***1. Explain the two specifications of TLS. (4.5M)***

Ans: Two important TLS concepts are the TLS session and the TLS connection, which are defined in the specification as follows: **(0.5M)**

Connection: A connection is a transport (in the OSI layering model definition) that provides a suitable type of service. For TLS, such connections are peer-to-peer relationships. The connections are transient. Every connection is associated with one session. **(2M)**

Session: A TLS session is an association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection. **(2M)**

**Schema: Each state: 2M\*2=4M+ Introduction: 0.5M=4.5M**

***2. Explain the SSH remote port forwarding. (4.5M)***

Ans: The remote forwarding, the user’s SSH client acts on the server’s behalf.

Steps involved:

1. From the work computer, set up an SSH connection to your home computer. The firewall will allow this, because it is a protected outgoing connection.
2. Configure the SSH server to listen on a local port, say 22, and to deliver data across the SSH connection addressed to remote port, say 2222.
3. You can now go to your home computer and configure SSH to accept traffic on port 2222.
4. You now have an SSH tunnel that can be used for remote logon to the work server.

**Schema: Introduction:0.5M + each step:1M\*4=4.5M**

***3. List and briefly define the parameters for SSL session state. (8M)***

Ans: A session state is defined by the following parameters:

* Session identifier: An arbitrary byte sequence chosen by the server to identify an active or resumable session state.
* Peer certificate: An X509.v3 certificate of the peer. This element of the state may be null. Figure 17.2 TLS Protocol Stack IP TCP Record protocol Handshake protocol Change cipher spec protocol Alert protocol HTTP Heartbeat protocol 17.2 / TRANSPORT LAYER SECURITY 551
* Compression method: The algorithm used to compress data prior to encryption.
* Cipher spec: Specifies the bulk data encryption algorithm (such as null, AES, etc.) and a hash algorithm (such as MD5 or SHA-1) used for MAC calculation. It also 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

**Schema: Introduction: List the parameters :2M + Description 6M=8M**

***4. List and briefly define the parameters that define an SSL session connection. (8M)***

Ans: A connection state is defined by the following parameters:

* Server and client random: Byte sequences that are chosen by the server and client for each connection.
* Server write MAC secret: The secret key used in MAC operations on data sent by the server.
* Client write MAC secret: The secret key used in MAC operations on data sent by the client.
* Server write key: The secret encryption key for data encrypted by the server and decrypted by the client.
* Client write key: The symmetric encryption key for data encrypted by the client and decrypted by the server.
* Initialization vectors: When a block cipher in CBC mode is used, an initialization vector (IV) is maintained for each key. This field is first initialized by the SSL Handshake Protocol. Thereafter, the final ciphertext block from each record is preserved for use as the IV with the following record.
* Sequence numbers: Each party maintains separate sequence numbers for transmitted and received messages for each connection. When a party sends or receives a change cipher spec message, the appropriate sequence number is set to zero. Sequence numbers may not exceed 264 – 1.

**Schema: Introduction: List the parameters 2M + Description 6M=8M**

***5. Show the message exchange process in SSH User Authentication protocol. Identify the significant changes in TLSv1.3 from TLSv1.2. ( 12.5M)***

Ans: SSH User Authentication Message Exchange process:

1. MESSAGE EXCHANGE The message exchange involves the following steps.
2. The client sends a SSH\_MSG\_USERAUTH\_REQUEST with a requested method of none.
3. The server checks to determine if the user name is valid. If not, the server returns SSH\_MSG\_USERAUTH\_FAILURE with the partial success value of false. If the user name is valid, the server proceeds to step 3.
4. The server returns SSH\_MSG\_USERAUTH\_FAILURE with a list of one or more authentication methods to be used.
5. The client selects one of the acceptable authentication methods and sends a SSH\_MSG\_USERAUTH\_REQUEST with that method name and the required method-specific fields. At this point, there may be a sequence of exchanges to perform the method.
6. If the authentication succeeds and more authentication methods are required, the server proceeds to step 3, using a partial success value of true. If the authentication fails, the server proceeds to step 3, using a partial success value of false.
7. When all required authentication methods succeed, the server sends a SSH\_MSG\_USERAUTH\_SUCCESS message, and the Authentication Protocol is over.

