Applying Cryptography within the ASPeCT Project

By Keith Martin, COSIC Research Group, Department of Electrical Engineering-ESAT, Katholieke Universiteit Leuven, Belgium.

Abstract: *This article considers the application of cryptographic techniques within the. ASPeCT project. ASPeCT is concerned with the provision of certain advanced security features within future mobile telecommunications systems, and we concentrate on two areas of interest within ASPeCT where cryptography is directly being used to provide security solutions: the provision of Trusted Third Party services and the secure billing of value-added information services.*

Introduction to ASPeCT

Mobile telecommunications present a challenging environment in which to provide security functionality, not just because of the international and dynamic nature of a mobile telecommunications network, but also because of the evolutionary process that is currently acting on the mobile telecommunications world. A rapidly increasing base of mobile users are becoming served by a growing number of operators, who are offering these users a proliferating variety of services. Addressing the evolving security needs of both users and service providers within existing and future personal communication networks by studying the feasibility and acceptability of new and advanced security features is the general objective of the ACTS[1] *Advanced Security for Personal Communications Technologies* (ASPeCT) project. This work is being done through the development, implementation and execution of several public trials and demonstrations.

ASPeCT is a European funded consortium of industrial and academic partners, each providing their own experience and technical expertise in security theory, application of security technology, development of security products and provision of security services for mobile telecommunications. The ASPeCT partners are Vodafone Ltd and Royal Holloway, University of London (United Kingdom), Siemens ATEA and Katholieke Universiteit Leuven (Belgium), Giesecke und Devrient GmbH and Siemens AG (Germany) and Panafon SA (Greece), plus associate partner Lernout and Hauspie (Belgium).

Developing on the first use of cryptographic technology to provide security features in second generation mobile systems, such as the *Global System for Mobile Telecommunications* (GSM), the focus of ASPeCT is on the provision of security services for third generation systems such as the *Universal Mobile Telecommunications System* (UMTS), which is likely to become a widely adopted international telecommunications standard. Recent advances in security technology and mechanisms mean that advanced cryptographic techniques can be applied for the first time within mobile networks to address the new diverse security requirements of systems such as UMTS. ASPeCT is exploiting this by

[1] The *Advanced Communications Technologies and Services* (ACTS) programme was established under the Fourth Framework Programme for European activities in the field of research and technological development and demonstration. The ACTS programme supports over 150 projects based in countries throughout the European Union. For more information see http://www.uk.infowin.org/ACTS/

making extensive use of these new capabilities, and in particular the use of public key cryptography, in the design of relevant security protocols.

ASPeCT Work Packages

The main technical work within ASPeCT has been divided into seven separate work packages, a brief description of which serves as a good overview of the main topics being explored by the project.

1. **Migration towards UMTS security.** It is important that the new security services offered within UMTS are developed in a manner that is sympathetic with those already offered in second generation systems. It is thus essential to consider the feasibility of more advanced authentication mechanisms and the movement of users towards UMTS, while entertaining requirements such as the need for cost-effective migration and the support of roaming capabilities[2] between second generation terminals and those of UMTS.

2. **Detection and management of fraud in UMTS networks.** Although technical security techniques such as the use of cryptography can be used to make the perpetration of fraud within a communications network more difficult, they can not prevent all forms of commercial fraud. The principle aim of this work package is to develop sophisticated fraud detection techniques that can assist in the early detection of fraud attempts. This development includes the identification of likely fraud scenarios and associated indicators. In particular, new tools are being developed using rule or neural network-based approaches that can be used for fraud scenarios where existing detection methods have proved inadequate.

3. **Trusted Third Parties.** Many of the new security services that it will become possible to offer within UMTS arise because it has become possible to use public key cryptographic techniques for the first time. A significant number of supporting services for public key cryptography require the active involvement of trusted organisations, normally referred to as *Trusted Third Parties* (TTPs). Investigations within ASPeCT concentrate on the identification and development of relevant TTP services within UMTS.

4. **UIM security functionality.** This work package is concerned with determining the possible functionality that can be provided by existing and future *User Identity Modules* (UIMs), which are the smart cards that provide access to UMTS. Particular issues include the migration towards UMTS from a UIM perspective, the provision of multiple smart card applications, and the exploration of possible biometrics techniques for authentication between users and their smart cards.

