The Use of Trusted Third Parties and Secure Billing in UMTS

L. Chen (***), H.-J. Hitz (*), G. Horn(*), K. Howker (***), V. Kessler(*),
L. Knudsen (**), C.J. Mitchell (***), C. Radu(**)

(*) = Siemens AG, Corporate Research and Development, Munich, Germany.
{hans-joachim.hitz, guenther.horn, volker.kessler}@zfe.siemens.de
(**) = Katholieke Universiteit Leuven, ESAT-COSIC, Leuven, Belgium.
{Lars.Knudsen, Cristian.Radu}@esat.kuleuven.ac.be
(***) = ISG, Royal Holloway, University of London, Egham, England.
{liqun,jkh,cjm}@dcs.rhbnc.ac.uk

Abstract

This paper presents solutions for certain advanced security features in UMTS developed in the ACTS project ASPeCT, namely Trusted Third Party services for certification and key escrow and a new scheme for the billing for WorldWideWeb based Value Added Services using micropayments. The solutions will be validated in demonstrators whose architectures are described.

1. Introduction

It is generally accepted now that adequate security features must form an integral part of a mobile telecommunications system. In second generation systems such as GSM and DECT, security features based on cryptographic techniques have been included in a systematic way for the first time. Their success is undeniable: second generation systems are much less susceptible to fraud than their predecessors. However, the increasing, and increasingly diverse, demand for security by users, operators and regulatory bodies calls for more advanced security features in third generation systems, such as the Universal Mobile Telecommunications System (UMTS). It is the goal of the ACTS project AC095 ASPeCT to specify such advanced features and propose solutions.

Some of these advanced security features to be provided in UMTS will be made possible by the use of more powerful smart card technology not yet available for second generation systems. This technology - together with the use of suitably adapted security mechanisms - will make the use of so-called public-key techniques in mobile systems possible for the first time. Trusted Third Parties (TTPs) will provide the infrastructure for the use of these techniques. The services which TTPs provide - among others certification of public keys, support of key management for end-to-end security services, and key escrow services - are described in more detail in section 2 of this paper. TTPs also enable the provision of non-repudiation services based on digital signatures, opening the possibility of secure billing services over UMTS. We concentrate on a new scheme to bill the user for value added services which we expect to become increasingly important. Corresponding details are presented in section 3 of this paper.

2. Trusted Third Parties and Key Escrow

The role of TTPs to support security services is recognised in a wide variety of application domains, and with a wide variety of cryptographic techniques. Within the ASPeCT project we will primarily be concerned with the use of TTPs in providing mobile telecommunications services using public key cryptography (at least for key management). More specifically we will be concerned with the provision of secure billing services and end-to-end security services.

To support secure billing services the TTP will act as a certification authority for the public keys of a mobile user and a value-added service provider (VASP). The mobile user and the VASP will each have its own TTP which may generate and distribute *public keys* for them.

To provide end-to-end encryption the TTP will act as a *secret key* distribution centre with a key escrow facility to satisfy lawful interception requirements. From associated TTPs, clients can obtain appropriate secret keys and interception agencies will be able to obtain escrowed keys when presenting a warrant.

2.1 Public key infrastructure

One well established role for TTPs in supporting public key cryptographic techniques is in the generation of public key certificates. Within ASPeCT we will be implementing this TTP function, and in demonstrating this functionality we hope to deal with some of the complex issues surrounding the verification of user identities, and the validation of user keys. The public key certificates generated using this TTP will be of fundamental importance in supporting the billing security functions.

ASPeCT TTPs will be responsible for the following tasks to support certification of public keys: generation of certificates, maintenance of *Directory Information Bases* (DIBs), which are used to store certificates, management of *Directory Information Trees* (DITs), which are used to issue a particular certificate to a particular entity via a suitable *certification path*, revocation of invalid certificates, and generation and maintenance of *Certificate Revocation Lists* (CRLs). We are currently concerned with the generation of certificates, including client certificates, TTP certificates and cross certificates in the first TTP demonstration of ASPeCT.

