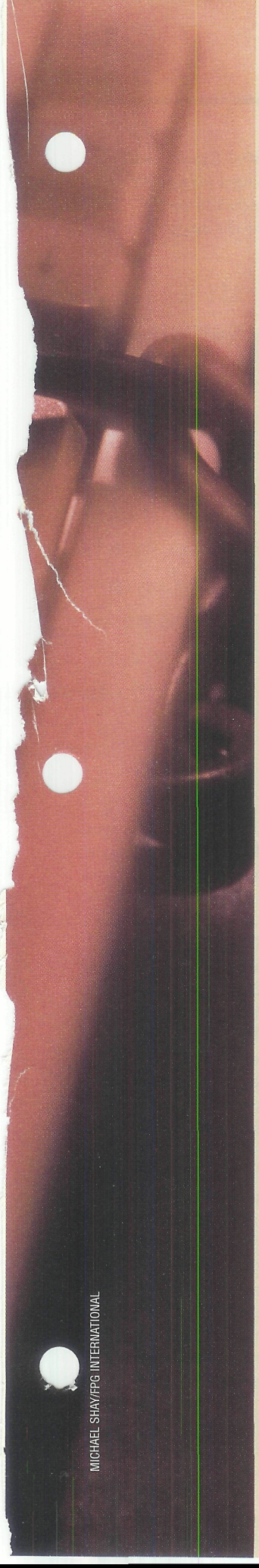


UNDERSTANDING SSL 3.0

by Simson L. Garfinkel

Secure Sockets Layer, or SSL, is a general-purpose protocol for sending encrypted information over the Internet. Developed by Netscape Communications Corp., SSL was first popularized in the Netscape Navigator Web browser and Netscape's Commerce Server. Since then, SSL has been widely adopted by other browser and Web server vendors. Increasingly, SSL is being used for non-Web applications, such as secure telnet, NNTP and SMTP.

Get out your notebooks and take note. This extensible and adaptive encryption protocol looks like it's here to stay.

Overview

SSL is a layer that exists between the raw TCP/IP protocol and the application layer. Whereas the standard TCP/IP protocol simply sends an anonymous, error-free stream of information between two computers (or between two processes running on the same computer), SSL adds several features to that stream:

- Authentication of the server, using digital signatures.
- Authentication of the client, using digital signatures.
- Data compression.
- Data encryption, through a variety of algorithms.
- Data integrity, through the use of Message Authentication Codes.

Cryptography is a fast-moving field, and cryptographic protocols don't work unless both parties to the communication implement the same algorithms. For that reason, SSL is extensible and adaptive. The protocol doesn't

MICHAEL SHAY/FPG INTERNATIONAL

specify that programs use a specific encryption algorithm or hash function; instead, it supports a range of algorithms, including RC2, RC4, DES, IDEA and even the National Security Agency's Clipper chip. When one program using SSL contacts another, the two programs electronically compare notes to determine which is the strongest cryptographic protocol they have in common. This communication is called the SSL Hello.

SSL was designed for worldwide use, but it was developed in the United States and is included in programs that are sold by U.S. corporations for use overseas. For this reason, SSL contains many features designed to implement the U.S. government's policy on the export of cryptographic systems.

Export versions of SSL programs must be crippled in two important ways:

• Public keys may not exceed 512 bits. Export versions of SSL products must use RSA keys that are limited to 512 bits. If an export-only SSL client connects to an SSL server that only has a 1,024-bit RSA public key, the server will create a 512-bit temporary RSA public key and sign the 512-bit key with its 1,024-bit key.

• Secret keys may not exceed 40 bits. Export versions of SSL products are further restricted to using a maximum secret-key length of 40 bits. In practice, SSL actually uses a 128-bit encryption key, but then sends 88 bits of the key unencrypted as part of the communication. The extra 88 bits prevent code-book attacks, but still allow a determined attacker to decrypt the SSL-encrypted communication by trying 2^{40} different keys. This frustrates the kinds of attacks likely to be launched by individuals and small corporations but doesn't hamper the attacks that would be mounted by well-funded intelligence organizations.

SSL was designed for worldwide use, but it was developed in the United States and is included in programs sold by U.S. corporations for use overseas.

Because U.S. export restrictions limit public keys to 512 bits and secret keys to only 40 bits, many people assume that it is just as hard to crack a 512-bit public key as it is to crack a 40-bit secret key. This is not true. A 40-bit secret key can easily be cracked in less than a month using a high-end workstation. There are many reported cases of 40-bit secret keys that have been cracked. On the other hand, there is still no reported case in which a 512-bit public key was cracked.