5. **Security and integrity of billing in UMTS.** Complexity of billing for mobile services is likely to increase in UMTS, not only because networks will involve more intricate relationships between an increased number of operators, service providers and users, but also because a greater number and variety of *value-added services* (VASs) will be being purchased by users over these networks. It is consequently important to study billing techniques that offer both security and integrity, in order to attempt to address the greater potential for occurrence of billing disputes within UMTS.

6. **Presentation of fraud detection.** It is an important practical issue to complement the development of fraud detection techniques within ASPeCT with a study of the legal issues arising from the use of computer generated evidence. The goal is to be able to provide evidence that is acceptable in courts

[2] Roaming capabilities in mobile cellular systems such as GSM and UMTS allow users to make and receive calls in mobile networks other than the home network to which they have subscribed.

of law under existing rules and that will support prosecution of fraud (or attempted fraud) with respect to services and other aspects of UMTS. One particular concern is to ensure that the developed processes are compatible with the different legal environments within the various participating nation states (within ASPeCT this is limited to the member states of the European Union).

7. **User authentication by vocal biometrics.** The last work package is concerned with seeking alternatives to traditional typed password and PIN based identification and authentication techniques. Objectives include conducting research on speaker verification by means of vocal properties extracted from both single and multiple passwords, and the demonstration of how vocal biometrics can be used to implement user authentication for UMTS networks.

In this article, in order to provide a taste of the type of cryptography being applied within ASPeCT, we concentrate on providing further details of work conducted by the two work packages that most heavily apply the use of cryptographic mechanisms: *Trusted Third Parties* and *Security and integrity of billing in UMTS*. Within each of these work packages the aims of ASPeCT are the identification of necessary security services, the design of cryptographic methods of providing these services, and the implementation and testing of these services by means of public demonstrators.

Trusted Third Parties

Trusted Third Parties are increasingly seen as essential components of any large scale security infrastructure. Potential TTP services include key management (of both symmetric and public key cryptosystems), key recovery, identification and authentication functionality, access control, and non-repudiation. ASPeCT is primarily concerned with the use of TTPs to provide mobile telecommunication services and to this end identified key management of public key cryptography and key recovery services for the first TTP demonstration within the project. This involves using TTPs to perform basic security functions such as public key generation and distribution, key storage and retrieval, and public key certification.

Key Recovery Services

The provision of key recovery (often referred to as *key escrow*) services is a particularly sensitive issue that has attracted a great deal of recent attention and vigorous debate. In any cryptographic system that intends to offer end-to-end confidentiality, there may be a need to permit certain parties under special circumstances to access keys used to encrypt communications. Such parties and circumstances could be law enforcement or security organisations in the event of a criminal investigation, or could be local security managers in the event of loss or damage to previously used cryptographic keys. With respect to key recovery within UMTS, the starting premise within ASPeCT is that it may be the case that legal requirements at both national and international level dictate that key recovery services must be provided within a mobile telecommunications network. The main objective of ASPeCT is thus to demonstrate the provision of key recovery services through the assistance of TTPs, given that such a service is required. It is not a central objective of ASPeCT to add to the debate over whether key recovery is a desirable service, although technical results obtained during the establishment of the working demonstrators may contribute to this wider discussion.

The First TTP Demonstrator

The first TTP demonstrator has been completed and was publicly exhibited[3] in May 1997. The demonstrator explicitly shows the use of TTPs in establishing an end-to-end encryption service, with the TTPs providing key generation, distribution and certification services. The demonstrator also implicitly provides a key recovery service using the same TTPs. We now describe the technical solution adopted in this demonstrator.

Demonstrator Players

The protocols used involve a pair of mobile users, where one user wishes to send the other user a confidential message and needs to be provided with a session key to protect it. Each user is associated with a TTP (the *home TTP* of the user) which is assumed to lie within the same legal jurisdiction (perhaps nation state) as the user. It is assumed that each TTP is located within the jurisdiction of some *interception authority* (a party requiring the key recovery service) and that each TTP operates subject to the relevant laws and regulations governing that authority. Note that the users may well be in different jurisdictions, a most important factor that affects the choice of technical solution within a mobile network. We label the users by A and B, and their respective home TTPs by TA and TB.