A non-standard, compact certificate format was chosen as both storage space on a smart card and bandwidth on the air interface are strictly limited. Each certificate for a public key consists of two parts: a certificate type identifier and a signed certificate information sequence. We have two types of certificates in ASPeCT, depending on the signature mechanisms used. The first type makes use of RSA-signature based on ISO/IEC 9796-2 [1], where the signed certificate information sequence includes a signed recoverable string and a non-recoverable part. The second type makes use of AMV-signature based on ISO/IEC 14888-3 [2], where the signed certificate information sequence is the certificate information sequence itself together with an appendix - the signature of the sequence.

The certificate information sequence includes a basic certificate information and a set of extended attributes providing other optional information about both the subject and the issuer. Table 1 shows a certificate information sequence format for ASPeCT.

Table 1. A certificate information sequence format

Field	Contents	Description
1	Map field	This field gives the map which fields and options will be presented in the certificate.
2	Version	The version number of the certificate.
3	Serial Number	Unique number of the certificate, assigned by the issuer.
4	Public key identifier	Optional. Unique identifier of the public key to be used to verify the signature on this certificate.
5	Issuer identifier	Two options for the issuer identifier: hashed or plain.
6	Validity	Including certificate valid period and optional private key usage period.
7	Subject identifier	Two options for the subject identifier: hashed or plain.
8	Subject key usage	Optional. The usage of the subject key being certified.
9	Cross certificate attributes	Optional. Two situations: either forward or backward cross certificate.
10	Certificate path attributes	Optional. The number of related certificates and a list of subject identifiers included in the certificate path.
11	Subject public key information	An algorithm identifier plus a public key value for the subject. Three examples of algorithms are foreseen: RSA, elliptic curve and Diffie-Hellman over GF(p).

2.2 End to end confidentiality

Another important role for TTPs in future mobile networks will be in supporting the particularly sensitive issue of end-to-end confidentiality. For a variety of reasons it may be necessary for copies of keys used to provide such a service to be held 'in escrow', so that they can be provided to authorised entities wishing to read encrypted messages at some future time. The use of TTPs to support such a service is widely seen as the way forward.

Thus, in parallel with the use of the TTPs to provide public key certificates, we will be implementing the key escrow functionality within the ASPeCT TTP infrastructure. This will, amongst other things, require the design and specification of an escrow interface, as well as the specification and design of a public key based key management system to support key escrow.

The following protocol, as one example of a key escrow mechanism, will be used in the initial TTP demonstration. It is based on the JMW proposal described in [3]. In this protocol every client has an associated TTP. If two clients, communicating with each other securely by using end-to-end encryption, are located in different domains, then the relevant pair of TTPs (one in each domain) collaboratively perform the dual role of providing the users with key management services and providing the two interception agencies with warranted access to the users' communications. A session key for end-to-end encryption is established based on the Diffie-Hellman algorithm for key exchange [4]. An asymmetric key agreement pair for one client (the receiver) is separately computed by both TTPs using a combination of a secret key shared between them and the receiver's name, and another asymmetric key agreement pair for the other user (the sender) is generated optionally either by himself or by his own TTP. The receiver computes the session key by combining his private key (transferred securely from his TTP) with the sender's public key (sent with the encrypted message). The sender computes the same session key by combining his private key with the receiver's public key (obtained from the sender's own TTP). Interception agencies in each domain can retrieve the session key from the TTP in the same domain.

2.3 Architecture of the TTP server

ASPeCT TTPs are designed and specified for implementation in PCs. These will be used in trials and demonstrations of secure billing services and end-to-end encryption services in UMTS.

