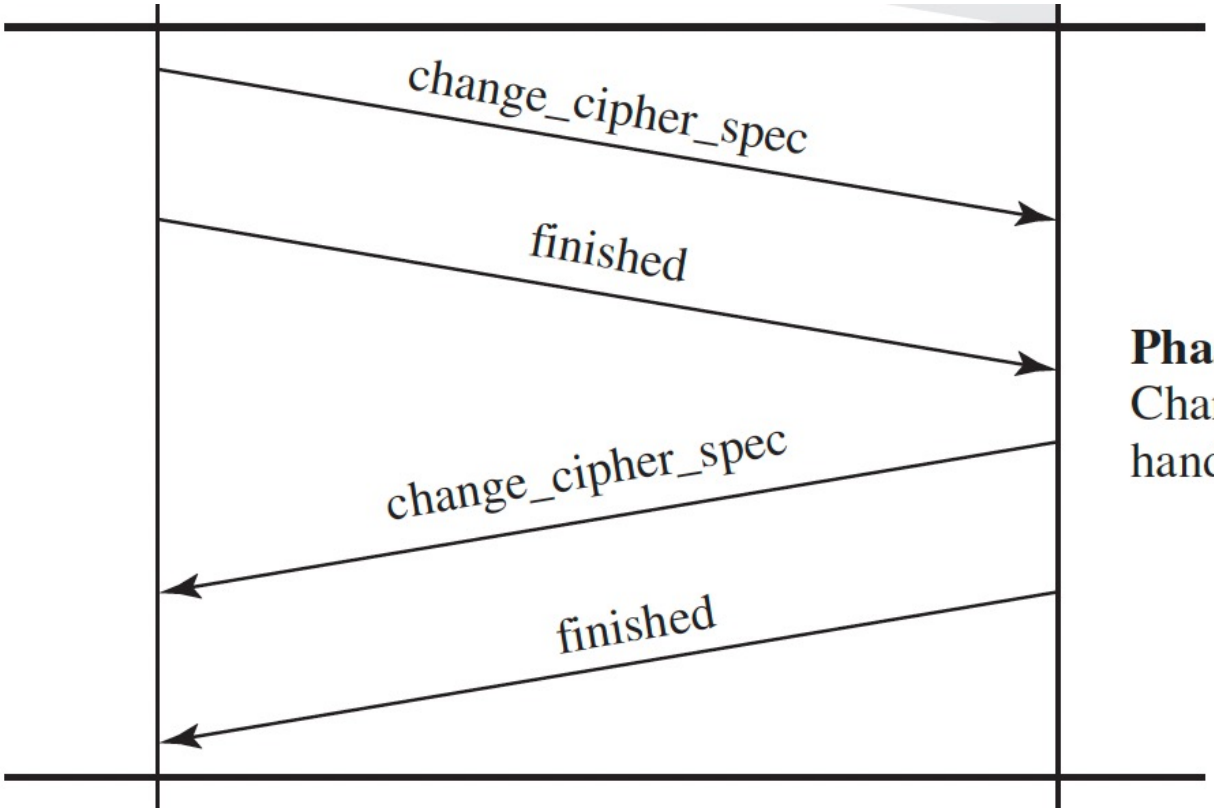


Phase 3
Client sends certificate if requested. Client sends key exchange. Client may send certificate verification.

Note: Shaded transfers are optional or Situation dependent messages that are not always sent.

certificate_verify	signature
client_key_exchange	parameters, signature

SSL Handshake Protocol



Phase 4
Change cipher suite and finish handshake protocol.

`finished`	hash value

Transport Layer Security (TLS)

- IETF standard RFC 5246 similar to SSLv3
- with minor differences
  - in record format version number
  - uses HMAC for MAC
  - a pseudo-random function (PRF) expands secrets
    - based on HMAC using SHA-1 or MD5
  - has additional alert codes
  - some changes in supported ciphers
  - changes in certificate types & negotiations
  - changes in cryptographic computations & padding

Transport Layer Security (TLS)

- One of the most widely used security services is Transport Layer Security (TLS); the current version is Version 1.3, defined in RFC 8446.
- In SSL, MD is used whereas in TLS pseudo random function is used.
- TLS is an Internet standard that evolved from a commercial protocol known as Secure Sockets Layer (SSL).
- Although SSL implementations are still around, it has been deprecated by IETF and is disabled by most corporations offering TLS software.

TLS Architecture

- TLS has two layers of protocols which are implemented above the TCP layer
- The TLS Record Protocol provides basic security services to various higher layer protocols e.g. Hypertext Transfer Protocol (HTTP).
- Three higher-layer protocols are defined as part of TLS:
  - Handshake Protocol,
  - Change Cipher Spec Protocol, and
  - Alert Protocol.

