WAP Security

Helsinki University of Technology
S-38.153 Security of Communication Protocols
Mikko.Kerava@iki.fi
15.4.2003

Contents

Wireless Application Protocol

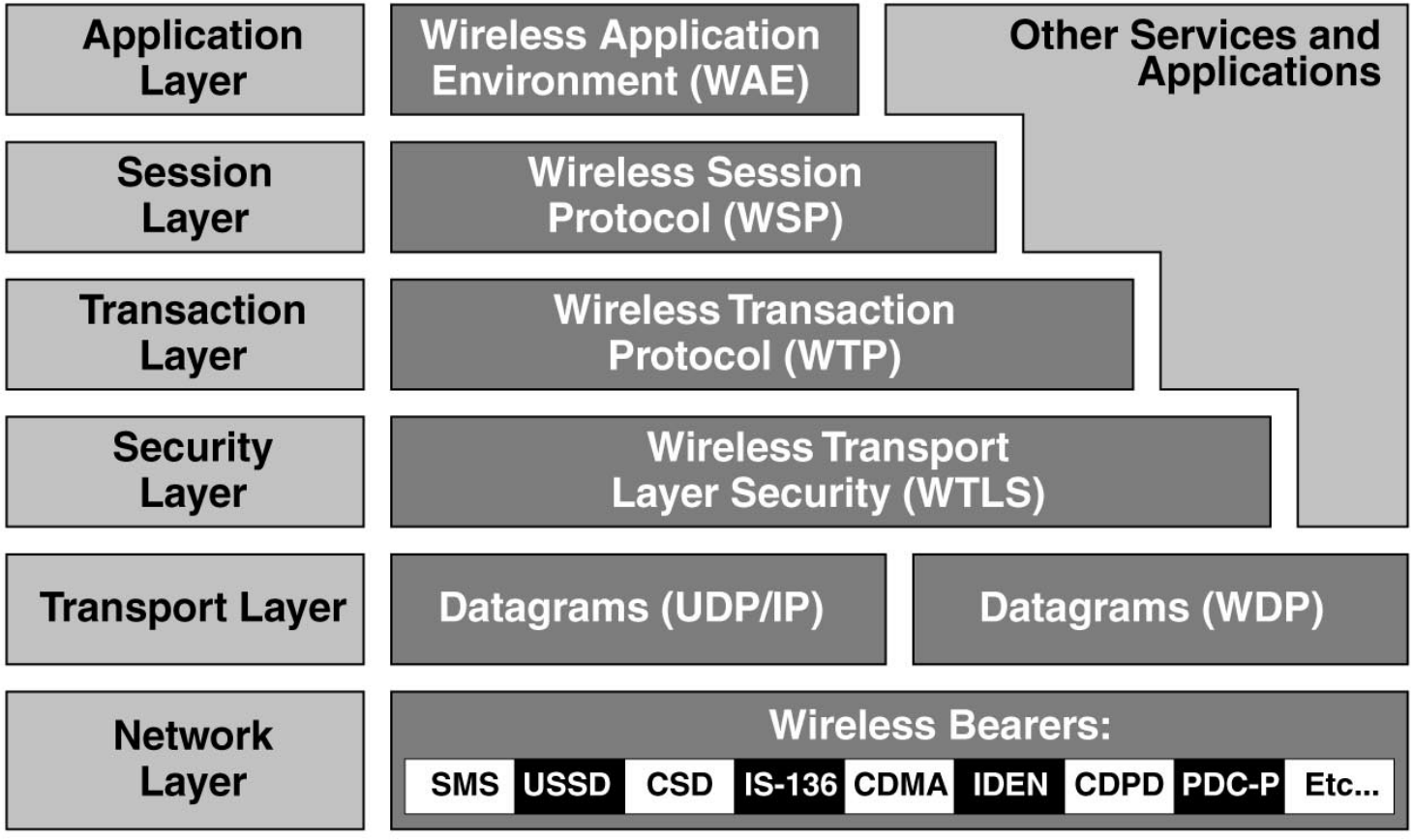
Designed to bring WWW look and feel and advanced services to mobile terminals

