CHAPTER 1

History Behind Internet Security

Computers have become ubiquitous and indispensable today. Internet usage has multiplied and is a vital global tool in technology.

1.1 What is Internet Security?

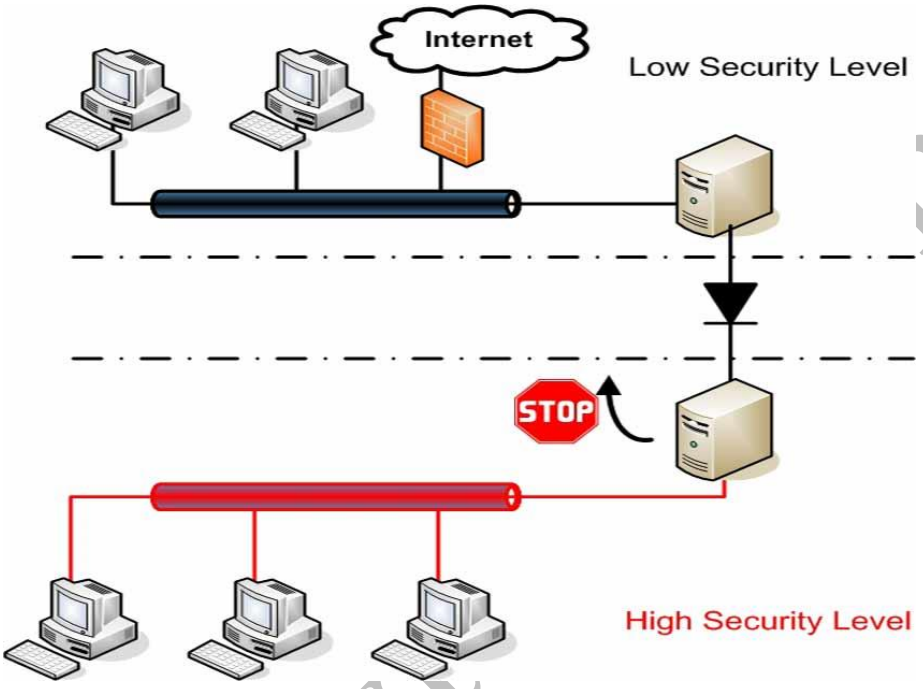
This subject came into being with the advent of the Internet. There are three basic issues - confidentiality, integrity and availability. When an unauthorized person reads or copies information, it is known as loss of confidentiality. When the information is modified in an irregular manner, it is known as loss of integrity. When the information is erased or becomes inaccessible, it is known as loss of availability. Authentication and authorization are the processes of the Internet security system by which numerous organizations make information available to those who need it and who can be trusted with it. When the means of authentication cannot be refuted later, it is known as non-repudiation. Internet security can be achieved through use of antivirus software, which quarantines or removes malicious software programs. Firewalls can determine which particular websites can be viewed and block deleterious content.

1.2 History of the Internet

In the beginning, the Internet did not exist. There were no computer networks to be found. There was no e-mail facility, and people used postal mail or the telephone to communicate. The extremely busy sent telegrams. Few people used ugly names as a euphemism for others whom they had never met. The Internet has dramatically changed all this. The Internet, started as the Advanced Research Projects Agency Network (ARPANET).

It was a tiny, isolated and restricted community. By 1996, the Internet connected an estimated 13 million computers in 195 countries on every continent, even Antarctica. The Internet is not a single network, but a worldwide collection of connected networks that are accessible by individual computer hosts by Internet service providers, gateways, routers and dial-up connections. The Internet is accessible to anyone with a computer

and a network connection. Individuals and organizations worldwide can reach any point on the network without regard to national or international boundaries or time. Today a local problem can become a global incident within a short span of time.