Starting Position

The two TTPs TA and TB have agreed upon a large prime number p (where $p-1$ is divisible by another large prime q) and agreed upon a primitive element[4] g modulo p. These values have been passed on to users A and B. Further, TA and TB agree upon a key generating function h which takes as input a user identity and a secret value, and outputs a key value. Each TTP has a *private* and *public signature key* pair. The private signature key is known only by the TTP, whereas an authenticated copy of the public (verification) signature key can be accessed by the client user of that TTP and the other TTP. Finally, each user and the home TTP of that user have access to a protected channel between them which provides origin authentication, data integrity and confidentiality.

Main Protocol

The protocol implemented in the first TTP demonstrator is based on the *JMW Protocol* first proposed in [1]. Figure 1 shows the message exchanges of the protocol among the four players. The protocol includes four parts, each of which can optionally be run separately.

[3] The first TTP demonstrator was publicly exhibited at IS&N ’97, the Fourth International Conference on Intelligence in Services and Networks, Cernobbio, Italy, May 27-29, 1997.

[4] A *primitive element g* generates all the numbers modulo *p*, in the sense that for each number *x* modulo *p*, there exists an integer *i* such that $x=g^i$.

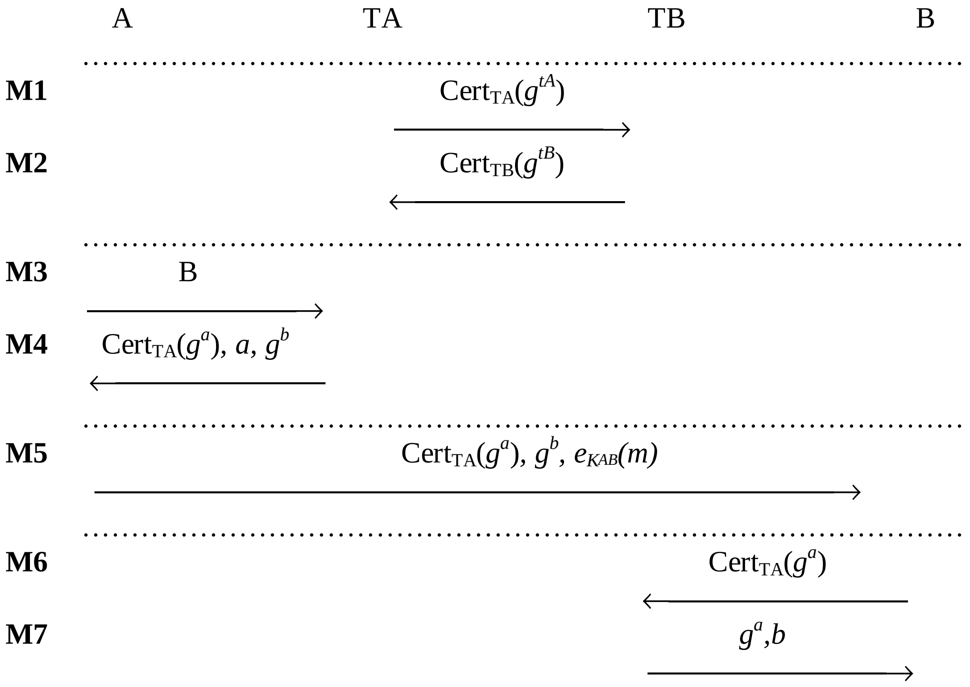
A
TA
TB
B
M1
$\mathrm{Cert}_{\mathrm{TA}}(g^{tA})$
M2
$\mathrm{Cert}_{\mathrm{TB}}(g^{tB})$
M3
B
M4
$\mathrm{Cert}_{\mathrm{TA}}(g^{a}), a, g^{b}$
M5
$\mathrm{Cert}_{\mathrm{TA}}(g^{a}), g^{b}, e_{KAB}(m)$
M6
$\mathrm{Cert}_{\mathrm{TA}}(g^{a})$
M7
g^{a},b

Figure 1: The First TTP Demonstrator Protocol

A more detailed explanation of the protocol exchanges is now given:

Establishing a shared secret key between TA and TB:

1. Each TTP generates a private and public (*key agreement*) key pair: (tA, g^{tA}) and (tB, g^{tB}).
2. In **M1**, **M2,** TA and TB exchange certified public keys $\mathrm{Cert}_{\mathrm{TA}}(g^{tA})$, $\mathrm{Cert}_{\mathrm{TB}}(g^{tB})$, each signed using the appropriate private signature key.
3. Each TTP verifies the received public key using the authenticated public signature verification key of the other.
4. Each TTP computes a shared secret, $K_{\mathrm{TATB}} = g^{tAtB}$, using his own private key and the other's public key (Diffie-Hellman key establishment [2]).

At the end of this part of the protocol TA and TB have jointly established a common key with which they can later both generate the private receive key of user B.

Certificate generation in A's domain:

5. A sends TA a request to communicate with user B in **M3**.
6. TA generates a random number a as A's *private send key* and computes the corresponding *public send key*, g^{a}.
7. TA generates a certificate for A's public send key, $\mathrm{Cert}_{\mathrm{TA}}(g^{a})$.
8. TA computes B's *private receive key,* $b = h(K_{\mathrm{TATB}}, \mathrm{B})$, and the corresponding *public receive key*, g^{b}.
9. In **M4**, TA sends A's certified public send key, A's private send key and B's public receive key to A.
10. A computes a shared key, $K_{\mathrm{AB}} = g^{ba}$, using his own private send key and B's public receive key.

At the end of this part of the protocol user A has all the information that he needs to send a message to user B.

Message transmission from A to B:

11. A sends to B in **M5** his public send key certificate issued by TA , B’s public receive key, and a message m encrypted by the shared key, K_{AB} .

The communication from A to B is now complete, however user B must now obtain the relevant information from TB that permits decryption.

Certificate generation in B’s domain:

12. After receiving **M5**, B sends TB a request **M6,** including A’s public send key certificate issued by TA, and B's public receive key sent by A.
13. TB computes B’s private receive key, $b = h(K_{\mathrm{TATB}}, \mathrm{B})$, and verifies g^b.
14. TB verifies A's public send key certificate using the public signature verification key of TA.
15. TB sends A’s public send key and B’s private receive key to B.
16. B computes the shared key, $K_{\mathrm{AB}} = g^{ab}$, using his own private receive key and A’s public send key.

The protocol is now complete since user B now has all the information necessary to compute the shared key K_{AB} (the *session key*) used to encrypt (and decrypt) the message. Further, both TA and TB have sufficient information to be able to reconstruct and communicate the session key if requested to do so by the interception authorities in either jurisdiction (see next section for more details on the full key recovery options). Note that many simplified versions and variations of this protocol are possible (for example, in a simplified version user B could be issued with his private receive key in advance, and in a variation user A could generate his own private receive key).

Choice of Cryptographic Primitives and Protocols

The cryptographic support is provided by the proprietary *Advanced Cryptographic Library* (*ACRYL*). The main ACRYL functions called upon for the first demonstrator are random key generation (provided by a random number generator based on the DES symmetric cryptosystem [3] in OFB mode [4]), the hash function RIPEMD-160 [5] (used as the key generating function h in the main protocol), RSA signatures based on ISO/IEC 9796-2 [6], exponentiation, and encryption using DES in CBC mode [4].

Within ASPeCT there are two types of certificate utilised, depending on the use of different signature mechanisms. As well as the RSA signatures used in the first demonstrator, other work packages use signatures based on ISO/IEC 14888-3 [7]. Within ASPeCT both these types of certificate make use of a universal certificate information sequence format, with the certificate type being specified by the left-most byte of the certificate string.

