

Colloquium on Information Security - June 22, 2000

keyon

WTLS – The security layer in the WAP stack

Martin Christinat, keyon

christinat@keyon.ch

Markus Isler, keyon

isler@keyon.ch

Abstract

The WAP-enabled wireless world represents a huge new market for anyone involved in e-commerce. Security is important as numerous intercept possibilities exist and modifying WAP data is easy due to the nature of the datagram transports used.

WTLS (Wireless Transport Layer Security) is the wireless version of the industry standard Transport Layer Security (TLS), the successor to the widely used Secure Sockets Layer (SSL). Using WTLS, all data exchanged between the WAP device and the WAP gateway is secured. This protects sensitive information while the data is sent through a number of different networks, including over the air, phone lines and IP networks.

In this talk keyon will introduce the WTLS protocol that was developed based upon the need to support datagrams in a high latency, low bandwidth environment. An overview of the current WAP security model and future models is given. Real life problems with WAP security and problems in the WTLS protocol and its implementations as well as possible attacks are discussed.

Martin Christinat

Founder and CTO of keyon. During spring 2000, Martin Christinat worked on the design and implementation of a WTLS layer in Java. He also designed and implemented the keyon / WTLSGateway product, a WTLS security gateway based on the implemented WTLS layer.

Markus Isler

Founder and President of keyon. During spring 2000, Markus Isler worked on the design and implementation of a WTLS layer in Java.

keyon

Founded in December 1999, keyon is a privately held company based in Rapperswil, Switzerland. keyon was founded by three former employees of r3 security engineering ag (now Entrust Technologies (Schweiz) GmbH) and is focused on IT security and applied cryptography. Visit www.keyon.ch for more information.

Agenda

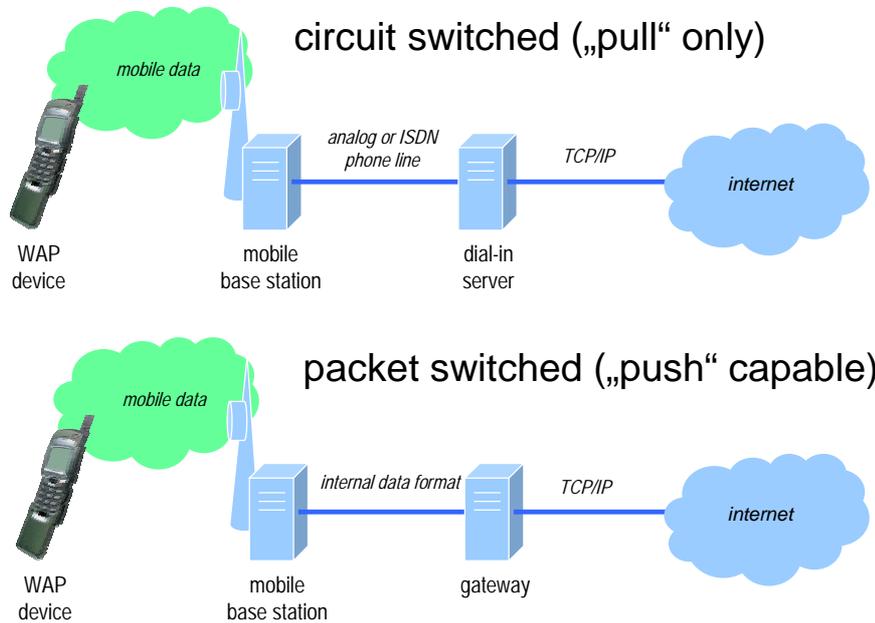
keyon

- Introduction
 - the wireless environment and the WAP model
 - the need for security and challenges
- The WTLS Protocol
 - elements of the protocol
 - how it works
- Practical aspects of today's WTLS security
 - Problems and attacks
 - Implementations
- Summary



The wireless environment (simplified)

keyon



There are numerous ways to transfer data in mobile environments. We distinguish between circuit switched services, in which the mobile device connects to a dial-in server, i.e. a modem server, and packet switched services, in which the mobile device is permanently connected to a network.

In the Circuit Switched Data (CSD) model, a mobile phone calls a dial-in server either by using an ISDN (V.110) or an analog modem protocol. In a GSM network, the data transfer speed using CSD is limited to 9.6 kbit/s. The High Speed Circuit Switched Data (HSCSD) model is exactly the same, only the data transfer rate is with up to 57.6 kbit/s six times as high. This is achieved by bundling several GSM channels for the data transfer. While HSCSD may allow to transfer data six times faster, it is also considerable more expensive. The connection is always initiated by the mobile device.

Packet switched services differ from the CSD model as the mobile device is always connected to the network and data is transmitted in packets. The best known packet switched service is SMS. In fact WAP allows using SMS for the data transfer, but due to the limitation to 160 characters and thus the requirement of sending several messages for a larger transfer unit it is not widely used for WAP. The General Packet Radio Service (GPRS) which will be launched shortly does not have the limitations of SMS and provides data transfer rates up to 171 kbit/s for fast access to the Internet or WAP services. Packet switched services can be used for push services as the mobile device is always on the network. A service provider may thus initiate a data transfer.