1.3 History of Internet Security

In 1987, the 'Vienna' virus emerged. Ralph Burger got a copy of it, disassembled it, and published the result in his book 'Computer Viruses: a High-tech Disease'. This particular book made the idea of writing viruses popular, explained how to do it, and resulted in creating up hundreds and in thousands of computer viruses implementing concepts from it. On November 2, 1988, Peter Yee at the NASA Ames Research Center sent a note out to the TCP/IP Internet mailing list that said, 'We are currently under attack from an Internet VIRUS! It has hit Berkeley, UC San Diego, Lawrence Livermore, Stanford, and NASA Ames.' Of course, this report was the first evidence of what was to be later known as The Morris Worm. Roberts, a 23-year-old Cornell University student, wrote some software code as part of a research project aimed at determining the size of the Internet. The worm was meant to infect computers, in order to see how many connections to the Internet existed. Because of a flaw in the software

code, however, it ended up exploiting vulnerabilities in Unix and spread rapidly, infecting multiple machines multiple times and rendering them unusable.

In 1994, Russian hacker Vladimir Levin broke into Citibank's cash management system and embezzled $10 million into his own accounts. The stolen accounts were unencrypted and all but $400,000 of the stolen cash was recovered and Levin was arrested He pled guilty to conspiracy to commit computer, wire and bank fraud. On April 11, 1994, a full-scale epidemic broke out, caused by file and boot polymorphic virus called 'Tequila'. In September 1994, the same thing happened with the 'Amoeba' virus.

In 1996, the 'Boza' virus emerged, which was the first virus designed specifically for Windows 95 files. In 1998, the first Java virus 'Strange Brew' affected computers.

In 2005, the Bropia Worm affected the Internet. It targeted MSN messenger for spreading.

The 2007 Storm Worm was a Trojan horse. It included an executable file as an attachment. When the e-mail recipient opened the attachment, he or she unknowingly became part of a botnet (a collection of infected computers) to spread viruses and Spam. Once infected, a computer is called as a bot. It is an instance of adaptive malware. It has been used in different kinds of criminal activities. The authors and the controllers, of the Storm Worm, have not yet been identified.

1.4 General ways of providing security

The concept of cryptography helps a lot in the security perspect. There are a lot of Encryption methods (Algorithms) are also using such as RSA. Although these applications need the security:-

- E-mail Encryption
- Web-site Encryption
- Application Encryption
- Remote user communiaction security by id and password
- Digital Signatures
- Using secure version of http (HTTPS) by SSL/TLS connection etc.

CHAPTER 2

Introduction

2.1 What is SSL ?

Secure Socket Layer (SSL) denotes the predominant security protocol of the Internet for World Wide Web (WWW) services relating to electronic commerce or home banking.

The majority of web servers and browsers support SSL as the de-facto standard for secure client-server communication. The Secure Socket Layer protocol builds up point-to-point connections that allow private and unimpaired message exchange between strongly authenticated parties.

In the ISO/OSI reference model [ISO7498], SSL resides in the session layer between the transport layer (4) and the application layer (7); with respect to the Internet family of protocols this corresponds to the range between TCP/IP and application protocols such as HTTP, FTP, Telnet, etc. SSL provides no intrinsic synchronization mechanism; it relies on the data link layer below.

The SSL protocol allows mutual authentication between a client and server and the establishment of an authenticated and encrypted connection. SSL runs above TCP/IP and below HTTP, LDAP, IMAP, NNTP, and other high-level network protocols.

In general:

SSL – Secure Socket Layer

- It provides a secure transport connection between applications (e.g., a web server and a browser),
- SSL was developed by Netscape,

  – V2 1994 netscape

  – V3 1996 netscape
- SSL version 3.0 has been implemented in many web browsers (e.g., Netscape

Navigator and MS Internet Explorer) and web servers and widely used on the Internet.

