HTTPS—HTTP Server and Client with SSL 3.0

First Published: March 31, 2003
Last Updated: August 6, 2007

The HTTPS—HTTP Server and Client with SSL 3.0 feature provides Secure Socket Layer (SSL) version 3.0 support for the HTTP 1.1 server and HTTP 1.1 client within Cisco IOS software. SSL provides server authentication, encryption, and message integrity to allow secure HTTP communications. SSL also provides HTTP client authentication. HTTP over SSL is abbreviated as HTTPS.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for HTTPS—HTTP Server and Client with SSL 3.0" section on page 35.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for HTTPS—HTTP Server and Client with SSL 3.0

To enable secure HTTP connections (encryption) without a configured certificate authority trustpoint, you must first ensure that each device has the key (such as a Rivest, Shamir, and Adleman [RSA] public key or a shared key) of the other device. In most cases, an RSA key pair will be generated automatically. The RSA key pair is used for creating a self-signed certificate (which is also generated automatically).

Restrictions for HTTPS—HTTP Server and Client with SSL 3.0

The HTTPS—HTTP Server and Client with SSL 3.0 feature is available only in Cisco IOS software images that support SSL. SSL is supported in "IPSec 56" (contains "k8" in the image name) and "IPSec 3DES" images (contains "k9" in the image name). "IPSec 56" images provide up to 64-bit encryption, "IPSec 3 DES" images provide greater than 64-bit encryption. The following CipherSuites are supported in IPSec Data Encryption Standard (DES) images:

- SSL_RSA_WITH_RC4_128_MD5—RSA key exchange (RSA Public Key Cryptography) with RC4 128-bit encryption and Message-Digest Algorithm 5 (MD5) for message digest
- SSL_RSA_WITH_RC4_128_SHA—RSA key exchange with RC4 128-bit encryption and Secure Hash Algorithm (SHA) for message digest
- SSL_RSA_WITH_3DES_EDE_CBC_SHA—RSA key exchange with 3DES and DES-EDE3-CBC for message encryption and SHA for message digest
- SSL_RSA_WITH_DES_CBC_SHA—RSA key exchange with DES-CBC for message encryption and SHA for message digest

For IPSec 56 images, only the SSL_RSA_WITH_DES_CBC_SHA CipherSuite is supported. For further details on these CipherSuites, see the *SSL Protocol Version 3.0* Internet-Draft document (see the "Related Documents" section on page 14).

RSA (in conjunction with the specified encryption and digest algorithm combinations) is used for both key generation and authentication on SSL connections. This usage is independent of whether a certificate authority (CA) trustpoint is configured.

Information About HTTPS—HTTP Server and Client with SSL 3.0

To configure the HTTP with SSL 3.0 (HTTPS) feature, you should understand the following concepts:

- Secure HTTP Server and Secure HTTP Client, page 2
- Certificate Authority Trustpoints, page 3
- CipherSuites, page 3

Secure HTTP Server and Secure HTTP Client

A secure HTTP connection means that data sent to and received from an HTTP server are encrypted before being sent out over the Internet. HTTP with SSL encryption provides a secure connection to allow such functions as configuring a router from a web browser. Cisco's implementation of the secure HTTP server and secure HTTP client uses an implementation of the SSL version 3.0. Application layer encryption provides an alternative to older methods such as having to set up a tunnel to the HTTP server for remote management. HTTP over SSL is abbreviated as HTTPS; the URL of a secure connection will begin with https:// instead of http://.

The Cisco IOS HTTP secure server's primary role is to listen for HTTPS requests on a designated port (the default HTTPS port is 443) and to pass the request to the HTTP 1.1 web server. The HTTP 1.1 server processes requests and passes responses (served pages) back to the HTTP secure server, which, in turn, responds to the original request.

The Cisco IOS HTTP secure client's primary role is to respond to Cisco IOS application requests for HTTPS User Agent services, perform HTTPS User Agent services on the application's behalf, and pass the response back to the application.

Certificate Authority Trustpoints

Certificate authorities (CAs) are responsible for managing certificate requests and issuing certificates to participating IPSec network devices. These services provide centralized security key and certificate management for the participating devices. Specific CA servers are referred to as "trustpoints."

The HTTPS server provides a secure connection by providing a certified X.509v3 certificate to the client when a connection attempt is made. The certified X.509v3 certificate is obtained from a specified CA trustpoint. The client (usually a web browser), in turn, has a public key that allows it to authenticate the certificate.

Configuring a CA trustpoint is highly recommended for secure HTTP connections. However, if a CA trustpoint is not configured for the routing device running the HTTPS server, the server will certify itself and generate the needed RSA key pair. Because a self-certified (self-signed) certificate does not provide adequate security, the connecting client will generate a notification that the certificate is self-certified, and the user will have the opportunity to accept or reject the connection. This option is available for internal network topologies (such as testing).

The HTTPS—HTTP Server and Client with SSL 3.0 feature also provides an optional command (**ip http secure-client-auth**) that, when enabled, has the HTTPS server request an X.509v3 certificate from the client. Authenticating the client provides more security than server authentication by itself.

For additional information on certificate authorities, see the "Configuring Certification Authority Interoperability" chapter in the *Cisco IOS Security Configuration Guide*.

CipherSuites

A CipherSuite specifies the encryption algorithm and digest algorithm to use on an SSL connection. Web browsers offer a list of supported CipherSuites when connecting to the HTTPS server, and the client and server will negotiate the best encryption algorithm to use from those that are supported by both. For example, Netscape Communicator 4.76 supports U.S. security with RSA Public Key Cryptography, MD2, MD5, RC2-CBC, RC4, DES-CBC, and DES-EDE3-CBC.