The choice of a key recovery method based on the JMW Protocol is due to the design features of the JMW Protocol that permit efficient key recovery in the jurisdictions of both the sending and the receiving user. This is essential in the case that the users are in different judicial domains, where it would seem unacceptable to demand that an interception authority in one domain would need the co-operation of a TTP in another domain in order to recover a particular cryptographic key. It is clear that either TTP has the ability to generate the session key (for example, TA can use its knowledge of the secret send key of A together with the public receive key of B to compute the session key). It is also possible that private send and receive keys could be issued with the intention of allowing them to be used in more than one communication (thus, for example, A does not always request a new private send key from TA, but reuses a given private send key for all session keys, regardless of the receiver, generated within a designated time period). In this case the JMW Protocol also offers the interesting feature that TTPs could, under appropriate conditions , present higher order keys than session keys to an interception authority, enabling the recovery of more than one session key through the release of just one item of information. For example, TA could present the private send key of A to an interception authority, allowing recovery of all session keys between A and any user with whom A has communicated within the time period for which the private send key of A is valid. It should be noted that such key releases by TTPs should only be permitted in the case that permission to do so (such as the presentation of a valid legal warrant) covers all possible time periods and targets whose session keys can be thus recovered.

Evaluation and Further Activities

Following the public presentation of the first demonstrator, a detailed evaluation of the demonstrator was conducted with a view to enhancing the demonstrator and identifying tasks for the next phase of the project. While the current implementation appears to work well, it has some technical limitations, and among further tasks identified was the desire to compare the JMW-based solution with some alternative methods of providing key recovery. Higher on the priority list for ASPeCT however is the desire to demonstrate further TTP services such as key revocation and time-stamping. The current phase of the ASPeCT project involves the establishment of a second demonstrator which shows the application of TTP services to assist secure billing, a topic which we now address.

Secure Billing

There is currently a great deal of interest, and an increasing range of solutions and products, within the general area of electronic commerce. It is an important aim of ASPeCT to investigate how best to apply this emerging technology to the problem of providing security and integrity of billing of mobile network value-added services, which represent a new range of services that will be available to users of UMTS. As an example of VASs, it can be imagined that similar to the developments in internet based purchasing, a future mobile user may be able to securely pay for various information services available to him. This information is likely to be more dynamic and sophisticated than information services currently available for purchase over a phone link. Thus a user may be able to purchase graphical information detailing recent changes in foreign exchange rates, or may be able to purchase travel information in the form of street maps, interactive railway timetables, or photographs of potential accommodation. The increase in power and sophistication of user terminals is likely to be matched by the increase in variety and sheer number of value-added services being offered by *value-added service providers* (VASPs).

Micropayments

The most significant factor in designing a secure electronic payment scheme suitable for application to the purchase of VASs over a mobile network is that the unit cost, and in many cases the total cost, of any purchase is likely to be very small. For this reason, such purchases are normally referred to as *micropayments*, and any suitable payment scheme referred to as a *micropayment scheme*. It is essential that any micropayment scheme involves extremely low processing and communication costs, otherwise the scheme is simply uneconomical. The minimisation of overheads can often be facilitated by a weakening of security requirements with respect to normal payment mechanisms, since the potential loss in any individual case of fraud is relatively low.

The First Secure Billing Demonstrator

The first Secure Billing demonstrator has been completed and, like the first TTP demonstrator, was exhibited at IS&N '97. The demonstrator explicitly shows a method of permitting mobile users to pay for access to information services in a flexible, efficient and secure way. The demonstrated method has potential application to charging for any telecommunications service.

Demonstrator Players

The protocols that we shortly describe involve just two parties: a mobile user, equipped with a UIM (smart card), and a VASP. The user wishes to purchase (using *micropayment tokens*) a VAS from the VASP. Although only these two entities communicate on-line during the protocols, a third entity plays the role of off-line broker in processing the payments. The relationship between these entities in the charging model is given in Figure 2.