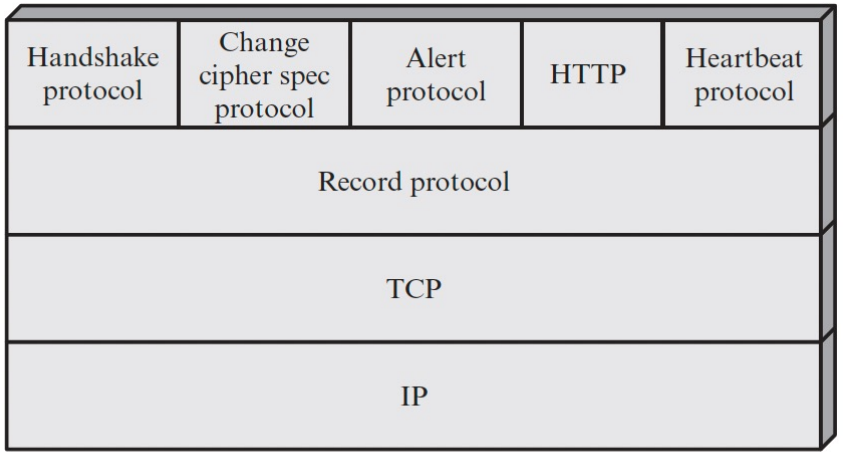


Figure 17.2 TLS Protocol Stack

TLS Architecture

Two TLS concepts: -

- TLS connection
  - a transient, peer-to-peer, communications link (in the OSI layering model definition)
  - Every TLS connection is associated with one TLS session
- TLS session
  - an association between client & server
  - created by the Handshake Protocol
  - defines a set of cryptographic parameters which can be shared by multiple TLS connections
- A connection and session are defined by their states.

TLS Architecture

Parameters of a connection state : -

- Server and client random
- Server write MAC secret
- Client write MAC secret
- Server write key
- Client write key
- Initialization vectors
- Sequence numbers

Parameters of a session state: -

- Session identifier
- Peer certificate
- Compression method
- Cipher spec
- Master secret
- Is resumable

- Once a session is established, there is a current operating state for both read and write (i.e., receive and send).
- During the Handshake Protocol, pending read and write states are created.
- Upon successful conclusion of the Handshake Protocol, the pending states become the current states.

TLS Record Protocol Services

- Confidentiality
  - uses symmetric encryption with a shared secret key defined by Handshake Protocol
  - AES or 3DES in block cipher mode
  - RC4-128 in stream cipher mode
  - message is compressed before encryption
- Message Integrity
  - uses a MAC with shared secret key
  - makes use of the HMAC algorithm defined in RFC 2104