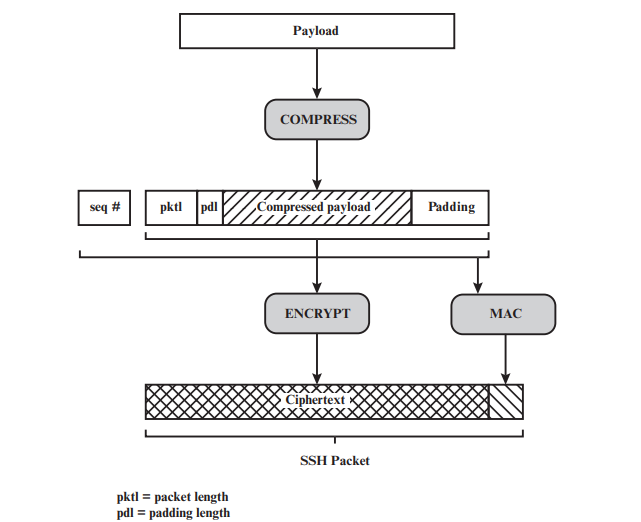
| **S.NO** | **TLS 1.2** | **TLS 1.3** |
| --- | --- | --- |
| **1.** | In TLS version 1.2 many messages move to-and-fro between Client and Server. | While TLS version 1.3 aims to reduce the time taken by the handshake process by reducing the to-and-fro messages between the Client and the Server. |
| **2.** | TLS version 1.2 has a slower TLS handshake | While; TLS version 1.3 has a faster TLS handshake |
| **3.** | It has a more complex handshake. | While; it has a simpler handshake. |
| **4.** | TLS version 1.2 has less secure Cipher suites. | While; TLS version 1.3 has more secure Cipher suites. |
| **5.** | Its round-trip time is not zero. | While; its round-trip time is zero. |
| **6.** | A typical handshake in TLS version 1.2 involves the exchange of 5 to 7 packets. | While; in TLS version 1.3 a typical handshake involves the exchange of 0 to 3 packets. |
| **7.** | It has a slower and less responsive connection. | While; it has a faster and more responsive connection. |
| **8.** | TLS version 1.2 does not shrink the size of cipher suites. | While TLS version 1.3 shrinks the size of cipher suites. |
| **9.** | Comparatively poor website performance and user experience. | While; it offers better website performance and user experience. |

**Schema: Message Exchange process: 6M + Difference: 6.5M=12.5M**

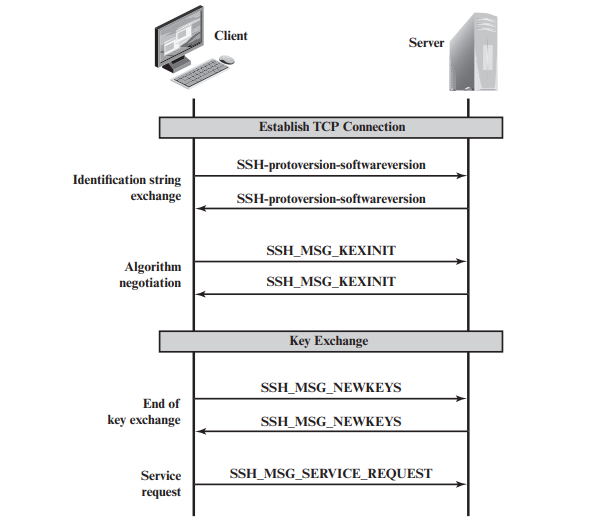
***6. Design the SSH transport layer protocol packet exchange scenario and packet format. (12.5M)***

Ans: 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. 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: Length of the random padding field.
* Payload: Useful contents of the packet. Prior to algorithm negotiation, this field is uncompressed. If compression is negotiated, then in subsequent packets, this field is compressed.
* Random padding: Once an encryption algorithm has been negotiated, this field is added. It contains random bytes of padding so that the total length of the packet (excluding the MAC field) is a multiple of the cipher block size, or 8 bytes for a stream cipher.
* Message authentication code (MAC): If message authentication has been negotiated, this field contains the MAC value. The MAC value is computed over the entire packet plus a sequence number, excluding the MAC field. The sequence number is an implicit 32-bit packet sequence that is initialized to zero for the first packet and incremented for every packet. The sequence number is not included in the packet sent over the TCP connection.