- A protocol widely used on the Web

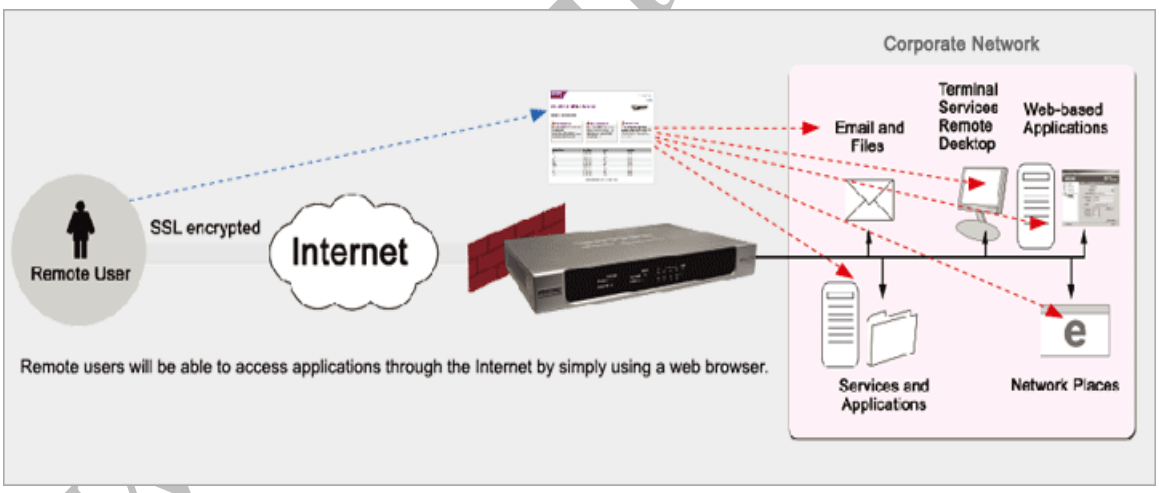
– Operates between the application and transport layers.

HTTP, FTP,SMTP
SSL **TCP IP**
Data Link
Physical

- Operations of SSL
  - Negotiation for PKI

    Server and browser negotiate to select cryptographic algorithm and create a session secret key.
  - Communications.

    Encrypted by using the key that was negotiated.



2.2 Evolution of SSL ?

Netscape developed the first specification of SSL in 1994, but only publicly released and deployed the next version, SSLv2, in the same year [SSL2]. With respect tc public key

cryptography, it relies mainly on RSA encryption (RSA cryptosystem) and X.509-compliant certificates. Block ciphers, such as DES, Triple DES (3DES), and RC4, along with hash functions like MD5 and SHA, complement the suite of algorithms. SSLv3

followed in 1995, adding cryptographic methods such as Diffie-Hellman key agreement (DH), support for the FORTEZZA key token, and the Digital Signature Standard (DSS) scheme [SSL3].

The most recent draft of the SSL 3.0 specification was published in November of 1996 by Netscape. The intent was to be a "security protocol that provides communications privacy over the Internet. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery." The goals included cryptographic security, interoperability, extensibility, and relative efficiency.

Interoperability was a goal so that applications could be written to the standard and expected to work with any other applications written to the standard. Interoperability, it was noted, does not imply that two programs will always be able to connect. One might not have the correct algorithm support or credentials necessary for the connection to the other.

Extensibility was descried as providing "a framework into which new public key and bulk encryption methods can be incorporated as necessary." It was noted that this should prevent the need to implement a new security protocol entirely should a weakness be found in one of the current encryption methods.

Cryptography, obviously, causes a higher CPU load than sending the data unencrypted. Still, they made some effort to minimize the network traffic and allow for session caching.

2.2.1 SSL v2.0

Released by Netscape Communications in 1994. The main goal of this protocol was to provide security for transactions over the World Wide Web. Unfortunately, very quickly a number of security weaknesses were found in this initial version of the SSL protocol, thus making it less reliable for commercial use:

- weak MAC construction
- possibility of forcing parties to use weaker encryption
- no protection for handshakes

Possibility of an attacker performing truncation attacks.

2.2.2 PCT v1.0

Developed in 1995 by Microsoft. Privacy Communication Technology (PCT) v1.0 addressed some weaknesses of SSL v2.0, and was aimed to replace SSL. However, this protocol has never gained as much popularity as SSL v3.0.

2.2.3 SSL v3.0

Released in 1996 by Netscape Communications. SSL v3.0 solved most of the SSL v2.0 problems, and incorporated many of the features of PCT. Pretty quickly become the most popular protocol for securing communication over WWW.

2.2.4 TLS v1.0 (also known as SSL v3.1)

Published by IETF in 1999 (RFC 2246). This protocol is based on SSL v3.0 and PCT and harmonizes both Netscape's and Microsoft's approaches. It is important to note that although TLS is based on SSL, it is not a 100% backward compatible with its predecessor. IETF did some security improvements, such as using HMAC instead of MAC, using a different calculation of the master secret and key material, adding additional alert codes, no support for Fortezza cipher suites, and so on. The end result of these improvements is that these protocols don't fully interoperate. Fortunately enough, TLS has also got a mode to fall back to SSL v3.0.

