



OpenSSL : An Overview

*Prepared By
Visolve SSL Team*

| [OpenSSL Introduction](#) | [SSL/TLS Overview](#) | [OpenSSL Overview](#) | [Cryptography](#) | [Cryptography Overview](#) | [Symmetric Key Encryption](#) | [Asymmetric Public Key Encryption](#) | [Digests](#) | [Certificates](#) | [SSL Protocol](#) | [SSL Protocol Overview](#) | [SSL Handshake](#) | [SSL Data Transfer](#) | [OpenSSL Toolkit](#) | [Library](#) | [Command Line Tool](#) | [Generating Certificates](#) | [Conclusion](#) | [Glossary](#) | [About Visolve.com](#) |

OpenSSL Introduction

SSL/TLS Overview

This document introduces the Secure Socket Layer (SSL) protocol and the OpenSSL library/toolkit. Secure Socket Layer is the most widely used protocol for secure network communication. The SSL protocol provides the following services for network applications:

Data privacy: Client/server session is encrypted
Client authentication: Server can verify the client identity
Server authentication: Client can verify the server identity
Message integrity: Data cannot be modified during transmission

SSL can be used to secure any network protocol that uses a single TCP connection. For

example, the HTTPS protocol uses SSL to provide secure HTTP.

The Transport Layer Security (TLS) protocol is simply a revision of the SSL protocol. It supports more cryptographic algorithms.

OpenSSL Overview

OpenSSL is a popular Open Source implementation of the SSL/TLS protocols. The project is managed by a worldwide community of volunteers. OpenSSL is the only free, full-featured SSL implementation currently available for use with the C and C++ programming languages. It works across most major platforms, including Microsoft Windows and Unix OSs.

Cryptography

Cryptography Overview

Cryptography is the formal term for transforming a message so that only the intended receiver can read the message. Cryptography is also used to verify the sender of a message, and to verify a message has not been modified during transmission.

SSL uses various cryptographic algorithms to ensure secure communication. There are four main cryptographic concepts used by SSL:

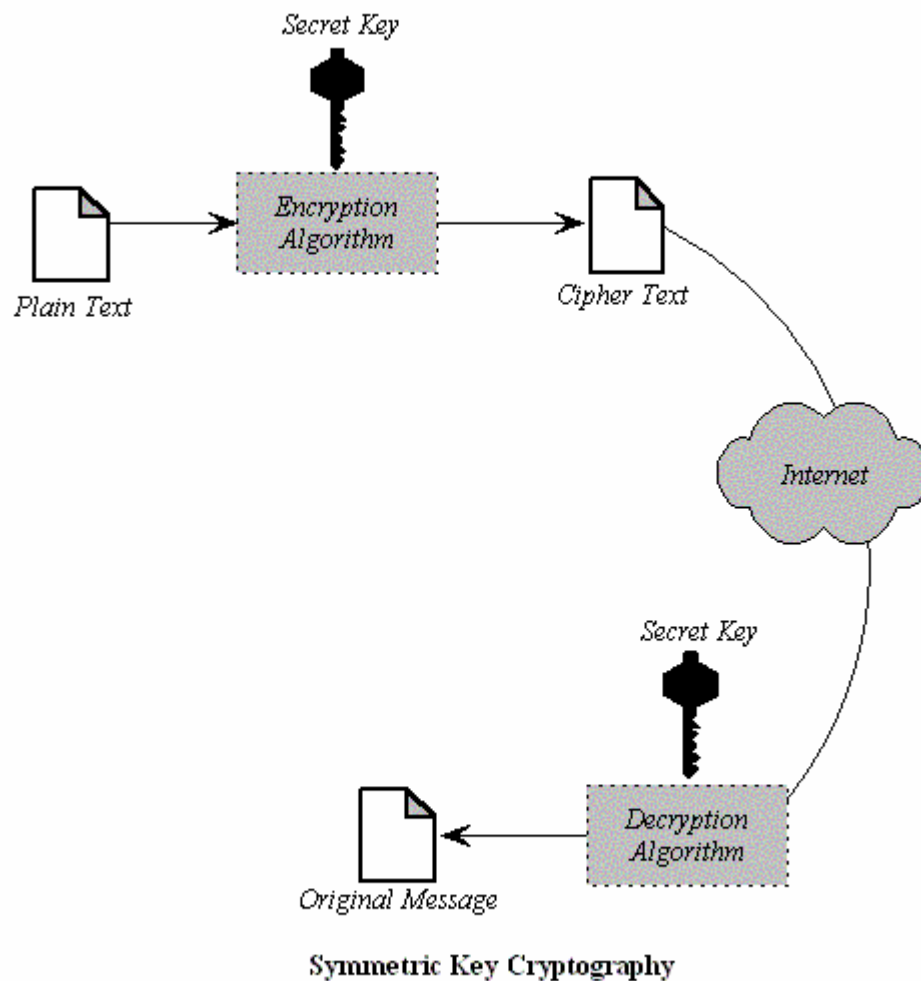
- Symmetric key (secret key) encryption
- Asymmetric key ([public key](#)) encryption
- Message Digests and digital signatures
- [Certificates](#)

