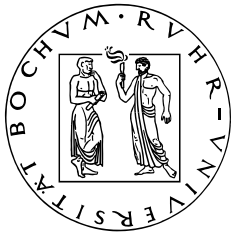
Security in WAP 1.x and WAP 2.0

Zhang chen

Sommersemester 2007

Seminararbeit
Ruhr-Universität Bochum



Chair for Communication Security
Prof. Dr.-Ing. Christof Paar

Contents

1 Introduction

In the modern society, the information and the access to information become more and more important, not only manifests in the area of technology, moreover manifests in people's daily life. In the recent several years, because of the existence of the WAP, Internet already does not limit to the desktop computers, many people can use their PDA (personal digital assistants) or mobile phones to browse web, glance over the news, get the important informations on the Internet.

What is WAP? The Wireless Application Protocol (WAP) is a protocol stack for wireless communication networks, which is specified by the WAP forum, WAP is equivalent to the Internet protocol stack (TCP/IP). Its application is to enable access to the Internet from a mobile phone or PDA. So that, if we have a mobile phone, which supports the Wireless Application Protocol, we can use the mobile phone easily portably at any time and anywhere to get to the internet.

The WAP uses The Wireless Transport Layer Security (WTLS), a wireless variant of the SSL/TLS protocol, to secure the communication between the mobile phone and other parts of the WAP architecture. In this paper, we will discuss the major security problems with the WAP, specifically the security problems with the Wireless Transport Layer Security.

2 WAP 1.x and WAP 2.0

At present, there are different versions of WAP in the modern society. WAP 2.0 edition is most popular. Now we will go through the security aspects and analyse the difference between WAP 1.x and WAP 2.0.

2.1 WAP 1.x

The Wireless Application Protocol (WAP) 1.x [4] architecture model is similar to the WWW model. It provides a good environment for application development for mobile communication devices. It consists of the origin server, gateway, and user-terminal environment, the server could be a WAP or HTTP server and the gateway is used to translate the protocol layer and application information. WAP 1.x has its own Mark-up Language, WML (Wireless Mark-up Language), which is the WAP equivalent of HTML. It is an XML-compliant format. Mobile internet sites, or WAP sites, are websites written in, or dynamically converted to, WML and accessed via the WAP browser.

Typical WAP 1.x defines three important entities: the WAP browser (the mobile device), which uses WML to display writings and pictures. the WAP gateway (also called WAP proxy) and a server on the Internet. When the mobile device wants to connect to the Internet, all the data of the communication will pass through the WAP gateway. The functionality of the WAP gateway is: It translates all the protocols used in WAP to the protocols used in Internet. It encodes or decodes the content to reduce the size of the data, so that the data can be sent over the wireless connection. The connection between the mobile device and the WAP gateway is secured by WTLS, a wireless variant of SSL/TLS(WTLS will be discussed in next section). It can be only used between the mobile device and the WAP gateway because of the very limited bandwidth, memory and computational power and battery power of mobile device. It can not perform heavy cryptographic computation (e.g., public key cryptography with a 2048-bit key). At the same time, SSL/TLS can be used between the gateway and the Internet, in order to secure the entire communication between the mobile device and the Internet server.

2.1.1 WTLS

Now we will introduce, what is the Wireless Transport Layer Security (WTLS) [4] . WTLS operates above the transport protocol layer. It is a security protocol based upon Transport Layer Security (TLS) protocol, is intended for use with the WAP transport protocols and has been optimized for use over narrow-band communication channels. The Wireless Transport Layer Security (WTLS) provides the following features for the wireless terminals:

- Privacy: WTLS contains facilities to ensure that data transmitted between the terminal and the server is private, it means, the data is encrypted, can not be understood by any intermediate parties that may have intercepted the data stream. Encryption algorithm for Privacy is chosen in the Server Hello message, WTLS supports block cipher algorithms like RC5, DES, 3DES, IDEA. There are no stream cipher algorithms expect NULL.

