## UNIT III ACCESS CONTROL AND SECURITY

Network Access Control: Network Access Control, Extensible Authentication Protocol, IEEE 802.1X Port-Based Network Access Control - IP Security - Internet Key Exchange (IKE). Transport-Level Security: Web Security Considerations, Secure Sockets Layer, Transport Layer Security, HTTPS standard, Secure Shell (SSH) application.

### 3.1 Network Access Control: Network Access Control

- Network access control (NAC) is an umbrella term for managing access to a network. NAC authenticates users logging into the network and determines what data they can access and actions they can perform.
- NAC also examines the health of the user's computer or mobile device (the endpoints).

NAC systems deal with **three categories of components**:

**Access requestor (AR):** The AR is the node that is attempting to access the network and may be any device that is managed by the NAC system, including workstations, servers, printers, cameras, and other IP-enabled devices. ARs are also referred to as supplicants, or simply, clients.

**Policy server: T**he policy server determines what access should be granted. The policy server often relies on backend systems, including antivirus, patch management, or a user directory, to help determine the host's condition.

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

Figure is a generic network access diagram.

1. A variety of different ARs seek access to an enterprise network by applying to some type of NAS.
2. The first step is generally to authenticate the AR.

3. Authentication typically involves some sort of secure protocol and the use of cryptographic keys.
4. Authentication may be performed by the NAS, or the NAS may mediate the authentication process.
5. The authentication process verifies a supplicant's claimed identity, which enables the policy server to determine what access privileges, if any, the AR may have.
6. The authentication exchange may result in the establishment of session keys to enable future secure communication between the supplicant and resources on the enterprise network.
7. The policy server or a supporting server will perform checks (health, suitability, screening, or assessment checks) on the AR to determine if it should be permitted interactive remote access connectivity.

8. If the user has acceptable authorization credentials but the remote computer does not pass the health check, the user and remote computer should be denied network access or have limited access to a quarantine.

9. Figure indicates that the quarantine portion of the enterprise network consists of the policy server and related AR suitability servers.

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

11. The NAS may mediate every exchange to enforce a security policy for this AR, or may use other methods to limit the privileges of the AR.

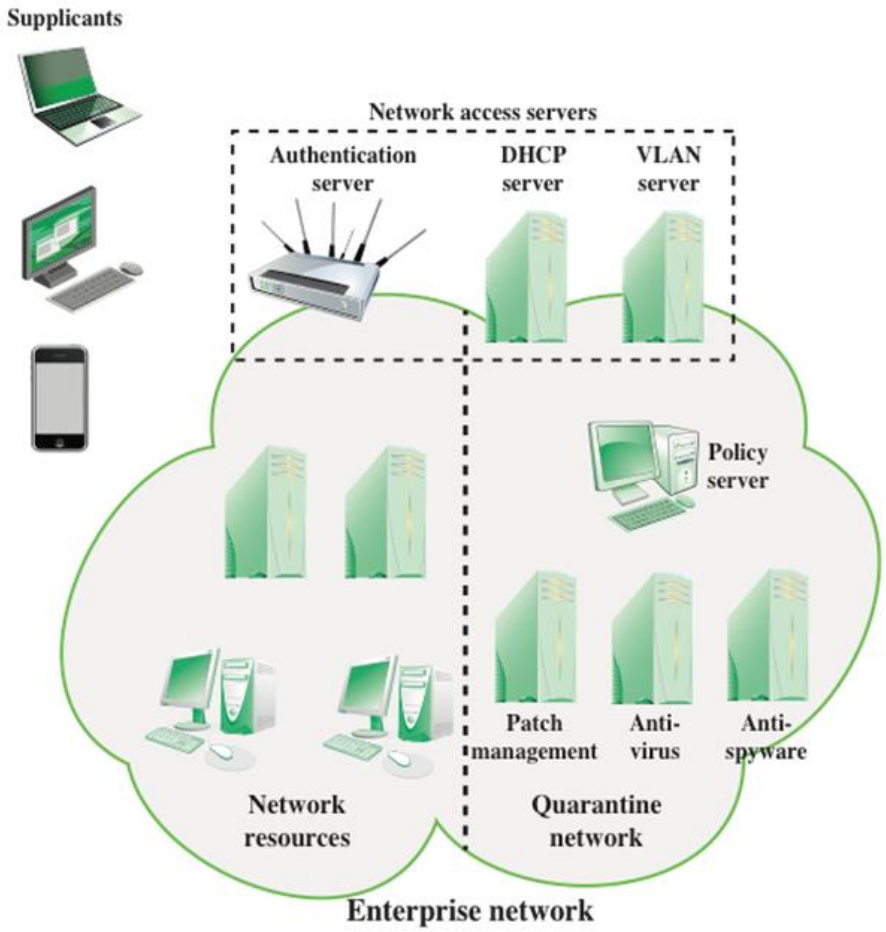



Figure Network Access Control Context

**Network Access Enforcement Methods**

The actions that are applied to ARs to regulate access to the enterprise network

**Common NAC enforcement methods:**

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

**3.2 Extensible Authentication Protocol**

The Extensible Authentication Protocol (EAP), defined in RFC 3748, acts as a **framework for network access and authentication protocols**. EAP provides a generic **transport service for the exchange of authentication information** between a client system and an authentication server.

The basic EAP transport service is extended by using a **specific authentication protocol** that is installed in both the EAP client and the authentication server.

**Commonly supported EAP methods:**

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

## Protocol layers

**EAP-TLS (EAP Transport Layer Security):** EAP-TLS (RFC 5216) defines how the TLS protocol can be encapsulated in EAP messages. EAP-TLS uses the handshake protocol in TLS, not its encryption method. Client and server authenticate each other using digital certificates.

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

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

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

Figure illustrates the protocol layers that form the context for EAP.

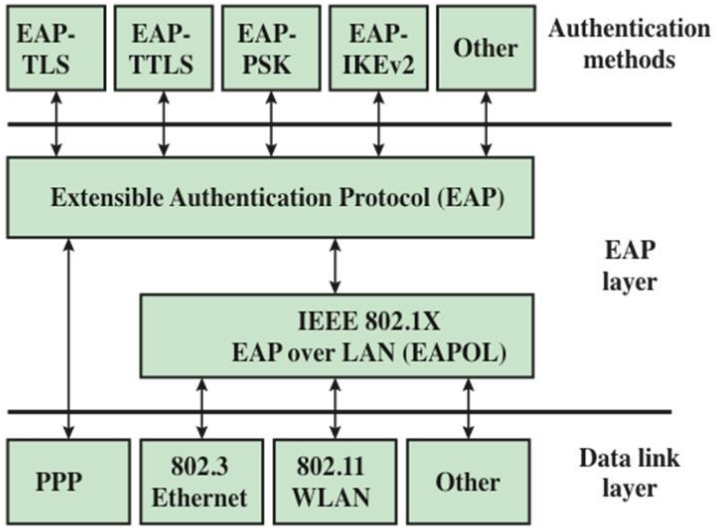



Figure 5.2 EAP Layered Context

## Components

EAP peer: Client computer that is attempting to access a network.

EAP authenticator: An access point or NAS that requires EAP authentication prior to granting access to a network.

Authentication server: A server computer that negotiates the use of a specific EAP method with an EAP peer, validates the EAP peer's credentials, and authorizes access to the network.

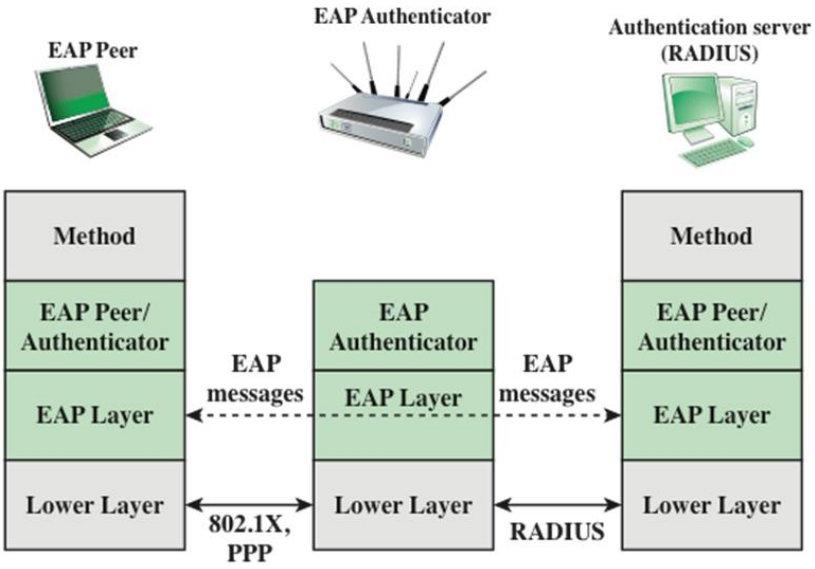



Figure [illegible] EAP Protocol Exchanges

Figure gives an example of an EAP exchange.

- The first pair of EAP Request and Response messages is of Type identity, in which the authenticator requests the peer's identity, and the peer returns its claimed identity in the Response message.
- This Response is passed through the authenticator to the authentication server. Subsequent EAP messages are exchanged between the peer and the authentication server.
- Upon receiving the identity Response message from the peer, the server selects an EAP method and sends the first EAP message with a Type field related to an authentication method.
- If the peer supports and accepts the selected EAP method, it replies with the corresponding Response message of the same type.
- Otherwise, the peer sends a NAK, and the EAP server either selects another EAP method or aborts the EAP execution with a failure message.

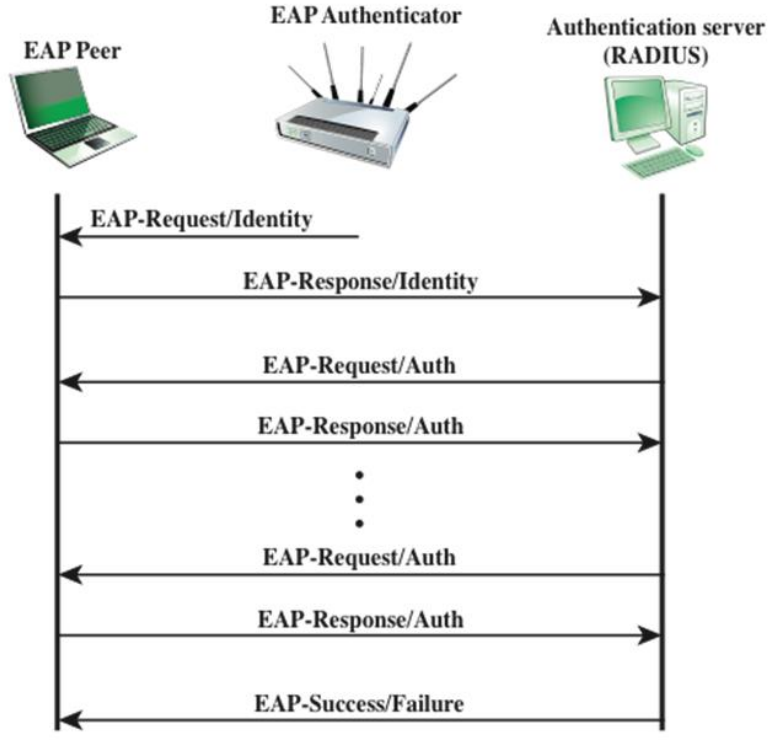



**Figure EAP Message Flow in Pass-Through Mode**

- The selected EAP method determines the number of Request/Response pairs. During the exchange the appropriate authentication information, including key material, is exchanged.
- The exchange ends when the server determines that authentication has succeeded or that no further attempt can be made and authentication has failed.

