# Performance of WTLS and its Impact on an M-Commerce Transaction

Ian Herwono and Ingo Liebhardt

Communication Networks
Aachen University of Technology
Kopernikusstraße 16, D-52074 Aachen, Germany
Phone: +49/241/80-7248, Fax: +49/241/8888-242
{ian|ilt}@comnets.rwth-aachen.de

**Abstract.** Transaction security is commonly seen as one of the key factors influencing the success of *Mobile Commerce*. In this paper simulation-based performance measurements of the *Wireless Transport Layer Security (WTLS)* protocol are presented. Its impact on an exemplary m-commerce transaction is discussed.

## 1 Introduction

Although saturation can be observed in Europe, the market for mobile telephony still faces an overwhelming growth in most of the world's regions. Globally, 240 million[1] people are predicted to use their mobile phones for wireless data exchange by the end of 2004—up from 26 million in 1999. As most of this data exchange is predicted to be business-centred, a considerable amount of users all over the world will be engaged in *Mobile Commerce (M-Commerce)*.

The *Wireless Application Protocol (WAP)* [1] specifies an application framework and network protocols to foster convergence of the Internet and wireless networks like CDPD or GSM/GPRS (Fig. 1). Within the context of m-commerce the *Mobile electronic Transaction (MeT) Initiative* has been formed by the leading mobile manufacturers to define common and consistent usage scenarios, e. g. , mobile payment or ticketing [2]. Rather than developing proprietary solutions to security problems, MeT embraces and extends existing industry standards and technologies—especially WAP. Therefore the performance of the employed WAP security mechanisms—WTLS and WMLScript signText—has a major impact on the overall transaction duration.

In [3] several alternatives for establishing secure channels to mobile devices have been compared whereby the influence of different key lengths and key exchange protocols has not been examined extensively. This work contributes detailed performance measurements of WTLS acquired from our WAP simulation platform.

After giving an overview on WTLS we briefly describe the simulator and present the measurement results. We then discuss the impact WAP's security

[1] according to Allied Business Intelligence

mechanisms impose on the overall duration of m-commerce transactions by exemplarily investigating a MeT payment using a SET Wallet Server.

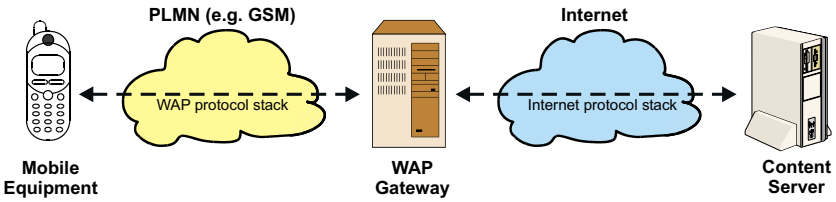


**Fig. 1.** Typical setup for accessing an Internet server via WAP

## 2 Wireless Transport Layer Security

The WTLS protocol is based upon the industry-standard Transport Layer Security (TLS) and offers various cryptographic algorithms to provide confidentiality, integrity, and authentication over the air interface. Several symmetric algorithms like DES, 3DES, RC5, or IDEA can be employed for en- and decryption whereas a keyed HMAC hash in combination with MD5 or SHA-1 is used for ensuring message authentication. RSA and ECDH are suggested for anonymous key exchange. In addition RSA-signing and ECDSA can be used for authenticated key exchange. It has to be noted that WTLS is unable to ensure nonrepudiation.

Unlike in RSA handshakes a provision is made for an optimised variant of the ECDH_ECDSA and ECDH handshakes. In this case the amount of data to be transferred across the air interface can be reduced since the server is able to retrieve the client's certificate from a certificate distribution service or from its own sources rather than obtaining it from the client. The flows of messages exchanged within full and optimised handshakes are depicted in Fig. 2. To resume

