| | |
|---|---|
| Project Number | AC095 |
| Project Title | ASPeCT: Advanced Security for Personal Communications Technologies |
| Deliverable Type | Intermediate |
| Security Class | Public |

| | |
|---|---|
| Deliverable Number | D16 |
| Title of Deliverable | Secure billing : evaluation report |
| Nature of the Deliverable | Report |
| Document reference | AC095/SAG/W25/DS/P/16/1 |
| Contributing WPs | WP2.5 |
| Contractual Date of Delivery | May 1997 (Y03M03) |
| Actual Date of Delivery | 25 June 1997 |
| Editors | Günther Horn, Hans-Joachim Hitz |


| | |
|---|---|
| **Abstract** | This document provides a report on the evaluation of the first implementation of the ASPeCT secure billing service. |
| **Keywords** | ACTS, ASPeCT, Value Added Services, security, billing |

## TABLE OF CONTENTS

# 1 EXECUTIVE SUMMARY

This deliverable is an evaluation report on the first secure billing demonstrator developed in ASPeCT WP 2.5. The demonstrator shows the feasibility of a scheme for secure billing for mobile value added information services.

The demonstrator was finalised in February 1997 and was shown at the IS&N conference in Como at the end of May 1997. It is described in ASPeCT deliverable 10. The first demonstrator will be followed by a second demonstrator in February 1998 which will be developed jointly with ASPeCT WP 2.3. The second demonstrator will be integrated with the UMTS platform provided by the ACTS project EXODUS to perform a trial in a realistic environment.

It is the purpose of this deliverable to

- evaluate the first demonstrator with respect to
  - stated goals;
  - achievements;
  - shortcomings;
- suggest new functions and enhancements for the second demonstrator;
- show the potential of the chosen approach by pointing out further possible developments of the demonstrator which cannot be implemented due to the lack of resources in the project.

It was found that the demonstrator works according to the specification laid down in ASPeCT deliverable 7 and works well and in a stable manner. The use of a smart card on the user side and the use of the demonstrator in a mobile environment (access over GSM to a fixed server) could be successfully demonstrated. The impact of the security procedures is such that the user is not bothered by additional delays.

The most important ones among the suggested new functions for the second demonstrator are:

- Addition of an on-line component to access a certificate server in real-time so as to be able to authorise a user on-line if required by the security policy;
- additional functionality to increase of the flexibility of the charging model;
- further elaboration of the Graphical User Interface.

# 2 DOCUMENT CONTROL

## 2.1 Document history

| | |
|---|---|
| Version A (Extended table of contents) | 19-3-97 |
| Version B (first draft) | 23-5-97 |

## 2.2 Changes forecast

| | |
|---|---|
| Version C (second draft sent to WP 2.3/2.5) | 16-6-97 |
| comments on version C due | 19-9-97 |
| Version D (sent to PMC) | 20-6-97 |
| Final version sent to EC | 25-6-97 |

## 2.3 Change control

In conformance with the ASPeCT Quality Plan.

# 3 ABBREVIATION AND GLOSSARY OF TERMS

| | |
|---|---|
| API | Applications Programming Interface |
| ASPeCT | Advanced Security for Personal Communications Technologies |
| CA | An authority trusted by one or more users to create and assign certificates. |
| Certificate | a collection of unforgeable information, *signed* by a *CA*, conveying trusted information about the entity to which it relates |
| DECT | Digital European Cordless Telephony |
| DLL | Dynamic Link Library |
| ETSI | European Telecommunications Standards Institute |
| FSM | Finite State Machine |
| GSM | Global System for Mobile Communications |
| GUI | Graphical User Interface |
| HTTP | Hypertext Transfer Protocol |
| HW | Hardware |
| ISO | International Standards Organisation |
| NO | Network Operator |
| Signature | a message, or a hash (fingerprint) of the message enciphered with the private key (signature key) of the signatory (signer) |
| SIM | Subscriber Identity Module |
| SMG | Special Mobile Group |
| SP | Service Provider |
| SW | Software |
| TCP/IP | Transmission Control Protocol / Internet Protocol |
| TTP | Trusted Third Party |
| UIM | User Identity Module |
| UMTS | Universal Mobile Telecommunications System |
| URL | Uniform Resource Locator |
| User | human user or an application using a service or network (even where the application may itself be providing a service) |
| VAS | Value Added Service |
| VASP | Value Added Service Provider |
| Winsocks | Windows Sockets |

# 4 INTRODUCTION

The remainder of the deliverable is structured as follows:

Section 5 provides **background** information and an **overview** of the first secure billing demonstrator. This section is included so as to make D16 self-contained for the convenience of the reader. Most of its material is taken from previous ASPeCT deliverables.