### 3.3 IEEE 802.1X Port-Based Network Access Control

It is an IEEE Standard for port-based Network Access Control (PNAC). It was designed to provide access control functions for LANs. It provides an authentication mechanism to devices wishing to attach to a LAN or a WLAN.

**Purpose:**

- Port authentication
- Access control

**802.1x Entities**

**Supplicant:**

- Requests to connect to a LAN

**Authenticator:**

- Responsible for initiating the authentication process
- Acting as a relay btwn the authentication server and the supplicant

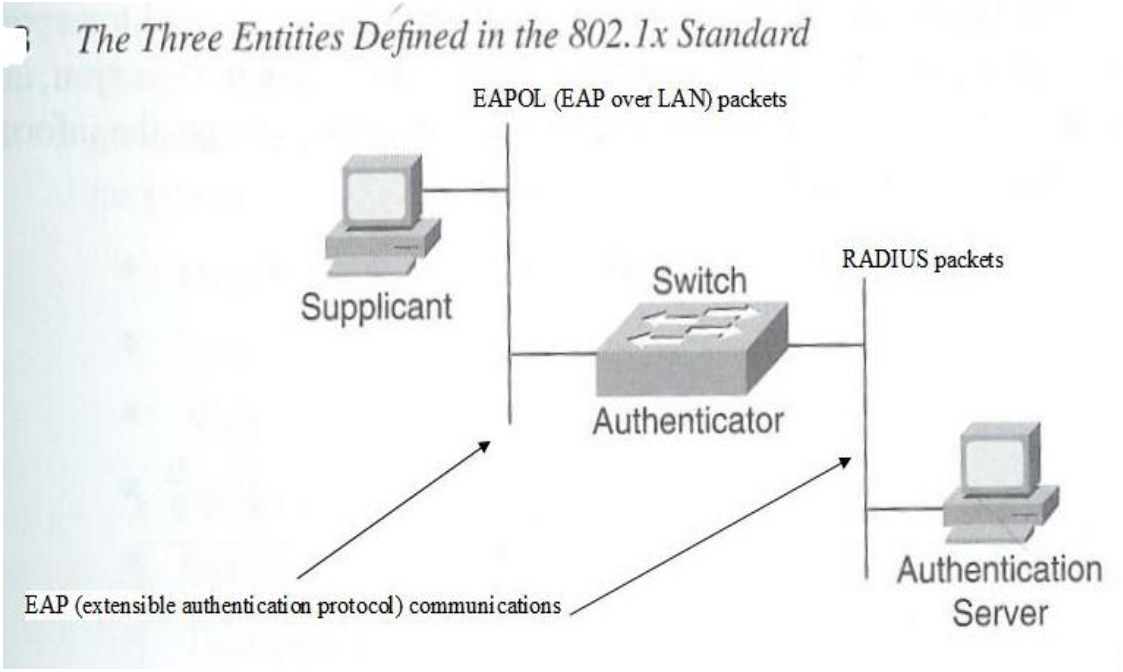
**Authentication server:**

- Responsible for doing the actual authentication & authorization

**Authentication of *supplicants***

- Until the AS authenticates a supplicant, the authenticator only passes control and authentication messages between the supplicant and the AS; the 802.1X control channel is unblocked, but the 802.1X data channel is blocked.

- Once a supplicant is authenticated and keys are provided, the authenticator can forward data from the supplicant. Under these circumstances, the data channel is unblocked.




- The essential element defined in 802.1X is a protocol known as **EAPOL (EAP over LAN)**.
    - EAPOL operates at the network layers and makes use of an IEEE 802 LAN, such as Ethernet or Wi-Fi, at the link level.
    - EAPOL enables a supplicant to communicate with an authenticator and supports the exchange of EAP packets for authentication.

**Controlled and uncontrolled ports**

- 802.1X uses the concepts of controlled and uncontrolled ports.
- Ports are logical entities defined within the authenticator and refer to physical network connections.
- Each logical port is mapped to one of these two types of physical ports.
- An uncontrolled port allows the exchange of protocol data units (PDUs) between the supplicant and the AS, regardless of the authentication state of the supplicant.

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

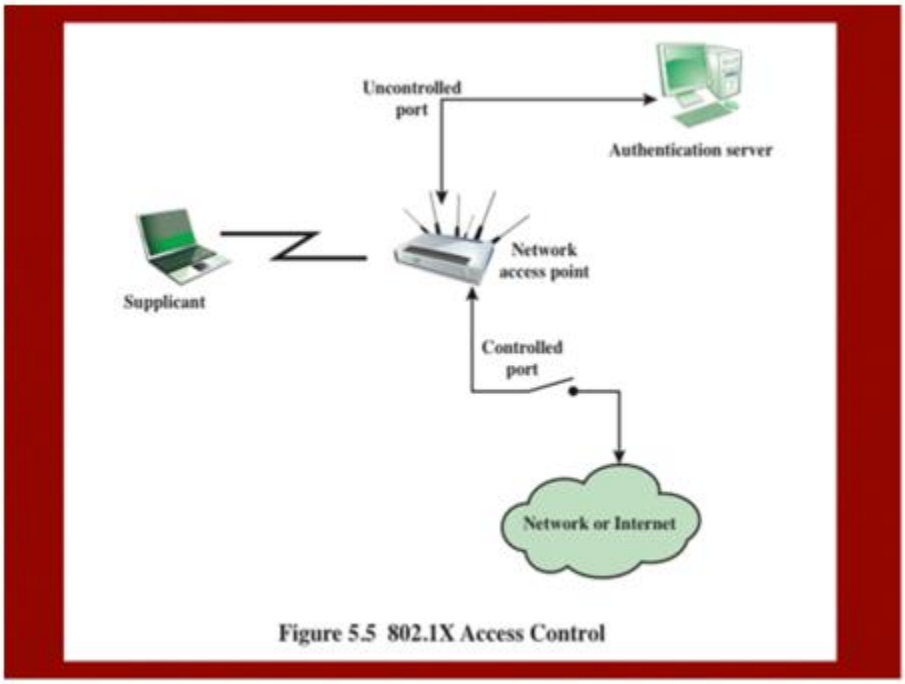



Figure 5.5 802.1X Access Control

**Controlled and uncontrolled access**

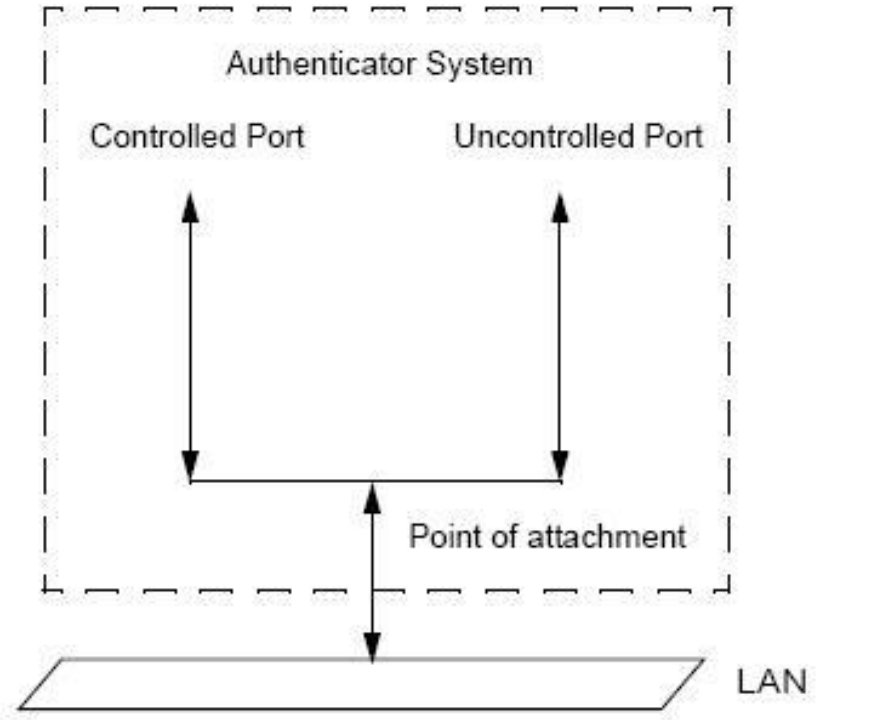



**Figure 6-1—Uncontrolled and controlled Ports**

## Common EAPOL Frame Types

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

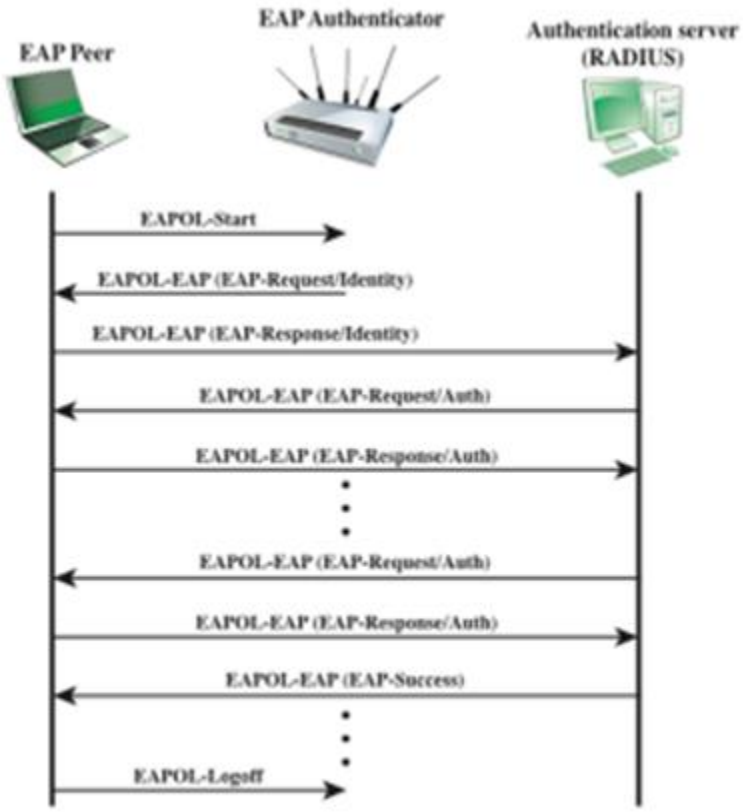



Figure Example Timing Diagram for IEEE 802.1X

## Messages

- EAPOL-Start
- EAPOL-EAP (EAP-Request/ Identity)
- EAPOL-EAP (EAP-Response/ Identity)

- EAPOL-EAP (EAP-Request/ Auth)
- EAPOL-EAP (EAP-Request/ Auth)
- EAPOL-EAP (EAP-Success)
- EAPOL-Logoff

### 3.4 IP Security

IPSec provides the capability to secure communications across a LAN, across private and public WANs, and across the Internet.

Examples are

- Secure branch office connectivity over the Internet
- Secure remote access over the Internet
- Establishing extranet and intranet connectivity with partners
- Enhancing electronic commerce security

**IP Security Architecture**

The IPSec specification consists of numerous documents.

- RFC 2401: An overview of a security architecture
- RFC 2402: Description of a packet authentication extension to IPv4 and IPv6
- RFC 2406: Description of a packet encryption extension to IPv4 and IPv6
- RFC 2408: Specification of key management capabilities

**Architecture**: Covers the general concepts, security requirements, definitions, and mechanisms defining IPSec technology.

**Encapsulating Security Payload (ESP)**: Covers the packet format and general issues related to the use of the ESP for packet encryption and, optionally, authentication.