TLS Record Protocol Operation

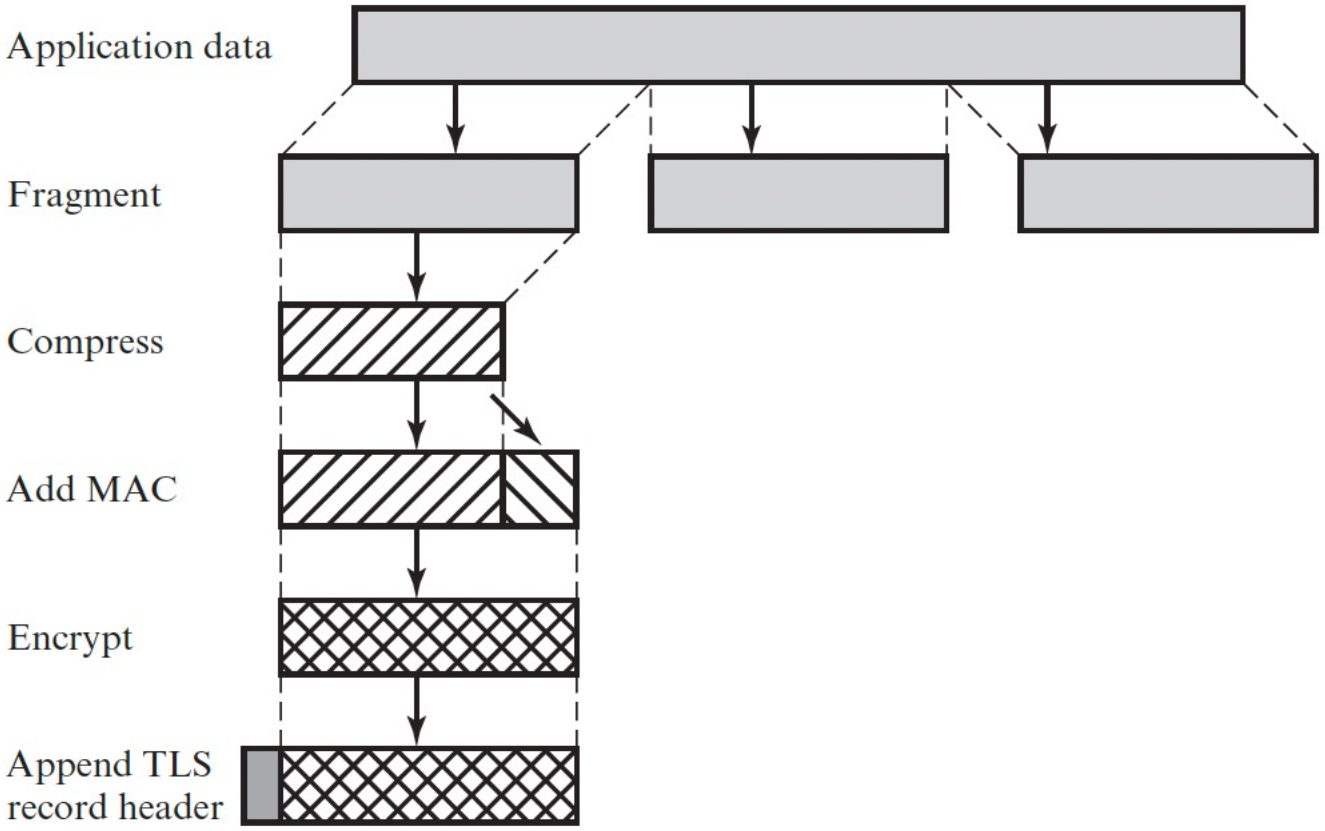


Figure 17.3 TLS Record Protocol Operation

TLS Record Protocol Operation

- In TLS Record Protocol application data is divided into fragments(2^{14} bytes).
- The fragment is optionally compressed (Lossless compression therefore kept optional).
- The next step in processing is to compute a MAC over the compressed data. TLS makes use of the HMAC algorithm defined in RFC 2104.
- After that encryption (Symmetric) of the data is done.

  The encryption algorithms permitted:

Block Cipher		Stream Cipher	
Algorithm	**Key Size**	**Algorithm**	**Key Size**
AES 3DES	128, 256 168	RC4-128	128

- and at last, TLS header(5 bytes) is appended to the data.

TLS Record Format

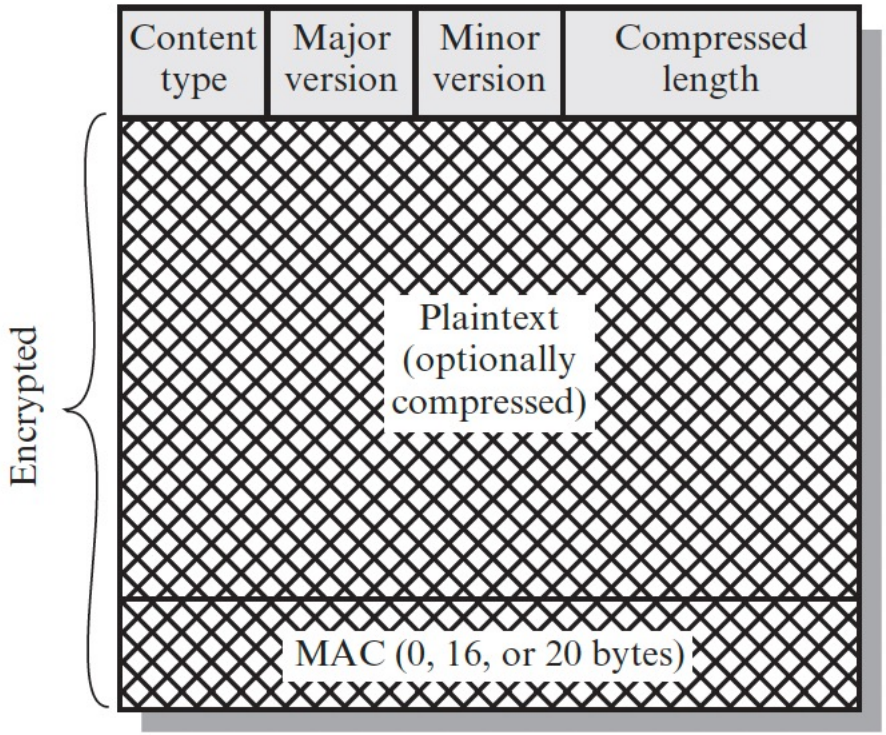
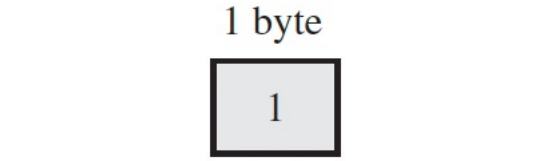


Figure 17.4 TLS Record Format

TLS Change Cipher Spec Protocol

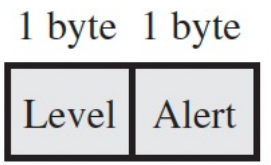
- This protocol uses TLS record protocol.
- Unless Handshake Protocol is completed, the TLS record Output will be in pending state.
- After handshake protocol the Pending state is converted into Current state.
- Change-cipher protocol consists of single message which is 1 byte in length and can have only one value (1).
- This protocol purpose is to cause the pending state to be copied into current state.



