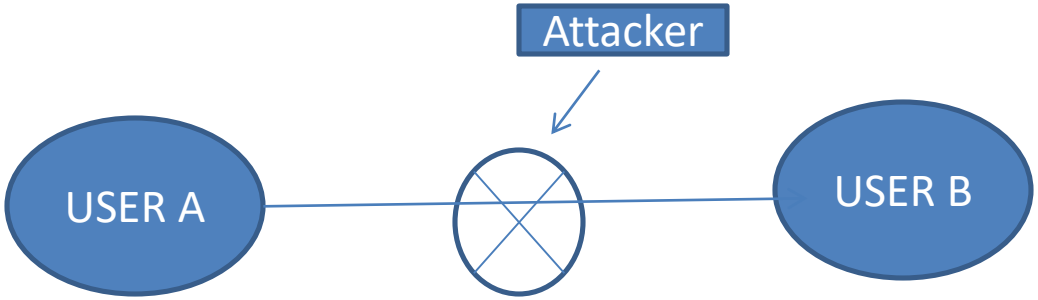
UNIT 4

Transport-Level Security:

- Web Security Considerations,
- Secure Sockets Layer,
- Transport Layer Security,
- HTTPS,
- Secure Shell(SSH)

- Web Security Considerations
- Why do we need security for our websites , internet and how can we provide security and what are different way to provide security.



When we are using internet, clicking the data or sending data to other there are always vulnerabilities(attackers) are there…there is chance of attacking the message

In order to avoid the attackers we need security…

Security is required all the web sites to protect our website content

- How to secure our web sites
- We have 6 ways
- 1. Updated Software
- 2.Beware of SQL Injections(modify the table data to disturb the integrity of data)
- 3.Cross site Scripting(XSS) (Attacker will send client side script in to our web sites Ex: Google forms—Faulty data injected to our data base)
- 4. Error Message
- 5.Data Validation
- 6 . Passwords

Web Security Threats

Table 17.1 provides a summary of the types of security threats faced when using the Web. 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.

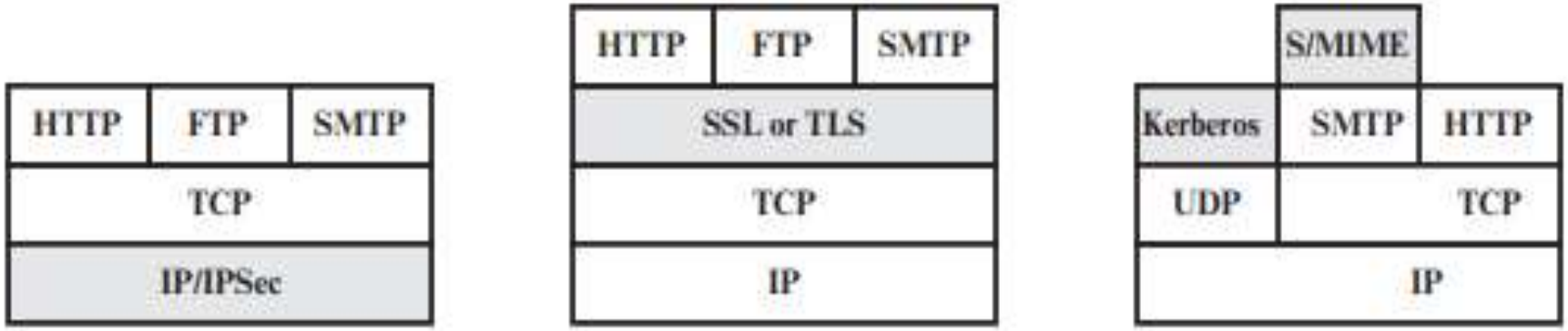
Another way to classify Web security threats is in terms of the location of the threat: Web server, Web browser, and network traffic between browser and server. Issues of server and browser security fall into the category of computer system security; Part Six of this book addresses the issue of system security in general but is also applicable to Web system security. Issues of traffic security fall into the category of network security and are addressed in this chapter.