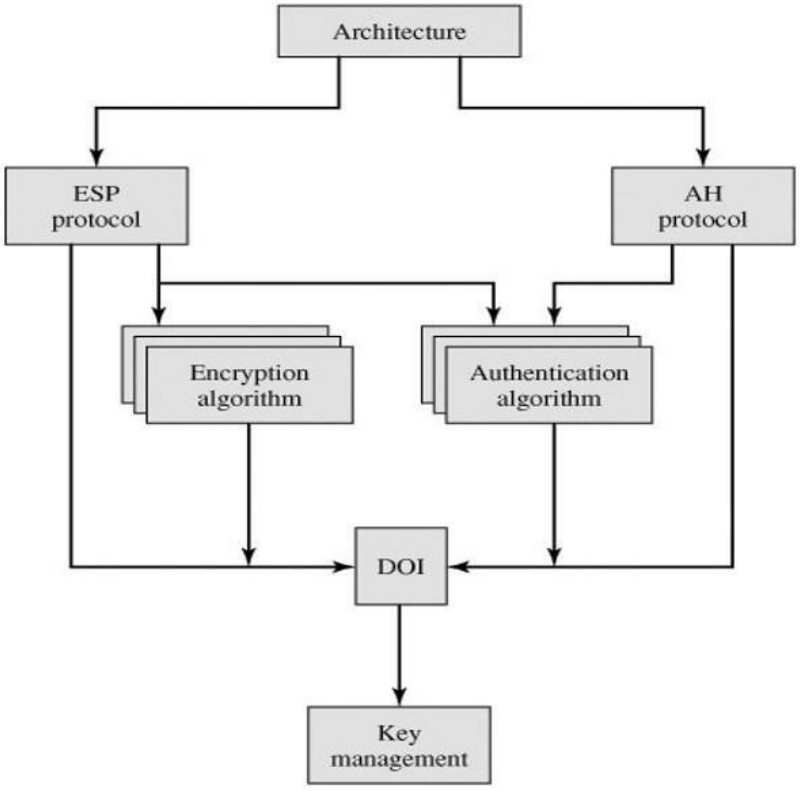
**Authentication Header (AH):** Covers the packet format and general issues related to the use of AH for packet authentication.

**Encryption Algorithm:** A set of documents that describe how various encryption algorithms are used for ESP.

**Authentication Algorithm:** A set of documents that describe how various authentication algorithms are used for AH and for the authentication option of ESP.

**Key Management:** Documents that describe key management schemes.

**Domain of Interpretation (DOI):** This document contains values needed for other documents to relate to each other.




**IPSec Services**

IPSec provides security services at the IP layer by enabling a system to select required security

protocols.

Two protocols are used to provide security:

- Authentication protocol
- Encryption/authentication protocol (ESP).

The services are

- 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**

An association is a one-way relationship between a sender and a receiver that affords security services to the traffic carried on it. If security exchange is needed in both directions then two-way security association is needed. A security association is uniquely identified by three parameters:

**Security Parameters Index (SPI):** A bit string assigned to this SA.

**IP Destination Address:** Only unicast addresses are allowed.

**Security Protocol Identifier:** This indicates whether the association is an AH or ESP security association

**SA Parameters**

A security association is normally defined by the following parameters:

**Sequence Number Counter:** A 32-bit value used to generate the Sequence Number field in AH or ESP headers.

**Sequence Counter Overflow:** A flag indicating whether overflow of the Sequence Number Counter should generate an auditable event and prevent further transmission of packets on this SA.

**Anti-Replay Window:** Used to determine whether an inbound AH or ESP packet is a replay or not.

**AH Information:** Specifies an authentication related parameters like authentication algorithm, authentication key, and key lifetimes.

**ESP Information:** Specifies the encryption and authentication algorithm, keys, initialization values, key lifetimes

**Lifetime of this SA:** This is the time interval after which SA must be replaced with a new SA.

**IPSec Protocol Mode:** This parameter specifies the mode of transfer.

**Path MTU:** Specifies the maximum transmission unit.

## SA Selectors

IPSec provides the user with flexibility in the way in which IPSec services are applied to IP traffic. The means by which IP traffic is related to specific SAs is the nominal Security Policy Database (SPD). SPD contains entries, each of which defines a subset of IP traffic and points to an SA for that traffic. Each SPD entry is defined by a set of IP and upper-layer protocol field values, called selectors. These selectors are used to filter outgoing traffic in order to map it into a particular SA.

The following selectors determine an SPD entry:

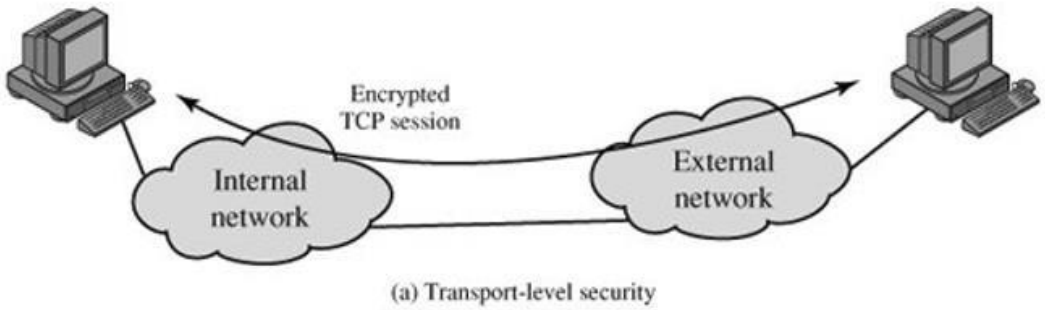
- Destination IP Address
- Source IP Address
- UserID
- Data Sensitivity Level
- Transport Layer Protocol
- Source and Destination Ports

**Modes of Transfer**

- Transport Mode
- Tunnel Mode

**Transport Mode**

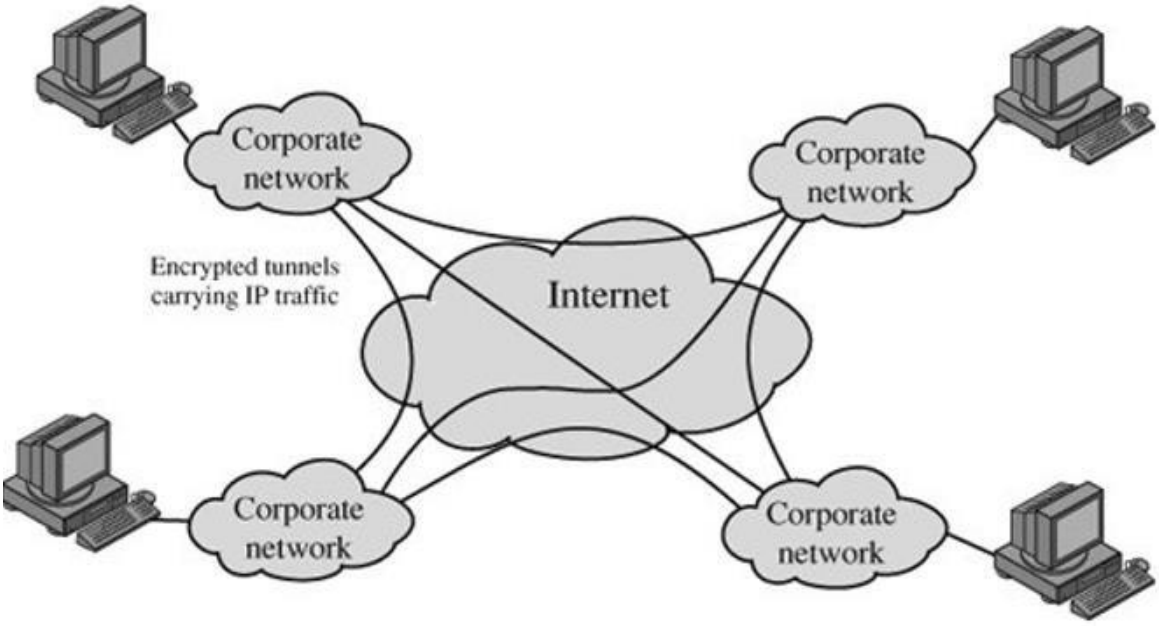
Transport mode provides protection primarily for upper-layer protocols. The transport mode protection extends to the payload of an IP packet. Transport mode is used for end to end connections.




(a) Transport-level security

**Tunnel Mode**

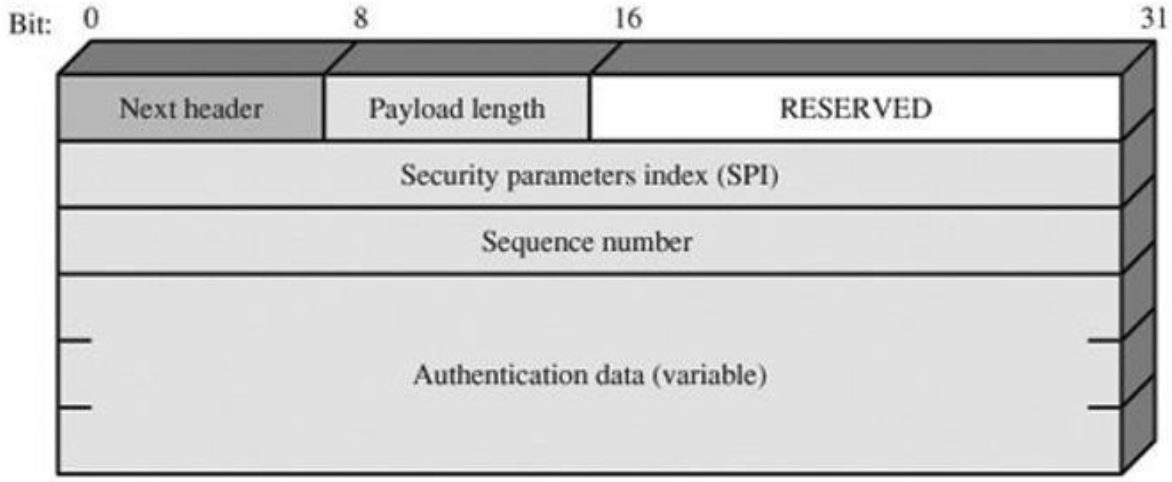
Tunnel mode provides protection to the entire IP packet. Tunnel mode authenticates the entire inner IP and selected portion of outer IP header, IP V6 extension header.




(b) A virtual private network via tunnel mode

**AUTHENTICATION HEADER**

The Authentication Header provides data integrity and authentication of IP packets. The data integrity assures that modification during transit is not possible. The authentication enables the system to authenticate the user and prevents the address spoofing attacks.



The Authentication Header consists of the following fields

**Next Header (8 bits):** Identifies the type of header immediately following this header. Payload Length (8 bits): Length of Authentication Header in 32-bit words, minus 2. Reserved (16 bits): For future use.

**Security Parameters Index (32 bits):** Identifies a security association.

**Sequence Number (32 bits):** A monotonically increasing counter value.

**Authentication Data (variable):** A variable-length which contains the Integrity Check Value.

**Anti-Replay Service**

A replay attack is one in which an attacker obtains a copy of an authenticated packet and later transmits it to the intended destination. The Sequence Number field is designed to overcome such attacks.

**Generation of sequence number by the sender**

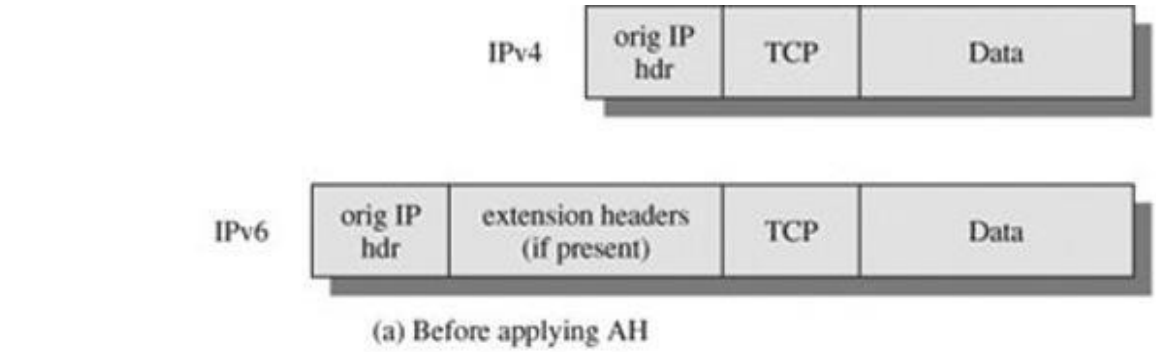
When a new SA is established, the sender initializes a sequence number counter to 0. Each time that a packet is sent on this SA, the sender increments the counter and places the value in the Sequence Number field. Thus, the first value to be used is 1.