(a) Change Cipher Spec Protocol

TLS Alert Protocol

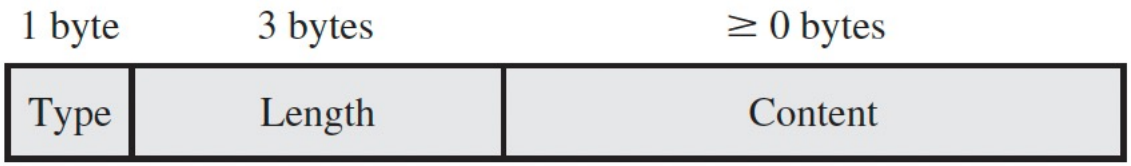
- conveys TLS-related alerts to peer entity
- alert messages are compressed and encrypted like all TLS data
- Severity:
  - The first byte takes the value warning (1) or fatal (2)
  - The second byte contains a code that indicates the specific alert.
- Specific alerts
  - fatal: unexpected message, bad record mac, decompression failure, handshake failure, illegal parameter
  - warning: close notify, no certificate, bad certificate, unsupported certificate, certificate revoked, certificate expired, certificate unknown



(b) Alert Protocol

TLS Handshake Protocol

- Allows server & client to:
  - authenticate each other
  - to negotiate encryption & MAC algorithms
  - to negotiate cryptographic keys to be used
- TLS Handshake protocol is used before transmitting of any application data.