The reason for this is simple: The 40-bit key is used only once but the 512-bit key is used over and over again. Thus, it makes sense for the 512-bit public key to be thousands of times more secure than the 40-bit secret key.

The SSL Version 3.0 Protocol

The SSL Version 3.0 protocol is arranged in two layers: the Transport Data Layer and the Record Layer. These two layers sit on top of a third layer, which is not strictly part of SSL: the Data Transport Layer (usually TCP/IP).

At the bottom of the SSL protocol is the SSL Record Layer. The Record Layer sends blocks of data, called records, between the client and the server. Each block can contain up to 16,383 bytes of data. According to the specification, "client message boundaries are not preserved in the record layer." This means that if higher-level processes send multiple messages very quickly, those messages may be grouped together into a single SSL record. Alternatively, they might be broken into many SSL records.

Each SSL record contains fields for the content type, protocol version number, length, data payload and a Message Authentication Code (MAC). The MAC provides for data integrity and prevents replay attacks within an SSL session, since each SSL message has a unique sequence number. The data payload may be optionally compressed and encrypted, in that order.

SSL protocols are specific types of messages that are sent using the record layer. SSL 3.0 defines three protocols: the Alert Protocol, the Change Cipher Spec Protocol and the

SSL 3.0 Features

SSL offers many features of both practical and theoretical interest:

• **Separation of Duties.** SSL uses separate algorithms for encryption, authentication and data integrity, with different keys (called secrets) for each function. This allows weak algorithms and short keys to be used for encrypting data that is sent over the Internet, rendering it vulnerable to eavesdropping by intelligence agencies, while still allowing strong algorithms and long keys to be used for authentication and data integrity. (SSL thus embodies the American belief that it's OK to spy on people, but unethical to manufacture evidence against them.)

• **Efficiency.** SSL implementations can cache a "master secret" that is carried over SSL connections. This allows new SSL connections to immediately begin cryptographic communications, without the need to perform more public key operations.

• **Certificate-based authentication.** SSL provides for authentication of both the client and the server through the use of digital certificates and digitally signed challenges.

• **Protocol agnostic.** SSL can run on top of any reliable connection-oriented protocol, such as X.25. SSL cannot run on top of an unreliable datagram protocol such as the IP User Datagram Protocol (UDP).

• **Protection against man-in-the-middle and replay attacks.**

• **Support for compression.**

• **Backwards compatibility with SSL 2.0.**

• **Interoperability between high-grade and low-grade encryption algorithms.**

• **Support for authentication-only.**

• **Support for politically motivated restrictions on cryptography.** –*slg*

Handshake Protocol. Alert Protocols are used for out-of-band signaling about: the state of the SSL connection. As its name implies, the Change Cipher Spec Protocol changes the current encryption algorithm that is being employed. The Handshake Protocol is the most complicated: It is used to create an SSL connection and exchange the necessary cryptographic keys.

The SSL Handshake

When an SSL client connects to an SSL server to initiate a SSL conversation, the first information transmitted consists of the SSL handshake. The SSL handshake establishes the protocols that will be used during the communication, selects the cryptographic algorithms, authenticates the parties, and uses

The SSL 3.0 Specification Defines Two Alert Levels

Alert Level	Level Name	Meaning
1	warning	SSL warnings indicate a problem that is not fatal.
2	fatal	SSL fatal alerts immediately terminate the current SSL session.

SSL 3.0 Defines 12 Alert Descriptions

Alert Number	Alert Name	Meaning
0	close_notify	Indicates that the sender will not send any more information. If a close_notify is sent with a warning alert level, then the session may be resumed. If a close_notify is sent with a fatal alert level, the session may not be resumed.
10	unexpected_message	Inappropriate message was received. "This alert is always fatal and should never be observed in communication between proper implementations."
20	bad_record_mac	Sender received a record with an incorrect MAC. Fatal.
30	decompression_failure	Information in record would not properly decompress. Fatal.
40	handshake_failure	Indicates that the sender was unable to negotiate an acceptable set of security parameters–for example, the sender was not satisfied with the encryption algorithms and strengths available on the recipient. Fatal.
41	no_certificate	Sent in response to a certification request if no appropriate certificate is available.
42	bad_certificate	Sent if a certification request fails–for example, if the certificate is corrupted, or the signature did not verify properly.
43	unsupported_certificate	Sent if the sender does not support the type of certificate sent by the recipient.
44	certificate_revoked	Sent if the sender receives a certificate that was already revoked.
45	certificate_expired	Sent if the sender receives a certificate that has expired.
46	certificate_unknown	Sent if some other error arises during the processing of the certificate.
47	illegal_parameter	Sent if the sender finds that another value in the handshake is out of range or inconsistent. Fatal.

