Authentication and Payment in Future Mobile Systems

Günther Horn[1] Bart Preneel[2⋆]

[1] Siemens AG, Corporate Technology, D-81730 München, Germany,
(guenther.horn@mchp.siemens.de)
[2] Katholieke Universiteit Leuven, Dept. Electrical Engineering-ESAT,
Kardinaal Mercierlaan 94, B–3001 Heverlee, Belgium
(bart.preneel@esat.kuleuven.ac.be)

Abstract. This article presents an efficient public-key protocol for mutual authentication and key exchange designed for third generation mobile communications systems. The paper also demonstrates how a micropayment scheme can be integrated into the authentication protocol; this payment protocol allows for the provision of incontestable charging. The problem of establishing authenticated public keys through cross-certification is addressed.

1 Introduction

1.1 The future of mobile systems

Mobile communications is one of the fastest growing sectors of the IT industry. For example, in Europe the number of mobile users was 22 million in 1995 and is estimated to reach more than 110 million by the year 2000. While current second generation systems such as GSM (cellular, [ETS1]) and DECT (cordless, [ETS2]) will continue to play an important role, a new third generation system, the UMTS (Universal Mobile Telecommunications System) is shortly to be introduced in Europe, with commercial UMTS services expected to commence by 2002 [UMTS,ETS3,ITU2]. UMTS will provide a wider spectrum of services than today's systems, ranging from simple voice telephony to high speed, high quality multimedia services, regardless of physical location of the user, using radio frequency access to a convergent network of fixed, cellular and satellite components.

1.2 Mobile systems security

As for second generation systems, the most fundamental security requirement for UMTS is to ensure that the level of security is at least as high as that in existing wired telecommunications networks. The necessary security features to meet

⋆ F.W.O. postdoctoral researcher, sponsored by the Fund for Scientific Research, Flanders (Belgium).

this requirement include: confidentiality on the air interface (which is much more vulnerable to eavesdropping than a wired interface), anonymity of the user and, most importantly, authentication of the user to the network in order to prevent fraudulent use of the system. While these features are already provided in existing second generation systems, they need to be further developed and enhanced by the incorporation of additional features in UMTS. The most important of the new security features in UMTS is that the user must also authenticate the network in order to prevent an intruder from masquerading as a network operator or service provider. This requirement is motivated by the observation that a user may want to make sure that he is connected to the network of an operator whom he trusts. This becomes increasingly important as the number and variety of competing public and private network operators and service providers grows larger. The resulting potential network complexity also suggests that new techniques of managing the cryptographic keys necessary for the provision of these security features may be required. This is a very natural setting for the application of public key cryptographic techniques. Although applied successfully in other areas, public key cryptography has not previously been used in mobile communication environments due to performance constraints. It was not deemed suitable for second generation systems because of the resulting length of messages and the necessary computational loads. To overcome these problems a new protocol was developed for authentication between user and network; it was particularly designed to fit the performance constraints of mobile networks. The protocol is described in Sect. 2 below. Its design exploits the advances in two fields: crypto-controller smart cards (which have a co-processor which efficiently supports public-key cryptographic mechanisms) and elliptic-curve cryptosystems (which permit the use of smaller cryptographic parameters). The new protocol was successfully implemented and tested in the collaborative research project "ASPeCT" (Advanced Security for Personal Communications Technologies) which is funded by the European Union under the "ACTS" programme. An extended version of the protocol to include on-line TTPs is presented in Sect. 4. The choice of cryptographic algorithms and certificate formats is discussed in Sect. 5. Another crucial issue in the security of mobile systems is that users may want to protect themselves against incorrect bills. Therefore it may be necessary to provide undeniable evidence that claims related to user charges are correct. The ASPeCT project demonstrated the feasibility of a solution to the problem of securely billing for the provision of a value added service using the aforementioned authentication protocol and a suitable micropayment scheme. The payment protocol and its integration with the authentication protocol are described in Sect. 3 below.

2 Authentication and Initialisation of Payment Protocol

In the following, U denotes the user, represented by his User Identity Module, and V denotes the UMTS value added service provider, or, in short, VASP. The protocol presented in this section, however, is also proposed for use between a

user and a mobile network operator during call set-up. The design principles for the protocol can be found in Sect. 2.2.

2.1 Protocol goals

The goals to be achieved by the end of a successful protocol run are:

1. mutual explicit authentication of U and V;
2. agreement between U and V on a secret session key K with mutual implicit key authentication;
3. mutual key confirmation between U and V;
4. mutual assurance of key freshness (mutual key control);
5. non-repudiation of origin by U for relevant data sent from U to V;
6. confidentiality of relevant data sent by U to V.

The non-repudiation feature is motivated by the requirement for incontestable charging.

