Advanced Security Protocols - SSL/TLS

Design and Verification of Security Protocols and Security Ceremonies

Programa de Pós-Graduacão em Ciências da Computacão
Dr. Jean Everson Martina

August-November 2016

Programa de Pós-Graduação em Ciência da Computação
UFSC
UNIVERSIDADE FEDERAL
DE SANTA CATARINA

About SSL/TLS

- Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL are both frequently referred to as "SSL";

About SSL/TLS

- Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL are both frequently referred to as "SSL";
- They are cryptographic protocols that provide communications security over a computer network;

About SSL/TLS

- Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL are both frequently referred to as "SSL";
- They are cryptographic protocols that provide communications security over a computer network;
  - Authentication;

About SSL/TLS

- Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL are both frequently referred to as "SSL";
- They are cryptographic protocols that provide communications security over a computer network;
  - Authentication;
  - Confidentiality;

About SSL/TLS

- Several versions of the protocols find widespread use in applications such as web browsing, email, Internet faxing, instant messaging, and voice-over-IP (VoIP);

About SSL/TLS

- Several versions of the protocols find widespread use in applications such as web browsing, email, Internet faxing, instant messaging, and voice-over-IP (VoIP);
- Major websites use TLS to secure all communications between their servers and web browsers;

About SSL/TLS

- Several versions of the protocols find widespread use in applications such as web browsing, email, Internet faxing, instant messaging, and voice-over-IP (VoIP);
- Major websites use TLS to secure all communications between their servers and web browsers;
- Although not common, TLS can be used for mutual authentication.
- TLS aims primarily to provide privacy, data integrity and peer authentication between two communicating computer applications.

TLS History

- TLS is a proposed Internet Engineering Task Force (IETF) standard;

TLS History

- TLS is a proposed Internet Engineering Task Force (IETF) standard;
- It was first defined in 1999 and updated in RFC 5246 (August 2008) and RFC 6176 (March 2011);

TLS History

- TLS is a proposed Internet Engineering Task Force (IETF) standard;
- It was first defined in 1999 and updated in RFC 5246 (August 2008) and RFC 6176 (March 2011);
- It builds on the earlier SSL specifications (1994, 1995, 1996) developed by Netscape Communications for adding the HTTPS protocol to their Navigator web browser.

TLS Properties

- The connection is private because symmetric cryptography is used;

TLS Properties

- The connection is private because symmetric cryptography is used;
- The keys for this symmetric encryption are based on a shared secret negotiated at the start of the session;

TLS Properties

- The connection is private because symmetric cryptography is used;
- The keys for this symmetric encryption are based on a shared secret negotiated at the start of the session;
- The identity of the communicating parties can be authenticated using public-key cryptography;

TLS Properties

- The connection is private because symmetric cryptography is used;
- The keys for this symmetric encryption are based on a shared secret negotiated at the start of the session;
- The identity of the communicating parties can be authenticated using public-key cryptography;
    - This authentication can be made optional, but is generally required for at least one of the parties (typically the server);

TLS Properties

- The connection is private because symmetric cryptography is used;
- The keys for this symmetric encryption are based on a shared secret negotiated at the start of the session;
- The identity of the communicating parties can be authenticated using public-key cryptography;
  - This authentication can be made optional, but is generally required for at least one of the parties (typically the server);
  - Loss or lack of authentication usually leads to major attacks on TLS;

TLS Properties

- The connection ensures integrity;

TLS Properties

- The connection ensures integrity;
- TLS can also provide forward secrecy, ensuring that any future disclosure of encryption keys cannot be used to decrypt any TLS communications recorded in the past.
- Most properties of the protocol are verified by different methods, ensuring no major flaws exist at conceptual level;

TLS Properties

- Nevertheless, SSL/TLS have seen a series of attacks happen in the last 5 years:

TLS Properties

- Nevertheless, SSL/TLS have seen a series of attacks happen in the last 5 years:
  - Beast (Browser Exploit Against SSL/TLS), POODLE (Padding Oracle On Downgraded Legacy Encryption), Sweet32 (attacks on half of the CBC block), Heartbleed (Implementation bug);

TLS Properties

- Nevertheless, SSL/TLS have seen a series of attacks happen in the last 5 years:
  - Beast (Browser Exploit Against SSL/TLS), POODLE (Padding Oracle On Downgraded Legacy Encryption), Sweet32 (attacks on half of the CBC block), Heartbleed (Implementation bug);
  - All of these have to do with implementation of the pre-conditions that fail in practice.

