Secure Sockets Layer FAQ 1.1.1

CONTENTS

1) THE SSL-TALK LIST
2) GENERAL SSL QUESTIONS
3) USING PROXIES, GATEWAYS AND FIREWALLS WITH SSL
4) SSL PROTOCOL QUESTIONS
5) CERTIFICATE RELATED QUESTIONS
6) SSL IMPLEMENTATION QUESTIONS
   6.1) NETSCAPE QUESTIONS
   6.2) MICROSOFT QUESTIONS
7) SSL TOOLKIT QUESTIONS
   7.1) SSLREF QUESTIONS
   7.2) SSL PLUS QUESTIONS
   7.3) SSLEAY QUESTIONS

Summary: This document is a summary of FAQ (Frequently Asked Questions)found on the SSL-Talk discussion list regarding technical implementation issues of the Secure Sockets Layer protocol, a transport level security protocol used for securing web servers and clients (such as Netscape Navigator) and other internet applications.

Date: 26 Sep 2001 08:42:07 GMT
Copyright-Notice: (c) Copyright 1996-1998 by Consensus Development Corporation -- All Rights Reserved

SSL-Talk FAQ
Secure Sockets Layer Discussion List FAQ v1.1.1

FAQ Maintained by:
Shannon Appel <SAppel@consensus.com>
Consensus Development Corporation
<http://www.consensus.com/>

The latest edition of this FAQ can always be found at:
<http://www.consensus.com/security/ssl-talk-faq.html>
<http://www.consensus.com/security/ssl-talk-faq.txt>

Copyright (c) 1996-1998 Consensus Development Corporation - All Rights Reserved

****
Due to the November 15, 1998 dissolution of the SSL-Talk mailing list, this will be the last version of this FAQ in its current form. It will be replaced by a more general TLS & SSL FAQ in the near future that is not tied to any mailing list or newsgroup.
****

All information contained in this work is provided "as is." All warranties, expressed, implied or statutory, concerning the accuracy of the information of the suitability for any particular use are hereby specifically disclaimed. While every effort has been taken to ensure the accuracy of the information contained in this work, the authors assume(s) no responsibility for errors or omissions or for damages resulting from the use of the information contained herein.

This work may be copied in any printed or electronic form for non-commercial, personal, or educational purposes if the work is not modified in any way, provided that the copyright notice, the notices of any other author included in this work, and this copyright agreement appear on all copies.

Consensus Development Corporation also grants permission to distribute this work in electronic form over computer networks for other purposes, provided that, in addition to the terms and restrictions set forth above, Consensus Development Corporation and/or other cited authors are notified and that no fees are charged for access to the information in excess of normal online charges that are required for such distribution.

This work may also be mentioned, cited, referred to or described (but not copied or distributed, except as authorized above) in printed publications, on-line services, other electronic communications media, and otherwise, provided that Consensus Development Corporation and any other cited author receives appropriate attribution.

Comments about, suggestions about, or corrections to this document are welcomed. If you would like to ask us to change this document in some way, the method we appreciate most is for you to actually make the desired modifications to a copy of the posting, and then to send us the modified document, or a context diff between the posted version and your modified version (if you do the latter, make sure to include in your mail the "Version:" line from the posted version). Submitting changes in this way makes dealing with them easier for us and helps to avoid misunderstandings about what you are suggesting.

Many people have in the past provided feedback and corrections; we thank them for their input.

In particular, many thanks to:

Christopher Allen <ChristopherA@consensus.com>
Shannon Appel <SAppel@consensus.com>
Nelson Bolyard <NelsonB@netscape.com>
Tim Dierks <TimD@consensus.com>
Eric Greenberg <ericg@netscape.com>
Charles Neerdaels <chuckn@netscape.com>
Bruce Schneier <schneier@counterpane.com>
Tom Weinstein <tomw@netscape.com>
Jonathan Zamick <JonathanZ@consensus.com>

Remaining ambiguities, errors, and difficult-to-read passages are not their fault. :)

---

1) THE SSL-TALK LIST

This section contains information about the SSL-Talk list.

1.1) What is the SSL-Talk List?

The SSL-Talk List was an email list intended for discussion of the technical issues of implementing the SSL protocol. It ceased to exist on November 15, 1998.

Past discussions included issues of software development, cryptanalysis of the protocol and of its various implementations, testing, interoperability, the applicability of SSL to additional TCP-based applications, infrastructure growth questions, etc.

1.1.1) Do archives of the SSL-Talk List exist?

Yes. An archive is maintained at:

<http://www.egroups.com/list/ssl-talk/>

It covers the list from 1995-1998 and is filled with useful information.

We are not aware of any plain text archives of the list.

1.2) What is SSL?

SSL is the Secure Sockets Layer protocol. Version 2.0 originated by Netscape Development Corporation, and version 3.0 was designed with public review and input from industry, and is defined at
<http://home.netscape.com/eng/ssl3/index.html>

1.2.1) What is TLS?

TLS is the Transport Layer Security protocol. It is effectively SSL 3.1 and was submitted to the IETF standards committee for change control in 1996. It should be close to release.

1.3) Has netscape replaced the SSL-Talk mailing list?

Yes. Netscape, the host of the old ssl-talk mailing list, has replaced it with a newsgroup.

The newsgroup netscape.dev.ssl is now available via two means:

a) from <snews://secnews.netscape.com/>

Note: snews is nntp over SSL. Supported in Communicator 4.x.

b) from

<http://www.dejanews.com/
dnquery.xp?QRY=netscape.dev.ssl&DBS=2&maxhits=25&format=terse>

1.4) Are there any other SSL mailing lists?

Some people prefer mailing lists to newsgroups. Fortunately, several

other mailing lists exist to discuss SSL.

THE SSL DEVELOPER LIST

This is a mailing list specifically geared toward application developers who are incorporating SSL or TLS into their products. It is hosted by Consensus Development, a division of Certicom, who has helped to develop the TLS specifications. To join, send a message to:

join-ssl-dev@lists.consensus.com

THE SSL LIST

As with the older SSL-Talk mailing list, the purpose of this mailing list is to discuss any SSL & TLS related issues. It covers the whole spectrum of issues, from beginners on up and is more oriented toward users of SSL-enabled applications. To subscribe simply send e-mail to:

ssl-subscribe@engine.ca

THE SSL-USERS LIST

This is a list concerning SSLeay, a public implementation of SSL. To subscribe send mail to:

factotum@lists.cryptsoft.com

The command "subscribe ssl-users" must appear in the body of the message.