Symmetric key (secret key) encryption

Symmetric key encryption uses a single key for both encrypting and decrypting data. As shown in the figure below the plain text message is passed through the encryption algorithm producing [ciphertext](#), which is unreadable, and therefore secure. The result is then sent over the insecure medium to the intended recipient. The recipient decrypts the message back to the [plaintext](#) using the same key.

Symmetric [ciphers](#) come in two types: block ciphers and stream ciphers. Block ciphers are traditionally the most popular. They operate by breaking up data into fixed-size blocks, and then encrypting each block individually. Leftover data is padded so that the length of the [plaintext](#) is a multiple of the cipher's block size. In contrast, stream ciphers are cryptographic pseudorandom number generators. They use a starting seed as a key to produce a stream of random bits known as the keystream. To encrypt data, one takes the plaintext and simply XORs it with the keystream.

The security of symmetric key encryption depends on the size of the key. The longer the key length, the more difficult for an intruder to break the encryption. However, longer keys take more time to decrypt for the recipient as well, and can lead to slight performance degradation.



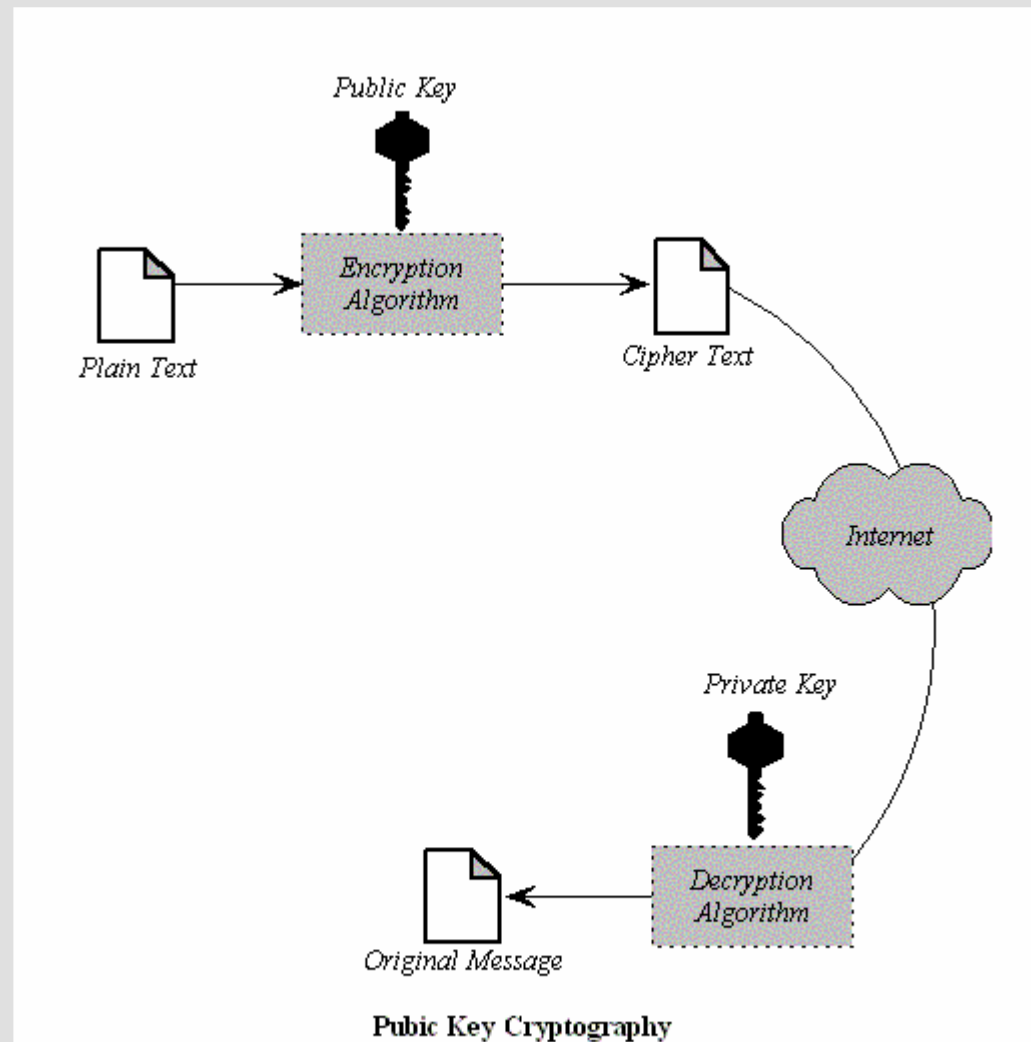
Asymmetric key (public key) encryption

In asymmetric encryption, a key pair, consisting of a [public key](#) and private key, is used to encrypt and decrypt data. The public key encrypts, but cannot be used to decrypt. Only the private key can decrypt the data. In addition, the private key can be used to encrypt the data, and the public key can be used to decrypt the data.

One major problem with private key encryption is how two machines decide on a private key securely. For ecommerce sites like amazon.com, it is not possible to assign private keys to every possible customer beforehand. Public key cryptography solves this problem. Any client can send encrypted data to a server using the server's public key. So any client can communicate securely with the server machine.