Table 17.1 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 • Vulnerability 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

Web Traffic Security Approaches

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



(a) Network level (b) Transport level (c) Application level

Figure 17.1 Relative Location of Security Facilities in the TCP/IP Protocol Stack

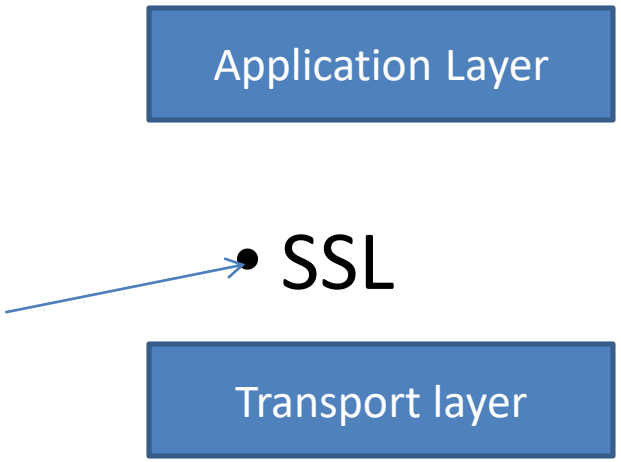
Figure 17.1 illustrates this difference. One way to provide Web security is to use IP security (IPsec) (Figure 17.1a). The advantage of using IPsec is that it is transparent to end users and applications and provides a general-purpose solution. Furthermore, IPsec includes a filtering capability so that only selected traffic need incur the overhead of IPsec processing.

Another relatively general-purpose solution is to implement security just above TCP (Figure 17.1b). The foremost example of this approach is the Secure Sockets Layer (SSL) and the follow-on Internet standard known as Transport Layer Security (TLS). At this level, there are two implementation choices. For full generality, SSL (or TLS) could be provided as part of the underlying protocol suite and therefore be transparent to applications. Alternatively, TLS can be embedded in specific packages. For example, virtually all browsers come equipped with TLS, and most Web servers have implemented the protocol.

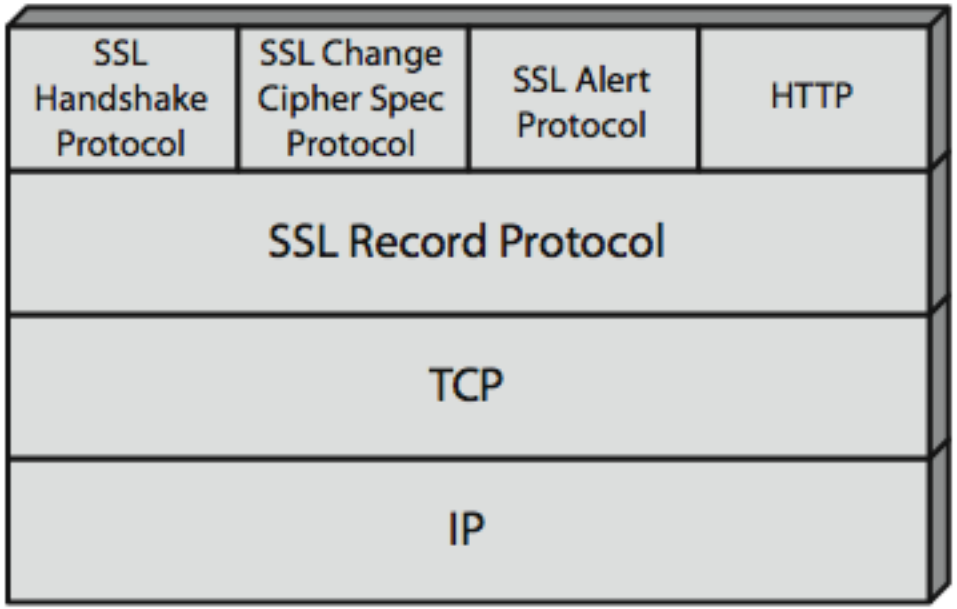
Application-specific security services are embedded within the particular application. Figure 17.1c shows examples of this architecture. The advantage of this approach is that the service can be tailored to the specific needs of a given application.

Secure Socket Layer(SSL)

- It is used to provide security for communication between two users(message is not altered ,not deleted,not known to the third person)
- In order to ensure security the message is transmitted safely from user A to user B we use the Secure socket layer protocol
- It ensures Integrity, Authentication, Confidentiality of the message
- It lies between Application layer and transport layer of TCP/IP Protocol



- Protocol Stack of SSL



- SSL Record Protocol:
- It has 2 services
- 1. Confidentiality--- message is not known to the 3rd person
- 2. Message integrity---by MAC

- The SSL Record Protocol provides basic security services to various higher-layer protocols. In particular, the Hypertext Transfer Protocol (HTTP), which provides the transfer service for Web client/server interaction, can operate on top of SSL. Three higher-layer protocols are also defined as part of SSL: the Handshake Protocol, Change Cipher Spec Protocol, and Alert Protocol. These SSL-specific protocols are used in the management of SSL exchanges.