THE IETF-TLS LIST

This is a mailing list dedicated to the writing of the TLS specification for the IETF. To subscribe, send a message to:

join-ietf-tls@lists.consensus.com

---

2) GENERAL SSL QUESTIONS

This section contains general information on SSL and the SSL protocol.

2.1) What is the current version of the SSL protocol?

The current version is SSL 3.0, as documented at
<http://home.netscape.com/eng/ssl3/index.html>

Errata to the SSL 3.0 Specification is periodically posted on the SSL discussion list, and is available at
<http://home.netscape.com/eng/ssl3/ssl-errata.html>

Netscape has submitted SSL 3.0 to the IETF-TLS Working Group as an Internet Draft (see the section 4.5 of this FAQ for more info on TLS). This will be TLS 1.0:
<http://www.ietf.org/internet-drafts/
draft-ietf-tls-protocol-05.txt>

The previous version of SSL, version 2.0 is documented at
<http://home.netscape.com/newsref/std/SSL_old.html>

2.2) Where can I get a "management overview" of SSL and web security?

There is a brief introduction on how Netscape uses public key cryptography in the SSL protocol called "Using Public Key Cryptography" at
<http://home.netscape.com/newsref/ref/rsa.html>

An overview on certificates and VeriSign's Digital IDs is at
<http://digitalid.verisign.com/crp_intr.htm>.

General information on Netscape security can be found in a set of web pages called "Network Security Solutions", at:
<http://home.netscape.com/products/security/index.html>

2.3) Where can I get a more in-depth look at SSL and web security?

The online version of the technical specifications for the SSL 3.0 protocol is at
<http://home.netscape.com/eng/ssl3/ssl-toc.html>

A PostScript version is also available at
<http://home.netscape.com/eng/ssl3/index.html>

A FAQ for SSLeay, a freeware implementation which support SSL 2.0, SSL 3.0, and TLS 1.0, is available at
<http://www.psy.uq.oz.au/~ftp/Crypto/>

A rather broad list of public key related documents, with a focus on certificates and standards can be found at
<http://www.xcert.com/~marcnarc/PKI/>

2.4) What software supports SSL 2.0 and SSL 3.0?

A list of web servers that support SSL 3.0 can be found using the powersearch at:
<http://webcompare.internet.com>

SSL is not just for web servers and is supported by numerous other internet clients and servers.

2.5) What are the laws regarding the import and export of cryptography in various countries?

There is an impressive "International Law Crypto Survey" of cryptographic laws and regulations throughout the world at
<http://cwis.kub.nl/~frw/people/koops/lawsurvy.htm>

RSA Data Security, Inc. offers an Acrobat version of their "Frequently Asked Questions: Export" at
<http://www.rsa.com/PUBS/exp_faq.pdf>

Other information on US export issues can be found on the Electronic Frontier Foundation's web site at
<http://www.eff.org/>

Canadian export issues are covered at
<http://insight.mcmaster.ca/org/efc/pages/doc/crypto-export.html>

---

3) USING PROXIES, GATEWAYS, AND FIREWALLS WITH SSL

This section contains information on how the SSL protocol interacts with proxy servers, security gateways, and firewalls.

3.1) What is a proxy server?

A proxy server is a computer program that resides on your firewall and acts as a conduit between your computer and the broader Internet. In addition to acting as network guardian and logging traffic, a proxy server can also provide an enterprise cache for files as well as replication and site-filtering services.

Any application which needs to communicate through a proxy has to negotiate with the proxy first before continuing through the firewall. Netscape Navigator works with many different types of proxies (such as the CERN proxy server and their own Netscape Proxy Server) and gateways that use the SOCKS protocol.

One problem with SSL-based traffic is that it does not allow caching and replication with proxy servers. For a proxy server to support SSL it must either support SOCKS (a protocol independent proxy mechanism), or use a special SSL Tunneling protocol. The Netscape Proxy Server supports both SOCKS and the SSL Tunneling protocol.

3.2) How does SSL work through (application level) firewalls, gateways and proxy servers?

SSL was designed to provide security between client and server and to avoid any kind of 3-way man-in-the-middle attack. Thus SSL cannot be proxied through traditional application level firewalls (such as the CERN proxy server) because SSL considers a proxy server to be a middleman.

The simplest alternative to this problem is to use a packet filtering firewall. You set it up to open a reserved and trusted port for the SSL+HTTP or SSL+NNTP services (443 or 563 respectively) allowing all traffic on those ports to be passed through unrestricted. The risk with this solution is that an internal attacker could attempt to use these trusted ports without using SSL and there is no way for the firewall to know.

SSL also can work with gateways that support the SOCKS protocol, a protocol independent proxy mechanism. SOCKS is a generic byte forwarding gateway between client and server and generally works at the socket level. If all you want is TCP/UDP restrictions based on client IP or server IP, SOCKS works fine.

However, most non-SSL HTTP proxies work at the protocol level and have the ability to understand header information related to the protocol. This goes beyond SOCKS to allow the firewall administrator to use the header information for filtering and/or monitoring the traffic. Also, SOCKS does not offer the firewall administrator enough information about the request to let it decide whether to allow it and whether to log the request.

A more secure approach is to use a firewall that supports the SSL Tunneling CONNECT extension method as described in the Internet draft

<http://search.ietf.org/internet-drafts/
draft-luotonen-web-proxy-tunneling-01.txt>

In SSL Tunneling, the client initiates an SSL connection via normal HTTP then handshakes and creates a secure connection to the server via a byte-forwarding tunnel. The proxy has access to the client-proxy request headers, but the session is encrypted. Once the handshake occurs, the proxy acts just like a SOCKS gateway. This allows the firewall to monitor the requests, but not the traffic.

The biggest difference between the two methods is that when using SOCKS, DNS resolution is the responsibility of the client, whereas when requests are forwarded through a proxy, DNS resolution is the responsibility of the proxy.

The are three additional things that the SSL Tunneling mechanism does with the proxy server that do not happen when using SOCKS:

* The client sends a "user agent" message (for example, "Mozilla/3.0/Macintosh").

* The proxy can send to the client an authorization request allowing the administrator to use passwords to control external Internet access.

* The standard is more easily extensible. For example, the client could, in theory, send the URL being requested (or anything else) to the firewall. However, there is no standard to support this behavior and as far as we know there are no products which do it.

