Use of SSL in Web Security

- Yashashree Kolhe (111508041)

**Introduction**

SSL stands for Secure Sockets Layer and, in short, it's the standard technology for keeping an internet connection secure and safeguarding any sensitive data that is being sent between two systems, preventing criminals from reading and modifying any information transferred, including potential personal details. The two systems can be a server and a client (for example, a shopping website and browser) or server to server (for example, an application with personal identifiable information or with payroll information).

It does this by making sure that any data transferred between users and sites, or between two systems remain impossible to read. It uses encryption algorithms to scramble data in transit, preventing hackers from reading it as it is sent over the connection. This information could be anything sensitive or personal which can include credit card numbers and other financial information, names and addresses.

HTTPS (Hyper Text Transfer Protocol Secure) appears in the URL when a website is secured by an SSL certificate. The details of the certificate, including the issuing authority and the corporate name of the website owner, can be viewed by clicking on the lock symbol on the browser bar.

**Working**

An SSL certificate is necessary to create SSL connection. You would need to give all details about the identity of your website and your company as and when you choose to activate SSL on your web server. Following this, two cryptographic keys are created - a Private Key and a Public Key.

The next step is the submission of the CSR (Certificate Signing Request), which is a data file that contains your details as well as your Public Key. The CA (Certification Authority) would then validate your details. Following successful authentication of all details, you will be issued SSL certificate. The newly-issued SSL would be matched to your Private Key. From this point onwards, an encrypted link is established by your web server between your website and the customer's web browser.

On the apparent level, the presence of an SSL protocol and an encrypted session is indicated by the presence of the lock icon in the address bar. A click on the lock icon displays to a user/customer details about your SSL. It's to be remembered that SSL Certificates are issued to either companies or legally accountable individuals only after proper authentication.

**SSL Certificate**

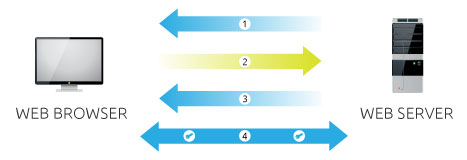
An SSL Certificate comprises of your domain name, the name of your company and other things like your address, your city, your state and your country. It would also show the expiration date of the SSL plus details of the issuing CA. Whenever a browser initiates a connection with a SSL secured website , it will first retrieve the site's SSL Certificate to check if it's still valid. It's also verified that the CA is one that the browser trusts, and also that the certificate is being used by the website for which it has been issued. If any of these checks fail, a warning will be displayed to the user, indicating that the website is not secured by a valid SSL certificate.

**Algorithms**

1. **ECC:** ECC provides stronger security and increased performance: it offers better protection than currently adopted encryption methods, but uses shorter key lengths (e.g. 256bit ECC key provides the same level of security as 3,072 RSA key). Requires fewer server processing cycles, allowing for more simultaneous SSL connections and faster processing. ECC key lengths increase at a slower rate than other encryption method keys as security levels increase, potentially extending the life of your existing hardware.
2. **RSA:** RSA is based on the presumed difficulty of factoring large integers (integer factorization). Full decryption of an RSA ciphertext is thought to be infeasible on the assumption that no efficient algorithm exists for integer factorization. A user of RSA creates and then publishes the product of two large prime numbers, along with an auxiliary value, as their public key. The prime factors must be kept secret. Anyone can use the public key to encrypt a message, but only someone with knowledge of the prime factors can feasibly decode the message.
3. **Pre-shared Key Encryption:** Pre-shared key encryption (symmetric) uses algorithms like Twofish, AES, or Blowfish, to create keys—AES currently being the most popular. All of these encryption algorithms fall into two types: stream ciphers and block ciphers. Stream ciphers apply a cryptographic key and algorithm to each binary digit in a data stream, one bit at a time. Block ciphers apply a cryptographic key and algorithm to a block of data (for example, 64 sequential bits) as a group. Block ciphers are currently the most common symmetric encryption algorithm.

**Asymmetric and Symmetric Encryption**

Public Key Infrastructure (PKI) is the set of hardware, software, people, policies, and procedures that are needed to create, manage, distribute, use, store, and revoke digital certificates. PKI is also what binds keys with user identities by means of a Certificate Authority (CA). PKI uses a hybrid cryptosystem and benefits from using both types of encryption. For example, in SSL communications, the server’s SSL Certificate contains an asymmetric public and private key pair. The session key that the server and the browser create during the SSL Handshake is symmetric.



1. Server sends a copy of its asymmetric public key.
2. Browser creates a symmetric session key and encrypts it with the server's asymmetric public key. Then sends it to the server.
3. Server decrypts the encrypted session key using its asymmetric private key to get the symmetric session key.
4. Server and Browser now encrypt and decrypt all transmitted data with the symmetric session key. This allows for a secure channel because only the browser and the server know the symmetric session key, and the session key is only used for that session. If the browser was to connect to the same server the next day, a new session key would be created.

**SSL Handshake for RSA Key Exchange Algorithm**

**1.  Client Hello:** Information that the server needs to communicate with the client using SSL. This includes the SSL version number, cipher settings, session-specific data.

**2.  Server Hello:** Information that the server needs to communicate with the client using SSL. This includes the SSL version number, cipher settings, session-specific data.

**3.  Authentication and Pre-Master Secret:** Client authenticates the server certificate. (e.g. Common Name / Date / Issuer) Client (depending on the cipher) creates the pre-master secret for the session, encrypts with the server's public key and sends the encrypted pre-master secret to the server.

**4.  Decryption and Master Secret:** Server uses its private key to decrypt the pre-master secret. Both Server and Client perform steps to generate the master secret with the agreed cipher.

**5.  Encryption with Session Key:** Both client and server exchange messages to inform that future messages will be encrypted.

**Best Practices**

1. Use 2048-Bit Private Keys
2. Protect Private Keys
3. Ensure Sufficient Hostname Coverage
4. Certificates from a Reliable CA
5. Use Strong Certificate Signature Algorithms
6. Use Complete Certificate Chains
7. Use Secure Protocols
8. Use Secure Cipher Suites
9. Select best cipher suites.
10. Use forward secrecy
11. Use strong key exchange
12. Mitigate known problems