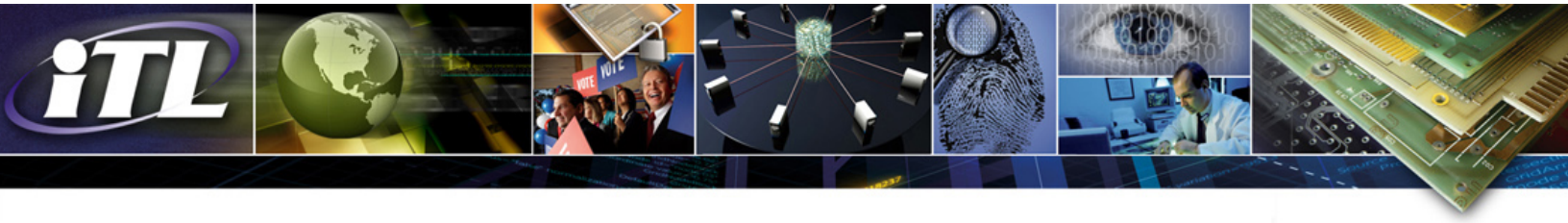


ITL BULLETIN FOR APRIL 2014

RELEASE OF NIST SPECIAL PUBLICATION 800-52 REVISION 1, *GUIDELINES FOR THE SELECTION, CONFIGURATION, AND USE OF TRANSPORT LAYER SECURITY (TLS) IMPLEMENTATIONS*

Kerry McKay, Kim Quill, and Greg Witte, Editors
Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
U.S. Department of Commerce

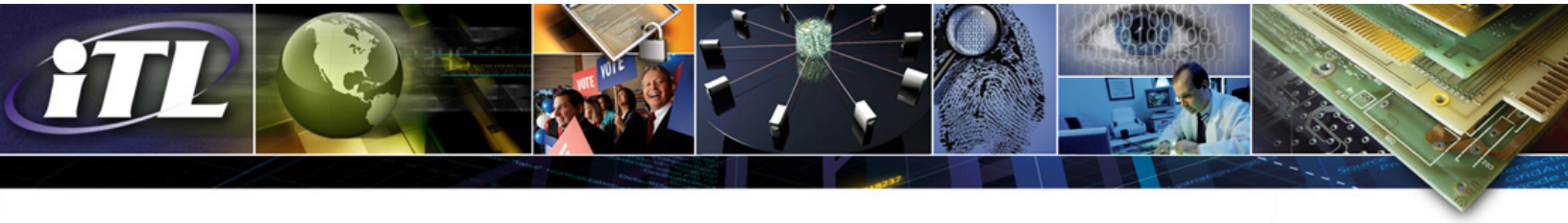
Background

Networked applications often need to protect sensitive information from interception or tampering as data is transmitted over insecure media. Examples of sensitive data include financial data (e.g., credit card numbers, bank information), personally identifiable information (PII) (e.g., passport numbers, Social Security numbers), medical history (e.g., health information, psychological records), and social networking details (e.g., emails, physical locations, private messages.) Technology that secures the transmission information among clients and servers is able to reduce the risk of interception of data and to reduce the likelihood of malicious use of that data.

The *Transport Layer Security (TLS)* Protocol, and its predecessor the *Secure Sockets Layer (SSL)* Protocol, are widely used protocols that protect data from eavesdropping and tampering while being sent across an insecure network. TLS is a standards track protocol in the Internet Engineering Task Force (IETF), first defined in 1999 in Internet Request for Comments (RFC) RFC 2246, *The TLS Protocol Version 1.0*. The protocol was updated in 2006 in RFC 4346, *The TLS Protocol Version 1.1*, and most recently updated again in 2008 in RFC 5246, *The TLS Protocol Version 1.2*. While TLS 1.0 is an improvement over SSL, TLS versions 1.1 and 1.2 fix many vulnerabilities present in TLS 1.0 and are, therefore, more secure.

