# WAP™ TLS Profile and Tunneling WAP-219-TLS

11-April-2001

## Wireless Application Protocol
## TLS Profile and Tunneling Specification

A list of errata and updates to this document is available from the WAP Forum™ Web site, http://www.wapforum.org/, in the form of SIN documents, which are subject to revision or removal without notice.

# Contents

# 1. Scope

The Wireless Application Protocol (WAP™) is a result of continuous work to define an industry wide specification for developing applications that operate over wireless communication networks. The scope for the WAP Forum is to define a set of specifications to be used by service applications. The wireless market is growing very quickly, reaching new customers and providing new services. To enable operators and manufacturers to meet the challenges in advanced services, differentiation, and fast/flexible service creation, WAP selects and defines a set of open, extensible protocols and content formats as a basis for interoperable implementations.

The objectives of the WAP Forum are:

- To bring Internet content and advanced data services to digital cellular phones and other wireless terminals.
- To create a global wireless protocol specification that will work across differing wireless network technologies.
- To enable the creation of content and applications that scale across a very wide range of bearer networks and device types.
- To embrace and extend existing standards and technology wherever appropriate.

This specification defines the WAP profile for using TLS 1.0 as the transport layer security protocol, and the TLS tunneling for enabling the transport level end to end security in the WAP-NG architecture [WAPArch].

# 2. Document Status

This document is available online in the following formats:

PDF format at http://www.wapforum.org/.

## 2.1 Copyright Notice



## 2.2 Errata

Known problems associated with this document are published at http://www.wapforum.org/.

## 2.3 Comments

Comments regarding this document can be submitted to the WAP Forum in the manner published at http://www.wapforum.org/.

# 3. References

## 3.1 Normative references

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels, " RFC 2119, S. Bradner, March 1997. URL: http://www.ietf.org/rfc/rfc2119.txt

[RFC2246] "The TLS Protocol, Version 1.0," rfc 2246, T. Dierks, C. Allen, January 1999. URL: http://www.ietf.org/rfc/rfc2246.txt

[RFC2459] "Internet X.509 Public Key Infrastructure Certificate and CRL Profile," rfc 2459, R. Housley, W. Ford, W. Polk, D. Solo, January 1999. URL: http://www.ietf.org/rfc/rfc2459.txt

[RFC2817] "Upgrading to TLS Within HTTP/1.1," rfc 2817, R. Khare, S. Lawrence, May 2000. URL: http://www.ietf.org/rfc/rfc2817.txt

[WAPCert] "WAP Certificate and CRL Profiles," WAP-211-WAPCert , WAP Forum. URL: http://www.wapforum.org.

[WCREQ] “Specification of WAP conformance requirements,” WAP-221-CREQ, WAP Forum. URL: http://www.wapforum.org.

## 3.2 Informative references

[WAPArch] “Wireless Application Protocol Architecture Specification”, WAP-210-WAPArch, Draft Version 17-October-2000, WAP Forum. URL: http://www.wapforum.org/

[RFC2818] "HTTP over TLS," rfc 2818, E. Rescorla, May 2000. URL: http://www.ietf.org/rfc/rfc2818.txt

# 4. Definitions and Abbreviations

## 4.1 Definitions

The following are terms and conventions used throughout this specification.

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described by [RFC2119].

**Client** – a device (or application) that initiates a request for a connection with a server.

**Client Authentication** - refers to the authentication of the client identity by using the client certificate in TLS 1.0 for the purpose of this specification.

**Origin Server** – the server on which a given resource resides or is to be created. Often referred to as a web server or an HTTP server.

**Proxy Server** - the server on which a given resource neither resides nor is to be created. To complete the client request, the proxy server must get the resource from the related origin server(s).

**Server** – a device (or application) that passively waits for connection requests from one or more clients. A server may accept or reject a connection request from a client.

**Server Authentication** - refers to the authentication of the server identity by using the server certificate in TLS 1.0 for the purpose of this specification.

## 4.2 Abbreviations

For the purposes of this specification, the following abbreviations apply.