Secret key decryption is much faster to execute on a computer than public key decryption. In practice, they are often used together, so that a public-key is used to

encrypt a randomly generated secret key, and the secret key is used to encrypt the actual message. This is called hybrid encryption, and is used by the SSL protocol.



Digests

Message digests are used to ensure that a message is valid and has not been modified during transmission. A digest is a short, fixed-length summary of a long message, usually around 128 bits. The digest is created by applying a hash function on the original message. It is extremely difficult to find two messages which create the same digest.

Both the message and digest are encrypted and sent to the recipient. After decryption, the recipient computes the digest of the message and compares it with the digest received to ensure the integrity of the message. If an intruder modifies the encrypted data during transmission, it is likely that the decrypted data will not have a valid digest.

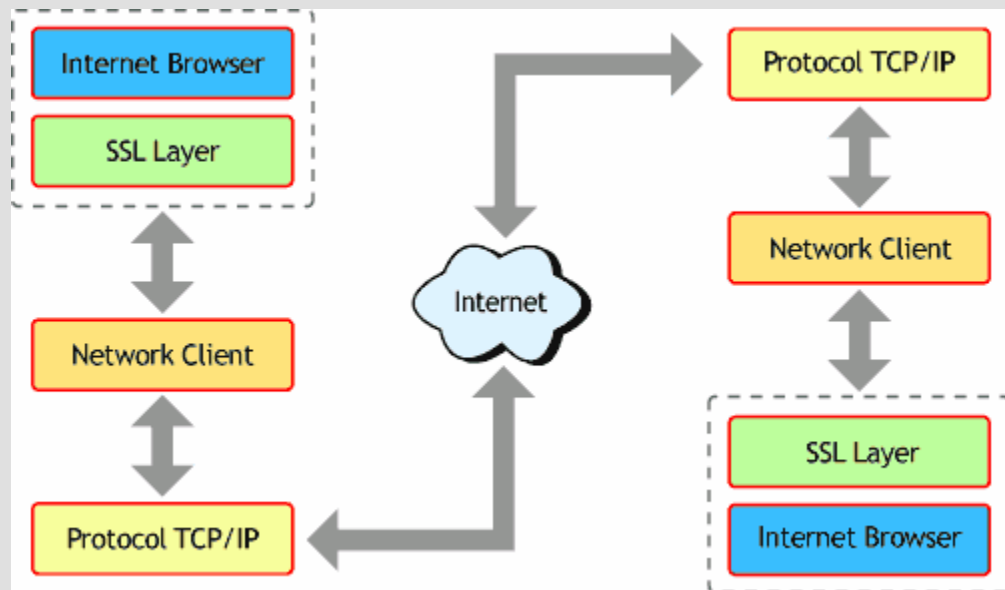
<p>Certificates</p>	<p>A certificate is a file used to securely identify a machine. A certificate includes the following information:</p> <ul style="list-style-type: none"> machine name organization/company location of machine (city, state, and country) time frame where certificate is valid public key/private key for secure communication Certificate authority name Certificate authority signature <p>The public key allows this machine to securely communicate with other machines. The company and location give information about the machine.</p> <p>A Certificate Authority is a company that issues and signs certificates. The Certificate Authority verifies that the information in the certificate (name, Company, etc) is correct. A certificate can also be self-signed, where is it not signed by any Certificate Authority. However, in practice, only certificates that are signed by trusted companies are trusted by Internet users.</p> <p>OpenSSL includes tools to create and manage certificates and public/private keys.</p>
----------------------------	--

