Wireless Personal Communications **29:** 303–321, 2004.
© 2004 *Kluwer Academic Publishers. Printed in the Netherlands.*

Experimental Analysis of an SSL-Based AKA Mechanism in 3G-and-Beyond Wireless Networks

GEORGIOS KAMBOURAKIS, ANGELOS ROUSKAS and STEFANOS GRITZALIS

Department of Information and Communication Systems Engineering, University of the Aegean, Samos 83200, Greece
E-mail: gkamb@aegean.gr

Abstract. The SSL/TLS protocol is a *de-facto* standard that has proved its effectiveness in the wired Internet and it will probably be the most promising candidate for future heterogeneous wireless environments. In this paper, we propose potential solutions that this protocol can offer to future "all-IP" heterogeneous mobile networks with particular emphasis on the user's side. Our approach takes into consideration the necessary underlying public key infrastructure (PKI) to be incorporated in future 3G core network versions and is under investigation by 3GPP. We focus on the standard 3G+ authentication and key agreement (AKA), as well as the recently standardized extensible authentication protocol (EAP)-AKA procedures and claim that SSL-based AKA mechanisms can provide for an alternative, more robust, flexible and scalable security framework. In this 3G+ environment, we perceive authentication as a service, which has to be performed at the higher protocol layers irrespectively of the underlying network technology. We conducted a plethora of experiments concentrating on the SSL's handshake protocol performance, as this protocol contains demanding public key operations, which are considered heavy for mobile devices. We gathered measurements over the GPRS and IEEE802.11b networks, using prototype implementations, different test beds and considering battery consumption. The results showed that the expected high data rates on one hand, and protocol optimisations on the other hand, can make SSL-based authentication a realistic solution in terms of service time for future mobile systems.

Keywords: EAP-TLS, mobile communications, performance evaluation, PKI, SSL

1. Introduction

According to beyond-3G current vision, an IP backbone will constitute the core network for all heterogeneous wireless technologies and secure communication provision will become one of the major goals of these systems. While secure sockets layer/transport layer security (SSL/TLS) protocol [1–3] is the predominant and most widely used security protocol on the wired Internet, to our knowledge, no wireless data service offers this protocol on a mobile device today.

Performance considerations using SSL in a resource-constrained environment drove wireless designers to choose a different and incompatible, mainly gateway oriented, security protocol called wireless transport level security (WTLS) for their mobile clients [4, 5]. Provided that is possible to develop a usable, in terms of performance, implementation of SSL for a handheld device as suggested in [6], we can view authentication between mobile users and network operators as a service which has to be performed at higher layers. By doing so, we can implement more secure, flexible, and reconfigurable authentication and key agreement (AKA) procedures for next generation mobile and wireless communication networks [7–9] which are independent of the underlying wireless access technology.

Considering SSL protocol and public key infrastructure (PKI) adaptation issues in future mobile networks, we propose two related SSL-based authentication and key agreement (AKA) schemes. We argue that standard packet switched (PS) domain AKA, extensible authentication protocol (EAP)-AKA and users applications authentication mechanisms can be alternatively SSL enhanced. Consequently, 3G AKA shortcomings can be tackled and scalable public key solutions can be implemented. The performance of an SSL-based user-to-network AKA scheme is experimentally evaluated, measuring service times and battery consumption under different scenarios. The results show that SSL-based authentication can be attainable, in terms of service time, in future mobile systems.

The rest of the paper is organized as follows. Section 2 provides an overview of the SSL protocol and examines how SSL and PKI can be included in future mobile communications. In Section 3, we argue about existing problems spotted in 3G's AKA and EAP-AKA mechanisms. Our alternative SSL-based AKA methods are discussed in Section 4, while Section 5 presents the measurements gathered from our experiments. Energy consumption results are presented in Section 6 and the paper is finally concluded in Section 7.

2. SSL and PKI in Mobile Communications

TSL/SSL establishes a transport-level secure channel for encrypted communications between two parties. SSL is offering many advantages, like different applications support, minimal changes at layers above and below, and is easy to develop in IP-enabled devices.

Moreover, SSL supports different protocols for creating pre-master keys (RSA, Diffie-Hellman, etc), several different cryptographic algorithms and different message authentication code (MAC) algorithms. In the context of an AKA procedure, these properties can provide the appropriate flexibility in a continuously evolving environment. The successful use of the SSL protocol in the wired Internet has proved its usability and effectiveness. Likewise, SSL can be part of an all-IP mobile environment [6]. Modern smart cards with advanced architectures that just appearing in the market can effectively store and protect the subscriber's private key, generate good pseudorandom values and take over of symmetric key (un)wrapping functions [6, 10]. Mobile's device processor can efficiently carry out the rest of the calculations needed by SSL protocol.

The ASPeCT project has demonstrated that public-key authentication is possible and GSM and universal mobile telecommunications system (UMTS) applications can coexist on a single smart card [11]. A recent study has also shown the feasibility of SSL in handheld wireless devices [6]. Similar papers [12] examine the potential use of SSL protocol in networked mobile devices using a proxy-based architecture.

Certainly to implement an AKA mechanism based on SSL, we need to utilize some sort of PKI, which is not necessarily part of the current 3G+ network core [13]. PKI is gradually being introduced in the market. Projects like ASPeCT [11] and USECA [14], Third Generation Partnership Project (3GPP) discussion documents [13, 15] especially for UMTS Release 6, as well as other recent works [16] anticipate that evolution. The eNorge 2005 strategy calls for a shared PKI for Norway [17], while advanced standards such MexE, WAP and i-mode from NTT DoCoMo have moved forward to introduce public key methods. Successful wireless PKI implementations and solutions from companies like Sonera Smarttrust, Lucent Technologies and Entrust, strengthens the assertion that PKI has become an acknowledged and promising component of standards.

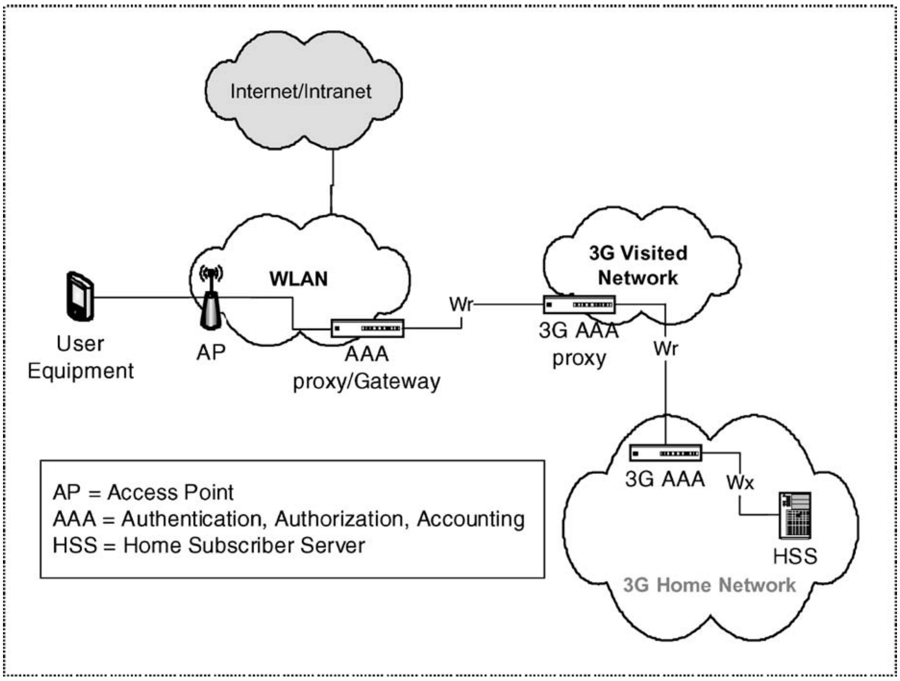

Figure 1. Wi-Fi integration in UMTS concept.

Such an infrastructure is protocol independent and heterogeneous, and can provide a wide range of security services, such as authorization and charging of mobile users for new services, authenticated and encrypted channels between network entities, as well as digital signatures and non-repudiation services. More on PKI and 3G integration requirements can be found in [7, 13, 15, 16, 18].