For the best possible encryption, you should use a browser that supports 128-bit encryption, such as Microsoft Internet Explorer version 5.5 (or later), or Netscape Communicator version 4.76 (or later). The SSL_RSA_WITH_DES_CBC_SHA CipherSuite provides less security than the other CipherSuites, because it does not offer 128-bit encryption.

In terms of router processing load (speed), the following list ranks the CipherSuites from fastest to slowest (slightly more processing time is required for the more secure and more complex CipherSuites):

1. SSL_RSA_WITH_DES_CBC_SHA

2. SSL_RSA_WITH_RC4_128_MD5
3. SSL_RSA_WITH_RC4_128_SHA
4. SSL_RSA_WITH_3DES_EDE_CBC_SHA

How to Configure the HTTPS—HTTP Server and Client with SSL 3.0

To configure the HTTPS—HTTP Server and Client with SSL 3.0 feature, complete the procedures in the following sections:

- Declaring a Certificate Authority Trustpoint, page 4
- Configuring the HTTPS Server with SSL 3.0, page 7
- Configuring Standard and Secure HTTP Server Options, page 9
- Configuring the HTTPS Client with SSL 3.0, page 11

Declaring a Certificate Authority Trustpoint

Configuring a CA trustpoint is highly recommended for secure HTTP connections. The certified X.509v3 certificate for the secure HTTP server (or client) is obtained from the specified CA trustpoint. If you do not declare a CA trustpoint, then a self-signed certificate will be used for secure HTTP connections. The self-signed certificate is generated automatically.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *name*
4. **ip domain-name** *name*
5. **crypto key generate rsa usage-keys**
6. **crypto ca trustpoint** *name*
7. **enrollment url** *url*
8. **enrollment http-proxy** *host-name port-number*
9. **crl** {**query** *url* | **optional** | **best-effort**}
10. **primary**
11. **exit**
12. **crypto ca authenticate** *name*
13. **crypto ca enrollment** *name*
14. **copy running-config startup-config**
    or
    copy system:running-config nvram:startup-config

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`hostname`** *`name`*
	Example:
	`Router(config)# hostname Router`
		Specifies the hostname of the router.
		• This step is needed only if you have not previously configured a hostname for your router. The hostname is required because a fully qualified domain name is needed for security keys and certificates.

Step 4	**`ip domain-name`** *`name`*
	Example:
	`Router(config)# ip domain-name example.com`
		Specifies the IP domain name of the router.
		• This step is needed only if you have not previously configured an IP domain name for your router. The domain name is required because a fully qualified domain name is needed for security keys and certificates.

Step 5	**`crypto key generate rsa usage-keys`**
	Example:
	`Router(config)# crypto key generate rsa usage-keys`
		(Optional) Generates an RSA key pair.
		• The **usage-keys** keyword specifies that two RSA special-usage key pairs should be generated (that is, one encryption pair and one signature pair) instead of one general-purpose key pair.
		• RSA key pairs are used to sign and encrypt Internet key exchange (IKE) key management messages and are required before you can obtain a certificate for your router.
		• RSA key pairs are generated automatically. This command can be used to regenerate the keys, if needed.
		Note There are other keywords and arguments for this command, but they do not pertain to this feature.

Step 6	**`crypto ca trustpoint`** *`name`*
	Example:
	`Router(config)# crypto ca trustpoint TP1`
		Specifies a local configuration name for the CA trustpoint and enters CA trustpoint configuration mode.
		Note The **crypto ca identity** command was replaced by the **crypto ca trustpoint** command in Cisco IOS Release 12.2(8)T.

Step 7	**`enrollment url`** *`url`*
	Example:
	`Router(ca-trustpoint)# enrollment url http://example.com`
		Specifies a URL of the CA where your router should send certificate requests.
		• If you are using Simple Certificate Enrollment Protocol (SCEP) for enrollment, the URL argument must be in the form **http://***CA-name*, where *CA-name* is the host Domain Name System (DNS) name or IP address of the CA trustpoint.

	Command or Action	Purpose

Step 8	**`enrollment http-proxy`** *`host-name port-number`*
	Example:
	`Router(ca-trustpoint)# enrollment http-proxy example.com 8080`
		(Optional) Configures the router to obtain certificates from the CA through an HTTP proxy server.

Step 9	**`crl`** `{`**`query`** *`url`* `|` **`optional`** `|` **`best-effort`**`}`
	Example:
	`Router(ca-trustpoint)# crl query ldap://example.com`
		Configures the router to request a certificate revocation list (CRL), make CRL checking optional, or perform CRL checking on a “best-effort” basis.
		• CRLs ensure that the certificate of the peer has not been revoked.
		• The **crl optional** command configures the router to accept certificates even if the appropriate CRL cannot be downloaded.
		• Use the **crl query** *url* command to specify the Lightweight Directory Access Protocol (LDAP) URL of the CA server; for example, **ldap://another-server**.

Step 10	**`primary`**
	Example:
	`Router(ca-trustpoint)# primary`
		(Optional) Specifies that this trustpoint should be used as the primary (default) trustpoint for CA requests.
		• Use this command if more than one CA trustpoint will be configured on this router.

Step 11	**`exit`**
	Example:
	`Router(ca-trustpoint)# exit`
		Exits CA trustpoint configuration mode and returns to global configuration mode.

Step 12	**`crypto ca authenticate`** *`name`*
	Example:
	`Router(config)# crypto ca authenticate TP1`
		Authenticates the CA by getting the public key of the CA.
		• Use the same name that you used when declaring the CA in the **crypto ca trustpoint** command.

Step 13	**`crypto ca enrollment`** *`name`*
	Example:
	`Router(config)# crypto ca enrollment TP1`
		Obtains the certificate from the specified CA trustpoint.
		• This command requests a signed certificate from the CA for each RSA key pair.