GPRS will be one of the enabling factors for WAP as it will allow fast and efficient data transfers with a volume based pricing.

Challenges of the wireless environment

keyon

- Limited bandwidth with current systems
- Latency
 - Round trip time may be high
- Dropouts
 - Weak signals
 - Unexpected loss of connection
- Datagram transfer
 - Forced when packet switched services are used
- Limited resources on the client available
 - Special transfer protocols are needed

The current data transfer rate in a GSM system is 9.6 kbit/s for circuit switched data. When SMS messages are used for data transfer, we may have a transfer rate as low as 160 bytes per second or even less. GPRS and the next generation mobile system UMTS will resolve the bandwidth issue.

There numerous internal protocols and gateways involved in the wireless environment introduce latencies in the data transfer. This is especially true if SMS messages are used as a means of transport.

One of the problems with wireless data transfer are dropouts due to the loss of reception, weak signals or interference while moving the mobile device.

As data transfer should work with all services, connection and packet oriented, datagram transport must be used. It is up to upper protocol layers to provide reliability.

Current internet protocols are not suitable for WAP. They have too much overhead and HTML for example is way too complicated to be rendered on a very small display using only limited processing power. As WAP is intended for all possible data transfer means including even SMS, it was designed as a lightweight protocol, every unnecessary bit was stripped and the lower layers are special for each of the so called bearers like GSM.

Why a security layer is needed

keyon

- WAP can be used for sensitive applications, e.g. banking or brokerage
- Available security in GSM systems applies only to the over the air channel
 - The GSM specific encryption algorithm A5/1 (or the weaker A5/2) is only used for securing data transfers over the air
 - A5/x strength is questionable
 - GSM Security is selected by the service provider
- Roaming features
 - The flow of the transferred data is not controllable when roaming is used

If the mobile device is only used as a modem, the security is handled by the application, e.g. the TLS layer in the browser. In the WAP model, the browser runs on the mobile device. As WAP client devices are usually personal devices, they are suitable for applications like banking or brokerage.

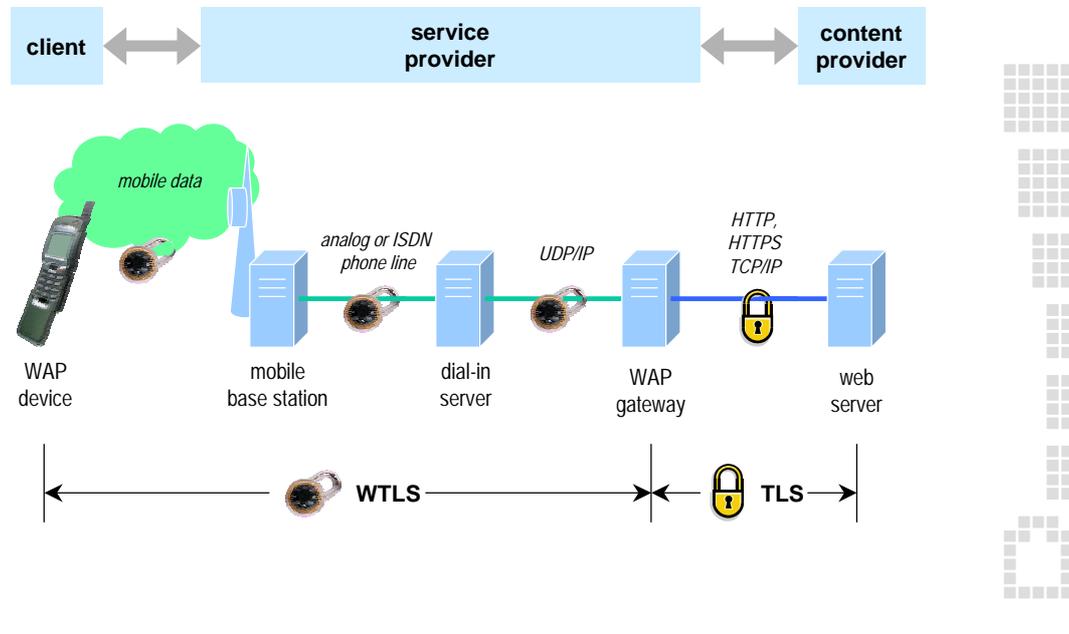
The data is transferred over different channels: Over the air, over the phone line and over IP networks. The GSM security A5/1, A5/2 is not enough as it is limited to the over the air channel. The strength of the A5/x algorithms is questionable, especially as they were never disclosed officially in public. The weaker A5/2 can be broken in real time, recent analysis showed that the stronger A5/1 can be broken within seconds using a PC with 128 MB RAM and a 73 GB hard disk. Also this kind of security is selected by the provider and almost none of the phones show security state during a call.

This all leads to the conclusion that if sensitive information is transferred over WAP, a new security protocol is needed.

Security solution used in WAP

keyon

CSD model



WAP introduces a new piece of software, the WAP gateway. It's purpose is to translate the WAP requests to HTTP requests and translate the received response from the web server into a binary form. All data transferred is in a compact compiled form, i.e. tags are replaced by tokens.

