EURASIP Journal on Wireless Communications and Networking 2004:1, 184–197

Performance Evaluation of Public Key-Based Authentication in Future Mobile Communication Systems

Georgios Kambourakis
Department of Information and Communication Systems Engineering, University of the Aegean, 83200 Karlovassi, Samos, Greece
Email: gkamb@aegean.gr

Angelos Rouskas
Department of Information and Communication Systems Engineering, University of the Aegean, 83200 Karlovassi, Samos, Greece
Email: arouskas@aegean.gr

Stefanos Gritzalis
Department of Information and Communication Systems Engineering, University of the Aegean, 83200 Karlovassi, Samos, Greece
Email: sgritz@aegean.gr

Received 1 September 2003; Revised 12 February 2004; Recommended for Publication by Alfred Hanssen

While mobile hosts are evolving into full-IP enabled devices, there is a greater demand to provide a more flexible, reconfigurable, and scalable security mechanism in mobile communication systems beyond 3G (B3G). Work has already begun on such an "all-IP" end-to-end solution, commonly referred to as 4G systems. Fully fledged integration between heterogeneous networks, such as 2.5G, UMTS, WLAN, Bluetooth, and the Internet, demands fully compatible, time-tested, and reliable mechanisms to depend on. SSL protocol has proved its effectiveness in the wired Internet and it will probably be the most promising candidate for future wireless environments. In this paper, we discuss existing problems related to authentication and key agreement (AKA) procedures, such as compromised authentication vectors attacks, as they appear in current 2/2.5G/3G mobile communication systems, and propose how SSL, combined with public key infrastructure (PKI) elements, can be used to overcome these vulnerabilities. In this B3G environment, we perceive authentication as a service, which has to be performed at the higher protocol layers irrespective of the underlying network technology. Furthermore, we analyze the effectiveness of such a solution, based on measurements of a "prototype" implementation. Performance measurements indicate that SSL-based authentication can be possible in terms of service time in future wireless systems, while it can simultaneously provide both the necessary flexibility to network operators and a high level of confidence to end users.

Keywords and phrases: authentication and key agreement, SSL, PKI, mobile communications, performance evaluation.

1. INTRODUCTION

In the past few years, there has been an explosive growth in the mobile phones and generally small wireless handheld devices market. In the years to come, most people will use their handheld device to make wireless security-sensitive transactions like online banking, stock trading, and shopping. It is also anticipated that the wireless Internet will have much more communication partners than the wired. Estimations show that at the end of 2005 there will be 1.6 billion wireless users, and more than 1 billion of them will be 3G mobile Internet users [1]. All of them will wish to use their devices, with the optimum radio interface available to the surrounding environment, for example, home environment, car, office, shopping, and so forth (Figure 1).

On the one hand, the underlying network infrastructure will remain heterogeneous with different wired and wireless technologies. On the other hand, the protocol architecture of the access networks will be based on Internet protocols; hence, integration into a homogeneous infrastructure on the IP layer will be possible. In this environment, it is obvious that the protection of personal and business data is very important, while the ultimate goal is seamless interconnection of different wireless and fixed network technologies to provide efficient, flexible, and secure communication based on the Internet protocol suite.

Secure sockets layer (SSL) [2, 3, 4, 5] is the predominant and most widely used security protocol on the wired Internet. Almost all web servers support some version of SSL.

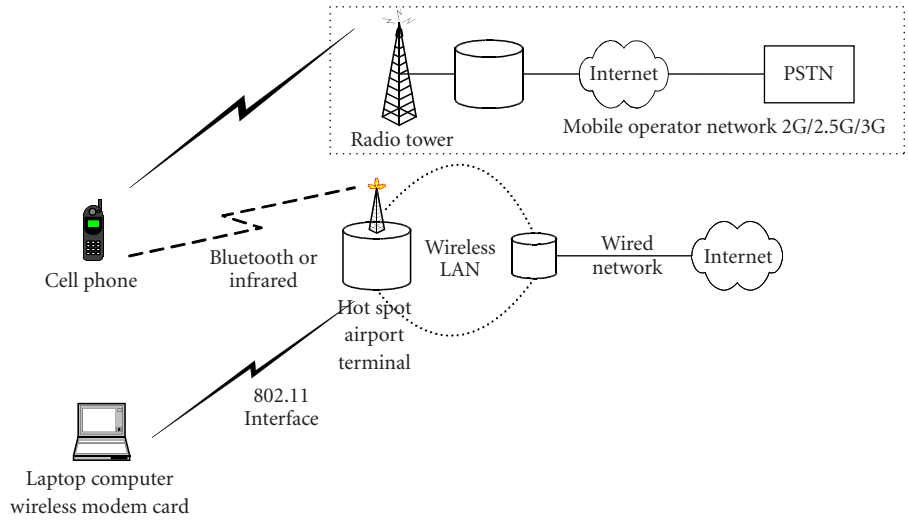

Figure 1: 4G environment.

Unlike wired networks, to the best of our knowledge, no wireless data service today offers this protocol on a mobile device. Performance considerations in using SSL in a resource-constrained environment drove wireless designers to choose for their mobile clients a gateway oriented security protocol called wireless transport layer security (WTLS) [6, 7]. Provided that it is possible to develop a usable, in terms of performance, implementation of SSL for a handheld device as suggested in [8], 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 a more secure, flexible, and reconfigurable authentication and key agreement (AKA) procedure for next generation mobile and wireless communication networks [9, 10, 11].

In this paper, we first discuss existing problems related to AKA procedures in 3G networks and then describe and evaluate the performance of a new AKA scenario, which is based on the SSL protocol and takes advantage of public key infrastructure (PKI). We show that SSL-based authentication can be possible in terms of service time in future mobile systems, while it can simultaneously provide both the necessary flexibility to network operators and a high level of confidence to end users.

Section 2 provides an overview of the SSL protocol and discusses how SSL and PKI can be included in future mobile communications. In Section 3, we propose and analyze an AKA mechanism based on SSL, while in Section 4 we evaluate the performance of this solution. The paper is finally concluded in Section 5.

2. INTRODUCING SSL AND PKI IN MOBILE COMMUNICATIONS

SSL establishes a transport-level secure channel for encrypted communications, between two parties. In fact, it is a new security protocol and layer between the application layer and the transport layer in the Internet protocol suite [12]. Hence, SSL needs a reliable transport protocol like TCP. 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.

SSL provides communications privacy through symmetric encryption and integrity through message authentication codes (MACs). Both the IETF's transport-layer security (TLS) and the WAP forum's WTLS protocol are direct descendants of SSL.

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 [8]. A mobile device that supports SSL protocol can be used for various applications, like HTTP, NNTP, and FTP. So it can actively secure electronic transactions via internet, including those which require the exchange of private information like passwords, PINs, and credit or prepaid card numbers, ensuring their secure transport through the network, and providing the subscriber with the essential level of confidence and certainty.