2.2 Principles for the selection of security mechanisms

One principle in the design of the protocol was to shift as much computational effort as possible from the user side to the network side because it is assumed that the user will be represented by a smart card which has limited computational capabilities. Another principle was to allow for messages that are as short as possible. A way to arrive at shorter messages is the use of elliptic-curve cryptosystems [Mene]. While the protocol does in no way mandate the use of these cryptosystems, it is designed in such a way that their advantages can be best exploited. Another way to arrive at shorter messages is the use of a streamlined certificate format which provides certificates much shorter than X.509 certificates. (For more details see Sect. 4.2.) The choice of the security mechanisms was guided by the following considerations: non-repudiation of data sent by the user requires a digital signature system on the user side. This signature system is then also used for authentication of the user for efficiency reasons. For session key establishment, a key agreement scheme (similar to the ElGamal scheme [ElGa]) with implicit key authentication [ISO3] of the network was chosen because then entity authentication of the network can be obtained with little extra cost. (See discussion in Sec. 2.5 below.) The protocols were chosen in such a way that their description is independent of the choice of the signature systems used by the user and the certification authority, respectively. They are also independent of the choice of the finite group in which the exponentiations required in the key agreement scheme are computed.

2.3 Prerequisites

Cryptographic functions

We assume that the following cryptographic functions can be executed by any participant:

- A symmetric encryption function where $\{M\}_K$ denotes the encryption of message M with key K. We assume that the cryptographic algorithm is resistant against known cryptanalytic attacks such as code-book attacks and chosen plaintext attacks.
- A (pseudo-) random number generator.
- Functions $h1$, $h2$ and $h3$ which are specified below.
- Multiplications in 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.

Further prerequisites

- The identity idV of V is assumed to be known to U at the start of the protocol.
- V has long-term secret and public key agreement keys v and g^v respectively, where g is as above.
- U possesses an asymmetric signature system with secret signature transformation Sig_U. In case of a signature with appendix, $\mathrm{Sig}_U()$ denotes only the appendix.
- There is a valid certificate $certU$ (issued by a certification authority CAU with identity $idCAU$) on the public key of the asymmetric signature system of U, available at U.
- There is a valid certificate $certV$ (issued by a certification authority CAV with identity $idCAV$) on the public key agreement key g^v of V, available at V.
- U possesses the public key necessary to verify certificates issued by CAV.
- V possesses the public key necessary to verify certificates issued by CAU.

Functions $h1$, $h2$, $h3$

Here, we explicitly list the requirements on the functions $h1$, $h2$ and $h3$. The following definitions are useful here: definitions 1, 2 and 3 are well-known [MvOV, p. 323ff.], while definitions 4 and 5 are weak forms of the MAC-property and of pseudo-randomness which we believe are sufficient in our context. Concatenation is indicated by $\|$.

1. A function h is *preimage resistant (one-way)*, if for essentially all outputs $y = h(x)$ it is computationally infeasible to find any input x' with $y = h(x')$ (x' may or may not be equal to x).
2. A function h is *partial-preimage resistant (local one-way)*, if for essentially all outputs, if part of the input is known it is still hard to find the remainder, i.e. it is not easier than brute-force.
3. A function h is *collision resistant (strong collision resistant)*, if it is computationally infeasible to find two inputs $x' \neq x$ which are mapped to the same value $y = h(x) = h(x')$.

4. A function h is *weakly computation resistant* (weak MAC-property) if it is computationally infeasible to find a pair $(x, h(K\|x))$ without knowing K, provided that no other pair $(x', h(K\|x'))$ is known (here the value of x can be chosen by the opponent).
5. A function h is a *weakly pseudo-random function* if, for secret random key K not used before and for known random x, the output $h(K\|x)$ is indistinguishable from a random output.

NOTE ON THE DEFINITIONS. There are some dependencies between these properties, e.g., a partial-preimage resistant function is especially preimage resistant. Note also that our definition 4 is indeed much weaker than the usual MAC-property (cf. [MvOV, p. 325]) because we assume for the weak MAC-property that an attacker does not even have one valid example *(message, MAC)* available against which he could test attempted forgeries. Definition 4 is not strictly weaker than definition 5 since definition 4 allows the opponent to choose the value of x. Definition 5 can be made operational in the following way: h is a publicly known function. Given parties Alice and Eve, Alice chooses random parameters K_i, x_i, R_i $(i = 1, 2, \ldots)$ and computes $H_i := h(K_i\|x_i)$. Alice makes x_i, R_i and H_i known to Eve and keeps K_i secret. Eve tries to break the function h by guessing which of the two parameters H_i and R_i is computed from K_i and x_i by applying h. If Eve has only about a 50% chance to make the correct guess then h fulfils definition 5. Note that this does not imply that such a function h (in practice a hash function) can be used to define a pseudo-random function because the opponent has no control over the input x_i and can only observe a single input-output pair (x_i, H_i) for each choice of K_i. Nothing is implied by definition 5 about repeated applications of h with the same value for K_i.

We require that the functions $h1$, $h2$ and $h3$ are hash functions (i.e. they map inputs of arbitrary finite lengths to fixed length outputs) which are easy to compute and that

1. $h1$ is a partial-preimage resistant, weakly computation resistant and weakly pseudo-random function.
2. $h2$ is a partial-preimage resistant, weakly computation resistant function.
3. $h3$ is collision resistant.