Section 6 contains the actual evaluation of the demonstrator. This evaluation is carried out under a number of aspects:

*functionality*
This subsection shows how the scheme implemented in ASPeCT relates to other proposed electronic payment schemes and lists possible functional extensions such as the on-line use of Trusted Third Party services or enhanced flexibility in the choice of charging parameters.
*performance*
This subsection contains measurements on protocol execution times.
*security aspects and smart cards*
Here, the gain in security compared to currently used solutions is assessed. The role of smart cards in the security concept is described .
*applicability*
The ASPeCT secure billing demonstrator concentrates on billing for mobile value added information services. In this subsection, potential other applications (e.g. payment for basic telecommunications services) of the chosen approach are investigated.
*architecture*
This subsection assesses whether the developed component is suitable to be easily integrated in other systems and applications and compares the implemented architecture with alternative architectures.
*test environment*
Here, the suitability of the test environment to demonstrate the concepts is discussed.
*user-friendliness*
This subsection deals with issues such as the user-perceived quality of service and the suitability of the Graphical User Interface.
*appearance of demonstration*
This subsection looks at the demonstration from the observer's point of view.

The new functions and enhancements for the second demonstrator which were suggested in section 6 are summarised in section 7 as an easy reference for the project in the specification phase of the second demonstrator. It should be emphasised, however, that section 7 does not contain a list of fixed requirements for the second demonstrator. The suggested enhancements still need to be discussed in the light of estimates of the development effort and the available resources. This will only be done in the specification phase. Furthermore, harmonisation with the second demonstrator of Trusted Third Parties is required, but not yet taken into account here.

# 5 SECURE BILLING DEMONSTRATOR - BACKGROUND AND OVERVIEW

In a demonstration exhibited at the IS&N '97 conference in Como, the ASPeCT project showed how mobile users can pay for access to information services in a flexible, efficient and secure way. The method has potential application to charging for any telecommunications service.

It is generally accepted 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. 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 verify their feasibility and acceptability as part of demonstrations and trials. Some of these advanced security features in UMTS, in particular the use of public key cryptography, will be made possible through the use of more powerful smart card technology and the availability of Trusted Third Parties (TTPs) acting as certification authorities for public keys.

It is expected that the number and variety of value added services (VASs) will greatly increase while current networks are evolving towards UMTS. One reason for this is that users will possess terminals with more powerful processing and display capabilities than today's mainly speech orientated terminals. These terminals will integrate the functions of a mobile phone and a laptop or palmtop PC. The terminals will be used to access a wider variety of more advanced services than those available today.

The charging for today's VASs typically consists of a basic charge for the telecommunication service and a premium for the value added service. Both are usually 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, to the variety of different possible tariffs and to the ease with which a certain tariff can be changed.

The value of a particular piece of information retrieved by a user from a VAS provider at any one time may be quite small. Therefore, the use of computationally expensive payment mechanisms may not be acceptable. In addition, the scheme has to take into consideration the specific requirements of a mobile telecommunications 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, service providers and VAS providers which may have serious implications for the trust relations among them. Thus it will be increasingly important that the charging scheme is secure against cheating, and that 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.

The demonstration will show a proposed charging scheme for VASs in UMTS which satisfies the above requirements. The charging scheme, is a credit-based micropayment scheme based on Pedersen's tick payments. In the demonstrator, the value-added information offered by the VAS provider is contained in hypertext documents which can be accessed by the user using the HTTP protocol.

We assume in our model that the user has a subscription with a UMTS service provider. The charge for using a Value Added Service is composed of two parts: A basic charge for the provision of the communication link between the user and the VAS provider by the network operator and a premium for the value added. The basic charge has to be paid by the user to the network operator (through the user's UMTS service provider who need not be actively involved in the provision of the call). The premium has to be paid by the user to the VASP (through the user's UMTS service provider). So, the communication relations differ from the relations in the billing process.

The subscriber enters into contractual relationship with the UMTS service provider (SP) on behalf of the user. Any payment scheme for the protection of the basic charge has to be run between the user and the network operator (NO). Any payment scheme for the protection of the premium has to be run between the user and the VAS provider (VASP).

The fact that the network operator need not be involved in the secure billing procedure for the premium has the advantage that the implementation of security enhancements to existing Value Added Services requires no modifications to whatever network is providing the connection. The only changes which are necessary are software changes at the end-points of the communication. In this way, the solution is not restricted to UMTS, it may also be used in a GSM or DECT environment.