For any incoming packet, the processing

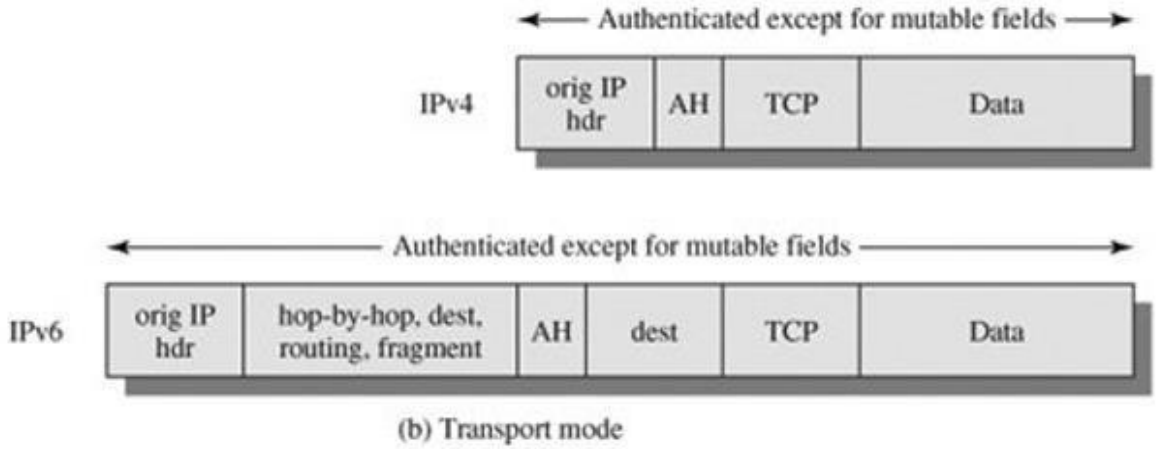
1. If the received packet falls within the window and is new, the MAC is generated.
2. If the packet is authenticated, the corresponding slot in the window is marked as received.
3. If the received packet is to the left of the window, or if authentication fails, the packet is discarded.

**Transport mode AH**

The AH is inserted after the original IP header and before the IP payload.



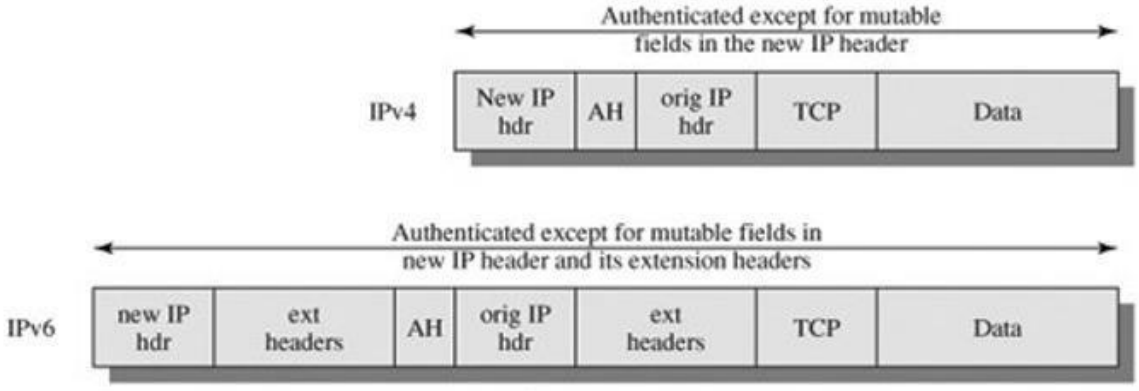

(a) Before applying AH




(b) Transport mode

**Tunnel Mode of AH**

The entire original IP packet is authenticated and the AH is inserted between the orginal IP header and new IP header.




(c) Tunnel mode

## 3.5 INTERNET KEY EXCHANGE:

Definition

- It is a Protocol for doing mutual authentication and establishing a shared secret key to create an IPSec SA.
- Before IPSec sends authenticated or encrypted IP data, both the sender and receiver must agree on the protocols, encryption algorithms and keys to use for message integrity, authentication and encryption. IKE is used to negotiate these and provides primary authentication.

**Uses:**

For long term keys (public signature only keys, pre shared secret keys, public encryption keys)

**Pieces:**

- ISAKMP (Internet Security Association and Key Management Protocol) framework (OAKLEY implementation)
- IKE (Internet Key Exchange) defines fields, chooses options of ISAKMP
- DOI (Domain of Interpretation) specifies particular use of ISAKMP

**ISAKMP:**

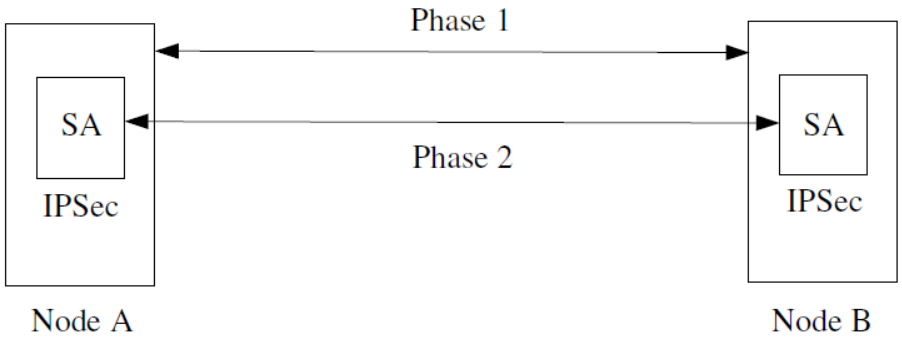
Framework developed by the NSA – mainly concerned with the details of Security Association management

Consists of procedures and fields for

- Authentication of peers
- Negotiation, modification, deletion of Security Associations
- Key generation techniques
- Threat mitigation (e.g. DoS, replay attacks)

An implementation requires a key exchange protocol like IKE. Common implementation is OAKLEY, a key agreement protocol using DH. (Diffie Hellman)

**PHASES OF IKE**




**Phase 1**: Does mutual authentication and establishes session keys based on identities such as names, and secrets

**Phase 2**: SAs are established between two entities

**Goal:**

- Create security association between 2 hosts
- Shared encryption and authentication keys, agreement on crypto algorithms

**Two phases:**

1st phase establishes security association (IKE-SA) for the 2nd phase

- Always by authenticated Diffie-Hellman (expensive)

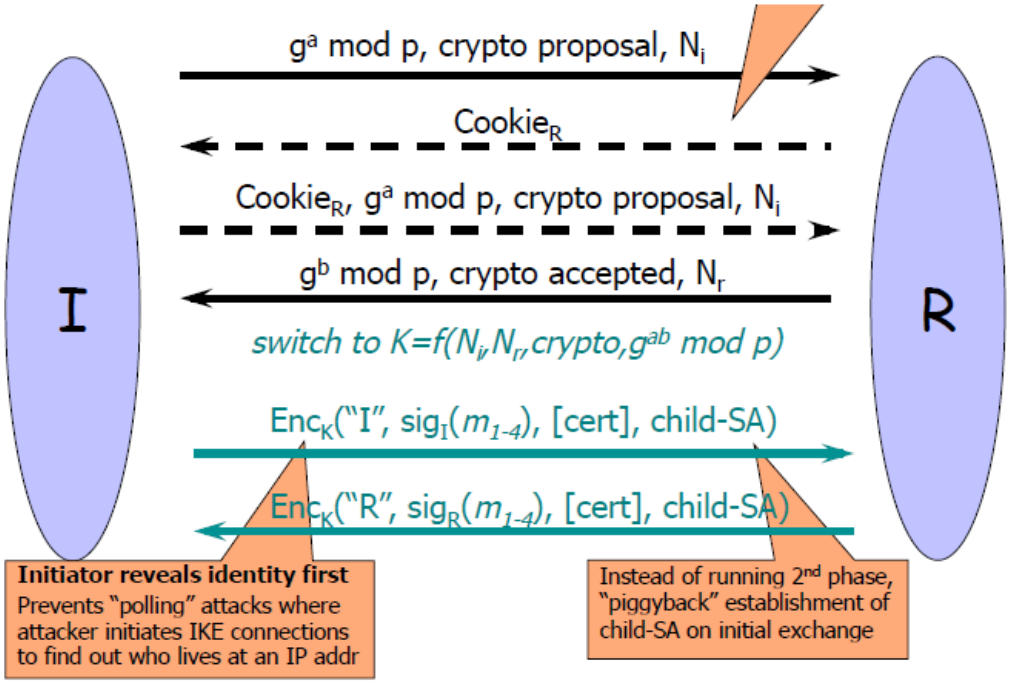
2nd phase uses IKE-SA to create actual security association (child-SA) to be used by AH and ESP

- Use keys derived in the 1st phase to avoid DH exchange
- Can be executed cheaply in “quick” mode

**Need for 2 phases**

- Expensive 1st phase creates “main” SA
- Cheap 2nd phase allows to create multiple child SAs (based on “main” SA) between same 2 hosts

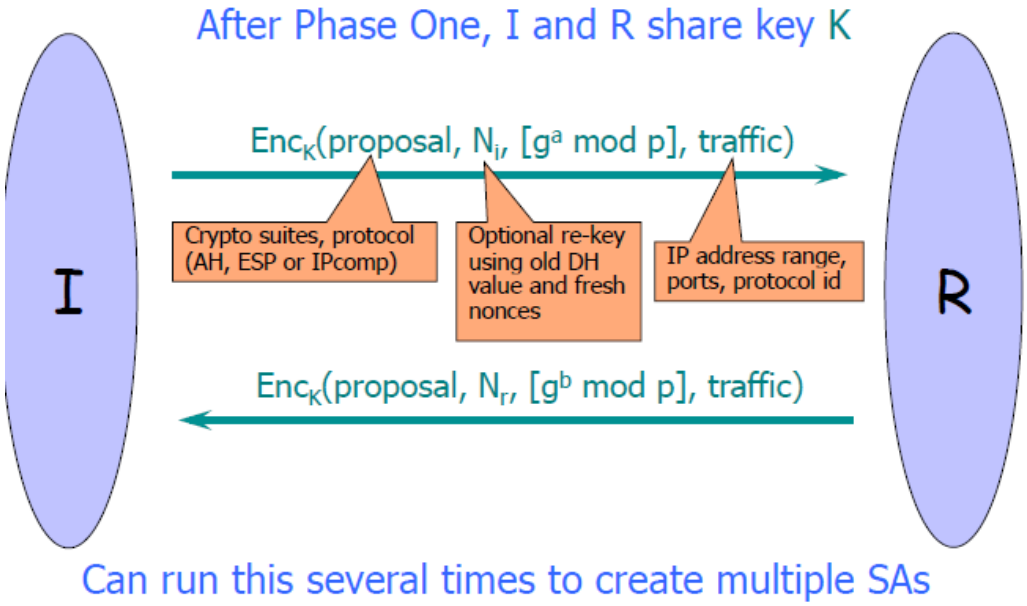
**IKE Phase1**




**Aggressive Mode:** Accomplishes mutual authentication in three messages.

**Main Mode:** Accomplishes mutual authentication in six msgs. Includes ability to hide endpoint identifiers from eavesdroppers and flexibility in negotiating crypto algorithms

**IKE Phase2**



**ISAKMP/IKE Encoding:**

Messages have a fixed header followed by a sequence of payloads. Each payload starts with "type of next payload" and "length of this payload".