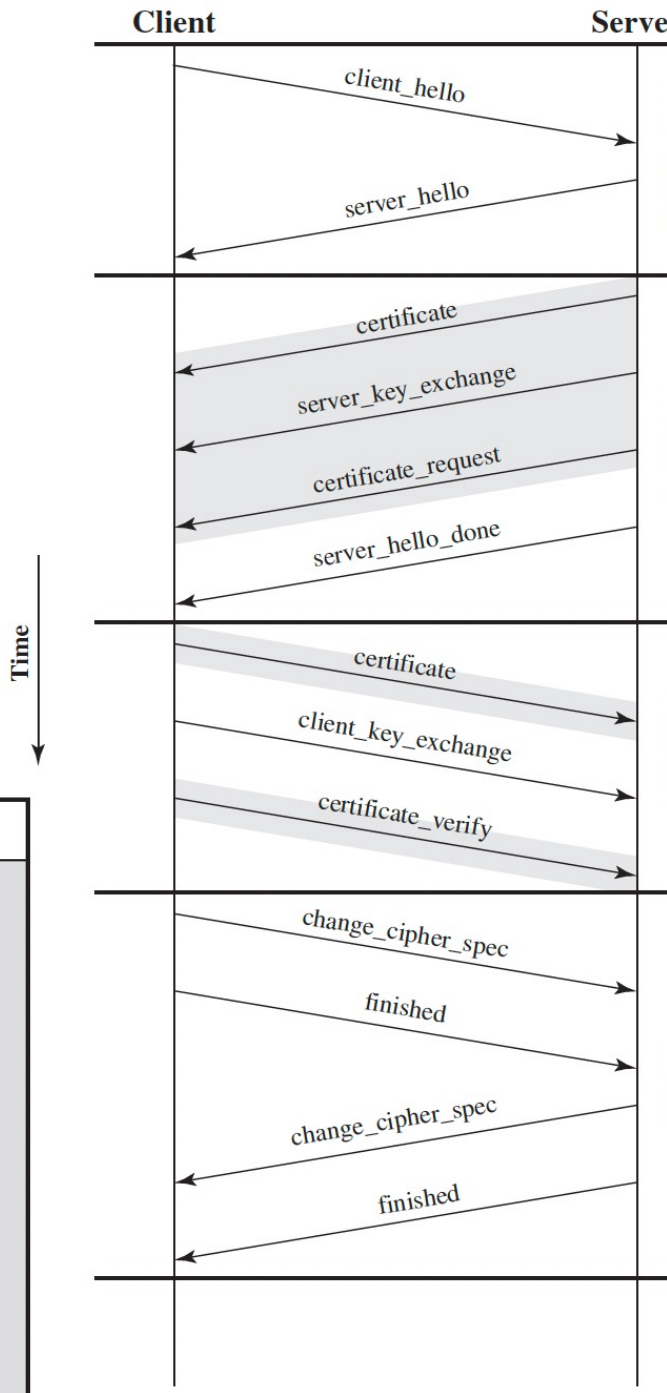


(c) Handshake Protocol

TLS Handshake Protocol

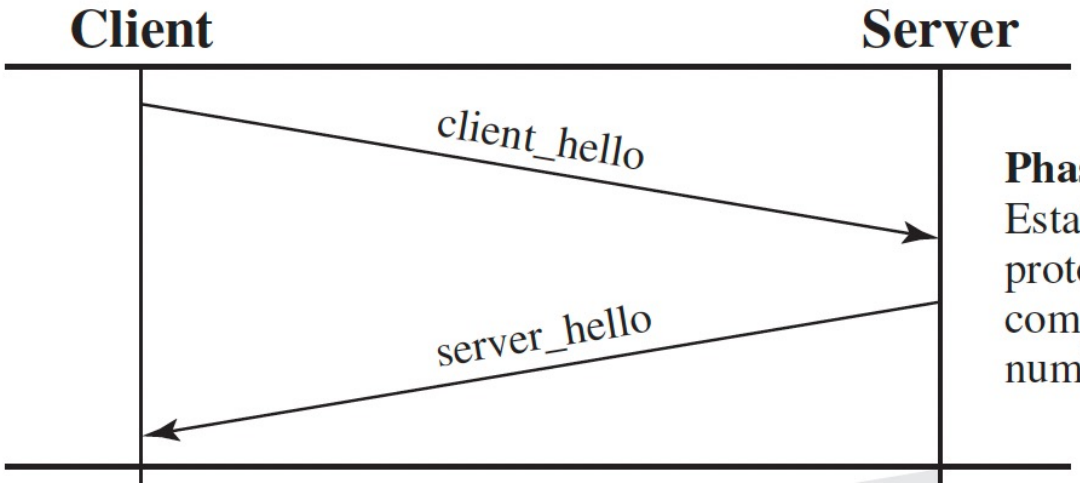
- Has a series of message exchanges in phases
  - Establish Security Capabilities
  - Server Authentication and Key Exchange
  - Client Authentication and Key Exchange
  - Finish

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



ure 16.6 Handshake Protocol Action

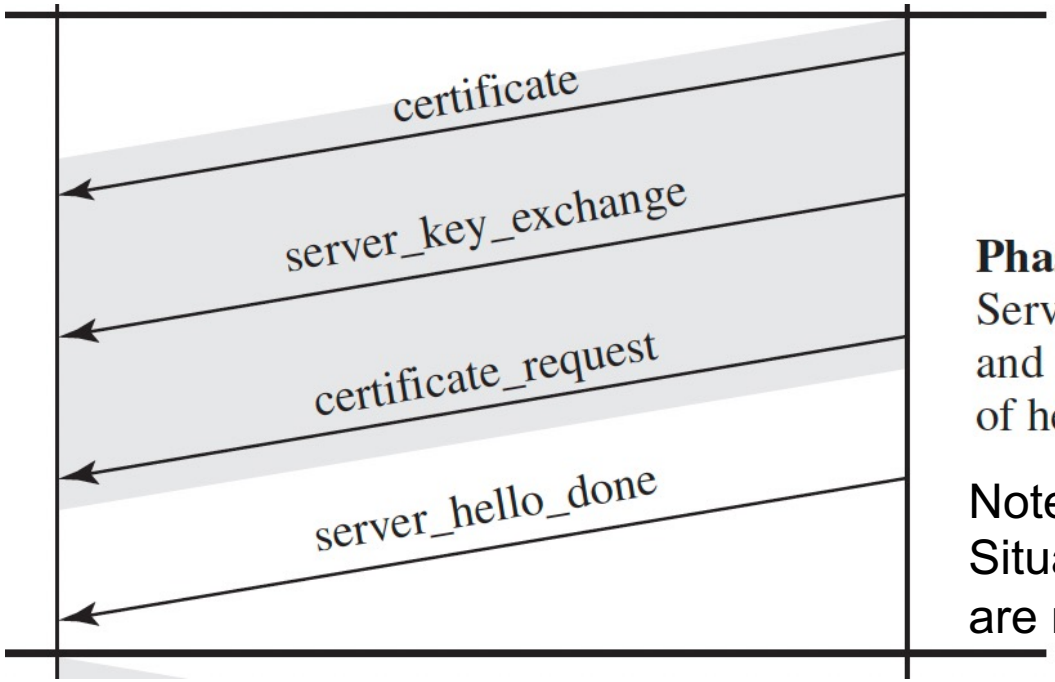
TLS Handshake Protocol



Phase 1
Establish security capabilities, including protocol version, session ID, cipher suite, compression method, and initial random numbers.

`client_hello`	version, random, session id, cipher suite, compression method
`server_hello`	version, random, session id, cipher suite, compression method