Furthermore, the necessity for more processing power and memory has driven smart cards toward more advanced architectures, all the way to where we are beginning to see 32-bit RISC-based ARM processors in smart cards. These cards based on modern chips have already appeared in the market, and they can effectively store and protect the subscriber's private key, generate good pseudorandom values and take over symmetric key (un)wrapping functions [8, 13]. The Schlumberger Cyberflex smart card, for example, can perform 1024 bit RSA operations (both public and private key) in less than one second. Mobile's device processor can efficiently carry out the rest of the calculations needed by SSL protocol. For example, the ASPeCT project has demonstrated that public-key authentication is possible and global system for mobile communication (GSM) and universal

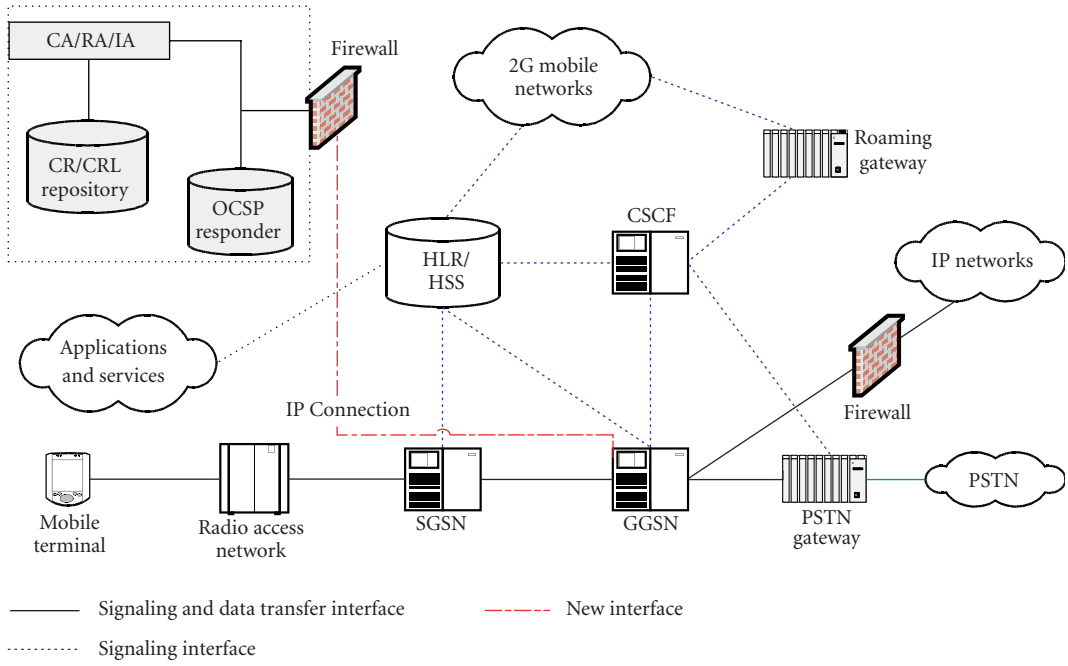

Figure 2: 3G network architecture and PKI elements.

mobile telecommunications system (UMTS) applications can coexist on a single smart card [14]. A recent study has also shown the feasibility of SSL in handheld wireless devices [8], while a relevant work showed that SSL handshake protocol time can be improved up to 5.7 times [15].

SSL supports different protocols for creating premaster keys (RSA, Diffie-Hellman, etc.), several different cryptographic algorithms, and different MAC algorithms. In the context of an AKA procedure, these properties can provide the appropriate flexibility in a continuously evolving environment, where the available means from a perspective of diversity and computational power at the attackers side are increasing rapidly. The incorporation of the international mobile subscriber identity (IMSI) transmission and the packet-temporary mobile subscriber identity (P-TMSI) allocation in SSL-based AKA far enhances its reliability and offers fewer opportunities to potential attackers.

Certainly, to implement an AKA mechanism based on SSL, we need to utilize some sort of PKI, which is not necessarily part of the beyond-3G (B3G) network core [16] (Figure 2). PKI is gradually being introduced in the market. Projects like ASPeCT [14] and USECA [17], Third Generation Partnership Project (3GPP) discussion documents [16, 18] especially for UMTS release 6, as well as other recent works [19] anticipate that evolution. The eNorge 2005 strategy [20] calls for a shared PKI for Norway, 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.

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 nonrepudiation services.

More specifically, in the context of an AKA implementation based on SSL, we assume the following.

(i) There is some sort of certification authority (CA), which issues and revokes certificates. This can be public in the form of a common certification service provider (CSP) or private; one or more per operator.

(ii) UMTS subscriber identity module (USIM) tamper-resistant card is storing the user's private and public keys, and certificate(s) along with the trusted CA certificates and IMSI. USIM card should be capable of generating pseudorandom values and executing public key operations. For instance, if RSA key exchange is being used in MS key-exchange message (see Section 3.2), the USIM generates a 48 bytes premaster secret and encrypts it under the server's RSA key found in server's certificate. This means that mobile equipment (ME) has to pass server's public RSA key to the USIM. Other operations required by the SSL protocol have to be supported by the ME. ME should also have a nonvolatile memory to store P-TMSI and SSL caching parameters in order to support session resumption discussed in Section 3.3.

(iii) Every network element, which takes part in AKA procedures, possesses a key pair similar to user's and the corresponding digital certificate. AKA functionality

can be provided by (serving GPRS support node) SGSNs, (session initiation protocol) SIP Servers or even Authentication Authorization and Accounting (AAA) servers. Moreover, these entities track and store possible CAs Cross reference certificates [13, 16] to support inter operator(s) authentication and trust.

(iv) There is one-at-least digital certificate repository, which stores all the digital certificates and is being managed by the mobile operator's CA.

3. AN AKA MECHANISM BASED ON SSL

3.1. Existing problems in 3G AKA

In UMTS, the AKA mechanism is somewhat similar to the authentication in GSM. The idea to use public keys in the process of authenticating the users was abandoned, mainly due to backwards compatibility and for performance considerations. The authentication in both systems is based on a symmetric secret key K, which is stored in the user's U(SIM) card and in the corresponding Authentication Centre (AuC) in which he made his subscription. The procedure, as described in [21], is based on the challenge/response protocol.

Several known weaknesses in AKA GSM seem to be now fixed in UMTS, through further study and investigation. However, there are still some "gaps" which an attacker can possibly exploit. Below we briefly describe these weaknesses, while for a detailed breakdown of 3G-AKA shortcomings refer to [21, 22, 23, 24].

(i) Passive or active attacks can compromise authentication vectors either from SGSN or home subscriber server (HSS) which store a number of vectors or quintuplets for each user, or from the SGSN-HSS communication link. The problem becomes more important in case the subscriber is roaming between two or more PLMNs, for example, in different countries, where the home network has always to send authentication vectors for use by the serving network. This means that the authentication vectors are very likely to become compromised or spoiled, traveling across different networks which possibly have dissimilar security features [22]. In any case, the user will still need to trust each serving network operator and its security technology in the same way he does in GSM networks.