Quotations taken from the specification.

public-key cryptography to create the master secret, from which encryption and authentication keys are derived.

The master secret for the SSL session is created by the client and sent to the server encrypted with the server's public key. The master secret is used to generate four more secrets (keys):

- An encryption key used for sending data from the client to the server.
- An encryption key used for sending data from the server to the client.
- An authentication key used for sending data from the client to the server.
- An authentication key used for sending data from the server to the client.

It is probably a design flaw to rely solely on the client to create the master secret, because this means that the security of the secret is entirely at the client's discretion. Not only must the client's cryptographic implementation be correct (for example, the random number generator must be properly seeded with truly random information), but the user of the client must ensure that their client is a true and faithful copy of the original client program that was verified (for example, the user must be certain that their random number generator has not been disabled by a computer virus). Last year, a team of researchers at the University of California at Berkeley demonstrated repeatedly that they could easily compromise the security of an SSL connection by attacking the Netscape Navigator browser's random number generator.

Sequence of Events

The SSL handshake is performed by a 10-part sequence between the client and the server:

1. The Client opens a connection and sends a ClientHello, which consists of the protocol version, a 32-bit timestamp, a 28-byte random number, a Session ID, a list of ciphers and a list of compression algorithms the client supports.

2. The server sends a ServerHello, which consists of another timestamp, another 28-byte random number, a Session ID, the cipher suite chosen by the server and the compression method chosen by the server. If the Session ID provided matches the Session ID specified by the client, then a previous SSL session is resumed.

3. The server sends its X.509.v3 certificates, if it has any.

4. A `CertificateRequest` is sent by the server, if it wishes the client to authenticate using a public key certificate.

5. If the server has no certificate, it sends a ServerKeyExchange to initiate a Diffie-Hellman key exchange. (Note: The Diffie-Hellman key exchange is not implemented in the current suite of Netscape products.)

6. The client sends its certificate, if it was requested to do so by the SSL server. If no certificate is available, the client sends the no certificate alert. It is up to the SSL server to decide what to do if a no certificate alert is received. The SSL server could continue the SSL transaction with an anonymous client. Alternatively, the SSL server could terminate the connection by sending a fatal handshake failure alert.

7. A `ClientKeyExchange` is sent by the client, which transmits the necessary public key encryption parameters.

8. If the client sends a public certificate that has signing capability (such as an RSA or a DSS certificate), the client now sends a `CertificateVerify` message. This message consists of two message authentication codes: one calculated with the MD5 algorithm, one calculated with SHA.

9. The client and server both send Change Cipher Spec messages.

10. The client and server both send finished messages, which consist of MD5 and SHA hash functions of the master secrets and additional padding. The finished message verifies that both the client and server are in proper synchronization. If they aren't, then the SSL link is terminated.

After the SSL Finished message is sent, application data is transported. All application data is divided into individual

It is probably a design flaw to rely solely on the client to create the master secret, because this means that the security of the secret is entirely at the client's discretion.

SSL URL Information

SSL
The SSL 3.0 protocol.
`http://home.netscape.com/newsref/std/SSL.html`

SSLRef
SSLRef is a reference implementation available from Netscape.
`http://home.netscape.com/newsref/std/sslref.html`

"On Internet Security"
Netscape has prepared a series of Web pages on the subject of Internet security.
`http://home.netscape.com/newsref/ref/internet-security.html`