- Data integrity: WTLS contains facilities to ensure that data transmitted between the terminal and the server is unchanged and uncorrupted, Data integrity is ensured using the Message Authentication Codes (MAC).

- Authentication: WTLS contains facilities to establish the authenticity of the terminal and the server.

The Wireless Transport Layer Security (WTLS) establishes a session between a client (the mobile phone) and a server (the WAP gateway). We call this phase is the handshake phase, which is for negotiating a secure session and consists of the following items: Session Identifier, Protocol Version, Peer Certificate, Compression Method, Cipher Spec, Master Secret, Sequence Number Mode, Key Refresh and Is Resumable. These items are then used to create security parameters for use by the Record Layer when protecting application data.

The WTLS Handshake produces a lot of cryptographic parameters, When a WTLS client and a server start communicating, they should agree on a protocol version, select cryptographic algorithms, authenticate each other, and use public-key encryption to secure the session.

During the WTLS Handshake Protocol phase, The client and the server exchange hello messages, random values and the necessary cryptographic parameters, certificates and cryptographic information to agree on algorithms, a pre-master secret, which is used to generate a master secret. The client and server use certificates to authenticate themselves.

These goals are achieved by the handshake protocol, which can be summarized: The client and the server agree on session capabilities and exchange random values for master secret calculation. Then they exchange the keys, the servers public key is used to conduct pre-master secret, and pre-master secret is encrypted with servers public key. During this phase, RSA, Anonymous RSA, Diffie-Hellman,

elliptic curve Diffie-Hellman are used for the key exchange. The client and server calculate pre-master secret based on one's private key and another's public key, master secret is calculated using pre-master secret and random values that were exchanged in Client Hello and Server Hello messages. WTLS also uses certificates. Because certificates were not really designed to be used by mobile devices, WAP defines a new format of certificate that is optimized for storage on mobile devices and transmission over wireless networks. These certificates have the same functionality as ordinary X.509 certificates, but rely on the server to perform more of the processing under some circumstances.

WTLS can support to terminate a session and resume it later. Thus, the sessions can last for days. The longer the session remains valid, the higher the probability for an attacker to find the sessionkey. So WTLS allows in a session to renegotiate keys.

2.2 WAP 2.0

The latest Wireless Application Protocol standard, WAP 2.0 [4], developed by the WAP Forum was revealed in August 2001. It is intended to bring mobile services closer to Internet standards on desktop PCs. WAP 2.0 uses language common to the fixed and wireless environments and contains new functionality that allows users to send sound and moving pictures over their telephones. WAP 2.0 ist based on the XHTML mark-up language, which is developed by the World-Wide Web Consortium (W3C). Other Internet standards that have been adopted in WAP 2.0 include Cascading Style Sheets (CSS), Transport Layer Security (TLS), HTTP/1.1 and TCP. WAP 2.0 further evolves WAP Push, which can be used for services such as online auctions, where it is important for users to receive information at the point of interest, rather than being forced to actively look for the information.

In previous versions of WAP, a WAP gateway was required to handle the protocol between the client and the origin server. The gateway communicated with the client using the WAP protocols that are based largely on Internet communication protocols, and it communicated with the origin server using the standard Internet protocols. WAP 2.0 does not require a WAP gateway, since the communication between the client and the origin server can be conducted using HTTP/1.1. However, deploying a WAP gateway can optimize the communications process and may offer mobile service enhancements, such as location, privacy, and presence based services. In addition, a WAP gateway is necessary to offer Push functionality.

The WAP 2.0 architecture applies TLS 1.0 between mobile terminals and an application server using TCP/HTTP as a protocol to provide the secure connection service (end-to-end security). In order to make interoperability more manageable and to improve over-the-air efficiency.

The WAP 2.0 architecture also includes the use of the gateway between a WAP client and an origin server. It is necessary to define the method for TLS tunneling to support the end-to-end security at the transport level. The client has a direct connection at the transport layer to the gateway, and the gateway has a direct connection at the transport layer to the origin server. The gateway relays the data flow at the transport layer between two connections so that a direct TLS session between the client and the origin server is established. That's why the WAP 2.0 architecture applies TLS protocol to provide the secure entire connection.