Open standard, developed by WAP Forum industry association

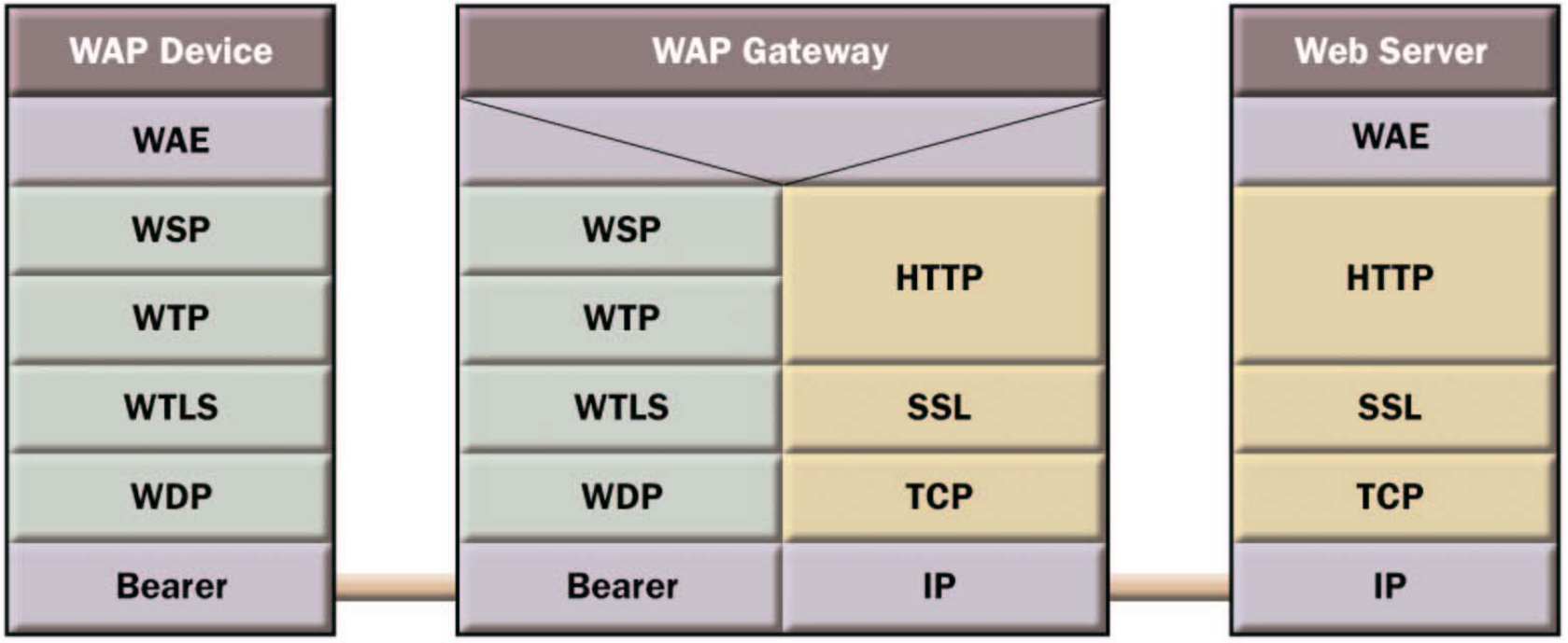
Bearer independence

Device independence

WAP Protocol Stack (version 1.x)