The security layer defined in the WAP stack is called wireless transport layer security (WTLS). WTLS is not compatible to the TLS protocol used on the web. The WAP gateway must therefore decrypt and re-encrypt data passing through it. The use of WTLS is optional.

Why this model?

The content provider does not need new hard- or software. He must however provide the content in WML format. WML (Wireless Markup Language) is based on XML.

The service provider operating the mobile base equipment also operates the WAP gateway. The gateway translates the WML pages into a format suitable for the WAP device.

WAP devices have only limited resources and little memory. Using tokenized pages lowers the requirements on the client side. Unlike TLS, the WTLS protocol is designed for devices with low computational power and little memory.

As one can see, locating the WAP gateway at the provider has the advantage that a content provider does not need new software. The drawback on the other hand is that we have no real end-to-end security between the client and the content provider.

WTLS protocol

keyon

- Based upon TLS (SSL v 3.1)
- Intended for WAP transport protocols
- Optimised for small bandwidth and long latency
- Features:
 - Privacy, data integrity, authentication
 - Datagram support
 - Optimised handshake
 - Dynamic key refreshing



WTLS is a security protocol based upon the industry-standard Transport Layer Security (TLS) protocol, formerly known as Secure Sockets Layer (SSL).

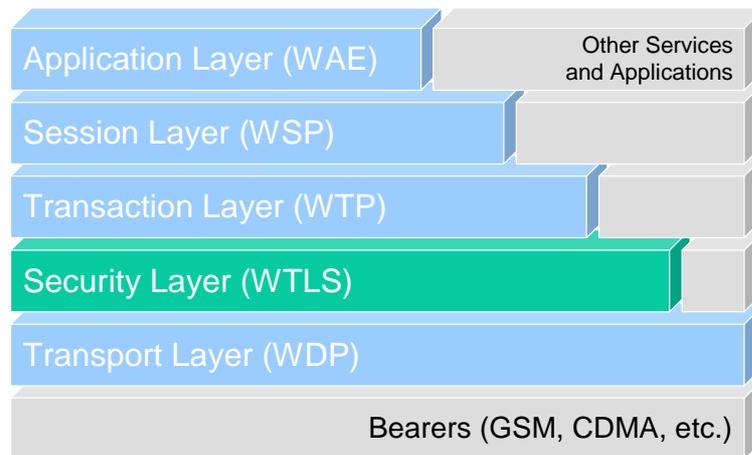
WTLS is intended for use with the WAP transport protocols and has been optimised for low-bandwidth bearer networks with relatively long latency.

The primary goal of the WTLS layer is to provide privacy, data integrity, and authentication between two communicating applications.

WTLS provides functionality similar to TLS 1.0 and incorporates new features such as datagram support, optimised handshake, and dynamic key refreshing.

WAP layer model

keyon



The WTLS layer operates above the transport protocol layer.

The WTLS layer is modular and it depends on the required security level of the given application whether it is used or not.

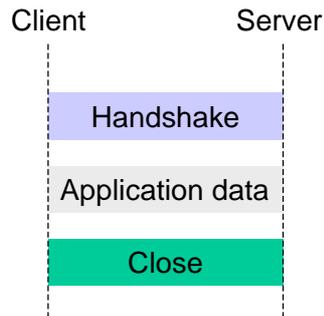
WTLS provides the upper-level layer of WAP with a secure transport service interface that preserves the transport service interface below it.

In addition, WTLS provides an interface for managing (e.g., creating and terminating) secure connections.

WTLS protocol layers

keyon

- Handshake protocol
 - Connection management
 - Client / server authentication
 - Key exchange
- Record layer
 - Privacy
 - Data integrity
- Alert layer
 - Alert management
 - Close handling



The WTLS protocol is composed of three layers.

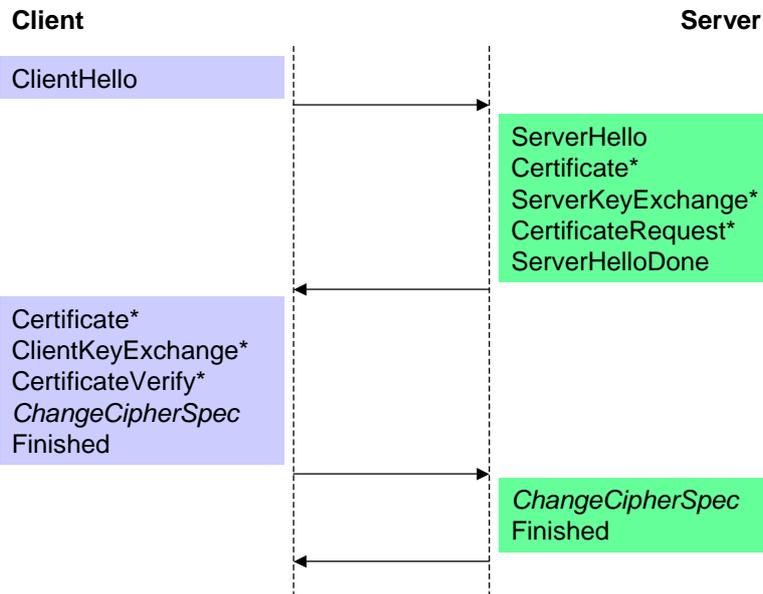
The WTLS handshake protocol manages secure connections, provides client and server authentication and is used to exchange key material.