Step 14	**`copy running-config startup-config`**
	or
	`copy system:running-config nvram:startup-config`
	Example:
	`Router(config)# copy running-config startup-config`
		Saves the configuration to NVRAM.
		• This command is required to save the certificates into NVRAM. If not used, the certificates would be lost at router reload.
		Note To execute EXEC mode commands in global configuration mode, you can add the **do** keyword before the command. For example, instead of **copy running-config startup-config**, you could enter **do copy running-config startup-config**.

Configuring the HTTPS Server with SSL 3.0

To disable the standard HTTP server and configure the HTTPS server with SSL 3.0, complete the procedures in this section.

Prerequisites

If a certificate authority is to be used for certification, you should declare the CA trustpoint on the routing device before enabling the secure HTTP server.

SUMMARY STEPS

1. **enable**
2. **show ip http server status**
3. **configure terminal**
4. **no ip http server**
5. **ip http secure-server**
6. **ip http secure-port** *port-number*
7. **ip http secure-ciphersuite** [**3des-ede-cbc-sha**] [**rc4-128-sha**] [**rc4-128-md5**] [**des-cbc-sha**]
8. **ip http secure-client-auth**
9. **ip http secure-trustpoint** *name*
10. **end**
11. **show ip http server secure status**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`Router# show ip http server status` **Example:** `Router# show ip http server status`	(Optional) Displays the status of the HTTP server. • If you are unsure whether the secure HTTP server is supported in the software image you are running, enter this command and look for the line "HTTP secure server capability: {Present | Not present}". • This command displays the status of the standard HTTP server (enabled or disabled).

Step 3	`configure terminal`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose

Step 4	**`no ip http server`**
	Example:
	`Router(config)# no ip http server`
		Disables the standard HTTP server.
		Note When enabling the HTTPS server you should always disable the standard HTTP server to prevent insecure connections to the same services. This is a precautionary step (typically, the HTTP server is disabled by default).

Step 5	**`ip http secure-server`**
	Example:
	`Router(config)# ip http secure-server`
		Enables the HTTPS server.

Step 6	**`ip http secure-port`** *`port-number`*
	Example:
	`Router(config)# ip http secure-port 1025`
		(Optional) Specifies the port number that should be used for the HTTPS server. The default port number is 443. Valid options are 443 or any number in the range 1025 to 65535.

Step 7	**`ip http secure-ciphersuite`** [**`3des-ede-cbc-sha`**] [**`rc4-128-sha`**] [**`rc4-128-md5`**] [**`des-cbc-sha`**] **Example:** `Router(config)# ip http secure-ciphersuite rc4-128-sha rc4-128-md5`	(Optional) Specifies the CipherSuites (encryption algorithms) that should be used for encryption over the HTTPS connection. • This command allows you to restrict the list of CipherSuites that the server offers the connecting clients. For example, you may want to allow only the most secure CipherSuite to be used. • Unless you have a reason to specify the CipherSuites that should be used, or you are unfamiliar with the details of these CipherSuites, you should leave this command unconfigured and let the server and client negotiate the CipherSuite that they both support (this is the default).

Step 8	**`ip http secure-client-auth`**
	Example:
	`Router(config)# ip http secure-client-auth`
		(Optional) Configures the HTTP server to request an X.509v3 certificate from the client in order to authenticate the client during the connection process.
		• In the default connection and authentication process, the client requests a certificate from the HTTP server, but the server does not attempt to authenticate the client. Authenticating the client provides more security than server authentication by itself, but not all clients may be configured for CA authentication.

Step 9	**`ip http secure-trustpoint`** *`trustpoint-name`*
	Example:
	`Router(config)# ip http secure-trustpoint trustpoint-01`
		Specifies the CA trustpoint that should be used to obtain an X.509v3 security certificate and to authenticate the connecting client's certificate.
		• Use of this command assumes you have already declared a CA trustpoint using the **crypto ca trustpoint** command and associated submode commands.
		• Use the same trustpoint name that you used in the associated **crypto ca trustpoint** command.

	Command or Action	Purpose

Step 10	**`end`**
	Example:
	`Router(config)# end`
		Ends the current configuration session and returns you to privileged EXEC mode.

Step 11	**`show ip http server secure status`**
	Example:
	`Router# show ip http server secure status`
		Displays the status of the HTTP secure server configuration.

What to Do Next

To verify the configuration of the HTTPS server, connect to the router running the HTTPS server with a web browser by entering **https://***url*, where *url* is the IP address or hostname of the router. If a port other than the default port is configured (using the **ip http secure-port** command), you must also specify the port number after the URL. For example:

```
https://209.165.202.129:1026
```

or

```
https://host.domain.com:1026
```

Generally, you can verify that you have a secure connection to an HTTP server by looking for an image of a padlock at the bottom of your browser window. Also note that secure HTTP connections have a URL that starts with “https:” instead of “http:”.

Configuring Standard and Secure HTTP Server Options

The configuration of the standard HTTP server applies to the secure HTTP server as well. The following optional commands are standard HTTP server commands that provide additional security and efficiency to both the standard HTTP server and the HTTPS server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http path** *path-name*
4. **ip http access-class** *access-list-number*
5. **ip http max-connections** *value*
6. **ip http timeout-policy idle** *seconds* **life** *seconds* **requests** *value*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`ip http path`** *`path-name`*
	Example:
	`Router(config)# ip http path slot1:`
		(Optional) Sets the base HTTP path for HTML files.
		• The base path is used to specify the location of the HTTP server files (HTML files) on the local system. Generally, the HTML files are located in system flash memory.