3. 3G AKA, EAP-AKA and Their Limitations

UMTS and GSM AKA mechanisms are based on a symmetric secret key K, which is stored in the user's universal subscriber identity module (USIM) card and in the corresponding home subscriber server (HSS). The procedure, as described in [19], is based on the challenge/response protocol. Note that AKA is used for authentication purposes at both the radio network and the IP multimedia subsystem (IM), introduced in UMTS release 5 [20, 21]. Consequently, the radio network uses international mobile subscriber identity (IMSI), whereas the IM uses network access identifier (NAI), which is of the form IMSI@realm or P-TMSI@realm where P-TMSI is the packet temporary mobile subscriber's identity. For more information about IM and NAI refer to [20] and [22], respectively.

Furthermore, current 3GPP specifications for UMTS Release 6 [23], describe an interworking architecture between UMTS and WLAN (Figure 1) where the home network is responsible for access control, while 3GPP authentication authorization and accounting (AAA) proxy, relays access control signalling to the home 3GPP AAA server. The access request is forwarded to the AAA proxy that translates the AAA request into the equivalent 3G AAA protocol request. Usually the EAP server is separate from the authenticator node, which resides closest to the user's machine (supplicant) for e.g. an AP or an 802.1X bridge. The supplicant communicates with the AAA server that provides EAP server functionality using an AAA protocol, such as RADIUS or DIAMETER.

3GPP seems to choose the EAP-AKA protocol described in [23, 24]. EAP is a general protocol for PPP authentication, which can support multiple authentication mechanisms. Consequently, EAP-AKA provides a way to exchange AKA authentication messages encapsulated within the EAP protocol.

Although several known weaknesses in GSM AKA seem to be now fixed in UMTS, there are still some “gaps” which affect the EAP-AKA mechanism too. For a detailed breakdown of 3G-AKA shortcomings refer to [7, 25–27].

In the following, we only mention additional deficiencies specific to EAP-AKA:

- The authentication procedure may require several request/response exchanges. When the user roams from one cell to another, he should gain or request authentication from his 3G AAA home server. This means that authentication efficiency should be significantly considered, since it is involved in the latency quality of the handover procedure.
- Identity privacy support can be optionally included in EAP-AKA to protect the privacy of the subscriber identity against passive eavesdropping. However, this mechanism cannot be used on the first connection with a given server and the IMSI must be sent in clear text. In addition, active attacks can be triggered from individuals that impersonate the network and try to obtain the subscriber’s IMSI.
- EAP-AKA does not support ciphersuite negotiation or protocol version negotiation.
- Other protocol attacks are also possible, for instance man-in-the-middle and negotiation attacks, as described in [24].

Last but not least, user application security, when needed, is provided by the WTLS protocol. It is well known that WTLS, at least until version 2.x, is using a WAP getaway, which is generally considered as insecure and certainly not “end-to-end”. At any rate, WTLS authentication procedure is on the one hand often anonymous, and if it is done, then it is done only once (towards the WAP gateway) for the whole time where the same WAP gateway is used.

4. Proposed AKA Mechanisms

Motivated by the aforementioned standard AKA deficiencies and technological trends mentioned in Section 2, we propose two alternative AKA procedures based on SSL and EAP-TLS, respectively. For standard PS domain authentication and key agreement we describe in short a mechanism based on SSL; for integrated 3G+/Wi-Fi networks we illustrate EAP-TLS, method [28] instead of EAP-AKA. Figures 2 and 3 depict the protocol messages flow, for standard SSL and EAP-TLS AKA mechanism respectively. All necessary adaptations are included focusing on public key operations in the client side which is generally considered as computationally weak. We note that the proposed AKA-SSL procedure protects IMSI from active or passive eavesdropping by making it part of the SSL. A short description on the necessary adaptations is following. For a detailed pure SSL and EAP-TLS protocols explanation respectively, refer to [1, 2, 28].

4.1. AKA Based on SSL

Figure 2 describes a standard SSL protocol negotiation, providing mutual authentication through X.509 subset certificates to both parties. The derived symmetric keys will serve for

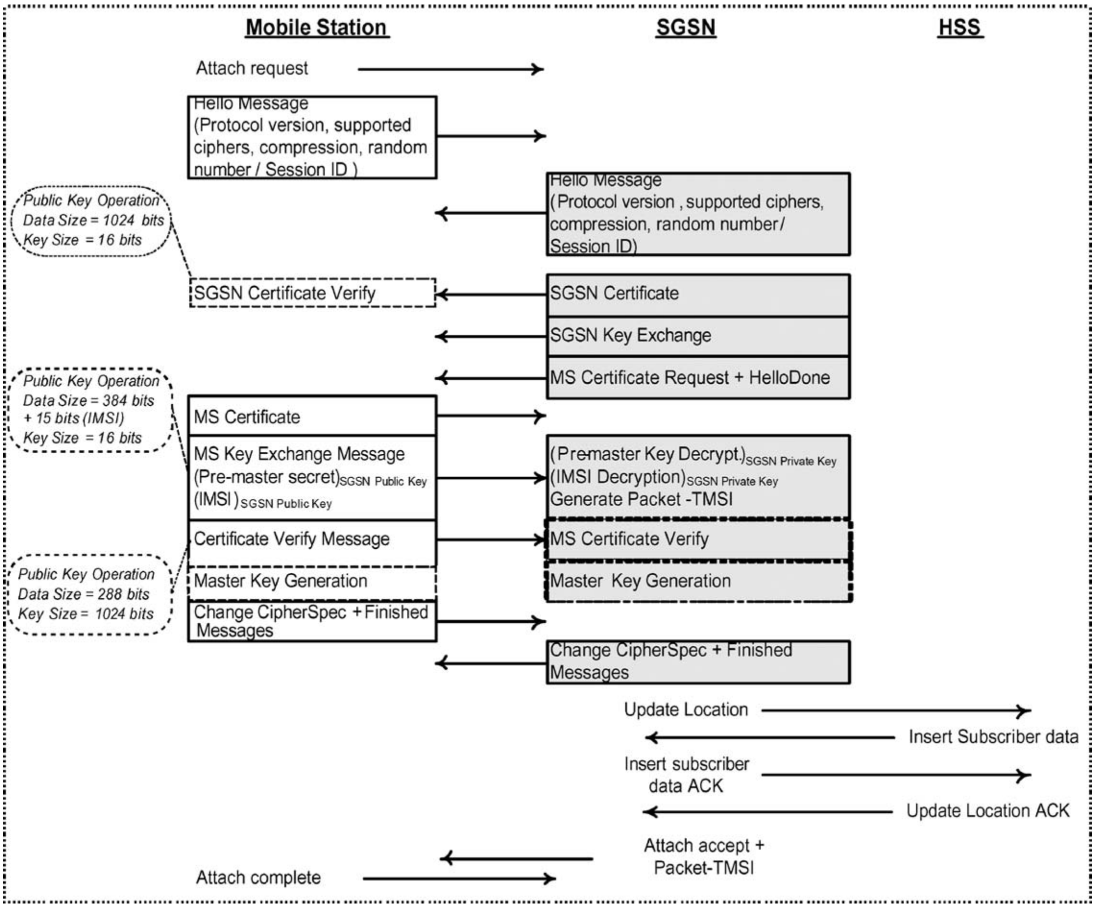
Mobile Station
SGSN
HSS
Attach request
Hello Message (Protocol version, supported ciphers, compression, random number / Session ID)
Hello Message (Protocol version , supported ciphers, compression, random number/ Session ID)
Public Key Operation Data Size = 1024 bits Key Size = 16 bits
SGSN Certificate Verify
SGSN Certificate
SGSN Key Exchange
MS Certificate Request + HelloDone
Public Key Operation Data Size = 384 bits + 15 bits (IMSI) Key Size = 16 bits
MS Certificate
MS Key Exchange Message (Pre-master secret)SGSN Public Key (IMSI) SGSN Public Key
(Pre-master Key Decrypt.)SGSN Private Key (IMSI Decryption)SGSN Private Key Generate Packet -TMSI
Certificate Verify Message
MS Certificate Verify
Public Key Operation Data Size = 288 bits Key Size = 1024 bits
Master Key Generation
Master Key Generation
Change CipherSpec + Finished Messages
Change CipherSpec + Finished Messages
Update Location
Insert Subscriber data
Insert subscriber data ACK
Update Location ACK
Attach accept + Packet-TMSI
Attach complete

Figure 2. AKA mechanism based on SSL (Packet-TMSI is unknown to SGSN).

both network signalling and user data integrity and confidentiality. At this point, we also suppose that no P-TMSI is included in the MS's attach request message to SGSN, which serves the routing area wherein the mobile is located. Later on, we will show how the procedure is extended when P-TMSI is included in the MS's attach request message to SGSN. The procedure may be slightly different, depending on the SSL cipher-suite, agreed by the communication parties.