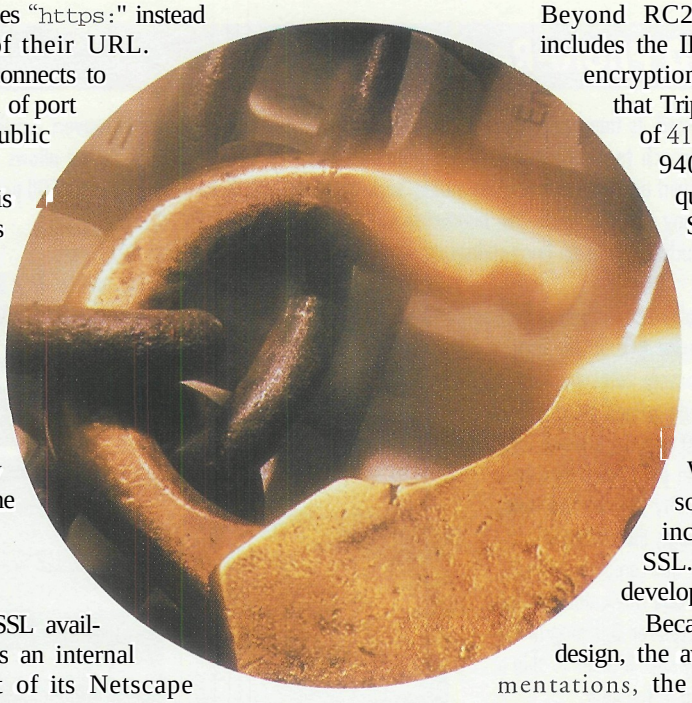
"Netscape Handbook"
The Netscape Handbook's index contains many interesting entries under the letter "S," including Secure Sockets Layer, security, site certification and SOCKS.
`http://home.netscape.com/eng/mozilla/2.0/handbook/docs/atoz.html#S`

"Using Public-Key Cryptography for Internet Security"
Netscape has prepared a series of Web pages that contain information about how SSL uses RSA public key cryptography.
`http://home.netscape.com/newsref/ref/rsa.html`

SSL record-layer messages. These messages are then compressed and encrypted according to the current compression method and cipher suite.

User's Perspective

From the user's perspective, SSL is virtually transparent. With Netscape Navigator, for instance, a user who seeks a secure connection to a Web site simply types "https:" instead of "http:" at the beginning of their URL. Nerscape Navigator automatically connects to the Web server on port 443 instead of port 80 and performs the necessary public key exchange.



This is nor to say that SSL is invisible to the user. SSL facilitates the exchange of public key certificates between the server and the client. These certificates can be inspected using Navigator's "View Document Info" command. If a Web server requests that a browser send a certificate, the browser can be configured to pop up a window and ask the user which certificate he or she wishes to be sent.

Getting SSL

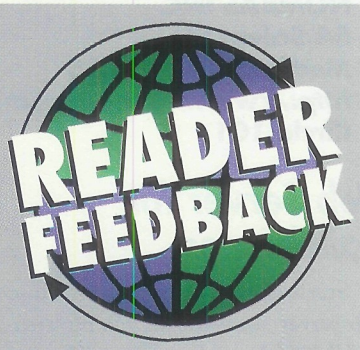
There are many versions of SSL available for general use. Netscape has an internal version that is shipped as part of its Netscape Navigator and other producrs. Netscape also distributes a reference SSL implementation called SSLRef. SSLRef can be freely used for noncommercial applications; licenses are available for commercial use from both Netscape and Consensus Developmenr of Berkeley.

SSLeay is an independent implementation of SSL developed by Eric Young, a computer programmer in Australia. It is freely available around rhe world on a number of anonymous FTP sites. SSLeay uses implementations of the RC2 and RC4 encryption algorithms that were anonymously published on the Usenet sci.crypt newsgroup in September 1994 (RC4) and February 1996 (RC2).

Beyond RC2 and RC4, SSLeay also includes the IDEA, DES and Triple DES encryption algorithms. "Considering that Triple DES can encrypt at rates of 410k/s on a Pentium 100, and 940k/s on a P6/200, this is quite reasonable performance. Single DES clocks in at 1,160k/s and 2,467k/s, respectively. It is actually quite fast for those not so paranoid," Young says.

Meanwhile, a version of SSL has been written in Java. Microsoft Corp.'s new WINSOCK 2.0 Internet socket library for Windows includes specific support for SSL. Other versions are under development as well.

Because of its clean and open design, the availability of different implementations, the ability of programs that employ SSL to abide by the U.S. restrictions on the export of cryptography while still providing the ability to support high-quality security, it is likely that SSL will become the standard Internet protocol for encrypted communications before the middle of next year. •



To help *Webserver Magazine* serve you better, please take a few minutes to close the feedback loop by circling the appropriate numbers on the Reader Service Card located elsewhere in this magazine. Rate the following column and feature topics in this issue.

	INTEREST LEVEL		
Features:	High	Medium	Low
Breaking the Bandwidth Bottleneck	170	171	172
Threads and Networking Communication	173	174	175
Understanding SSL 3.0	176	177	178
Columns:			
WebAdmin - All the Information That's Fit to Log	179	180	181
WebSecurity –Secure CGI Programming in Perl	182	183	184
WebTools –Play That Funky Music	185	186	187