Step 4	**`ip http access-class`** *`access-list-number`*
	Example:
	`Router(config)# ip http access-class 20`
		(Optional) Specifies the access list that should be used to allow access to the HTTP server.

	Command or Action	Purpose

Step 5	**`ip http max-connections`** *`value`*
	Example:
	`Router(config)# ip http max-connections 10`
		(Optional) Sets the maximum number of concurrent connections to the HTTP server that will be allowed. The default value is 5.

Step 6	**`ip http timeout-policy idle`** *`seconds`* **`life`** *`seconds`* **`requests`** *`value`*
	Example:
	`Router(config)# ip http timeout-policy idle 30 life 120 requests 100`
		(Optional) Sets the characteristics that determine how long a connection to the HTTP server should remain open. The characteristics are:
		• **idle**—The maximum number of seconds the connection will be kept open if no data is received or response data cannot be sent out on the connection. Note that a new value may not take effect on any already existing connections. If the server is too busy or the limit on the **life** time or the number of **requests** is reached, the connection may be closed sooner. The default value is 180 seconds (3 minutes).
		• **life**—The maximum number of seconds the connection will be kept open, from the time the connection is established. Note that the new value may not take effect on any already existing connections. If the server is too busy or the limit on the idle time or the number of requests is reached, it may close the connection sooner. Also, because the server will not close the connection while actively processing a request, the connection may remain open longer than the specified **life** time if processing is occurring when the life maximum is reached. In this case, the connection will be closed when processing finishes. The default value is 180 seconds (3 minutes). The maximum value is 86,400 seconds (24 hours).
		• **requests**—The maximum limit on the number of requests processed on a persistent connection before it is closed. Note that the new value may not take effect on any already existing connections. If the server is too busy or the limit on the idle time or the life time is reached, the connection may be closed before the maximum number of requests are processed. The default value is 1. The maximum value is 86,400.

Configuring the HTTPS Client with SSL 3.0

To configure the HTTPS client with SSL 3.0, complete the procedures in this section.

Prerequisites

The standard HTTP client and the secure HTTP client are always enabled.

A certificate authority is required for secure HTTP client certification; the following steps assume that you have previously declared a CA trustpoint on the routing device. If a CA trustpoint is not configured, and the remote HTTPS server requires client authentication, connections to the secure HTTP client will fail.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http client secure-trustpoint** *trustpoint-name*
4. **ip http client secure-ciphersuite** [**3des-ede-cbc-sha**] [**rc4-128-sha**] [**rc4-128-md5**] [**des-cbc-sha**]
5. **end**
6. **show ip http client secure status**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`ip http client secure-trustpoint`** *`trustpoint-name`*
	Example:
	`Router(config)# ip http client secure-trustpoint trustpoint_01`
		(Optional) Specifies the CA trustpoint that should be used if the remote HTTP server requests client authentication.
		• Use of this command assumes you have already declared a CA trustpoint using the **crypto ca trustpoint** command and associated submode commands.
		• Use the same trustpoint name that you used in the associated **crypto ca trustpoint** command.
		• This command is optional if client authentication is not needed, or if a primary trustpoint has been configured. If the **ip http client secure-trustpoint** command is not used, the router will use the primary trustpoint, as specified by the **primary** CA trustpoint configuration mode command.

Step 4	**`ip http client secure-ciphersuite`** `[`**`3des-ede-cbc-sha`**`] [`**`rc4-128-sha`**`] [`**`rc4-128-md5`**`] [`**`des-cbc-sha`**`]`
	Example:
	`Router(config)# ip http client secure-ciphersuite rc4-128-sha rc4-128-md5`
		(Optional) Specifies the CipherSuites (encryption algorithms) that should be used for encryption over the HTTPS connection.
		• This command allows you to restrict the list of CipherSuites that the client offers when connecting to a secure HTTP server. For example, you may want to allow only the most secure CipherSuites to be used.
		• Unless you have a reason to specify the CipherSuites that should be used, or you are unfamiliar with the details of these CipherSuites, you should leave this command unconfigured and let the server and client negotiate the CipherSuite that they both support (this is the default).

	Command or Action	Purpose

Step 5	**`end`**
	Example:
	`Router(config)# end`
		Ends the current configuration session and returns to privileged EXEC mode.

Step 6	**`show ip http client secure status`**
	Example:
	`Router# show ip http client secure status`
		Displays the status of the HTTP secure server configuration.

Configuration Examples for the HTTPS—HTTP Server and Client with SSL 3.0 feature

The following example shows a configuration session in which the secure HTTP server is enabled, the port for the secure HTTP server is configured as 1025, and the remote CA trustpoint server “CA-trust-local” is used for certification.

```
Router# show ip http server status

HTTP server status: Disabled
HTTP server port: 80
HTTP server authentication method: enable
HTTP server access class: 0
HTTP server base path:
Maximum number of concurrent server connections allowed: 5
Server idle time-out: 600 seconds
Server life time-out: 600 seconds
Maximum number of requests allowed on a connection: 1
HTTP secure server capability: Present
HTTP secure server status: Disabled
HTTP secure server port: 443
HTTP secure server ciphersuite: 3des-ede-cbc-sha des-cbc-sha rc4-128-md5 rc4-12a
HTTP secure server client authentication: Disabled
HTTP secure server trustpoint:

Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip http secure-server
Router(config)# ip http client secure-trustpoint CA-trust-local
Router(config)# ip http secure-port 1024
Invalid secure port value.

Router(config)# ip http secure-port 1025
Router(config)# ip http secure-ciphersuite rc4-128-sha rc4-128-md5
Router(config)# end
Router# show ip http server secure status
HTTP secure server status: Enabled
HTTP secure server port: 1025
HTTP secure server ciphersuite: rc4-128-md5 rc4-128-sha
HTTP secure server client authentication: Disabled
HTTP secure server trustpoint: CA-trust-local
```

In the following example, the CA trustpoint CA-trust-local is specified, and the HTTPS client is configured to use this trustpoint for client authentication requests:

```
Router# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# crypto ca trustpoint CA-trust-local
Router(ca-trustpoint)# enrollment url http://example.com
Router(ca-trustpoint)# crl query ldap://example.com
Router(ca-trustpoint)# primary
Router(ca-trustpoint)# exit
Router(config)# ip http client secure-trustpoint CA-trust-local
Router(config)# end
Router# copy running-config startup-config
```

Additional References

The following sections provide references related to the HTTPS—HTTP Server and Client with SSL 3.0 feature.

