SSL and TLS: A Beginners Guide

Holly McKinley

Holly Lynne McKinley

GSEC Practical v.1.4b

SSL and TLS: A Beginners' Guide

ABSTRACT:

This practical serves to explain the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols, how they can be applied to a web application, and the requirements necessary to create a secure link between a server and a client machine. In addition, a development history of the protocols will be given, and a brief discussion of the impact that secure communications protocols have had on the electronic commerce arena.

This paper particularly serves as a resource to those who are new to the information assurance field, and provides an insight to two common protocols used in Internet security. Though SSL and TLS are not the only secure protocols currently in use, they are very common for sites dealing with transactions that could involve sensitive data (ie: passwords, personal and financial information, etc.).

INTRODUCTION:

As database driven applications are increasing their hold on the systems market, the security of the retained information is also increasing.

Databases such as Oracle, SQL Server and Access are being used to create more complex and intelligent systems and are used to establish a developer's command of the computer's capabilities. Just as the systems are being developed rapidly, unauthorized and malicious users are finding ways into these systems – and creating new ways to retrieve sensitive information. Since E-Government is a part of President Bush's management agenda, the confidentiality and integrity of information is more important than ever. Now that security is under the scrutiny of the public eye, security breaches are making headlines and the media has turned more attention towards information security and made breaches more prominent. This is not meant to say that hacking, viruses, and white-collar crimes involving computers did not exist for decades before President Bush's administration. In fact, the opposite is true. Dr. Alan Solomon of S&S International explains that computer viruses began in the mid 1980's with codes that were simple in comparison to today's malicious computer viruses. (Solomon).

As newly developed systems become more complex and interact with other systems, the aggregate of information they contain can be very sensitive. For this reason, the Federal Government has passed several legislations to protect the privacy of the data stored in those systems.

The Health Insurance Portability and Accountability Act (HIPAA) was passed in 1999 with the goal of reducing the government's costs of healthcare payments. HIPAA requires the protection of personally identifiable information and requires standard procedures to be created and followed for the execution of electronic transactions. More records are restricted with HIPAA regulations as increasing numbers of transactions are made that involve healthcare benefits. (Coleman, 2/9/03)

The Gramm-Leach-Bliley Act (GBLA) was also passed in 1999 to require the protection of consumers' financial records. Banking institutions must disclose privacy policies and allow consumers to opt-out of any information sharing in which the institution might participate. (Hiller, 81)

While these legislations include acts to protect the *data* stored in a system from malicious use, nothing is mentioned about protecting the system from attacks.

Examples of these attacks are: active wiretapping, masquerading and spoofing. These attacks can occur while information is being transmitted through the pathways of the Internet.

One way of mitigating a potential attack during a user's session would be to use a secure communication protocol to encrypt data in transit between the user and the server on which the sensitive information resides. Two of these communication protocols will be explained within this paper: Secure Sockets Layer (SSL) and Transport Layer Security (TLS).

Secure Sockets Layer Protocol

Definition of SSL

SSL is the secure communications protocol of choice for a large part of the Internet community. There are many applications of SSL in existence, since it is capable of securing any transmission over TCP. Secure HTTP, or HTTPS, is a familiar application of SSL in e-commerce or password transactions. (Viega, 10)

According to the Internet Draft of the SSL Protocol, the point of the protocol "is to provide privacy and reliability between two communicating applications." (Freier, 3.) The protocol release further explains that three points combine to provide connection security. These points are:

- Privacy - connection through encryption
- Identity authentication – identification through certificates, and
- Reliability –dependable maintenance of a secure connection through message integrity checking.

The current version of SSL is version 3.0, released by Netscape in 1999. The Internet Engineering Task Force (IETF) has created a similar protocol in an attempt to standardize SSL within the Internet community. This protocol, the Transport Layer Security (TLS) protocol, will be discussed later in this paper.

Using a series of nine messages (explained later), the server authenticates itself to a client that is transmitting information. Though it is a good idea for the user to hold a digital certificate, it is not required for the SSL connection to be established. Keep the following scenario in mind, as it shows a common application of SSL: A user without a certificate wishes to check her e-mail on a web-based e-mail system. Since she has requested a secure connection from the e-mail web page, she expects to send her username and password to the e-mail site. The identification of the e-mail server to her current workstation is critical. To the e-mail server though, it is not critical that the user has an identifying certificate on her machine because she can check her e-mail from any computer. For this reason, SSL does not require a client certificate.