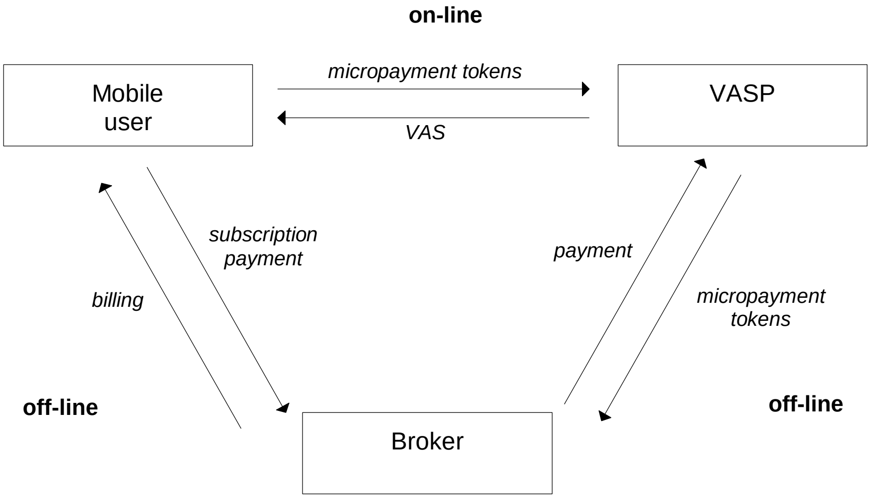

Figure 2: Charging Model

The role of broker in this model is likely to be filled by the UMTS service provider. The broker takes care of all payments to the network operators and, perhaps through an appropriate TTP, offers facilities such as issuing certificates of credentials to the mobile user, checking certificates of credentials on behalf of the VASP, billing the user, and clearing payments.

The billing process in the first Secure Billing demonstrator can be split into two protocols. In the *Initialisation Protocol*, the user and the VASP authenticate one another, and the user commits himself to a starting value and tariff for the micropayment scheme. In the *Micropayment Protocol* the user actually transfers the micropayment tokens to the VASP. We describe each of these protocols separately.

Starting Position (Initialisation Protocol)

The user U and VASP V are assumed to jointly possess a symmetric encryption function, a random number generator, a one-way function h2, hash functions h1 and h3, and a length preserving one-way function F. User U possesses a private and public signature key pair signature transformation Sig_U .

An authenticated copy of the public signature key is assumed to be available at V. Finally, V possesses private and public key pair (v, g^v) respectively, where an authenticated copy of g^v is available at U[5].

Initialisation Protocol

The goals of the Initialisation Protocol are to provide mutual authentication of the user and the VASP, to establish an agreed session key and to initialise the Micropayment Protocol. The main exchanges of the Initialisation Protocol are shown in Figure 3.

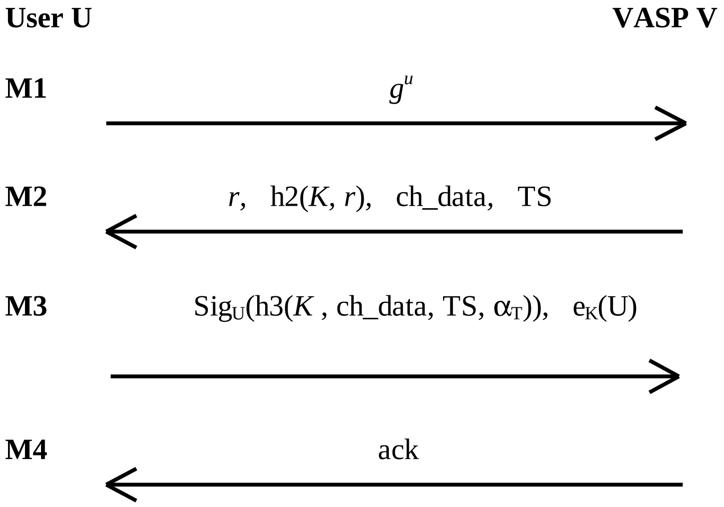
User U
VASP V
M1
g^u
M2
r, h2(K, r), ch_data, TS
M3
$\text{Sig}_U(\text{h3}(K, \text{ch\_data}, \text{TS}, \alpha_T))$, $e_K(\text{U})$
M4
ack

Figure 3: Secure Billing Initialisation Protocol

Initiating the communication session:

1. U generates a random number u and computes a temporary key g^u
2. U sends g^u to V in message **M1.**

At the end of this part of the protocol, V is aware that U has requested a communication session.

Confirmation of derived key and tariff setting:

3. V generates a random number r.
4. V computes the session key $K = \text{h1}(\text{f}((g^u)^v), r)$ and h2(K, r).
5. V sends message **M2** to U, where ch_data contains information on the tariff to be applied, and TS is the time according to V.

After this part of the protocol is completed, U has received V's commitment to key K and has received the tariff information regarding the billing charge rates.