Related Documents

Related Topic	Document Title
SSL 3.0	*The SSL Protocol Version 3.0* This document is available from various sources online.
Standard Cisco Web Client	*HTTP 1.1 Web Client*
Standard Cisco Web Server	*HTTP 1.1 Web Server*
Certification Authority Interoperability	• *Configuring Certification Authority Interoperability* • *Certificate Autoenrollment* • *Certificate Enrollment Enhancements, Release 12.2(8)T feature document* • *Trustpoint CLI* • *Source Interface Selection for Outgoing Traffic with Certificate Authority*

Standards

Standard	Title
No new or modified standards are supported by this feature.	—

Related MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Related RFCs

RFCs	Description
RFC 2616	Cisco's implementation of HTTP is based on *RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1*.

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register on Cisco.com.	http://www.cisco.com/techsupport

Command Reference

This section documents only commands that are new or modified.

- **debug ip http ssl error**
- **ip http client secure-ciphersuite**
- **ip http client secure-trustpoint**
- **ip http secure-ciphersuite**
- **ip http secure-client-auth**
- **ip http secure-port**
- **ip http secure-server**
- **ip http secure-trustpoint**
- **show ip http client secure status**

- **show ip http server secure status**

debug ip http ssl error

To enable debugging messages for the HTTPS web server and client, use the **debug ip http ssl error** command in privileged EXEC mode. To disable debugging messages for the secure HTTP web server and client, use the **no** form of this command.

debug ip http ssl error

no debug ip http ssl error

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging message output is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(33)SRA	This command was integrated into the 12.2(33)SRA release.
12.2(33)SXH	This command was integrated into the 12.2(33)SXH release.

Usage Guidelines

This command displays output for debugging purposes related to the secure HTTP server and secure HTTP client. Secure HTTP services use the Secure Socket Layer (SSL) protocol, version 3.0, for encryption.

Examples

The following is sample debugging output from the **debug ip http ssl error** command:

```
000030:00:08:01:%HTTPS:Key pair generation failed
000030:00:08:10:%HTTPS:Failed to generate self-signed cert
000030:00:08:15:%HTTPS:SSL handshake fail
000030:00:08:21:%HTTPS:SSL read fail, uninitialized hndshk ctxt
000030:00:08:25:%HTTPS:SSL write fail, uninitialized hndshk ctxt
```

Table 1 describes the significant fields shown in the display.

Table 1 debug ip http ssl error Field Descriptions

Field	Description
%HTTPS:Key pair generation failed	The RSA key pair generation failed.
%HTTPS:Failed to generate self-signed cert	The HTTPS server or client failed to generate a self-signed certificate.

Table 1 debug ip http ssl error Field Descriptions (continued)

Field	Description
%HTTPS:SSL handshake fail	SSL connection handshake failed.
%HTTPS:SSL read fail, uninitialized hndshk ctxt	A read operation failed for SSL with an unitialized handshake context.

Related Commands

Command	Description
ip http secure-server	Enables the HTTPS server.

ip http client secure-ciphersuite

To specify the CipherSuite that should be used for encryption over the secure HTTP connection from the client to a remote server, use the **ip http client secure-ciphersuite** command in global configuration mode. To remove a previously configured CipherSuite specification for the client, use the **no** form of this command.

ip http client secure-ciphersuite [**3des-ede-cbc-sha**] [**rc4-128-sha**] [**rc4-128-md5**] [**des-cbc-sha**]

no ip http client secure-ciphersuite

Syntax Description

3des-ede-cbc-sha	SSL_RSA_WITH_3DES_EDE_CBC_SHA—Rivest, Shamir, and Adleman (RSA) key exchange with 3DES and DES-EDE3-CBC for message encryption and Secure Hash Algorithm (SHA) for message digest.
rc4-128-sha	SSL_RSA_WITH_RC4_128_SHA—RSA key exchange (RSA Public Key Cryptography) with RC4 128-bit encryption for message encryption and SHA for message digest.
rc4-128-md5	SSL_RSA_WITH_RC4_128_MD5—RSA key exchange (RSA Public Key Cryptography) with RC4 128-bit encryption for message encryption and Message Digest 5 (MD5) for message digest.
des-cbc-sha	SSL_RSA_WITH_DES_CBC_SHA—RSA key exchange with DES-CBC for message encryption and SHA for message digest.

Command Default

The client and server negotiate the best CipherSuite that they both support from the list of available CipherSuites.

Command Modes

Global configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

This command allows you to restrict the list of CipherSuites (encryption algorithms) that the client offers when connecting to a secure HTTP server. For example, you may want to allow only the most secure CipherSuites to be used.

Unless you have a reason to specify the CipherSuites that should be used, or you are unfamiliar with the details of these CipherSuites, you should leave this command unconfigured and let the server and client negotiate the CipherSuite that they both support (this is the default). The **no** form of this command returns the list of available CipherSuites to the default (that is, all CipherSuites supported on your device are available for negotiation).