Other practical applications of SSL communications are found in e-mail and financial transaction communications.

Application to a Web System

The need to send sensitive information over the Internet is increasing, and so is the necessity to secure information in transit through the Internet. A common application of SSL with a web system is an online store where a client machine is

sending a request to a merchant's server. In order to apply the SSL protocol to a web system, some requirements must be met. Since the SSL protocol is integrated into most web browsers, and those browsers are normally used to access web applications, no further configuration is required from the client's side of the SSL connection.

Configuration is relatively simple from the server side of the communication equation. First, the web server administrator must acquire a digital certificate. This can be obtained from a Certification Authority (CA) such as VeriSign or RSA Data Security. CAs require that certificates be renewed after a set length of time, as a mechanism for ensuring the identity of the owner of the application's server. (MSDN, 2/16/03).

The second requirement is the proper configuration of the web server to allow SSL connections**.** For example, the iPlanet Web Server has the capability to store multiple certificates for multiple sites on one web server. This capability allows the administrators to prove the identity of each application hosted by this server, and allows the application users to correctly identify each application separately.

The third piece of the puzzle is not necessarily a requirement, but a strong suggestion: to add an accelerator to the web server. SSL accelerators are PCI cards sold by several companies (Cisco, Broadcom, etc) to speed up the processing actions required to encrypt information for secure communications. There is a balance struck frequently between security and functionality, and this balance changes on a case-by-case basis. SSL connections do slow communications, mostly due to the exchanging of keys and other information during the startup phase of the session. The use of public key cryptography requires a "sizeable amount of information" to be passed between the client and server machines. (Viega,12). Though there are several ways to mitigate this issue, but the most commonly accepted strategy is to use an SSL accelerator.

How SSL Works

The four protocol layers of the SSL protocol (Record Layer, ChangeCipherSpec Protocol, Alert Protocol, and Handshake Protocol) encapsulate all communication between the client machine and the server.

Record Layer

The record layer formats the Alert, ChangeCipherSpec, Handshake and application protocol messages. This formatting provides a header for each message, and a hash, generated from a Message Authentication Code (MAC) at the end. The fields that comprise the five-byte header of the Record Layer are: Protocol Definition (1 byte), Protocol Version (2 bytes) and the Length (2 bytes). The protocol messages that follow the header cannot be longer than 16,384 bytes, as specified by the SSL protocol. (Thomas, 70)

ChangeCipherSpec Protocol

The ChangeCipherSpec layer is composed of one message that signals the beginning of secure communications between the client and server. Though the ChangeCipherSpec Protocol uses the Record Layer format, the actual ChangeCipherSpec message is only one byte long, and signals the change in communications protocol by having a value of '1'.

Alert Protocol

This protocol sends errors, problems or warnings about the connection between the two parties. This layer is formed with two fields: the Severity Level and Alert Description.

Severity Level

The Severity Level sends messages with a '1' or '2' value, depending on the level of concern. A message with a value of '1' is a cautionary or warning message, suggesting that the parties discontinue their session and reconnect using a new handshake. A message with a value of '2' is a fatal alert message, and requires that the parties discontinue their session.

Alert Description

The Alert Description field indicates the specific error that caused the Alert Message to be sent from a party. This field is one byte, mapped to one of twelve specific numbers, and can take on one of the following meanings. Those descriptions that always follow a "fatal" alert message are underlined. (Thomas, 73)

CloseNotify	HandshakeFailure	CertificateRevoked
UnexpectedMessage	NoCertificate	CertificateExpired
BadRecordMAC	BadCertificate	CertificateUnknown
DecompressionFailure	UnsupportedCertificate	IllegalParameter

Handshake Protocol

Messages passed back and forth between the user's browser (client) and web application (server) establish a handshake that begins a secure connection. The following steps are how a SSL handshake is performed. The messages that compose this handshake are: ClientHello, ServerHello, ServerKeyExchange, ServerHelloDone, ClientKeyExchange, ChangeCipherSpec, Finished, ChangeCipherSpec, Finished. (Thomas, 40) The following sections will detail these messages and, where appropriate, will explain how they are used in the webmail example seen earlier in this paper. A visual explanation of the Handshake Protocol is found in Figure 1.