(ii) In some cases, the system allows the identification of a user by means of the permanent subscriber identity (IMSI) in clear text. The procedure should be invoked by the serving network when (a) the subscriber registers for the first time in a serving network, or after a long time interval in which the mobile station (MS) has not been used and (b) when the network cannot retrieve IMSI from P-TMSI, due to SGSN's database malfunctions or in handover cases when the (IMSI, P-TMSI) pair is transmitted from one SGSN to another and the IP address of the old SGSN cannot be resolved. The procedure is open to passive attacks, where the intruder is waiting for potential IMSI transmissions in clear text or active man-in-the-middle attacks [22]. The unprotected transmission of permanent identity is primarily a threat against user identity confidentiality. But there is also a possibility that the user identity could be tampered with. However, it is not obvious how it could be exploited beyond creating general commotion in the system. Below, we shall examine a way in which the P-TMSI allocation procedure can be part of the SSL AKA mechanism.

(iii) The key sizes and the ciphering and deciphering algorithms are fixed. This makes the whole mechanism inflexible and less secure, whenever security vulnerabilities are discovered in an existing algorithm, like in the case of GSM A5/1 algorithm [25]. Having a dynamic security mechanism that is able to negotiate and load new encryption modules with different key lengths on demand allows greater flexibility.

(iv) One security enhancement in UMTS is the inclusion of integrity protection service. However, integrity is only guaranteed for signalling data between RNC and MS. The user data do not have an associated MAC, and are consequently vulnerable to manipulation.

(v) In addition to securing mobile communications, the security mechanisms in mobile devices should be able to provide security services for multimedia applications and IP-based services. The challenge is even greater when mobile communication involves multiple domains in wireless, self-configuring, and heterogeneous environments.

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 2.0, is using a WAP gateway, which is generally considered as insecure and certainly not "end-to-end." Moreover, it allows the use of weak encryption algorithms and features that make chosen-plaintext attacks and brute force attacks easier to mount. At any case, WTLS authentication procedure is often anonymous, and if it is done, then it is performed only once (towards the WAP gateway) for the whole time the same WAP gateway is used.

3.2. AKA based on SSL protocol

As we already discussed, the AKA mechanism is based on the symmetric key K, which is used by the (common to AuC and USIM) one-way functions f1 through f5 to compute and to verify the authentication vectors respectively. Functions f1 and f2 are message authentication functions while f3, f4, and f5 are key generating functions. For a detailed explanation of f1 to f5 functions, refer to [21, 24]. The USIM-network authentication process is mutual as an attempt to enhance GSM. Despite that, during the authentication process, there is always the chance for an intruder to use a compromised authentication vector that he has intercepted either from HSS or from the HSS-SGSN link. If that vector has not been used again, then the attacker can possibly achieve his goals, using a false base station (BS) (impersonating the network), or a false MS (impersonating the user).

An SSL-based AKA mechanism, which includes P-TMSI allocation, is described hereunder and is depicted in

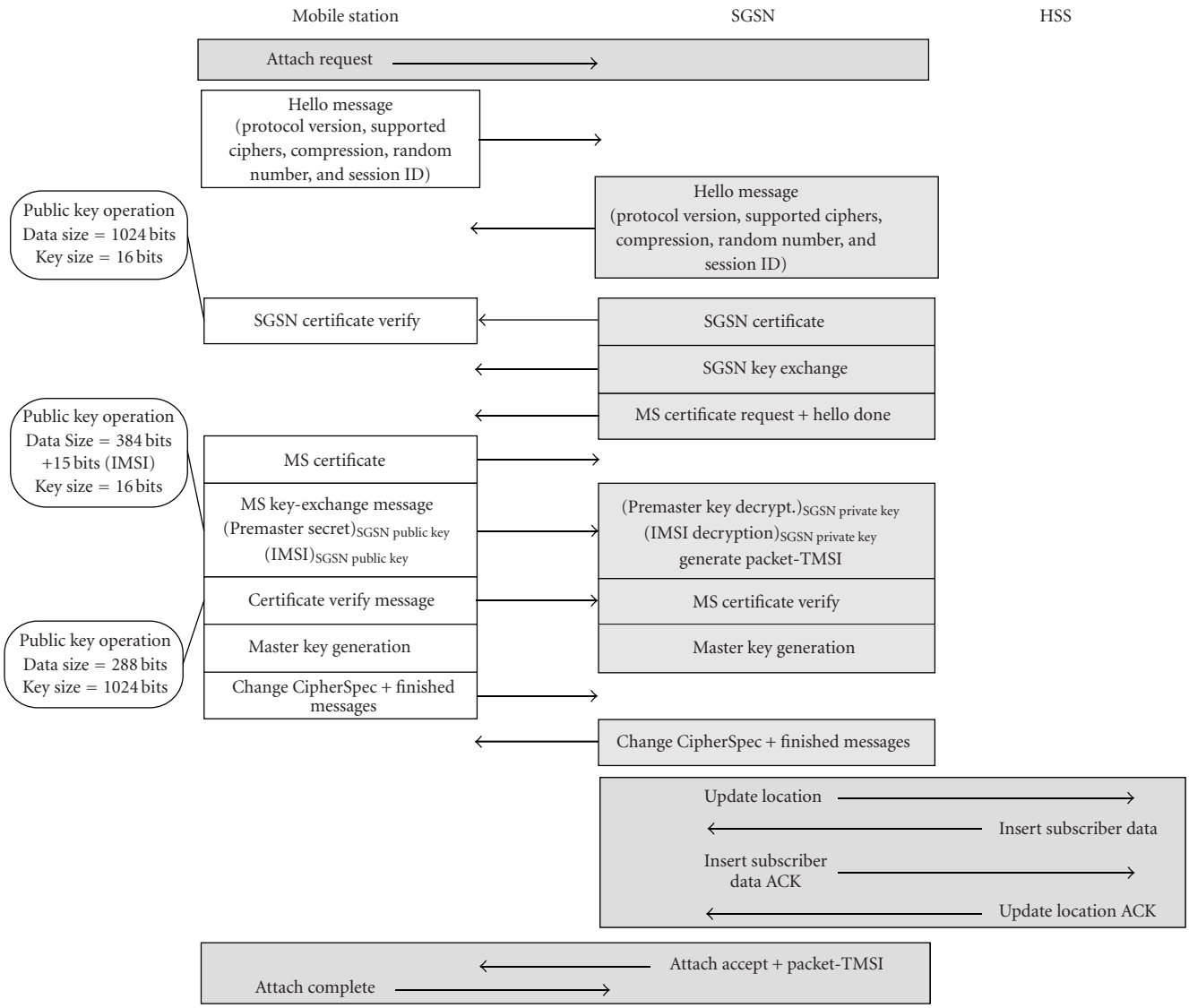

Figure 3: AKA mechanism based on SSL (P-TMSI is unknown to SGSN).

(Figure 3). The proposed mechanism can be implemented in some future time or in AKA dual-mode devices. 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, we will show how the procedure is extended when P-TMSI is included in the MS's attach request message to SGSN.

(i) MS initiates the SSL handshake sending an MS hello message to SGSN. This includes the MS security options: the highest version of the SSL protocol that the MS can support, an ordered preference list of cryptographic and compression parameters that the MS can sustain, a 32-bit timestamp, a 28-bit random number, and a session ID. This ID should be empty if no SSL session currently exists or if the client wishes to generate new security parameters.