SSL Record Protocol Operation

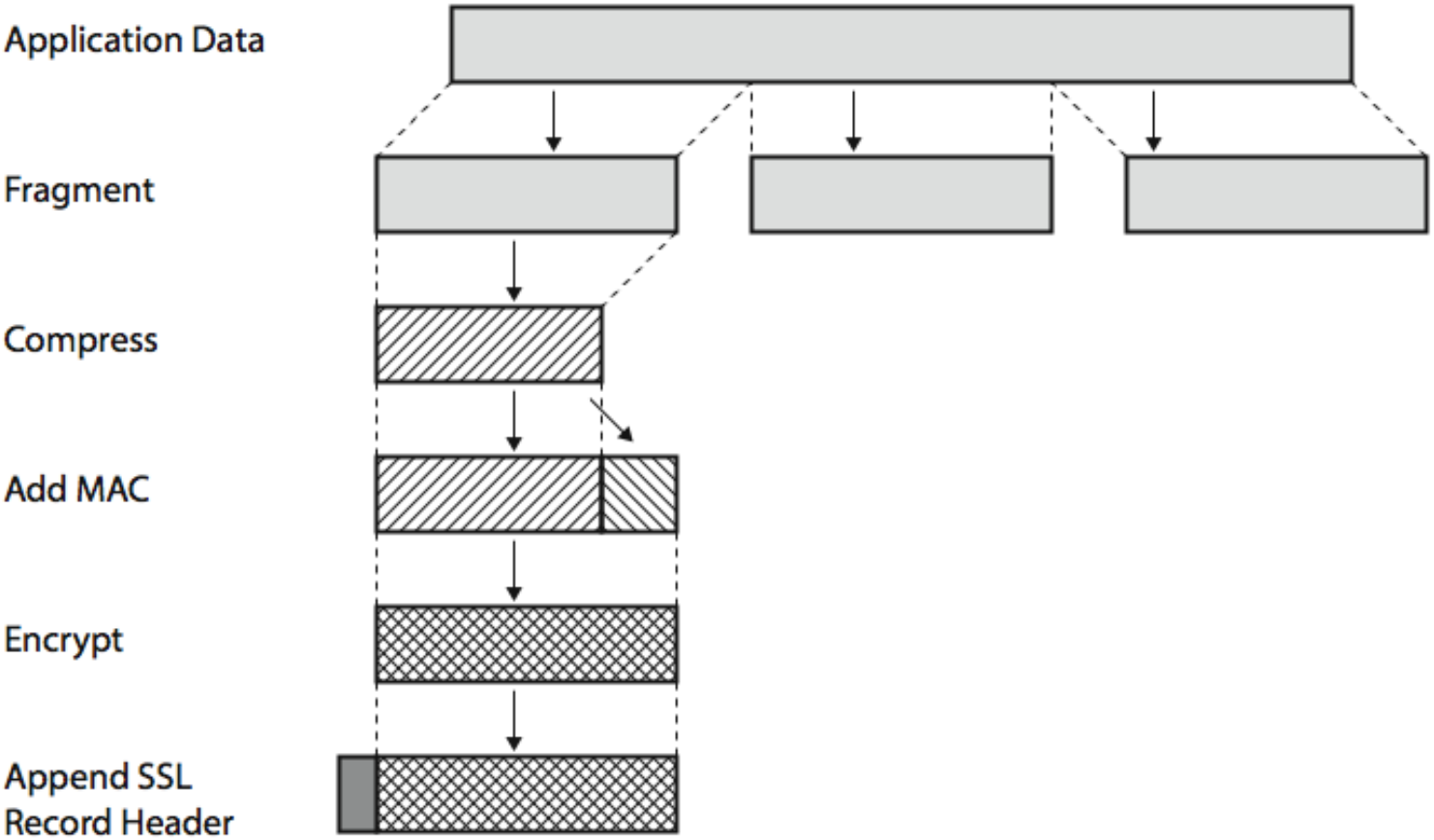


Figure shows the overall operation of the SSL Record Protocol. The Record Protocol takes an application message to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, adds a header, and transmits the resulting unit in a TCP segment. Received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

- SSL Handshake Protocol:
- -- Ensures Authentication
- -- Most complicated part it SSL
- --Key exchange between client and server
- Working:
- 1. Connection establishment with server
- 2. Key exchange from server to client
- 3. Key exchange from client to server
- 4. Handshake done from server

- The most complex part of SSL is the Handshake Protocol. This protocol allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in an SSL record. The Handshake Protocol is used before any application data is transmitted. The Handshake Protocol consists of a series of messages exchanged by client and server, which can be viewed in 4 phases:
- Phase 1. Establish Security Capabilities - this phase is used by the client to initiate a logical connection and to establish the security capabilities that will be associated with it
- Phase 2. Server Authentication and Key Exchange - the server begins this phase by sending its certificate if it needs to be authenticated.
- Phase 3. Client Authentication and Key Exchange - the client should verify that the server provided a valid certificate if required and check that the server_hello parameters are acceptable
- Phase 4. Finish - this phase completes the setting up of a secure connection. The client sends a change_cipher_spec message and copies the pending CipherSpec into the current CipherSpec