ClientHello

The first message is the ClientHello. Since the client machine is requesting the secure communication session, this message involves a set of options that the client is willing to use in order to communicate with the server. The option categories are: Version of SSL to be used, CipherSuites supported by the client, and CompressionMethods used by the client. Other information that is included in this message is a 32-byte RandomNumber that assists the client in establishing encrypted communications, and a SessionID field that is blank. This message is generated by the client in the web e-mail example when our user wants to check her email and clicks on the "secure connection" option that is made available on many websites.

ServerHello

The second message of the SSL handshake is the ServerHello. In this message, the server makes choices based on the ClientHello message. The server returns five fields, just like the ClientHello message, but fills in the SessionID, and makes firm decisions on the Version of SSL to be used, the CompressionMethod and CipherSuite. The date and time stamp replaces four bytes of the RandomNumber field to avoid repeated random values, and Thomas adds that "the remaining bytes should be created by a cryptographically secure random number generator."(Thomas, 44).

ServerKeyExchange

Now that the server has made decisions for the transmission of data, information must be passed between the parties to determine how data will be encrypted. Since no algorithm has been previously agreed upon, this information is sent with no encryption. This means that all communication for this segment must already be in the public domain. The server's public key is used to encrypt a separate session key to be maintained for this secure communication. Both the client and server will use this same key to encrypt data to be transmitted.

To ensure that the communicating parties are who they claim to be, digital certificates are used to provide electronic identification. Digital certificates combine the public key and connect it to the name of the certificate owner. Additionally, these certificates contain public keys to certification authorities like RSA Security or VeriSign and an expiration date so that the person receiving the digital certificate can verify the link between the certificate owner and the certification authority. The certificate only contains the public key, and should never include the private key, else the private key would be compromised, and the entire purpose of having the digital certificate would be voided. (Martin, 3/14/03)

ServerHelloDone

Once the Server has completed the ServerKeyExchange message, the client receives a ServerHelloDone message to indicate that the server is through with its messages. It is similar to a two-way radio conversation when the sending party says "OVER" to announce that he is done sending a message, and signals the receiving party to acknowledge the message that was sent.

ClientKeyExchange

Since SSL does not require a client to have public and private keys in order to establish a SSL session, the ClientKeyExchange message contains information about the key that the client and server will use to communicate. Thomas explains that this is the point where the "man in the middle" attack is mitigated since a masquerader must know the server's private key in order to decrypt this message. (Thomas, 46) This message completes the negotiation processes between the client and the server.

ChangeCipherSpec

The two ChangeCipherSpec messages signal the change of data transmission from an insecure state to a secure state. As each computer sends the ChangeCipherSpec message, it changes its side of the connection into the agreed-upon secure state.

Finished

The two messages signaling the final messages of the SSL handshake ensure that three things are verified before the initial handshake is complete. These are:

- Key Information
- Contents of all previous SSL handshake messages exchanged by the systems
- A special value indicating whether the sender is a client or server (Thomas, 51)

At the end of this handshake process, the user will see a lock icon in the corner of her browser to indicate that a secure protocol has been agreed upon, and is in use by her browser and the web e-mail server.

Message Authentication

Once this information is checked, the communication can continue, appending a message authentication algorithm to the end of each message. Message Authentication is performed by using "an algorithm that uses cryptographic technology to create a digital summary of information so that if the information is altered, the summary (known as a hash) will also change." (Thomas, 186) MD5 and SHA are common hash functions used in SSL communications.

Resuming a Disconnected Session

If an Alert message disconnects a sessions before the parties are through communicating, that session can be resumed if the client sends a HelloRequest to the server with the properly encrypted SessionID information. The server then determines if the SessionID is valid, exchanges ChangeCipherSpec and Finished messages with the client machine, and secure communication can resume.

Development History

Netscape developed SSL version 1.0 in 1994 for the secure transmission of documents over the Internet. SSL 2.0 was developed about a year later, and was released with version 1.0 of Netscape Navigator. SSL 3.0 was released in 1999, and Netscape has allowed the IETF to take over the development of future versions. The name of future versions of the SSL protocol will be changed to TLS, with version numbers of the protocol beginning at 1.0. Since the Version numbers negotiated in the ClientHello and ServerHello messages of SSL are 3.0 and below, version numbers to be negotiated with TLS and future revisions will continue by negotiating as version 3.1 or higher. This will be done to denote a revision of SSL 3.0, but to promote backwards compatibility between clients and servers using either SSL or TLS.

Transport Layer Security Protocol

Definition of TLS

TLS was released in response to the Internet community's demands for a standardized protocol. The IETF provided a venue for the new protocol to be openly discussed, and encouraged developers to provide their input to the protocol.