TLS Basics

- TLS consists of two main protocols:

TLS Basics

- TLS consists of two main protocols:
  - Handshake protocol - Use public-key cryptography to establish a shared secret key between the client and the serve;

TLS Basics

- TLS consists of two main protocols:
  - Handshake protocol - Use public-key cryptography to establish a shared secret key between the client and the serve;
  - Record protocol - Use the secret key established in the handshake protocol to protect communication between the client and the server;

TLS Basics

- TLS consists of two main protocols:
  - Handshake protocol - Use public-key cryptography to establish a shared secret key between the client and the serve;
  - Record protocol - Use the secret key established in the handshake protocol to protect communication between the client and the server;
  - We can also have Alert protocol, ChangeCipherSpec protocol and Application protocol, but this a variants of the Handshake or Record protocols;

TLS Basics

- TLS consists of two main protocols:
  - Handshake protocol - Use public-key cryptography to establish a shared secret key between the client and the serve;
  - Record protocol - Use the secret key established in the handshake protocol to protect communication between the client and the server;
  - We can also have Alert protocol, ChangeCipherSpec protocol and Application protocol, but this a variants of the Handshake or Record protocols;
- Formal verification usually focuses on the Handshake and not on the Record;

TLS Basics

- TLS consists of two main protocols:
  - Handshake protocol - Use public-key cryptography to establish a shared secret key between the client and the serve;
  - Record protocol - Use the secret key established in the handshake protocol to protect communication between the client and the server;
  - We can also have Alert protocol, ChangeCipherSpec protocol and Application protocol, but this a variants of the Handshake or Record protocols;
- Formal verification usually focuses on the Handshake and not on the Record;
- Record protocol is a simple symmetric encryption engine.

TLS Record

- The TLS protocol exchanges records, which encapsulate the data to be exchanged in a specific format;

TLS Record

- The TLS protocol exchanges records, which encapsulate the data to be exchanged in a specific format;
- Each record can be compressed, padded, appended with a message authentication code (MAC), or encrypted, all depending on the state of the connection;

TLS Record

- The TLS protocol exchanges records, which encapsulate the data to be exchanged in a specific format;
- Each record can be compressed, padded, appended with a message authentication code (MAC), or encrypted, all depending on the state of the connection;
- Each record has a content type field that designates the type of data encapsulated, a length field and a TLS version field;

TLS Record

- The TLS protocol exchanges records, which encapsulate the data to be exchanged in a specific format;
- Each record can be compressed, padded, appended with a message authentication code (MAC), or encrypted, all depending on the state of the connection;
- Each record has a content type field that designates the type of data encapsulated, a length field and a TLS version field;
- The data encapsulated may be control or procedural messages of the TLS itself, or simply the application data needed to be transferred by TLS.

TLS Record Frame

+	Byte +0	Byte +1	Byte +2	Byte +3
Byte 0	Content type			
Bytes 1..4	Version		Length	
	(Major)	*(Minor)*	*(bits 15..8)*	*(bits 7..0)*
Bytes 5..(*m*-1)	Protocol message(s)			
Bytes *m*..(p-1)	MAC (optional)			
Bytes *p*..(q-1)	Padding (block ciphers only)			

TLS Handshake

- This protocol is used to exchange all the information required by both sides for the exchange of the actual application data by TLS;

TLS Handshake

- This protocol is used to exchange all the information required by both sides for the exchange of the actual application data by TLS;
- The specifications (cipher suite, keys etc.) required to exchange application data by TLS, are agreed upon in the "TLS handshake";

TLS Handshake

- This protocol is used to exchange all the information required by both sides for the exchange of the actual application data by TLS;
- The specifications (cipher suite, keys etc.) required to exchange application data by TLS, are agreed upon in the "TLS handshake";
- This happens between the client requesting the data and the server responding to requests;

Basic TLS Handshake

- A client sends a ClientHello message specifying the highest TLS protocol version it supports, a random number, a list of suggested cipher suites and suggested compression methods;

Basic TLS Handshake

- A client sends a ClientHello message specifying the highest TLS protocol version it supports, a random number, a list of suggested cipher suites and suggested compression methods;
- The server responds with a ServerHello message, containing the chosen protocol version, a random number, CipherSuite and compression method from the choices offered by the client;