The Netscape Proxy Server supports the SSL Tunneling CONNECT extension method for tunneling SSL, and the use of the proxy is described in

<http://developer.netscape.com/docs/manuals/proxy/adminux/
encrypt.htm>

Another solution, also available using the Netscape Proxy Server, is that the proxy server can spoof SSL on behalf of the internal client. The proxy will initiate SSL between itself and other servers on the Internet, but be unsecure inside the firewall between the proxy server and the client.

This compromise means that client authentication is not possible; only server authentication of the remote sites is available. However, you gain the ability for client authentication between the client and the proxy. The administrator must decide which is more important, until such time as a better solution arises. The description of this feature of the Netscape Proxy Server is at

<http://developer.netscape.com/docs/manuals/proxy/adminux/
encrypt.htm>

Reverse proxies are a solution for serving secure content inside a firewall to outside clients. For the Netscape Proxy Server this is described at

<http://developer.netscape.com/docs/manuals/proxy/adminux/
revpxy.htm>

It is possible for a proxy server to hold both client and server keys for its internal clients. This allows SSL sessions to be

carried out twice: once between the client and proxy server, and again between the proxy server and the secure server. Thus, the proxy server can to listen in on the conversation without having the private keys of external servers. Clearly this isn't reasonable for the general internet, but it is a viable solution for corporate requirements inside a firewall.

Netscape Proxy Server 3.5 supports this feature. It can be used as described above, or simply to create a secure tunnel between sites across an insecure network. This is really multiple sessions of SSL, not an end-to-end secure connection.

This means that 3.5 has full SSL support as opposed to just SSL tunneling. It can therefore do client authentication and serve documents like a secure server, or request documents like an SSL-enabled client. SSL doesn't allow recursive encryption, so by using it this way you lose the transparency of the proxy and get multiple segments of secure connections, rather than a single end-to-end connection.

3.3) Since SSL is supposed to withstand replay attacks, does this preclude proxy servers from caching the data?

A proxy server must pass SSL directly through without caching.

3.4) What ports does SSL use?

Theoretically SSL can transparently secure any TCP-based protocol running on any port if both sides know the other side is using SSL. However, in practice, separate port numbers have been reserved for each protocol commonly secured by SSL -- this allows packet filtering firewalls to allow such secure traffic through.

As of October 1998, SSL has the following port numbers reserved with the Internet Assigned Numbers Authority (IANA), a part of the Internet Engineering Task Force (IETF):

Keyword	Decimal	Description
nsiiops	261/tcp	IIOP Name Service over TLS/SSL
https	443/tcp	http protocol over TLS/SSL
ddm-ssl	448/tcp	DDM-SSL
smtps	465/tcp	smtp protocol over TLS/SSL
nntps	563/tcp	nntp protocol over TLS/SSL
sshell	614/tcp	SSLshell
ldaps	636/tcp	ldap protocol over TLS/SSL
ftps-data	989/tcp	ftp protocol, data, over TLS/SSL
ftps	990/tcp	ftp, control, over TLS/SSL
telnets	992/tcp	telnet protocol over TLS/SSL
imaps	993/tcp	imap4 protocol over TLS/SSL
ircs	994/tcp	irc protocol over TLS/SSL
pop3s	995/tcp	pop3 protocol over TLS/SSL

A listing of all IANA port assignments can currently be found at <http://www.isi.edu/in-notes/iana/assignments/port-numbers>.

3.5) Do you have any information on sftp?

SSL FTP has been assigned port 990 under the name ftps.

---

4) SSL PROTOCOL QUESTIONS

This section contains more detailed information on the SSL protocol.

4.1) Does SSL protect users from replay attack by eavesdroppers or message interceptors?

Yes. The client and the server each provide part of the random data used to generate the keys for a connection. (The random portions of the connection that initiate a session, drawn from both the client and the server, are used to generate the master secret associated with that session.) Additionally, each record is protected with a MAC (Message Authentication Code) that contains a sequence number for the message.

4.2) Isn't encrypt-only SSL open to "man-in-the-middle" attacks?

Yes, even though SSL 3.0 defines an encrypt-only cipher suite (the SSL_DH_anon_WITH_DES_CBC_SHA cipher suite), there are many possible attacks against it, and some recommend against using it. SSL *MUST* have strong server authentication or it becomes open to some attacks. Netscape's browser and server products do not presently support encrypt-only cipher suites for this reason.

4.3) When did MD5 get "disavowed"?

It hasn't been truly "disavowed", but weaknesses have been discovered such that some people believe that an alternative should be found. These weaknesses were found by Dr. Hans Dobbertin <dobbertin@skom.rhein.de> of the German Information Security Agency in a paper called "Cryptanalysis of MD5 Compress" dated May 2, 1996. A postscript version of the paper is at
<http://www.cs.ucsd.edu/users/bsy/dobbertin.ps>.

SSL uses MD5 in combination with SHA for all negotiation. It also uses MD5 alone in most negotiated cipher suites. However, in these cases it is used with the HMAC construction, which strengthens it such that there are no known problems with this construction.

It has been proposed with TLS to start phasing out all use of MD5.

4.4) The record protocol sits underneath the other protocols, right? It appears that information can be sent only in blocks. Does there have to be a one-to-one mapping between write() calls on the client/server and SSL records? Is there some other blocking taking place when user data is being sent?

The record layer takes a data stream from the higher layers and fragments it into records. If the write is longer than 2^14 bytes (with headers), the record layer will generate multiple records. Multiple writes can be condensed into a single record.

4.5) It appears that there is no way in the SSL protocol to resynchronize blocks if they get out of synch. Is that true?

Yes. SSL relies on an underlying reliable protocol to assure that bytes are not lost or inserted. There was some discussion of reengineering the future TLS protocol to work over datagram

protocols such as UDP, however TLS 1.0 does not support this.

4.6) Why does SSL3 have Diffie-Hellman encryption at all? What good is it? Exchanging random numbers that are encrypted with the server's (or client's) public key would seem to be an adequate way of getting the secret bits across. Why have DH as well?

Anonymous DH key exchange doesn't require the use of certificates. Ephemeral DH allows you to use signing-only certificates, and it protects the session from future compromise of the server's private key. Another advantage of DH is that the patent expired in 1997.

4.7) What is TLS? What happened at these meetings? Has anything come out of them yet?

TLS is the Transport Layer Security Working Group of the IETF (Internet Engineering Task Force). It is the working group responsible for moving transport layer protocols such as SSL through the standards tracks.

IETF working groups do most of their activities through mailing lists and thrice-annual IETF meetings. The first official IETF-TLS Working Group meeting was June 1996 in Montreal. (Before then it was an unofficial BOF "birds of a feather" group.)