The WTLS record layer provides privacy and data integrity.

The alert layer is used to report error conditions to each other and to handle the close alert.

Full handshake

keyon



The cryptographic parameters of the secure session are produced by the WTLS Handshake Protocol, which operates on top of the WTLS Record layer. When a WTLS client and server first start communicating, they agree on a protocol version, select cryptographic algorithms, optionally authenticate each other, and use public-key encryption techniques to generate a shared secret.

The WTLS Handshake Protocol involves the following steps:

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.

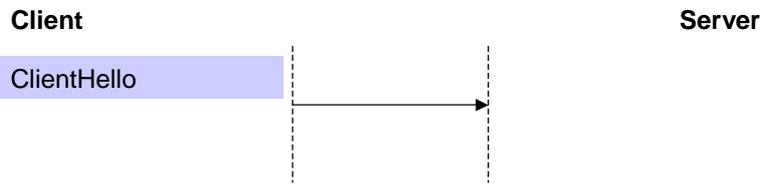
Generate a master secret from the pre-master secret and exchanged random values.

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.

ClientHello

keyon



when this message will be sent:

- first message when client connects to server

purpose of this message:

- initiate secure connection
- provide random for session key calculation
- identify session to resume
- tell sever cryptographic and other capabilities



When a client first connects to a server it is required to send the client hello as its first message. The client can also send a client hello in response to a hello request or on its own initiative in order to renegotiate the security parameters in an existing secure connection.

The purpose of this message is to initiate a secure connection with a server.

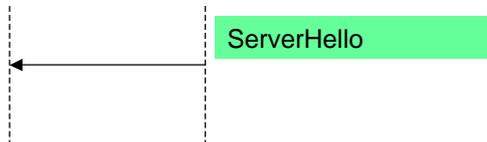
The client provides a random value that is used to calculate the session key, a session id that identifies a previously established secure connection with this server and a list of cryptographic algorithms and other protocol options.

ServerHello

keyon

Client

Server



when this message will be sent:

- in response to a ClientHello message

purpose of this message:

- provide random for session key calculation
- provide unique session ID to identify session
- select cryptographic algorithms (key exchange, cipher, hash)
- select other options (compression, key refresh)



The server will send this message in response to a client hello message when it was able to find an acceptable set of algorithms. If it cannot find such a match, it must respond with a handshake failure alert.

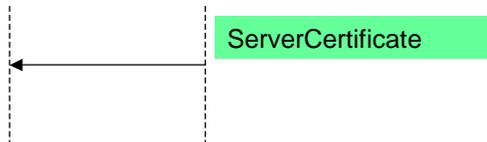
The purpose of this message is to provide random for the session key calculation, provide a unique session ID to identify the session and to select suitable cryptographic algorithms and other protocol parameters.

ServerCertificate

keyon

Client

Server



when this message will be sent:

- if sent this message must immediately follow a ServerHello message

purpose of this message:

- provide server certificate for key exchange
- X.509 v3, WTLS v1 or X9.68 certificates



If sent this message must always immediately follow the server hello message.

The purpose of this message is to provide the server certificate for the key exchange.

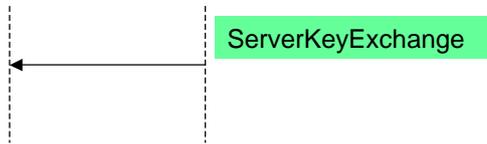
The certificate type must be appropriate for the selected key exchange suite's algorithm. It can be a X.509v3 certificate, a WTLS certificate that is optimised for size or a X9.68 certificate.

ServerKeyExchange

keyon

Client

Server



when this message will be sent:

- this message is sent after a ServerHello message

purpose of this message:

- provide cryptographic information for key exchange for anonymous key exchange methods



This message will be sent immediately after the server hello message.

The purpose of this message is to provide additional cryptographic information to allow the client to communicate the pre-master secret. This is true for the anonymous key exchange methods.

Key exchange methods

keyon

- anonymous
 - RSA key exchange
 - Diffie-Hellman key exchange
 - EC Diffie-Hellman key exchange
- with authentication
 - RSA key exchange with RSA based certificates
 - EC Diffie-Hellman key exchange with ECDSA based certificates
- others
 - shared secret
 - no key exchange



The WTLS protocol defines the following key exchange algorithms:

Anonymous key exchange algorithms based upon RSA and Diffie-Hellman key exchange

Key exchange algorithms with authentication based upon RSA key exchange with RSA based certificates or EC Diffie-Hellman key exchange with ECDSA based certificates

Symmetric-key based handshake where both parties share a secret key that is used as the pre-master key as such.

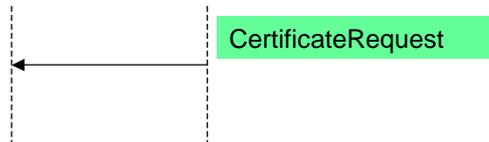
no key exchange

CertificateRequest

keyon

Client

Server



when this message will be sent:

- this message is sent immediately after a ServerCertificate or ServerKeyExchange message

purpose of this message:

- request a certificate from the client for client authentication
- send a list of acceptable certificate authorities



A server can optionally request a certificate from the client, if appropriate for the selected cipher suite. This message, if sent, will immediately follow the Server Certificate message and Server Key Exchange message (if sent).