The architecture of the TTP server has four layers, namely *external communication layer*, *TTP security control layer*, *TTP function and operation layer* and *cryptographic function layer*. Between the first and second layers, there is a *TTP access interface*; between the second and third layers, there is a *TTP security service interface*; and between the third and fourth layers, there is a *Cryptographic interface.*

3. Secure Billing

3.1 Charging for value added services

A UMTS user will be offered a much larger variety of services than in today's networks. But there will still be a distinction between basic tele- and bearer- services, such as traditional telephony, video telephony or high speed data services, and services offering added value to the user, such as the provision of a particular piece of information the user needs. Our work on secure billing in UMTS concentrates on a new scheme to bill the user for such *value added services* (VAS).

It is expected that the number and variety of VASs will greatly increase while current networks are evolving towards UMTS. One reason for this is that users will increasingly possess terminals with much larger processing and display capabilities than today's mainly speech orientated terminals. *Personal mobile communicators* will integrate the functions of a mobile phone and of a laptop or palmtop PC. These devices may be used to access information of a much more complex nature than that available to users of VASs in mobile systems today: Instead of being restricted to the character oriented display of his handy, the user will be able to display e.g. hypertext-documents with included graphics, so, instead of numbers giving the prices of stock he may view charts of stock indices,

instead of textual information on the nearest hotels he may view a street map of his surroundings indicating the location of the hotels, or he may view a coverage map of the mobile operator in whose domain he is roaming.

The charging for today's VASs consists of a basic charge for the basic service and a premium for the added value. Both are based on the duration of the call. In the future, due to the greater variety of services offered more *flexible charging schemes* for the premium would be desirable. Flexibility relates to the parameters which determine the charge (in addition to the duration of the call, the charge may depend e.g. on the amount of data transferred, different tariffs may apply for different information items), to the variety of different possible tariffs and to the ease with which a certain tariff can be changed.

Also, the value of a particular piece of information retrieved by a user from a VAS provider at one time may be quite small so that the charging scheme would not warrant a large financial overhead to process the charge. In addition, the scheme has to have a performance compatible with the requirements of a mobile system. In short, the *charging scheme* must be also *efficient.*

It is expected that the evolution of current mobile systems towards UMTS will also see the emergence of many new network operators, UMTS service providers and VAS providers which may have serious implications for the trust relations among them. The *charging scheme* must be *secure* against cheating, and the parties involved should have the assurance that justified claims relating to charges can be proved and that unjustified claims cannot be successfully made. This is called *incontestable charging.*

ASPeCT will demonstrate a proposed charging scheme for VASs in UMTS which satisfies the above requirements. The charging scheme is a credit-based payment scheme using *micropayments* according to Pedersen [5]. A similar scheme was recently published by Rivest and Shamir [6]. In the demonstrator, the user - acting as a Web client - will be able to retrieve WorldWideWeb pages from a VAS provider - acting as a Web server - over a mobile link.

We assume in our model that the user has a subscription with a UMTS service provider. Then, the only on-line communication required in the charging procedure is that between the user and the VAS provider while the service is being provided. The VAS provider will forward the information proving his claims on the user to the user's UMTS service provider off-line who in turn will bill the user, also off-line. The UMTS service provider will also take care of the payments to the network operators involved in providing the needed connectivity.

A crucial element in our model is the User Identity Module which is a smart card held by the user and issued by his UMTS service provider. This *smart card* will be *multi-functional,* and will contain the security procedures to access basic UMTS services as well as advanced payment features. A particular feature of our solution is that the authentication protocol used for basic service access may be re-used in our charging scheme for VASs.

In section 3.2 below, the protocols of our proposed charging scheme are presented. In section 3.3, the architecture of our demonstrator is described.