The Transport Layer Security (TLS) protocol was released in January 1999 to create a standard for private communications. The protocol "allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering or message forgery." (Dierks, 1)

According to the protocol's creators, the goals of the TLS protocol are cryptographic security, interoperability, extensibility, and relative efficiency. (Dierks, 4) These goals are achieved through implementation of the TLS protocol on two levels: the TLS Record protocol and the TLS Handshake protocol.

TLS Record Protocol

The TLS Record protocol negotiates a private, reliable connection between the client and the server. Though the Record protocol can be used without encryption, it uses symmetric cryptography keys, to ensure a private connection. This connection is secured through the use of hash functions generated by using a Message Authentication Code.

TLS Handshake Protocol

The TLS Handshake protocol allows authenticated communication to commence between the server and client. This protocol allows the client and server to speak the same language, allowing them to agree upon an encryption algorithm and encryption keys before the selected application protocol begins to send data. (Dierks, 3).

Using the same handshake protocol procedure as SSL, TLS provides for authentication of the server, and optionally, the client. Several changes were made to the handshake protocol, and those will be discussed in a later section. (MSDN,2/16/03).

Comparison of SSL and TLS

Stephen Thomas explains that there are seven main differences between SSL and TLS. These differences range from protocol version numbers to the generation of key material. (118)

Protocol Version in Messages

To differentiate TLS Version 1.0 and SSL Version 3.0, the protocol version number negotiated by a client and server communicating through TLS Version 1, is version number 3.1,

Alert Protocol Message Types

The following message types are those that are allowed as Alert Descriptions within the TLS protocol. Upon examination of the list, one would notice that "NoCertificate" has been removed from the SSL list, since it is assumed that if no certificate exists for the user, there is no need for a separate message. TLS uses the

assumption that the client can return an empty certificate message if it does not have a certificate to use.

Additionally, several more descriptions have been added to bring the number of Alert Descriptions to 23 from 12. A list of these descriptions is below. Again, those resulting in fatal errors are underlined. (Thomas, 119-120).

CloseNotify	UnsupportedCertificate	DecryptError
UnexpectedMessage	CertificateRevoked	ExportRestriction
BadRecordMAC	CertificateExpired	ProtocolVersion
DecryptionFailure	CertificateUnknown	InsufficientSecurity
RecordOverflow	IllegalParameter	InternalError
DecompressionFailure	UnknownCA	UserCancelled
HandshakeFailure	AccessDenied	NoRenegotiation
BadCertificate	DecodeError	

Message Authentication

TLS implements a standardized MAC (H-MAC) that has been proven in many other implementations. The main benefit to this change is that H-MAC operates with any hash function, not just MD5 or SHA, as explicitly stated by the SSL protocol.

Key Material Generation

TLS uses the HMAC standard and its pseudorandom function (PRF) output to generate key material. Thomas explains that "each system starts with the premaster secret; next it creates the master secret. Then it generates the required key material." (125).

The major difference is that SSL uses RSA, Diffie-Hellman or Fortezza/DMS output to create key material. This output generates secret information based on the cipherSuite and Parameters selected during session negotiations.

CertificateVerify

In SSL, the CertificateVerify message requires a complex procedure of messages. With TLS, however, the verified information is completely contained in the handshake messages previously exchanged during the session. (Thomas, 125).

Finished

In TLS, the PRF output of the H-MAC algorithm is used with the master secret and either a "client finished" or a "server finished" designation to create the Finished message. In SSL, the finished message is created in the same ad-hoc manner that key material is generated: using a combination of hash output, selected ciphersuite and parameter information.

Baseline Cipher Suites

As mentioned earlier, SSL specifically supports RSA, Diffie-Hellman and Fortezza/DMS ciphersuites. TLS has stopped allowing Fortezza/DLS support, but allows for ciphersuites to be added to the protocol in future revisions.

Discussion of the Impact of Secure Communications on the E-Commerce Arena

The development of the SSL protocol in the e-commerce arena showed an effort on behalf of the browsers' and applications' developers to protect the information sent over the Internet. This effort gave customers of online stores a sense of safety while using their credit cards online, and guaranteed users of online applications that they were communicating with their intended recipient.

An additional security point to consider, however, is that even though SSL protects information that is passed through the channels of the Internet, it doesn't necessarily protect data that is held on the server. This is why legislations are in effect protecting the data, and why it is important to secure the web servers in addition to using secure connections.