The home page for the IETF-TLS Working Group is at
<http://www.ietf.org/html.charters/tls-charter.html>

The discussion list for IETF-TLS is at IETF-TLS@CONSENSUS.COM. You subscribe and unsubscribe by sending to IETF-TLS@CONSENSUS.COM with subscribe or unsubscribe in the SUBJECT of the message. Archives of the list are at
<http://www.imc.org/ietf-tls/mail-archive/>

Minutes are available for a number of past IETF-TLS meetings.

August 1998:
Not currently available

March 1998:
<http://www.ietf.org/proceedings/98mar/98mar-edited-114.htm>

December 1997:
<http://www.ietf.org/proceedings/97dec/97dec-final-116.htm>

April 1997:
<http://www.consensus.com/ietf-tls/minutes-9704.txt>

December 1996:
<http://www.consensus.com/ietf-tls/minutes-9612.txt>

June 1996:
<http://www.consensus.com/ietf-tls/minutes-9606-1.txt>
<http://www.consensus.com/ietf-tls/minutes-9606-2.txt>

May 1996:
<http://www.consensus.com/ietf-tls/minutes-9605.txt>

A number of internet-draft documents have been submitted to the IETF-TLS Working Group.

The TLS Protocol 1.0 (Current Version 06):
<ftp://ftp.ietf.org/internet-drafts/
draft-ietf-tls-protocol-06.txt>

Addition of Kerberos Cipher Suites to Transport Layer Security (TLS):
<http://www.ietf.org/internet-drafts/
draft-ietf-tls-kerb-cipher-suites-03.txt>

ECC Cipher Suites for TLS
<http://www.ietf.org/internet-drafts/draft-ietf-tls-ecc-00.txt>

HTTP over SSL:
<http://www.ietf.org/internet-drafts/draft-ietf-tls-https-01.txt>

An Internet AttributeCertificate Profile for Authorization
<http://www.ietf.org/internet-drafts/
draft-ietf-tls-ac509prof-00.txt>

TLS extensions for AttributeCertificate based authorization
<http://www.ietf.org/internet-drafts/
draft-ietf-tls-attr-cert-01.txt>

The following internet drafts are expired, but are of historical interest:

Addition of Shared Key Authentication to Transport Layer Security (TLS):
<draft-ietf-tls-passauth-00.txt>
(16885 bytes, expires May '97)

Modifications to the SSL protocol for TLS:
<draft-ietf-tls-ssl-mods-00.txt> (9271 bytes, expires May '97)

Secure FTP over SSL:
<draft-murray-auth-ftp-ssl-00.txt>
(14238 bytes, expires June '97)

SSH Transport Layer Protocol (originally <draft-ietf-tls-ssh-00.txt>)
<http://www.consensus.com/ietf-tls/tls-ssh-00.txt>
(44411 bytes, expired December '96)

4.8) What is the purpose of pad1 and pad2, and why were the numbers 0x36 and 0x5c chosen?

The purpose of the construction of a "keyed-MAC" in the form of HASH(K,pad2,HASH(K,pad1,text)). It was proposed by the cryptographer Hugo Krawczyk of IBM as a much more secure alternative to traditional MACs. In a paper last year he demonstrated a proof that even if the hash function was relatively weak (as MD5 has since proven itself to be) the addition of the secret key in the function makes it significantly more secure. The particular method proposed by Krawczyk is now known as an HMAC.

The particular construction that Netscape uses for SSL is based on the original internet-draft, and since that time it has been revised such that it XORs the pads rather than appending them -- a nice consequence of which is that pads are of the same size whether you use MD5 or SHA; it also allows for long keys and has some security advantages. This version may now be found as RFC 2104:

<http://andrew2.andrew.cmu.edu/rfc/rfc2104.html>

In the proposals we've seen for the IETF-TLS Working Group the scheme SSL 3.0 uses will be replaced by the official RFC HMAC technique.

The particular pad bytes used are the ones defined in Krawczyk's original HMAC paper. We believe that they are relatively arbitrary. The salient property is that half the bits differ: the hamming distance between 0x36 and 0x5c is 4 out of a possible 8. We don't know if the fact that each of the pads also has a hamming weight of 4 is significant or not.

4.9) Are you aware of any SSL toolkits supporting client authentication?

SSLRef 3.0 and SSL Plus both support SSL 3.0 client authentication. SSLeay supports SSL 2.0 and 3.0 client authentication as well as the proposed TLS standard for client authentication.

4.10) What SSL implementations should I test against?

There is no formal conformance testing, but Netscape does currently offer an interoperability test server that has been used to test conformance with many other implementations of SSL 3.0. This server is located at
<https://ssl3.netscape.com/>

Another interoperability test server can be found at:
<https://ssl3.c2.org>

VeriSign also has an "Authentic Site" program listing various sites that use SSL authentication. Also included is a test page that requires that you present a valid VeriSign client certificate. More information on the Authentic Site program is at
<http://www.verisign.com/authentic/>

Client authentication can be tested at:
<https://in-103.infospace.com/>

4.11) What is the difference between SSL 2.0 and 3.0?

Security improvements:

1. SSL 2.0 is vulnerable to a "man-in-the-middle" attack. An active attacker can invisibly edit the list of ciphersuite preferences in the hello messages to invisibly force both client and server to use 40-bit encryption. SSL 3.0 defends against this attack by having the last handshake message include a hash of all the previous handshake messages.

2. SSL 2.0 uses a weak MAC construction, although post-encryption seems to stop attacks. This is fixed in 3.0.

3. SSL 2.0 feeds padding bytes into the MAC in block cipher modes, but leaves the padding-length field unauthenticated, which could allow active attackers to delete bytes from the end of messages. This, too, is fixed in 3.0.

4. In SSL 3.0, the Message Authentication Hash uses a full 128 bits of keying material, even when using an Export cipher. In SSL 2.0,

Message Authentication used only 40 bits when using an Export cipher.

Functionality improvements:

1. In SSL 2.0, the client can only initiate a handshake at the beginning of the connection. In 3.0, the client can initiate a handshake routine, even in the middle of an open session. A server can request that the client start a new handshake. Thus, the parties can change the algorithms and keys used whenever they want.

2. SSL 3.0 allows the server and client to send chains of certificates. This allows organizations to use a certificate hierarchy that is more than two certifications deep.

3. SSL 3.0 has a generalized key exchange protocol. It allows Diffie-Hellman and Fortezza key exchanges and non-RSA certificates.