(ii) SGSN replies with its hello message. If it supports some cryptographic and compression methods common to MS, those are included in its message. Otherwise, the connection is terminated. In addition, SGSN sends its 32-bit and 28-bit numbers and a session ID. If the latter is equal to the MS session ID, it is implied that the parties are going to use security parameters agreed on in a previous session. Otherwise, SGSN generates a fresh session ID number denoting a new connection.

(iii) SGSN sends its digital certificate to MS. This can be an X509v3 certificate or an X509v3 subset.

(iv) SGSN sends a key-exchange message to MS. This step is essential to give the MS a way to verify that the SGSN really possesses the private key corresponding to its public key certificate.

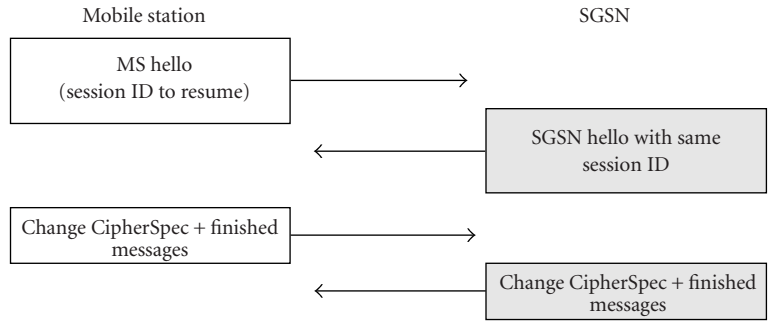

Figure 4: Resuming an SSL session (abbreviated handshake).

(v) MS has to validate the SGSN's certificate. First, it verifies that the certificate is published by a CA that it can trust and that the certificate time-validity period has not expired. Given that changes on those lists are rare, a CAs list with their corresponding public keys can be prestored in the USIM card. It is important that the procedure uses the public key from its internal store to verify the certificate, rather than the public key from the CA certificate, that might be provided by the SGSN. Finally, it computes the certificate's hash and compares it with that existing in the received certificate. We also assume that there is no need for the MS to check if SGSN's certificate has been revoked. This is true as SGSNs, or authentication servers in general, administered by the mobile service provider are relatively few and in the order of tenths according to the size of the network. Considering the provider's moderately closed network and assuming that the private key of each SGSN is adequately protected, the following options are possible. (a) Each operator can periodically, for example, 12–18 months, refresh SGSN's certificates by revoking the old ones. (b) Issued certificates for SGSN can have a limited life of 12–18 months. After that, they must be replaced with the new ones. (c) The last option is to replace an SGSN's certificate only when the machine operation is suspended or upgraded. This procedure will still happen quite infrequently.

(vi) SGSN requests MS certificate and concludes its part of the negotiation with a hello done message.

(vii) MS/USIM generates a premaster secret value and encrypts it using the SGSN's public key. Similarly, it encrypts its IMSI and sends both values back to SGSN. If a Diffie-Hellman key exchange is performed, the SGSN and the MS exchange their public parameters as part of the SGSN key exchange and MS key-exchange messages.

(viii) MS has to prove that it possesses the private key that corresponds to the public key, which is included in the certificate it previously sent to SGSN. The message contains a digitally signed hash from the information, available to both parties (keys and messages). This hash will be checked by SGSN.

(ix) SGSN checks the validity of the MS's certificate carrying all the above-mentioned tests.

(x) SGSN decrypts the premaster secret value and IMSI using its private key. It also generates P-TMSI.

(xi) Both parties convert the premaster secret into master secret. The converting procedures may include MD5 & SHA-1 hashes with the session IDs exchanged by both parties as input parameters. The master key will be used for ciphering and MAC calculations.

(xii) MS sends change CipherSpec and finished messages and SGSN replies accordingly. Note that the finished messages are cryptographically and integrity protected making use of the previously negotiated parameters. Finally, P-TMSI is included in the attach accept message send from SGSN to MS.

3.3. *Resuming a previous session*

To minimize the overhead of sophisticated cryptographic calculations and of a significant number of protocol messages in both parties, SSL defines a mechanism by which two parties can reuse previously negotiated SSL parameters. As Figure 4 shows, resuming earlier sessions notably streamlines the SSL AKA negotiation. The two hello messages define if the session can be resumed or not. More specifically, if MS wishes to resume a previous session, then it includes its session ID in the MS hello message suggesting its value to SGSN. If SGSN 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 fresh session ID value and then the full negotiation takes place [4, 12].

The entire attach procedure, incorporating our proposed SSL AKA procedure, is illustrated in Figure 5, in the form of a message sequence diagram. The MS retrieves its P-TMSI from its nonvolatile 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 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 presented in Figure 6.

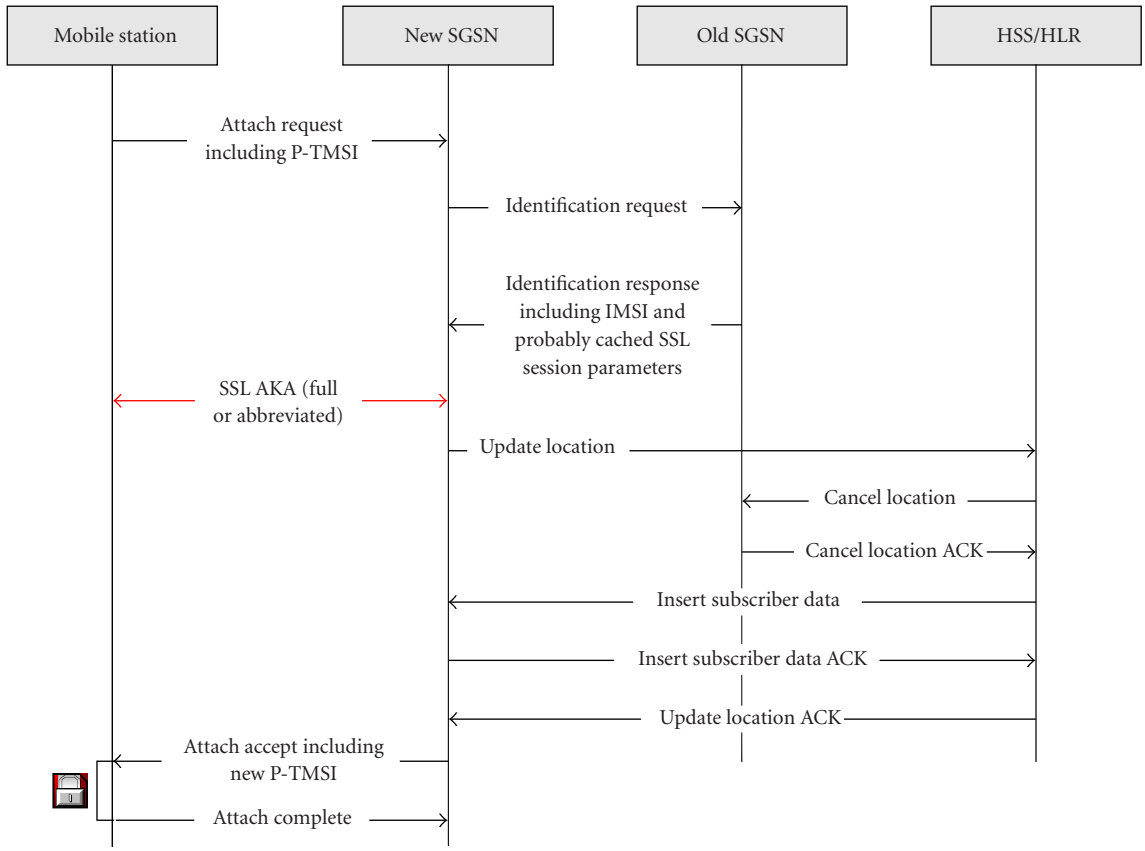

Figure 5: MS's attach procedure.