4.2. AKA Based on EAP-TLS

Figure 3 depicts the case of EAP-TLS. EAP-TLS is based on SSL Version 3.0, and SSL handshake is performed over EAP, instead of TCP as in the Internet case. As EAP-TLS performs mutual SSL authentication, each side is required to prove its identity to the other using its certificate and its private key. The appropriate 3G AAA server is chosen based on the NAI. Note that since the client claimed his identity in the EAP-response identity packet, the EAP server should verify that the claimed identity corresponds to the certificate presented by the peer. This means that a user ID must be included in the peer certificate. From the AAA server side, a mapping from the temporary identifier (P-TMSI) to the IMSI is also required likewise,

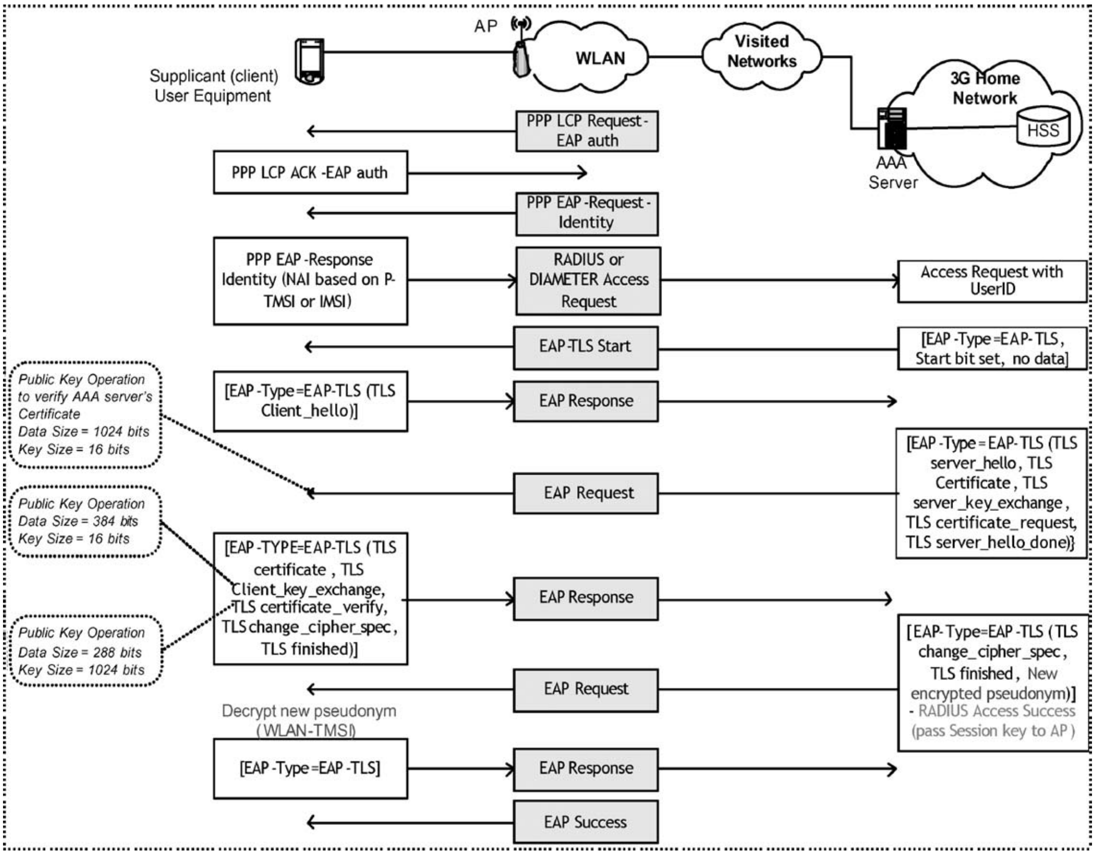
AP
WLAN
Visited Networks
3G Home Network
HSS
Supplicant (client) User Equipment
AAA Server
PPP LCP Request - EAP auth
PPP LCP ACK -EAP auth
PPP EAP -Request - Identity
PPP EAP -Response Identity (NAI based on P-TMSI or IMSI)
RADIUS or DIAMETER Access Request
Access Request with UserID
EAP-TLS Start
[EAP -Type =EAP-TLS, Start bit set, no data]
Public Key Operation to verify AAA server's Certificate Data Size = 1024 bits Key Size = 16 bits
[EAP -Type =EAP-TLS (TLS Client_hello)]
EAP Response
EAP Request
[EAP -Type = EAP-TLS (TLS server_hello, TLS Certificate, TLS server_key_exchange, TLS certificate_request, TLS server_hello_done)}
Public Key Operation Data Size = 384 bits Key Size = 16 bits
[EAP -TYPE=EAP-TLS (TLS certificate , TLS Client_key_exchange, TLS certificate_verify, TLS change_cipher_spec, TLS finished)]
EAP Response
Public Key Operation Data Size = 288 bits Key Size = 1024 bits
[EAP- Type=EAP -TLS (TLS change_cipher_spec, TLS finished, New encrypted pseudonym)] - RADIUS Access Success (pass Session key to AP)
EAP Request
Decrypt new pseudonym (WLAN-TMSI)
[EAP -Type =EAP -TLS]
EAP Response
EAP Success

Figure 3. AKA mechanism based on EAP-TLS.

supplicant must check against EAP's server certificate validity (expiration time/name/signed by trusted certification authority (CA) etc).

4.3. Resuming Sessions

To minimize the overhead of sophisticated cryptographic calculations and the number of protocol messages in both parties (MS–SGSN–AAA server), SSL defines a mechanism by which two parties can reuse previously negotiated SSL parameters. Resuming earlier sessions notably streamlines the AKA-SSL negotiation. The two hello messages in Figure 2 determine whether the session can be resumed or not. More specifically, if MS wishes to resume a previous session, then it includes its session ID in its hello message suggesting its value to SGSN/AAA server. If SGSN/AAA server agrees with that and has cached that session parameters, it responds with the same session ID in its own hello message. Otherwise, it generates a different session ID value and the full negotiation then takes place.

The entire attach procedure, incorporating our proposed AKA-SSL procedure, is illustrated in Figure 4, in the form of a message sequence diagram. The MS retrieves its P-TMSI from its non-volatile memory and places it in the attach request message, sent to the new SGSN. The P-TMSI has been previously allocated possibly by another SGSN (old SGSN) and perhaps at another routing area. Obviously, if the P-TMSI has been allocated by the new SGSN in the