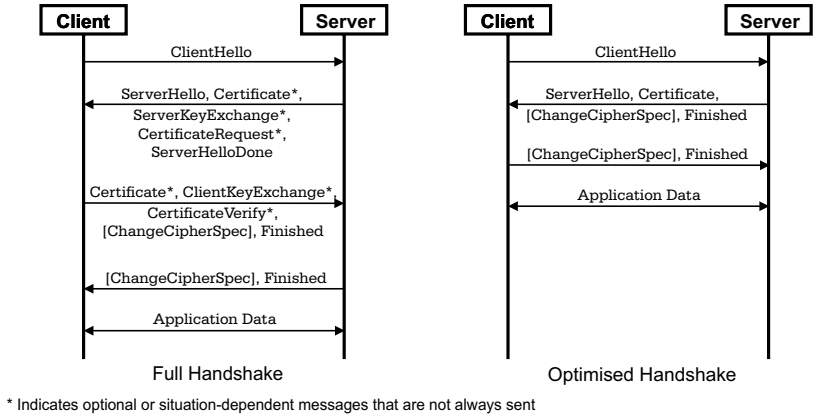


**Fig. 2.** Message flow for WTLS handshakes

a previous secure session and reuse negotiated security parameters, an abbreviated handshake can be performed. Further details concerning WTLS can be found in [1].

## 3 Performance Evaluation

Our simulation platform is a prototypical, standard-conformant implementation of the relevant protocols WTP, WTLS, WDP, and IP. It is formally specified in SDL[2] and coded using C/C++. Most of the implementations concerning cryptographic computations originate from the free C++ class library Crypto++ 4.1[3]. All following measurements are the results of tests carried out on a SUN Enterprise server equipped with 1664 Mbyte RAM and using one single *dedicated* processor of 400 MHz clock frequency.

### 3.1 Throughput of WTLS

The WTLS throughput results from the processing times needed for the generation of record IV, the calculation and verification of keyed MAC, and the en- and decryption respectively. Hence the values given in Table 1 do not correspond with the ones resulting from investigations on the pure cipher throughput as—for example—done in [4]. WTLS user data ranging from 256 up to 8192 bytes have been used and the measured throughputs have been averaged. Note that only user data has been taken into account for calculating the throughput whilst the encrypted data additionally includes the MAC and padding. All ciphers operate in CBC mode and a key length of 128 bits has been chosen for AES, Serpent, Twofish, and Mars.

**Table 1.** Throughput of the WTLS-layer in Mbit/s

| keyed MAC | DES | 3DES | RC5 | IDEA | AES | Serpent | Twofish | Mars |
|---|---|---|---|---|---|---|---|---|
| MD5 (enc) | 5.81 | 2.38 | 9.01 | 6.52 | 15.10 | 3.32 | 5.63 | 5.82 |
| MD5 (dec) | 5.76 | 2.37 | 8.70 | 5.75 | 14.76 | 3.13 | 5.52 | 5.69 |
| SHA-1 (enc) | 5.14 | 2.26 | 7.45 | 5.61 | 10.56 | 2.96 | 4.78 | 4.97 |
| SHA-1 (dec) | 5.13 | 2.26 | 7.40 | 5.15 | 11.36 | 2.92 | 4.89 | 5.08 |

Table 1 shows that AES (Rijndael) in combination with MD5 provides the highest performance. The faster the investigated cipher algorithm, the more weight lies in the selection of the hashing algorithm. However we observe that—contrary to our expectation—the encryption throughput of AES is higher than its decryption throughput when SHA-1 is employed. Even after repeated simulations on different machines and thorough analysis this behaviour remained inexplicable.

[2] Specification and Description Language

[3] Please refer to `http://www.eskimo.com/~weidai/cryptlib.html` for further information.

### 3.2 Handshake

In contrast to the WTLS throughput, which—even when implemented within a constrained environment—is higher than the underlying network's throughput, attention has to be paid to the duration of a handshake. The overall durations of several handshake procedures have been measured while varying the effective mean throughput of the underlying bearer, which is determined by numerous factors such as available radio resources, network latency and channel quality. The measurement results of four types of full handshakes and one optimised handshake are shown in Fig. 3. The key lengths of RSA and ECDH have been set to 1024 bits and 160 bits respectively. The time needed for the server retrieving a certificate in the optimised variant has been assumed to be 500 ms.

Interestingly, the impact of the various cryptographic methods becomes negligible as the network throughput decreases, and the amount of data[4] transferred during a single handshake (Table 2) gets more important.

