Secure Socket Layer

M S.Bhiogade
Patni Computer Services, Mumbai, India

mittal.bhiogade@patni.com or mittalb@hotmail.com

Abstract

This paper tells about the need for security on the Internet, SSL Protocol, and how a Certificate is used to meet the demand for safe interaction over the Internet.

Keywords: SSL, Certificate, and Certificate Authority.

Introduction

When we walk into a store, we know whom we are dealing with. We see the products, the branding and the store assistant. We can be sure that if something would be wrong with our purchase, we'll recourse to the store manager or owner. But on the Internet, website visitors generally have no reliable way of knowing who owns the website (the virtual store). When customers visit a website with the intent of making an online purchase, they want to know whom they'll be paying. They want proof of the identity of the website owner, and they want to know that the personal information they send to the website cannot be intercepted by other Internet users. This is where SSL certificates come to the fore.

SSL (Secure Socket Layer) is a protocol developed by Netscape that enables a web browser and a web server to communicate securely; it allows the web browser to authenticate the web server. The SSL protocol requires the web server to have a digital certificate installed on it in order for an SSL connection to be made. SSL works by using a public key to encrypt data that's transferred over the SSL connection. Both Netscape Navigator and Internet Explorer support SSL, and many websites use the protocol to obtain confidential user information, such as credit card numbers. By convention, URLs that require an SSL connection start with *https:* instead of *http*.

SSL

SSL stands for Secure Sockets Layer protocol developed by Netscape and is the standard Internet protocol for secure communications. The secure hypertext transfer protocol (HTTPS) is a communications protocol designed to transfer encrypted information between computers over the World Wide Web. HTTPS is http using a Secure Socket Layer (SSL). A secure socket layer is an encryption protocol invoked on a Web server that uses HTTPS. SSL is a type of sockets communication and resides between TCP/IP and upper layer applications, requiring no changes to the application layer

SSL is used typically between server and client to secure the connection. A common TCP/IP sockets call is substituted for a call to SSL sockets and a variety of application programming interfaces (APIs) are offered. This approach of "plugging in" security at the socket layer can significantly reduce development time in contrast to building and incor-

porating the necessary cryptographic components to assure the same security.

SSL Protocol

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.

Figure 1, shows SSL runs above TCP/IP and below high-level application protocols

The SSL protocol includes two sub-protocols: the SSL record protocol and the SSL handshake protocol. The SSL record protocol defines the format used to transmit data. 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:

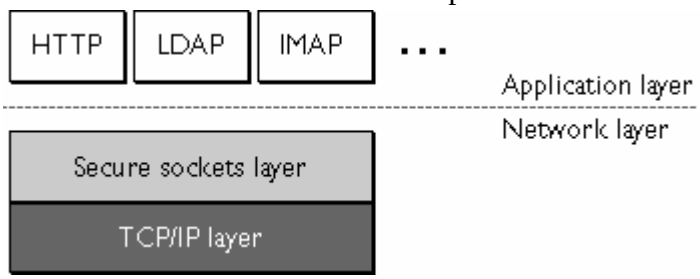

Figure 1. Location of Secure Socket Layer

- 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.

The SSL protocol supports the use of a variety of different cryptographic algorithms, or ciphers, for use in operations such as authenticating the server and client to each other, transmitting certificates, and establishing session keys. Clients and servers may support different cipher suites, or sets of ciphers, depending on factors such as the version of SSL they support, company policies regarding acceptable encryption strength, and government restrictions on export of SSL-enabled software. Among its other functions, the SSL handshake protocol determines how the server and client negotiate which cipher suites they will use to authenticate each other, to transmit certificates, and to establish session keys.

The cipher suite descriptions that follow refer to these algorithms [1]:

- **DES**. Data Encryption Standard, an encryption algorithm used by the U.S. Government.
- **DSA**. Digital Signature Algorithm, part of the digital authentication standard used by the U.S. Government.
- **KEA**. Key Exchange Algorithm, an algorithm used for key exchange by the U.S. Government.
- **MD5**. Message Digest algorithm developed by Rivest.
- **RC2** and **RC4**. Rivest encryption ciphers developed for RSA Data Security.
- **RSA**. A public-key algorithm for both encryption and authentication. Developed by Rivest, Shamir, and Adleman.
- **RSA** key exchange. A key-exchange algorithm for SSL based on the RSA algorithm.