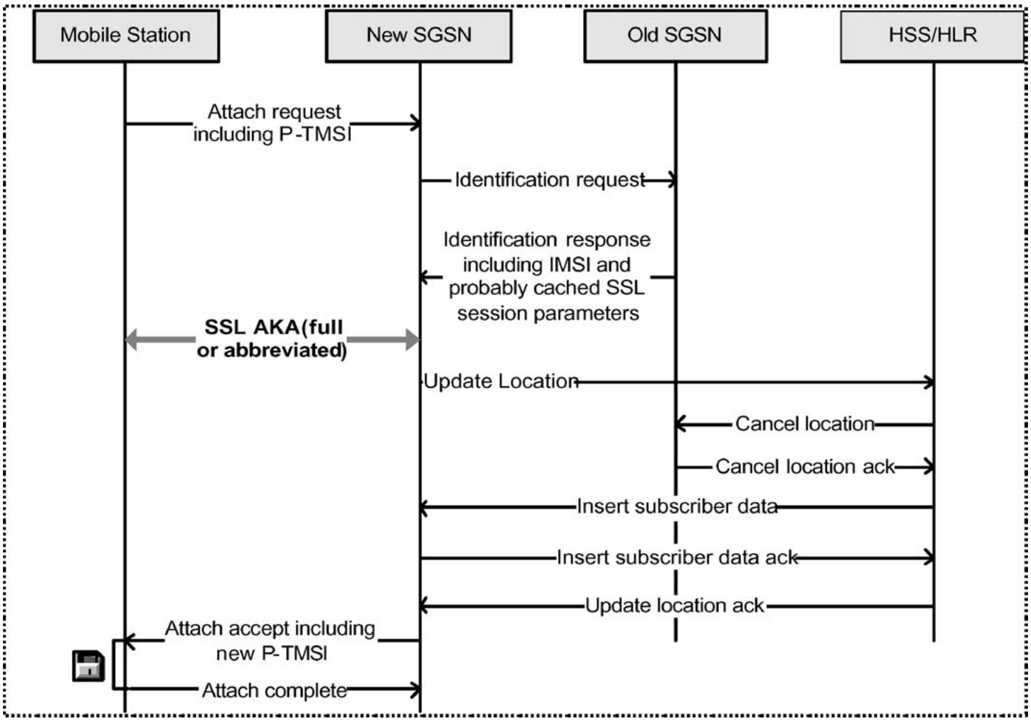
Mobile Station
New SGSN
Old SGSN
HSS/HLR
Attach request including P-TMSI
Identification request
Identification response including IMSI and probably cached SSL session parameters
SSL AKA(full or abbreviated)
Update Location
Cancel location
Cancel location ack
Insert subscriber data
Insert subscriber data ack
Update location ack
Attach accept including new P-TMSI
Attach complete

Figure 4. MS's attach procedure.

past, then the new SGSN and the old SGSN are identical and the new SGSN is aware of the IMSI of the MS. A decision flowchart describing whether session can by resumed or not, is also presented in Figure 5.

A comparable procedure could be initiated for EAP-TLS authentication. In that case, session resumption is straightforward, as the home AAA server is caching SSL session parameters. Also note that any AAA server (WLAN or 3G) that resides near the supplicant can provide for authentication, thus improving mobility. This is possible as the "Any-AAA server" can pre-exchange cross-reference certificates with the home AAA server, or both can have a signed certificate from a common root CA. Accounting details could be transferred in batch, according to bilateral pre-arrangements.

Although session resumption offers a great deal of convenience and efficiency to both parties involved, systems should exercise some care in employing it. When a single key is employed, encryption inevitably becomes less secure, as more information is protected and the time passes. So the SGSN/AAA server has to set thresholds on the number of resumptions allowed per session, as well as on the time elapsed between consecutive resumptions per session. If one of the constraints is not met then the full negotiation should be mandatory.

4.4. AKA-SSL Procedure in a Serving Network

In case the subscriber is using his mobile device in a serving network, AKA-SSL procedure remains as is, with the assumption that the home network CA and the serving network CAs, have pre-exchanged cross reference certificates. In that case, SGSN is bound to send to MS the corresponding cross-reference certificate too. Of course, another option might be to have

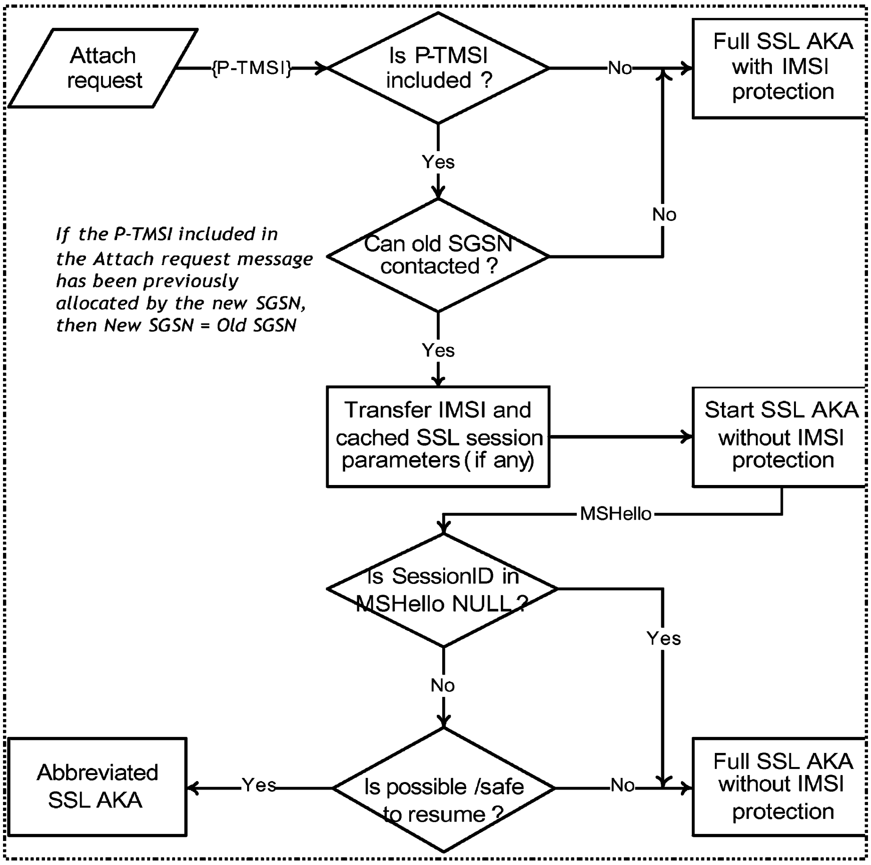
Attach request
{P-TMSI}
Is P-TMSI included ?
No
Full SSL AKA with IMSI protection
Yes
No
If the P-TMSI included in the Attach request message has been previously allocated by the new SGSN, then New SGSN = Old SGSN
Can old SGSN contacted ?
Yes
Transfer IMSI and cached SSL session parameters (if any)
Start SSL AKA without IMSI protection
MSHello
Is SessionID in MSHello NULL?
Yes
No
Abbreviated SSL AKA
Yes
Is possible /safe to resume ?
No
Full SSL AKA without IMSI protection

Figure 5. Checking whether a session can be resumed.

a common Root CA, acting as a trusted third party (TTP), and being entrusted by both the serving and home network CAs. A hierarchical approach can be proposed as well to support inter-provider trust.

4.5. Further Adaptation Issues

Some further issues need to be resolved before the proposed SSL-based authentication is introduced. Currently 2.5G and 3G systems, as discussed in Section 2, lack such a large-scale of infrastructure to authorize and consequently charge mobile users for new services, as well as to provide digital signatures and non-repudiation services. However, in the years to come it is very likely that mobile operators will incorporate PKI technology or become associated to TTPs.

Integration between 3G mobile systems and PKI has not been standardized yet. However, recent 3GGP discussion documents deal with that particular subject. Apparently, possible solutions to IP connect PKI entities to 3G+ core network architecture, are Gateway GPRS support node (GGSN) and SGSN nodes, with GGSN being the "natural" choice. Certainly, other possible and perhaps more optimal solutions for e.g. gateway or proxy oriented, that do not affect SGSN or GGSN signalling can be proposed. For a more detailed discussion refer to [15, 18].

Another barrier for someone to implement a straight AKA-SSL procedure between an MS and 2.5G-SGSN is the fact that currently there is not direct IP connectivity between MS and SGSN. However, 3G-SGSN will communicate towards all directions (RNC, GGSN) using IP. Even more, it is very likely that SGSN will be finally integrated with GGSN, at a later time. Regardless of which node will provide AKA functionality, it is more efficient to implement the AKA procedure as a service, provided to the user in an "anywhere – anytime" manner. According to 3G+ all-IP vision, this technology independent approach shall probably be more suitable.