Examples

The following example shows how to configure the HTTPS client to use only the SSL_RSA_WITH_3DES_EDE_CBC_SHA CipherSuite:

```
Router(config)# ip http client secure-ciphersuite 3des-ede-cbc-sha
```

Related Commands

Command	Description
show ip http client secure status	Displays the configuration status of the secure HTTP client.

ip http client secure-trustpoint

To specify the remote certificate authority (CA) trustpoint that should be used if certification is needed for the secure HTTP client, use the **ip http client secure-trustpoint** command in global configuration mode. To remove a client trustpoint from the configuration, use the **no** form of this command.

ip http client secure-trustpoint *trustpoint-name*

no ip http client secure-trustpoint *trustpoint-name*

Syntax Description

trustpoint-name	Name of a configured trustpoint. Use the same trustpoint name that was used in the associated **crypto ca trustpoint** command.

Command Default

If the remote HTTPS server requests client certification, the secure HTTP client will use the trustpoint configured using the **primary** command in the CA trustpoint configuration. If a trustpoint is not configured, client certification will fail.

Command Modes

Global configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

This command specifies that the secure HTTP client should use the certificate associated with the trustpoint indicated by the *trustpoint-name* argument. Use the same trustpoint name that you used in the associated **crypto ca trustpoint** command.

The specified X.509v3 security certificate will be used by the HTTPS client for cases when the remote HTTPS server requires client authorization.

Use this command only if you have already declared a CA trustpoint using the **crypto ca trustpoint** command and associated submode commands. If the remote HTTPS server requires client authorization and a trustpoint is not configured for the client, the remote HTTPS server will reject the connection.

If this command is not used, the client attempts to use the certificate associated with the primary trustpoint. The primary trustpoint is configured using the **primary** command.

Examples

In the following example, the CA trustpoint is configured and referenced in the secure HTTP server configuration:

```
!The following commands specify a CA trustpoint that can be used
!to obtain a X.509v3 security certificate.

Router(config)# crypto ca trustpoint tp1
Router(config-ca)# enrollment url http://host1:80
Router(config-ca)# exit
!The following command is used to actually obtain the security certificate.
!A trustpoint NAME is used because there could be multiple trust points
!configured for the router.

Router(config)# crypto ca enrollment TP1

!The following command specifies that the secure HTTP client
!should use the certificate associated with the TP1 trustpoint for HTTPS connections.
Router(config)# ip http client secure-trustpoint tp1
```

Related Commands

Command	Description
crypto ca trustpoint	Specifies a name for a certificate authority trustpoint and enters CA trustpoint configuration mode.
primary	Indicates that the CA trustpoint being configured should be used as the primary (default) trustpoint.

ip http secure-ciphersuite

To specify the CipherSuites that should be used by the secure HTTP server when negotiating a connection with a remote client, use the **ip http secure-ciphersuite** command in global configuration mode. To return the configuration to the default set of CipherSuites, use the **no** form of this command.

ip http secure-ciphersuite [**3des-ede-cbc-sha**] [**rc4-128-sha**] [**rc4-128-md5**] [**des-cbc-sha**]

no ip http secure-ciphersuite

Syntax Description

3des-ede-cbc-sha	SSL_RSA_WITH_3DES_EDE_CBC_SHA—Rivest, Shamir, and Adleman (RSA) key exchange with 3DES and DES-EDE3-CBC for message encryption and Secure Hash Algorithm (SHA) for message digest.
rc4-128-sha	SSL_RSA_WITH_RC4_128_SHA —RSA key exchange (RSA Public Key Cryptography) with RC4 128-bit encryption for message encryption and SHA for message digest.
rc4-128-md5	SSL_RSA_WITH_RC4_128_MD5 —RSA key exchange (RSA Public Key Cryptography) with RC4 128-bit encryption for message encryption and Message Digest 5 (MD5) for message digest.
des-cbc-sha	SSL_RSA_WITH_DES_CBC_SHA—RSA key exchange with DES-CBC for message encryption and SHA for message digest.

Command Default

The HTTPS server negotiates the best CipherSuite using the list received from the connecting client.

Command Modes

Global configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

This command is used to restrict the list of CipherSuites (encryption algorithms) that should be used for encryption over the HTTPS connection. For example, you may want to allow only the most secure CipherSuites to be used.

Unless you have a reason to specify the CipherSuites that should be used, or you are unfamiliar with the details of these CipherSuites, you should leave this command unconfigured and let the server and client negotiate the CipherSuite that they both support (this is the default).

The supported CipherSuites vary by Cisco IOS software image. For example, “IP Sec56” (“k8”) images support only the SSL_RSA_WITH_DES_CBC_SHA CipherSuite in Cisco IOS Release 12.2(15)T.

In terms of router processing load (speed), the following list ranks the CipherSuites from fastest to slowest (slightly more processing time is required for the more secure and more complex CipherSuites):

1. SSL_RSA_WITH_DES_CBC_SHA
2. SSL_RSA_WITH_RC4_128_MD5
3. SSL_RSA_WITH_RC4_128_SHA
4. SSL_RSA_WITH_3DES_EDE_CBC_SHA

Additional information about these CipherSuites can be found online from sources that document the Secure Sockets Layer (SSL) 3.0 protocol.

Examples

The following exampleshows how to restrictsthe CipherSuites offered to a connecting secure web client:

```
Router(config)# ip http secure-ciphersuite rc4-128-sha rc4-128-md5
```

Related Commands

Command	Description
ip http secure-server	Enables the HTTPS server.
show ip http server secure status	Displays the configuration status of the secure HTTP server.

ip http secure-client-auth

To configure the secure HTTP server to authenticate connecting clients, use the **ip http secure-client-auth** command in global configuration mode. To remove the requirement for client authorization, use the **no** form of this command.

ip http secure-client-auth

no ip http secure-client-auth

Syntax Description

This command has no arguments or keywords.