SSH Transport Layer Protocol Packet Formation



SSH Transport Layer Protocol Packet Exchanges

**Schema: Packet Format: 6M + Message Exchange: 6.5M =12.5M**

***7. Illustrate the application-level gateway (4.5M)***

Application-level gateways, also known as proxy firewalls, operate at the application layer of the OSI model. They act as intermediaries between clients and servers, intercepting and filtering all incoming and outgoing traffic. Proxy firewalls inspect the entire application layer payload, allowing for more granular control and enhanced security. Configuring proxy firewalls involves setting up specific proxies for each application or network service. (2.5M)

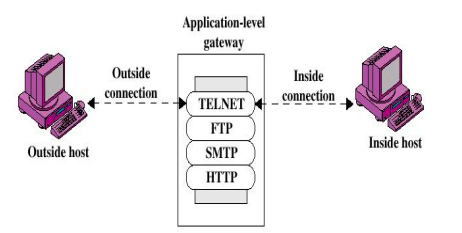
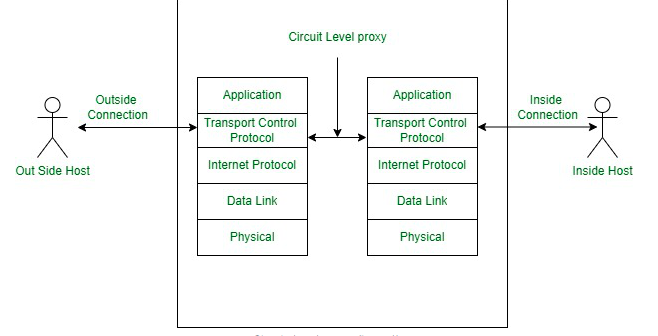


Figure: Application-level gateways. (2M)

**Schema: 2.5M+2M =4.5M**

***8. Explain the circuit-level gateway. (4.5M)***

A Circuit-Level Gateway (CLG) functions as a network security device operating within the transport layer (Layer 4) of the OSI model. In contrast to packet-filtering firewalls, which assess individual packets, CLGs specialize in overseeing and controlling the connections or circuits established between different networks. (2.5M)

**2.5M**

**Schema: 2M+2.5M**

***9. Explain the purpose padding field in ESP (8M)***

A diagram of a security system

Description automatically generated**2M**

■ Pad Length (8 bits): Indicates the number of pad bytes immediately preceding

this field.

■ Padding (0–255 bytes): The purpose of this field is mentioned below:

The Padding field serves several purposes:

1. Alignment Requirements **4M**

If an encryption algorithm requires the plaintext to be a multiple of some number of bytes (e.g., the multiple of a single block for a block cipher), the Padding field is used to expand the plaintext (consisting of the Payload Data, Padding, Pad Length, and Next Header fields) to the required length.

The ESP format requires that the Pad Length and Next Header fields be right aligned within a 32-bit word. Equivalently, the ciphertext must be an integer multiple of 32 bits. The Padding field is used to assure this alignment.

2. Confidentiality **2M**

Additional padding may be added to provide partial traffic-flow confidentiality by concealing the actual length of the payload.

**Schema: 2M+4M+2M=8M.**

***10. Give the differences between an internal and an external firewall. (8M)***

For the comparison of internal and external firewalls we need to consider the following key properties like purpose, location, focus, threats, performance and configuration complexity.

**(2M)**

Internal firewalls:

1. Protects against threats that have already breached the perimeter.
2. Deployed within the internal network.
3. Monitors and controls east-west traffic.
4. Malicious insiders, compromised devices, lateral movement.
5. Must be fast enough to handle internal traffic.
6. More complex to configure.

External firewalls:

1. Protects against threats from outside the organization.
2. Deployed at the network perimeter.
3. Monitors and controls north-south traffic.
4. Malware, phishing attacks, denial-of-service attacks.
5. Can be slower than an internal firewall.
6. Less complex to configure.

**Comparison 4\*1.5M=6M.**

**Schema: 2M+6M=8M.**

***11. Identify the difference between top-level format and payload data of an ESP packet. (12.5M)***

The Encapsulating Security Payload (ESP) protocol is a member of the IPSecc and encrypts and authenticates data packets sent between computers via a virtual private network (VPN). VPNs can work securely because of the emphasis and layers under which ESP functions. The Encapsulating Security Payload (ESP) protocol provides: (**2M)**