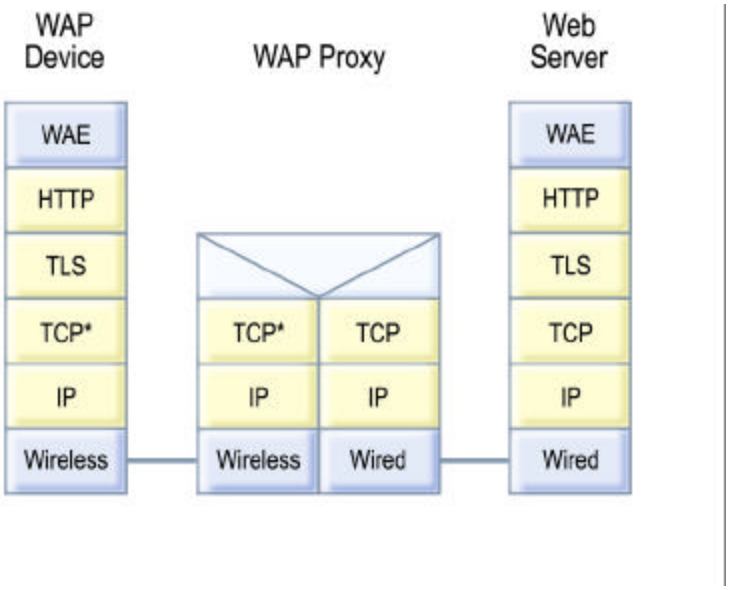
| | |
|---|---|
| **CA** | Certificate Authority |
| **HTTP** | Hypertext Transfer Protocol |
| **IP** | Internet Protocol |
| **RFC** | Request For Comments |
| **TCP** | Transport Control Protocol |
| **TLS** | Transport Layer Security |
| **URL** | Uniform Resource Locator |
| **WAE** | Wireless Application Environment |
| **WAP** | Wireless Application Protocol |

# 5. Introduction

This section is informative.

The WAP Architecture [WAPArch] cites TLS as a protocol to provide the Secure Connection service. In order to make interoperability more manageable and to improve over-the-air efficiency, this document specifies a WAP profile for the use of TLS 1.0. The WAP profile includes cipher suites, certificate formats, signing algorithms, and the use of session resume, as described below.

In addition to the direct access, the WAP-NG architecture also includes the use of the proxies between a WAP client and an origin server [WAPArch]. It is necessary to define the method for TLS tunneling to support the end to end security at the transport level. The following diagram illustrates the TLS tunneling in the WAP-NG architecture [WAPArch],




**Stack View of TLS Tunneling**

The client has a direct connection at the transport layer to the proxy, and the proxy has a direct connection at the transport layer to the origin server. The proxy 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. The wireless profile of TCP (TCP*) [WAPArch] is used over the air in the above diagram.

## 5.1 Note on use of various RFC's as normative references

Implementers should note that RFC 2459 (PKIX profile of X.509 certificates) is scheduled to be superseded sometime during 2001. RFC 2246 (TLS 1.0 specification) may be superseded in the same time frame. However, there are a large number of implementations following these RFC's and the superseding RFC's will preserve backward compatibility with their predecessors. RFC's 2459 and 2246 can therefore be used for implementation of TLS according to the profile of TLS in this document

# 6. TLS Profile

TLS 1.0 [RFC2246] is used as the baseline from which this specification profiles. The client and server implementations MUST conform to TLS 1.0.

## 6.1 Cipher Suites

The server MUST support all of the following cipher suites

```
TLS_RSA_WITH_RC4_128_SHA
TLS_RSA_WITH_3DES_EDE_CBC_SHA
```

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.

## 6.2 Session

### 6.2.1 Session Resume

The client and server MUST support the session resume as defined in TLS. The longer session life (eg, 12 hours) is RECOMMENDED. The guidelines on the session resume as documented in TLS 1.0 should be respected.

### 6.2.2 Session Identifier

This section is informative.