## Fixed Header:

| | | |
|---|---|---|
| (64 bits) | initiator's cookie | |
| (64 bits) | responder's cookie | |
| (8 bits) | next payload type | |
| (32 bits) | version | exchange type |
| (80 bits) | flags | message ID |
| (64 bits) | message length | |

**Payload type:**

End, SA, Proposal, Transform (crypto choices), Key Exchange, ID, Certificate, Certificate Request, Checksum (hash), signature, nonce, Notification, delete (closing the SPI), vendor ID (for telling the Implementation being used)

**flags:** encrypted, commit, authentication only (set only during phase 2),

**messageID:** differentiates messages with same phase 1 SA

**3.6 Transport-Level Security- Web Security Considerations**

The World Wide Web is a client/server application running over the Internet and TCP/IP intranets.

- The internet is two-way; - the Web is vulnerable to attacks on the Web servers over the Internet
- Reputations can be damaged and money can be lost if the Web servers are subverted.
- Web servers are relatively easy to configure and manage, the underlying software is extraordinarily complex. This complex software may hide many potential security flaws.

- Casual and untrained (in security matters) users are common clients for Web-based services. Such users are not necessarily aware of the security risks

**WEB SECURITY THREATS**

One way to group these threats is in terms of passive and active attacks.

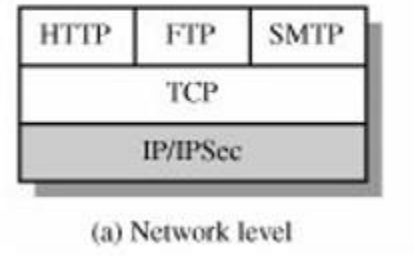
- Passive attacks include eavesdropping on network traffic between browser and server and gaining access to information on a Web site that is supposed to be restricted.
- Active attacks include impersonating another user, altering messages in transit between client and server, and altering information on a Web site.

Another way to classify Web security threats is in terms of the **location of the threat**: Web server, Web browser, and network traffic between browser and server. The various types of security threats faced when using the Web are listed below.

- Integrity – modification of user data
    - Modification of user data
    - Trojan horse browser
- Confidentiality – eavesdropping on the Net
    - Eavesdropping on the net
    - Theft of info from server
    - Theft of data from client
- Denial of Service – preventing any part of system from functioning
    - Killing of user threads
    - Flooding machine with bogus requests
- Authentication – impersonation of legitimate users
    - Impersonation of legitimate users
    - Data forgery

**Web Traffic Security Approaches**

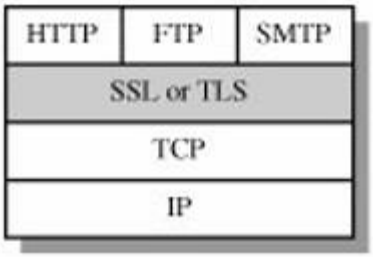
**Approach1**



(a) Network level

This approach is transparent to end users and provides a general purpose solution. IPSec data filtering, so that only selected traffic can incur IPSec processing.

**Approach 2**

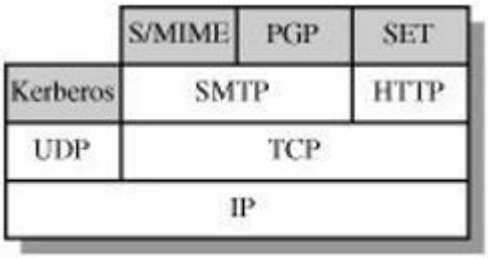


(b) Transport level

Implement the security above TCP. SSL can be embedded with SSL to Netscape Navigator, Microsoft explorer.

**Approach 3**

Security services are embedded in a particular application. SET is an example of this approach.



(c) Application level

Web Security Requirements

Web security can be done through several ways

1.IPSEC

2.SSL

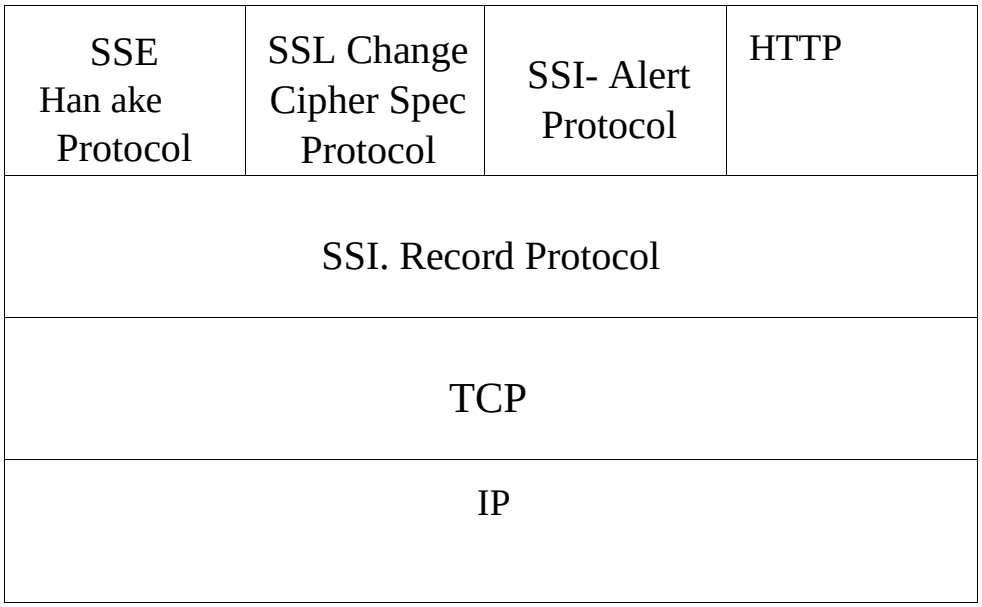
3.SET

**3.7 Secure Sockets Layer (SSL)**

- One of the most widely used security services is the Secure Sockets Layer (SSL) and the follow-on Internet standard known as Transport Layer Security (TLS).
- SSL was originated by Netscape.

**SSL Architecture**

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




The SSL Record Protocol provides basic security services to various higher-layer protocols. Three higher layer protocols are part of SSL viz

- Handshake Protocol,
- Change Cipher Spec Protocol,
- Alert Protocol

*SSI. concepts*

Two important SSI. concepts are

- SSI. session
- SSI. connection

***Parameters of a session state***

**Session identifier:** An arbitrary byte sequence chosen by the server to identify an active or resumable session state.

**Peer certificate:** An X509.v3 certificate of the peer.

**Compression method:** The algorithm used to compress data prior to encryption.

**Cipher spec:** defines cryptographic attributes such as the hash size.

Master secret: 48-byte secret shared between the client and server.

**Is resumable:** A flag indicating whether the session can be used to initiate new connections

***Parameters that define the connection state***

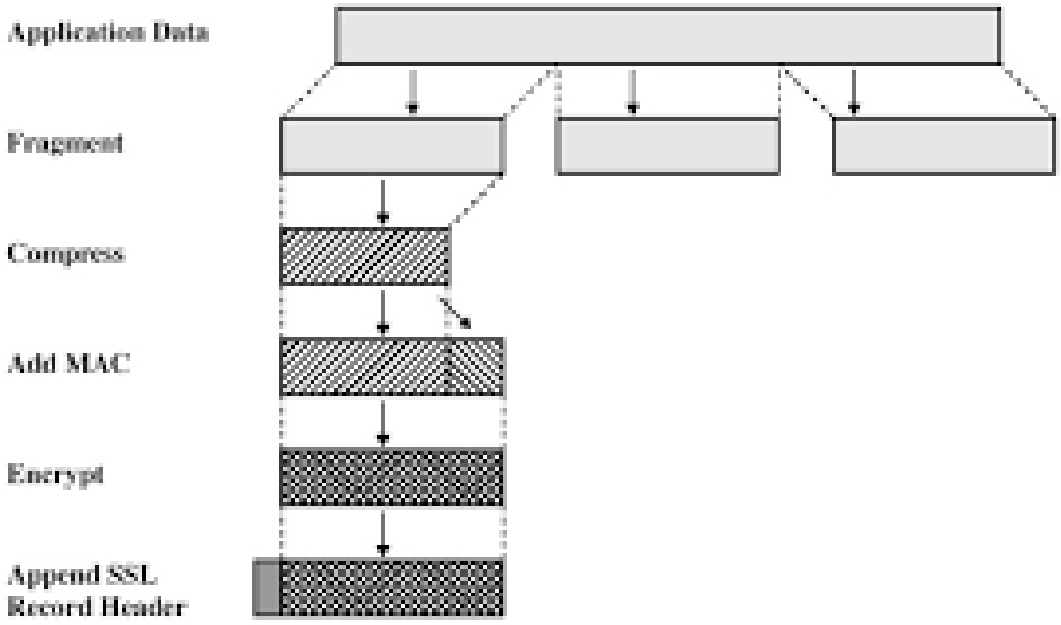
- Server and client random
- Server write MAC secret: . Client write MAC secret
- Server write key
- Client write key
- Initialization vectors
- Sequence numbers

## SSL protocol stack

The SSL Record Protocol provides two services for SSL connections:

Confidentiality: The Handshake Protocol defines a shared secret key that is used for conventional encryption of SSL payloads.

Integrity: The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC). SSI. record protocol operations




The Record Protocol takes an application message to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, adds a header, and transmits the resulting unit in a TCP segment. Received data are decrypted, verified, decompressed, and reassembled and then delivered to higher-level users.

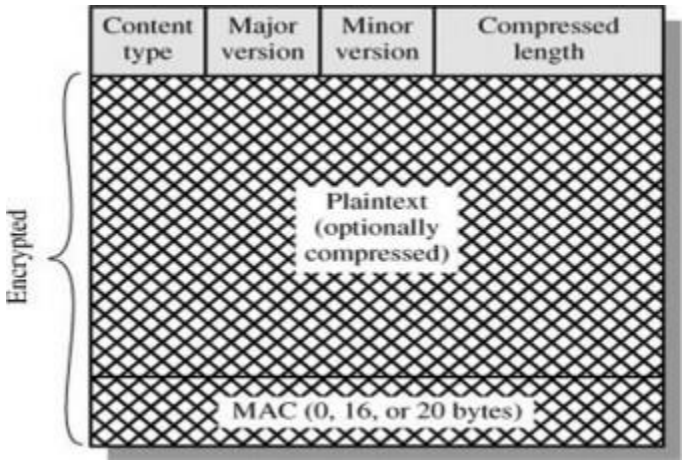
## SSI. record format

Content Type (8 bits): The higher layer protocol used to process the enclosed fragment.

Major Version (8 bits): Indicates major version of SSL in use which is 3.

Minor Version (8 bits): Indicates minor version in use which is 0.

Compressed Length (16 bits): The length in bytes of the plaintext fragment, maximum value is 214 4 2048.




