Global Information Assurance Certification Paper

Transport Layer Security Protocol In WAP

Youngwan Son
Version 2.0 (revised August 13, 2001)

Introduction

The Wireless Application Protocol (WAP) is the de facto worldwide standard for providing Internet communications and advanced telephony services on digital mobile phones, pagers, personal digital assistants and other wireless terminals developed by WAP Forum. WAP is designed in a layered fashion and divided into five layers – Transport, Security, Transaction, Session, and Application. (1)

According to WAP Forum, one of the architectural goals of WAP is as the following: "Provide support for secure and private applications and communication in a manner that is consistent and interoperable with Internet security models". Security forms a fundamental part of the WAP Architecture and appears in the form of Wireless Transport Layer Security (WTLS), the Security Layer of WAP Architecture.

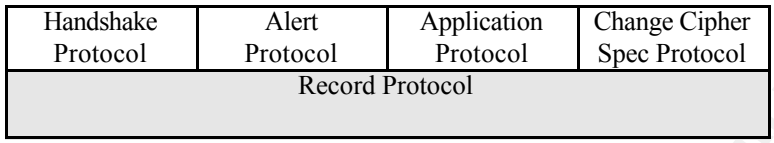
Wireless Transport Layer Security (WTLS)

WTLS is based on the Internet de facto standard Transport Layer Security (TLS) v1.0 and TLS is derived from the widely used Secure Sockets Layer (SSL) 3.0. WTLS is poised to do for the wireless internet what SSL did for the internet and optimized for wireless communication environments. WTLS provides functionality similar to SSL/TLS and incorporates new features such as datagram support, optimized handshake and dynamic key refreshing. Applications are able to selectively enable or disable WTLS features depending on their security requirements and the characteristics of the underlying network like SSL.

Despite of there close resemblance, WTLS has been amended partially to meet the requirements of wireless network. The common requirements set by wireless networks are described below: (2)

- Both datagram and connection oriented transport layer protocols must be supported. It must be possible to cope with, for example, lost, duplicated, or out of order datagrams without breaking the connection state.
- The protocol must take into account that round-trip times with some bearers can be long.
- The slowness of some bearers is a major constraint. Therefore, the amount of overhead must be kept in the minimum.
- The processing power of many mobile terminals is quite limited. This must be taken into account when cryptographic algorithms are chosen.
- The memory capacity of most mobile terminals is very modest. Therefore, the number of cryptographic algorithms must be minimized and small-sized algorithms must be chosen.
- International restrictions and rules for using, exporting, and importing cryptography must be taken into account. This means that it must be possible to achieve the best permitted security level according to the legislation of each area.

The internal architecture of WTLS is shown in [Figure 1] (3). The WTLS Record Protocol is layered protocol and divided into four protocol clients – the handshake protocol, the change cipher spec protocol, the alert protocol and the application data protocol. The application protocol is not described here, since it is the interface for the upper layers.



[Figure 1] Internal architecture of WTLS

When a WTLS client and server first start communication, the Handshake Protocol is initiated. In the handshake procedure between a client and server, all the security related parameters are agreed on. Then the Change Cipher Spec is sent to peer either by the client or the server to notify the other party that subsequent records will be protected under the newly negotiated cipher spec and keys. When an error is detected the detecting party sends an alert message containing the occurred error. The peer decides further procedure depending on the content type of alert messages.

Record Protocol

The Record Protocol takes messages to be transmitted, optionally compresses the data, applies a MAC, encrypts and transmits the result. Received data is decrypted, verified, decompressed and then delivered to higher level clients. Moreover, the Record Protocol takes care of the data integrity and authentication. (4)

Handshake Protocol

The Handshake Protocol is composed of three sub-protocols, which are used to allow peers to agree upon security parameters for the Record Layer, authenticate themselves, instantiate negotiated security parameters, and report error conditions to each other. This layer is responsible for negotiating a secure session and produces the cryptographic parameters described in [Table 1] (2,4).