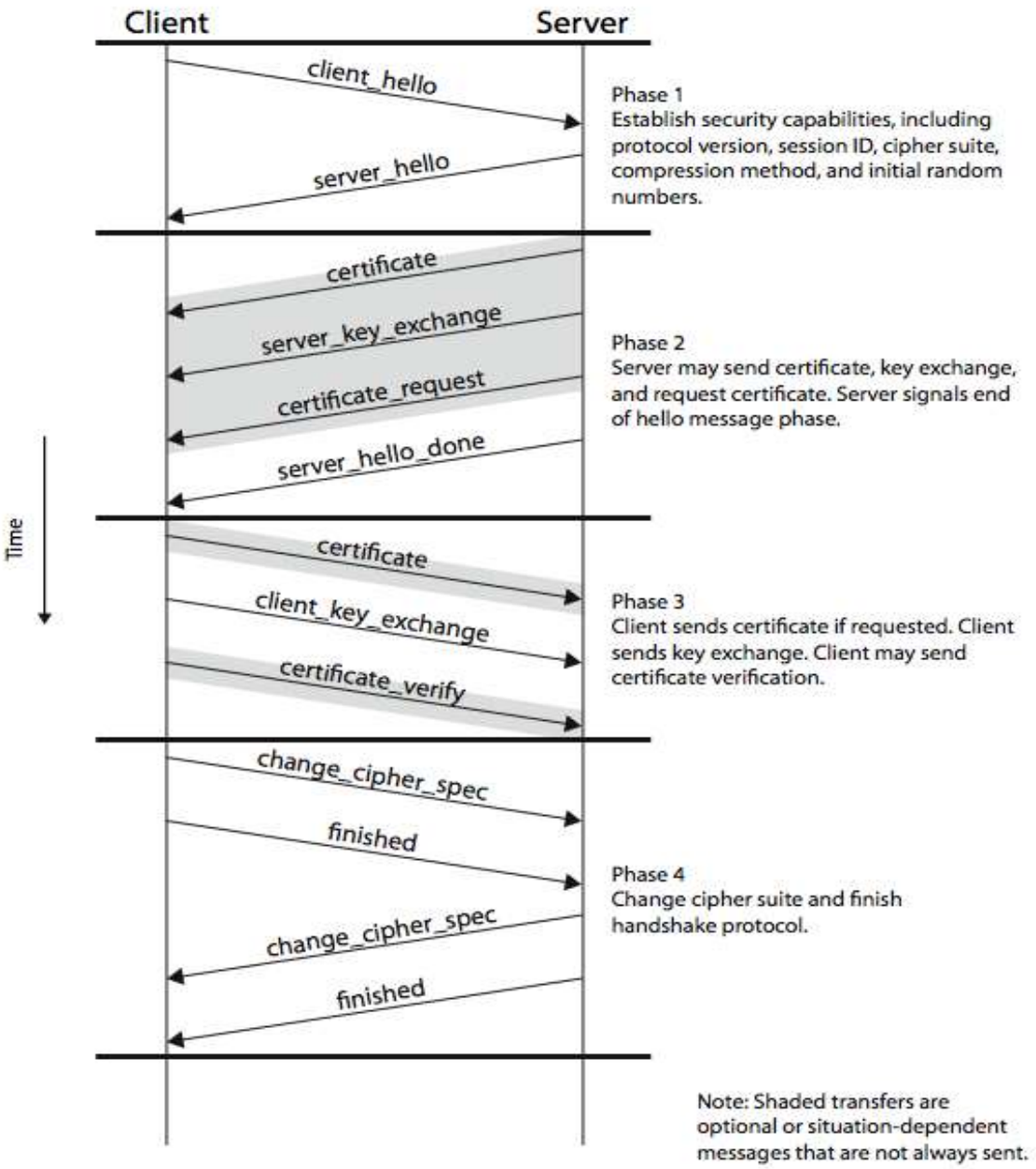


Figure shows the initial exchange needed to establish a logical connection between client and server. The exchange can be viewed as having the four phases discussed previously.

SSL Change Cipher Spec Protocol

It has only one message → single byte(1 byte)
Copies the pending state into the current state

- one of 3 SSL specific protocols which use the SSL Record protocol
- a single message
- causes pending state to become current
- hence updating the cipher suite in use

SSL Alert Protocol

-- alerts (notfications)are related to SSL are send to the clients
-- has 2 bytes

Byte1 : can have values as 1 or 2
1→ Warning 2→ Fatal Error(terminated)
Byte2 : it will specify the type of error