Introduction to Special Publication 800-52 Revision 1

NIST Special Publication (SP) 800-52, *Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations,* provides guidance to U.S. government information system managers for the selection and configuration of TLS protocol implementations. U.S. Office of Management and Budget (OMB) Circular A-130, *Management of Federal Information Resources*, requires managers of publicly accessible federal systems to ensure that sensitive data is protected. SP 800-52 Revision 1 provides guidelines that focus specifically on the transport layer as described in the Open Systems Interconnection (OSI) model (ISO/IEC 7498-1). TLS is applicable to a variety of situations where clients and servers need to interact and where authentication is performed using public key certificates. SP 800-52 is used in conjunction with other NIST information technology security publications to ensure the protection and security of an entire information system.



NIST published the original version of NIST SP 800-52 in 2005 and specified the requirements for government use of TLS protocols based on TLS 1.0. Several vulnerabilities in TLS version 1.0 were later exposed. IETF subsequently released TLS versions 1.1 and 1.2, and NIST withdrew the original release of SP 800-52 in March 2013. In September 2013, NIST released Draft Revision 1 of SP 800-52, recommending that government system managers move both clients and servers to TLS 1.1 and 1.2. The final version of Revision 1 was released in April 2014 (available at http://csrc.nist.gov/publications) and provides updated recommendations on the use of specific schemes and algorithms for implementing TLS, reflecting updates to NIST's approved cryptographic algorithms and key lengths. The motivations and details for this update, with additional TLS background, are presented below.

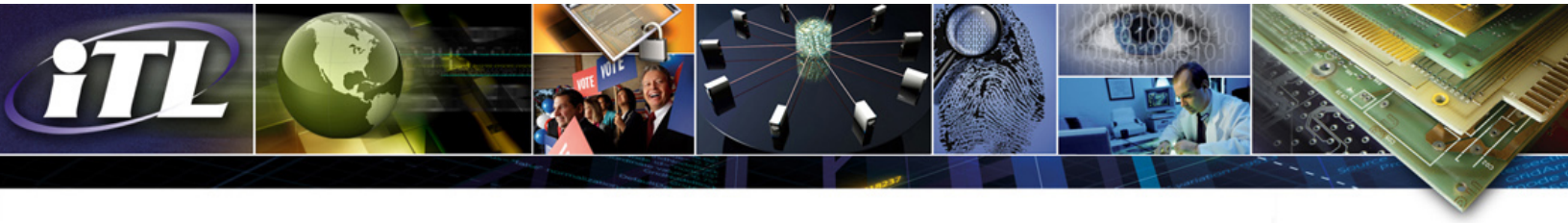
Introduction to Transport Layer Security

TLS works at the Transport Layer of the OSI protocol stack and is composed of the TLS Record Protocol and the TLS Handshake Protocol. The TLS Handshake Protocol is used for client-server authentication and for negotiation of the cipher suite and cryptographic keys that the client and server will use for information exchange. The TLS Record Protocol secures application data using the encryption methods and keys agreed upon during the handshake. Session keys that are determined during the handshake procedure are used to encrypt and decrypt all further messages sent between the client and server, and to verify the integrity of the data until the connection is closed.

Vulnerabilities in Deprecated TLS Versions

Although TLS 1.0 and 1.1 are still widely used, it is important to note that IETF released version 1.0 in 1999, and that portions of it have become outdated and insecure. SP 800-52 describes the manner in which the client and server negotiate algorithms for authentication, confidentiality, and integrity, as well as derive symmetric keys and establish other session parameters, such as data compression. The negotiated set of authentication, confidentiality, and integrity algorithms is known as the ***cipher suite***. NIST SP 800-131A*, Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths,* deprecated and removed a number of algorithms and key lengths that are used in TLS 1.0 and TLS 1.1. Many of the cipher suites used in TLS 1.0 and TLS 1.1 are still allowed, but the user must accept the risk, and that risk will increase over time. For example, to address some of the timing attacks on authenticate and then encrypt structure of TLS records, TLS 1.2 defined block cipher modes of operation that provide authenticated encryption with associated data (AEAD). Examples of such modes are GCM and CCM.