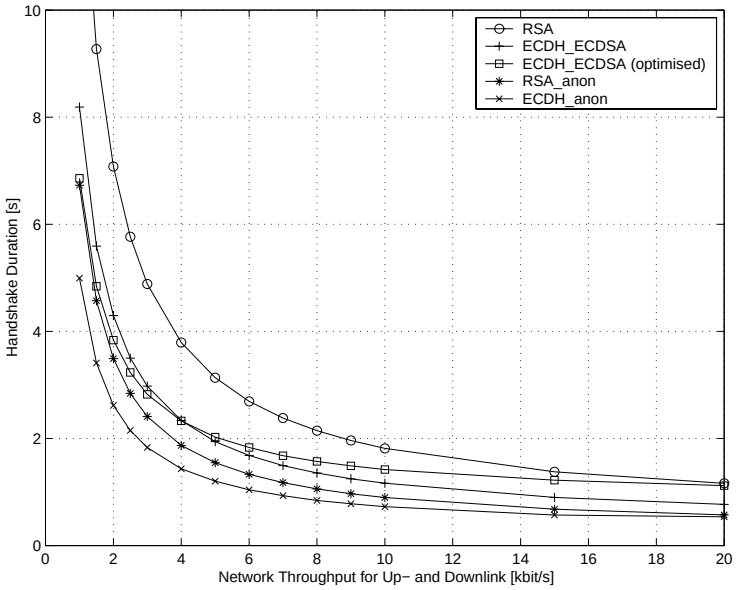


**Fig. 3.** Duration of examined WTLS handshake procedures

### 3.3 Impact on an M-Commerce Transaction

The *MeT Initiative* has specified a usage scenario for mobile payments using a SET Wallet Server wherein nonrepudiation is granted by application level digital

[4] includes WDP- and IP-overhead

**Table 2.** Size of message groups exchanged during a handshake

| | RSA | ECDH_ECDSA | RSA_anon | ECDH_ECDSA(opt) | ECDH_anon |
|---|---|---|---|---|---|
| $1^{st}$ msg. | 263 bytes | 263 bytes | 263 bytes | 263 bytes | 263 bytes |
| $2^{nd}$ msg. | 485 bytes | 287 bytes | 210 bytes | 338 bytes | 98 bytes |
| $3^{rd}$ msg. | 780 bytes | 310 bytes | 221 bytes | 154 bytes | 115 bytes |
| $4^{th}$ msg. | 116 bytes | 116 bytes | 116 bytes | – | 116 bytes |

signatures (WMLScript signText) and SET messages are exchanged between the server and merchants only. Assuming that the WAP gateway and the SET Wallet Server are both hosted by the corresponding credit institute, a secured channel between mobile devices and the server can be established by means of WTLS.

Measurements carried out with ECDSA-signing have resulted in an increase of the transaction duration by 0.11 s if no certificate has been included in the signed string and 0.35 s in case the certificate has been appended. With RSA-signing, the duration increases by 1.73 s and 2.50 s respectively. Given that most of the time is being spent in the SET Wallet Server itself [5], the slight increase is acceptable although these values are not taking a handshake, which eventually is to be performed, into account.

## 4 Conclusions

According to our performance evaluation it is obvious that, as the WTLS throughput is higher than the expected one of the underlying bearers, the impact of symmetric en- and deciphering becomes negligible. However, costs for the completion of WTLS handshakes have still to be taken into consideration. As exemplarily shown in 3.3, in case a full handshake is to be carried out prior to each transaction, a significant increase in transaction duration is to be expected—depending on the chosen key exchange suite and the available channel quality. Based on this fact, the decision whether to execute a full or an abbreviated handshake should be deliberated.

## References


1. WAP Technical Specifications Version 1.2.1, WAP Forum, URL: http://www.wapforum.org/
2. "MeT Core Specification" V0.1, MeT, 21-2-2001, URL: http://www.mobiletransaction.org/
3. Linder, D., "Transport Security for the next Generation Mobile Terminals", Master's Thesis, Royal Institute of Technology, Stockholm, Sweden, 29-11-2000
4. Schneier, B. et al., "Performance Comparison of the AES Submissions", 3-1-1999
5. Wrona, K. and Zavagli, G., "Adaptation of the SET Protocol to Mobile Networks and to the Wireless Application Protocol", Proc. European Wireless '99, Munich, Germany, October 1999