Attempts at "man in the middle" attacks are still possible, and even though the third party could capture the encrypted information, incorrect message authentication would cause the main parties of the secure session to disconnect the current insecure session and reinstantiate a secure session.

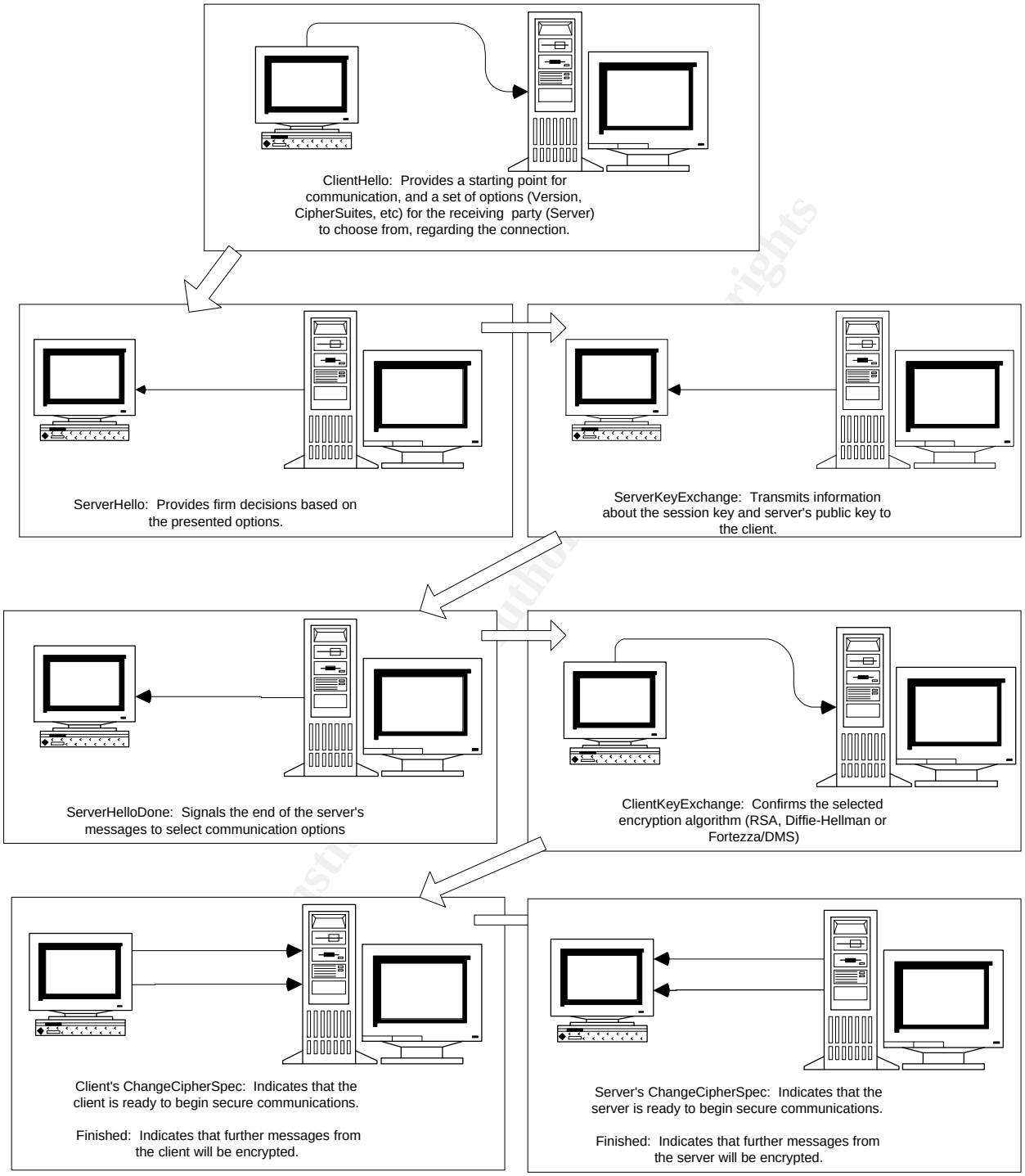
Hackers, as well as developers building more robust systems, are constantly testing the strength of encryption techniques, and the techniques' applications to SSL and TLS. In late February of 2003, researchers at the Swiss Federal Institute of Technology claimed that they cracked IMAP, the version of the SSL protocol used to transmit secure email. Several researchers do not believe that this cracking of the IMAP protocol will have a profound impact on the future of SSL and TLS. Patrick Gray of ZDNet states, "[a]lthough several news sources … have proclaimed that a Swiss research team… has "cracked" SSL, experts are keen to water down the claims" (Gray, 3/14/03). This is because the exploit used to "crack" SSL was a known vulnerability and according to another expert interviewed by Gray, "the problem is with the implementation of the SSL protocol, not the protocol itself. " (Gray, 3/14/03)