***Change Cipher Spec Protocol***

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

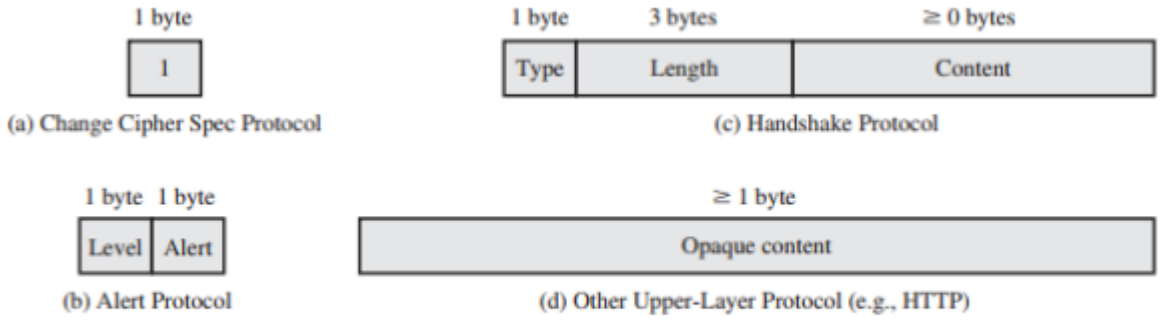



*Figure SSL record Protocol Payload.*

***Alert Protocol***

The Alert Protocol is used to convey SSL-related alerts to the peer entity. The SSL messages are compressed and encrypted, as specified by the current state. The alerts are

unexpected_message: An inappropriate message was received. bad record mac: An incorrect MAC was received.

decompression_failure: The decompression function received improper input handshake failure: Sender was unable to negotiate an acceptable set of security parameters given the options available.

***Handshake Protocol***

The Handshake Protocol consists of a series of messages exchanged by client and server

**Phase 1: Establish Security Capabilities**

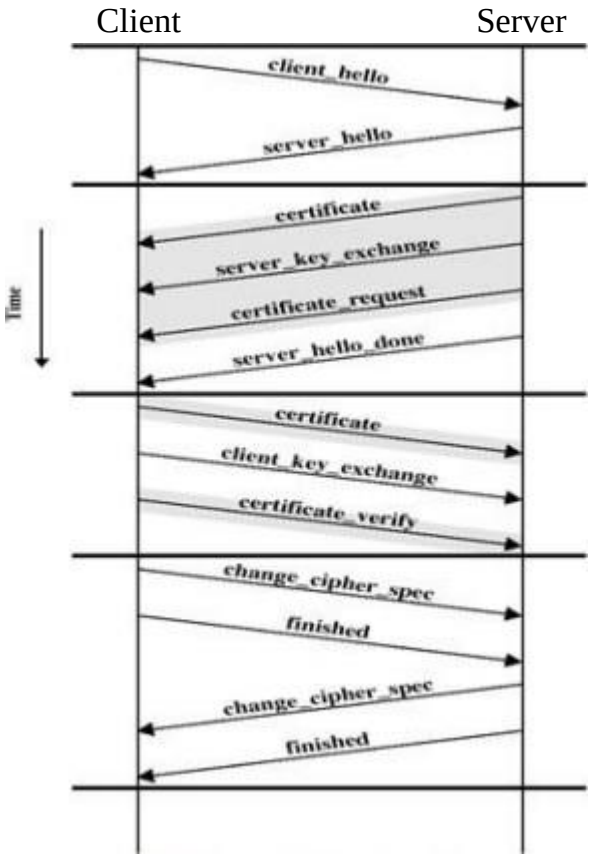
This phase is used to initiate a logical connection and to establish the security capabilities that will be associated with it. The exchange is initiated by the client by sending the 'client hello' message. It contains the

- Version
- Random
- Session id
- Cipher Suite
- Compression Method

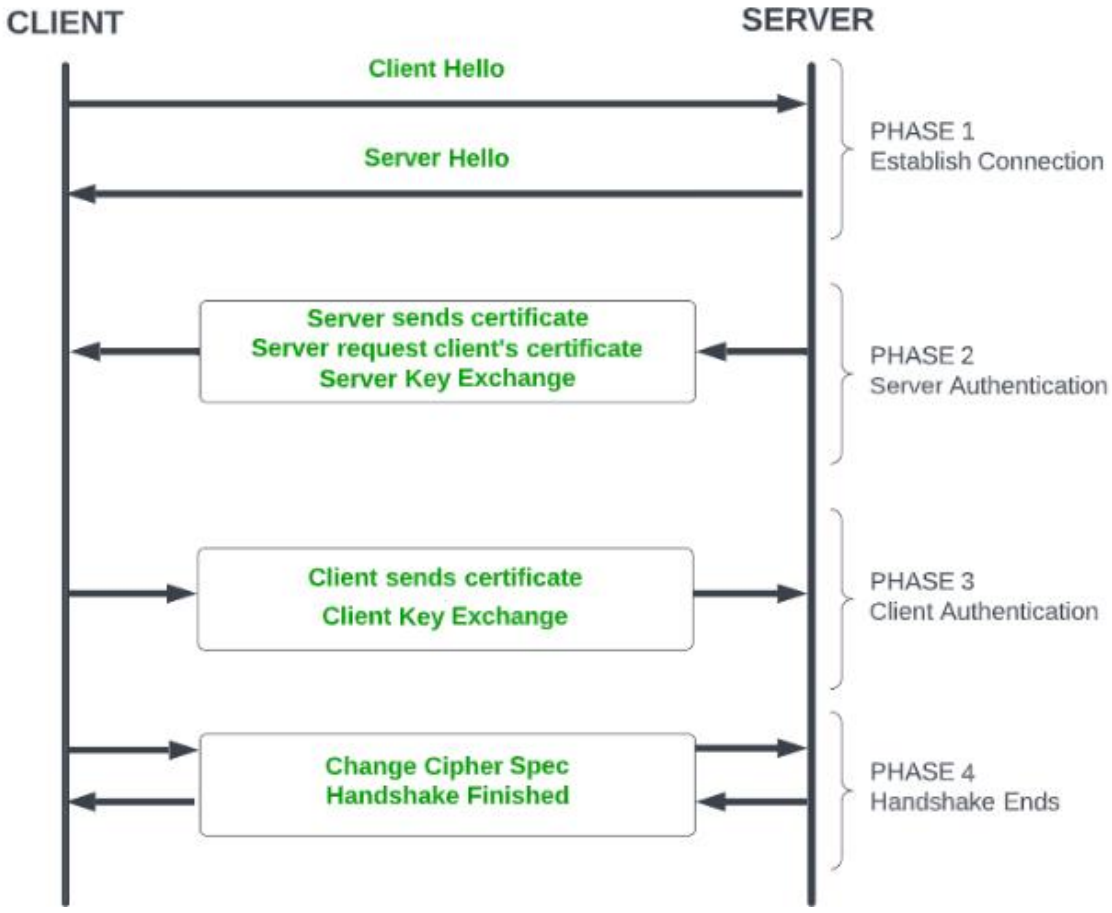
The server also sends a 'server hello' message to the client.

***Phase 2: Server Authentication and Key Exchange***

- The server sends its certificate, if it needs to be authenticated.
- Server key exchange message is optional
- Server request a certificate from the client
- 'Server done' is sent by the server to indicate the end of server hello and associated messages. After this message the server will not wait for the client response.




**SSL Handshake Protocol**



***Phase 3. Client Authentication and Key Exchange***

The client should verify that the server provided a valid. If all are satisfied the client sends one or more messages back to the server.

***Phase 4. Finish***

The client sends a change cipher spec message and copies the pending CipherSpec into the current CipherSpec. The client then immediately sends the finished message under the new algorithms, keys, and secrets. The finished message verifies that the key exchange and authentication processes were successful.

In response to these two messages, the server sends its own change cipher_spec message, transfers the pending to the current CipherSpec, and sends its finished message. Now, the handshake is complete and the client and server exchange data.

## 3.8 Transport Layer Security

TLS is an IETF standardization initiative whose goal is to produce an Internet standard version of SSL. TLS is defined as a Proposed Internet Standard in RFC 5246. RFC 5246 is very similar to SSLv3 the differences are listed below:

**➢ Version Number**

The TLS Record Format is the same as that of the SSL Record Format and the fields in the header have the same meanings. The one difference is in version values. For the current version of TLS, the major version is 3 and the minor version is 3.

**➢ Message Authentication Code**

There are two differences between the SSLv3 and TLS MAC schemes: the actual algorithm and the scope of the MAC calculation. TLS makes use of the HMAC algorithm defined in RFC 2104.

The MAC calculation covers all of the fields covered by the SSLv3 calculation, plus the field TLS Compressed Version, which is the version of the protocol being employed.

➢ **Pseudorandom Function**

TLS makes use of a pseudorandom function referred to as PRF to expand secrets into blocks of data for purposes of key generation or validation. The objective is to make use of a relatively small shared secret value but to generate longer blocks of data in a way that is secure from the kinds of attacks made on hash functions and MACs.

To make PRF as secure as possible, it uses two hash algorithms in a way that should guarantee its security if either algorithm remains secure.

➢ **Alert Codes**

TLS supports all of the alert codes defined in SSLv3 with the exception of no_certificate. A number of additional codes are defined in TLS; of these, the following are always fatal.

• record_overflow: A TLS record was received with a payload (ciphertext) whose length exceeds bytes, or the ciphertext decrypted to a length of greater than bytes.

• unknown_ca: A valid certificate chain or partial chain was received, but the certificate was not accepted because the CA certificate could not be located or could not be matched with a known, trusted CA.

• access_denied: A valid certificate was received, but when access control was applied, the sender decided not to proceed with the negotiation.

• decode_error: A message could not be decoded, because either a field was out of its specified range or the length of the message was incorrect.

• protocol_version: The protocol version the client attempted to negotiate is recognized but not supported.

The remaining alerts include the following.

• user_canceled: This handshake is being canceled for some reason unrelated to a protocol failure.

• no_renegotiation: Sent by a client in response to a hello request or by the server in response to a client hello after initial handshaking. Either of these messages would

normally result in renegotiation, but this alert indicates that the sender is not able to renegotiate. This message is always a warning.

### ➢ Cipher Suites

There are several small differences between the cipher suites available under SSLv3 and under TLS:

• Key Exchange: TLS supports all of the key exchange techniques of SSLv3 with the exception of Fortezza.

• Symmetric Encryption Algorithms: TLS includes all of the symmetric encryption algorithms found in SSLv3, with the exception of Fortezza.

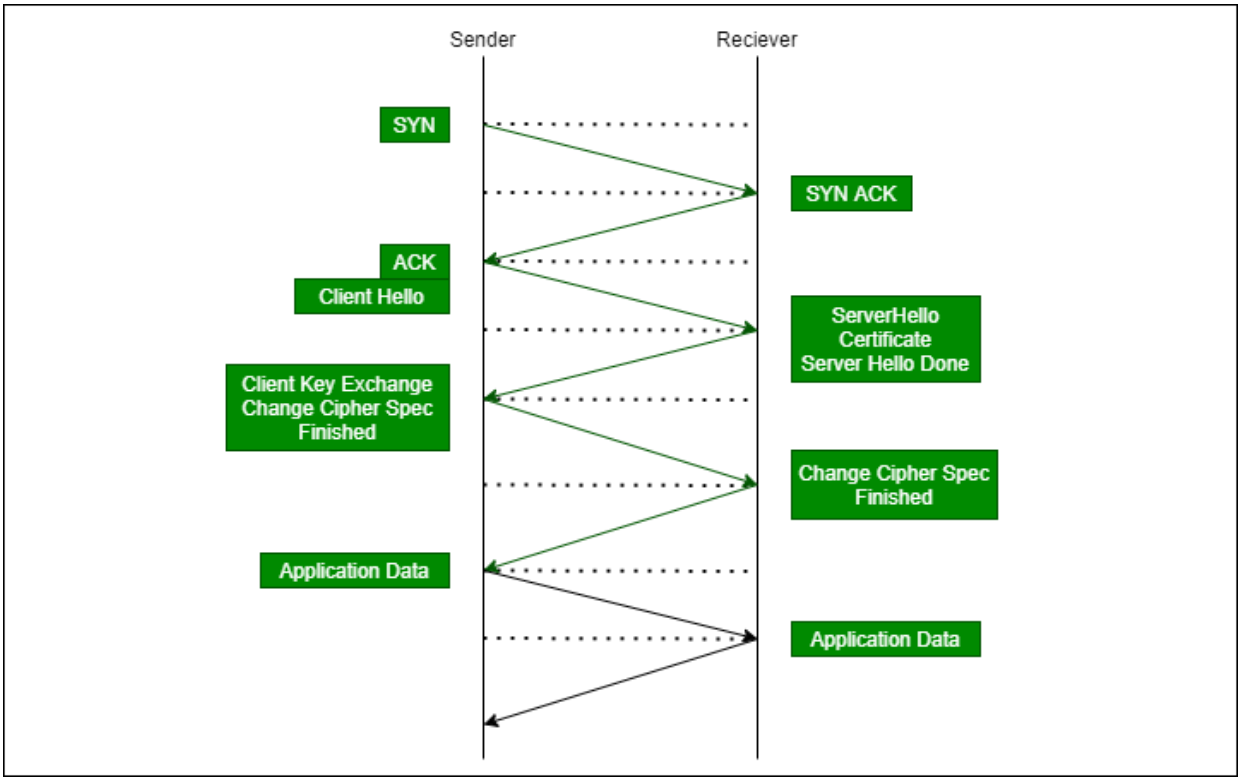
### ➢ certificate_verify and Finished Messages

In the TLS certificate_verify message, the MD5 and SHA-1 hashes are calculated only over handshake_messages. As with the finished message in SSLv3, the finished message in TLS is a hash based on the shared master_secret, the previous handshake messages, and a label that identifies client or server.