<p>SSL protocol</p>	<p>SSL protocol overview</p> <p>The Secure Sockets Layer Handshake Protocol usually abbreviated as SSL is an excellent protocol capable of securing any protocol that works over TCP. And is short to say, SSL is recognized as the bottom line in security, which stands for securing the communications between servers-to-client and server-to-server, load balancing devices.</p> <p>The protocol is composed of two layers.</p> <ul style="list-style-type: none"> • SSL Record Protocol is layered on top of some reliable transport protocol (e.g., TCP). The SSL Record Protocol is used for encapsulation of various higher-level protocols. • SSL Handshake Protocol allows the server and client to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before the application protocol transmits or receives its first byte of data. <p>The SSL protocol provides connection security that has three basic properties:</p> <ul style="list-style-type: none"> • The connection is private. Encryption is used after an initial handshake to define a secret key. Symmetric cryptography is used for data encryption (e.g., DES) • The peer's identity can be authenticated using asymmetric, or public key, cryptography (e.g., RSA, DSS , etc.). • The connection is reliable. Message transport includes a message integrity check using a keyed MAC. Secure hash functions (e.g., SHA, MD5, etc.) are used for MAC computations.
----------------------------	--

SSL is a layered protocol. At each layer, messages may include fields for length, description, and content. SSL takes messages to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, and transmits the result. Received data is decrypted, verified, decompressed, and reassembled, then delivered to higher level clients.

In a typical Internet scenario, the browser will incorporate an SSL layer. This will allow data that is sent from the client to become encrypted before being sent to the network client. Once the data leaves the network client, it is converted to a network protocol (typically TCP/IP) and then sent through the Internet to the appropriate server. When the server receives the data, the server's SSL protocol layer decrypts the data and passes it to the proper server application for processing. When the server sends a reply back to the client the process is repeated - only in reverse.

SSL is most commonly used for transmitting sensitive information, such as credit card numbers and passwords, and as an authentication mechanism. SSL is the standard protocol in use today for secure communication over the Internet. SSL servers are typically used for Web hosts, while the SSL clients are typically Web browsers.



SSL Handshake

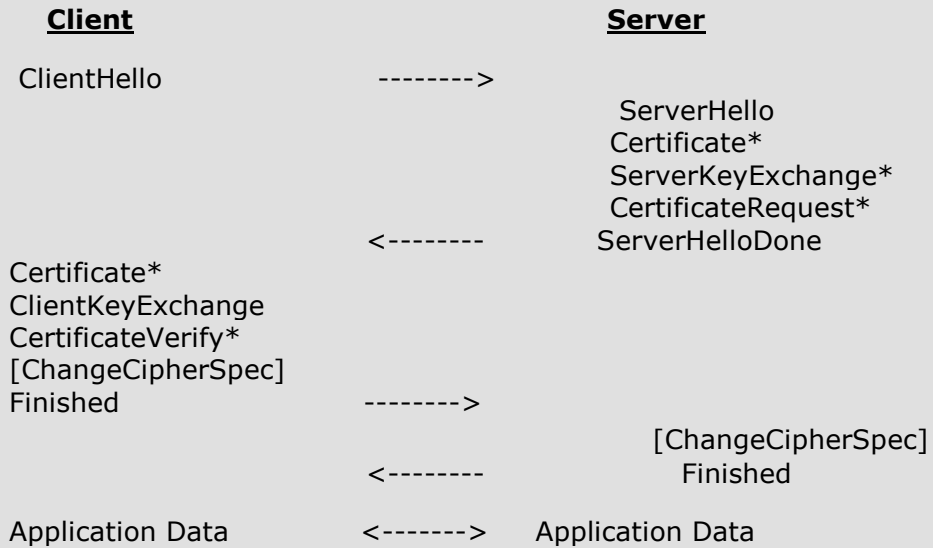
When an SSL client initially connects to an SSL server, they perform a handshake to establish security attributes and exchange certificates. The client and server exchange "hello" messages which establish:

- Protocol version
- Session ID
- Cipher Suite
- Compression method

Following the hello messages, the server and client exchange their certificates. The

client will also send a key exchange method, to determine the type of session key to use for encryption.

The flow chart below summarizes the initial SSL handshake:



* Indicates optional or situation-dependent messages that are not always sent.