Commitment to charge data:

[5] In the implementation we use an elliptic curve public key cryptosystem E over a field of prime p elements whose parameters q (size of the curve), g_x and g_y (co-ordinates of a generator g of the curve), a and b (coefficients of the defining equation) are configurable, and for which there is a function f mapping E onto the numbers in the range [0..q-1].

6. U computes K = h1(f(g^v)^u), r) and confirms that V has computed the same value of K by computing and checking h2(K, r).
7. U checks that TS matches the time according to U sufficiently closely.
8. U generates a random value a_0 and computes a_T = F ^{T}(α_0), where T is a predetermined system parameter (see description of Micropayment Protocol).
9. U sends message **M3** to V, where e_K(U) is the symmetric encryption of the identity of user U, protected in this way to preserve the anonymity of the transaction.

V has now received U's commitment to key K and has received the important initial commitment a_T that will be used in the Micropayment Protocol.

Initialisation of Micropayment Protocol variables:

10. V verifies the signed part of **M3** and in doing so confirms U's knowledge of K.
11. V stores the signature and relevant parameters for later billing purposes.
12. V initialises the Micropayment Protocol variables j ¬ T, tck_cnt ¬ 0, a ¬ a_T.
13. V sends an acknowledgement message **M4** back to U.

The Initialisation Protocol is now complete and the user and the VASP have mutually authenticated one another, have agreed upon session key K, and are ready to run the Micropayment Protocol. It is important to note that for this, and other protocols in ASPeCT, in the event that a check fails the protocol is abandoned and no service is provided.

Micropayment Protocol

The Micropayment Protocol used in ASPeCT is a credit-based system based on Pedersen's *Tick Payment Scheme* [8], (see also related schemes [9,10,11,12]). Micropayment tokens are transferred from the user to the VASP by releasing pre-images of the one-way function F (one pre-image represents one unit charge). During the Initialisation Protocol, the user commits to a starting value a_0 and its T-th image $F^T(\alpha_0)$, then pays for the k-th micropayment by transferring the micropayment token $F^{T-k}(\alpha_0)$. The VASP cannot forge a payment token as the VASP should not be able to compute pre-images by inverting function F. The parameter T represents the maximum number of micropayment tokens that can be transferred with respect to a given (signed) commitment. When the maximum number T tokens is reached, then a shortened version of the Initialisation Protocol is re-run in order to re-initialise the Micropayment Protocol, and then the Micropayment Protocol is re-run using the new commitment. The two essential steps in the Micropayment Protocol are indicated in Figure 4.

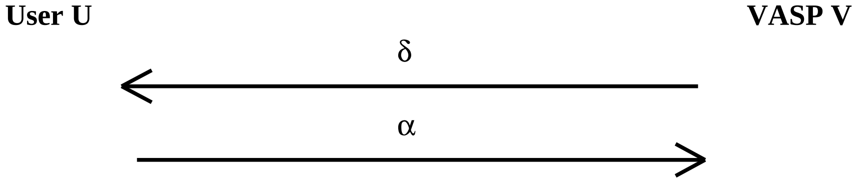

Figure 4: Micropayment Protocol

At a given instant in the Micropayment Protocol let t be the number of tokens already sent by U and let $d$be the number of tokens whose payment is requested by V in the current protocol run.

Request of payment:

1. V checks to see if $t + d >$ T (if so, then re-initialise the Micropayment Protocol).
2. V sends a request for δ tokens to U.

Transfer of micropayment tokens:

3. U sets $\alpha \leftarrow F^{T-(t+d)}\left(a_0\right)$
4. U sends the micropayment token α to V.

Check token and store variables:

5. V checks that $a^{t+d} = F^{T}\left(a_0\right)$.
6. V stores a, tok_cnt , where tok_cnt is the number of micropayment tokens that the user has transferred during the current run of the Micropayment Protocol.

At the completion of the Micropayment Protocol (either due to the termination of the communication session or due to the number of tokens exceeding the maximum number T permitted per commitment), the VASP stores a transcript of the session. This transcript forms the bill that is presented to the broker for payment and includes *K*, ch_data, TS, a_T, Sig_U(h3(*K*, ch_data, TS, α_T)), and the last received micropayment token a.

