

Secure Your Server Through SSL

In the recent times, we have been using the Internet for many of our online transactions such as Online Ticket reservation, online banking etc. We also observe that at times the data / information are being hacked without the knowledge of the user. However, technology can be used to minimize the risk of such untoward incidents and ensure that the data is transmitted in a secure and safe way. This article focuses on the basics of Secured Socket Layer (SSL) and the implementation of the same.



SRINIVASA RAGHAVAN
Senior Technical Director
NIC Tamil Nadu
ks.raghavan@nic.in



SIVARAMA SELVAN
Senior Systems Analyst
NIC Tamil Nadu
ss.selvan@nic.in

Edited by **R. Gayatri**

SECURED Socket Layer (SSL) is a protocol designed by Netscape Communications to enable encrypted, authenticated communications across the Internet. SSL is the standard security technology for establishing an encrypted link between a web server and a browser. This link ensures that all data passed between the web server and client remains private and integral. SSL is an industry standard and is used by millions of websites for protecting online transactions with their customers. Various Versions of SSL are in place and the Current SSL Version is 3.

To be able to create an SSL connection a web server requires an SSL Certificate. When you choose to activate SSL on your web server you will be prompted to answer a number of questions about the identity of your website and the business or company. The web server then creates two cryptographic keys - a Private Key and a Public Key.

The Public Key need not be kept secret and is placed into a Certificate Signing Request (CSR) - a data file also containing these details. The CSR is to be submitted. During the SSL Certificate application process, the Certification Authority will validate the given details and issue an SSL Certificate containing the validated details and allowing the user to use the SSL. The web server will match the issued SSL Certificate to your Private Key. The web server will then be able to establish an encrypted link between the website and the customer's web browser.

SSL uses a cryptographic system

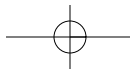
that uses two keys to encrypt data. The encryption using a private key/public key pair ensures that the data can be encrypted by public key but can only be decrypted by the other key pair, private key. By convention, URLs that require an SSL connection start with https (HTTP on SSL) instead of http (Hyper Text Transfer Protocol).

BENEFITS

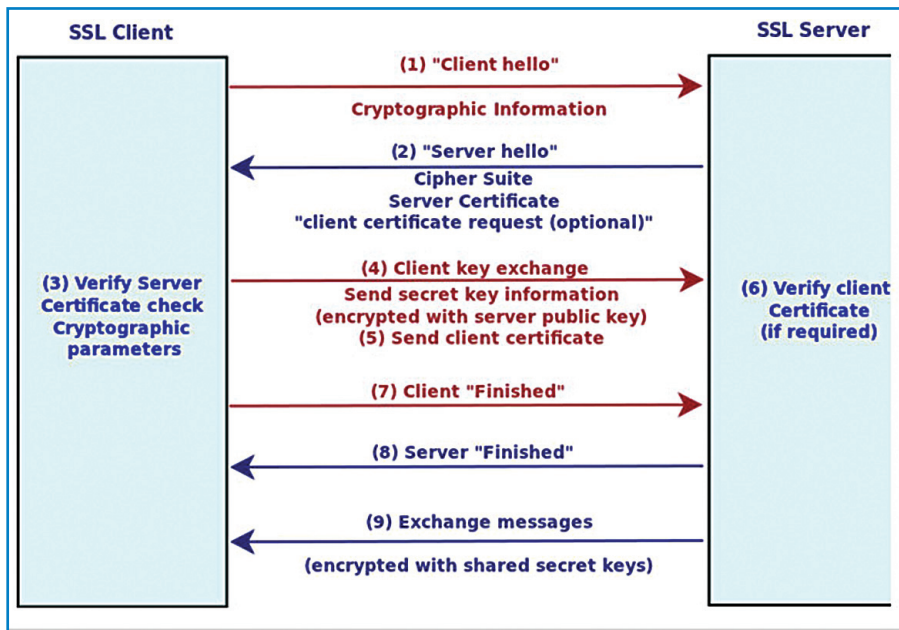
- Higher security: it ensures the security of data transmitted over the network.
- Support for various application layer protocols: SSL resides between the transport layer and the application layer, it provides security service for any application layer protocol that employs TCP connections. SSL runs above TCP/IP and below HTTP, LDAP, IMAP, NNTP, and other high-level network protocols.
- Simple to deploy: SSL has become a global standard for identity authentication between the client and server. It has been integrated into most browsers, such as IE, Netscape, Opera and Firefox. This means that almost every computer with a browser supports SSL connection, without requiring any extra client software.
- Identity authentication: SSL, supports certificate-based identity authentication of the server and client by using the Digital Certificates, with the authentication of the client being optional.
- Message integrity verification: SSL uses Message Authentication Code (MAC) algorithms to verify message integrity.