TLS Handshake Protocol

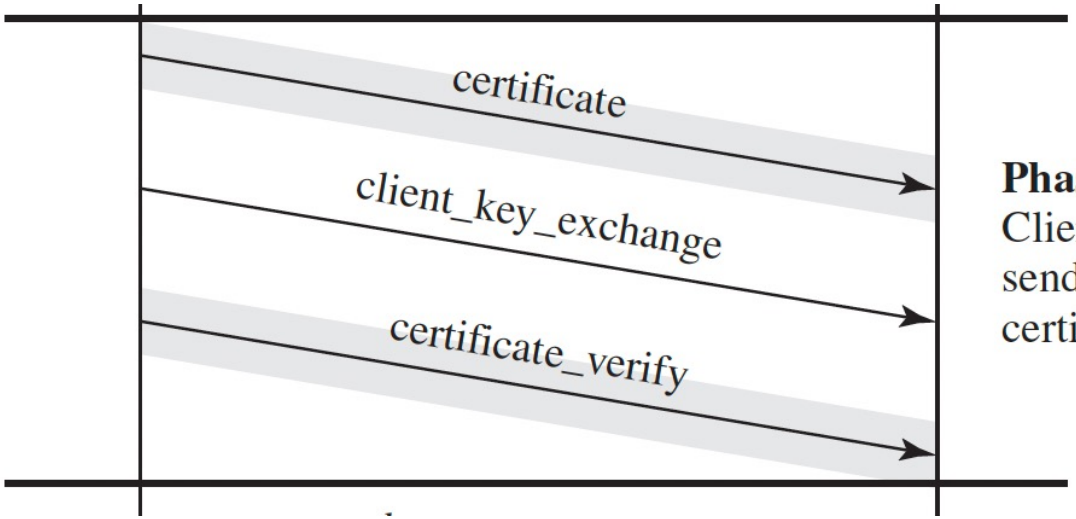


Phase 2
Server may send certificate, key exchange, and request certificate. Server signals end of hello message phase.

Note: Shaded transfers are optional or Situation dependent messages that are not always sent.

certificate	chain of X.509v3 certificates
server_key_exchange	parameters, signature
certificate_request	type, authorities
server_done	null

TLS Handshake Protocol

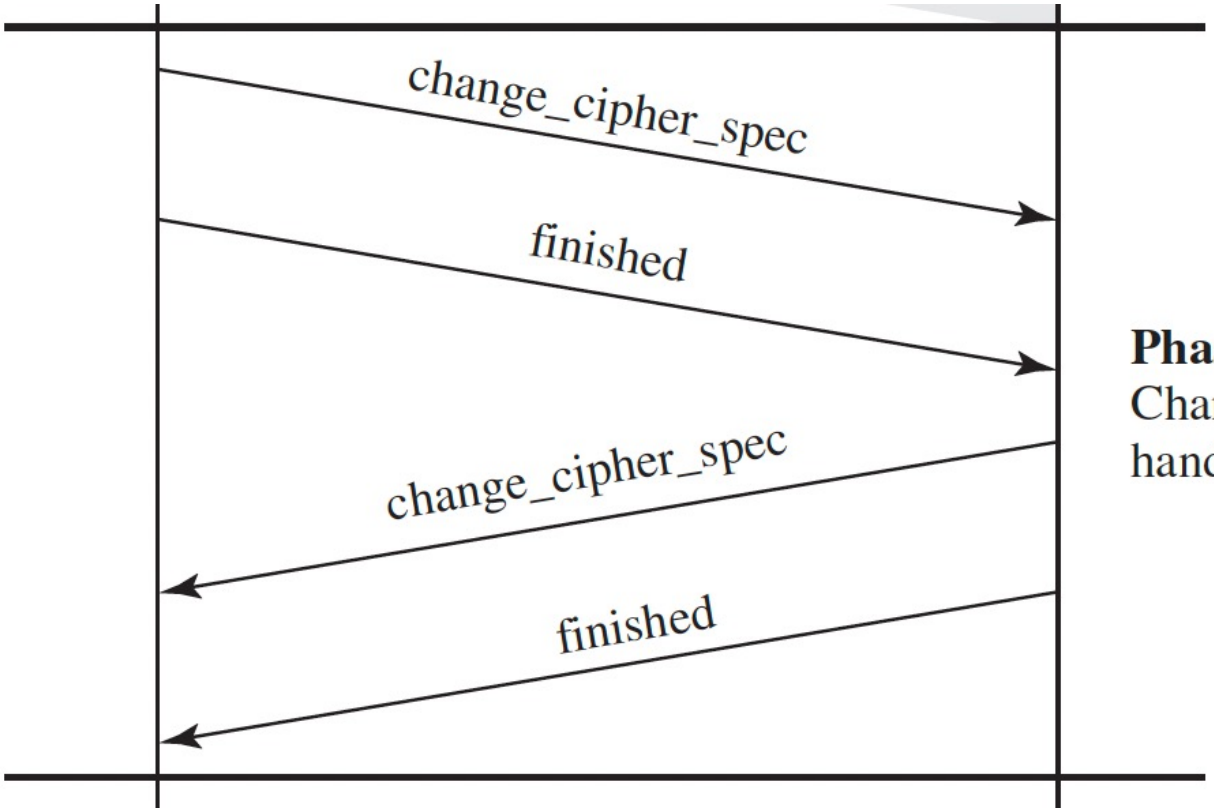


Phase 3
Client sends certificate if requested. Client sends key exchange. Client may send certificate verification.

Note: Shaded transfers are optional or Situation dependent messages that are not always sent.

certificate_verify	signature
client_key_exchange	parameters, signature

TLS Handshake Protocol



Phase 4
Change cipher suite and finish handshake protocol.