Command Default

Client authentication is not required for connections to the secure HTTP server.

Command Modes

Global configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

This command configures the HTTP server to request an X.509v3 certificate from the client in order to authenticate the client during the connection process.

In the default connection and authentication process, the client requests a certificate from the HTTP server, but the server does not attempt to authenticate the client. Authenticating the client provides more security than server authentication by itself, but not all web clients may be configured for certificate authority (CA) authentication.

Examples

In the following example the secure web server is enabled and the server is configured to accept connections only from clients with a signed security certificate:

```
Router(config)# no ip http server
Router(config)# ip http secure-server
Router(config)# ip http secure-client-auth
```

Related Commands

Command	Description
ip http secure-server	Enables the HTTPS server.
show ip http server secure status	Displays the configuration status of the secure HTTP server.

ip http secure-port

To set the secure HTTP (HTTPS) server port number for listening, use the **ip http secure-port** command in global configuration mode. To return the HTTPS server port number to the default, use the **no** form of this command.

ip http secure-port *port-number*

no ip http secure-port

Syntax Description

port-number	Integer in the range of 0 to 65535 is accepted, but the port number must be higher than 1024 unless the default is used. The default is 443.

Command Default

The HTTPS server port number is not set for listening.

Command Modes

Global configuration

Command History

Release	Modification
12.1(11b)E	This command was introduced.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

An HTTP server and an HTTPS server cannot use the same port. If you try to configure both on the same port, the following message is displayed:

```
% Port port_number in use by HTTP.
```

where port_number is the port number that is already assigned to the HTTP server.

If you change the HTTPS port number, clients attempting to connect to the HTTPS server must specify the port number in the URL, in this format:

https://device:port_number

where port_number is the HTTPS port number.

Examples

The following example shows how to assign port 1025 for HTTPS server connections:

```
Router(config)# ip http secure-port 1025
```

Related Commands

Command	Description
ip http secure-server	Enables an HTTPS server.

ip http secure-server

To enable a secure HTTP (HTTPS) server, use the **ip http secure-server** command in global configuration mode. To disable an HTTPS server, use the **no** form of this command.

ip http secure-server

no ip http secure-server

Syntax Description

This command has no arguments or keywords.

Command Default

The HTTPS server is disabled.

Command Modes

Global configuration

Command History

Release	Modification
12.1(11b)E	This command was introduced.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

The HTTPS server uses the Secure Sockets Layer (SSL) version 3.0 protocol.

Note When enabling an HTTPS server, you should always disable the standard HTTP server to prevent unsecured connections to the same services. Disable the standard HTTP server using the **no ip http server** command in global configuration mode (this step is precautionary; typically, the HTTP server is disabled by default).

If a certificate authority (CA) is used for certification, you should declare the CA trustpoint on the routing device before enabling the HTTPS server.

Examples

In the following example the HTTPS server is enabled, and the (previously configured) CA trustpoint CA-trust-local is specified:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# ip http secure-server
Router(config)# ip http secure-trustpoint CA-trust-local
Router(config)# end
```

```
Router# show ip http server secure status

HTTP secure server status: Enabled
HTTP secure server port: 443
HTTP secure server ciphersuite: 3des-ede-cbc-sha des-cbc-sha rc4-128-md5 rc4-12a
HTTP secure server client authentication: Disabled
HTTP secure server trustpoint: CA-trust-local
```

Related Commands

Command	Description
ip http secure-trustpoint	Specifies the CA trustpoint that should be used for obtaining signed certificates for the HTTPS server.
ip http server	Enables the HTTP server on an IP or IPv6 system, including the Cisco web browser user interface.
show ip http server secure status	Displays the configuration status of the HTTPS server.

ip http secure-trustpoint

To specify the certificate authority (CA) trustpoint that should be used for obtaining signed certificates for a secure HTTP (HTTPS) server, use the **ip http secure-trustpoint** command in global configuration mode. To remove a previously specified CA trustpoint, use the **no** form of this command.

ip http secure-trustpoint *trustpoint-name*

no ip http secure-trustpoint *trustpoint-name*

Syntax Description

trustpoint-name	Name of a configured trustpoint. Use the same trustpoint name that was used in the associated **crypto ca trustpoint** command.

Command Default

The HTTPS server uses the trustpoint configured when you use the **primary** command. If a trustpoint is not configured, the HTTPS server uses a self-signed certificate.

Command Modes

Global configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

This command specifies that the HTTPS server should use the X.509v3 certificate associated with the trustpoint indicated by the *trustpoint-name* argument. Use the same trustpoint name that you used in the associated **crypto ca trustpoint** command.

The specified X.509v3 security certificate will be used to authenticate the server to connecting clients, and, if remote client authentication is enabled, to authenticate the connecting clients.

Use this command only if you have already declared a CA trustpoint using the **crypto ca trustpoint** command and associated submode commands. If a trustpoint is not configured, the HTTPS server will use a self-signed certificate.

If this command is not used, the server will attempt to use the certificate associated with the primary trustpoint. The primary trustpoint is configured using the **primary** command.

Examples

In the following example, the CA trustpoint is configured, a certificate is obtained, and the certificate is referenced in the HTTPS server configuration:

```
!The following commands specifies a CA trustpoint that can be used
!to obtain a X.509v3 security certificate.
!A trustpoint NAME is used because there could be multiple trustpoints
!configured for the router.