Vulnerabilities in TLS 1.0 have been exposed by a number of demonstrated attacks including the BEAST (Browser Exploit Against SSL/TLS) attack. The BEAST attack is one example of a class of cipher block chaining (CBC) attacks that can be carried out on TLS 1.0. In this attack vector, an adversary injects malicious JavaScript into content returned from a server and then opens a secure connection to a given server, comparing encrypted information with known unencrypted data sent by the injected script. Through this comparison, the attacker is able to learn information about the encrypted session that



results in the ability to decrypt the data, resulting in the disclosure of sensitive data (e.g., usernames and passwords) or to improperly authenticate. Several of the underlying vulnerabilities were remediated with the release of TLS 1.1; TLS 1.2 further improved security by adding AEAD modes of cipher suites. Although TLS 1.2 has been available since 2008, many browsers and websites still do not use this more secure protocol.

The CRIME (Compression Ratio Info-Leak Made Easy) attack takes advantage of a side channel that opens during compression and describes the length of compressed data. In this attack, the attacker injects plaintext content alongside source content before compression. If the injected plaintext content matches the source content, the redundancy will cause the encryption to produce a smaller result. By monitoring the length of the resulting compressed data and observing the size of the resulting ciphertext, an attacker can determine if the some part of the injected content matches some part of the source content. All versions of TLS are susceptible to the CRIME attack; hence SP 800-52 Revision 1 specifies the use of the null compression method, which disables TLS compression.

TLS Handshake

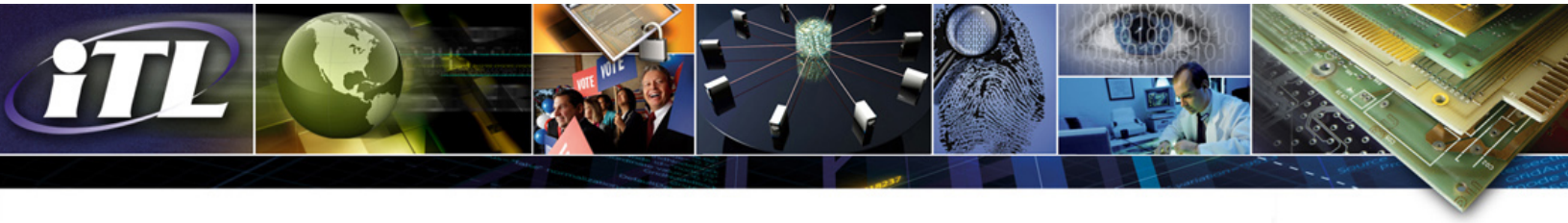
TLS protects the application data by using a set of cryptographic algorithms to ensure the confidentiality, integrity, and authenticity of exchanged application data. The TLS handshake protocol is used to establish and change security parameters, and communicate error and warning conditions to the server and client. Three sub-protocols in the TLS protocol are used to control the session connection:

- The TLS handshake protocol is used to negotiate session parameters;
- The alert protocol is used to notify the other party of an error condition; and
- The change cipher spec protocol is used to change cryptographic parameters.

In addition, the client and the server exchange application data that is protected by the security services provisioned by the negotiated cipher suite. These security services are negotiated and established with the handshake. Clients and servers can be configured so that one or more of the following security services are negotiated during the handshake:

- Confidentiality service provides assurance that data is kept secret;
- Message integrity service provides confirmation that unauthorized data modification is detected, thus preventing undetected deletion, addition, or modification of data;
- Authentication service provides assurance of the sender or receiver's identity; and/or
- Replay protection ensures that an unauthorized user does not capture and successfully replay previous data.

The handshake protocol is used to exchange public key certificates to authenticate the server to the client and to optionally identify the client to the server. The server presents an X.509 public key



certificate to assert its provenance, and, for client-authenticated connections, the client presents its own public key certificate for authentication.

When all security parameters are in place, a message informs the other side to begin using the negotiated security services. Subsequent traffic is secured (i.e., encrypted and/or integrity protected) using the negotiated cipher suite and derived symmetric keys. Additional messages throughout the session provide integrity checking and help to provide assurance of cryptographic integrity – nothing was modified, added or deleted, and all key derivation was performed correctly.