There are a number of practical cryptographic functions which are assumed to be at least collision resistant, e.g., SHA-1 [FIPS,ISO2], RIPEMD-160 [DoBP,ISO2] and ISO/IEC 10118-2 [ISO2]. It is common practice in cryptographic applications to assume that these hash functions are also preimage and partial-preimage resistant, though this issue needs further investigation. Due to [MvOV, p. 331] hash functions should not be used as pseudo-random functions unless the randomness requirements are "clearly understood". This does not contradict our assumption that a practical hash function fulfils our definition 5, (see corresponding note following definition 5). Let h be one of these well-known hash functions. Then we claim that the functions

1. $h1(x) := h(x)$

2. $h2(x) := \text{trunc}(n, h(x))$ where $trunc(n, y)$ returns the n least significant bits of y (n being significantly shorter than the length of the hash value, but long enough to prevent successful guessing).
3. $h3(x) := h(x)$

satisfy our requirements.

2.4 Protocol description

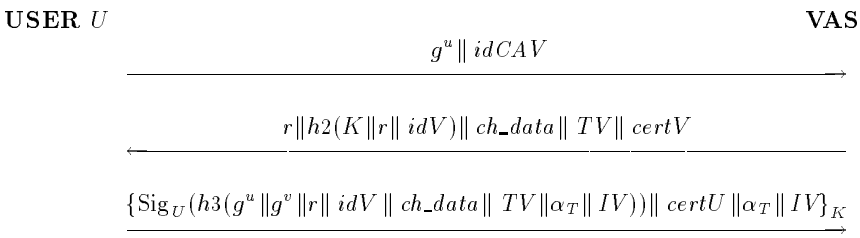

Fig. 1. The authentication protocol

At the start of the protocol (see Fig. 1), U generates a random number u and then computes g^u which he sends to V, together with the identity $idCAV$ (see remark on certificate verification keys in Sect. 2.5).

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 then a session key $K := h1((g^u)^v \| r)$. He demonstrates knowledge of the session key K by computing the hash value $h2(K \| r \| idV)$ which he sends to U, together with r, his certificate $certV$ and additional data needed as input to the payment scheme (a time-stamp TV and charging-relevant data ch_data).

On receipt of the second message, U computes the key $K = h1((g^v)^u \| r)$. He then checks the hash value $h2(K \| r \| idV)$ and he thus knows that V actually has the session key K. U generates random numbers IV and α_0, computes $\alpha_T = F_{IV}^T(\alpha_0)$ and signs the hash value of the concatenation $g^u \| g^v \| r \| idV \| ch\_data \| TV \| \alpha_T \| IV$. Here, F_{IV}, T, α_0, α_T and IV have significance only for the payment scheme, not for the protocol goals stated above, and are therefore explained in Sect. 3 below. U concatenates the signed hash with his certificate $certU$ and with α_T and IV. U then encrypts the concatenated parameters with K. On receipt of the third message, V first deciphers the message elements using K. He retrieves the certificate $certU$, and after the verification of the certificate, he can verify U's signature. V stores the signature and the corresponding message for later use in the payment scheme (see Sect. 3 below).

2.5 Discussion

General

Key confirmation and authentication of V: the inclusion of $h2(K\|r\|\ idV)$ in the second message gives key confirmation from V to U and hence, together with key freshness (cf. remark 3 below) and implicit key authentication of V, also entity authentication of V (see, for example, [RuvO]). Note that explicit key confirmation demands that the session key is used in the key distribution itself[1].

Key confirmation and authentication of U: the encryption of the certificate $certU$ with K in the third message provides key confirmation. The inclusion of $g^u\|g^v\|r$ in the signed part of the third message provides implicit key authentication from U to V as it confirms the origin of g^u and links this value to g^v and r. The inclusion in the signed part of the third message of the random number r generated by V provides entity authentication of U.

Replay of old keys and key freshness: the key K is constructed using the random number r generated by V in order to prevent an old key K being forced on V by U. This, together with the use of the random number u generated by U, guarantees key freshness to both sides.

Non-repudiation: U's signature provides non-repudiation of the signed data. Note also that in order to achieve non-repudiation it has to be ensured by other means that the certificate $certU$ of U is valid. Furthermore, in order to achieve non-repudiation, it may be required in addition that V submits $\mathrm{Sig}_U(\ldots)$ to a trusted time-keeper who signs $\mathrm{Sig}_U(\ldots)$ together with a timestamp and returns it to V. Otherwise, U could repudiate a signature claiming that the signature was generated by an impostor after the certificate had been revoked. Whether this additional measure should be implemented, however, depends on the security policy and on a trade-off between a higher security level and additional effort.

Confidentiality of the data in the third message: it is ensured by encrypting the data with the symmetric encryption function using K.

Encryption of the signature: the signature is encrypted for two reasons. Firstly, in order to guarantee that the signer knows the session key K. Secondly, in order to protect the user's identity; if the signature would not be encrypted, an attacker would be able to detect the identity of U by verifying the signature. This might be possible in a scenario where the attacker has access to the public keys of the users and he assumes that the originator of the signature is one of a small group of users (here 'small' depends on the time needed to verify a signature).

Inclusion of idV in the second message: this prevents so-called source-substitution attacks (as described in [MvOV, remark 12.54]). (These can also be

[1] Some protocols do not use the session key in the key distribution protocol itself in order to avoid leaking information on the session key. Such leaking of information on the session key is minimized in our protocol by the inclusion of random numbers in the encrypted messages and by the assumptions on the symmetric encryption function.

avoided by making sure that the TTP checks that the user is in possession of the corresponding private key before it issues a certificate on the public key. The inclusion of idV is simpler and prevents the need to detail the duties of a TTP in this context.)