WAP version 1.x Communication Model



Problems

Data is decrypted and again encrypted in WAP gateway

No end to end security => man-in-the-middle-attack

No control of SSL part of the connection

Some problems in WTLS security

WAP version 2.0

Released January 2002

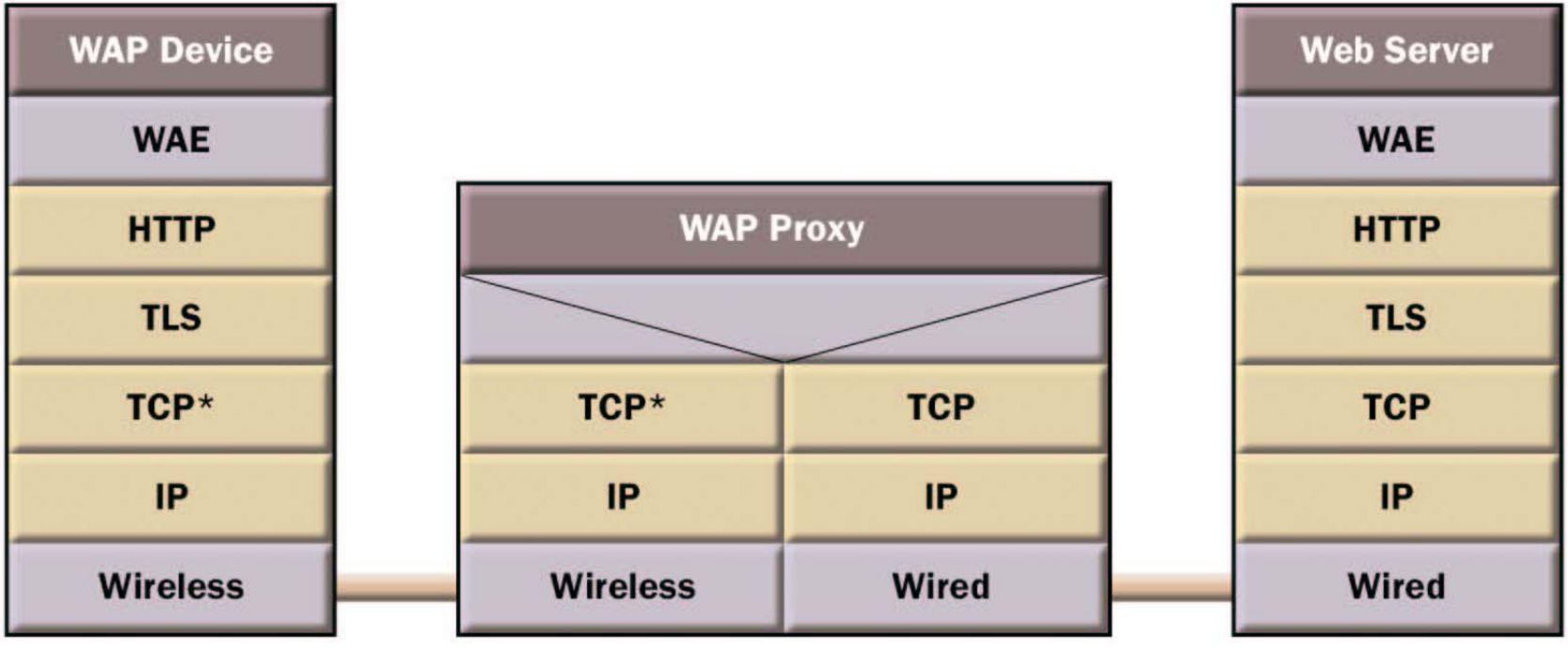
For mobile networks supporting IP

Support for TLS => End to end security

Wireless profiled TCP and HTTP

Dual Stack => WAP 1.x support

WAP version 2.x Communication Model



TCP*: Wireless Profiled TCP (WP-TCP)