4. SSL 3.0 allows for record compression and decompression.

Backward compatibility:

1. SSL 3.0 can recognize an SSL 2.0 client hello and fall back to SSL 2.0. An SSL 3.0 client can also generate an SSL 2.0 client hello with the version set to SSL 3.0, so SSL 3.0 servers will continue the handshake in SSL 3.0, and SSL 2.0 server will cause the client to fall back to SSL 2.0.

Other:

1. SSL 3.0 separates the transport of data from the message layer. In 2.0, each packet contained only one handshake message. In 3.0, a record may contain part of a message, a whole message, or several messages. This requires different logic to process packets into handshake messages. Therefore, the formatting of the packets had to be completely changed.

2. Cipher specifications, handshake messages, and other constants are different.

---

5) CERTIFICATE RELATED QUESTIONS

This section contains information on certificates used by the SSL protocol.

5.1) How does Netscape handle client certificates in Communicator 4.X? Navigator 3.X?

Netscape describes their framework for web-based key generation and certificate issuing on their web pages at
<http://home.netscape.com/eng/security/certs.html>

5.2) What is the format of the SSL certificates used by Netscape Navigator?

Netscape has documented their SSL 2.0 certificate format at
<http://home.netscape.com/newsref/std/ssl_2.0_certificate.html>.

5.3) I am distributing load on several different web servers and I don't want to have to have a different certificate for each. How can I do this?

When establishing a secure connection in SSL, many SSL clients applications, including Netscape's Navigator, check the common name of the certificate against the name of the site in the URL. If it doesn't match, the client application warns the user. Thus the preferred format of a common name of an SSL server is a simple DNS name like "www.consensus.com".

To support multiple servers you can use a round-robin DNS to send each request for "www.consensus.com" to different IP addresses. As Netscape Navigator does not check to see that the IP address matches the original domain name (reverse-IP), this will work for each round-robin server.

Netscape's Navigator will also allow for some simple pattern matching. Netscape has documented a number of different possibilities in their SSL 2.0 Certificate Format web pages at:
<http://home.netscape.com/newsref/std/ssl_2.0_certificate.html>

Note, however, none of these regular expression/pattern matching choices are accepted by VeriSign. In the past they have accepted server certificate common names with regular expressions, but these are no longer allowed.

Other CAs may have different policies regarding use of regular expressions in common names.

5.4) When comparing a URL against the common name of the certificate, why don't you do a reverse-DNS lookup?

DNS is not a secure name service, and trying to treat it like one could be a security hole. The purpose of checking the common name against the URL is to make sure that at least the user's expectation of what site the user is visiting is not compromised.

5.5) Does Netscape require hierarchical naming (that is, distinguished names) for its certificates?

Yes, Netscape requires distinguished names.

5.6) Where can I get more information on certificates?

PKIX is an IETF working group dedicated to providing standards for an X509-based PKI. You can find their charter at:
<http://www.ietf.org/html.charters/pkix-charter.html>

VeriSign, the default CA (Certificate Authority) used by Netscape and most other WWW browsers has a FAQ at:
<http://digitalid.verisign.com/id_faqs.htm>

Entrust has a primer on Web Security with an emphasis on Certificate Authorities at:
<http://www.entrust.com/products/library/primer.htm>

There is also a good resource of links to a variety of certificate technical and policy issue sites available at:

<http://www.xcert.com/~marcnarc/PKI/>

5.7) What other CAs exist besides VeriSign?

We know of these CAs:

Thawte Consulting <http://www.thawte.com/certs/>
COST Computer Security Technologies <http://www.cost.se/>
CompuSource <http://www.compusource.co.za/id/personal/>
XCert <http://www.xcert.com>

Numerous other CAs now exist; additional links will be included in the replacement TLS/SSL FAQ intended for the future.

5.8) How do I set up my own Certificate Authority?

There is some support for creating your own CA in SSLeay; there is information on how to integrate it with Netscape available at:
<http://www.webvision.com/developers_new/casetup.html>

Several specific products also exist; additional links will be included in the replacement TLS/SSL FAQ intended for the future.

5.9) What criteria should I use in deciding between one CA and another?

The purpose of a Certificate Authority is to bind a public key to the common name of the certificate, and thus assure third parties that some measure of care was taken to ensure that this binding is valid. A measure of a Certificate Authority is their "Policy Statement" which states what measures they take for each class of certificate they offer to ensure that this binding of identity with public key is valid.

5.10) What are Attribute Certificates?

Attribute Certificates are a new type of certificate proposed by Netscape. These are signed objects that assert additional properties about a particular identity certificate.

An attribute cert has no associated key pair and consequently cannot be used to establish identity. Informally, one can think of them as a mechanism for extending the attributes of an identity certificate without requiring that the identity certificate be reissued.

More details of the proposal are at
<http://lists.w3.org/Archives/Public/ietf-tls/msg00796.html>

---

6) SSL IMPLEMENTATION ISSUES

This section offers specific implementation details of different SSL clients and servers that are not specific to the protocol.

6.1) NETSCAPE QUESTIONS

This is not an official statement by Netscape, and Netscape has not reviewed this for accuracy. For additional information, please see:

<http://home.netscape.com/info/security-doc.html>
<http://home.netscape.com/eng/security/>

6.1.1) I just downloaded a new version of Netscape's browser, and it doesn't have 128-bit encryption. What version(s) of the browser have 128-bit encryption?

All versions of Netscape Navigator and Communicator, except "Preview Release" versions, are available in two flavors: a "domestic" flavor with 128-bit encryption for use in the USA and Canada, and an "export" or "international" flavor with only 40-bit encryption. (There is also a third flavor of Communicator 4.x available for use in France.) Preview releases are only available in the export flavor. To get 128-bit encryption, you must download the U.S. flavor.

6.1.2) I just downloaded a newly released version of Netscape's browser and my bank's server tells me my browser does not have adequate security. What's wrong?

Here are the likely explanations for this:

a) You have downloaded an "export" flavor of the browser with only 40-bit encryption, but your bank requires that you use the "domestic" flavor of the browser with 128-bit encryption. In this case, you must download the domestic 128-bit version.

b) Your bank's server keeps a list of the browser versions with which it will work, and that list has not yet been updated to include the very latest version(s) of Netscape's browser. In this case, please ask your bank to add the newest version of Netscape's browser to their server's list of acceptable versions. Note that many banks will not accept "Preview Release" versions because they do not contain domestic (128-bit) encryption.

