

***Software Engineering 4C03  
Research Project***

***An Overview of Secure  
Transmission on the  
World Wide Web***

***Sean MacDonald  
0043306***

***Tuesday April 5, 2005***

**Software Engineering 4C03  
Research Project**

***An Overview of Secure Transmission on the World Wide Web***

**Sean MacDonald  
0043306**

***Submitted on Tuesday April 5, 2005  
Last Revised on Monday April 4, 2005***

**Introduction**

The question of Internet security is one that is becoming increasingly prevalent in the world of computing today. The number of online transactions taking place each day is increasing at an astronomical rate, and people, whether they are doing their banking, making online purchases or simply checking their email, take comfort in the fact that the information they are transmitting over the Internet is protected. If a system of protection was not in effect, individuals and corporations would most certainly incur monetary loss as well as compromise confidential information.

**How can you protect your information?**

The use of encryption is a primary means of protecting information that is transferred on the Internet. Encryption defined, is the process of obscuring information to make it unreadable without special knowledge. In this case, it is desired that the data being transferred be readable by the source and destination computer systems that the data is being transferred between, but not to any external observers. The special knowledge that must be known to read the message being transferred, comes in the form of a key. This key is used with a special function, and when applied to the encrypted message, decrypts it.

There are two types of data encryption that are commonly used today. The first method, symmetric cryptography, uses the same key for both the encryption and decryption of a message. Both the client and server systems must "know" each other, and have agreed upon the key that is going to be used in advance. This method requires minimal CPU cycles in utilizing the key, but an inherent weakness in the system is introduced due to the fact that the key must be distributed in advance, outside of the actual cryptography system. The second method, asymmetric cryptography, uses one key for encrypting data, and a second key for decrypting data. The use of this method is advantageous as even if the key used for encryption is discovered by an external source, the key used for decryption must still be discovered if a message is to be compromised. This method does have its disadvantages however, as its performance is up to 1000 times more CPU intensive than symmetric cryptography, and therefore, inherently slower.

Each of the cryptography systems described above has its own merits, but the user should not have to manually go through the process of utilizing either of them. This brings us to the Secure Sockets Layer Protocol, or SSL, which makes use of both cryptography system to transfer data securely over the Internet.

## What is the Secure Sockets Layer Protocol?

The Secure Sockets Layer Protocol, or SSL, was a system developed by Netscape Communications to provide security for Internet data traffic. The three primary objectives for this system are:

1. Authenticating client and server systems to each other.
2. Securing data privacy, such that a message can be sent between client and server systems and only be readable by the intended recipient.
3. Ensuring data integrity, such that data cannot be tampered with while it is being transmitted.

The SSL protocol was designed such that it could be integrated with the TCP/IP protocol suite. The communications tasks of this protocol suite are divided into five independent layers, with these layers being the Application Layer, the Transport Layer, the Internet Layer, the Network Interface Layer and the Physical Layer. The SSL Protocol is integrated between the Transport Layer and the Application Layer which allows virtually any application to make use of it. Through making use of the TCP/IP protocol suite, SSL provides a reliable and secure end-to-end service.

Simply stated, the SSL protocol uses asymmetric cryptography, which is inherently slow, to encrypt a generated symmetric key, which is then distributed to both the client and server systems. This symmetric key is then decrypted and used for encrypting all other data that is sent along the connection between the client and server systems. How this is done specifically, will be explained in the following section.

## How does the Secure Sockets Layer Protocol work?

The SSL protocol can be thought of as not a single protocol, but as a set of protocols that can be divided into low level and high level layers. The low level layer contains only the Record Protocol, which is used to provide basic security services to the protocols in the high level layer. The high level layer contains three protocols, the Handshake Protocol, the Change CipherSpec Protocol and the Alert Protocol, all of which are used in the management of SSL data transmissions. The use of both layers of protocols is required to create and maintain a connection between the client and server systems. Each of these protocols will be explained in the following section.

When it is initially desired that a secure connection be made, the Handshake Protocol is used to initiate a session between the client and server systems, where a session is defined as a logical peer-to-peer connection between two nodes. When this takes place, the client system sends the server system a *client\_hello* message which contains the following information:

Version	The version of SSL that is supported by the client system.
Random Data	Random data used to protect the key exchange between the client and server systems.
Session ID	A unique identifier for the data transmission session.