HOW IT WORKS

- A browser requests a secure page (usually https://).



Technology Update



- The web server sends its public key with its Digital Certificate.
- The browser checks that the certificate was issued by a trusted party (usually a trusted root CA), that the certificate is still valid and that the certificate is related to the site contacted.
- The browser generates a random symmetric encryption key which is to be used for encrypting data sent through the SSL channel. This key is encrypted using the public key of the server and sent to the server along with the https request.
- After exchanging the session key the client and the server exchange data by encrypting using the session key.
- The web server sends back the requested html document and http data encrypted with the symmetric

- key.
- The browser decrypts the http data and html document using the symmetric key and displays the information.

To enforce security uniformly, various standards have been enforced from time to time and following section describes the most relevant standards for encryption algorithms of the data.

SECURE HASH SIGNATURE STANDARD (SHS)

National Institute of Standards and Technology (NIST) has defined standards (Secure Hash Signature Standards - SHS) for making uniformity in hashing and encryption methods of data files or messages. This Standard specifies a Secure Hash Algorithm, SHA-1, for computing a condensed representation of a mes-

sage or a data file. When a message of any length < 264 bits is input, the SHA-1 produces a 160-bit output called a message digest. The message digest can then be input to the Digital Signature Algorithm (DSA) which generates or verifies the signature for the message. Signing the message digest rather than the message often improves the efficiency of the process because the message digest is usually much smaller in size than the message. The same hash algorithm must be used by the verifier of a digital signature as was used by the creator of the digital signature.

Secure Hash Signature Standard (SHS) specifies four secure hash algorithms, SHA-1, SHA-256, SHA-384, and SHA-512 for computing a condensed representation of electronic data (message). When a message of any length < 264 bits (for SHA-1 and SHA-256) or < 2128 bits (for SHA-384 and SHA-512) is input to an algorithm, the result is an output called a message digest. The message digests range in length from 160 to 512 bits, depending on the algorithm. Secure hash algorithms are typically used with other cryptographic algorithms, such as digital signature algorithms and keyed-hash message authentication codes, or in the generation of random numbers (bits).


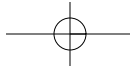
The **SHA** is one of a number of cryptographic hash functions published by the National Institute of Standards and Technology as a U.S. Federal Information Processing Standard.

contd. on next page

Salute to Sh. Mahendra Singh, DIO, NIC Kutch for Saving Two Young Lives !!

On 22nd September, 2011 evening around 7.15 pm, Sh. Mahendra Singh, DIO, NIC Kutch, Gujarat was passing near Hamirsar Lake in the heart of Bhuj city with his daughter Jahnavi. There was a huge crowd near the lake. Curiously, approaching there he saw two boys aged 13 and 15 drowning in the deep water and crying for help. Without any hesitation, Sh. Singh jumped into the lake and made utmost efforts to save the lives of the boys. After great effort, he saved the boys.

All NICians are very proud of him !!

SHA 1: A 160-bit hash function which resembles the earlier MD5 algorithm. This was designed by the National Security Agency (NSA) to be part of the Digital Signature Algorithm.