WTLS

Based on TLS version 1.0

Some modifications to suite better in wireless environment

Privacy

- Symmetric cryptography

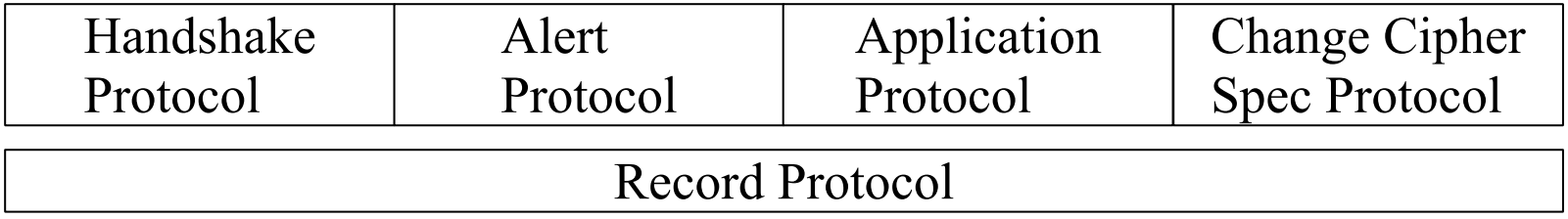
Authentication

- Certificates

Integrity

- Message Authentication Codes (MAC)

WTLS internal architecture



Handshake Protocol	Authentication, key-exchange and agreement of security parameters
Alert Protocol	Error Handling. Warning, Critical and Fatal messages
Application Protocol	Interface for upper layers
Change Cipher Spec Protocol	Announce switch to negotiated algorithms and values
Record protocol	Takes care of encryption, decryption, data integrity and authentication

Authentication in the WTLS

Three classes of authentication:

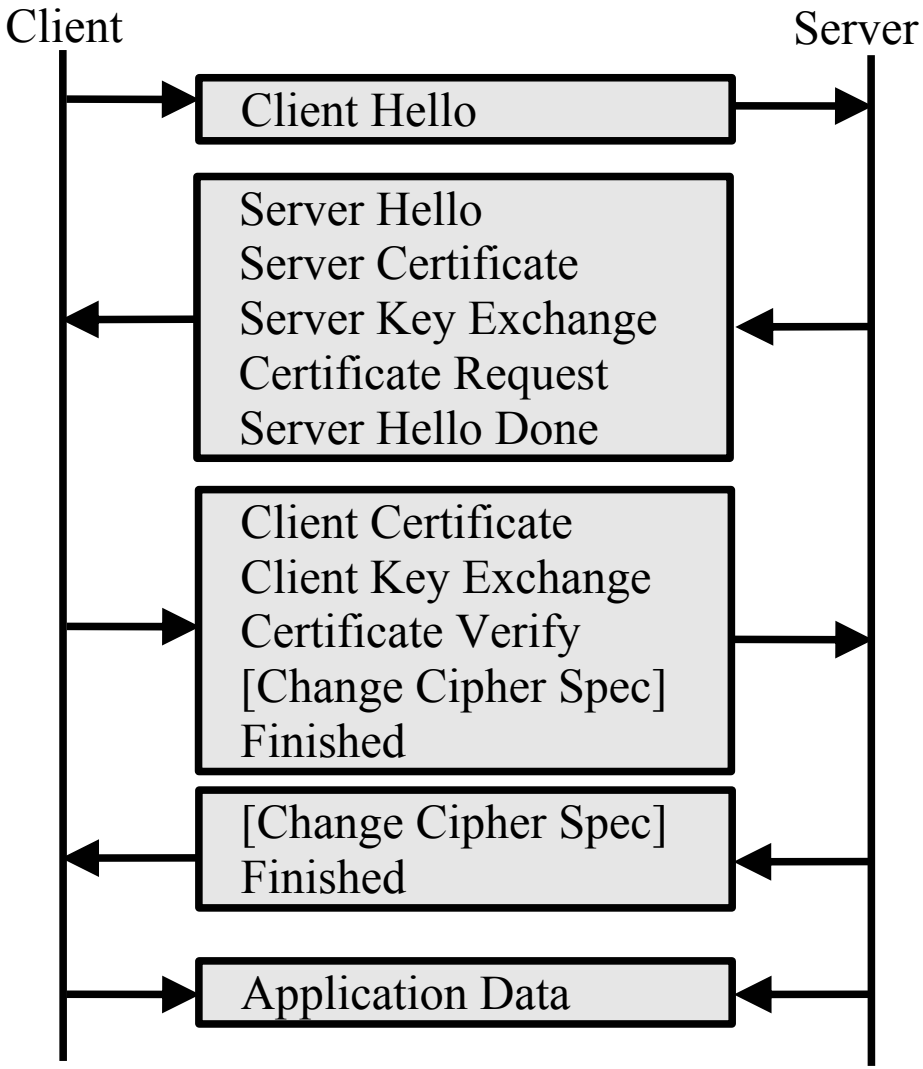
- Class 1: Anonymous
- Class 2: Server Authentication
- Class 3: Client and Server Authentication