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. Potential attackers gain more data to analyze and more time to perform analysis. So the SGSN 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.

3.4. AKA procedure in a serving network

In case the subscriber is using his mobile device in a foreign network, SSL AKA procedure remains as is, with the assumption that the home network CA and the serving network CA, have pre-exchanged cross reference certificates. In that case, SGSN is bound to send to MS the corresponding cross-reference certificate too [16]. Of course, another option is to have a common root CA, which is acting as a CSP and is entrusted by both the serving and home network CAs.

3.5. Additional requirements

Some further issues need to be resolved before the proposed SSL-based authentication is introduced. Currently it is not possible to implement a straight SSL AKA procedure between an MS and 2.5G-SGSN, because there is not a direct IP connectivity between MS and SGSN. However, 3G-SGSN will eventually communicate towards all directions (RNC, GGSN) using IP as specified by 3GPP and thus it is very likely that it will be finally integrated with GGSN, at a later time. More importantly, regardless which node will provide AKA functionality, we believe that it is more efficient to implement the AKA procedure as a service, provided to the user, irrespective of the access network/domain he is connected to and the service he is requesting.

For instance, 3GPP AKA is used for authentication purposes at both the radio network and the IP multimedia subsystem (IM), introduced in UMTS release 5. Moreover, current 3GPP specifications for UMTS release 6 describe an interworking architecture between UMTS and WLAN 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. 3GPP seems to choose the extensible authentication protocol (EAP)-AKA protocol [26, 27, 28] to support such interworking scenarios.

In this B3G environment, we perceive authentication as a service performed at the higher protocol layers irrespective of the underlying network technology. According to B3G "all-IP" vision, this technology independent approach will probably be more suitable.

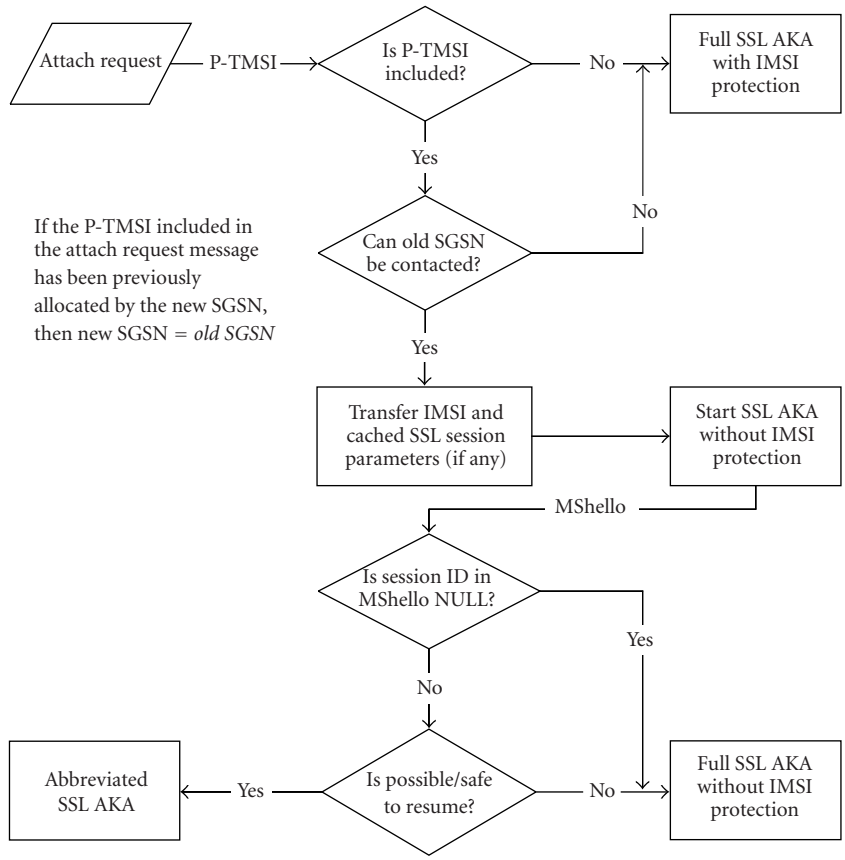

Figure 6: Checking whether a session can be resumed.

Another issue is that SSL AKA protects effectively user's data but leaves signalling communication between RNC and MS unprotected. In a similar way, as current UMTS specifications provide ciphering and integrity protection to signalling traffic, the ciphering and integrity keys constructed during SSL handshake could be used for the protection of only signalling messages at the lower layers. A mechanism similar to "the security mode command" message of the radio access network application protocol (RANAP) [21] is necessary to transfer these keys to RNC. Ciphering on signalling data could also be avoided as the critical issue is the integrity of this data. Additionally, the USIM card must support algorithms f8 and f9 for ciphering and integrity protection of signalling, respectively.

4. SSL AKA SERVICE TIME MEASUREMENTS

4.1. Testbed setup

In order to evaluate the performance of the SSL AKA mechanism, we constructed an experimental hardware and software architecture. The objective was to take measurements of the AKA procedure and at the same time simulate the consumption of processing power and communication resources of a real full deployment. We excluded error processing and optional features from our study, because these increase development time without consuming significant additional resources.

The development and test model topology is illustrated in Figure 7. The presumed mobile device is an IBM ThinkPad 380 laptop computer that uses Windows 95B operating system. The "client" uses a Siemens ME45 mobile phone, in order to connect to the Internet over GPRS. The IBM 380 incorporates a 150 MHz Pentium CPU and has 16 MB of RAM available. Contemporary wireless devices are featuring advanced architectures with Strong-Arm processors up to 400 MHz, memory capacities of 64 MB RAM and 48 MB ROM, support for java applications and strong operating systems. At the other end, the "server" machine has a Pentium III 733 MHz processor with 256 MB RAM, running the Windows 2000 Professional SP2 operating system. The server has also a WAN connection available. Comparable testbeds for GPRS and WAP performance evaluation can be found in the literature [29, 30, 31].

We wrote the applications in C++ and employed the well-known open-source Apache-style license OpenSSL toolkit in version 0.9.6g (http://www.openssl.org) [32] 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 from server's side only. Moreover, it is argued that Java performance in crypto code is by far worse than C performance [4].