Cipher Suite	A list of encryption algorithms and key exchange methods supported by the client system.
--------------	--

In response to the *client\_hello* message, the server issues a similar *server\_hello* message which contains the following information:

Version	The version of SSL that is supported by the server system.
Random Data	Random data used to protect the key exchange between the client and server systems. The random data generated by the server system is in no way related to the random data generated by the client system.
Session ID	The unique identifier for the data transmission session as sent by the client system.
Cipher Suite	A list of encryption algorithms and key exchange methods that will be used throughout the data transmission. This list is created by the server based on the corresponding list that it has received from the client.

After the server completes the sending of the *server\_hello* message, it sends its digital certificate to the client system for authentication. The purpose of the digital certificate is to attest to the identity of an individual or other entity and allow verification that an individual or entity is in fact who it claims to be. SSL specifically makes use of X.509 certificates for identify verification, which should be issued by a trusted third party organization to verify that the certificate is from a reputable individual or entity. An X.509 contains information such as the issuer of the certificate, the start and end validity dates of the certificate and the public key of the owner of the certificate, as well as the algorithm used to generate the public key. Additionally, the public key of the owner is encrypted using the private key of the trusted third party organization. Following the transmission of this certificate, the *server\_hello* message will be complete.

When the client computer system receives the *server\_hello* message, it first examines the digital certificate that it has received. It decrypts the encrypted public key it has received, using the public key of the trusted third party. If the client system is able to successfully verify the identity of the system, it will then send a 384 bit premaster key to the server system, which when functionally combined with the Random Data generated by both the client and server systems, forms the symmetric encryption key that will be used for all further data encryption and decryption.

After the symmetric key has been transmitted, the host and client systems each make use of the ChangeCipher Spec Protocol, which consists of a single message that carries the value of 1. This message allows the pending connection to be formally established. After the host and client systems have received this message from the other, the subsequent application data transmission takes place.

As with any computer system, errors are bound to happen. The Alert Protocol is therefore present and is used by both the client and server to indicate that errors have taken place. Each message used by the Alert Protocol is of length 2 bytes, where the first byte contains a value corresponding to either “warning” or “fatal”, depending on the severity of the error that has occurred. If a “warning” value is sent, the receiving party will determine the specific error that has taken place from the second byte of the message, and attempt to recover from it. If a “fatal” value is sent, the SSL connection will be terminated immediately.

The overall functioning of the SSL protocol is illustrated in the following diagram.

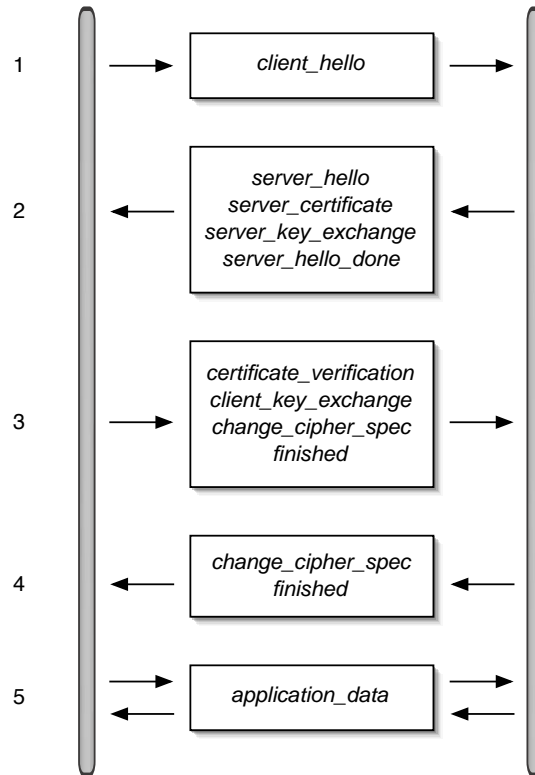


Figure 1 - Making an SSL Connection

The SSL Record Protocol, although mentioned above, was not explicitly discussed due to the length requirements of this paper. More information can be obtained in the listed references with respect to this topic.

**Will the use of the Secure Sockets Layer Protocol ensure data security?**

As in life, there are very few things that can be stated with absolute certainty. The SSL protocol is indeed not excluded from this, and the use of it will definitely not ensure data security. A determining factor in the overall security of an SSL implementation, is the length of the key that is used for encryption and decryption. A key with a shorter length is inherently more insecure than a key with a longer length. This was in fact the reason why the first version of SSL that was developed was never released, as it could only support 40 bit keys which are relatively easy to discover through the use of brute force methodologies. Modern implementations of SSL support keys with lengths of 128 and 256 bits,