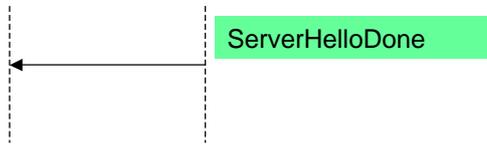
The purpose of this message is to request a certificate from a client for client authentication and to define a list of acceptable certificate authorities.

ServerHelloDone

keyon

Client

Server



when this message will be sent:

- this message is sent to indicate the end of the ServerHello and associated messages

purpose of this message:

- server is done to send messages to support key exchange
- client can proceed with its phase of key exchange

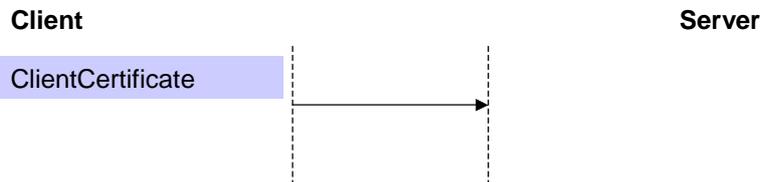


The server hello done message is sent by the server to indicate the end of the server hello and associated messages. After sending this message the server will wait for a client response.

This message means that the server is done sending messages to support the key exchange, and the client can proceed with its phase of the key exchange.

ClientCertificate

keyon



when this message will be sent:

- this message is sent after the client received a ServerHelloDone message only if the server request a client certificate

purpose of this message:

- provide client certificate for client authentication
- X.509 v3, WTLS v1 or X9.68 certificates



This message from the client can be sent after receiving a server hello done message. This message is only sent if the server requests a certificate.

The purpose of this message is to send the client certificate to the server for client authentication.

If no suitable certificate is available, the client must send a certificate message containing no certificates. If client authentication is required by the server for the handshake to continue, it may respond with a fatal handshake failure alert.

The certificate type must be appropriate for the selected key exchange suite's algorithm. It can be a X.509v3 certificate, a WTLS certificate that is optimised for size or a X9.68 certificate.

ClientKeyExchange

keyon



when this message will be sent:

- this message follows the ClientCertificate message, if sent

purpose of this message:

- set the pre-master-secret
 - through direct transmission of RSA-encrypted secret
 - by transmission of Diffie-Hellman public key



This message will immediately follow the client certificate message, if it is sent. Otherwise, it will be the first message sent by the client after it receives the server hello done message.

With this message, the pre-master secret is set, either through direct transmission of the RSA-encrypted secret, or by the transmission of EC Diffie-Hellman public key which will allow each side to agree upon the same pre-master secret.

CertificateVerify

keyon



when this message will be sent:

- follows the ClientKeyExchange message, if sent
- only sent if the client has signing capability

purpose of this message:

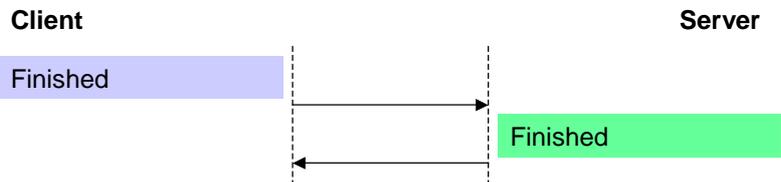
- provide explicit verification of a client certificate



This message is only sent by the client following a client certificate that has signing capability (i.e., RSA certificates) and will immediately follow the client key exchange message. This message is used to provide explicit verification of a client certificate.

Finished

keyon



when this message will be sent:

- at the end of the handshake

purpose of this message:

- verify that the key exchange and authentication processes were successful
- the Finished message is the first message protected with the just-negotiated algorithms, keys and secrets



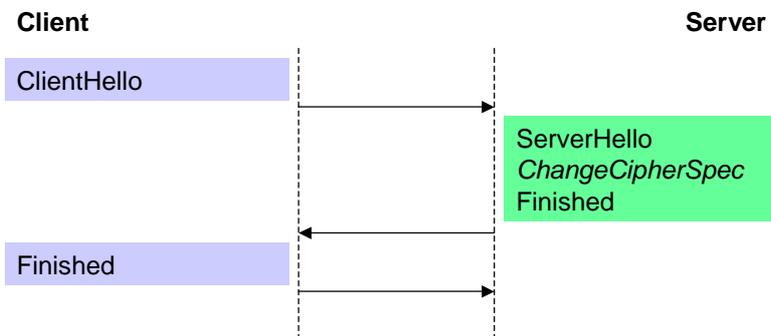
A finished message is always sent at the end of the handshake.

The purpose of this message is to verify that the key exchange and authentication processes were successful.

The finished message is the first protected with the just-negotiated algorithms, keys, and secrets.

Abbreviated handshake

keyon



- client provides session ID of the previous secure session to be resumed
- server accepts to re-establish the secure connection

The client sends a ClientHello using the session id of the secure session to be resumed.