The experimental software architecture is depicted in Figure 8. Two processes are running on the SSL enabled "authentication server": process A1, which opens a TCP-SSL

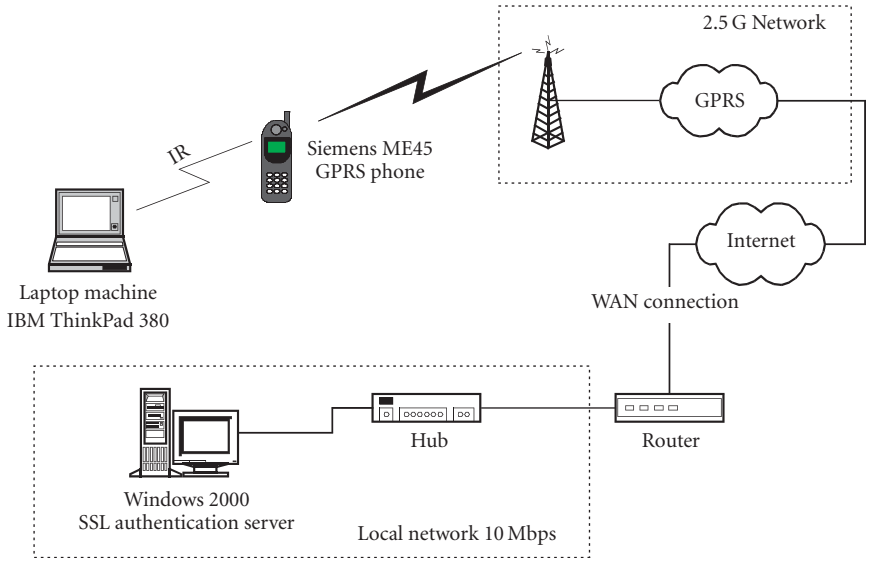

Figure 7: Hardware architecture.

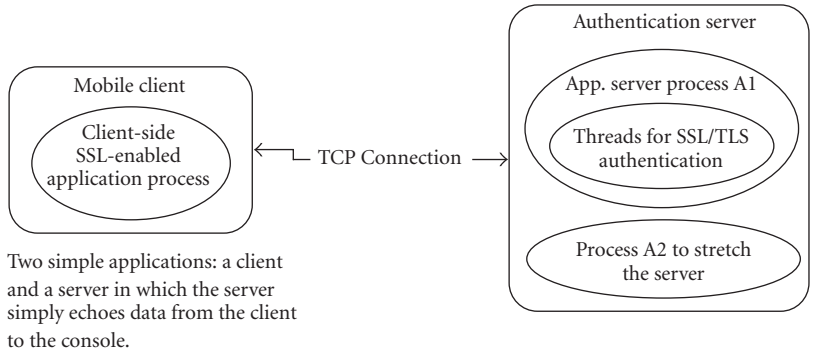

Figure 8: Software architecture.

listening socket and waits for transactions, and process A2, which locally generates a large number of SSL transactions and loads the local-host server. The interarrival times between successive SSL authentication requests, generated by process A2, follow the negative exponential distribution. Application server process A1 is multithreaded. When it receives a message, it dispatches a thread to process and respond to the request.

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, and verify certificates paths and certificates chain depths above four, and we enabled the client to support only SSL version 3 and TLS version 1. Moreover, we excluded from the client and the server, support for ciphers and MACs algorithms that are generally considered anonymous or weak, for example, 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 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 network authentication by stretching the client even more. As shown in Figure 9, our certificate "tree" includes a root CA, a server CA, and a client CA. Different service providers may operate client CA and server CA accordingly. Root CA certificate is self-signed. Similarly, root CA signs server CA and client CA certificates. Client holds a certificate signed by client CA, whereas server CA signs server's certificate.

We used a standard X.509 certificate format that contained the following custom fields: country name, state or province name, locality name, organization name, organization unit name, common name, email address. Additionally, we used two more optional (attribute) fields: optional company name and optional common name. Common name or subject field that serves as the client's distinguished name was filled up with the client's IP address. For the other fields, we provided demo values.

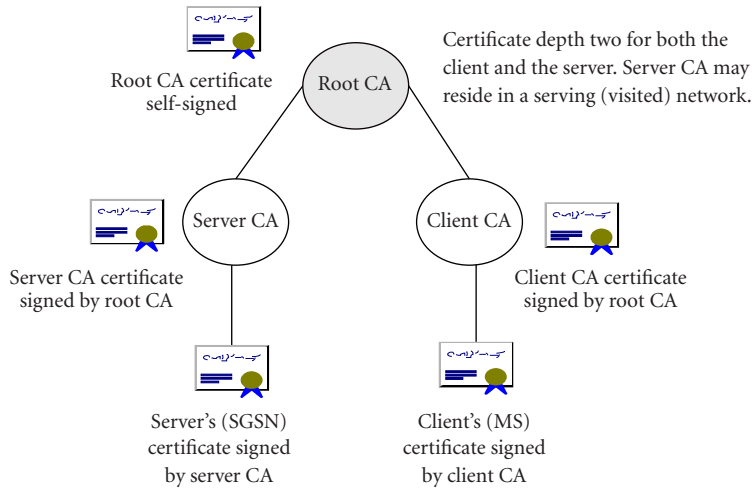

Figure 9: Certificate tree.

Both parties check certificates validity, against time expiration, issuing CA and IP address. An implementation must not only ensure that the certificate is valid, but must also make sure that it certifies the right party. Since a certificate binds a public key with a distinguished name, the "natural" option is to use a unique ID, like IMSI, in the common name field. Of course this will reveal the real identity of the user. In the following, we will present a way to avoid this.

It is well known that CSPs are capable and liable to issue "anonymous" X.509 certificates, which means that the common name field can contain any pseudonym. The mapping of pseudonyms to IMSIs is known only to that particular CSP and a potential eavesdropper cannot derive the real identity of the subscriber. Even if an attacker compromises both the subscriber certificate and TMSI, he still cannot be SSL AKA authenticated, as he will not be able to prove (to SGSN) that he holds the private key that corresponds to that certificate's public key. This is true because it is infeasible to create the right certificate verify message (Figure 3 and (viii) in Section 3.2).

No party checks certificates validity against any fresh revocation list. Only the authentication server is bound to do so by checking against MS's IMSI. This will require an IMSI revocation list transferred from HSS to SGSNs. Having that an IMSI has a relatively long life, this list can be transferred once a day during nonpeak hours. In the meantime, if IMSI has been revoked but is not included in the last list received, the handshake will successfully start. However, when SGSN sends an update location message to HSS, it will receive an invalid IMSI response, denoting that the specific IMSI has been revoked.

All RSA keys are 1024 bits in length, premaster secret exchange is based on ephemeral Diffie-Hellman key with RSA signatures, thus supporting forward secrecy and the resulting symmetric SSL session key is 128 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 for encryption, and HMAC-SHA-1 for integrity (EDH-RSA-DES-CBC3-SHA).

4.2. 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 times. The general packet radio service (GPRS) coding scheme was CS1 (9.05 Kbps) and the time slots for GPRS were varying from 3 to 4, thus having wireless network speeds in the range from 27 to 36 Kbps. 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 roundtrip 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 the NRT, plus the preparation time, for example, 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 includes 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 values in milliseconds in Table 1 and the probability density functions of these time durations in Figures 10 and 11.