Inclusion of idV in the third message: idV must be included in order to indicate the intended recipient of the signature. This is related to the use of the signature in the payment scheme: the signature, together with payment tokens sent in the payment protocol (see below), serves as a proof of payment. So, anyone who intercepts the signature and the tokens and presents them to the broker of the payment scheme could collect money fraudulently.

Certificate verification keys: since U has limited space for storing public keys, U will not be able to verify certificates issued by an arbitrary certification authority. Therefore, in the first message, U tells V the identity $idCAV$ of a certification authority CAV (certification authorities $CAV1$, $CAV2$, ...) whose certificates he is able to verify. We assume that V has got a certificate issued by this authority (one of these authorities). If this is not the case then an extended version of the protocol has to be run to obtain such a certificate (see Sect. 4.1).

Identification of multiple users: when the protocol is run concurrently by many users it is necessary to identify which message belongs to which user. This problem is assumed to be taken care of by the underlying communication system and, hence, need not be addressed by the security protocol. Temporary channel identifiers can be used to provide user anonymity.

Possible attacks and the choice of the properties of h1, h2 and h3

The weak pseudo-randomness property makes $h1$ suitable for key derivation in our context. Note that only one session key K is derived from the master key g^{uv}. Furthermore, the proposed protocol avoids the following attacks: an attacker Eve may try to forge the second message in three ways by attacking the function $h1$ or the function $h2$ or the concatenated function H (see 3. below):

1. Eve may try to find a valid pair $(r, K) = (r, h1(g^{uv}\|r))$. If she could do that she could compute $h2(K\|r\|\, idV)$ from this. This is impossible by the weak MAC-property of $h1$ because she does not know g^{uv}.
2. Eve may try to find a valid pair $(r', h2(K\|r'\|\, idV))$, possibly after having seen the valid pair $(r, h2(K\|r\|\, idV))$ generated by V. But this would not help Eve because when U tries to verify the forged second message $(r', h2(K\|r'\|\, idV))$, U computes $K' = h1(g^{uv}\|r')$ which by the weak pseudo-randomness of $h1$ is almost certainly different from K for $r \neq r'$. U's verification would be successful only if $h2(K'\|r'\|\, idV) = h2(K\|r'\|\, idV)$, but then Eve would have generated a valid pair $(r', h2(K'\|r'\|\, idV))$ without knowing K' and without having seen another valid pair involving K', in contradiction to the weak MAC-property of $h2$.
3. Eve may attack the concatenated function $H(L, r) := h2(h1(L\|r)\|r\|\, idV)$ directly and may try to find a valid pair $(r', H(g^{uv}, r'))$, possibly after having

seen the valid pair $(r, H(g^{uv}, r))$, generated by V. However, Eve would then also have generated a valid pair $(r', h2(K'\|r'\| idV))$ where $K' = h1(g^{uv}\|r')$, and $K' \neq K$ almost certainly by the weak pseudo-randomness of $h1$, hence this contradicts the weak MAC-property of $h2$.

3 The Payment Protocol

3.1 General

There is a recognised need for so-called micropayment systems, that are suitable for the efficient payment of small, frequently recurring, variable amounts. Applications range from electronic publishing to metering, telecommunications and information services and video-on-demand. A series of payments should be made to the same vendor over a period of time so that the vendor can aggregate the individual payments and spread the cost for clearing them with the broker over a larger number of payments. The micropayment system presented here is based on Pedersen's tick payment protocol. For a detailed discussion, the reader is referred to [Pede], however we summarise its main features. The novelty is not the payment protocol itself, but the way in which it is integrated with the authentication protocol proposed for the mobile system UMTS (cf. Sect. 1) and the payment scenario for basic and value added services in UMTS. The cryptographic mechanism employed in the tick payment protocol is based on Lamport's password scheme [Lamp]. Later Pedersen, Rivest and Shamir [RiSh] and Anderson et al. [AnMS] independently proposed the same mechanism for the payment of small amounts. Hauser et al. [HaSW], who were aware of Pedersen's work, proposed to integrate the mechanism with IBM's iKP (Internet Keyed Payment Systems) protocol for credit-card based electronic transactions. An extension of the approach can be found in [JuYu]. The proposed system has two phases: an initialisation phase in which the payer (in our scenario the user of a value added UMTS service) commits to initial values of the payment scheme by a digital signature, and an actual payment phase in which payments are made to the payee (in our scenario a UMTS value added service provider or VASP) by successively releasing preimages of an initial value α_T under a one-way function F. Initialisation is performed as part of the authentication protocol described in the previous section. The meaning of the parameters signed in the third message of that protocol and the detailed working of the tick payment protocol performed in the payment phase are described in the following.

3.2 Goals

From the payer's point of view

1. a payment in his name can be made only by him;
2. the amount of the payment is exactly what the payer has specified;
3. only the payee specified by the payer can receive the payment.

From the payee's point of view:

4. a payee can verify the correctness of a payment;
5. the payer cannot deny having made a verified payment;
6. the payee can be certain of being credited for verified payments by the broker.

From the broker's point of view:

7. the broker can verify the correctness of a payment.