SHA 2: A family of two similar hash functions, with different block sizes, known as SHA-256 and SHA-512. They differ in the word size; SHA-256 uses 32-byte (256 bits) words whereas SHA-512 uses 64-byte (512 bits) words. SHA-2 includes a significant number of changes from its predecessor, SHA-1. SHA-2 consists of a set of four hash functions with digests that are 224, 256, 384 or 512 bits. The following table lists various parameters associated with different SHA algorithms.

CIPHER

A cipher (or cypher) is an algorithm for performing encryption or decryption, a series of well-defined steps that can be followed as a procedure. An alternative term is encipherment.

When using a cipher the original information is known as plaintext, and the encrypted form as ciphertext. The cipher-text message contains all the information of the plain-text message, but is not in a format readable by a human or computer without the proper mechanism to decrypt it.

Types of Ciphers: Ciphers can be classified by two criteria- by type of key used, and by type of input data.

By type of key used ciphers are divided into:

- **Symmetric key algorithms (Private-key cryptography),** where the same key is used for encryption and decryption, i.e., in the symmetric key algorithm (e.g., DES and AES), the sender and receiver must have a shared key set up in advance and kept secret from all other parties; the sender uses

this key for encryption, and the receiver uses the same key for decryption.

- **Asymmetric key algorithms (Public-key cryptography),** where two different keys are used for encryption and decryption, i.e., in the asymmetric key algorithm (e.g., RSA), there are two separate keys: a public key is published and enables any sender to perform encryption, while a private key is kept secret by the receiver and enables only the receiver to perform correct decryption.

By the type of input data are divided into:

- **Block ciphers** - which encrypt block of data of fixed size, and
- **Stream ciphers** - which encrypt continuous streams of data

The strength of a cipher is graded as Anonymous, Weak and Strong based on the number of bits used for encrypting the data.

The web servers such as Apache, Apache-Tomcat, IIS facilitate the use of SSL through the configuration files. By default, the web servers allow all the ciphers.

Strengthening Ciphers /

Disabling weak ciphers: In order to strengthen the data that is encrypted, ciphers such as anonymous, weak and null ciphers have to be disabled. Ciphers with 128 bit encryption and higher should be allowed. This will ensure that the weak ciphers are blocked and only strong ciphers are enforced during the data encryption during transmission.

DIGITAL SIGNATURE CERTIFICATE

SSL Certificate: The SSL certificate gives confidence to the user that the user is working in the correct site and not on a spoofed site.

SSL certificate is a unique credential identifying the certificate owner. An SSL certificate contains

information about the owner of the certificate, like e-Mail address, Owner's name, Certificate Usage, duration of validity, resource location or Distinguished Name (DN) which includes the Common Name (CN) (Name of the person / Web site address) and the certificate ID of the person who certifies (signs) this information. A certificate Authority such as NIC-CA under CCA, India authenticates the identity of the certificate owner before it is issued.

The certificate contains the reference to the issuer, the public key of the owner of this certificate, the date of validity of this certificate and the signature of the certificate to ensure this certificate hasn't been tampered with. The certificate does not contain the private keys as it should never be transmitted in any format.

X.509 Standard: X.509 is a standard in public key infrastructure (PKI). X.509 specifies standard formats for public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithm.

In the X.509 system, a certification authority issues a certificate binding a public key to a particular distinguished name or to an alternative name such as an e-mail address or a DNS-entry.

Certificate formats: Digital Signature Certificates are available in various formats say PEM, DER, P7B and PFX.

- **PEM format:** The PEM format is the most common format that Certificate Authorities issue certificates in this format. ".PEM" certificates usually have extensions such as .pem, .crt, .cer and .key. They are Base 64 encoded ASCII files and contain "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----" statements. Server certificates, intermediate certificates, and private keys can all be

put into the PEM format.