Alerts are used to convey information about the session, such as errors or warnings. For example, an alert can be used to signal a decryption error or that access has been denied. Some alerts are used for warnings, and others are considered fatal and lead to immediate termination of the session. A close_notify alert message is used to signal normal termination of a session. Like all other messages after the handshake protocol is completed, alert messages are encrypted and optionally compressed.

Details of the handshake, change cipher spec, and alert protocols are described in RFC 5246.

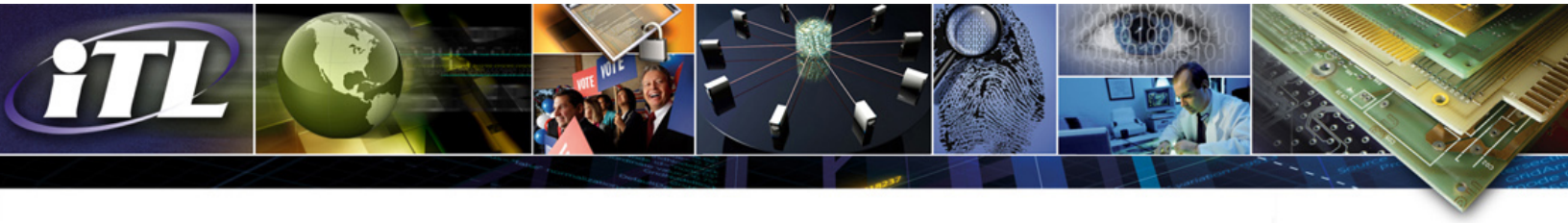
Shared Secret Negotiation

Using the key exchange method agreed upon by the TLS handshake protocol, the client and server determine a pre-master secret. That pre-master secret, along with random values exchanged by the client and server in the “hello” messages, is used to compute a master secret. The master secret, in turn, is used to derive session keys that are used by the negotiated security services to protect the data exchanged between the client and the server. Anti-replay protection is provided, since each packet has a monotonically increasing sequence number.

Session Confidentiality, Integrity, and Authentication

When the current TLS protocol is used in accordance with SP 800-52 Revision 1 guidelines, the application data, as well as the secrets, are protected from attackers who place themselves in the middle of the connection. The attacker cannot modify the handshake messages without being detected by the client and the server because the Finished message, exchanged after security parameter establishment, provides integrity protection to the entire exchange. This enables assurance of the following security attributes:

- Confidentiality is provided for a communication session by the negotiated encryption algorithm to guard against eavesdropping and disclosure of the data transported. Only the client and server know the relevant keys and decrypt the messages using the same key that was used for encryption, derived from the shared master secret;
- The keyed algorithm, specified by the negotiated cipher suite, provides message integrity. Message Authentication Code (MAC) keys are derived from the shared master secret and



provide integrity checks that confirm that no tampering has occurred from outside the secure session;

- Server authentication is performed by the client using the server's public key certificate, which the server presents during the handshake. The exact nature of the cryptographic operation for server authentication is dependent on the negotiated cipher suite and extensions. In most cases, authentication is performed explicitly through verification of digital signatures present in certificates, and implicitly by the use of the server public key by the client during the establishment of the master secret. A successful Finished message implies that both parties calculated the same master secret, and thus the server must have known the private key corresponding to the public key used for key establishment; and
- Client authentication is optional, and occurs only at the server's request. Client authentication is based on the client's public key certificate. The exact nature of the cryptographic operation for client authentication depends on the negotiated cipher suite's key exchange algorithm and the negotiated extensions. For example, when the client's public key certificate contains an RSA public key, the client signs a portion of the handshake message using the private key corresponding to that public key, and the server verifies the signature using the public key to authenticate the client.

