SSL - Secure Sockets Layer

The Internet Engineering Task Force (IETF) standard called Transport Layer Security (TLS) is based on SSL.

TCP/IP Protocol Suite

- The Transmission Control Protocol/Internet Protocol (TCP/IP) governs the transport and routing of data over the Internet.
- Other protocols, such as the HyperText Transport Protocol (HTTP), Lightweight Directory Access Protocol (LDAP), or Internet Messaging Access Protocol (IMAP), run "on top of" TCP/IP in the sense that they all use TCP/IP to support typical application tasks such as displaying web pages or running email servers.

TCP/IP Protocol Suite and Security

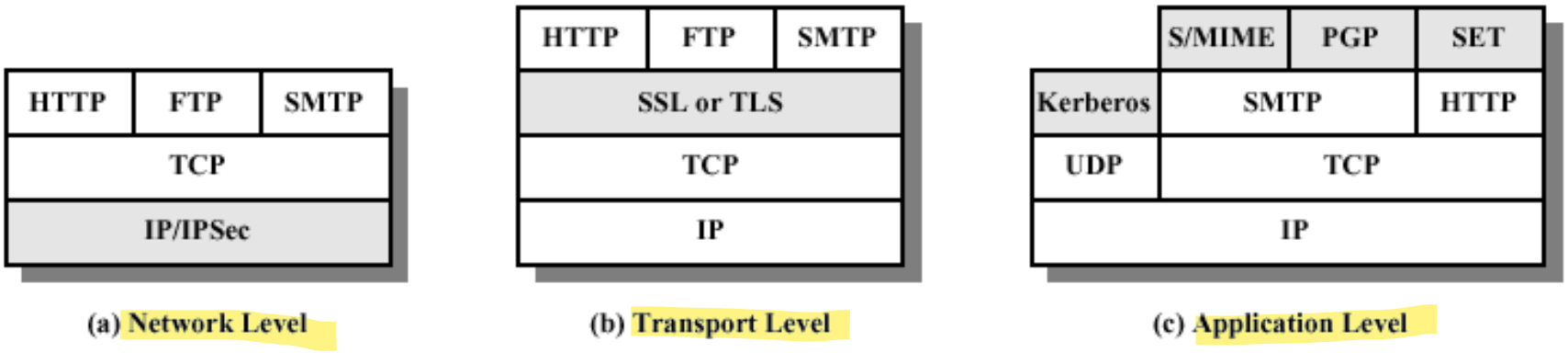
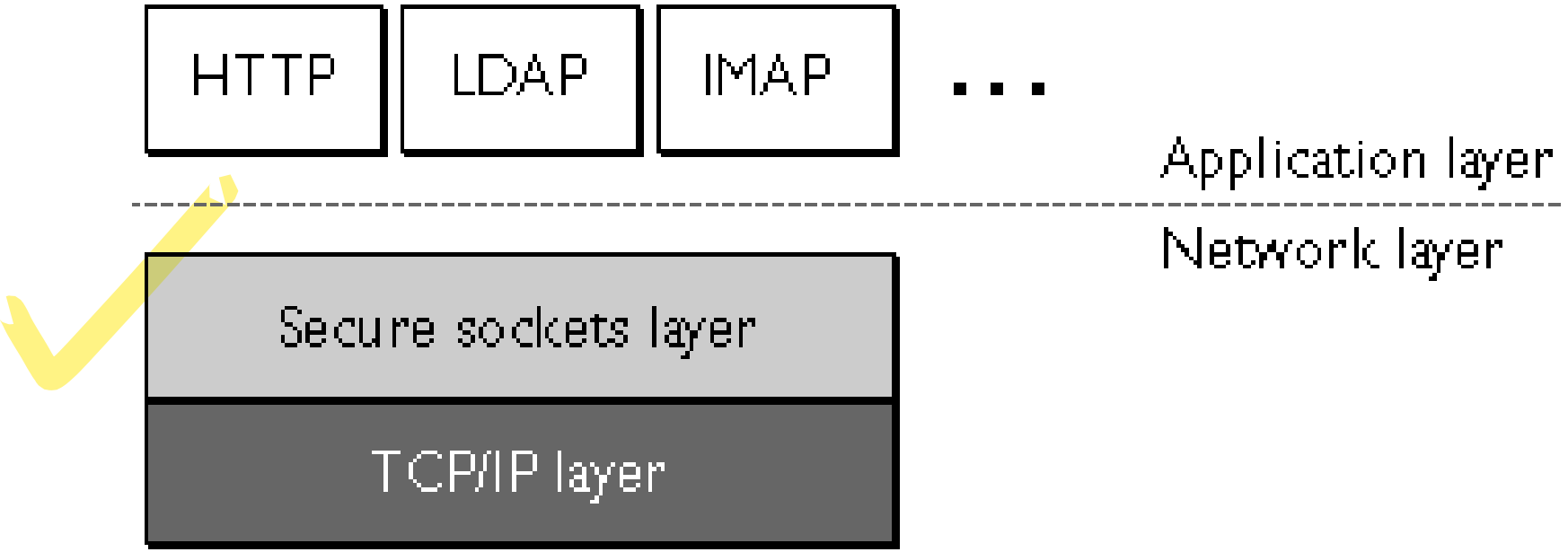