Basic TLS Handshake

- A client sends a ClientHello message specifying the highest TLS protocol version it supports, a random number, a list of suggested cipher suites and suggested compression methods;
- The server responds with a ServerHello message, containing the chosen protocol version, a random number, CipherSuite and compression method from the choices offered by the client;
- The server also sends its Certificate message, its ServerKeyExchange message and a ServerHelloDone message, indicating it is done with handshake negotiation;

Basic TLS Handshake

- The client responds with a ClientKeyExchange message, which may contain a PreMasterSecret, public key, or nothing;

Basic TLS Handshake

- The client responds with a ClientKeyExchange message, which may contain a PreMasterSecret, public key, or nothing;
- This PreMasterSecret is encrypted using the public key of the server certificate;

Basic TLS Handshake

- The client responds with a ClientKeyExchange message, which may contain a PreMasterSecret, public key, or nothing;
- This PreMasterSecret is encrypted using the public key of the server certificate;
- The client and server then use the random numbers and PreMasterSecret to compute a common secret, called the "master secret";

Basic TLS Handshake

- The client responds with a ClientKeyExchange message, which may contain a PreMasterSecret, public key, or nothing;
- This PreMasterSecret is encrypted using the public key of the server certificate;
- The client and server then use the random numbers and PreMasterSecret to compute a common secret, called the "master secret";
- All other key data for this connection is derived from this master secret;

Basic TLS Handshake

- The client now sends a ChangeCipherSpec record and an authenticated and encrypted Finished message;

Basic TLS Handshake

- The client now sends a ChangeCipherSpec record and an authenticated and encrypted Finished message;
- The server will attempt to decrypt the client's Finished message and verify the hash and MAC. If the decryption or verification fails, the handshake is considered to have failed and the connection should be torn down;

Basic TLS Handshake

- The client now sends a ChangeCipherSpec record and an authenticated and encrypted Finished message;
- The server will attempt to decrypt the client's Finished message and verify the hash and MAC. If the decryption or verification fails, the handshake is considered to have failed and the connection should be torn down;
- Finally, the server sends a ChangeCipherSpec and an authenticated and encrypted Server Finished message;

Basic TLS Handshake

- The client now sends a ChangeCipherSpec record and an authenticated and encrypted Finished message;
- The server will attempt to decrypt the client's Finished message and verify the hash and MAC. If the decryption or verification fails, the handshake is considered to have failed and the connection should be torn down;
- Finally, the server sends a ChangeCipherSpec and an authenticated and encrypted Server Finished message;
- Application phase starts using TLS record and parameters established at handshake time.

Variations of the Basic TLS Handshake

- The server may require client authentication on his first set of messages;

Variations of the Basic TLS Handshake

- The server may require client authentication on his first set of messages;
- The client will answer a CertificateVerify signed with his private key;

Variations of the Basic TLS Handshake

- The server may require client authentication on his first set of messages;
- The client will answer a CertificateVerify signed with his private key;
- Everything else is the same;

Variations of the Basic TLS Handshake

- The server may require client authentication on his first set of messages;
- The client will answer a CertificateVerify signed with his private key;
- Everything else is the same;
- We can also have a Resumed handshake which avoids the use of asymmetric crypto;

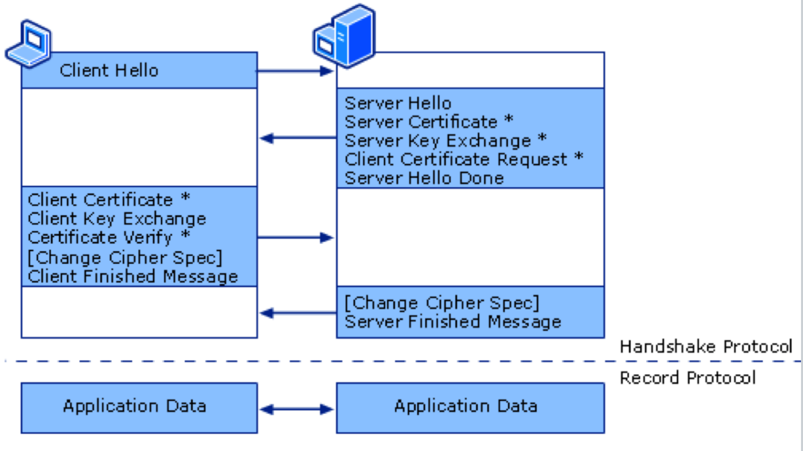
Variations of the Basic TLS Handshake

- The server may require client authentication on his first set of messages;
- The client will answer a CertificateVerify signed with his private key;
- Everything else is the same;
- We can also have a Resumed handshake which avoids the use of asymmetric crypto;
- They are based on sessionId or sessions tickets established at ChangeCipherSpec time;