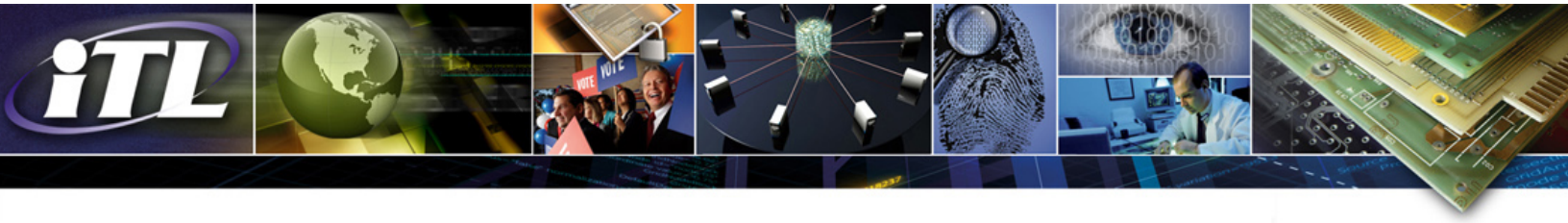
Key Management

The server public key certificate, a corresponding private key, and, optionally, the client public key certificate and corresponding private key, are used in the establishment of the pre-master secret, according to the key exchange algorithm dictated by the selected cipher suite. The pre-master secret, server random value, and client random value are used to determine the master secret, which is then used to derive the symmetric session keys.

The security of the server's private key is critical to the security of TLS. If the server's private key is weak or can be obtained by a third party, the third party can masquerade as the server to all clients. If a third party can obtain a public key certificate for a public key corresponding to their own private key in the name of a legitimate server from a certification authority (CA) trusted by the clients, the third party can masquerade as the server to the clients. Similar threats exist for clients. If a client's private key is weak or can be obtained by a third party, the third party can masquerade as the client to the server. Similarly, if a third party can obtain a public key certificate for a public key corresponding to their own private key in the name of a client from a certificate authority that is trusted by the server, the third party can masquerade as that client to the server.

Server Requirements

As SP 800-52 Revision 1 provides guidance to U.S. government information providers, it describes specific minimum security requirements for managers of publicly accessible information repositories or dissemination systems that contain sensitive but unclassified data. To ensure that sensitive data is



adequately protected, the requirements described make effective use of Federal Information Processing Standards (FIPS) and NIST-recommended cryptographic algorithms. They describe the use of TLS configured with FIPS-based cipher suites as the minimum appropriate secure transport protocol and recommend that agencies develop migration plans to TLS 1.2 by January 1, 2015. This Special Publication also identifies TLS extensions for which mandatory support must be provided and other recommended extensions. While these requirements are mandatory for those federal systems affected by U.S. Office of Management and Budget (OMB) Circular A-130, *Management of Federal Information Resources*, they are useful for all who utilize interconnected networks to share information.

Throughout SP 800-52 Revision 1, the key words "**shall,**" "**shall not,**" "**should,**" and "**should not**" are used to describe requirements. These words are a subset of RFC 2119 key words, and have been chosen based on convention in other normative documents. The key word "Approved" is used to indicate that a scheme or algorithm is described in a Federal Information Processing Standard (FIPS) or is recommended by NIST.

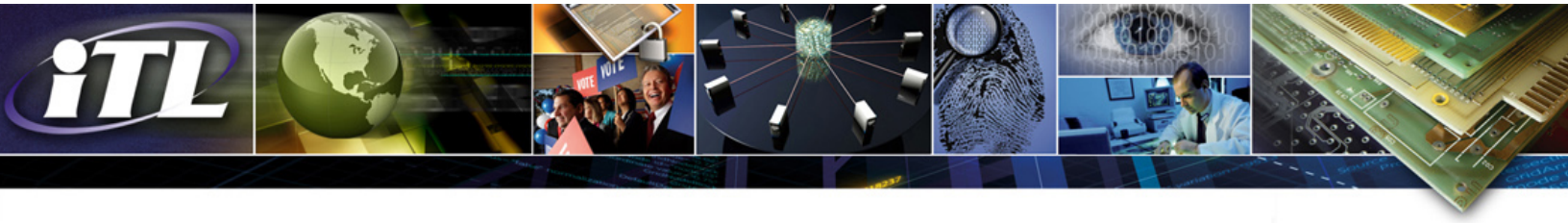
Protocol Version Support

To address vulnerabilities in TLS 1.0, the minimum set of requirements that both a government server and client must implement has been updated with SP 800-52 Revision 1. At a basic level, government servers that support government-only applications **shall** be configured to support TLS 1.1, and **should** be configured to support TLS 1.2. TLS 1.2 support for these systems is mandatory by January 2015, and agencies **shall** develop a migration plan to achieve this goal. These government servers **shall not** support TLS 1.0 or SSL 3.0 or earlier.