6.1.3) I downloaded a version of Netscape's browser that is newer than version 4.05. Now, when I go to certain https web sites that used to work for me (like my bank) I get an error message telling me that "Netscape has received bad data from the server." I've been told the problem is with SSL v3 in my new browser, and that I should disable SSL v3 in my browser. What's wrong with SSL v3 in these new browsers? Should I disable it?

Newer versions of Netscape's browsers enforce the legal export control requirements of the SSL v3 specification and will not work with servers that violate the export control provisions of the SSL v3 specification.

Some SSL servers do not properly follow the SSL v3 specification's requirements for the U.S. Government's export control regulations. Netscape's server products, and most other brands of server products, conform to the specification, but a few others do not.

We strongly advise you to NOT disable SSL v3 in your browser. If you do disable SSL v3, you lose the extra security protections of SSL v3 with ALL the https web sites you visit. By keeping SSL v2 and v3

enabled in your browser, you get the best protection each site can provide.

Please ask the failing web site to upgrade to conforming servers.

Web sites whose servers violate the specification have several options at their disposal, including falling back on the less secure SSL v2, by disabling the non-conformant SSL v3 in their servers, or replacing their servers with servers that conform to the SSL v3 spec.

6.1.4) Do Netscape's browsers cache data on disk that has been received via https?

Navigator 3.0 and Communicator 4.x have an option to allow on-disk caching of data fetched over SSL connections. The default setting is to not cache https data on disk.

In Navigator 2.0, documents fetched using SSL were cached in the same way as non-SSL documents. You could use the "Pragma: no-cache" HTTP header to disable caching for a particular page.

In Navigator 1.0, documents fetched with SSL were not cached on disk.

6.1.5) Is the cached data encrypted using some key?

No, Navigator and Communicator do not encrypt documents that are stored in the cache.

6.1.6) Does Netscape use "regular" RSA libraries (such as BSAFE) or "custom" RSA code? More specifically, is Netscape using BSAFE 3.0?

Netscape is a BSAFE source licensee. Much of the code in BSAFE 3.0 has been integrated into Netscape's products. However, the BSAFE API is not available to plugins.

6.1.7) Are the 512-bit RSA keys used by exportable servers generated on the fly by Netscape's servers? How often are they changed?
Does the Netscape server take care of changing them automatically?

In Netscape's server products, if the server's public key is longer than 512 bits, the server generates a temporary 512-bit export key at start-up time. This key is regenerated only when the server is restarted.

6.1.8) How can additional root CA certificates be added to the browser's certificate database?

Root keys for CA (Certificate Authority) certificates may be loaded using an SSL connection to a previously unknown CA. Please see:

<http://home.netscape.com/eng/security/comm4-cert-download.html>

for more information. Also, new releases of the Navigator have added additional CA root keys.

6.1.9) What X.509v3 certificate extensions are supported by the various versions of Netscape browsers?

Please see <http://home.netscape.com/eng/security/certs.html>.

6.1.10) The Help Information for Netscape's Enterprise server indicates that the server supports 6 ciphers for SSL 2.0 and 6 ciphers for SSL 3.0. However, the Encryption|Security Preferences menu in the server Manager displays only 2 choices for SSL 2.0 and 3 choices for SSL 3.0. How can I select the other choices?

The Enterprise server is available in two flavors, the "domestic" flavor with 128-bit encryption for use in the USA and Canada, and the "export" or "international" flavor with only 40-bit encryption. If you do not have all the ciphers, then you have the export flavor of the server. If you want to use the others, you must use the domestic (non-export) flavor.

6.1.11) When will Netscape support SSL sockets for Java browser applets?

There are presently no announced plans to do so.

6.2) MICROSOFT QUESTIONS

The text for sub-section 6.2 was grabbed from various documents found at

<http://www.microsoft.com/workshop/security/default.asp>

6.2.1) Which of Microsoft's products will support SSL?

Internet Explorer 3.0 provides support for SSL versions 2.0 and 3.0 and for Private Communication Technology (PCT) version 1.0. It will include support for the Transport Layer Security Protocol (TLS), which is being considered by IETF.

6.2.2) Which Microsoft products support Client Authentication?

Client authentication as implemented by Microsoft Internet Explorer 3.0 is interoperable with popular Web servers that support secure sockets layer (SSL) 3.0 client authentication.

Internet Information Server 3.0 supports client authentication using standards-based X.509 version 3 certificates. Webmasters can easily add client authentication to their Web sites by creating an Active Server Pages (ASP) application.

---

7) SSL TOOKIT QUESTIONS

This section offers specific details of different SSL development toolkits that are not specific to the protocol.

7.1) SSLREF QUESTIONS
This subsection contains information on SSLRef 3.0 which was

codeveloped by Netscape Communications Corp. of Mountain View, California <http://home.netscape.com/> and Consensus Development Corporation of Berkeley, California <http://www.consensus.com/>.

7.1.1) What is SSLRef 3.0?

SSLRef 3.0 is a reference implementation of the SSL (Secure Sockets Layer) protocol. SSLRef 3.0 is intended to aid and accelerate developers' efforts to provide security within TCP/IP applications. It can also be used to qualify other implementations of version 3.0 of the SSL protocol.

SSLRef 3.0 consists of a software library, distributed as ANSI C source-code, that can be compiled on Windows 95/NT and Solaris platforms and then linked into TCP/IP application programs. SSLRef 3.0 was also designed to be easily ported to a wide variety of other platforms and operating systems.

More information on SSLRef can be found at
<http://home.netscape.com/newsref/std/sslref.html>

If you are a US citizen you can download SSLRef 3.0 at
<http://test-drive.netscape.com/tdrive-new/sslref.html>

7.1.2) How can I license SSLRef 3.0? What does it cost? With what restrictions?

The SSLRef 3.0 distribution includes a license for non-commercial use. For commercial licensing, send mail to <sslref@netscape.com>.

The SSLRef 3.0 commercial license is Part Number 70-01128-00 and the price is $30,000. The license agreement is a flat one-time fee, not a recurring royalty.

SSLRef 3.0 may not be exported. However, the encryption options in SSLRef 3.0 can be limited to make exportable products.

SSLRef 3.0 does not include an RSA/BSAFE license for required cryptographic functions. Most users would use BSAFE.

For BSAFE information contact RSA at
<http://www.rsa.com/>

7.2) SSL PLUS QUESTIONS

This sub-section contains information specific to the SSL Plus: SSL 3.0 Integration Suite(tm) software toolkit developed by Consensus Development Corporation of Berkeley, California <http://www.consensus.com/>.

7.2.1) What is the relationship between SSLRef and SSL Plus?