Conclusion

The C-I-A (Confidentiality, Integrity, Availability) Model for information security is addressed in several ways by the use of a secure communications protocol. Confidentiality of the information being passed is the main purpose of the SSL and TLS protocols. Integrity is addressed through the use of message authentication in each message from the first handshake. Additionally, non-repudiation is accounted for through certificate passing in addition to the integrity check from the message authentication. Though more responsibility for the Availability portion of the model (in this example) is placed on the server, Availability is slightly addressed since secure communications prevent malicious users from having direct access to the system.

SSL and TLS are proven and effective methods of securing sensitive communications, and as the aggregate of larger amounts of information should be properly secured, secure communications protocols will provide additional useful tools for developers of web systems to implement.

Figure 1: The SSL Handshake Protocol.



References:

Bartlett, Alan and Richard Silverman "SSH: The Secure Shell The Definitive Guide." 31 July 2001. URL: http://www.snailbook.com/ (3 March 2003)

Coleman, Mirean. "HIPAA Electronic Transactions Standards, Code Sets, and the Clinical Social Worker." National Association of Social Workers. July 2002. http://www.socialworkers.org/practice/hipaa/hippa.PDF(9 February 2003)

Cotter, Sean. "SSL Reference." 18 October 2000.
URL: http://www.mozilla.org/projects/security/pki/nss/ref/ssl/ (10 March 2003).

Deitel, Harvey M., Paul J. Deitel & Tem R. Nieto. e-Business & e-Commerce How to Program. Upper Saddle River: Prentice Hall. 2001.

Dierks, T. & C. Allen. "The TLS Protocol Version 1.0." January 1999. (16 February 2003)

Elgamal, Taher. "The Secure Sockets Layer Protocol (SSL)." April 1995. URL: http://www.ietf.org/proceedings/95apr/sec/cat.elgamal.slides.html (3 March 2003)

Freier, Alan O. and Philip Karlton and Paul C. Kocher. "The SSL Protocol Version 3.0." November 1999. (14 March 2003)

Gray, Patrick. "AU experts dampen SSL break claim."
URL: http://www.zdnet.com.au/newstech/security/story/0,2000024985,20272274,00.htm (14 March 2003).

Hiller, Janine S. and Ronnie Cohen. Internet Law & Policy. Upper Saddle River: Prentice Hall, 2000.

URL:http://www.kentlaw.edu/classes/rwarner/legalaspects_ukraine/preventing_access/tutorials/WSSL.html (20 February 2003).

URL: http://ece.gmu.edu/courses/ECE636/viewgraphs/lecture14_protocols3.pdf (26 February 2003)

Martin, Franck. "SSL Certificates HOWTO." 20 October 2002.
URL: http://www.tldp.org/HOWTO/SSL-Certificates-HOWTO/ (14 March 2003)

Microsoft Developers' Network (MSDN). "TLS Handshake Protocol." MSDN. 2003.
URL: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/security/security/tls_handshake_protocol.asp (16 February 2003).

Netscape. "Secure Sockets Layer." 2000. URL:
http://wp.netscape.com/security/techbriefs/ssl.html (14 March 2003).

Reuters. "Swiss Crack E-mail Encryption Code." New York.
URL: http://www.cnn.com/2003/TECH/internet/02/21/email.encryption.reut/index.html (22 February 2003)

Schneider, Gary P. and James T. Perry. Electronic Commerce. 2nd Edition. Boston: Course Technology. 2001.

Solomon, Dr. Alan. "A Brief History of PC Viruses."
URL: http://www.bocklabs.wisc.edu/~janda/solomhis.html (9 February 2003)

Thomas, Stephen. SSL and TLS Essentials: Securing the Web. New York: John Wiley & Sons, Inc. 2000.

Viega, John, Matt Messier and Pravir Chandra. Network Security with OpenSSL. Sebastopol: O'Reilly & Associates, Inc. 2002.