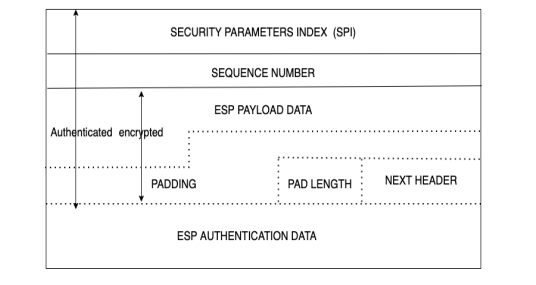
1. Data confidentiality

2. Data origin authentication

3. Data integrity and

4. Replay protection.

The top-level format of an ESP packet is given below**. (3M**)



The major difference between top-level format and payload data are. **(4.5M)**

1. The top-level format gives the architecture of the ESP packet where as the Payload is the actual data to be transported by the ESP packet.
2. The top-level format covers various parameters for the transportation of the packet such as:
   1. Security parameter index (SPI)
   2. Sequence number
   3. ESP payload data
   4. Padding
   5. Pad Length
   6. Next header
   7. ESP authentication data:
3. The payload will be encrypted while transported from the sender to receive or gateway to gateway based on Security associations (CA).

For the understanding of the ESP the above components of the ESP packet are explained: (**3M)**

1. Security parameter index (SPI): The SPI is a 32-bit value that, when combined with the packet’s destination IP address and security protocol, uniquely identifies the Security Association (SA).
2. Sequence number: The sequence number is a 32-bit counter that increases monotonically to protect against replay attacks. The sequence number is reset to 0 when a SA is established. On the sender’s and receiver’s ends, it is first set to 0. As packets move from sender to receiver, the counter is incremented. Finally, the counter is checked on the receiver’s side.
3. ESP payload data: ESP payload data is a transport-level segment or IP packet that is protected by encryption. This is where our actual message resides, and it is encrypted for confidentiality. This is a variable-length field that normally holds the data payload.
4. Padding: It is used to fill the payload data to a specific block size multiple required by a specific encryption scheme or to randomize the length of the payload in order to protect it against traffic.
5. Pad length: It is an 8-bit field whose value shows the padding field’s length in bytes.
6. Next header: The next header identifies the type of data contained in the payload data field by identifying the first header in that payload (e.g., an extension header in IPv6 or an upperlayer protocol such as TCP).
7. ESP authentication data: It is a variable-length field containing the integrity check value (ICV). ICV verifies the sender’s identity and the integrity of the message. ICV is an optional field. 9.6.2 Encryption and Authentication Algorithms and Protocols

**Schema: 2M+3M+4.5M+3M=12.5M**

***12. Develop a model to show the interrelationship between the standardized protocols of SP 800-177 for assuring message Authenticity and Integrity. (12.5M)***

The Special Publication 800-177, on "Trustworthy Email," states that the Email communications cannot be made trustworthy with a single package or application. It involves incremental additions to basic subsystems, with each technology adapted to a particular task. Some of the techniques use other protocols such as DNS to facilitate specific security functions like domain authentication, content encryption and message originator authentication. These can be implemented discretely or in aggregate, according to organizational needs.

The Special Publication 800-177, recommended various Technologies in support of core Simple Mail Transfer Protocol (SMTP): **(3M)**

1. The Domain Name System (DNS) include mechanisms for authenticating a sending domain: Sender Policy Framework (SPF), Domain Keys Identified Mail (DKIM) and Domain based Message Authentication, Reporting and Conformance (DMARC).
2. Recommendations for email transmission security include Transport Layer Security (TLS) and associated certificate authentication protocols.
3. Recommendations for email content security include the encryption and authentication of message content using S/MIME (Secure/Multipurpose Internet Mail Extensions) and associated certificate and key distribution protocols.

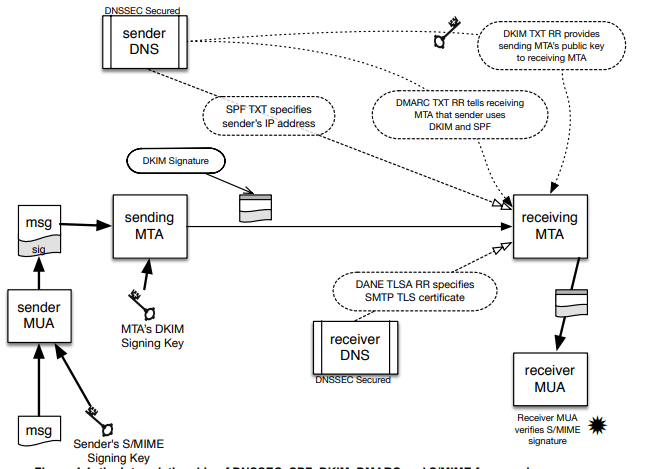
Some of the important remediations are: **(3M)**