Variations of the Basic TLS Handshake

- The server may require client authentication on his first set of messages;
- The client will answer a CertificateVerify signed with his private key;
- Everything else is the same;
- We can also have a Resumed handshake which avoids the use of asymmetric crypto;
- They are based on sessionId or sessions tickets established at ChangeCipherSpec time;
- This is very handy for busy server and it heavily depended on forward secrecy.

TLS Handshake



TLS Handshake Frame

+	Byte +0	Byte +1	Byte +2	Byte +3
Byte 0	22			
Bytes 1..4	Version		Length	
	(Major)	*(Minor)*	*(bits 15..8)*	*(bits 7..0)*
Bytes 5..8	Message type	Handshake message data length		
		(bits 23..16)	*(bits 15..8)*	*(bits 7..0)*
Bytes 9..(*n*-1)	Handshake message data			
Bytes *n*..(*n*+3)	Message type	Handshake message data length		
		(bits 23..16)	*(bits 15..8)*	*(bits 7..0)*
Bytes (*n*+4)..	Handshake message data			

TLS Formal Verification

- Wagner and Schneier analysed SSL 3.0 in detail using computational approach;

TLS Formal Verification

- Wagner and Schneier analysed SSL 3.0 in detail using computational approach;
- Sven Dietrich analysed SSL 3.0 using the belief logic NCP (non-monotonic cryptographic protocols);

TLS Formal Verification

- Wagner and Schneier analysed SSL 3.0 in detail using computational approach;
- Sven Dietrich analysed SSL 3.0 using the belief logic NCP (non-monotonic cryptographic protocols);
- Although NCP is a formal logic, Dietrich appears to have generated his lengthy derivations by hand;

TLS Formal Verification

- Wagner and Schneier analysed SSL 3.0 in detail using computational approach;
- Sven Dietrich analysed SSL 3.0 using the belief logic NCP (non-monotonic cryptographic protocols);
- Although NCP is a formal logic, Dietrich appears to have generated his lengthy derivations by hand;
- Mitchell et al. apply model checking to a number of simple protocols derived from SSL 3.0

TLS Formal Verification

- Wagner and Schneier analysed SSL 3.0 in detail using computational approach;
- Sven Dietrich analysed SSL 3.0 using the belief logic NCP (non-monotonic cryptographic protocols);
- Although NCP is a formal logic, Dietrich appears to have generated his lengthy derivations by hand;
- Mitchell et al. apply model checking to a number of simple protocols derived from SSL 3.0
- Most of the protocols are badly flawed (no nonces, for example) and the model checker finds many attacks;

TLS Formal Verification

- Wagner and Schneier analysed SSL 3.0 in detail using computational approach;
- Sven Dietrich analysed SSL 3.0 using the belief logic NCP (non-monotonic cryptographic protocols);
- Although NCP is a formal logic, Dietrich appears to have generated his lengthy derivations by hand;
- Mitchell et al. apply model checking to a number of simple protocols derived from SSL 3.0
- Most of the protocols are badly flawed (no nonces, for example) and the model checker finds many attacks;
- Paulson proved correctness of TLS using Isabelle/HOL and the Inductive method.

Paulson's TLS Formal Verification

- Was part of a big EPSRC grant;

Paulson's TLS Formal Verification

- Was part of a big EPSRC grant;
- Involved at least 3 people;

Paulson's TLS Formal Verification

- Was part of a big EPSRC grant;
- Involved at least 3 people;
- Took around 6 months do be conducted;

Paulson's TLS Formal Verification

- Was part of a big EPSRC grant;
- Involved at least 3 people;
- Took around 6 months do be conducted;
- Is up to-date one of the best abstractions and most precise abstract verification of the suite.

Discussion

- How complex can real-world security protocols become?

Discussion

- How complex can real-world security protocols become?
- How many real world protocols where never subject to proper scrutiny with formal verification?

Discussion

- How complex can real-world security protocols become?
- How many real world protocols where never subject to proper scrutiny with formal verification?
- How come deeply verified and widely deployed protocols can still have problems?

Questions????

UNIVERSIDADE FEDERAL
DE SANTA CATARINA