Re version numbers: Both SSL2 and SSL3 have 16-bit (two-byte) version number fields. SSL2 interprets this as a single 16-bit integer, and the official number is 2, e.g. 0x0002. SSL3 interprets two-byte version numbers as a one byte "major" number and a one byte "minor" (or fractional) number. So the value 0x0002 is interpret by SSL3 as version 0.2, not 2.0.

2.3 What is TLS ?

SSL 3.0 was the basis for the TLS 1.0 (RFC 2246) specification published by the Internet Engineering Task force (IETF) in 1999.

In actual TLS was just a minor modification in SSL.

In general:

TLS – Transport Layer Security

- TLS can be viewed as SSL v3.1,
- SSL v3.0 was specified in an Internet Draft (1996), it evolved into RFC 2246 and was renamed to TLS (Transport Layer Security),
- V1.0 1999 RFC2246 IETF minor update from SSL v3.0,
- V1.1 April 2006 RFC4346 updates to prevent specific security attacks.

2.4 Evolution of TLS ?

SSL v3.0 was actually renamed into TLS. SSL version 3.0 and its designated successor protocol Transport Layer Security (TLS) 1.0, which the Internet Engineering Task Force (IETF) published for the first time in 1999 [RFC2246]. The IETF published the most recent Internet-Draft for TLS 1.1 in Oct. 2002 [TLS].

The TLS 1.0 specification described itself as being similar to but not backwards compatible with the SSL 3.0 specification. It did include a fallback mechanism for SSL 3.0 if TLS was not available. The IETF made some small changes and clarifications and published RFC4346 in 2006 detailing TLS 1.1. There is currently a working draft for TLS 1.2 (RFC Draft 4346) which expired in September 2007.

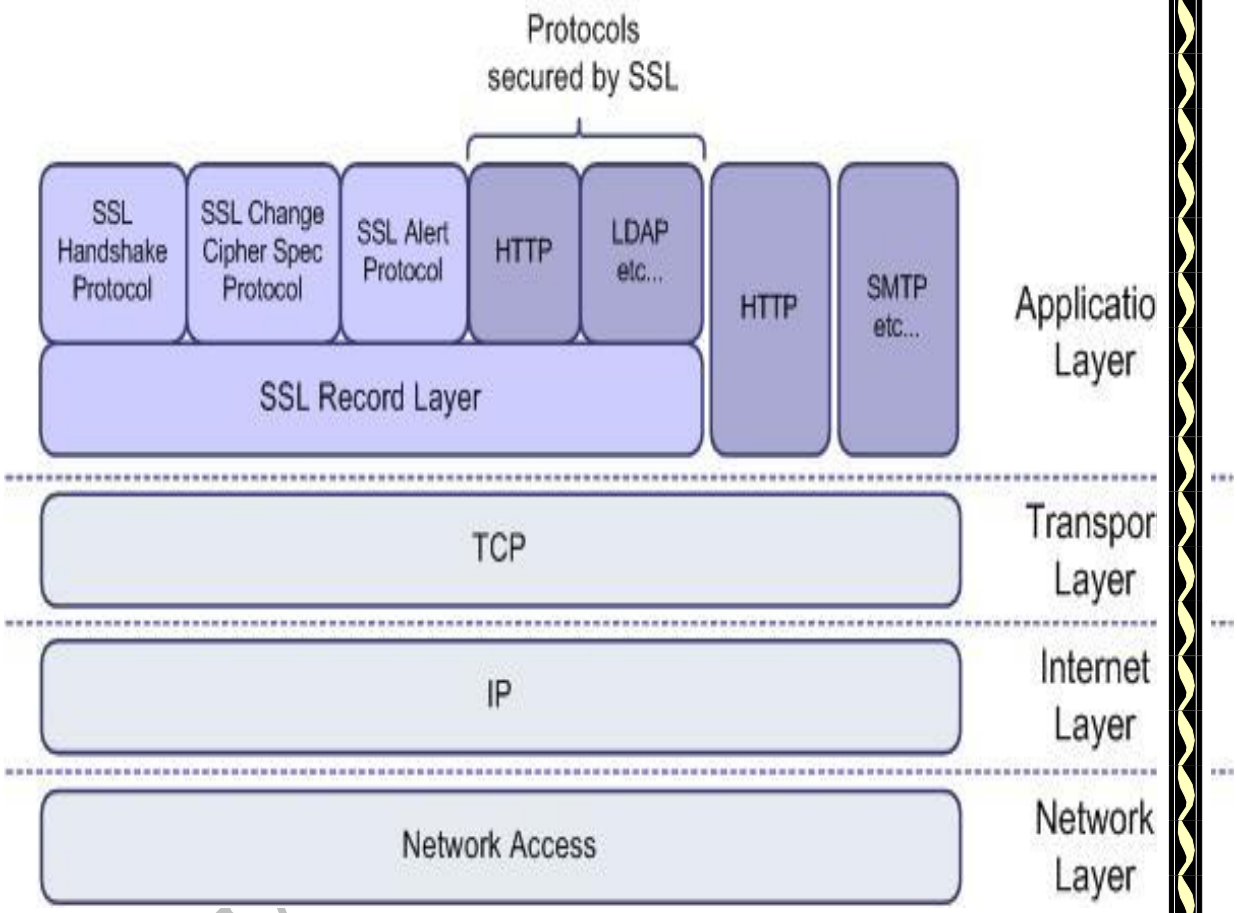
2.5 SSL/TLS Architecture :