In Figure 10, we can see the relation between NRT and RPT. We easily observe that RPT plot is actually a right shift of about 0.03 second, of the NRT plot, which means that the

Table 1: Average service times in milliseconds.

Mobile station				Authentication server
NRT	RPT	THT	TST	SCRT
803	833	6858	7684	6133

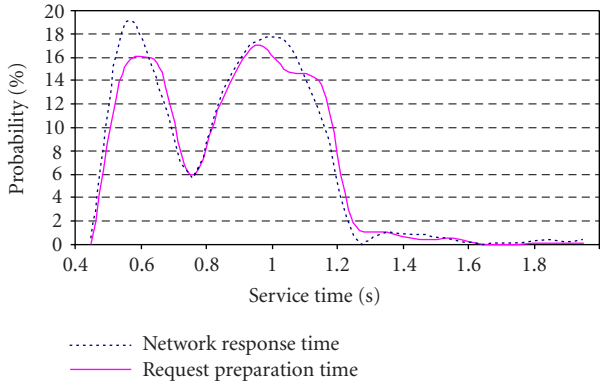

Figure 10: Service times 1/2.

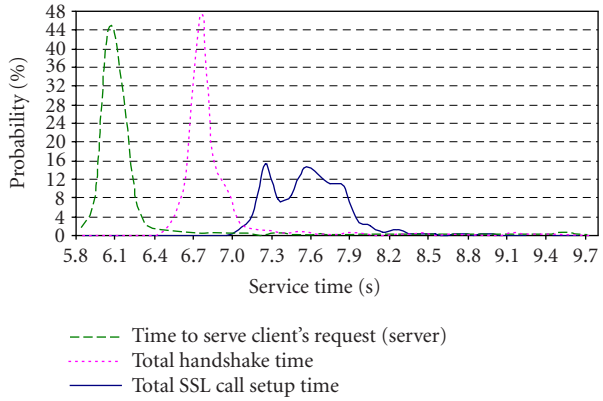

Figure 11: Service times 2/2.

preparation process takes nearly constant time to complete and the total time before SSL handshake is mainly dependent on the distribution of NRT, that is, the network speed.

Similarly, in Figure 11 we present the relation between TST, THT, and SCRT. We notice that the THT plot is a right shift of about 0.7 second of the SCRT plot. This 0.7 second is actually the MS's time to prepare, for example, the request and to load the certificates, plus half round trip time (0.4 second) from the initial MS hello. TST has nearly the same distribution, as the NRT and RPT of the previous figure. This can be explained from the protocol's 10 exchanged messages, in other words, approximately 5 NRTs. Therefore, 803 milliseconds average round trip time multiplied by 5 equals 4 second, plus other times for calculations, verifications, and hashing, we get the total average SSL setup time of 7.7 seconds. We note that all measurements gathered were highly immune to server's workload as this was generated by process A2.

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

4.3. Remarks on the measurements results

We believe that the average time of 7.7 seconds can be, under certain circumstances discussed hereunder, an accepted authentication time duration for the user of a wireless B3G device. Comparing our work with other works, we can mention the following. First, 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 showed a time of approximately 10 seconds [8]. Second, to the best of our knowledge, two other works describe and measure public key-based authentication mechanisms for 3G mobile devices. The first one [33] discusses an alternant to SESAME architecture called Tiny SESAME. SESAME extends Kerberos by providing additional services. Performance measurements using an HP Jornada 680 with a 133 MHz CPU running Windows CE 2.11 and a version of Tiny SESAME in PersonalJava, showed 10-to-16 second client authentication times depending on the call setup scheme. The other one [34] uses proxy assisted Kerberos protocol architecture between the client and the server. Measurements using a Vadem Clio C-1000 with a 100 MHz CPU running Windows CE and applications written in C++ showed 8-to-15 second client authentication times depending on the scenario architecture and network speed.

Regarding our results we can also mention the following.

(i) We have to take into account that the network speeds for 3G will be 144 Kbps up to 348 Kbps for wide and up to 2 Mb/s for low coverage and mobility, which will substantially reduce round trip times. A rough calculation assuming 144 Kbps network speed could diminish NRT to 200.8 ms showing an improvement of a factor of four.

(ii) A higher GPRS coding scheme, if offered by the operator and possible by the prevailing link conditions, could improve protocol's handshake performance considerably [31].

(iii) We have also 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 average extra time spend in each roundtrip was about 200 milliseconds.

(iv) Someone has to consider that the exchanged certificates provide for authentication in a serving network. In 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. A real 2.5G standard AKA mechanism, assuming that someone activates his device in a roaming network, takes about 5–7 seconds to complete.

Table 2: Memory recourses in kilobytes.

Item	Kilobytes
RAM space for process A1	2662
Disk or EPROM space for OpenSSL .dll files	1064
Disk or EPROM space for MS's application program	96.2
EPROM space in chipcard for client's certificate (.pem file)	2.93
EPROM space in chipcard for one trusted CA-root certificate (.pem file)	1.89

Excluding hardware improvements, further optimizations may come from either the protocol structure (messages exchange) or the network architecture. As we discussed in Section 3.3, session resumption option provides a dramatic performance improvement. That is because resumed sessions use the master secret from a previous connection, thus reducing the computation load and the number of messages. Testing with 512-bit RSA keys shows a performance improvement of a factor of 20 only in the handshake. An even larger improvement can be observed with larger keys [4]. More important, we would consequently expect corresponding improvement in network throughput, as a significant number of sessions are resumed. 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 mentioned in Section 2, the SSL's handshake protocol time can be improved up to 5.7 times [15].

Another important thing is that 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 and sizes of messages exchanged) to generate a number of messages and transmit them all at once using a buffer at the transmitting side [4]. Regardless the buffer size, it is also possible that the data may be quite large and should be segmented in more than one TCP segments.

Finally, two recent works [36, 37] on battery consumption of mobile devices with comparable testbeds and using the SSL protocol, show that energy utilization can be adequately controlled.

5. CONCLUSIONS AND FUTURE WORK

The AKA mechanism, as described by 3GPP, is based on GSM AKA and it was designed to correct known GSM's problems and weaknesses. As we already discussed, and until the providers improve their inter/intra-networks security, there are still some "weak points" which potential intruders can exploit. However, the greater drawback of the existing procedure is that it cannot offer a dynamic and flexible AKA mechanism, acting rather statically, and facing the problem shortsightedly. On the contrary, 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 Internet environment and heterogeneous wireless and wired technologies. SSL AKA mechanism can be used to overcome these inefficiencies providing real end-to-end security and importing the benefits of PKI in future mobile communication systems.

In this paper, we perceive authentication as a service and propose an SSL-based authentication scenario for future mobile communication systems that takes advantage of PKI and protects user's IMSI. We examined and evaluated its performance focusing on handshake protocol. We showed that SSL-based authentication can be possible, in terms of service time, while it can simultaneously 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, cross-reference certificates to support internetwork authentication, and battery consumption.

ACKNOWLEDGMENTS

We would like to thank Mr. Dimitrios Geneiatakis, Mr. George Karopoulos, Mr. Anastassios Katsigiannis, and Mr. George Sarkos for providing us with the network measurements, and the reviewers for their valuable comments that helped us improve the quality and presentation of our work.