1. Sender Policy Framework (SPF) is the standardized way for a sending domain to identify and assert the authorized mail senders for a given domain.
2. Domain based Message Authentication, Reporting and Conformance (DMARC) was conceived to allow email senders to specify policy on how their mail should be handled, the types of security reports that receivers can send back, and the frequency those reports should be sent.
3. Standardized handling of SPF and DKIM removes guesswork about whether a given message is authentic, benefitting receivers by allowing more certainty in quarantining and rejecting unauthorized mail.
4. Secure Multipurpose Internet Mail Extensions (S/MIME) is the recommended protocol for email end-to-end authentication and confidentiality. This usage of S/MIME is not common at the present time, but it is recommended.

**Model explanation: 4M+2.5M=6.5M**

The following model will ensure the compliance to the above standard the framework shall implement the above protocols in combination. The protocols are

1. Simple Mail Transfer Protocol (SMTP) as this is very old protocol it is not explained.
2. Domain based Message Authentication, Reporting and Conformance (DMARC)
3. Domain Keys Identified Mail (DKIM)
4. Transport Layer Security (TLS) and associated certificate authentication protocols
5. Secure Multipurpose Internet Mail Extensions (S/MIME)



The interrelation of SPF, DKIM, DMARC, and S/MIME signatures are shown in the above Figure.

The purpose of authenticating the sending domain is to guard against senders (both random and malicious actors) from spoofing another’s domain and initiating messages with bogus content, and against malicious actors from modifying message contents in transit. Sender **Policy Framework (SPF)** is the standardized way for a sending domain to identify and assert the authorized mail senders for a given domain. Domain Keys Identified Mail (DKIM) is the mechanism for eliminating the vulnerability of man-in-the-middle content modification by using digital signatures generated from the sending mail server.

**Domain based Message Authentication, Reporting and Conformance (DMARC)** was conceived to allow email senders to specify policy on how their mail should be handled, the types of security reports that receivers can send back, and the frequency those reports should be sent. Standardized handling of SPF and DKIM removes guesswork about whether a given message is authentic, benefitting receivers by allowing more certainty in quarantining and rejecting unauthorized mail. In particular, receivers compare the “From” address in the message to the SPF and DKIM results, if present, and the DMARC policy in the DNS. The results are used to determine how the mail should be handled. The receiver sends reports to the domain owner about mail claiming to originate from their domain. These reports should illuminate the extent to which unauthorized users are using the domain, and the proportion of mail received that is “good.”

**The Transport Layer Security Protocol (TLS)** uses an encrypted channel to protect message transfers from man-in-the-middle attacks. Email message confidentiality can be assured by encrypting traffic along the path. TLS relies on the Public Key Infrastructure (PKI) system of X.509 certificates to carry exchange material and provide information about the entity holding the certificate. These are usually generated by a Certificate Authority (CA).

**Secure Multipurpose Internet Mail Extensions (S/MIME)** is the recommended protocol for email end-to-end authentication and confidentiality. This usage of S/MIME is not common at the present time, but is recommended. Certificate distribution remains a significant challenge when using S/MIME, especially the distribution of certificates between organizations. Research is underway on protocols that will allow the DNS to be used as a lightweight publication infrastructure for S/MIME certificates

S/MIME is also useful for authenticating mass email mailings originating from mailboxes that are not monitored, since the protocol uses PKI to authenticate digitally signed messages, avoiding the necessity of distributing the sender’s public key certificate in advance. Encrypted mass mailings are more problematic, as S/MIME senders need to possess the certificate of each recipient if the sender wishes to send encrypted mail.

**Mail Transfer Agents (MTAs)** Email is transmitted, in a “store and forward” fashion, across networks via Mail Transfer Agents (MTAs). MTAs communicate using the Simple Mail Transfer Protocol (SMTP) described below and act as both client and server, depending on the situation.

**Mail User Agents (MUAs)** Most end users interact with their email system via a Mail User Agent (MUA). A MUA is a software component (or web interface) that allows an end user to compose and send messages and to one or more recipients. A MUA transmits new messages to a server for further processing (either final delivery or transfer to another server). The MUA is also the component used by end users to access a mailbox where in-bound emails have been delivered.

**Schema: 3M+3M+6.5M=12.5M**