`finished`	hash value

TLS Heartbeat Protocol

- The Heartbeat protocol is a new protocol running on top of the Record Layer.
- The heartbeat is used to make sure that the peer is still alive.
- A HeartbeatRequest message can arrive almost at any time during the lifetime of a connection.
- Whenever a HeartbeatRequest message is received, it SHOULD be answered with a corresponding HeartbeatResponse message.

IPSec

- IPsec, also known as the Internet Protocol Security or IP Security protocol, defines the architecture for security services for IP network traffic.
- IPsec describes the framework for providing security at the IP layer, as well as the suite of protocols designed to provide that security, through authentication and encryption of IP network packets.
- Also included in IPsec are protocols that define the cryptographic algorithms used to encrypt, decrypt and authenticate packets, as well as the protocols needed for secure key exchange and key management.

IPSec

- IPsec originally defined two mechanisms for imposing security on IP packets:
  - Encapsulating Security Payload (ESP) protocol, which defined a method for encrypting data in IP packets, and
  - Authentication Header (AH) protocol, which defined a method for digitally signing IP packets.
- The Internet Key Exchange (IKE) protocol is used to manage the cryptographic keys used by hosts for IPsec.
- IPsec can be used to protect network data, for example, by setting up circuits using IPsec tunneling, in which all data being sent between two endpoints is encrypted, as with a Virtual Private Network (VPN) connection; for encrypting application layer data.

Applications of IPSec

IPsec provides the capability to secure communications across a LAN, across private and public WANs, and across the Internet. Examples of its use include:

- Secure branch office connectivity over the Internet:
  - A company can build a secure virtual private network over the Internet or over a public WAN.
  - This enables a business to rely heavily on the Internet and reduce its need for private networks, saving costs and network management overhead.
- Secure remote access over the Internet:
  - An end user whose system is equipped with IP security protocols can make a local call to an Internet Service Provider (ISP) and gain secure access to a company network.
  - This reduces the cost of toll charges for traveling employees and telecommuters.

Applications of IPSec

- Establishing extranet and intranet connectivity with partners:
  - IPsec can be used to secure communication with other organizations, ensuring authentication and confidentiality and providing a key exchange mechanism.
- Enhancing electronic commerce security:
  - Even though some Web and electronic commerce applications have built-in security protocols, the use of IPsec enhances that security.
  - IPsec guarantees that all traffic designated by the network administrator is both encrypted and authenticated, adding an additional layer of security to whatever is provided at the application layer.

IPSec Services

- IPsec provides security services at the IP layer by enabling a system to select required security protocols, determine the algorithm(s) to use for the service(s), and put in place any cryptographic keys required to provide the requested services.
- Two protocols are used to provide security:
  - an authentication protocol designated by the header of the protocol, Authentication Header (AH); and
  - a combined encryption/authentication protocol designated by the format of the packet for that protocol, Encapsulating Security Payload (ESP).

IPSec Services

- These are the services provided by IPSec:
  - Access control
  - Connectionless integrity
  - Data origin authentication
  - Rejection of replayed packets (a form of partial sequence integrity)
  - Confidentiality (encryption)

Modes of Operation

- Both AH and ESP support two modes of use: -
  - Transport mode and
  - Tunnel mode.

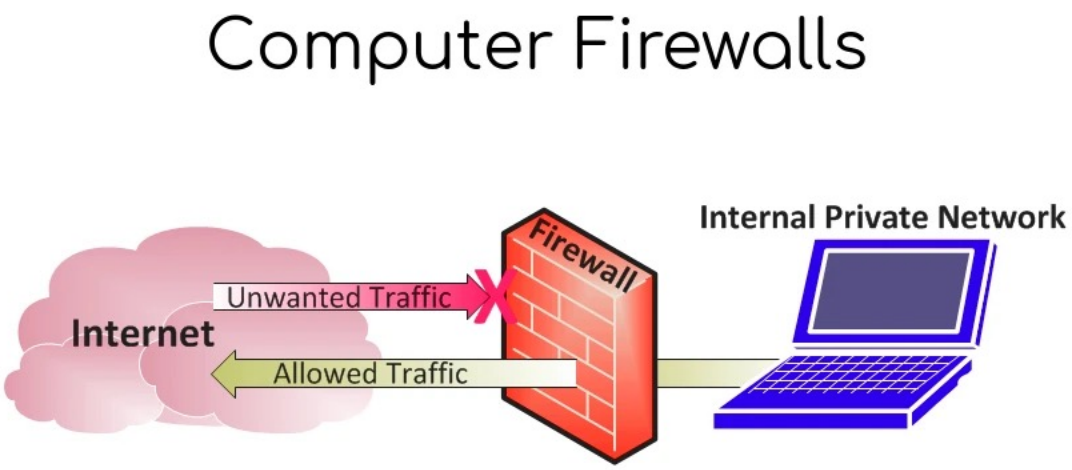
Table 19.1 Tunnel Mode and Transport Mode Functionality

	Transport Mode SA	Tunnel Mode SA
AH	Authenticates IP payload and selected portions of IP header and IPv6 extension headers.	Authenticates entire inner IP packet (inner header plus IP payload) plus selected portions of outer IP header and outer IPv6 extension headers.
ESP	Encrypts IP payload and any IPv6 extension headers following the ESP header.	Encrypts entire inner IP packet.
ESP with Authentication	Encrypts IP payload and any IPv6 extension headers following the ESP header. Authenticates IP payload but not IP header.	Encrypts entire inner IP packet. Authenticates inner IP packet.