### ➢ Padding

In SSL, the padding added prior to encryption of user data is the minimum amount required so that the total size of the data to be encrypted is a multiple of the cipher's block length. In TLS, the padding can be any amount that results in a total that is a multiple of the cipher's block length, up to a maximum of 255 bytes.

**TLS HANDSHAKE**




### 3.9 HTTPS

➢ (HTTP over SSL) refers to the combination of HTTP and SSL to implement secure communication between a Web browser and a Web server. The HTTPS capability is built into all modern Web browsers. Its use depends on the Web server supporting HTTPS communication.

➢ The principal difference seen by a user of a Web browser is that URL (uniform resource locator) addresses begin with https:// rather than http://.

➢ A normal HTTP connection uses port 80. If HTTPS is specified, port 443 is used, which invokes SSL. When HTTPS is used, the following elements of the communication are encrypted:

- ✓ URL of the requested document
- ✓ Contents of the document
- ✓ Contents of browser forms (filled in by browser user)
- ✓ Cookies sent from browser to server and from server to browser

✓ Contents of HTTP header

HTTPS is documented in RFC 2818, HTTP Over TLS. There is no fundamental change in using HTTP over either SSL or TLS, and both implementations are referred to as HTTPS.

**Connection Initiation**

• The client initiates a connection to the server on the appropriate port and then sends the TLS ClientHello to begin the TLS handshake. When the TLS handshake has finished, the client may then initiate the first HTTP request. All HTTP data is to be sent as TLS application data. Normal HTTP behavior, including retained connections, should be followed.

• At the HTTP level, an HTTP client requests a connection to an HTTP server by sending a connection request to the next lowest layer.

• At the level of TLS, a session is established between a TLS client and a TLS server. This session can support one or more connections at any time.

**Connection Closure**

• An HTTP client or server can indicate the closing of a connection by including the following line in an HTTP record: Connection: close.

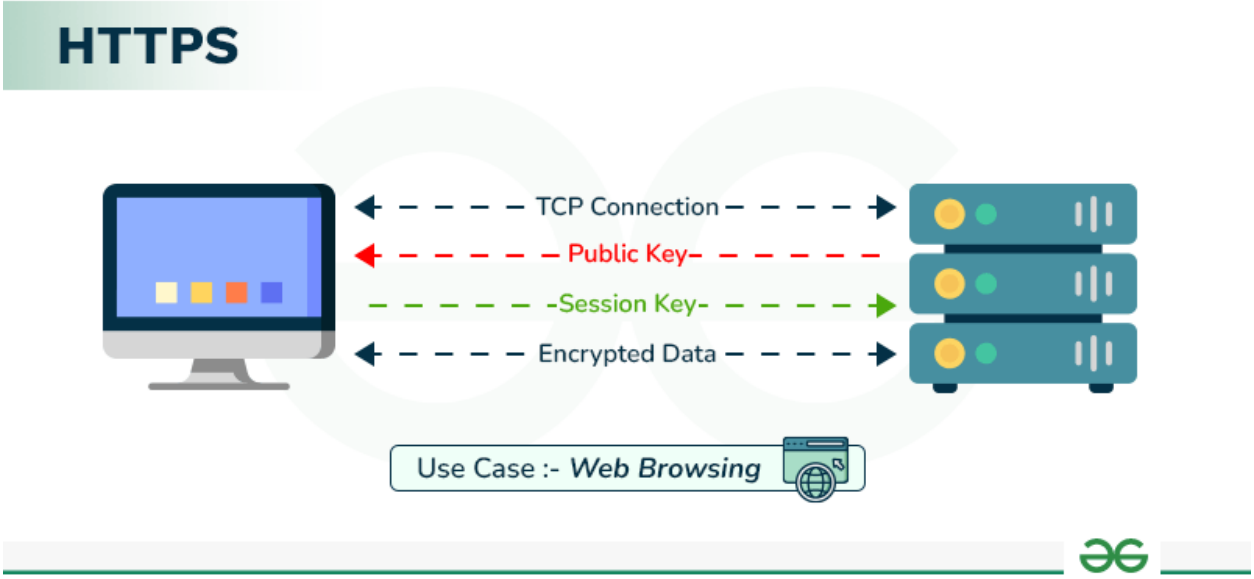
• This indicates that the connection will be closed after this record is delivered. The closure of an HTTPS connection requires that TLS close the connection with the peer TLS entity on the remote side, which will involve closing the underlying TCP connection.

• At the TLS level, the proper way to close a connection is for each side to use the TLS alert protocol to send a close_notify alert. TLS implementations must initiate an exchange of closure alerts before closing a connection.

• A TLS implementation may, after sending a closure alert, close the connection without waiting for the peer to send its closure alert, generating an “incomplete close”.

HTTP clients also must be able to cope with a situation in which the underlying TCP

connection is terminated without a prior close_notify alert and without a Connection: close indicator. The unannounced TCP closure could be evidence of some sort of attack. So the HTTPS client should issue some sort of security warning when this occurs.




**3.10 Secure Shell (SSH) application.**

Secure Shell (SSH) is a protocol for secure network communications designed to be relatively simple and inexpensive to implement.

• The initial version, SSH1 was focused on providing a secure remote logon facility to replace TELNET and other remote logon schemes that provided no security.

• A new version, SSH2, fixes a number of security flaws in the original scheme.

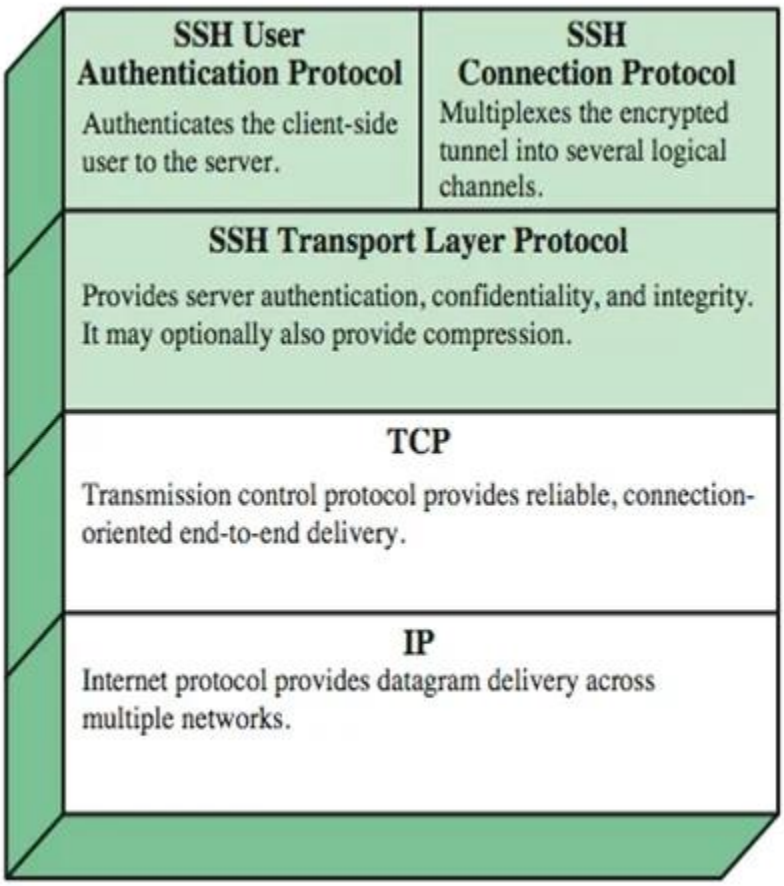
• SSH2 is documented as a proposed standard in IETF RFCs 4250 through 4256.

• SSH client and server applications are widely available for most operating systems. It has become the method of choice for remote login and is rapidly becoming one of the most pervasive applications for encryption technology outside of embedded systems.

• SSH is organized as three protocols that typically run on top of TCP

Transport Layer Protocol: Provides server authentication, data confidentiality, and

data integrity with forward secrecy (i.e., if a key is compromised during one session, the knowledge does not affect the security of earlier sessions). The transport layer may optionally provide compression.




• User Authentication Protocol: Authenticates the user to the server.

• Connection Protocol: Multiplexes multiple logical communications channels over a single, underlying SSH connection.

**TRANSPORT LAYER PROTOCOL**

❖ HOST KEYS:

The server host key is used during key exchange to authenticate the identity of the host. Server authentication occurs at the transport layer, based on the server possessing a public/private key pair.

## ❖ PACKET EXCHANGE

Each packet is in the following format

• Packet length: Length of the packet in bytes, not including the packet length and MAC fields.

• Padding length • Payload • Random padding

• Message authentication code (MAC

• Once an encryption algorithm has been negotiated, the entire packet (excluding the MAC field) is encrypted after the MAC value is calculated.