[Table 1] Attributes negotiated in handshake procedure

Attributes	**Descriptions**
Session Identifier	An arbitrary byte sequence chosen by the server to identify an active or resumable secure
Protocol Version	WTLS protocol version number
Peer Certificate	Certificate of the peer (may not be null)
Compression Method	The algorithm used to compress data prior to encryption
Cipher Spec	Specifies the bulk data encryption algorithm (such as null, RC5, DES, etc.) and a MAC algorithm (such as SHA-1)
Master Secret	20-byte secret shared between the client and server

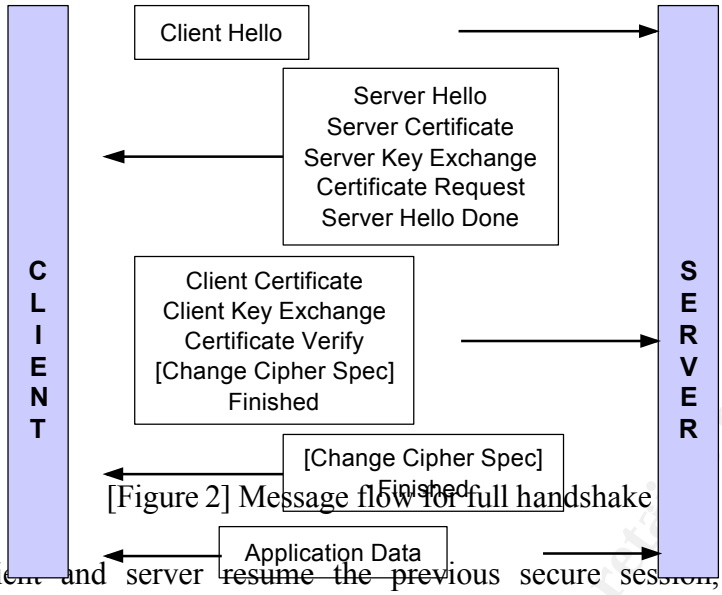
Sequence Number Mode	Which sequence numbering scheme (off, implicit, or explicit) is used to this secure connection
Key Refresh	Defines how often some connection state values (encryption key, MAC secret, and IV) calculations are performed
Is Resumable	A flag indication whether the secure session can be used to initiate new secure connection

The Handshake Protocol involves the following steps: (2)

- Exchange hello messages to agree on algorithms, exchange random values.
- Exchange the necessary cryptographic parameters to allow the client and server to agree on a pre-master secret.
- Exchange certificates and cryptographic information to allow the client and server to authenticate themselves.
- Provide security parameters to the record layer.
- Allow the client and server to verify that their peer has calculated the same security parameters and that the handshake occurred without tampering by an attacker.

There are four types of handshake in WTLS – full handshake, abbreviated handshake, shared-secret handshake, and optimized full handshake. Every handshake starts with exchanging Client Hello message and Server Hello message, which are used to establish security enhancement capabilities between client and server. (2)

In full handshake then the server will send its Server Certificate, if it is to be authenticated. Additionally, a Server Key Exchange message may be sent, if it is required. The server may request a certificate from the client if that is appropriate to the key exchange suite selected. Now the server will send the Server Hello Done message. Then the server will wait for a client response. If the server has sent a certificate request message, the client must send the Client Certificate message. Now the Client Key Exchange message is sent if the client certificate does not contain enough data for key exchange or if it is not sent at all. At this point, the client sends the ChangeCipherSpec message and also the Finished message under the new algorithms, keys, and secrets. In response, the server will also send its own ChangeCipherSpec message, and send its own Finished message under the new cipher spec. At this point, the handshake is complete and the client and server may begin to exchange application layer data. The flow of full handshake is showed in [Figure 2] (2).