SSL/TLS has 4 underlying protocols: *Handshake*, *Record*, *Change Cipher Spec*, and *Alert*.

All other SSL/TLS protocols reside inside of the Record protocol. This is laid out as:

8 bit	8 bit	8 bit	16 bit	16384 byte
Type	Minor version	Major version	Record Length	Record Data ....

Protocol Architecture:



CHAPTER 3

SSL/TLS working and description

3.1 SSL layers and working?

SSL splits into distinct layers and message types.SSL/TLS has 4 underlying protocols: *Handshake*, *Record*, *Change Cipher Spec*, and *Alert.*

The working of these layers as follows:-

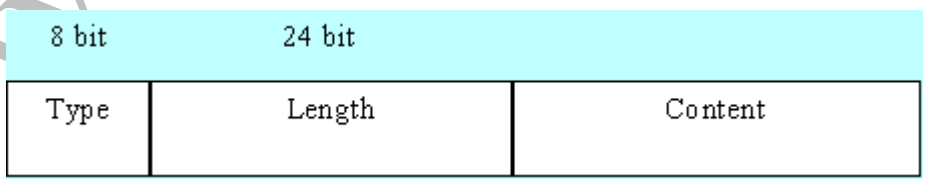
3.1.1 SSL Handshake protocol :-

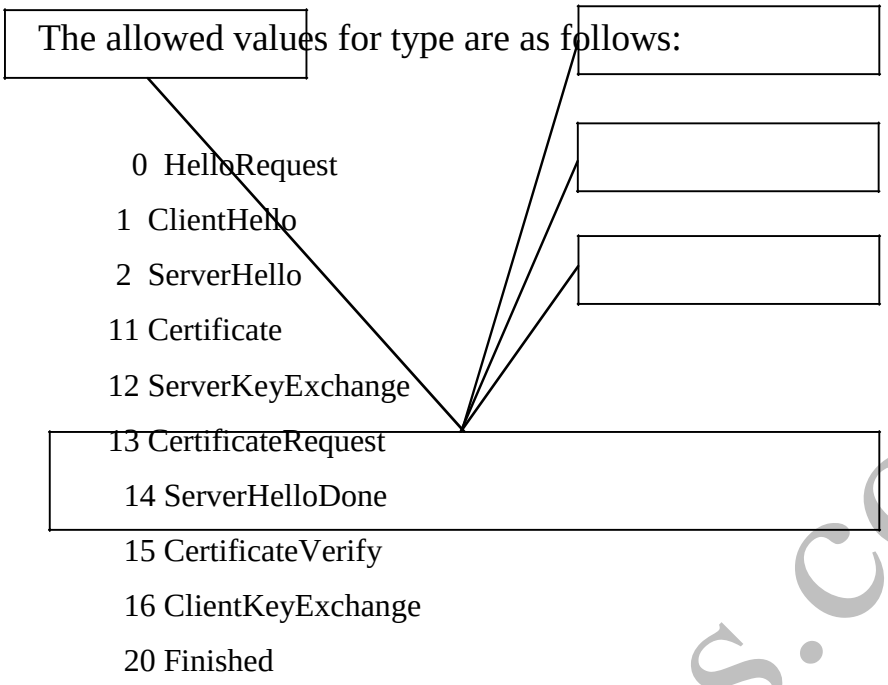
The *handshake* message sequence initiates the communication, establishes a set of common parameters like the protocol version, applicable cryptographic algorithms (*cipher suites*), and assures the validity of the message sequence. During the handshake, the participants accomplish the negotiated authentication and derive the session key material.