On the other hand, the application of 3G+ SSL-based (EAP-TLS) authentication mechanisms into Wi-Fi networks, supposes that AAA procedures of a mobile Wi-Fi user can be controlled in a "centralized" or "semi-centralized" way by his home core 3G+ network. According to this arrangement, the Wi-Fi networking settings are considered as multiple entry points of a common 3G+ infrastructure. A Wi-Fi user needs to know only his home 3G+ network operator, who is responsible to establish and maintain roaming agreements (RAs) with various ending WLAN operators. Considering the high number of operators expected and the diversity of their services, a layered/hierarchical approach for RA settlement would probably fit better. Depending on the RA between the two operators, the user may receive Internet access through his home 3G+ network, via the Wi-Fi network, or directly through the current Wi-Fi access network, after being authenticated by his home 3G network, Obviously, such a solution also assumes that the user has a dual mode mobile station supporting both WLAN and UMTS, or the WLAN device can be linked with a UE, which supports 3G+ SIM capabilities e.g. Bluetooth, USB or IrDA. Finally, it is worth mentioning that EAP Tunnelled TLS (EAP-TTLS) protocol was also under consideration in 3GPP to support 3G and WLAN integration, but as is explained in [29] it was vulnerable to man-in-the-middle attacks.

5. Experimental Framework for AKA-SSL/TLS

5.1. Test-Bed Setup

In order to evaluate the performance of an AKA-SSL mechanism, we constructed an experimental hardware and software architecture. The objective was to evaluate the performance of the proposed AKA-SSL methods in terms of service times and thus show that this mechanism is realizable with current and future technology, while EAP-TLS performance is left for future work. The topology of our test architecture is illustrated in Figure 6(a). The presumed mobile device is a Compaq iPAQ H3970 Pocket PC (PPC) that uses Windows PPC 2002 operating system. The client uses a Nokia D211 dual GPRS class 7/WLAN IEEE802.11b PCMCIA card inserted in iPAQ's expansion pack plus module. The PPC incorporates a 400 MHz Intel X-Scale PXA250 CPU and has 64 MB of RAM and 48 MB of flash ROM available. It also utilizes a user-accessible section of ROM that can hold approximately 22 MB of data, applications, and other files. At the other end, the server machine has dual processor Pentium III 600 MHz with 256 MB RAM, running the Windows 2000 professional SP4 operating system. The server has also a WAN connection available. Comparable test-beds for GPRS and WAP performance evaluation can be found in the literature [30–32].

We wrote the applications in Microsoft's Embedded C++ version 4.0 and employed the well-known open-source Apache-style license OpenSSL toolkit in version 0.9.7b [3, 33, 34] to make them SSL enabled. Lightweight SSL packages like Java 2 Micro Edition (J2ME) Kilobyte-SSL from Sun and RSA's Bsafe SSL-C/SSL-J, offer certificate based authentication

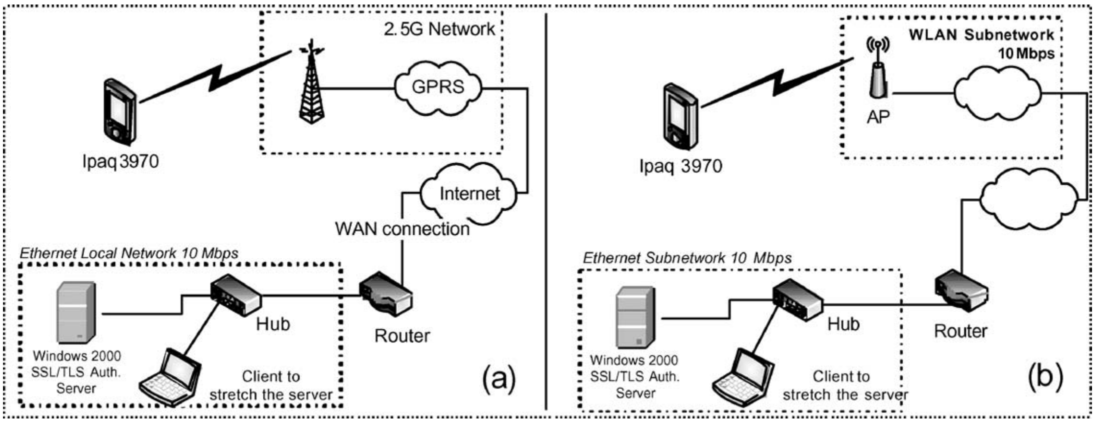
2.5G Network
GPRS
Ipaq 3970
Internet
WAN connection
Ethernet Local Network 10 Mbps
Hub
Router
Windows 2000 SSL/TLS Auth. Server
Client to stretch the server
(a)
WLAN Subnetwork 10 Mbps
AP
Ipaq 3970
Ethernet Subnetwork 10 Mbps
Hub
Router
Windows 2000 SSL/TLS Auth. Server
Client to stretch the server
(b)

Figure 6. Experiment set-up.

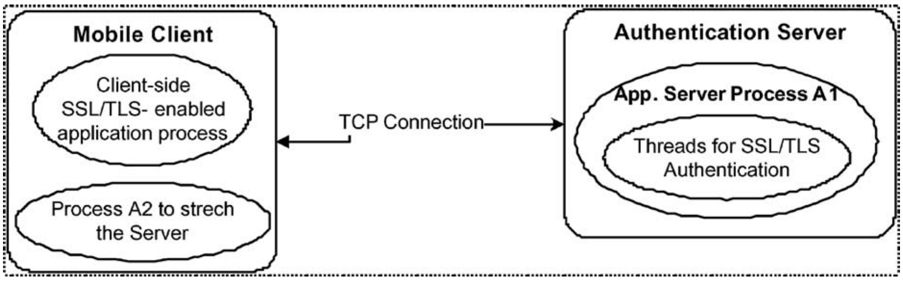
Mobile Client
Client-side SSL/TLS- enabled application process
Process A2 to strech the Server
TCP Connection
Authentication Server
App. Server Process A1
Threads for SSL/TLS Authentication

Figure 7. Software architecture.

from server's side only, while Java performance in crypto code is by far worse than C++ performance [3].

The experimental software architecture is illustrated in Figure 7. One process is running on our SSL enabled authentication server: Process A1, which opens a TCP-SSL listening socket and waits for transactions. A1 is multi-threaded, that is when it receives a message, it dispatches a thread to process and respond to the request. Process A2, running on a separate machine, generates a large number of SSL transactions to virtually load the server. The times between successive SSL authentication requests follow the negative exponential distribution.

We tried to minimize client's application demand in processing power and memory capacity, by removing dispensable memory and power consuming calls to OpenSSL functions. Therefore, we left out functions that load libraries with error strings, verify certificates paths, certificates chain depths above four and we enabled the client to support only SSL version 3 and TLS version 1. Likewise, we excluded from the client and the server, support for ciphers and MACs algorithms that are generally considered anonymous or weak, e.g. MD5.

The handshake and authentication procedures are mutual, meaning that both the client (MS) and the server (SGSN) exchange their certificates, which are kept locally along with the corresponding trusted CAs public keys list. We decided to evaluate a depth-two certificate chain schema in order to weigh up a serving or visited network authentication by stretching the client even more. Both parties check certificates validity, against time expiration and issuing CA. Neither party check certificates validity agasinst any revocation list. Only the authentication server is bound to do so by checking against MS's P-TMSI (mapping it to the correct IMSI).

Table 1. Service times (iPAQ\GPRS)

	Mobile station (iPAQ) (ms)				Authentication server (ms)
	NRT	RPT	THT	TST	SCRT
Average	1391	1427	9.480	10.950	8.469
S.D.	610	620	750	980	760

All RSA keys are 1024 bits in length, pre-master secret exchange is based on ephemeral Diffie-Hellman key with RSA signatures, thus supporting forward secrecy and the resulting symmetric SSL/TLS session key is 128 or 256 (for AES256) bits long. The complete cipher suite algorithms that our applications employed are: EDH-RSA for key exchange with key size 512 or 1024 bits, DES-CBC3 or AES256 for encryption and SHA for hashing. Both suites can be characterized as heavy, compared e.g. to weaker but faster Kilobyte-SLL's cipher-suites options (RSA_RC4_128_MD5 and RSA_RC4_40_MD5).

Figure 6(b), depicts our hardware architecture for normal SSL authentication over 802.11b link. Client and server machines, the corresponding applications and the setup options remain the same. We used a D-link DWL-900AP+ wireless access point to connect to our WLAN sub-network. Our server resides in another sub-network and ping times from one sub-network to another showed an average value of 60 ms.