3.2 Security protocols

The charging consists of two phases: In the *initialization phase* (see Figure 1) , the user (denoted by U in the protocol description) and the VAS provider (denoted by V) authenticate each other and agree on a session key, and the user commits himself to a starting value for the micropayment scheme and a certain tariff by performing a digital signature on corresponding data. The authentication protocol is identical with one submitted to ETSI SMG for UMTS user-to-network authentication [7]. The starting value is the n-th iterate of a one-way function applied to a random value chosen by the user, as described below. In the *data transfer phase*, the user pays by releasing the pre-images of the starting value, so-called "ticks" which represent unit charges. The value of one unit charge is agreed upon in the initialization phase. The "ticks" serve as proof to the VAS provider that the user incurred certain charges because only the user could have generated them. They are presented by the VAS

provider to the user's UMTS service provider to clear the charges. The particular efficiency of the scheme stems from the fact that the user may commit himself to a large number of payments of unit charges with only one signature. Images of one-way functions are much less expensive to compute and to transmit than signatures.

The two protocols corresponding to the two phases are presented in the following. For the sake of brevity, we omit a third protocol, the so-called "re-initialization protocol" which is used when the user runs out of "ticks" while the call is still in progress.

For the description of the protocols, a few bits of mathematical notation are needed. The protocols run in the framework of a finite group G with generator g, e. g. the multiplicative group of a finite field or a subgroup of an elliptic curve, in which the Discrete Logarithm Problem is hard. Both U and V can efficiently perform operations and randomly select elements in G. There is a function f which maps G onto a set of bit strings, a length-preserving one-way function $F: \{0,1\}^n \rightarrow \{0,1\}^n$, a one-way function $h2$, and hash functions $h1$ and $h3$, which are considered common knowledge in the system. U and V possess a symmetric encryption function, where $\{M\}_K$ denotes the encryption of message M with key K. V has secret and public key agreement keys v and g^v respectively. U possesses an asymmetric signature scheme with secret key KU^-, with public key KU^+, and with the secret signature transformation Sig_U. Authentic copies of g^v and of KU^+ are available to U and V respectively or may be sent by V to U (U to V respectively), accompanied by a certificate issued by a Trusted Third Party (see section 2 above). The identity of V is assumed to be known to U at the start of the protocols.

The *authentication and initialization of charge ticks protocol* (see Figure 1) works as follows: U generates a random number u and then computes a temporary key agreement key g^u which he sends to V. On receipt of the first message, V does not know with whom he is communicating. V generates a random number r, computes $(g^u)^v$ and a sort of Diffie-Hellman session-key with additional freshness assurance for both sides $K := h1(f((g^u)^v)||r) = h1(f((g^v)^u)||r)$. He confirms possession of the derived key K by sending a hash value $h2(K, r)$. V also sends to U some charge data ch_data, e.g., the tariff on which the charging is to be based, and a time stamp tV. U has to check if the time-stamp and the tariff information received is acceptable to him. In the third message, U commits himself to the charge data, the time-stamp and a starting value for the tick payment scheme by including this data in the signature in the third message. Let T be a system-wide fixed parameter, specifying the number of ticks that can be paid by U with a single signature. U generates a random seed α_0 and computes $\alpha_i = F(\alpha_{i-1})$ for i=1,...,T, i.e. $\alpha_i = F^i(\alpha_0)$. He includes α_T in the third message as shown in Figure 1. α_T need not be kept confidential. U confirms knowledge of K by signing a hash value which must be different from $h2(K, r)$. In the third message U also encrypts his identity IdU to ensure anonymity. We assume the use of an El Gamal type signature system. In the general case, the signature in the third message has to be encrypted as well to ensure anonymity.

The main feature of the *charge ticks* protocol is that within a short time a large number of small charges (ticks) must be confirmed by U. On one hand, both U and V do not want to carry out a complete charging procedure for each of the ticks, due to the waste of computation resources and the short time between ticks. On the other hand, V does not want to compute the corresponding bill