SSLRef 3.0 was written by Netscape Development Corporation and Consensus Development Corporation. SSL Plus, a derivative of SSLRef 3.0, is fully supported and offers unique value-added features.

SSL Plus 2.0 includes numerous updates to SSLRef 3.0, support, a VeriSign certificate request tool, and a "signer" file format for storing keys and certificates. It is qualified for additional platforms, and system integration services are available. When the TLS spec becomes official, SSL Plus will be upgraded to that new protocol.

SSLRef 3.0 offers 4 ciphersuites:

* RSA authenticated, unencrypted, with MD5
  (SSL_RSA_WITH_NULL_MD5)

* RSA authenticated with exportable RC4 encryption, and MD5
  (SSL_RSA_EXPORT_WITH_RC4_40_MD5)

* RSA authenticated with DES encryption, and SHA
  (SSL_RSA_WITH_DES_CBC_SHA)

* Diffie-Hellman anonymous key exchange with DES encryption, and SHA
  (SSL_DH_anon_WITH_DES_CBC_SHA)

SSL Plus 2.0 adds support for an additional 6 ciphersuites:

* RSA authenticated, unencrypted, with SHA
  (SSL_RSA_WITH_NULL_SHA)

* RSA authenticated with non-exportable RC4 encryption, with MD5 or SHA
  (SSL_RSA_WITH_RC4_128_MD5 & SSL_RSA_WITH_RC4_128_SHA)

* RSA authenticated with Triple-DES encryption, with SHA
  (SSL_RSA_WITH_3DES_EDE_CBC_SHA)

* Diffie-Hellman anonymous key exchange with RC4 encryption, with MD5
  (SSL_DH_anon_WITH_RC4_128_MD5 &
   SSL_DH_anon_WITH_3DES_EDE_CBC_SHA)

* Diffie-Hellman anonymous key exchange with Triple-DES encryption and SHA
  (SSL_DH_anon_WITH_RC4_128_MD5 &
   SSL_DH_anon_WITH_3DES_EDE_CBC_SHA)

For more information on SSL Plus features see
<http://www.consensus.com/SSLPlus/sslplus_stats.html>

7.2.2) What is the relationship between SSL Plus and SSLRef 2.0?

There is no relationship between SSLRef 2.0 and SSL Plus -- SSL Plus was originally based on SSLRef 3.0 which was not based on SSLRef 2.0.

7.2.3) How can I license SSL Plus?

SSL Plus is available for commercial use only. Certicom will work with you to provide a license tailored to fit your company's business model, whether you prefer flat yearly licenses, royalty based payment or even one time buyouts.

Since SSL Plus is a protocol toolkit, SSL Plus customers will also require a license for one of the standard crytographic libraries such

as Certicom's Security Builder, or RSA's BSAFE.

You can get more information about SSL Plus at <http://www.consensus.com/SSLPlus>, and about Certicom's Security Builder toolkit at <http://www.certicom.com/cSecure/sdk.htm>

In addition, you can contact Consensus directly for more information on SSL Plus, including pricing, technical information, or more information about cryptographic library tools, at <mailto:sales@consensus.com> or at 510/649-3300.

7.2.4) Is there any relationship between SSL Plus and Winsock 1.1 or Winsock 2.0? Which Winsock would you recommend using to test our SSL? Does it matter if Winsock 1.1 or 2.0 architecture is used?

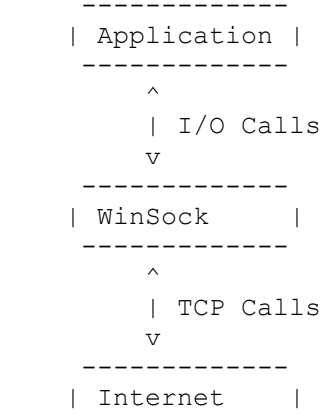
No -- SSL Plus is designed to be transport independent and work with both socket and stream styles of I/O. SSL Plus includes some examples of using WinSock 1.1 in the Win32 builds of our sample code. However, we recommend that you write your own callback code if you want better handling of your I/O than what our sample routines provide.

7.2.5) How does the data flow within the application, WinSock, SSL, TCP/IP stack layers?

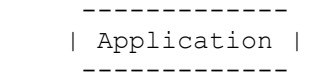
The short answer is that you insert SSL Plus between your I/O and your application code.

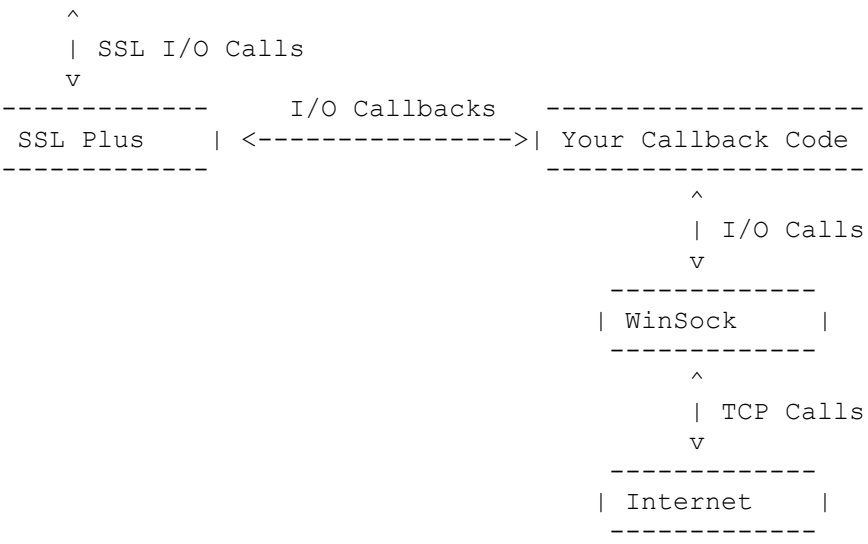
Basically, you call SSL Plus instead of your read and write. SSL Plus does its stuff and calls your callback code to do the I/O. Data comes through your I/O routines, through SSL Plus, and then finally to your application. SSL Plus only manages the data flowing through the connection; it does not handle setting up and tearing down the underlying network connection; your application should open the network connection, then hand it off to SSL Plus for SSL handshaking and data transfer. (This step is not shown in the diagram).

Normal:



SSL Plus:





7.2.6) With the WinSock 2.0 architecture, the application need only chose an appropriate SSL-enabled service provider. Does SSL Plus support this?