- **SHA-1**. Secure Hash Algorithm, a hash function used by the U.S. Government.
- **SKIPJACK**. A classified symmetric-key algorithm implemented in FORTEZZA-compliant hardware used by the U.S. Government.
- **Triple-DES**. DES applied three times.

Key-exchange algorithms like KEA and RSA key exchange govern the way in which the server and client determine the symmetric keys they will both use during an SSL session. The most commonly used SSL cipher suites use RSA key exchange.

Public key encryption is a technique that uses a pair of asymmetric keys for encryption and decryption. Each pair of keys consists of a public key and a private key. The public key is made public by distributing it widely. The private key is never distributed; it is always kept secret. Data that is encrypted with the public key can be decrypted only with the private key. Conversely, data encrypted with the private key can be decrypted only with the public key. This asymmetry is the property that makes public key cryptography so useful

SSL Handshake

An SSL session always begins with an exchange of messages called the SSL handshake. The handshake allows the server to authenticate itself to the client using public-key techniques.

The steps involved in messages exchanged during SSL handshake is summarized below [1]:

- The client sends the server the client's SSL version number, cipher settings, randomly generated data, and other information the server needs to communicate with the client using SSL.
- The server sends the client the server's SSL version number, cipher settings, randomly generated data, and other information the client needs to communicate with the server over SSL. The server also sends its own certificate and, if the client is requesting a server resource that requires client authentication, requests the client's certificate.
- The client uses some of the information sent by the server to authenticate the server. If the server cannot be authenticated, the user is warned of the problem and informed that an encrypted and authenticated connection cannot be established. If the server can be successfully authenticated, the client goes on to next step.
- Using all data generated in the handshake so far, the client (with the cooperation of the server, depending on the cipher being used) creates the premaster secret for the session, encrypts it with the server's public key, and sends the encrypted premaster secret to the server.
- If the server has requested client authentication (an optional step in the handshake), the client also signs another piece of data that is unique to this handshake and known by both the client and server. In this case the client sends both the signed data and the client's own certificate to the server along with the encrypted premaster secret.
- If the server has requested client authentication, the server attempts to authenticate the client. If the client cannot be authenticated, the session is terminated. If the client can be successfully authenticated, the server uses its private key to decrypt the premaster secret, then performs a series of steps (which the client also performs, starting from the same premaster secret) to generate the master secret.
- Both the client and the server use the master secret to generate the session keys, which are symmetric keys used to encrypt and decrypt information exchanged during the SSL session and to verify its integrity--that is, to detect any changes in the data between the time it was sent and the time it is received over the SSL connection.

- The client sends a message to the server informing it that future messages from the client will be encrypted with the session key. It then sends a separate (encrypted) message indicating that the client portion of the handshake is finished.
- The server sends a message to the client informing it that future messages from the server will be encrypted with the session key. It then sends a separate (encrypted) message indicating that the server portion of the handshake is finished.
- The SSL handshake is now complete, and the SSL session has begun. The client and the server use the session keys to encrypt and decrypt the data they send to each other and to validate its integrity.

Certificates

Certificates are digital documents attesting to the binding of a public key to an individual or other entity. They allow verification of the claim that a specific public key does in fact belong to a specific individual. Certificates help prevent someone from using a phony key to impersonate someone else. In their simplest form, certificates contain a public key and a name. As commonly used, a certificate also contains an expiration date, the name of the certifying authority that issued the certificate, a serial number, and perhaps other information. Most importantly, it contains the digital signature of the certificate issuer.

An SSL certificate contains the following information [2]:

- The domain for which the certificate was issued.
- The owner of the certificate (who is the also the person/entity who has the right to use the domain).
- The physical location of the owner.
- The validity dates of the certificate.