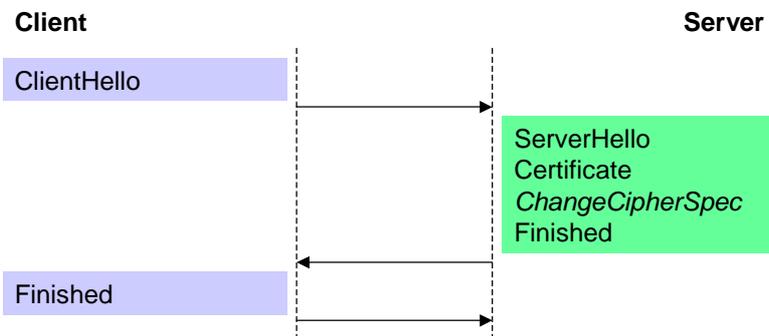
The server then checks its secure session cache for a match. If a match is found, and the server is willing to re-establish the secure connection under the specified secure session, it will send a ServerHello with the same session id value.

Once the re-establishment is complete, the client and server may begin to exchange Application layer data.

If a session id match is not found, the server generates a new session id and the WTLS client and server perform a full handshake.

Optimised handshake

keyon



- server gets client certificate from certificate distribution service or own source
- EC Diffie-Hellman parameters provided in the certificates allow calculating pre-master-secret

Unlike in the full handshake, the server looks up the client certificate from its own source without requesting it over the air from the client.

If the EC Diffie-Hellman parameters are provided in the certificates, the server can calculate the pre-master secret and master secret at this point. In this case, the server sends its certificate, a ChangeCipherSpec, and a Finished message.

Record layer

keyon

- Data compression
 - currently no compression is supported
- Data integrity
 - calculates/verifies message authentication code
 - HMAC based: SHA-1, MD5
 - XOR based: SHA-1
- Privacy
 - encrypts/decrypts data using bulk ciphers
 - bulk cipher algorithms: RC5, DES, 3DES, IDEA



The primary goal of the record layer is to provide data compression, data integrity and privacy.

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, and decompressed, then delivered to higher-level clients.

Record layer modifications

keyon

- no fragmentation
 - transport layer takes care of necessary fragmentation and reassembly
- explicit sequence numbering
 - must be used with datagram transport protocol
 - records can be lost, duplicated or out of order
- sliding window
 - keep books on received messages to discard duplicated records



Unlike in TLS, the record layer does not fragment information blocks. It is assumed that the transport layer takes care of the necessary fragmentation and reassembly.

Explicit sequence numbering must be used with a datagram transport protocol, meaning that records can be lost, duplicated, or received out of order.

The receiver must keep books about received records in order to discard duplicated records. This can be implemented using a sliding window.

Alert protocol

keyon

- Closure alerts
 - connection/session closed
- Error alerts
 - warning, critical, fatal
 - fatal error → close secure connection, invalidate session id
 - critical error → close secure connection
 - MAC error is only a warning



The client and the server must share knowledge that the secure connection is ending. Either party may initiate the exchange of closing messages.

Upon transmission or receipt of a fatal alert message, both parties immediately close the secure connection. Servers and clients are required to forget any session identifiers, keys, and secrets associated with a failed secure connection.

Upon transmission or receipt of a critical alert message, both parties immediately close the secure connection but may preserve the session identifiers and use that for establishing new secure connections.

A MAC error is only a warning and not an error. If the MAC verification fails, the received record is discarded but the secure connection is kept open to prevent from denial of service attacks.

Implementation classes

keyon

<i>Feature</i>	<i>Class 1</i>	<i>Class 2</i>	<i>Class 3</i>
Public-key exchange	M	M	M
Server certificates	O	M	M
Client Certificates	O	O	M
Shared-secret handshake	O	O	O
Compression	-	O	O
Encryption	M	M	M
MAC	M	M	M
Smart card interface	-	O	O

M - mandatory
O - optional



WTLS implementations may have support for various features.

A class may have mandatory (M) or optional (O) support for a certain feature.

Certain features are not yet defined in the current version of the specification.

Cryptographic aspects

keyon

- XOR-MAC attack
 - Proprietary message authentication code algorithm
- Key-exchange attacks
 - Key exchange, encryption and MAC algorithm negotiated independently
- CBC calculated for each record
 - Sequence numbers guarantee order of records
- Key refresh
 - Makes crypto analysis less attractive



There is a known MAC attack on the proprietary XOR message authentication code defined in the WTLS protocol.

Key exchange, encryption and MAC algorithm negotiated independently. This leads to a very high flexibility which allows nonsense combinations, e.g. NULL key exchange with integrity or confidentiality protection.

CBC is calculated for each record as the order of the records is not guaranteed. The order of the records is guaranteed by using sequence numbers.

In WTLS many connection state parameters can be recalculated during a secure connection. This feature is called the key refresh. It is performed in order to minimize the need for new handshakes. In the key refresh, the values of MAC secret, encryption key, and IV will change due to the sequence number. Key refresh makes crypto analysis less attractive.