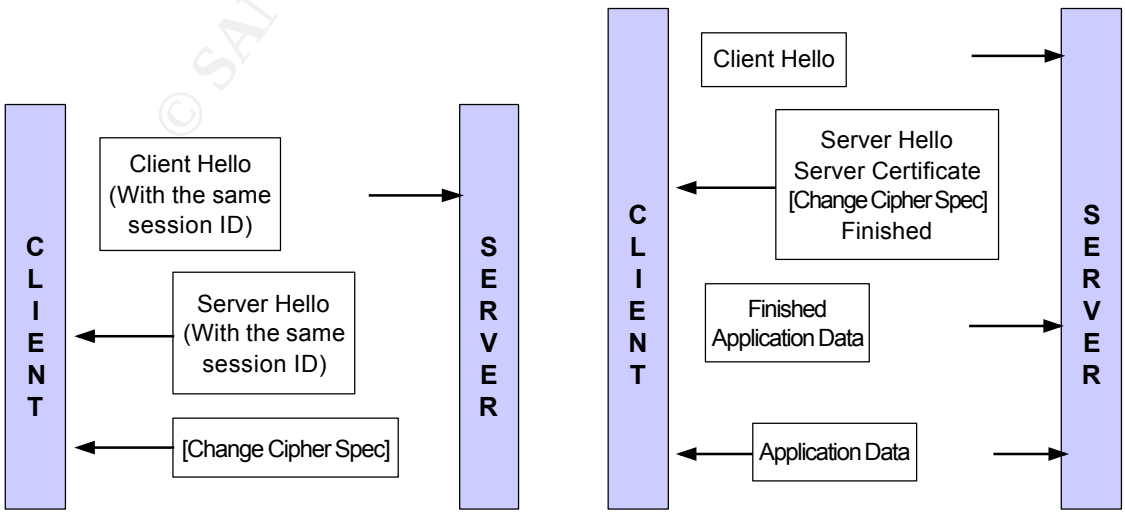


[Figure 2] Message flow for full handshake

When the client and server resume the previous secure session, the abbreviated handshake[Figure 3] (2) is carried on. The Session ID is included in also exchanged during the exchange of Hello messages. The client sends a Client Hello using the Session ID of the secure session to be resumed. The server then checks its secure session cache for a match. If a match is found, and the server is willing to reestablish the secure connection under the specified secure session, it will send a Server Hello with the same Session ID value. Once the re-establishment is complete, the client and server may begin to exchange application layer data.

The shared-secret handshake means that the new secure session is based on a shared secret already implanted in both ends. The shared secret is used as the pre-master secret and the client requests the SHARED_SECRET key exchange suite in Client Hello. This message flow may also be resumable and is similar to the abbreviated handshake.

Another variation is the optimized full handshake[Figure 4] (2) where the server, after receiving the ClientHello, can retrieve client's certificate using a certificate distribution service or from its own sources. The server sends its certificate, a ChangeCipherSpec, and a Finished message. The client responds with a ChangeCipherSpec and Finished message and appication data can now be exchanged.



[Figure 3] Abbreviated handshake [Figure 4] Optimized full handshake

Change Cipher Spec Protocol
The change cipher spec protocol exists to signal transitions in ciphering strategies. The protocol consists of a single message, which is encrypted and compressed under the current connection state. The message consists of a single byte of value 1. (4)

Alert Protocol
Alert messages convey the severity of the message and a description of the alert. There are three types of alert messages: warning, critical, and fatal.

Alert messages, labeled as critical or fatal, result in termination of the current secure connection. Other connections using the secure session may continue. In case of the fatal message, the session identifier must be invalidated so that the failed connection is not used to establish new secure connections. However, in case of the critical message, the session identifier may be used for establishing new secure connections. (2,4)

Security in WTLS
The primary goal of the WTLS layer is to provide privacy, data integrity and authentication for applications in cellular phones and other wireless terminals. The privacy is implemented using strong encryption and MAC algorithms are supported to keep data integrity. Finally, authentication is implemented with key exchange suites and certificates. WTLS also provides dynamic key refreshing which allows encryption keys to be updated on a regular and configurable basis during a secure session. (4)

The available bulk encryption algorithms are RC5 with 40, 56, 64 and 128 bit keys, DES with 40 and 56 bit keys, 3DES, and IDEA with 40, 56, 64 and 128 bit keys. For MAC algorithms such as SHA and MD5 are available. RSA, Diffie-Hellman, or the elliptic curve Diffie-Hellman are available for the key exchange suites and X.509v3, X9.68 and WTLS certificates are supported. (5)

Even though the WTLS protocol is closely modeled after the well studied TLS protocol, a number of potential security problems have been identified - a chosen plaintext data recovery attack, a datagram truncation attack, a message forgery attack, and a key-search shortcut for some exportable keys. The WTLS protocol appears to be more vulnerable to attacks than TLS. (6)

References

(1) WAP Forum. “WAP Architecture Specification”. 12 July 2001. URL: http://www1.wapforum.org/tech/terms.asp?doc=WAP-210-WAPArch-20010712-a.pdf

(2) WAP Forum. “Wireless Transport Layer Security Specification”. 6 April 2001. URL: http://www1.wapforum.org/tech/terms.asp?doc=WAP-261-WTLS-20010406-a.pdf

(3) Espen Kristensen. “WAP Security”. Wireless Security Group of WAP Forum. 3 February 1999. URL: http://www1.wapforum.org/member/developers/slides/WAP-Security/index.htm

(4) Saarinen, Markku-Juhani. "Attacks against the WAP WTLS Protocol". University of Jyväskylä. 1999

(5) Sandra Laquina. "Wireless Application Protocol". September 4 2000. URL: http://www.sans.org/infosecFAQ/wireless/WAP2.htm

(6) Jormalainen, Sami and Laine, Jouni. "Security in the WTLS". Helsinki University of Technology. 10 January 2000.

Last Updated: May 28th, 2017

Upcoming Training

SANS Stockholm 2017	Stockholm, Sweden	May 29, 2017 - Jun 03, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event