Installing a digital certificate on the server lets you:

- **Authenticate your site**. A digital certificate on your server automatically communicates your site's authenticity to visitors' web browsers, confirming that the visitor is actually communicating with you, and not with a fraudulent site stealing credit card numbers or personal information.
- **Keep private communications private**. Digital certificates encrypt the data, that visitor’s exchange with your site to keep it safe from interception or tampering using SSL technology, the industry-standard method for protecting web communications.

When you connect to a secure web server for example https://www.xyz.com, that server authenticates itself to the web browser by presenting a digital certificate. This authentication is quite a complex process that involves the exchange of a “public key” and the use of a “session key “ for encryption. The process is seamless to the user. The certificate serves as proof that an independent trusted third party, such as Verisign, has verified that the server belongs to the company it claims to belong to. A valid certificate gives customer’s confidence that they are sending personal information securely, and to the right place. [2]

Certificates are issued by a certifying authority (CA), which can be any trusted central administration willing to vouch for the identities of those to whom it issues certificates and their association with a given key. In order to prevent forged certificates, the CA's public key must be trustworthy: a CA must either publicize its public key or provide a certificate from a higher-level CA attesting to the validity of its public key.

Certificate issuance proceeds as follows. Alice generates her own key pair and sends the public key to an appropriate CA with some proof of her identification. The CA checks the identification and takes any

other steps necessary to assure itself the request really did come from Alice and that the public key was not modified in transit, and then sends her a certificate attesting to the binding between Alice and her public key along with a hierarchy of certificates verifying the CA's public key. Alice can present this certificate chain whenever desired in order to demonstrate the legitimacy of her public key.

Working of Certificate

Server certificates take advantage of SSL to work seamlessly between your site and your visitors' web browsers.

- A customer contacts the website, accessing a secured URL (indicated by a URL that begins with "https:" instead of just "http:").
- The server responds, automatically sending the customer, website’s digital certificate, which authenticates your site.
- The customer's web browser generates a unique "session key" to encrypt all communications with the site.
- The customer's browser encrypts the session key itself with the website's public key so only the site can read the session key.
- A secure session is now established. It all takes only seconds and requires no action by the user. Depending on the browser, the user may see a key icon becoming whole or a padlock closing, indicating that the session is secure.

SSL Benefits

A customer connecting to a secure website is assured of three things:

- Authentication: The company that installed the certificate really owns the website.
- Message privacy: Using a unique “session key”, SSL encrypts all information exchanged between your web server and your customers, such as credit card numbers and other personal data. This ensures that personal information cannot be viewed if it is intercepted by unauthorized parties.
- Message integrity: The data cannot be tampered with over the Internet.
- Increasing Business: Certificates let you securely exchange sensitive information online and increase business by giving your customers confidence that their transactions are safe.

Conclusion

If it is important for you to assure your customers that they are not at risk when sending data over the Internet, you should get a certificate. If you have more than one domain name to secure, then you should have more than one certificate. The certificates are domain name and host name specific, so we will need as many certificates as we have domain names. Reassurance pays, the e-commerce business will benefit from the SSL enabled web server and certificate, and we can forecast an increase in online purchases from customers who feel more secure buying online. SSL provides confidence in the integrity and security in online business and network infrastructure. Customers are becoming increasingly aware of the advantages of SSL security and will often not purchase online from non-secure stores. All major web merchants use SSL security backed by strong warranties to encourage customers to buy online.

Acknowledgements

The author thanks Mrs. Sonal Desai for her valuable insights in this work.

References

[1] http://www.netscape.com

[2] http://www.verisign.com

Biography

M.S.Bhiogade is currently working as Software Engineer with Patni Computer Services. He has received B.E. (Computer Engineer.) in 1999 from Sardar Patel College of Engineering. (S.P.C.E), Mumbai University Maharashtra. His research areas are Internet Security, Networking, and Object Oriented Programming.

HomePage: http://personal.vsnl.com/mittalbhiogade/index.htm