Outlook

keyon

- Client authentication
 - Implementation class 3
 - SIM (WIM) cards needed that hold the personal credentials
- WIM (wireless identity module)
 - kernel of WTLS security
 - specifies storage of personal credentials (i.e. private keys, certificates)
 - performs optimised cryptography during handshake, especially for client authentication



In implementation class 3 client authentication is mandatory.

As the private key and certificate are personal data they have to be stored on the SIM (WIM) card rather than on the phone itself.

The kernel of WTLS security is the WIM (Wireless Identity Module).

WIM specifies storage of personal credentials (i.e. private keys, certificates) on tokens such as SIM (WIM) cards.

The WIM performs optimised cryptography during handshake, especially for client authentication, and forges long-term, secure WTLS connections.

WTLS in practice

keyon

- No end-to-end security with the web server
 - All data is available in plain in the WAP gateway
 - Client authentication only available to the WAP gateway
- No minimal set of cipher suites defined
 - No guaranteed common set of algorithms between client and WAP gateway
- Nonsense algorithm combinations possible
- WTLS must implement reliability
 - Security unrelated elements in the WTLS layer
- WTLS is not a mandatory layer in the WAP stack



WTLS security is only used between the WAP device and the WAP gateway. Between the WAP gateway and the web server, SSL is the security protocol used. While this model does not require additional software at the content provider side, it lacks the possibility of end-to-end security unless the secure WAP gateway is not located at the service provider. The data is always decrypted and re-encrypted in the WAP gateway. While this may be sufficient for some e-commerce applications, it is not well suited for banking and brokerage. Especially client authentication does not provide any additional level of security for the content provider. The WAP gateway is the only entity which can verify the client using client certificates. The web server of the content provider must therefore ultimately trust the WAP gateway.

WAP 1.1 and 1.2 does not define a mandatory cipher suite. Two key exchange suites using certificates are defined: RSA key exchange and EC Diffie-Hellman key exchange are possible. For WTLS Class 2 compliant devices, only one of those two is needed.

The location of the WTLS layer in the WAP stack requires security unrelated actions in the implementations such as handling of duplicates, retransmissions if no response is received within a certain time frame. Also the close alert is often not received as the connection is physically closed before the alert is transmitted.

Attacking WTLS

keyon

- Attacks to gain access to the exchanged data
 - Man-in-the-middle attacks as at least one anonymous key exchange is mandatory in WTLS
- Denial of service (DoS) attacks
 - Do not allow access to exchanged data
 - Prevent users from accessing a service
 - Interrupt existing secure connections



Any WTLS compliant client device must implement one of the anonymous handshakes. The problem is that mobile phones are devices which must be fool proof. Mobile phone developers avoid displaying any information that could possibly confuse a customer at any price. Unlike on a normal computer screen, there is not enough space on a small phone screen for explanatory comments, so they tend to leave information away.

The problem is however that a user must be able to check if the key exchange was performed using an anonymous handshake or if the server authenticated itself. In the later case it is also desirable that the user can check the server certificate as there are no means defined like that the host name must be present. As the host is a gateway, usually located in an intranet this would not make much sense either.

DoS attacks do allow access to any confidential information. They may however prevent users from accessing a service. Quality of service is an important factor for financial services like brokerage. While DoS attacks usually prevent new connections to be set up, they may also cause existing secure connections to be dropped.

Denial of service attacks

keyon

- Spoofing of UDP packets
 - Only resolvable at the dial-in server
 - High risk when a WAP gateway is accessible from the internet
- New initial ClientHello on an existing connection
 - Close alerts of the client may not reach the server, connection state is unknown
 - Responding with a ServerHello consumes resources
- Clear text alerts and known checksums
 - The checksum of e.g. the change cipher spec message is known



DoS attacks are very easy to carry out. In case of WTLS, the problem of IP spoofing is especially serious as UDP is used for transport. IP spoofing can only be prevented at the dial-in server. If a WAP gateway is publicly available on the internet, IP spoofing cannot be prevented at all.

Using IP spoofing, an attacker can always send a ClientHello message disguised as coming from e.g. a connection from the dial-in server. The problem in real life environments is that the closure alert of the client may never reach the WAP gateway as the connection is dropped before. The WAP gateway now has no means to decide if it is a new connection or an attack when a client hello is received. As we have a datagram transport layer, we cannot determine the connection state from it. Closure alerts or information from the upper layers are the only way to determine if a connection is closed. Also some phone, e.g. the Nokia 7110, always use the same client side port.

Alerts too may be easily spoofed. Certain alerts are sent in clear text even over secured connections. Although a checksum is used in the alert to make such attacks harder, there is still at least one message with a known checksum: The ChangeCipherSpec message. It has a known sequence number and content, thus the checksum is known. By flooding the WAP gateway (or the client) with fake alerts, it is almost impossible to establish a secure connection.

DoS attacks may be used as a resource or license hog without the need of spoofing the address. Today most WAP gateways only allow a certain number of concurrent connections, depending on the purchased license and the server resources. A ClientHello DoS attack may be used to allocate all concurrent connections at once. As we have more than 64000 ports for this attack available at the client side and we use the connection-less UDP protocol, such an attack is easily performed. Note that in Switzerland, pre-paid mobile phone cards could be used to conduct such an attack completely anonymous. The last kind of attack is prevented by allowing only a limited amount of connections from one IP address. However, this kind of attacks will always remain an issue.