Figure 14.1 Relative Location of Security Facilities in the TCP/IP Protocol Stack

TCP/IP Protocol Suite and Security

The SSL protocol runs above TCP/IP and below higher-level protocols such as HTTP or IMAP. It uses TCP/IP on behalf of the higher-level protocols, and in the process allows an SSL-enabled server to authenticate itself to an SSL-enabled client, allows the client to authenticate itself to the server, and allows both machines to establish an encrypted connection.



Services Provided by SSL

- SSL encrypts data so that no one who intercepts is able to read it.
- SSL can assure a client that they are dealing with the real server they intended to connect to.
- SSL can prevent any unauthorized clients from connecting to the server.
- SSL prevents anyone from meddling with data going to or coming from the server.

Services Provided by SSL

- These capabilities address fundamental concerns about communication over the Internet and other TCP/IP networks:
- SSL server authentication
- SSL client authentication
- An encrypted SSL connection

SSL Server Authentication

- SSL server authentication allows a user to confirm a server's identity.
- SSL-enabled client software can use standard techniques of public-key cryptography to check that a server's certificate and public ID are valid and have been issued by a certificate authority (CA) listed in the client's list of trusted CAs.
- This confirmation might be important if the user, for example, is sending a credit card number over the network and wants to check the receiving server's identity.

SSL Client Authentication

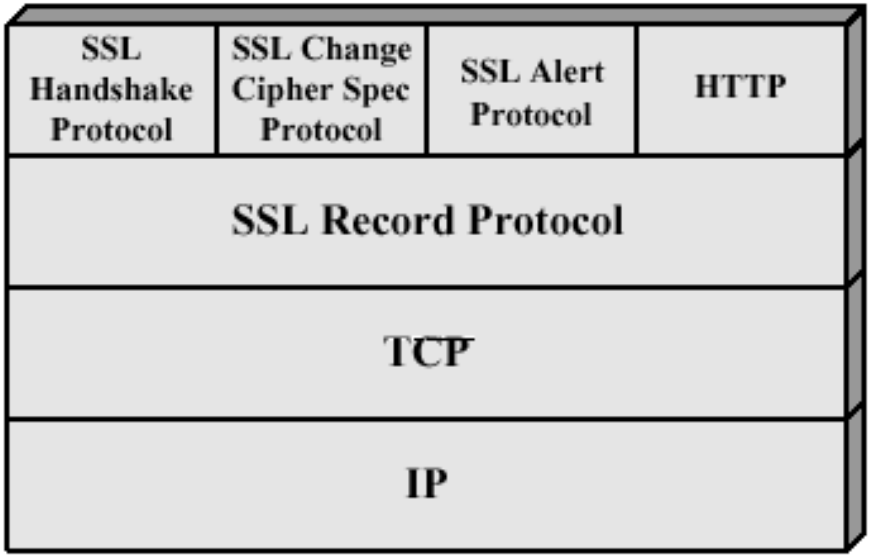
- SSL client authentication allows a server to confirm a user's identity.
- Using the same techniques as those used for server authentication, SSL-enabled server software can check that a client's certificate and public ID are valid and have been issued by a certificate authority (CA) listed in the server's list of trusted CAs.
- This confirmation might be important if the server, for example, is a bank sending confidential financial information to a customer and wants to check the recipient's identity.

An encrypted SSL connection

- An encrypted SSL connection requires all information sent between a client and a server to be encrypted by the sending software and decrypted by the receiving software, thus providing a high degree of confidentiality.
- Confidentiality is important for both parties to any private transaction.
- In addition, all data sent over an encrypted SSL connection is protected with a mechanism for detecting tampering--that is, for automatically determining whether the data has been altered in transit.

SSL Sub-protocols

- The SSL protocol includes two major sub-protocols:
- the SSL record protocol
- the SSL handshake protocol



The SSL Record Protocol

- The SSL record protocol defines the format used to transmit data
- The SSL record protocols provides two services for SSL connections:
  - Confidentiality: The Handshake Protocol defines a shared secret key that is used for conventional encryption of SSL payloads
  - Message Integrity: The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC)

The SSL Handshake protocol

- The SSL handshake protocol involves using the SSL record protocol to exchange a series of messages between an SSL-enabled server and an SSL-enabled client when they first establish an SSL connection. This exchange of messages is designed to facilitate the following actions:
  - Authenticate the server to the client.
  - Allow the client and server to select the cryptographic algorithms, or ciphers, that they both support.
  - Optionally authenticate the client to the server.
  - Use public-key encryption techniques to generate shared secrets.
  - Establish an encrypted SSL connection.