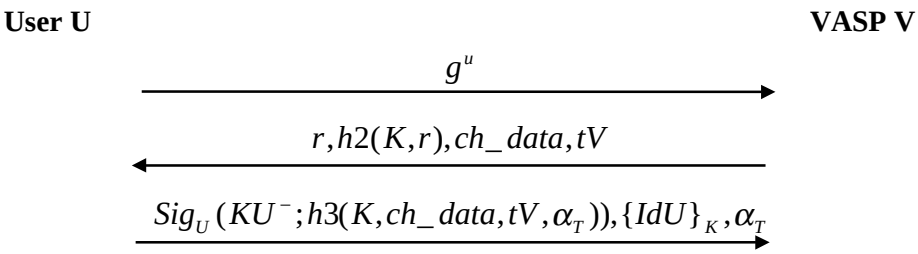

Figure 1 Authentication and initialization of charge ticks protocol

covering the cumulative amount *after* the completion of the call session, because U could maliciously interrupt the protocol previous to providing V with a signature on the cumulative amount. The solution for the charge ticks protocol is as follows. Assume that the charge ticks protocol is

ongoing, and that U has been asked by V to pay for tick t. U confirms for tick *t* by releasing the pre-image $F^{T-t}(\alpha_0)$. Note that several ticks maybe released at once. When the call session has ended, or the maximum number T of ticks per signature has been reached, V stores the last received pre-image α and the number of ticks *tck_cnt* consumed by U during the current run of the charge ticks protocol. V composes and stores the transcript of the charge ticks transaction, representing the cheque to be claimed.

3.3 Architecture of the demonstrator

The demonstrator will be built in *two stages.* The first version will be based on PCs representing the user and the VAS provider. The user's smart card will be attached via a card reader to the PC. Connectivity will be provided initially by wired links and by GSM data connections. The second version of the demonstrator will be trialled in a UMTS testbed provided by ACTS project EXODUS and will feature on-line Trusted Third Parties.

Regarding the *protocol architecture,* it is particularly worth emphasizing that our concept permits the use of existing applications (WorldWideWeb client and server) as well as of existing communication stacks (TCP/IP). Most applications useful in our context are based on a standardized interface, namely sockets, or, in a Windows environment, more specifically Windows sockets. There are three protocol layers (cf. Figure 2 below): An application layer, a communication layer and a security layer in between, realizing the security functionality as described in section 3.2. The security layer provides Windows sockets to the application layer. There is no need to modify the application and there is no need for an extended security interface. The security layer uses Windows Sockets to access the communication stack. In this way, the security layer is independent from particular applications and is transparent for the application. This is seen as a major advantage of our approach.

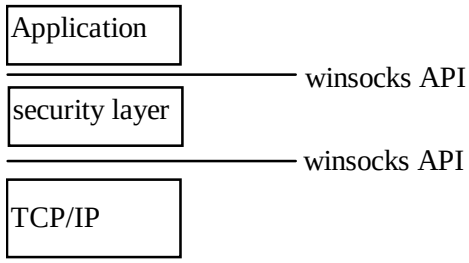

Figure 2 - protocol stack

References

[1] ISO/IEC CD 9796-2 (review). Information technology - Security techniques - Digital signature schemes giving message recovery - Part 2: Mechanisms using a hash - function. June 1996.

[2] ISO/IEC 2nd CD 14888-3. Information technology - Security techniques - Digital signature with appendix - Part 3: Certificate - based mechanisms. June 1996.

[3] N. Jefferies, C. Mitchell and M. Walker. A proposed architecture for trusted third party services. In E. Dawson and J. Golic, editors, Lecture Notes in Computer Science 1029, Cryptography: Policy and Algorithms Conference, pages 98-104, Springer-Verlag, 1996.

[4] W. Diffie and M.E. Hellman. New directions in cryptography. IEEE Transactions on Information Theory, 22:644-654, November 1976.

[5] T.P. Pedersen. Electronic payments of small amounts. DAIMI PB-495, Computer Science Department, Aarhus University, August 1995.

[6] R.L.Rivest, A. Shamir. PayWord and MicroMint: Two simple micropayment schemes. May 1996, available from the authors under {rivest, shamir}@theory.lcs.mit.edu

[7] ETSI SMG SG DOC 73/95. A public-key based protocol for UMTS providing mutual authentication and key agreement.