In this way Handshake protocol does :-

- Negotiation of security algorithms and parameters,
- Key exchange,
- Server authentication and optionally client authentication.

TLS connections begin with a 6-way handshake. The handshake protocol structure is:

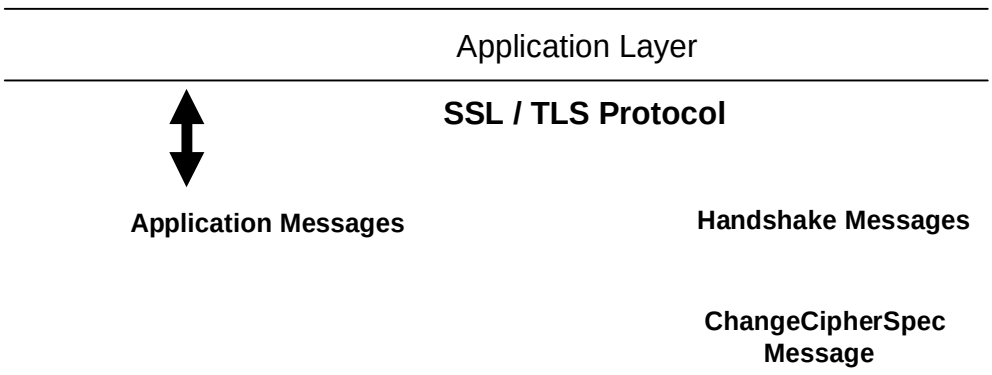


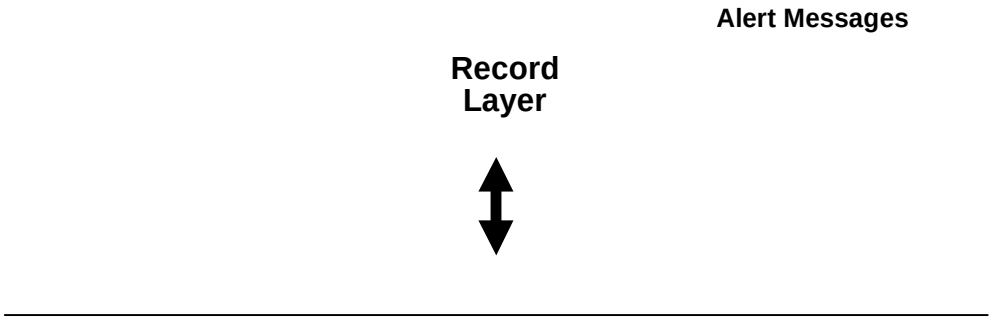


3.1.2 SSL Record protocol :-

The record layer fragments the full data stream into records with a maximum size of 2^{14} bytes and envelopes them cryptographically under the current session keys. Records contain a keyed message authentication code (HMAC). The initial handshake presupposes a NULL cipher suite applying no encryption and no HMAC. The record layer fully provides the use of compression. However, for patent reasons the core specifications name no method explicitly, except for the mandatory NULL algorithm, which practically makes compression an incompatible, implementation-dependent feature.