3.3 Prerequisites

For the **tick payment protocol**, we have the following prerequisites:

- There is a public system parameter T which gives the maximum number of ticks (the currency unit of the payment system, reminiscent of phone ticks) to which the user can commit himself by one signature.
- There is a public family F of length-preserving one-way functions $F_{IV} : \{0,1\}^n \to \{0,1\}^n$, where n is a public system parameter and IV is an initialisation vector. (To be more precise, the functions F_{IV} need to be one-way on T-th iterates, cf. [Pede].)

Both F and T have to be chosen with care in order to avoid certain attacks (see discussion in Sect. 3.5).

3.4 Tick payment protocol description

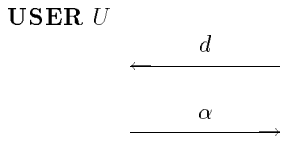

Fig. 2. The tick payment protocol

In the "authentication and initialisation of payments" protocol, the user commits to the parameters idV, ch_data, TV, α_T and IV. Parameter idV is the identity of the payee (the VASP V), ch_data gives the conditions under which the payment is made (such as applicable tariff etc.), TV is a time-stamp generated by V, giving date and time of day, IV is an initialisation vector by which the user selects a particular function F_{IV} from the family F of one-way functions, α_0 is a random number selected by the user and $\alpha_T = F_{IV}^T(\alpha_0)$ is the initial value for the tick payments. All payments made are multiples of one "tick". Whenever V requests a payment from U, he sends a message to U with the requested number d of unit payments (ticks) to be made. U responds by releasing preimages of $\alpha_T = F_{IV}^T(\alpha_0)$ under the function F_{IV}. For the payment of the first $d1$ ticks,

the user sends the payment token $\alpha_{T-d1} = F_{IV}^{T-d1}(\alpha_0)$. On receipt, V checks if $F_{IV}^{d1}(\alpha_{T-d1}) = \alpha_T$. For the payment of the next $d2$ ticks, the user sends the payment token $\alpha_{T-d1-d2} = F_{IV}^{T-d1-d2}(\alpha_0)$, V checks if $F_{IV}^{d2}(\alpha_{T-d1-d2}) = \alpha_{T-d1}$, and so on. Only the last value received from the user has to be stored by V, together with the signature in the third message of the "authentication and initialisation of payments" protocol. If the total number of requested ticks exceeds the maximum amount T to which the user can commit himself by one signature, then either the "authentication and initialisation of payments" protocol has to be run again or, preferably, a simplified re-initialisation protocol is run which has only two messages and does not repeat mutual authentication. This re-initialisation protocol is not presented in this contribution for lack of space. V can aggregate the payments from a single user, and clear the aggregated payments with the broker of the payment system, by presenting the signature by which the user committed to the initial values of the payment process and the last tick payment α_{T-d} made by the user. The VASP V is then credited d ticks by the broker.

3.5 Discussion

Achievement of goals

1. A payment in the user's name can be made only by him because he commits to the starting value α_T by his signature, and only he can know the preimages of α_T under the one-way function F_{IV}, provided the one-way function is appropriately selected. For the same reason, the payer cannot deny having made a verified payment.
2. The amount of the payment is exactly what the payer has specified: again, this depends on the appropriate choice of the one-way function, see the discussion in [Pede].
3. The identity idV of the payee is included in the user's signature, therefore only the payee specified by the payer can receive the payment.
4. The payee verifies the correctness of a payment by verifying the signature and verifying that the d-th iterate of the one-way function F_{IV} applied to the payment token received last equals the payment token received before (the starting value α_T respectively), where d is the amount due.
5. The payee can be certain of being credited for verified payments by the broker because payee and broker, as well as any arbiter, can verify payments (provided of course that an agreement exists that the broker honours verifiable payments). The broker can verify the correctness of a payment in the same way as the payee.

Further issues

1. Re-use of starting value α_T: a payer could use the same starting value α_T in two different sessions, either with the same payee or with a different payee.

This is to be strictly avoided by the payer as it would be to his disadvantage: assume that he pays for a total of $d1$ ticks in the first session and a total of $d2$ ticks in the second. The payees could then for both instances claim $\max(d1, d2)$ from the broker (provided they could intercept the communication), and the broker would have no way to check.

2. Re-use of initialisation vector: the purpose of using a whole family of one-way functions instead of just one function is to make it much harder for a fraudulent payee to invert the one-way function by brute force with the help of pre-computations. Therefore, a user should avoid re-using IV.
3. Check for double submission: note that the signed commitment contains idV and r, hence the commitment is different for each session and payee. The broker checks for double submission of the same payment by comparing idV and the nonce r in the signed commitment with previous commitments. If there is a match the broker refuses to honour the payment. No checking for double spending is required of the payee V. (Note that the checking by the broker could also be done using TV instead of r, provided that the granularity of TV is fine enough.)
4. Collision resistance is not required for the function F_{IV}: it does not help a fraudulent payee if he is able to generate collisions for F_{IV} because the starting value α_T is given to him by the payer.
5. The maximum amount T cannot be arbitrarily large. Rather, F and T must be chosen such that the probability that ${F_{IV}}^T(\alpha_0)$ lies on a cycle is small for all IV. If ${F_{IV}}^j(\alpha_0)$ lies on a cycle with length c and $j + c < T$, then payments for $T-j$ and $T-j-c$ ticks cannot be distinguished as ${F_{IV}}^j(\alpha_0) = {F_{IV}}^{j+c}(\alpha_0)$. Bounds for T as well as other results on the tick payment scheme will be presented in a forthcoming paper [KnPr].