2.2.1 TLS 1.0

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL) are cryptographic protocols that provide secure communications on the Internet for such things as web browsing, e-mail, Internet faxing, instant messaging and other data transfers. TLS 1.0 is specified in RFC 2246 [2]. The primary goal of the TLS 1.0 Protocol is to provide privacy and data integrity between two communicating applications. The protocol is composed of two layers: the TLS Record Protocol and the TLS Handshake Protocol. At the lowest level, layered on top of some reliable transport protocol (e.g., TCP), is the TLS Record Protocol. It provides connection security that has two basic properties:

- The connection is private. Symmetric cryptography is used for data encryption (e.g., DES, RC4, etc.). The keys for the symmetric encryption are generated uniquely for each connection.
- The connection is reliable. Keyed-MAC is used for the data integrity, secure hash functions (e.g., SHA, MD5, etc.) are used for MAC computations.

The TLS Handshake Protocol allows the server and client to authenticate each other and to agree encryption algorithm and cryptographic keys before the application protocol transmits or receives data. It provides connection security that has three basic properties:

- The peer's identity can be authenticated using asymmetric or public key cryptography (e.g., RSA, DSS, etc.).
- The negotiation of a shared secret is secure: the negotiated secret is unavailable to eavesdroppers, and for any authenticated connection the secret cannot be obtained, even by man-in-the-middle attack.
- The negotiation is reliable: no attacker can modify the negotiation communication.

WAP profile defines cipher suites, certificate formats, signing algorithms, and the use of session resume. In TLS 1.0 Protocol the server must support the following two cipher suites:

- **TLS_RSA_WITH_RC4_128_SHA**
- **TLS_RSA_WITH_3DES_EDE_CBC_SHA**

and the client must support at least one of the following cipher suites:

- **TLS_RSA_WITH_RC4_128_SHA**
- **TLS_RSA_WITH_3DES_EDE_CBC_SHA**

The client and server may support any other cipher suites and must support the session resume as defined in TLS.

TLS 1.0 allows server authentication and client authentication. The client and server must support server authentication. The client must support processing of X.509 server certificates from the server. The server should use the WAP profiled X.509 server certificate, and may use the X.509 server certificate.

The server is recommended to support client authentication. If client authentication is supported, the server must support the client certificates in the form of the WAP profiled X.509 client certificate and the X.509 client certificate. The server must also include the RSA certificate type (e.g., *rsa_sign*) in the certificate request for client certificates, and support verification of the RSA client certificate and signature. The client may support client authentication. If the client authentication is supported, the client must support use of the WAP profiled X.509 client certificate and should support use of the X.509 certificate. The client must support RSA client certificate and signature.

The Certification Authority (CA) should issue the WAP profiled X.509 client certificates.

One advantage of TLS is that it is application protocol independent. Higher level protocols can layer on top of the TLS Protocol transparently.

2.2.2 TLS 1.1

TLS 1.1 is the current approved version of the TLS protocol, which is specified in RFC 4346 [3]. Then we will see differences from TLS 1.0 in the specification: TLS 1.1 contains some small security improvements, clarifications, and editorial improvements. The major changes are:

- The implicit Initialization Vector (IV) is replaced with an explicit IV to protect against CBC attacks.
- Handling of padding errors is changed to use the `bad_record_mac` alert rather than the `decryption_failed` alert to protect against CBC attacks.

3 Security

Now the security problems with WAP will be discussed.

3.1 WAP Gateway security problem

WAP 1.x architecture doesn't offer end-to-end security. WTLS is only used between the mobile device and the gateway, while SSL/TLS can be used between the gateway and the web server on the Internet.As we know, the WAP Gateway is a device for converting the TCP/IP protocol to the WAP protocol. It is able to translate HTML to WML. The problem with WAP is apparently the WAP Gateway. Since the access to data on the Internet using WAP protocol suite isn't directly compatible to TCP/IP. It should be converted. That is where the data security is low. During the protocol translation, data will be decrypted while process in the gateway. This means that data is plaintext within the process and some of the data maybe store in the caches, if the gateway is being attacked,like man-in-the-middle attack, all the confidential information compromised.