To improve over the air efficiency, it is desirable that the server uses session identifiers of length 8 bytes or less.

## 6.3 Server Authentication

The client and server MUST support server authentication.

The client MUST support processing of X.509 server certificates as detailed in "WAP Certificate and CRL Profile" [WAPCert]. The client implementation should respect the guidelines for server identity as documented in RFC 2818 [RFC2818]. Furthermore, the client should use the guidelines for handling X.509 server certificates as described in "WAP Certificate and CRL Profiles Specification" [WAPCert]. The client should follow the guidelines in [WAPCert] for handling of unknown attributes and extensions if the server certificate does not conform to [WAPCert].

The server SHOULD use the WAP profiled X.509 server certificate [WAPCert], and MAY use the X.509 server certificate [RFC2459].

## 6.4 Client Authentication

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 [WAPCert] and the X.509 client certificate [RFC2459]. The server MUST also include the RSA certificate type (ie, `rsa_sign`) in the certificate request [RFC2246] 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 [RFC2459]. The client MUST support RSA client certificate and signature.

CA should issue the WAP profiled X.509 client certificates [WAPCert].

The server implementation should respect the guidelines for client identity as documented in RFC 2818 [RFC2818].

# 6.5 Certificate Chain Depth

## 6.5.1 CA Practice Recommendation

This section is informative.

To improve over-the-air efficiency for the TLS full handshake, the CA should use the minimum possible chain depth for the client and server certificates.

# 7. TLS Tunneling

As exemplified in the WAP-NG architecture [WAPArch], a HTTP proxy may be used between a client and an origin server. In order to maintain the end to end security at the transport layer while using a proxy, TLS tunneling must be used between the client and the origin server. The proxy functions as a transport level data relay element and is isolated from the TLS session between the client and the origin server. The client MUST support the TLS tunneling if it supports the HTTP proxy.

To establish a TLS tunnel, the client MUST use HTTP CONNECT method as defined in RFC 2817 [RFC2817]. Furthermore, the client MUST only establish the tunnel over a raw TCP connection, not an "upgraded" connection per RFC 2817 [RFC2817].

The HTTP proxy server should support the HTTP CONNECT method in the manner as defined in RFC 2817 [RFC2817]. It should be noted that a chain of HTTP proxy servers, including proxy servers that do not support HTTP CONNECT method, may be involved for a desired TLS tunnel, the client should not assume that a TLS tunnel can always be successfully established. The client MUST abort the attempt to establish a TLS tunnel if a non-successful response for an HTTP CONNECT request is received.

# Appendix A Static Conformance Requirements

This static conformance clause defines a minimum set of features that should be implemented to ensure interoperability. A feature can be optional (O), or mandatory (M) [WCREQ].

## A.1 General Requirements

This section applies to all the clients and servers that conform to this specification.

| Item | Functionality | Reference | Status | Requirement |
|---|---|---|---|---|
| TLS-001 | Conform to TLS 1.0 | 6 | M | [RFC2246] |

## A.2 Client Options

### A.2.1 Basic

This section applies to all clients that conform to this specification.

| Item | Functionality | Reference | Status | Requirement |
|---|---|---|---|---|
| TLS-C-010 | RSA based Cipher Suites (TLS-C-011 and TLS-C-012); at least one supported. | 6.1 | M | TLS-C-011<br>OR<br>TLS-C-012 |
| TLS-C-011 | TLS_RSA_WITH_RC4_128_SHA | 6.1 | O | |
| TLS-C-012 | TLS_RSA_WITH_3DES_EDE_CBC_SHA | 6.1 | O | |
| TLS-C-020 | Session Resume | 6.2.1 | M | |
| TLS-C-030 | Server Authentication | 6.3 | M | TLS-C-031 |
| TLS-C-031 | Support X.509 certificate processing in accordance with the WAP Certificate and CRL Profile Specification [WAPCert] | 6.3 | M | Cert-SrvA-C-01<br>AND<br>Cert-SrvA-C-02<br>AND<br>Cert-SrvA-C-03<br>AND<br>Cert-SrvA-C-04<br>AND<br>Cert-SrvA-C-05<br>AND<br>Cert-SrvA-C-06<br>AND<br>Cert-SrvA-C-07 |