4 TTP Related Issues

The protocol in Sect. 2 assumes that the user and the VASP possess the public keys necessary to verify each other's certificates, or that they possess an authenticated copy of each other's public keys. However, if this is not the case, the parties might have to contact an on-line TTP during the protocol in order to obtain a cross-certificate. Moreover, contacting an on-line TTP also allows for checking whether the certificate of the other party has been revoked. First the protocol with on-line TTP is described, and next the different options for cross-certificates are sketched. Due to space constraints, the protocol is only briefly described.

4.1 Protocol with on-line TTP

The protocol in Fig. 3 differs from the protocol in Sect. 2 in the following aspects: U also sends to V the identity $idTTP$ of his TTP, together with his own identity idU encrypted with the key $L := g^{uw}$ where g^w is the public key agreement key of the TTP. On receipt of the first message, V contacts the TTP of the

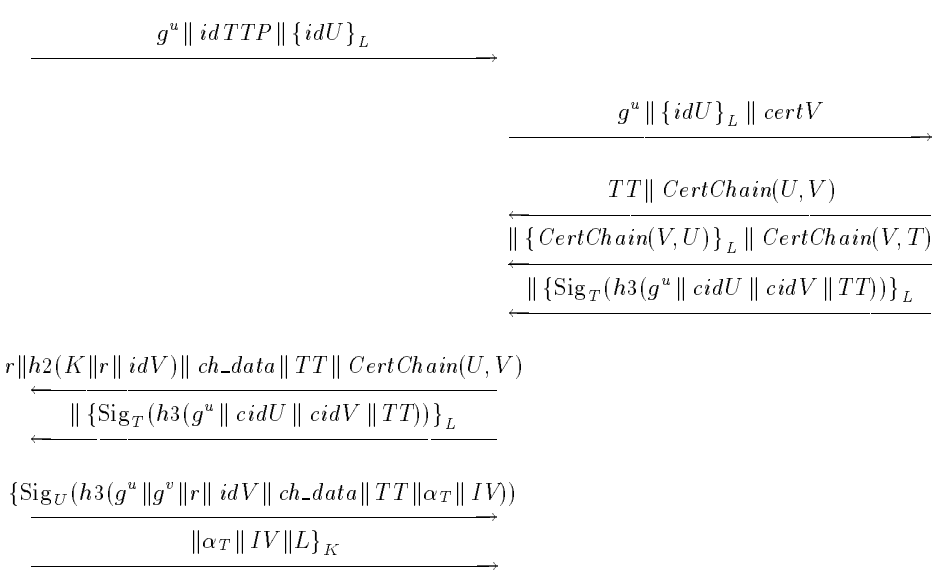

Fig. 3. The authentication protocol with on-line TTP

user and forwards the information sent by the user together with his certificate $certV$. It is assumed that the user's identity is sufficient for the TTP to retrieve the appropriate certificate. The TTP decrypts $\{idU\}_L$ and verifies whether the certificate of U has been revoked; it might also perform the same verification for the certificate of V (which might involve contacting his TTP, if V has a different TTP). If the certificates are still valid, the TTP generates a time-stamp TT, and sends to V a cross-certificate chain (see Sect. 4.2 below); the part that may reveal U's identity is encrypted using L. Together with the time-stamp, the TTP signs the unique certificate identifiers $cidU$ and $cidV$ as well as g^u. The TTP Sends TT, the (partly encrypted) cross-certificate chain and the encrypted signature to V, who then forwards the encrypted signature and the cross-certificate for his public key to U. U can verify the freshness of the signature by the TTP since it also includes g^u (note that U may not have a reliable clock). It is assumed that the user knows the unique identifier number of the certificate on his own public key and on that of V so that he can verify the received signature. If the signature is correct, the user sends the final message to V, which also includes the key L encrypted under the key K. Using the key L, V can decrypt the answer received from the TTP in the third step and verify the signature. It is assumed that V has a reliable clock, so he can verify TT.

NOTE: this protocol has the property that V can only verify the signature of the TTP after the last step, which results in some additional delay. If this is not acceptable, the user can add in the first message a MAC computed with a

key derived from L on idU and idV (as an opponent will only see a single *(text, MAC)* pair for a given key, the required MAC property is weak – but slightly stronger than the weak MAC property discussed in Sect. 2.3). In that case the encryption in the fourth message can use the key K and there is no need to send L in the fifth message. However, this latter solution does not provide protection on the interface between V and TTP.