Router(config)# crypto ca trustpoint tp1
Router(config-ca)# enrollment url http://host1:80
Router(config-ca)# exit
Router(config)# crypto ca authenticate tp1

!The following command is used to actually obtain the security certificate.

Router(config)# crypto ca enrollment tp1
Router(config)# ip http secure-server

!The following command specifies that the secure HTTP server
!should use a certificate associated with the TP1 trustpoint for HTTPS connections.
Router(config)# ip http secure-trustpoint tp1
```

Related Commands

Command	Description
crypto ca trustpoint	Declares the CA that your routing device should use.
ip http secure-server	Enables the HTTPS server.
primary	Assigns a specified trustpoint as the primary trustpoint of the router.
show ip http server secure status	Displays the configuration status of the secure HTTP server.

show ip http client secure status

To display the status of the secure HTTP client configuration, use the **show ip http client secure status** command in privileged EXEC mode.

show ip http client secure status

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Examples

The following is sample output from the **show ip http client secure status** command:

```
Router# show ip http client secure status

HTTP secure client ciphersuite: 3des-ede-cbc-sha des-cbc-sha rc4-128-md5 rc4-12a
HTTP secure client trustpoint: TP-1
```

Table 2 describes the significant fields shown in the display.

Table 2 show ip http client secure status Field Descriptions

Field	Description
HTTP secure client ciphersuite:	Displays the configuration of the **ip http client secure-ciphersuite** command.
HTTP secure client trustpoint:	Displays the configuration of the **ip http client secure-trustpoint** command.

Related Commands

Command	Description
ip http client secure-ciphersuite	Specifies the CipherSuites that should be used for encryption over the secure HTTP connection from the client to a remote server.
ip http client secure-trustpoint	Specifies the CA trustpoint that should be used if the remote HTTP server requests client authentication.

show ip http server secure status

To display the status of the HTTP secure server configuration, use the **show ip http server secure status** command in privileged EXEC mode.

show ip http server secure status

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Examples

The following is sample output from the **show ip http server secure status** command:

```
Router# show ip http server secure status

HTTP secure server status: Enabled
HTTP secure server port: 1025
HTTP secure server ciphersuite: rc4-128-sha rc4-128-md5
HTTP secure server client authentication: Disabled
HTTP secure server trustpoint: CA-trust-local
```

Table 3 describes the significant fields shown in the display.

Table 3 show ip http server secure status Field Descriptions

Field	Description
HTTP secure server status:	Displays the state of secure HTTP server (“Enabled” or “Disabled”). Corresponds to the configuration of the **ip http secure-server** command.
HTTP secure server port:	Displays the configuration of the **ip http secure-port** command.
HTTP secure server ciphersuite:	Displays the configuration of the **ip http secure-ciphersuite** command.

Table 3 show ip http server secure status Field Descriptions (continued)

Field	Description
HTTP secure server client authentication:	Displays the configuration of the **ip http secure-client-auth** command.
HTTP secure server trustpoint:	Displays the configuration of the **ip http secure-trustpoint** command. If no trustpoint is configured, the line will appear blank after the colon.

Related Commands

Command	Description
ip http secure-ciphersuite	Specifies the CipherSuites that should be used for encryption over the secure HTTP connection from the server to a remote client.
ip http secure-client-auth	Configures the HTTP server to authenticate the remote client during the connection process.
ip http secure-port	Specifies the port (socket) to be used for HTTPS connections.
ip http secure-server	Enables the HTTPS server.
ip http secure-trustpoint	Specifies the CA trustpoint that should be used for obtaining signed certificates for the secure HTTP server.

Feature Information for HTTPS—HTTP Server and Client with SSL 3.0

Table 4 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 4 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 4 Feature Information for HTTPS—HTTP Server and Client with SSL 3.0

Feature Name	Releases	Feature Information
HTTPS—HTTP Server and Client with SSL 3.0	12.2(15)T 12.2(33)SRA 12.2(33)SXH	This feature provides Secure Socket Layer (SSL) version 3.0 support for the HTTP 1.1 server and HTTP 1.1 client within Cisco IOS software. SSL provides server authentication, encryption, and message integrity to allow secure HTTP communications. SSL also provides HTTP client authentication. This feature is supported only in Cisco IOS software images that support SSL. Specifically, SSL is supported in “IPSec 56” and “IPSec 3DES” images (contains “k8” or “k9” in the image name) in Cisco IOS Releases 12.2(15)T, 12.2(33)SRA, and 12.2(33)SXH.

Glossary

RSA—RSA is a widely used Internet encryption and authentication system that uses public and private keys for encryption and decryption. The RSA algorithm was invented in 1978 by Ron Rivest, Adi Shamir, and Leonard Adleman. The abbreviation RSA comes from the first letter of the last names of the three original developers. The RSA algorithm is included in many applications, such as the web browsers from Microsoft and Netscape. The RSA encryption system is owned by RSA Security.

SHA—The Secure Hash Algorithm. SHA was developed by NIST and is specified in the Secure Hash Standard (SHS, FIPS 180). Often used as an alternative to Digest 5 algorithm.

signatures, digital—In the context of SSL, "signing" means to encrypt with a private key. In digital signing, one-way hash functions are used as input for a signing algorithm. In RSA signing, a 36-byte structure of two hashes (one SHA and one MD5) is signed (encrypted with the private key).

SSL 3.0—Secure Socket Layer version 3.0. SSL is a security protocol that provides communications privacy over the Internet. The protocol allows client and server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery. SSL uses a program layer located between the Internet's HTTP and TCP layers. SSL is included as part of most web server products and as part of most Internet browsers. The SSL 3.0 specification can be found at *http://home.netscape.com/eng/ssl3/*.

Note See the *Internetworking Terms and Acronyms* document for terms not included in this glossary.