| Item | Functionality | Reference | Status | Requirement |
|---|---|---|---|---|
| | | | | AND<br>Cert-SrvA-C-09<br>AND<br>Cert-SrvA-C-10<br>AND<br>Cert-SrvA-C-12<br>AND<br>Cert-SrvA-C-13<br>AND<br>Cert-TLS-C-01<br>[WAPCert] |
| TLS-C-040 | Client Authentication | 6.4 | O | TLS-C-100<br>AND<br>TLS-C-102 |

## A.2.2 Client Authentication

This section only applies to the client that supports client authentication.

| Item | Functionality | Reference | Status | Requirement |
|---|---|---|---|---|
| TLS-C-100 | Support use of WAP profiled X.509 client certificate | 6.4 | O | WAPCert:MCF<br>[WAPCert] |
| TLS-C-101 | Support use of X.509 client certificate | 6.4 | O | [RFC2459] |
| TLS-C-102 | Support RSA client certificate and signature | 6.4 | O | |

## A.2.3 TLS Tunneling

This section only applies to clients that support proxy.

| Item | Functionality | Reference | Status | Requirement |
|---|---|---|---|---|
| TLS-C-200 | Support TLS tunneling | 7 | M | |
| TLS-C-201 | Establish the tunnel over the raw TCP connection | 7 | M | |
| TLS-C-202 | Use HTTP CONNECT to establish a TLS tunnel | 7 | M | [RFC2817] |
| TLS-C-203 | Abort the attempt to establish a TLS tunnel if a non-successful response for an HTTP CONNECT request is received | 7 | M | |

# A.3 Server Options

## A.3.1 Basic

This section applies to all servers (ie, origin server or proxy server) that conform to this specification.

| Item | Functionality | Reference | Status | Requirement |
|---|---|---|---|---|
| TLS-S-011 | TLS_RSA_WITH_RC4_128_SHA | 6.1 | M | |
| TLS-S-013 | TLS_RSA_WITH_3DES_EDE_CBC_SHA | 6.1 | M | |
| TLS-S-020 | Session Resume | 6.2 | M | |
| TLS-S-030 | Server Authentication | 6.3 | M | TLS-S-031<br>OR<br>TLS-S-032 |
| TLS-S-031 | Use of WAP profiled X.509 server certificate | 6.3 | O | [WAPCert] |
| TLS-S-032 | Use of X.509 server certificate | 6.3 | O | [RFC2459] |
| TLS-S-040 | Client Authentication | 6.4 | O | TLS-S-100<br>AND<br>TLS-S-101<br>AND<br>TLS-S-102<br>AND<br>TLS-S-103 |

## A.3.2 Client Authentication

This section only applies to servers that support client authentication.

| Item | Functionality | Reference | Status | Requirement |
|---|---|---|---|---|
| TLS-S-100 | Support WAP profiled X.509 client certificate | 6.4 | O | WAPCert:MSF<br>[WAPCert] |
| TLS-S-101 | Support X.509 client certificate | 6.4 | O | [RFC2459] |
| TLS-S-102 | Support verification of RSA client certificate and signature | 6.4 | O | |
| TLS-S-103 | Request RSA certificate type for client certificate | 6.4 | O | |

# Appendix B History and Contact Information

| Document history | | |
|---|---|---|
| **Date** | **Status** | **Comment** |
| 10-Jan-2001 | Prototype | Frozen for formal architecture consistency review. |
| 06-Apr-2001 | Prototype | Incorporated CRs approved since last update. |
| 11-Apr-2001 | Prototype | One minor editorial change. |
| 08-May-2001 | Proposed | Editorial changes for Proposed status |
| 22-June-2001 | Approved | Editorial changes for Approved status |

**Contact Information**

http://www.wapforum.org.

technical.comments@wapforum.org