4.2 Cross-certificates and TTP scenarios

The protocol in Sect. 4.1 uses cross-certificate chains. Such chains are required when parties in the protocol do not have the same TTP, or when the parties do not have on-line access to their TTP. Here $CertChain(X, Y)$ consists of a sequence of certificates, c_0, $c_1, \ldots$, c_n, where the signer of certificate c_0 is the Certification Authority (CA) of entity X, the subject of c_i is equal to the signer of c_{i+1} $(0 \leq i < n)$, and the subject of certificate c_n is entity Y. Such a certificate is verified starting with c_0 (using the public key of the CA of entity X); this guarantees the public key required to verify c_1, etc. The verification is completed after verification of c_n. In order to speed up the verification process and reduce the communication overhead, the CA of entity X might also verify the complete chain, and then create a new certificate for entity Y. However, this provides slightly different guarantees to the entity verifying the cross-certificate.

For the protocol of Sect. 4.1, a CA structure where the CA of a user (the TTP) and the CA of a VASP V always cross-certify one another, will provide short certificate chains. Under this assumption — which is, of course, not necessary for the correct functioning of the protocol — we have:

- $CertChain(U, V)$ consists of a cross-certificate signed by the user's TTP, and $certV$.
- $CertChain(V, U)$ consists of a cross-certificate signed by V's CA, and $certU$.
- $CertChain(V, T)$ consists of a cross-certificate signed by V's CA, and a certificate on the signature key used in Sig_T by the user's TTP.

Of course the protocol can be simplified if U and V have the same TTP. Moreover, one can also consider the case where V contacts his own TTP, rather than that of the user. In that case the user should send its certificate $certU$ over the air interface in the first protocol step, reducing the efficiency of the protocol and compromising user identity confidentiality.

5 Choices for Cryptographic Algorithms and Certificate Formats

The protocol and cryptographic mechanisms were chosen in such a way that they are particularly suited to the low bandwidth and low computational capabilities on the user's smart card. The payment protocol itself is very lightweight; the elementary payment operation does not require any public-key operation. The authentication protocol is (essentially) identical to one proposed to ETSI for

UMTS user-to-network authentication [ETS4]. The cross-certification approach is chosen in order to minimise communication overheads.

5.1 Cryptographic algorithms

The use of elliptic-curve cryptography allows for much shorter signatures and keys; some additional storage is required for the system parameters, but this can be minimised by selecting common system parameters. Current estimates [Wien] indicate that elliptic curves with a 170-bit subgroup order (which typically corresponds to an elliptic curve over a group with $171 \ldots 180$ bits) offer a security level comparable to 1024-bit RSA. The signature scheme used is the AMV scheme of ISO/IEC FCD 14888-3 [ISO4], but the construction of the RSA based signature scheme of ISO/IEC 9796-2 [ISO1] has also been planned as an option for the signature on certificates. The hash function RIPEMD-128 can be used for $h1$ and $h2$ in the authentication protocol; this hash function offers a performance in between MD5 (the security of which is questionable) and SHA-1 and RIPEMD-160. For collision resistance, 128 bits is on the low side, but for the specific needs of $h1$ and $h2$ in the authentication protocol it certainly provides a high security level. For $h3$ RIPEMD-160 has been selected. For the tick payment protocol, RIPEMD-160 has been selected, restricted to an output of 64 bits.

5.2 Certificate format

A special certificate format has been designed that minimises the storage space on the smart card and the bandwidth on the air interface. The size of a public-key certificate is less than 200 bytes, which should be compared to about 1 Kbyte for a typical X.509 v.1 certificate [ITU1] (and certificates proposed within IETF). The certificate allows for all necessary information: version number, serial number, issuer identifier, four validity dates (begin and end of validity and two optional dates for usage of the private key), subject identifier and public key information (algorithm type identifier and a public key value). Other optional fields include key usage, cross certificate attributes and certificate path attributes. Similar ad hoc certificate formats are being used in the financial sector (e.g., for the EMV specifications by Europay, Mastercard and VISA).

6 Related Work

There is a vast literature on authentication and key agreement protocols. For overviews, the reader is referred to [MvOV] or [RuvO]. A protocol related to the one presented in Sect. 2 is the well-known Station-to-Station protocol with its variations [DvOW]. As opposed to our protocol it has longer messages and higher computational requirements, irrespective of the choice of the signature systems used by the user and the certification authority, respectively, and the choice of the finite group in which the exponentiations required in the key agreement scheme are computed. Proposals for authentication protocols specifically designed for

mobile systems were made in [AzDi,BeYa,LiHa] (see also Chapter 12 in [MvOV] for additional references). However, they are either too inefficient for use in UMTS, do not achieve all the protocol goals stated in Sect. 2.1 or do not allow the integration of a payment system as described in Sect. 3 of this contribution. An overview of the ASPeCT trial protocol and some trial details are presented in [MaPM].

7 Conclusion

The protocols presented in this paper provide an efficient way to achieve mutual authentication, key establishment and incontestable charging in a mobile environment. The protocol satisfies the needs for UMTS: it requires a low computational load on the user's side and requires only a limited amount of communication. Moreover, it can be extended to a large scale system with multiple TTPs through cross-certification.

Acknowledgements

We would like to thank the following colleagues for their help and advice: Peter Howard, Volker Kessler, Lars Knudsen, Keith Martin, Chris Mitchell, Klaus Müller and Konstantinos Rantos. We would also like to thank all the partners in the ASPeCT project for their good cooperation and the anonymous referees for their helpful comments.