Government servers and clients that support citizen or business applications need the capability of interaction with nongovernment entities. These nongovernment systems may not support higher versions of TLS, and therefore government clients and servers that interact with nongovernment systems may support TLS 1.0. However, these systems **shall not** support SSL 3.0 or earlier versions. These systems **shall** support TLS 1.1 and **should** support TLS 1.2, and **shall** prefer the use of later versions of TLS over TLS 1.0. Agencies **shall** develop a plan to support TLS 1.2 by January 2015.

Keys and Certificates

SP 800-52 Revision 1 recommends updates to server keys and certificates. Government servers **shall** be configured with one or more public key certificates that are issued by a trusted Certificate Authority (CA) that publishes revocation information in either a Certificate Revocation List (CRL) or an Online Certificate Status Protocol. The source for a certificate's revocation information will be included in the certificate in the appropriate extension. These systems **should** support multiple server certificates with their associated private keys to maintain flexibility when communicating with clients. To satisfy the requirement for an approved certificate, government servers **shall** at least be configured with an RSA



key encipherment certificate, and **should** be configured with an Elliptic Curve Digital Signature Algorithm (ECDSA) or RSA signature certificate.

The server certificate profile, described in the publication's Section 3.2, provides requirements and recommendations for the format of the server certificate. For these guidelines, the TLS server certificate shall be an X.509 version 3 certificate; both the public key contained in the certificate and the signature shall have at least 112 bits of security. The certificate **shall** be signed with an algorithm consistent with the public key. SP 800-52 Revision 1 Table 3-1 lists the server certificate profile that **should** be used for the server certificate, in the absence of agency-specific certificate profile requirements. The Server Certificate section further describes requirements for revocation checking of the client certificate, when client authentication is used, and for server public key certificate assurance. Notably, it describes specific certificate policy to be used in the absence of agency-specific policies.

Cryptographic Support

TLS provides cryptographic support through the negotiation of various cipher suites during the handshake protocol between server and client. As described above, the client presents its set of supported cipher suites to the server, and the server responds with the preferred selected suite. SP 800-52 Revision 1 designates a number of acceptable cipher suites, which vary based upon the key exchange algorithm, encryption algorithm, and message authentication algorithm used for cryptographic support. Servers and clients **shall** only be configured to support cipher suites with at least 112-bit security strength.

Two suites are specified for mandatory support; both specify that a server or client has been configured with an RSA private key and corresponding certificate, use SHA-1 for generating digital signatures, and has cipher block chaining mode support. Of the two mandatory suites, one supports Triple DES encryption with encrypt decrypt encrypt mode, and one supports Advanced Encryption Standard (AES) 128-bit encryption. SP 800-52 Revision 1 Section 3.3.1 provides a detailed list of required and requested cipher suites for all versions of TLS.

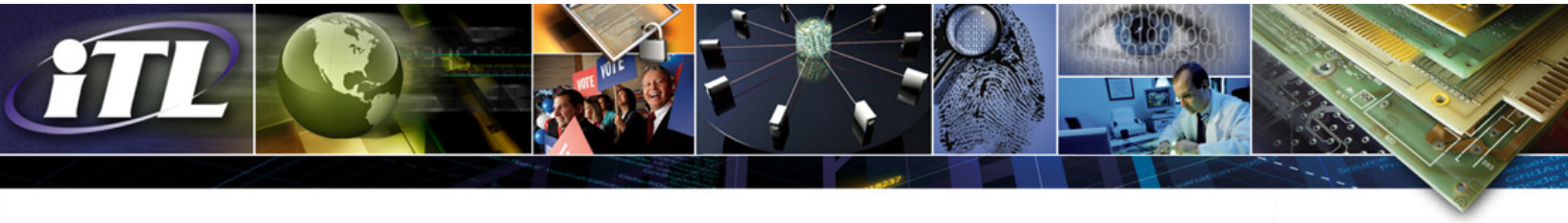
In some situations, such as closed environments, it may be permissible to used pre-shared keys. Pre-shared keys are symmetric keys that are already in place prior to the initiation of a TLS session, which are used in the derivation of the pre-master secret.