The WAP gateway contains unencrypted data at least for some period of time. The gateway vendors have to therefore take steps to ensure that the decryption and re-encryption takes place in memory, that keys and unencrypted data are never saved to disk, and that all memory used as part of the encryption and decryption process is cleared before being handed back to the operating system.

The WAP architecture implicitly assumes that the user of the mobile phone (and the web server) trust the WAP gateway. All the unencrypted data will by the WAP gateway. This means, in sensitive services, such as electronic banking application, the bank should not rely on the client's default (and untrusted) WAP gateway. This problem was described in [5]

A solution for this problem is: The solution is to switch to a trusted and secure gateway instead of using the default WAP gateway. This is important in sensitive services like electronic banking applications. The problem with this solution is that it is not always very easy for a (non-technical) user to switch to another gateway. Note that if WAP is deployed over GSM, switching from one gateway to another can be done by sending a SMS message. Another possibility would be to change the gateway automatically on request of the target web server.

3.2 Security problems with WTLS

Although the WTLS looks reasonably good, many of the changes that were made by WAP Forum have led to security problems. Now let's analyse the security weaknesses in the WTLS protocol. The following attacks were described in [1].

3.2.1 Predictable IVs lead to chosen-plaintext attacks against low-entropy secrets

While the TLS protocol was designed to be used over a reliable transport (such as TCP/IP), the WTLS protocol should be able to operate over an unreliable datagram transport where datagram may be lost, duplicated, or reordered. If CBC-encryption mode is being used, this requirement makes it necessary for the IV (Initial Value) to be contained in the packet itself or that the IV for that block can be derived from data already available to the recipient. WTLS always uses a linear IV computation, even for reliable transports.

When a block cipher is used in CBC-encryption mode, the IV for encrypting each packet is computed as follows:

$$IV_s = IV_0 \oplus (s|s|s|s) \tag{3.1}$$

where s is a 16-bit sequence number of the packet and IV_0 is the original IV, derived during key generation.

The plaintext blocks $P_{s,0}\,, P_{s,1}\,, ...$in the packet s are encrypted as:

$$C_{s,0} = E_k(IV_s \oplus P_{s,0}) \tag{3.2}$$

$$C_{s,i} = E_k(C_{s,i-1} \oplus P_{s,i}),\ \text{for } i > 0 \tag{3.3}$$

When CBC-encryption is used mode in combination with a terminal application (such as telnet), where each keypress is sent as an individual packet, this will come problems when low-entropy secrets (passwords) are entered. Alice enters the password into the application, an attacker gets this packets. So the attacker has blocks of type to guess every character of the password:

$$C_{s,0} = E_k(P_{s,0} \oplus IV_s) = E_k(P_{s,0} \oplus IV_0 \oplus (s|s|s|s)) \tag{3.4}$$

where $P_{s,0}$ contains an unknown letter of Alice's password. The attacker therefore knows s.

The attacker somehow gets hold of Alice's channel and guesses that the unknown letter in the password is L, then sends the following packet through Alice's channel to check if his guess was correct:

$$P_{r,0} = L \oplus (s|s|s|s) \oplus (r|r|r|r) = L \tag{3.5}$$

where r is the sequence number of this packet.

The attacker can make a right guess $L = P_{s,0}$, which leads to matching ciphertexts $C_{r,0} = C_{s,0}$. We can say this is an oracle that tells whether the guessed password letter was correct. The entire password can be easily brute forced with this oracle.

3.2.2 The weak XOR MAC and stream ciphers

The WTLS protocol supports a 40-bit XOR MAC, which works by padding the message with zeros, dividing it into 5-byte blocks and xoring these blocks together.