References

[AnMS] R. Anderson, H. Manifavas, C. Sutherland. "A practical electronic cash system." Available from `http://www.cl.cam.ac.uk/users/rja14/`

[AzDi] A. Aziz, W. Diffie, "Privacy and Authentication for Wireless Local Area Networks," *IEEE Personal Communications*, 1st Q 1994, pp. 25–31.

[BeYa] M.J. Beller, Y. Yacobi, "Authentication and key agreement protocol for PCS," *Joint experts meeting on privacy and authentication for PCS,* P&A JEM/93-012, Nov. 8, 1993.

[DiHe] W. Diffie, M.E. Hellman, "New directions in cryptography," *IEEE Trans. on Information Theory*, Vol. IT–22, No. 6, 1976, pp. 644–654.

[DvOW] W. Diffie, P. van Oorschot, M. Wiener, "Authentication and authenticated key exchanges," *Design, Codes and Cryptography*, Vol. 2, 1992, pp. 107-125.

[DoBP] H. Dobbertin, A. Bosselaers, B. Preneel, "RIPEMD-160: a strengthened version of RIPEMD," *Fast Software Encryption, LNCS 1039*, D. Gollmann, Ed., Springer-Verlag, 1996, pp. 71–82.

[ElGa] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. on Information Theory*, Vol. IT–31, No. 4, 1985, pp. 469–472.

[ETS1] ETSI ETS GSM 02.09, *"European Digital Cellular Telecommunications System (Phase 2); Security Aspects,"* Version 4.2.4, September 1994.

[ETS2] ETSI ETS 300175-7, *"DECT Common Interface, Part 7: Security Features,"* October 1992.

[ETS3] ETSI ETR 33.20, *"Security Principles for the Universal Mobile Telecommunications System (UMTS),"* Draft 1, 1997.

[ETS4] ETSI SMG SG DOC 73/95, *"A public key based protocol for UMTS providing mutual authentication and key agreement."*

[FIPS] FIPS 180-1, *"Secure Hash Standard,"* Federal Information Processing Standard (FIPS), Publication 180-1, National Institute of Standards and Technology, US Department of Commerce, Washington D.C., April 1995.

[HaSW] R. Hauser, M. Steiner, M. Waidner, "Micro-payments based on iKP," *Presented at SECURICOM 96.* Available from `http://www.zurich.ibm.com`.

[ISO1] ISO/IEC 9796-2, *"Information technology – Security techniques – Digital signature schemes giving message recovery, Part 2: Mechanisms using a hash-function,"* 1997.

[ISO2] ISO/IEC 10118, *"Information technology – Security techniques – Hash-functions, Part 1: General,"* 1994, *"Part 2: Hash-functions using an n-bit block cipher algorithm,"* 1994, *"Part 3: Dedicated hash-functions,"* 1998.

[ISO3] ISO/IEC 11770-3, *"Information technology – Security techniques – Key management, Part 3: Mechanisms using asymmetric techniques,"* 1997.

[ISO4] ISO/IEC FCD 14888-3, *"Information technology – Security techniques – Digital signature with appendix, Part 3: Certificate-based mechanisms",* 1997.

[ITU1] ITU-T Recommendations X.509, *"Authentication Framework,"* Geneva 1989.

[ITU2] ITU, *"Security Principles for Future Public Land Mobile Telecommunication Systems,"* Rec. ITU-R M. 1078.

[JuYu] C.S. Jutla, M. Yung, "Paytree: amortised-signature for flexible micropayments," *Proc. of Second USENIX Association Workshop on Electronic Commerce*, November 1996, pp. 213-221.

[KnPr] L.R. Knudsen, B. Preneel, "One-way functions for tick payments," in preparation.

[Lamp] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, Vol. 24, 1981, pp. 770–772.

[LiHa] H. Lin, L. Harn, "Authentication in Wireless Communications," *Proc. GLOBECOM 1993.*

[MaPM] K.M. Martin, B. Preneel, C.J. Mitchell, H.J. Hitz, G. Horn, A. Poliakova, P. Howard, "Secure billing for mobile information services in UMTS," *Proc. IS&N'98*, to appear.

[Mene] A. Menezes, *"Elliptic Curve Public Key Cryptosystems,"* Kluwer Academic Publishers, Boston, 1993.

[MvOV] A. Menezes, P. van Oorschot, S. Vanstone, *"Handbook of Applied Cryptography,"* CRC Press, Boca Raton, 1997.

[Pede] T. Pedersen, "Electronic payments of small amounts," *Security Protocols, LCNS 1361*, M. Lomas, Ed., Springer-Verlag, 1997, pp. 59–68. See also *DAIMI PB-495*, Computer Science Department, Aarhus University, August 1995.

[RiSh] R.L. Rivest, A. Shamir, "PayWord and MicroMint: two simple micropayment schemes," *Security Protocols, LCNS 1361*, M. Lomas, Ed., Springer-Verlag, 1997, pp. 69–87.

[RuvO] R. Rueppel, P. van Oorschot, "Modern key agreement techniques," *Computer Comm.*, Vol. 17, No. 7, July 1994.

[UMTS] UMTS Forum, *"A regulatory framework for UMTS,"* Report no. 1, 1997.

[Wien] M. Wiener, "Performance comparisons of public-key cryptosystems," *Presented at the 1998 RSA Data Security Conference*, San Francisco, January 12-16, 1998.