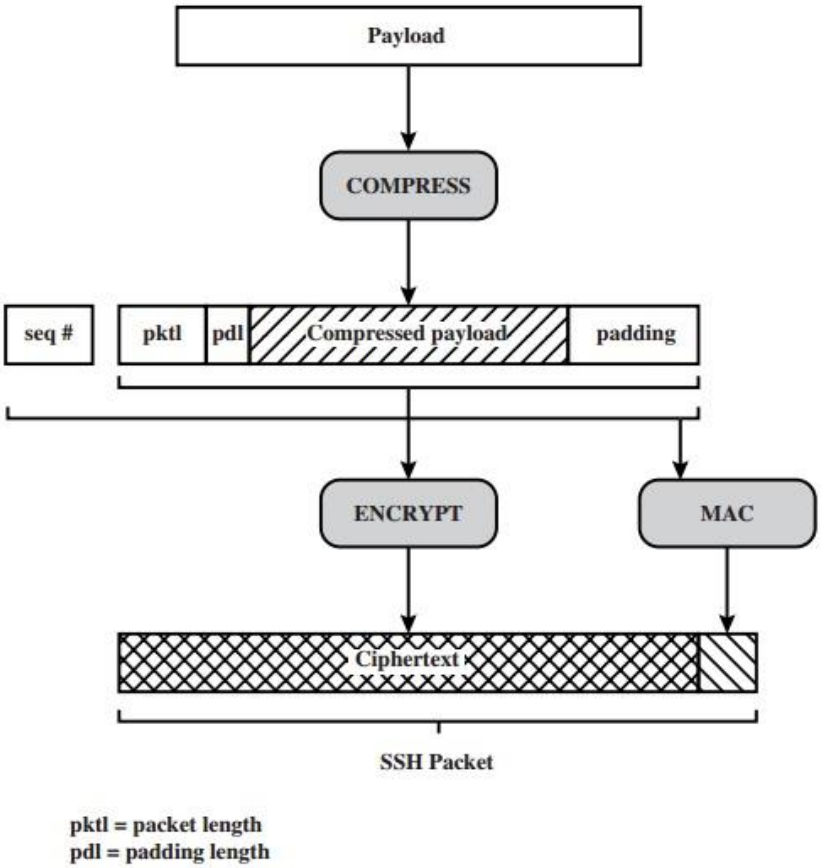



Figure 16.10 SSH Transport Layer Protocol Packet Formation

The SSH Transport Layer packet exchange consists of a sequence of steps

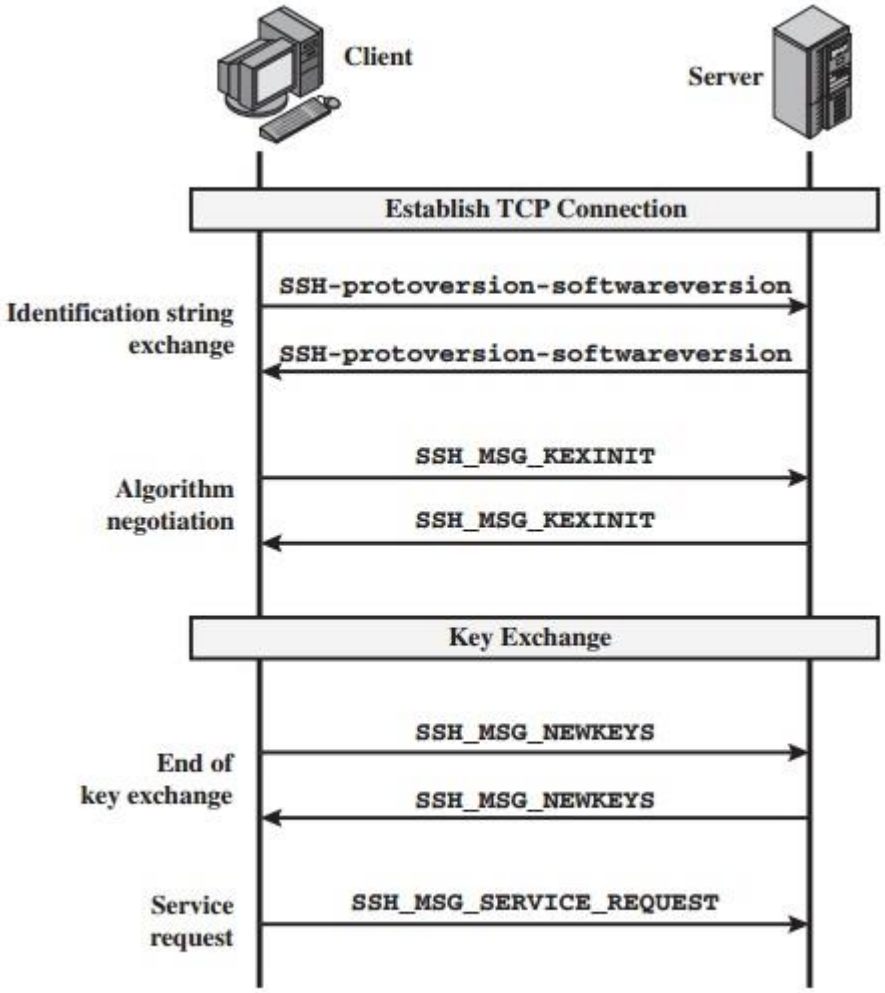



Figure 16.9 SSH Transport Layer Protocol Packet Exchanges

First, the client establishes a TCP connection to the server. This is done via the TCP protocol and is not part of the Transport Layer Protocol.

• Once the connection is established, the client and server exchange data, referred to as packets, in the data field of a TCP segment.

**User Authentication Protocol**

The User Authentication Protocol provides the means by which the client is authenticated to the server.

**CONNECTION PROTOCOL**

The SSH Connection Protocol runs on top of the SSH Transport Layer Protocol and assumes that a secure authentication connection is in use. The secure authentication connection, referred to as a tunnel, is used by the Connection Protocol to multiplex a number of logical channels.

## CHANNEL MECHANISM

• All types of communication using SSH, such as a terminal session, are supported using separate channels. Either side may open a channel.

• For each channel, each side associates a unique channel number, which need not be the same on both ends.

• Channels are flow controlled using a window mechanism. No data may be sent to a channel until a message is received to indicate that window space is available.

• The life of a channel progresses through three stages: opening a channel, data transfer, and closing a channel.

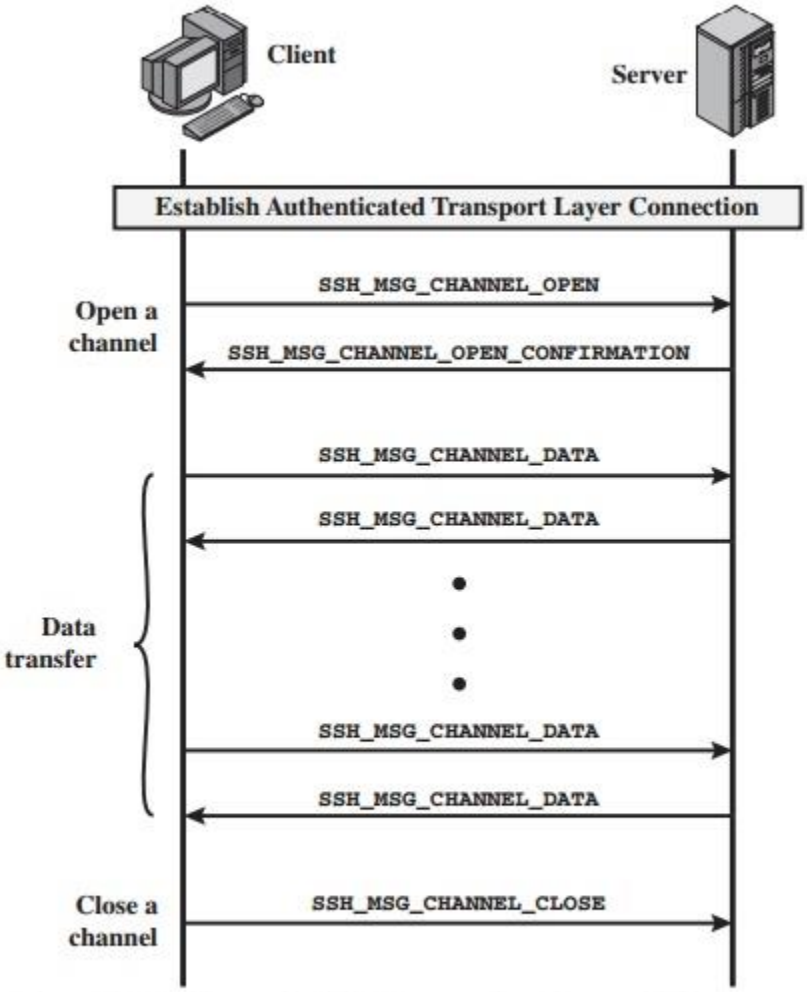



Figure 16.11 Example SSH Connection Protocol Message Exchange

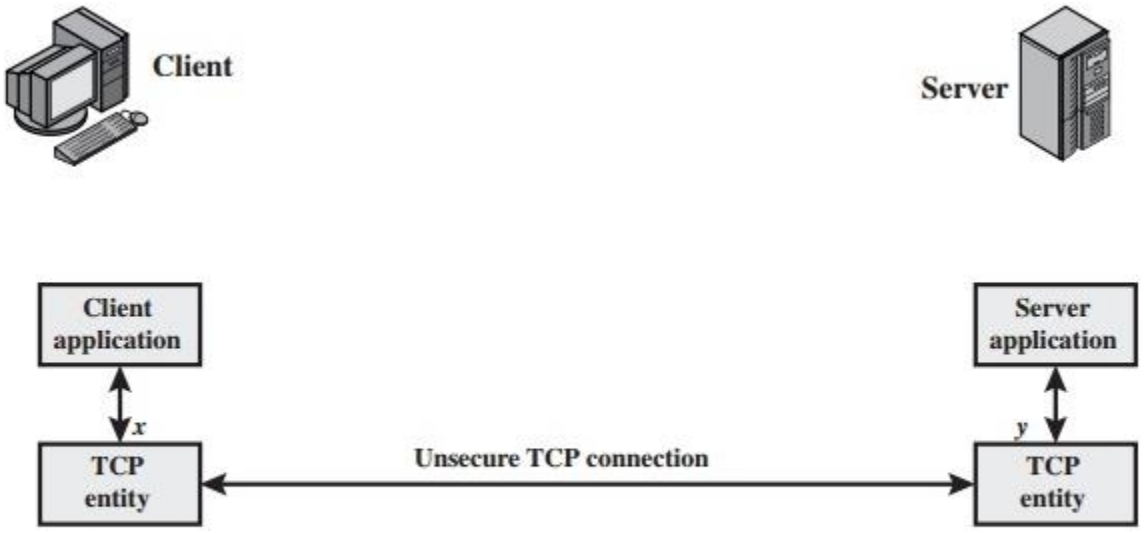
**PORT FORWARDING**

One of the most useful features of SSH is port forwarding. Port forwarding provides the ability to convert any insecure TCP connection into a secure SSH connection. This is also referred to as SSH tunnelling.

SSH supports two types of port forwarding: local forwarding and remote forwarding.

**Local forwarding**

It allows the client to set up a "hijacker" process. This will intercept selected application-level traffic and redirect it from an unsecured TCP connection to a secure SSH tunnel.

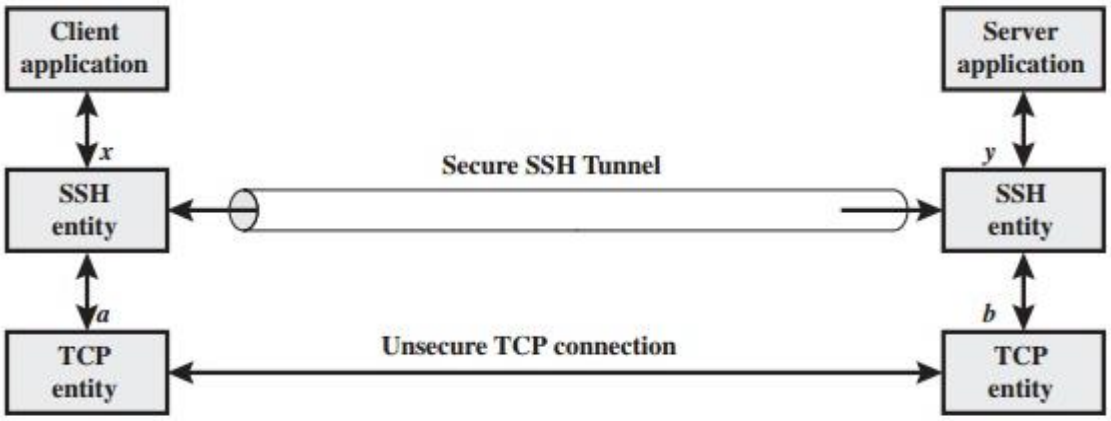



(a) Connection via TCP

**Remote forwarding**,

The user's SSH client acts on the server's behalf.

• The client receives traffic with a given destination port number, places the traffic on the correct port and sends it to the destination the user chooses.




(b) Connection via SSH tunnel