SSL Data Transfer

Once the SSL handshake is completed, a session key is used by the client and server to encrypt/decrypt all network traffic between them. SSL acts as another network layer on top of TCP. The SSL layer performs the following actions to the plaintext data before sending:

- fragments the data into blocks
- Optionally compresses the data
- Appends a Message Authentication Code (MAC)
- Encrypts the data

After receiving data, SSL performs the same operations in reverse:

- Decrypts the data
- Validates the MAC is correct
- Optionally decompresses the data
- Reconstructs the blocks into the message

OpenSSL toolkit

Library

OpenSSL provides a general purpose cryptography library (libcrypto.a) and Secure Sockets Layer library (libssl.a) written in C. Applications written in C can use the

OpenSSL libraries for either client or server side SSL encryption.

Command line tool

The openssl program is a command line tool for using the various cryptography functions of OpenSSL as crypto library from the shell. It can be used for

- Creation of RSA, DH and DSA key parameters
- Creation of [X.509](#) certificates, CSRs and CRLs
- Calculation of Message Digests
- Encryption and Decryption with Ciphers
- SSL/TLS Client and Server Tests
- Handling of S/MIME signed or encrypted mail

The openssl program provides a rich variety of commands, each of which often has a wealth of options and arguments. The pseudo-commands output a list (one entry per line) of the names of all standard commands, message digest commands, or cipher commands, respectively, that are available in the present openssl utility.

Generating certificates

The OpenSSL toolkit is often used to create and manage certificates for SSL enabled applications. To create a standard certificate request, use the following command:

```
# openssl req -new -newkey rsa:<bits> -out <filename> -keyout <filename>
```

For example

```
# openssl req -new -newkey rsa:1024 -out mycert.csr -keyout mykey.pem
```

The arguments have the following meaning:

- req : perform a certificate operation
- new : create a new certificate request
- newkey : create a new RSA public/private key pair, with the given number of bits
- out : the file to write the certificate request to.
- keyout : the file to write the RSA private key to.

OpenSSL will prompt for the following information while creating your certificate. These values will be stored in the certificate:

- PEM pass phrase: a password that grants access to your private key.
- Country Name (2 letter code)
- State
- City
- Organization/Company
- Organizational Unit
- Common Name

A certificate request is not complete until it is signed by a Certificate Authority. To create a self signed certificate, use the command:

```
# openssl req -new -x509 -newkey rsa:<bits> -out <filename> -keyout <filename>
-days <number>
```

For example

```
# openssl req -new -x509 -newkey rsa:1024 -out mycert.csr -keyout mykey.pem -
days 365
```

The parameter -x509 indicates a self-signed certificate. The parameter -days indicates the number of days the certificate is valid for. For more information on the openssl tool, see the openssl man page.

Conclusion

Secure Socket Layer is the most common tool used for secure network communication. It is used especially in transmitting sensitive information, such as credit card numbers and passwords. OpenSSL is the most popular library for writing SSL applications. OpenSSL is an essential tool for system and application administrators to learn in order to develop a secure network environment.

Glossary

Certificate

A data record used for authenticating network entities such as a server or a client. A certificate contains X.509 information pieces about its owner (called the subject) and the signing Certificate Authority (called the issuer), plus the owner's public key and the signature made by the CA. Network entities verify these signatures using CA certificates.

Certification Authority (CA)

A trusted third party whose purpose is to sign certificates for network entities it has authenticated using secure means. Other network entities can check the signature to verify that a CA has authenticated the bearer of a certificate.

Cipher

An algorithm or system for data encryption. Examples are DES, IDEA, RC4, etc.

Ciphertext

The result after a Plaintext passed a Cipher.

Plaintext

The unencrypted text.

Private Key

The secret key in a Public Key Cryptography system, used to decrypt incoming

messages and sign outgoing ones. The private key is never distributed; it is always kept secret. The public key is made public by distributing it widely.

Public Key

The publically available key in a Public Key Cryptography system, used to encrypt messages bound for its owner and to decrypt signatures made by its owner.

X.509

An authentication certificate scheme recommended by the International Telecommunication Union (ITU-T) which is used for SSL/TLS authentication.

About ViSolve.com

ViSolve is an international corporation that provides technical services, for Internet based systems, for clients around the globe. ViSolve is in the business of providing software solutions since 1995. We have experience of executing several major projects and we are now completely focused on leading Internet technologies, Testing QA and support. We are committed to the Open source movement and in the same lines we provide free support for products like Linux, Apache and Squid to the user community.