5.2. Measurement Results

We run our experiments with various values of the request arrival rate λ for process A2 which adds virtual load to the server process A1, during different days and peak hours times. The GPRS coding scheme was CS-1 (9.05 kb/s) and the time slots for GPRS were varying from 3 to 4, thus having wireless network speeds in the range from 27 to 36 kb/s. We tracked and measured the following times in the mobile's client process:

(a) Network response time (NRT): Time to complete a connection to the server socket. It includes network round-trip plus client and server processing related to the acceptance of the connection.
(b) Request preparation time (RPT): Elapsed time before the actual SSL handshake. This time is NRT, plus the preparation time e.g. MS's time to load the certificates.
(c) Total handshake time (THT): Elapsed time from MS hello to finished message.
(d) Total SSL call setup time (TST): Total elapsed time until both parties acquire the symmetric key and are ready to start the actual communication (it is actually the sum of RPT and THT).
At the server side, we measured the following time:
(e) Time to serve client's request (SCRT): Elapsed time from server hello message until MS's request has been accepted and served.

During our experiments, we gathered 1000 measurements of the aforementioned times, from an equal number of transactions initiated by our client. We present the corresponding average and standard deviation values in milliseconds in Table 1 and the probability density functions of these time durations in Figures 8(a) and 8(b).

In Figure 8(a), we can see the relation between NRT and RPT. We easily observe that RPT plot is actually a right shift of about 0.1 s, of the NRT plot, which means that the preparation process takes nearly constant time to complete and the total time before SSL handshake is mainly dependent on the distribution of NRT i.e. the network speed.

Table 2. Memory requirements for the client and server

Characteristics	Memory (KB)
Disk or Flash ROM space for OpenSSL.dll files	1132
Disk or Flash ROM space for MS's application	16
EPROM space in smartcard for client's certificate	4.07
(same for the server)	4.07
EPROM space in smartcard for one trusted CA-root certificate	1.02
RAM space for process server application	100

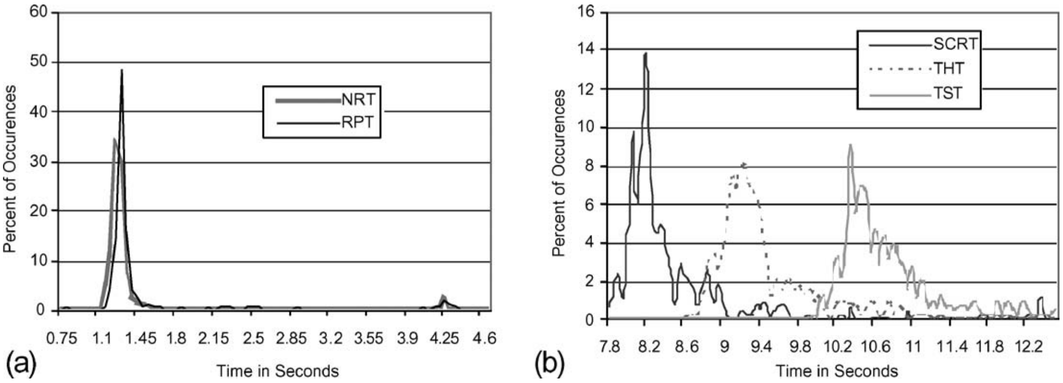

Figure 8. GPRS service times (iPAQ\GPRS).

Similarly, in Figure 8(b) we present the relation between TST, THT and SCRT. We notice that the THT plot is a right shift of about 1 s of the SCRT plot. This 1 s includes half round-trip time (0.7 s) from the initial MS hello message. All the three plots have nearly the same distribution. This can be explained from the protocol's 10 exchanged messages, in other words approximately 5 NRTs. Therefore, 1391 ms average round-trip time multiplied by 5 equals 7 s, plus other times for calculations, verifications and hashing, we get the total average SSL setup time of 10.95 s. We also mention that all measurements gathered, were highly immune to server's workload as this was generated by process A2.

Finally, Table 2 presents the resources in kilo bytes, which comprise indicative values for a mobile device to run our scenario.

5.3. Comments on the Measurements Results

An average time of 11 s, as it appears in Table 1, is certainly an unacceptable authentication time duration for the users of a wireless 3G+ devices, since a real 2.5G standard AKA mechanism, assuming that someone activates his device in a roaming network, takes about 5–7 s to complete. However, comparing TST with actual WTLS service times, we note that WTLS is in several cases even slower, while comparable results with Kilobyte-SSL, 20 MHz Palm client CPU and server-side only certificate verification show a time of approximately 10 s [6].

Nevertheless, 11 s is an overestimated result, because we have to subtract the extra network delay, derived from the fact that our server did not reside inside the provider's core network. Performing measurements with a ping tool, we discovered that the extra time spend in each roundtrip was about 220 ms. Thus, the actual TST time is below 10 s (9.85).

Furthermore, someone has to consider that the ciphers used were the heaviest possible and that the exchanged certificates provide for authentication in a serving network. In the present 2.5 and 3G specifications, this means that the serving network has to ask the subscriber's home network, to provide it with authentication vectors to authenticate the user. The size of the certificates as presented in Table 2, not only decelerates the certificate verification process in each party, but at the same time adds several extra bytes to the relevant handshake messages.

Additionally, the network speed was relatively low, since the expected network speeds for 3G will be 144 kb/s up to 348 kb/s for wide and up to 2Mb/s for low coverage and mobility, which will substantially reduce round-trip times. A rough calculation assuming (the minimum) 144 kb/s network speed could diminish NRT to 348 ms showing an improvement of a factor of four. Even a higher GPRS coding scheme, if offered by the operator and possible by the prevailing link conditions, could improve protocol's handshake performance considerably [32].

Excluding hardware improvements, further optimizations may come from either the protocol structure (messages exchange) or the network architecture. As we discussed in Section 4.1, session resumption option provides a dramatic performance improvement. Making the appropriate adjustments to our client and server software we measured an average of 2.1 s in THT client's time (77% improvement). More important, we would consequently expect corresponding improvement in network throughput, as a significant number of sessions are resumed. In any case, the client will need for each session that will attempt to resume, a maximum buffer size of 80 bytes, which is 32 bytes for session ID and 48 bytes for the pre-master secret.

Finally, during the SSL handshake, the server must wait for a client message and vise versa. OpenSSL for example can buffer network output for increased performance. So, it is often computationally cheaper and network faster (considering round-trip times) to generate a number of messages and transmit them all at once [3]. The simplest buffering approach is to set a fixed buffer size and transmit the data when it gets full. It's also necessary to flush the buffer after each transmission. In all cases the choice of buffer size is critical, as in some cases improves latency and in other cases makes it worse. Recent studies also showed that session reuse could be further improved, using an SSL session aware dispatcher, when the operator is planning to install a cluster of SSL authentication servers [35] and as is mentioned in [36] the SSL's handshake protocol time can be improved up to 5.7 times.

5.4. Enhancing the Client's Processing Power

It was important to test the effect of the client computing power on the overall AKA mechanism. Thus, we replaced the iPAQ device with a Compaq Presario laptop machine which incorporates a Pentium 4 2.2 GHz processor and 256 MB RAM. The rest of our initial test-bed setup remained unchanged. We logged another set of 1000 measurements of the aforementioned times, from an equal number of transactions initiated by our laptop client.

The new measurement results are presented now in Table 3 and the probability density functions of THT, TST and SCRT time durations are shown in Figure 9.

We observe that RPT is equal to NRT, meaning that the preparation phase is zero, which is expected for the fast 2.2 GHz machine. Once more, all the three plots are shifted versions of the same distribution.