Client devices

keyon

Advantages

- Personal device
- Secure end device, no trojans possible at present
- Smart card reader built in

Disadvantages

- WTLS optional
- Manufactured all over the world; export regulations
- Software not easily upgradeable
- No additional software can be installed

Mobile phone users are usually not computer experts!



Mobile phones are usually very personal devices. People carry them always along, other people's access is controlled. As they usually don't run an open operating system where people can install additional software, we have sort of a secure end device as the threat of viruses or trojans does not exist. This may change in future and is of course not true if the client is a PDA (Personal Digital Assistant). Another interesting point is that each GSM phone has a smart card reader built in. Once the Subscriber identification Module (SIM) supports cryptographic operations other than those needed in GSM, very secure client authentication means can be built.

There however some drawbacks. WTLS is an optional layer and the focus of mobile phone manufacturers is definitely not the security area. Currently there are three major developers of WAP stacks for mobile devices: Nokia, Ericsson and Phone.com. Phone.com delivers the WAP software for Motorola, Siemens and other manufacturers. The mobile devices are manufactured all over the world, export regulations regarding cryptographic functionality may apply. Also there is no easy way to upgrade the software in the mobile devices. One usually has to send the device to a service center to update the software in case of bug fixes. The advantage that not software can be installed, is a disadvantage in that one has to trust the device manufacturer regarding the security. It is not possible to install a third party security layer or proxy.

Mobile phone users are not necessarily computer experts. Additional displays of security information or special security settings may easily confuse users. Security must be implemented in a fool proof way.

Current client implementations

keyon

- Mostly WTLS Class 1 clients available
 - Class 1 only supports anonymous handshakes
- The Nokia 7110 issue I
 - Non-anonymous and anonymous handshakes are indistinguishable for the user
 - Server certificates are never displayed
 - Security state information not available
 - Configurable via SMS
- The Nokia 7110 issue II
 - Does not follow the WTLS standard for the ClientKeyExchange message



Today, most WTLS implementation support WTLS Class 1 features only, i.e. they support only anonymous handshakes. Currently, the only available WTLS Class 2 device is the Nokia 7110 which supports server certificates.

While developing a WTLS server layer and the keyon / WTLS Gateway, two issues regarding the Nokia 7110 were found:

With the Nokia 7110, a non-anonymous handshake with an appropriate root certificate installed is completely indistinguishable from an anonymous handshake. This makes the device susceptible to a man-in-the-middle attack. Also the server certificate is never displayed and the subject of the certificate cannot be retrieved. As long as the server certificates are issued using the same root certificate, one cannot proof that a connection to the correct server was made. Like the server certificates, the security algorithms and key length used cannot be retrieved. The user has no idea if he connects using export grade security or strong security. Needless to say that the algorithms to use cannot be configured.

The man-in-the-middle attack is easily conducted by forcing the user to change the dialed number. As the Nokia 7110 can be configured over the air using a special configuration message, such an attack is almost trivial.

Nokia has confirmed the problem and intends to provide an upgraded firmware later this year.

The second issue is that the Nokia 7110 does not follow the WTLS standard in the ClientKeyExchange message. The encrypted pre-master-secret is represented using a wrong data type. This issue can be resolved at the server side based on length information for the key exchange key and the message.

Summary

keyon

- WTLS provides an equally high security level as TLS if appropriate algorithms are used
- No end-to-end security unless the secure WAP Gateway is operated by the content provider
- Enhancement made to TLS lead to an incompatible protocol with some issues
- Attacks are possible depending on the implementation and environment
- DoS attacks against WTLS are very easy
- Migration problems must be considered



WTLS provides an equal high level of security as TLS does, however the provider gateway model favored in WAP does not provide end-to-end security. This approach especially questions the use of client authentication as it does not provide additional security for the content provider. If end-to-end security is requested, the content provider must operate his own WAP gateway in a controlled environment. This may also mean that he has to install a whole dial-in infrastructure or team up with e.g. a internet service provider which could guarantee that spoofing is not possible using his dial-ins.

The TLS protocol used as a base was modified in so many ways, that WTLS is totally incompatible with TLS. Some of the limitation of TLS were solved, e.g. algorithms may be selected independently in WTLS, but this also lead to new problems, e.g. nonsense algorithm combinations.

We also saw that even if the protocol is secure we are faced with possible denial of service attacks unless IP spoofing is not possible. Attacks cannot be prevented by the protocol alone, the environment must also be appropriate, in the WTLS case, the dial-in server must reject spoofed packets. This problem should actually warn people to put a WAP gateway directly accessible on the internet. The question is what will happen when packet oriented mobile data communication is available with GPRS, where mobile devices are connected to the internet all the time.

The firmware of mobile devices is not easily upgradeable and WTLS is optional. We may come up with the same security level like on the internet, i.e. although client authentication is available, almost no one uses it. Especially open points like how client certificates are issued may delay WTLS Class 3 support for a long time. Unlike computers however, mobile phones usually have a shorter lifecycle.

Questions

keyon



This presentation is available at

www.keyon.ch/events

WTLS protocol details are available at

www.wapforum.org