- **DER format:** The Distinguished Encoding Rules (DER) format is simply a binary form of a certificate instead of the ASCII PEM format. All types of certificates and private keys can be encoded in DER format. DER is typically used with Java platforms.
- **P7B format:** The PKCS#7 or P7B format is usually stored in Base64 ASCII format and has a file extension of .p7b or .p7c. P7B certificates contain "-----BEGIN PKCS7-----" and "-----END PKCS7-----" statements. A P7B file only contains certificates and chain certificates, not the private key.
- **PKCS12/PFX:** The PKCS#12 or PFX format is a binary format for storing the server certificate, any intermediate certificates, and the private key in one encrypt-able file. PFX files usually have extensions such as .pfx and .p12.

Certificate Trust Chain: A Certificate Authority (say NIC-CA) can issue multiple certificates in the form of a tree structure. A root certificate is the top-most certificate of the tree, the private key of which is used to "sign" other certificates. The top most Certifying Authority in India is the Controller of Certifying Authority (<http://cca.gov.in>). All certificates immediately below the root certificate inherit the trustworthiness of the root certificate - a signature by a root certificate.

The Chain of Trust of a Certificate Chain is an ordered list of certificates, containing an end user subscriber certificate, intermediate certificates and then the CA certificate that enables the receiver to verify that the sender and all intermediates certificates are trustworthy.

Certificate Revocation List (CRL): The Certificate Revocation List (CRL) is a periodically issued list of digital signature certificates that have been suspended or revoked prior to their

expiration dates due to loss or damage or theft of private key associated with the Digital Signature Certificates.

The CRL is exactly what its name implies: A list of subscribers paired with digital certificate status. The list enumerates the serial number of the revoked certificates along with the reason(s) for revocation. The dates of certificate issue, and the entities that issued them, are also included. In addition, each list contains a proposed date for the next release. When a potential user attempts to access a server, the server allows or denies access based on the CRL entry for that particular user.

CONFIGURING SSL IN TOMCAT

In order to strengthen the data during transmission, the certificate has to be generated locally in the server using a tool and the generated certificate will be submitted to CA for authorization. The CA signed server certificate is used for configuring http over SSL. One of the common tools to generate server certificate is OpenSSL.

OpenSSL: The OpenSSL program is a command line tool for using the various cryptography function of OpenSSL's crypto library. This is automatically bundled in all the latest version of Linux. It can be used for

- Creation and management of private keys, public keys and parameters
- Public key cryptographic operations
- Creation of X.509 certificates, CSR and CRL's
- Calculations of Message Digests
- SSL/TLS client and server tests
- Handling of S/MIME signed or encrypted mail

Generation of Key pairs: A key pair is generated normally using OpenSSL or similar programs and the same is stored in a key file.

Generation of Certificate Signing Request (CSR): The generated certificate is trusted after it is validated by the CA. Hence the Certificate request should be generated and sent to the Certifying Authority.

Sending Certificate request to Certifying Authority (CA): The generated certificate should be sent to the Certifying Authority which will be digitally signed and sent back to the user in .cer format.

Convert .cer file to PKCS#12 format: The .cer file is transferred to the concerned server, and converted into PKCS#12 format.

Installing the certificate in the Tomcat server: This file can be customized in Tomcat and after restarting the tomcat, the server can render the pages in SSL mode.

EMERGING STANDARDS

SSL has recently been succeeded by Transport Layer Security (TLS) which is based on SSL. SSL uses a program layer located between the Internet's HTTP and TCP layers. TLS and SSL are not interoperable. However, a message sent with TLS can be handled by a client that handles SSL but not TLS. The current version of TLS is 1.2.

TLS has a variety of security measures.

The major ones are listed below:

- Protection against a downgrade of the protocol to a previous (less secure) version or a weaker cipher suite.
- Numbering subsequent Application records with a sequence number and using this sequence number in the message authentication codes (MACs).
- Using a message digest enhanced with a key (so only a key-holder can check the MAC).
- TLS uses an enhanced HMAC over SSL hash-based MAC.