Table 3. Service times (laptop\GPRS)

	Mobile station (laptop) (ms)				Authentication server (ms)
	NRT	RPT	THT	TST	SCRT
Average	1342	1343	5.503	6.840	4.354
S.D.	574	574	1298	1381	793

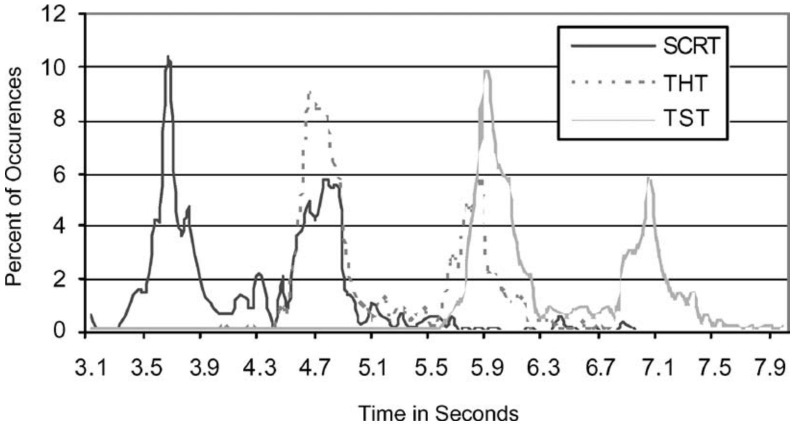

Figure 9. GPRS service times (laptop\GPRS).

5.5. Performing AKA-SSL Over an 802.11b WLAN

We used the test bed depicted in Figure 6(b) to measure the processing cost of the *pure* (excluding network transfer times) SSL handshake with the available mobile device. Performing SSL handshake in a fast wireless environment (10 Mb/s), network times are nearly diminished and therefore, the remaining time is very close to the net handshake protocol's performance time. The results are presented in Table 4 and Figure 10.

TST time is very close to THT, as NRT is very small and standard deviation of all values remains low, supporting our reasoning. Remember that THT measured with iPAQ over GPRS was 9.480 ms. Is nearly the same as the current THT (2.605 ms) plus the total round-trip times in GPRS. The aforementioned results are also indicative of the performance expected when EAP-TLS is employed. Concluding, Figure 11(a) gives a picture of THT time comparison between the three deployed scenarios.

6. Battery Measurements

Energy consumed by secure wireless sessions on mobile devices, can be categorized in two major groups that include cryptographic computations for secure session establishment and data transactions and message exchanges during handshake and secure data transactions. Related work [37] conducted on PPC in a wireless 11 Mbps LAN and WTLS using Diffie-Hellman key exchange protocol, shows a consumption of 1062 mJ, 7% of which is consumed by the cryptographic computations and 93% by the message exchanges. A comprehensive analysis of the energy requirements of SSL using OpenSSL and iPAQ, can also be found in [38].

Table 4. Service times (iPAQ\WLAN)

	Mobile station (iPAQ) (ms)				Authentication server (ms)
	NRT	RPT	THT	TST	SCRT
Average	103	467	2.605	2.730	2.600
S.D.	377	459	370	376	161

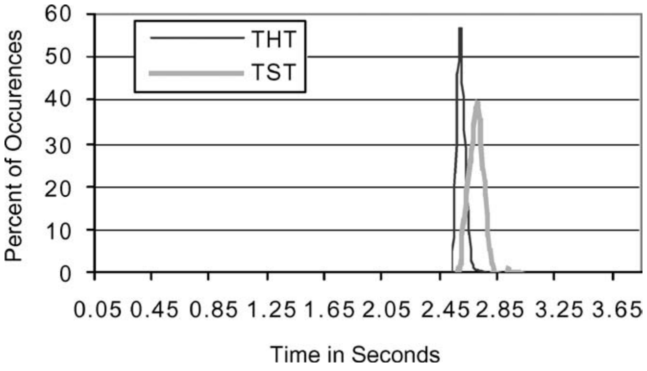

Figure 10. TST and THT service times (iPAQ\WLAN).

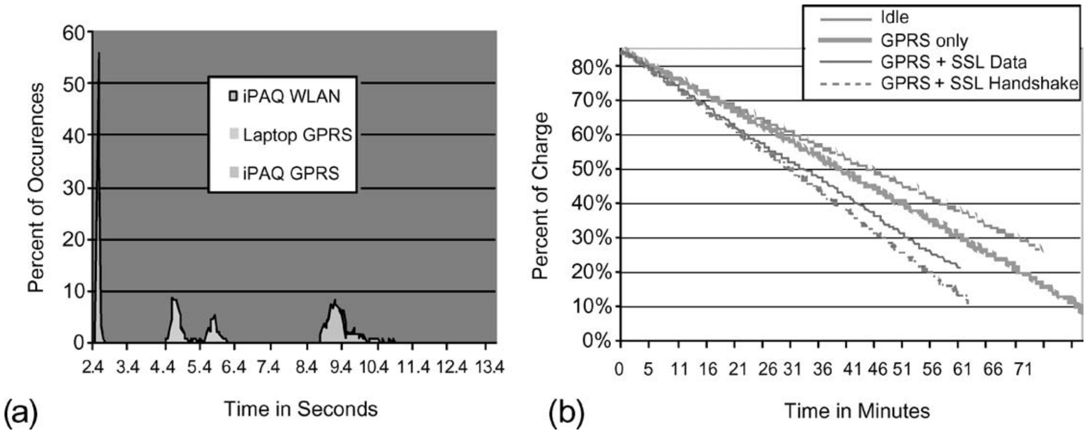

Figure 11. Comparing THT service times and battery consumption.

The aforementioned works cover SSL's energy consumption to a great deal. In this paper we will bound our study, measuring battery consumption for four different situations. (a) Device is idle; (b) device is connected to GPRS without performing any transaction; (c) device is connected to GPRS performing an SSL handshake every 10 s and (d) device connected to GPRS with an SSL connection active, sending and receiving an amount of 10 KB of data continuously.

iPAQ has a lithium-polymer 400 mAh rechargeable main battery. Expansion pack plus module where the Nokia D211 card was inserted has also a removable battery. This battery was removed during our tests. In all scenarios iPAQ's screen was always turned on and the battery initialization status was 80% charged. Energy consumption for Nokia D211 card when it is connected to GPRS is 23 mA (idle) and 150–200 mA (downloading), respectively.

Measurement results which are presented in Figure 11(b), show that iPAQ can be SSL connected, receive and sent ciphered data for somewhat less than 2 h, with screen always turned on. Likewise, it can be connected to the GPRS network for almost 2.5 h. At any rate, protocol's message compression could be an important contributing factor, as decompression is always faster and more energy-efficient from compression. This means that we can enable server-side compression and client-side decompression for both handshake and data transaction messages to achieve better battery administration.

7. Conclusions and Future Work

The greater drawback of the existing AKA procedures is that they are dependent on the underlying network infrastructure and thus cannot offer a dynamic and flexible authentication and key agreement mechanism, acting rather statically. On the contrary, in next generation mobile environments more flexible, dynamic and scalable security mechanisms are necessary in order to support on-demand services and all-IP end-to-end solutions, integrated with the Internet and over heterogeneous wireless and wired technologies.

In this paper, we described two SSL-based authentication procedures, which take advantage of an underlying PKI, for future mobile communication systems. We examined and evaluated the performance of one of the proposed schemes that also protects user's IMSI. We tried to estimate protocol's handshake time and battery consumption by testing different devices and set-ups. Taking into consideration the forthcoming 3G+ networks speed, SSL protocol optimisations, as well as hardware improvements we conclude that SSL-based authentication can be possible in terms of service times. Simultaneously, it can deliver the appropriate flexibility and scalability to network operators and a high level of trust and assurance to end-users.

Topics to be further investigated include roaming and authentication, layered oriented approaches for inter-domain issues, like cross-certification, EAP-TLS authentication performance and battery consumption improvements.

Acknowledgements

We would like to thank Mr. Dimitrios Geneiatakis for helping us with the network measurements.

References

1. A. Frier, P. Karlton and P. Kocher, "The SSL 3.0 Protocol Version 3.0", http://home.netscape.com/eng/ssl3/draft302.txt.
2. T. Dierks and C. Allen, "The TLS Protocol Version 1.0", IETF RFC 2246, January 1999.
3. E. Rescorla, *SSL and TLS Designing and Building Secure Systems*, Addison-Wesley, 2001.
4. WAP forum WAP-217-WPKI, "Wireless Application Protocol Public Key Infrastructure Definition", www.wapforum.org/what/technical.htm.
5. R. Khare, "W* Effect Considered Harmful", *IEEE Internet Computing*, Vol. 3, No. 4, pp. 82–92, July/August 1999.
6. V. Gupta and S. Gupta, "Experiments in Wireless Internet Security", in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC 2002)*, No. 1, pp. 859–863, March 2002.