Firewall

- A firewall is a network security device, either hardware or software-based, which monitors all incoming and outgoing traffic and based on a defined set of security rules it accepts, rejects or drops that specific traffic.
  - Accept : allow the traffic
  - Reject : block the traffic but reply with an "unreachable error"
  - Drop : block the traffic with no reply
- A firewall establishes a barrier between secured internal networks and outside untrusted network, such as the Internet.

Firewall



the first line of defense in network security

Characteristics of Firewall

Physical Barrier:

- A firewall does not allow any external traffic to enter a system or a network without its allowance.
- A firewall creates a choke point for all the external data trying to enter the system or network and hence can easily block access if needed.

Multi-Purpose:

- A firewall has many functions other than security purposes.
- It configures domain names and Internet Protocol (IP) addresses.
- It also acts as a network address translator. It can act as a meter for internet usage.

Characteristics of Firewall

Flexible Security Policies:

- Different local systems or networks need different security policies.
- A firewall can be modified according to the requirement of the user by changing its security policies.

Security Platform:

- It provides a platform from which any alert to the issue related to security or fixing issues can be accessed.
- All the queries related to security can be kept under check from one place in a system or network.

Characteristics of Firewall

Access Handler:

- Determines which traffic needs to flow first according to priority or can change for a particular network or system.
- Specific action requests may be initiated and allowed to flow through the firewall.

Types of Firewall

- A firewall can either be software or hardware.
- Software firewalls are programs installed on each computer, and they regulate network traffic through applications and port numbers.
- Meanwhile, hardware firewalls are the equipment established between the gateway and your network.
- Additionally, one may call a firewall delivered by a cloud solution as a cloud firewall.
- There are multiple types of firewalls based on their traffic filtering methods, structure, and functionality. A few of the types of firewalls are:

Types of Firewall

Packet Filtering

- A packet filtering firewall controls data flow to and from a network.
- It allows or blocks the data transfer based on the packet's source address, the destination address of the packet, the application protocols to transfer the data, and so on.

	Source IP	Dest. IP	Source Port	Dest. Port	Action
1	192.168.21.0	--	--	--	deny
2	--	--	--	23	deny
3	--	192.168.21.3	--	--	deny
4	--	192.168.21.0	--	>1023	Allow

Sample Packet Filter Firewall Rule

1. Incoming packets from network 192.168.21.0 are blocked.
2. Incoming packets destined for internal TELNET server (port 23) are blocked.
3. Incoming packets destined for host 192.168.21.3 are blocked.
4. All well-known services to the network 192.168.21.0 are allowed.

Types of Firewall

Stateful Inspection Firewall :

- Stateful firewalls (performs Stateful Packet Inspection) can determine the connection state of packet, unlike Packet filtering firewall, which makes it more efficient.
- It keeps track of the state of networks connection travelling across it, such as TCP streams.
- So the filtering decisions would not only be based on defined rules, but also on packet's history in the state table.

Types of Firewall

Application Layer Firewall :

- Application layer firewall can inspect and filter the packets on any OSI layer, up to the application layer.
- It can block specific content, also recognize when certain application and protocols (like HTTP, FTP) are being misused.
- In other words, Application layer firewalls are hosts that run proxy servers.
- A proxy firewall prevents the direct connection between either side of the firewall, each packet has to pass through the proxy.
- It can allow or block the traffic based on predefined rules.

Note: Application layer firewalls can also be used as Network Address Translator(NAT).

Types of Firewall

Next Generation Firewalls (NGFW) :

- Next Generation Firewalls are being deployed these days to stop modern security breaches like advance malware attacks and application-layer attacks.
- NGFW consists of Deep Packet Inspection, Application Inspection, SSL/SSH inspection and many functionalities to protect the network from these modern threats.

Benefits of Firewall

- It provides enhanced security and privacy from vulnerable services. It prevents unauthorized users from accessing a private network that is connected to the internet.
- Firewalls provide faster response time and can handle more traffic loads.
- A firewall allows you to easily handle and update the security protocols from a single authorized device.
- It safeguards network from phishing attacks.

Limitations of Firewall

- Firewalls cannot stop users from accessing malicious websites, making it vulnerable to internal threats or attacks.
- Firewalls cannot protect against the transfer of virus-infected files or software.
- Firewalls cannot prevent misuse of passwords.
- Firewalls cannot protect if security rules are misconfigured.
- Firewalls cannot protect against non-technical security risks, such as social engineering.
- Firewalls cannot stop or prevent attackers with modems from dialing in to or out of the internal network.
- Firewalls cannot secure the system which is already infected.