TLS Extensions

SP 800-52 Revision 1 designates several of the TLS extensions (described in RFC 6066) as mandatory, conditional, and discouraged. These extensions work to extend the functionality of TLS, focusing on the TLS protocol message formats.

The following extensions are specified as mandatory by SP 800-52 Revision 1:

- TLS Session Renegotiation (Servers and Clients);



- Server Name Indication (Servers and Clients);
- Certificate Status Request (Servers); and
- Trusted CA Indication (Servers).

The following TLS extensions **shall** be supported by government TLS clients and servers, when the conditions in Section 3.4.2 and Section 4.4.2 of SP 800-52 Revision 1 are met:

- Supported Elliptic Curves (Servers and Clients);
- EC Point Format (Servers and Clients);
- Signature Algorithms (Servers and Clients);
- Multiple Certificate Status (Servers and Clients); and
- Certificate Status Request (Clients).

The following TLS extension **should not** be supported by government TLS clients and servers, except in the scenarios described in SP 800-52 Revision 1, Section 4.4.3:

- Client Certificate URL (Clients and Servers).

Client and Server Authentication

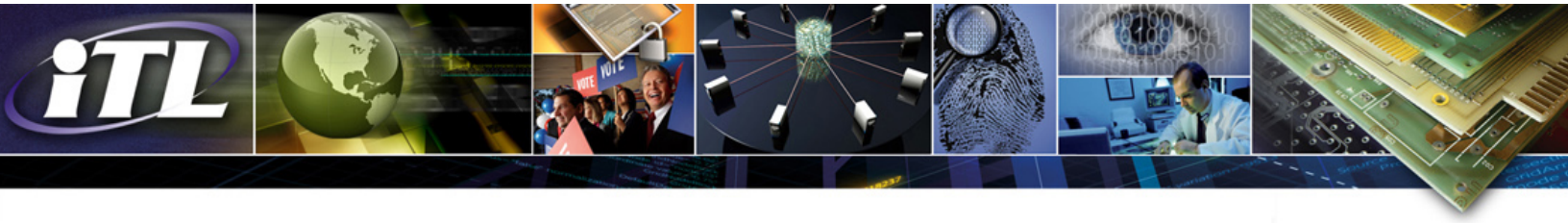
When a TLS server sends its certificate to a client during the handshake, the client must be able to perform path validation to build the certificate chain for the process to move forward. When TLS servers need to cryptographically authenticate a client, they may use the TLS client authentication option to request a client certificate. To authenticate a client, the server sends the client a certificate request message containing a list of trusted certificate authorities (CAs). The exact process of authentication depends on the negotiated cipher suite during earlier steps in the handshake, but the certificates provided must be sufficient for the server to perform path validation to a trusted CA. Both client and server authentication require the authenticating entity to have a trust anchor store containing a minimal list of trusted CAs.

The trust anchor store of a government TLS server or client **shall** only be configured with trust anchors that the server or client trusts. In situations where a server supports client authentication, all trust anchors included in the trust store are required to authenticate clients. The list of trusted CAs is best kept minimal to decrease the chance of using a compromised CA.

The path validation mechanism used during this authentication **shall** be in accordance with the path validation rules specified in RFC 5280. Servers shall validate the revocation status of each certificate in the certification path, determine the certificate policies that the client is trusted for, and be able to provide both the certificate and valid certificate policies to applications for access control decisions.