Choice of Cryptographic Primitives and Protocols

As for the first TTP demonstrator, the cryptographic support for the first Secure Billing demonstrator is provided by ACRYL. The functions h1, h2 and h3 are implemented using RIPEMD-128 [3,13], and the function F is implemented using RIPEMD-160, restricted to an output of 64 bits. The user computes signatures based on ISO/IEC 14888-3 and the VASP uses RSA-signatures. Symmetric encryption is by means of DES in CBC mode. In the first Secure Billing demonstrator $T = 2^{10}$.

The choice of authentication process in the Initialisation Protocol is largely due to the fact that a closely related version of this protocol has been submitted to ETSI-SMG for user-to-network authentication within UMTS [14]. Thus, by integrating the Initialisation Protocol into the user-to-network authentication process, operational costs are greatly reduced.

The choice of Micropayment Protocol is largely due to the tight storage and computational restrictions at the user end. During the Micropayment Protocol described, the user does not perform any computationally expensive signatures, but rather relies on one signature that is computed during the Initialisation Protocol. A further important property is that there is no need for any on-line communication with the broker during the entire protocol, allowing all clearing of payments and billing procedures to be conducted off-line at later convenience.

Evaluation and Further Activities

A detailed evaluation of the first Secure Billing demonstrator was conducted following the public presentation at IS&N ’97. The cryptographic component of the demonstrator was regarded as successful and several possible extensions were identified. These include adapting the current Micropayment Protocol to handle multiple micropayment token chains (perhaps for handling different currencies or values) and to permit the payment of multiple vendors per commitment. These extensions need some further research. Investigations also hope to shed some light on the selection of optimal parameter values, such as for the maximum token limit per commitment. The next phase of the secure billing study within ASPeCT is to incorporate TTP services within a working demonstrator, thus conducting a secure billing operation while involving the on-line use of TTPs in providing time-stamping facilities and certificate services.

Concluding Comments

We have described in some detail the type of cryptographic technology being applied in the first phase of just two of the work packages within the ASPeCT project, thus providing an illustration of theoretical cryptography being applied to build working demonstrators of security solutions for future mobile telecommunication networks. These are by no means the only areas within ASPeCT that cryptography is being applied. Other work packages are using cryptographic mechanisms to provide, amongst other things, user-to-network authentication, identity confidentiality and session key generation. Anyone wishing any further details of any of the activities within ASPeCT is encouraged to contact the project directly for more information [15].

The assistance of all ASPeCT project members, especially those involved in the TTP and Secure Billing work packages, is acknowledged in the preparation of this article.

References

[1] N. Jefferies, C. Mitchell and M. Walker. A proposed architecture for trusted third party services, Cryptography : Policy and Algorithms, Lecture Notes in Comput. Sci., 1029:98-104, 1996.

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

[3] H. Dobbertin, A. Bosselaers and B. Preneel, RIPEMD-160: a strengthened version of RIPEMD, Fast Software Encryption, Third International Workshop, Lecture Notes in Comput. Sci., 1039:71-82, 1996.

[4] FIPS 46, Data Encryption Standard, U.S. Department of Commerce / National Bureau of Standards, Springfield, Virginia, 1977.

[5] FIPS 81, DES modes of operation, U.S. Department of Commerce / National Bureau of Standards, Springfield, Virginia, 1980.

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

[7] ISO/IEC FCD 14888-3. Information technology - Security techniques - Digital signature with appendix - Part 3: Certificate-based mechanisms. November 1997.

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

[9] R. L. Rivest and A. Shamir. PayWord and MicroMint: Two simple micropayment schemes. Cryptobytes Vol 2, No 1, pp7-11, May 1996. Extended version also available from `http://theory.lcs.mit.edu/~rivest`

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

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

[12] C. S. Jutla and M. Yung. Paytree: "Amortised-signature" for flexible micropayments. Proceedings of Second USENIX Association Workshop on Electronic Commerce, 213-221, November 1996.

[13] ISO/IEC 10118-3:1998, Information technology - security techniques - hash-functions - Part 3: Dedicated hash-functions, to appear 1998.

[14] ETSI SMG SG DOC 73/95. A public key based protocol for UMTS providing mutual authentication and key agreement, September 1995.

[15] http://www.esat.kuleuven.ac.be/cosic/aspect/aspect.html