With WinSock 2.0 there is some discussion of functionality that allows you to create a module that you could add to WinSock 2.0.

At this time we do not believe that this functionality is actually shipping (as Microsoft was supporting PCT but is now supporting SSL 3), but we do know that it is part of their plans. See the MS-ISF (Microsoft Internet Security Framework) description at
<http://premium.microsoft.com/msdn/library/backgrnd/html/msdn_misf.htm>

We can't speak to when or if Microsoft will add it to their system software, or if another third-party offers such a module.

Meanwhile, there has been some discussion on what changes might be required under WinSock 2.0 to do SSL. See:
<http://home.netscape.com/newsref/std/ssl_integration.html>

7.2.7) Does SSL Plus support yielding?

SSL Plus includes support for processor yielding during cryptographic operations. Because developers provide their own I/O routines, they can do yielding during I/O. Our sample code include examples of I/O yielding.

7.2.8) I don't understand the nomenclatures of constants such as "SSL_RSA_EXPORT_WITH_RC4_40_MD5" -- where are they defined?

They are actually defined by the SSL 3.0 specification, but also see section 7.2.1 for an overview.

7.2.9) In what order are the cipher suites called?

The default order of the cipher suites is:

SSL_RSA_WITH_3DES_EDE_CBC_SHA
SSL_RSA_WITH_RC4_128_SHA
SSL_RSA_WITH_RC4_128_MD5

SSL_RSA_WITH_DES_CBC_SHA
SSL_RSA_EXPORT_WITH_RC4_40_MD5
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA
SSL_DH_anon_WITH_RC4_128_MD5
SSL_DH_anon_WITH_DES_CBC_SHA
SSL_RSA_WITH_NULL_SHA
SSL_RSA_WITH_NULL_MD5

7.2.10) Can I change the order of the cipher suites?

Yes. This is easily done with the SSLSetCipherSuites function.

7.2.11) Does SSL Plus support compression?

Not as of 2.0. If there is a specific customer requirement, or if a compression cipher suite is defined we expect to support it in the future, but otherwise we have no plans here.

7.2.12) In the function SSLWriteRecord(), the data buffer is copied, encrypted, then enqueued on the SSL write queue. The function then returns. What thread services the write queue? How is the thread created?

The write queue is serviced by the public function called SSLServiceWriteQueue(). It is called in a number of places in ssltrspt.c, including with every call to SSLWrite(). Data to be written is sent to the I/O layer as you exit out of the write function (for example, right near the bottom of SSLWrite).

If SSLWrite() returns SSLWouldBlockError, then make a call to SSLServiceWriteQueue() to service the write queue. (You could instead make a call to SSLWrite() with more data to be written, but this is unlikely.)

The write queue is not serviced by a separate execution thread. The write queue mechanism was designed to support non-blocking I/O without undue overhead.

7.2.13) When I call SSLRead(), on returning, the length argument should be replaced with the number of bytes actually read. In practice, this doesn't seem to be happening. What am I doing wrong?

The difficulty is that it's hard for SSL to precisely emulate the behavior of Unix-style socket calls.

The problem is that you are using SSL Plus in its blocking mode; if you return SSLWouldBlock from your I/O Read callback, the library will return the data it has along with the SSLWouldBlock error.

The best way to solve this is always to know how much data you're waiting for and to request exactly that much. I know this doesn't work with a lot of free-form Internet protocols.

Alternatively, you would like the call to block until it gets some data, then return it to you, even if it's less than 512 bytes. Ideally, you'd like to do this without busy-looping the CPU waiting for data. The best way to do this using SSL Plus is to write a

wrapper for SSLRead() which does the following:

* Make a blocking select() call until there is some data available on the TCP/IP connection over which you're speaking SSL. This will cause you to block in a friendly way until data arrives.

* Call SSLRead(). If zero bytes are returned from the read, loop and do the select() again. Otherwise, return whatever came back.

* Make your Read() callback non-blocking. The easiest thing to do is to check how much data is available on the incoming connection and return SSLWouldBlockErr if you can't completely fulfill the request. (You can optionally read what data there is and return it first; this won't affect functionality).

This will result in the following behavior:

1. Your program will block gracefully in the select() call until something arrives on the connection.

2. You will then ask SSL Plus to read some data.

3. SSL Plus will ask the Read() callback to read the header of the next record (3 or 5 bytes).

4. The Read() callback will fulfill that, if possible

5. SSL Plus will ask to read the body of the record (whose length will be equal to how much data was sent by the other side, plus MAC and encryption padding).

6. The Read() callback will fulfill that, if possible.

7. If the amount of data received is greater than or equal to how much was requested in 2., the data will be returned

8. Otherwise, go back to 3.

What will happen in practice looks something like this: because the SSL peer on the other end of the connection generates record layer records monolithically, and they're relatively small, the header and content of a record will arrive at your machine all together. Thus, when your select() call returns, you will be able to successfully read a header and body without blocking. When SSL Plus goes to read another one, your Read() callback will see that there's no data available on the connection (assuming another record hasn't arrived) and return SSLWouldBlockErr. SSL Plus will then return the data it has received and the error SSLWouldBlockErr; you can return that data as a partial completion of the desired read.

If a partial record arrives, your select() will wake up, but SSL Plus won't be able to decrypt and check a complete record before the Read() callback returns SSLWouldBlockErr; thus, your read will return with zero bytes returned. Since this isn't the behavior your client expects, you should select() again until more data arrives, hopefully completing the record.

7.2.14) If session cache is stored in a database, can multiple Unix processes share the same session data?

There is no information stored in the session database which can't be passed between processes. Specifically, there is no pointer indirection. Of course, you'll have to figure out how to pass session database records (and their changes or deletions) between processes; that is not part of SSL Plus.

7.3) SSLEAY QUESTIONS

This sub-section contains information specific to the SSLeay toolkit developed by Eric Young <eay@mincom.com>

7.3.1) Where is the SSLeay FAQ?

There is a very complete SSLeay FAQ at:
<http://www.psy.uq.oz.au/~ftp/Crypto/>

.. Shannon Appel Consensus Development Corporation ..
.. Research Assistant a subsidiary of Certicom Corporation ..
.. <SAppel@consensus.com> 2930 Shattuck Ave. #206 ..
.. <http://www.consensus.com> Berkeley, CA 94705-1883 ..
.. <http://www.certicom.com> o510/649-3300 f510/649-3301 ..

Send corrections/additions to the FAQ Maintainer:
Shannon Appel <SAppel@consensus.com>

Last Update October 21 2001 @ 10:34 PM