Authentication is carried out with certificates

Supperted certificates:

- X.509v3
- WTLS (Optimised with size)
- X9.68 (Currently in draft)

WTLS Handshake Procedure



WTLS Handshake Procedure

Client Hello and Server Hello:

Parties agree on session capabilities and exchange random values for master secret calculation

Server Key Exchange:

Servers public key to conduct pre-master secret.

Client Key Exchange:

Pre-master secret encrypted with servers public key

Other Handshaking Procedures

Resumed Connection

If previously negotiated session, client sends Client Hello with sessionID. If both have the same sessionID they may continue the secure session

Abbreviated Handshake

Client and server have shared secret, which is used as a pre-master secret

Optimised Full Handshake

Server retrieve client's certificate using the trusted third party. Master key calculated using Diffie-Hellman method

Key Exhange

Algorithms:

- RSA, Anonymous RSA

  Server Key Exchange: the public key of the server.

  Client Key Exhange: pre-master secret encrypted with the public key of the server

- Diffie-Hellman, elliptic curve Diffie-Hellman

  Client and server calculate pre-master secret based on one's private key and the counetrpart's public key

Master secret is calculated using pre-master secret and random values that were exchanged in Client Hello and Server Hello messages

Privacy – Encryption

Encryption algorithm is chosen in the Server Hello message

Supported block cipher algorithms:

- RC5
  - 40, 56, 128 bit keys
- DES
  - 40, 56 bit keys
- 3DES
  - 40, 56, 128 bit keys
- IDEA
  - 40, 56, 128 bit keys

No stream cipher algorithms expect NULL

Integrity – MAC

Data integrty is ensured using the Message Authentication Codes

MAC algorithm is chosen in the Server Hello message

WTLS supports many versions of common MAC algorithms

- SHA
- MD5

MAC is generated over the compressed WTLS data

Security problems in WTLS

Developed to support wide range of mobile device, including devices with limited CPU and memory recources => NULL and weak encryption methods available

Allowing anonymous authentication opens door to man-in-the-middle attack

Attacks against WTLS by Markku-Juhani Saarinen: http://www.jyu.fi/~mjos/wtls.pdf

- Weak MAC available
- RSA PKCS#1 1.5
- Unauthenticated alert messages
- Plaintext leaks

Is WTLS security level sufficient?

Security level is a compromise between usability and the strength of the encryption method

Many WAP services may not require strong encryption

Preventing use of weak algorithms and using strong authentication (RSA, big enough key size), good encryption algorithms (RC5) and full MAC algorithm provides high enough security for commercial purposes (in my opinion)

But end user is not always able to affect on or even find out what algorithms is in use

Other WAP Security Components

WIM – WAP Identification Module

A tamper-resistant device which is used in performing WTLS and application level security functions and to store and process information needed for user identification and authentication.

WMLScript Crypto API

Application programming interface providing basic security functions, such as digital signatures to be used for authentication or non-repudiation purposes in application level

WPKI – WAP Public Key Infrastructure