7. G. Kambourakis, A. Rouskas and S. Gritzalis, “Using SSL in Authentication and Key Agreement Procedures of Future Mobile Networks”, in *Proceedings of the 4th IEEE International Conference On Mobile and Wireless Communication Networks (MWCN 2002)*, pp. 152–156, September 2002.
8. S. Dixit and R. Prasad (eds.), *Wireless IP and Building the Mobile Internet*, Artech House, 2003.
9. D. Wisely, P. Eardlay and L. Burness, *IP for 3G,* Wiley, 2002.
10. N. Duane and J. Brink, *PKI Implementing and Managing E-Security*, Berkeley, RSA press, 2001.
11. ASPeCT Project, “Securing the Future of Mobile Communications”, http://www.esat.kuleuven.ac.be /cosic/aspect, 1999.
12. M. Burnside, D. Clarke, T. Mills, S. Maywah, S. Devadas and R. Rivest, “Proxy-based Security Protocols in Networked Mobile Devices”, in *Proceedings of ACM SAC 2002 Conference*, Madrid, Spain, pp. 265–272, 2002.
13. 3GPP TSG, “Using PKI to Provide Network Domain Security”, *Discussion Document*, (S3-010622 SA WG3 Security – S3#15bis), November 2000.
14. USECA Project, “UMTS Security Architecture: Intermidiate Report on a PKI Architecture for UMTS”, *Public Report*, July 1999.
15. 3GPP TSG, “Architecture Proposal to Support Subscriber Certificates”, *Discussion and Approval Document*, Tdoc S2-022854, October 2002.
16. G. Kambourakis, A. Rouskas and S. Gritzalis, “Introducing PKI to Enhance Security in Future Mobile Networks”, in *Proceedings of the IFIPSEC’2003 18th IFIP International Information Security Conference*, pp. 109–120, Athens, Greece, May 2003.
17. eNorge 2005, *Naerings – og handelsdepartmentet*, 2002.
18. 3GPP Technical Specs, “Bootstrapping of Application Security Using AKA and Support of Subscriber Certificates”, *System Description*, TS ab.cde v.3.0, September 2003.
19. 3GPP Technical Specs, *Security Architecture*, TS 33.102 v.5.1.0, December 2002.
20. 3GPP Technical Specs, *Access Security for IP-Based Services*, TS 33.203 v.6.0.0, September 2003.
21. Y. Lin and A. Pang, “An All-IP Approach for UMTS Third-Generation Mobile Networks”, *IEEE Network*, pp. 8–19, September/October 2002.
22. 3GPP Technical Specs, *3GPP System to WLAN Interworking*, TS 24.234 v.0.2.0 Release 6, November 2003.
23. 3GPP Technical Specs, *WLAN Interworking Security*, TS 33.cde v0.1.0, July 2002.
24. J. Arkko and H. Haverinen, “EAP-AKA Authentication”, <draft-arkko-pppext-eap-aka-11.txt>, October 2003.
25. 3GPP Technical Specification, *A guide to 3rd Generation Security*, TR 33.900 v.1.2.0, January 2000.
26. T. Aamodt, T. Friiso, G. Koien and O. Eilertsen, *Security in UMTS – Integrity*, Telenor R&D, February 2001.
27. V. Niemi and K. Nyberg, *UMTS Security*, Wiley, 2003.
28. IETF RFC 2716, “PPP EAP-TLS Authentication Protocol”, October 1999.
29. N. Asokan, N. Valtteri and K. Nyberg, “Man-in-the-Middle in Tunnelled Authentication”, Nokia Research Center, October 2002.
30. R. Chakravorty and I. Pratt, “Performance Issues with General Packet Radio Service”, *Journal of Communication and Networks,* 2002, submitted.
31. R. Chakravorty, J. Cartwright and I. Pratt, “Practical Experience with TCP over GPRS”, in *Proceedings of IEEE GLOBECOM 2002*, Taipei, November 2002.
32. J. Korhonen, O. Aalto, A. Gurtov and H. Laamanen, “Measured Performance of GSM HSCSD and GPRS”, in *Proceedings of the IEEE International Conference On Communications (ICC’01)*, Helsinki, June 2001.
33. The OpenSSL project web page, http://www.openssl.org.
34. J. Viega, M. Messier and P. Chandra, *Network Security with OpenSSL*, O’Reilly, 2002.
35. G. Apostolopoulos, V. Peris, P. Pradhan and D. Saha, “Securing Electronic Commerce: Reducing the SSL Overhead”, *IEEE Network Magazine*, No 4, pp. 8–16, July/August 2000.
36. P. Nachiketh, R. Srivaths, R. Anand and L. Ganesh, “Optimizing Public-Key Encryption for Wireless Clients”, in *Proceedings of the IEEE International Conference on Communications (ICC 2002)*, No 1, pp. 1050–1056, April 2002.
37. R. Karri and P. Mishra, “Minimization of Energy Consumption of Secure Wireless Session with QOS Constraints”, in *Proceedings of IEEE International Conference on Communications*, New York city, NY, April 2002.
38. R. Nachiketh, R. Srivaths, A. Raghunatan and J. Niraj, “Analysing the Energy Consumption of Security Protocols”, in *Proceedings of ACM ISLPED 2003 Conference*, Seoul, August 25–27, 2003, pp. 30–35.

Georgios Kambourakis was born in Samos, Greece, in 1970. He received his Diploma in Applied Informatics from the Athens University of Economics and Business (AUEB) in 1993. Today he is a PhD student in the department of Information and Communications Systems Engineering of the University of Aegean (UoA) and a postgraduate student in "Master in Education" program in the Department of Social Studies of the Hellenic Open University. His research interests are in the fields of mobile and ad-hoc networks security, security protocols, public key infrastructure and mLearning. Since 2001, he is a visiting lecturer in the department of Information and Communications Systems Engineering of the UoA. He is a member of the Greek Computer Society.

Angelos Rouskas was born in Athens, Greece, in 1968. He received the five-year Diploma in Electrical Engineering from the National Technical University of Athens (NTUA), the M.Sc. in Communications and Signal Processing from Imperial College, London, and the PhD in Electrical and Computer Engineering from NTUA. He is an assistant professor in the Department of Information and Communication Systems Engineering of the University of the Aegean (UoA), Greece, and Associate Director of the Computer and Communication Systems Laboratory. Prior to joining UoA, Dr. Rouskas worked as a research associate at the Telecommunications Laboratory of NTUA, in the framework of several European and Greek funded research projects, and at the network performance group of the Greek Cellular Operator CosmOTE S.A. His current research interests are in the areas of resource management of mobile communication networks, mobile and ad-hoc networks security, and pricing and congestion control in wireless and mobile networks and he has several publications in the above areas. He is a reviewer of several IEEE, ACM and other international journals and has served as a technical program committee member in several conferences. Dr. Rouskas is a member of IEEE and of the Technical Chamber of Greece.

Stefanos Gritzalis was born in Greece in 1961. He holds a BSc in Physics, an MSc in Electronic Automation, and a PhD in Distributed Systems Security, all from the University of Athens, Greece. Currently he is an associate professor at the Department of Information and Communication Systems Engineering, University of the Aegean, Greece, and an associate director of the *Information and Communication Systems Security Laboratory* (www.icsd.aegean.gr/info-sec-lab). He has been involved in more than 30 national and CEC funded R&D projects in the areas of information and communication systems. His published scientific work includes seven books or chapters in books (in Greek) on information and communication technologies topics, and more than 60 journals, and national and international conference papers. The focus of these publications is on information and communication systems security. He has served on program and organizing committees of national and international conferences on informatics and is a reviewer for several scientific journals. His professional experience includes senior consulting and researcher positions in a number of private and public institutions. He was a member of the board (secretary general, treasurer) of the Greek Computer Society. He is a member of the ACM and IEEE Computer Society.