The basic layer structre and message is as follows:





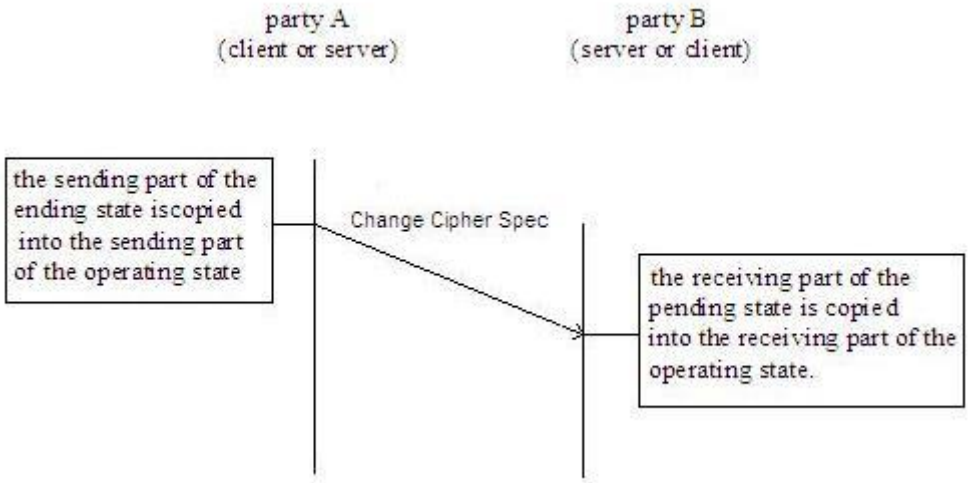
SSL Layer and Message Structure

In this way the Record protocol does:-

- Fragmentation,
- Compression,
- Message authentication and integrity protection,
- Encryption.

3.1.3 SSL Change Cipher Spec Protocol:-

It is a single message that indicates the end of the SSL handshake. The change cipher can understood as follows:



State Changes

Here

Operating state

– currently used state.

Pending state

- state to be used,
- built using the current state.

3.1.4 SSL Alert Protocol:-

Alert messages inform on exceptional protocol conditions (fatal alerts and warnings) or on a participant's request to end the communication (closure alert).

Each alert message consists of 2 fields (bytes)

1. first field (byte): "warning" or "fatal"
2. second field (byte):
   - 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

- In case of a fatal alert
  - connection is terminated,
  - session ID is invalidated,
  - no new connection can be established within this session.

3.2 SSL Encryption and Header ?

SSL can use following Encrption and Header types:-

3.2.1 Encryption:-

It supports following algorithms:-

- block ciphers (in CBC mode)
  - RC2_40
  - DES_40
  - DES_56
  - 3DES_168
  - IDEA_128
  - Fortezza_80
- stream ciphers
  - RC4_40
  - RC4_128

If a block cipher is used, than padding is applied, last byte of the padding is the padding length.

3.2.2 Header :-

It supports following header types:-

- change_cipher_spec
- alert
- handshake
- application_data

The higher level protocol used to process the enclosed fragment. Length (in bytes) of the enclosed fragment or compressed fragment max value is $2^{14}+2048$.

3.3 SSL Working ?

The working of SSL can be show as following:-

The SSL handshake accomplishes three goals. *Firstly*, both parties agree on a cipher suite, i.e. the set of cryptographic algorithms that they intend to use for application data protection. *Secondly*, they establish a common master_ secret in order to derive their session key material. *Thirdly*, the participant's identities are authenticated. Although the SSL specification permits anonymous, server-only and mutual authentication, it is customary to only assert the server's identity.

This Figure gives an overview of the SSL protocol variants. It comprises four different handshake sequences, each identified by a capital letter:
S,C,E,R which denotes:-

- S denotes the server-authenticated message flow.
- C marks the sequence with additional client authentication.
- E shows the handshake variant with ephemeral Diffie-Hellman key agreement.
- R stands for the handshake of resumed sessions. Note, that the message pairs.

ChangeCipherSpec / Finished of client and server are drawn in reverse order; the server message pair follows ServerHello immediately.

REFERENCES

Websites:-

1. www.google.com
2. http://en.wikipedia.org/wiki/Ssl
3. www.open**ssl**.org/
4. www.VeriSign.com

Books:-

1. SSL & TLS Essentials- *by Stephen A. Thomas*
2. HTTP Essentials- *by Stephen Thomas*
3. Network Security with OpenSSL- *by John Viega, et al*
4. SSL and TLS- *by Eric Rescorla (Author)*