Session Resumption

After completion of the handshake, both the server and the client store the session ID and cryptographic information used for the session. Often a client will make a request to a server to resume a session. The



master secret is known only to the client and server, so resuming a session is considered a secure mode of operation. If the server is willing, it responds to the request to resume the session with the original session ID and cipher suite at the start of the handshake.

If there is a requirement to authenticate each client as it initiates a session, government TLS servers **shall** be configured to ignore requests to resume a session, generate a new session ID, and restart the handshaking procedure. Similarly, if there is a requirement to authenticate each server as it initiates a connection, government TLS clients shall be configured not to request a session resumption.

Compression Methods

Several attacks on TLS focus on exploiting vulnerabilities created during compression (e.g., CRIME and its successor, BEAST). To prevent these attacks, compression may not be used. SP 800-52 Revision 1 states that only the null compression method, which disables TLS compression, **should** be used. If compression is necessary, only the methods defined in RFC 3749 **shall** be used.

Operational Considerations

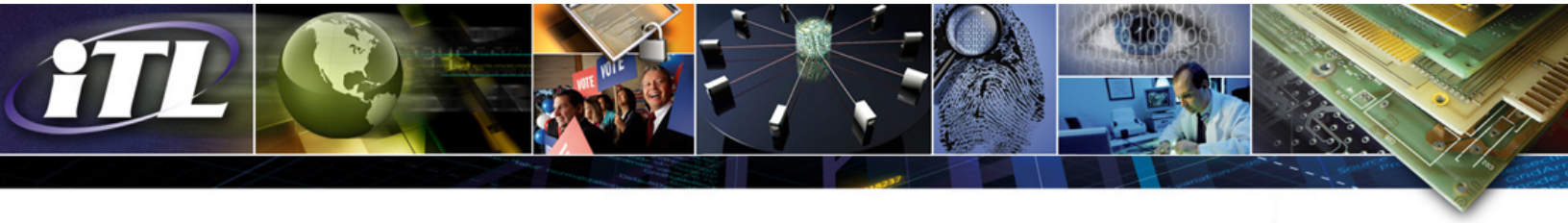
While SP 800-52 specifies functionalities sufficient to achieve security in the transport layer, security across an entire operation environment is necessary for complete security. SP 800-52 requires that federal agencies include appropriate network security protections such as those specified in NIST SP 800-53. Servers must operate on a secure operating system, protecting software and private keys as applicable, and be kept up-to-date in terms of security patches. Similarly, government client platforms must be kept up-to-date with security patches, and incorporate the appropriate security standards to secure systems and applications.

Additional Information in SP 800-52 Revision 1

In addition to the guidance and requirements described above, SP 800-52 Revision 1 provides five appendices. Appendix A provides a list of Acronyms. Appendix B provides guidance on interpreting the names of cipher suites recommended in the publication. Appendix C provides guidelines for the use of pre-shared keys (PSKs); the appendix reminds the reader that PSKs are discouraged, but that their use may be appropriate for some closed environments that have adequate key management support. Appendix D identifies emerging concepts and capabilities that are applicable to TLS. Appendix E lists documents, publications, and organizations that provide a variety of information on aspects of TLS.

Conclusion

The TLS protocol provides a measure of data integrity and confidentiality that may be appropriate to protect public-facing U.S. government systems. Managers of such systems are encouraged to read and understand the specific requirements and recommendations of NIST Special Publication (SP) 800-52 Revision 1. The publication will help readers to fulfill FIPS 200, *Minimum Security Requirements for*



Federal Information and Information Systems, responsibilities to protect confidentiality, integrity, and availability of federal information systems and the information processed, stored, and transmitted by those systems (e.g., minimum security requirements for security-related areas including *Identification and Authentication* and *System and Communications Protection*.) System managers are encouraged to stay aware of system vulnerabilities, such as through the National Vulnerability Database (http://nvd.nist.gov), and to use current versions of security products, where practical.

ITL Bulletin Publisher: Elizabeth B. Lennon
Information Technology Laboratory
National Institute of Standards and Technology
elizabeth.lennon@nist.gov