The specification states that the XOR MAC is only used for "some devices with very limited CPU resources", and the XOR MAC "may not provide as strong message integrity protection as SHA", when exportable encryption is being used. The XOR MAC does not provide any message integrity protection if stream ciphers are being used, regardless of the key length. Because MAC can be made to match by inverting the bit (n mod 40) in the MAC, if one inverts a bit position n in the ciphertext. This can be repeated arbitrary number of times. In this case, if stream ciphers are used, the weak XOR MAC does not provide any integrity protection.

3.2.3 35-bit DES encryption

The encryption used to encrypt data during a WTLS session is negotiated in the handshake phase. There is the possibility to choose the 40-bit DES encryption method.

The 40-bit DES encryption uses five bytes of keying material. Because of the parity bit in each byte of a DES key, In other words, a 5 byte key is used which contains 5 parity bits. There are only $5 * 7 = 35$ effective key bits in five bytes in DES key. so it's easy to perform a brute force attack on the DES key. We can say that the 40-bit DES encryption is a weak algorithm.

3.2.4 Unauthenticated alert messages

One of the content types supported by the WTLS Record layer is the alert type. Alert messages convey the severity of the message and a description of the alert.

An alert message is either sent as specified by the current connection state (e.g., compressed and encrypted), or under null cipher spec (e.g., without compression or encryption).

Some of the alert messages used in the protocol are sent in cleartext and are not properly authenticated. Most of these messages are warnings and do not cause the session to be terminated.

Since an alert message can take up a sequence number in the protocol, an active attacker may replace an encrypted datagram with an unauthenticated plaintext alert message with the same sequence number without being detected. This leads to a truncation attack that allows arbitrary packets to be removed from the data stream.

The alert messages can be sent in cleartext. When sending the alert, messages are sent in cleartext no compression, MAC protection or encryption is used. As mentioned under Security features this means that the cipher suite is assigned to NULL.

Solution: All messages affecting the protocol state should be properly authenticated.

3.2.5 The RSA PKCS #1 attack

The RSA signatures and encryption are performed according to PKCS #1, version 1.5. Daniel Bleichenbacher and others have demonstrated that if the protocol includes an oracle that tells whether a given packet has a correct PKCS #1 version 1.5 padding, RSA messages can be decrypted with approximately 2^{20} chosen ciphertext queries. In some implementations, the WTLS error messages *bad_certificate* and *decode_error* may provide such an oracle to the attacker.

Solution: The 2.0 version of the PCKS #1 should be used instead of PKCS #1,version 1.5.

3.2.6 Plaintext leaks

An eavesdropper can determine the initial IV of each packet under exportable from the Hello messages and the sequence number alone.The exportable keys have a minimal key-length, which is only forty bits is considered weak. Thus, an attacker can easily get the initial IV just by looking at the Hello messages. Also; Hello Request messages are omitted from handshake hashes.

An eavesdropper can determin the change of keys,because the *record_type* field is sent unencrypted. This field determines the type of the message; one type being the Change Cipher Spec type.

The existence of encrypted error messages can be determined from the *record_type* field. The exact nature of the encrypted error messages can not be determined.

3.2.7 Probable plaintext attacks

If we have enough know or probable plaintext, we can perform an exhaustive key search (brute force) on a symmetric encryption, so that the correct key can be

recognized with trial decryption of one or more blocks.

We should prevent to use of weak algorithms and use strong authentication (RSA, big enough key size), good encryption algorithms (RC5) and full MAC algorithm to provide high enough security for commercial purposes.

Bibliography

[1] Markku-Juhani Saarinen, Attacks Against The WAP WTLS Protocol
http://web.freeprotocols.org/harmOfWap/wtls.pdf

[2] T.Dierks and C.Allen, The Transport Layer Security (TLS) Protocol Version 1.0
www.ietf.org/rfc/rfc2246.txt

[3] T. Dierks and E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.1
http://www.ietf.org/rfc/rfc4346.txt

[4] Open Mobile Alliance
http://www.openmobilealliance.org

[5] Dave Singelee and Bart Preneel, The Wireless Application Protocol
www.cosic.esat.kuleuven.be/publications/article-600.pdf