REFERENCES

[1] M. Frodigh, S. Parkvall, C. Roobol, P. Johansson, and P. Larsson, "Future-generation wireless networks," *IEEE Personal Communications Magazine*, vol. 8, no. 5, pp. 10–17, 2001.

[2] A. Frier, P. Karlton, and P. Kocher, *The SSL 3.0 Protocol Version 3.0*, http://home.netscape.com/eng/ssl3.

[3] T. Dierks and C. Allen, *The TLS Protocol: Version 1.0*, IETF RFC 2246, January 1999.

[4] E. Rescorla, *SSL and TLS Designing and Building Secure Systems*, Addison-Wesley, Boston, Mass, USA, 2001.

[5] R. Oppliger, *Security Technologies for the World Wide Web*, Artech House, Boston, Mass, USA, 2000.

[6] WAP forum WAP-217-WPKI, *Wireless Application Protocol Public Key Infrastructure Definition*, www.wapforum.org/what/technical.htm.

[7] R. Khare, "W* effect considered harmful," *IEEE Internet Computing*, vol. 3, no. 4, pp. 89–92, 1999.

[8] V. Gupta and S. Gupta, "Experiments in wireless Internet security," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC 2002)*, vol. 2, pp. 860–864, Orlando, Fla, USA, 2002.

[9] G. Kambourakis, A. Rouskas, and S. Gritzalis, "Using SSL/TLS in authentication and key agreement procedures of future

mobile networks," in *Proc. of 4th International Workshop on Mobile and Wireless Communications Network (MWCN 2002)*, pp. 152–156, Stockholm, Sweden, 2002.

[10] S. Dixit and R. Prasad, Eds., *Wireless IP and Building the Mobile Internet*, Artech House, Norwood, Mass, USA, 2003.

[11] D. Wisely, P. Eardley, and L. Burness, *IP for 3G: Networking Technologies for Mobile Communications*, John Wiley & Sons, New York, NY, USA, 2002.

[12] S. Thomas, *SSL and TLS Essentials: Securing the Web*, John Wiley & Sons, New York, NY, USA, 2000.

[13] A. Nash, B. Duane, D. Brink, and C. Joseph, *PKI: Implementing and Managing E-Security*, RSA Press, Berkeley, Calif, USA, 2001.

[14] ASPeCT Project, *Securing the future of mobile communications*, 1999, http://www.esat.kuleuven.ac.be/cosic/aspect.

[15] N. R. Potlapally, S. Ravi, A. Raghunathan, and G. Lakshminarayana, "Optimizing public-key encryption for wireless clients," in *Proc. IEEE International Conference on Communications (ICC 2002)*, vol. 2, pp. 1050–1056, New York, NY, USA, April 2002.

[16] 3GPP TSG, "Using PKI to provide network domain security," Discussion Document S3- 010622 SA WG3 Security – S3# 15bis, November 2000.

[17] USECA Project, "UMTS security architecture: Intermediate report on a PKI architecture for UMTS," Public Report, July 1999.

[18] 3GPP TSG, "Architecture proposal to support subscriber certificates," Discussion and Approval document, Tdoc S2-022854, October 2002.

[19] G. Kambourakis, A. Rouskas, and S. Gritzalis, "Introducing PKI to enhance security in future mobile networks," in *Proc. of the IFIPSEC'2003 18th IFIP International Information Security Conference*, P. Samarati and S. K. Katsikas, Eds., pp. 109–120, Kluwer Academic, Athens, Greece, May 2003.

[20] eNorge 2005, *Naerings – og handelsdepartmentet*, 2002.

[21] 3GPP Technical Specification, *Security Architecture*, TS 33.102 v.5.1.0, December 2002.

[22] 3GPP Technical Specification, *A guide to 3rd Generation Security*, TR 33.900 v.1.2.0, January 2000.

[23] T. Aamodt, T. Friiso, G. Koien, and O. Eilertsen, *Security in UMTS - Integrity*, Telenor R&D, Norway, February 2001.

[24] V. Niemi and K. Nyberg, *UMTS Security*, John Wiley & Sons, New York, NY, USA, January 2004.

[25] A. Biryukov and A. Shamir, *Real-time Cryptanalysis of the alleged A5/1 on a PC*, Preliminary draft, December 1999.

[26] 3GPP Technical Specification, 3GPP system to WLAN interworking, TS 24.234 v.0.2.0 Release 6, November 2003.

[27] 3GPP Technical Specification, WLAN interworking security, TS 33.cde v0.1.0, July 2002.

[28] J. Arkko and H. Haverinen, "EAP-AKA authentication," draft-arkko-pppext-eap-aka-11.txt, October 2003.

[29] R. Chakravorty and I. Pratt, "Performance issues with general packet radio service (GPRS)," *Journal of Communications and Networks (JCN)*, vol. 4, no. 2, pp. 266–281, 2002.

[30] R. Chakravorty, J. Cartwright, and I. Pratt, "Practical experience with TCP over GPRS," in *Proc. IEEE Global Communications Conference (IEEE GLOBECOM 2002)*, vol. 2, pp. 1678–1682, Taipei, Taiwan, November 2002.

[31] J. Korhonen, O. Aalto, A. Gurtov, and H. Lamanen, "Measured performance of GSM HSCSD and GPRS," in *Proc. IEEE International Conference on Communications (ICC 2001)*, vol. 5, pp. 1330–1334, Helsinki, June 2001.

[32] J. Viega, M. Messier, and P. Chandra, *Network Security with OpenSSL*, O'Reilly & Associates, 2002.

[33] J. Al-Muhtadi, D. Mickunas, and R. Campbell, "A lightweight reconfigurable security mechanism for 3G/4G mobile devices," *IEEE Wireless Communications*, vol. 9, no. 2, pp. 60–65, 2002.

[34] A. Harbitter and D. Menasce, "The performance of public key-enabled kerberos authentication in mobile computing applications," in *Proc. 8th ACM Conference on Computer and Communications Security (CCS-8 2001)*, pp. 78–85, Philadelphia, Pa, USA, November 2001.

[35] G. Apostolopoulos, V. Peris, P. Pradhan, and D. Saha, "Securing electronic commerce: reducing the SSL overhead," *IEEE Network*, vol. 14, no. 4, pp. 8–16, 2000.

[36] R. Karri and P. Mishra, "Minimizing energy consumption of secure wireless session with QoS constraints," in *Proc. IEEE International Conference on Communications (ICC 2002)*, vol. 4, pp. 2053–2057, New York, NY, USA, 2002.

[37] R. Nachiketh, R. Srivaths, A. Raghunatan, and J. Niraj, "Analysing the energy consumption of security protocols," in *Proc. ACM ISLPED 2003 Conference*, pp. 30–35, Seoul, Korea, 2003.

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 Ph.D. 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 has been 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.S. in communications and signal processing from Imperial College, London, and the Ph.D. 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 B.S. in physics, an M.S. in electronic automation, and a Ph.D. 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 thirty 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 fifty-five journal 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. 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.