In our approach, the protection mechanisms for the basic charge and for the premium may be handled separately. How the approach may be used to provide secure billing also for basic services is described in section 6 below. The demonstrator is concerned only with a protection scheme for the payment of the premium.

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 (possibly through the network operator) 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.

# 6 Evaluation of the demonstrator

## 6.1 Functionality

### 6.1.1 Relation of the payment system used in ASPeCT with other payment systems

The first demonstrator provides UMTS users and value-added service providers with the security functionality required to support a micropayments protocol. The protocol provides a secure and efficient means of allowing users to pay relatively small amounts for information received from value-added service providers. The system implemented in the first demonstrator is referred to as the ASPeCT payment scheme in this document. In the first demonstrator the value-added information is in the form of World-Wide Web documents which can be retrieved by the user using the HTTP protocol. The ASPeCT payment system is particularly suited for the mobile environment.
A large number of payment systems have been proposed for use on the World-Wide Web. An overview can be found under [BoKn], [gang].
We explain in this document how the payment system implemented in the first demonstrator relates to other payment systems.

The terminology used for the roles in such systems varies. Typically three roles are distinguished:

- the buyer or customer, in our case the UMTS user: she wants to obtain a service or a piece of information or software and pay for it electronically;
- the vendor or merchant or seller, in our case the UMTS value-added service provider: he provides a service or sells a piece of information or software and wants to receive payment for it electronically;
- the broker or bank or issuer/acquirer or payment systems provider, in our case the UMTS service provider: he sets up the payment system and, in most systems, acts as an intermediary in the payment flow between the buyer and the vendor. The specific role played varies depending on the nature of the system. In some systems, such as Mondex, payment received by one party may be used to pay another party without the bank being directly involved. This is impossible in most other systems.

The roles of the parties involved in the ASPeCT payment scheme are explained in subsection 6.1.3 below.

Payment schemes proposed for the Internet can be classified according to various criteria:
**on-line vs. off-line:** An on-line system requires access to an authentication or authorisation server, typically an acquirer or a bank, for each payment. Examples of such systems are all credit-card based systems (see below) and Netbill [Netb]. In an off-line system there is no need for contacting a third party during a payment. Electronic purse systems typically fall into this category.
The system implemented in the ASPeCT first demonstrator is an off-line system. It is planned, however, to add an on-line component in the second demonstrator.
**credit-based vs. debit-based:** In credit-based systems, the user's / buyer's account with the bank is debited after he makes an electronic payment, in debit-based system, the user has to pay before he can make an electronic payment.

The ASPeCT payment scheme is a credit-based scheme. With some modifications, however, it could be implemented as a debit-based scheme.

**cryptography**: Almost all systems use cryptography in some form as it provides for higher security. **No cryptography** is used, however, in the First Virtual system [Firs]. **Secret-key cryptography** is used by most electronic purse systems, **public-key cryptography** is used in untraceable electronic cash systems or in the SET protocol. The use of public-key cryptography usually imposes a higher computation and communication load on the system. The advantage is a more flexible key management.

The ASPeCT payment scheme uses public-key cryptography. Like other micropayment systems, it is designed to minimise the negative impact of the use of public-key cryptography on the performance of the system.

**software-based systems vs. tamper-resistant hardware**: In some systems, tamper-resistant hardware, both on the buyer and on the vendor side, is indispensable for the secure functioning of the system from the point of view of all three parties involved. All electronic purse solutions fall into this category. Other systems may be securely implemented as software-only solutions. In the latter systems, the buyer may wish to have part of the system implemented on a tamper-resistant device, typically a smart card, e.g. to protect his secret key.

The ASPeCT payment scheme may be used as a software-only system. In the first demonstrator, the user has a smart card protecting the user's key and generating digital signatures. Depending on the assumptions on the trustworthiness of the user's terminal, other parts of the system may be implemented on the user's smart card.

**anonymity / untraceability**: Some systems provide strong guarantees of untraceability, i.e. Similarly to the case of real cash, the processing of the payment does not allow the tracing of information such as the identity of the payer and the amount of the payment. These systems are therefore termed electronic cash systems, cf. e.g. [ecash], [CAFE]. They typically come with a substantial overhead in terms of cryptographic computations. Credit-based systems cannot provide this kind of untraceability as the buyer has to keep an account with the broker. However, they can provide anonymity in the following sense: The buyer may remain anonymous with respect to the vendor by choosing a pseudonym (which however would be constant over a certain period of time and would therefore provide somewhat limited protection). The buyer's actions are not necessarily completely traceable by the broker in that a vendor needs to submit to the broker only the sum of the payments made by the buyer to that particular vendor. The individual items purchased by the buyer and their price need not be known to the broker.

The ASPeCT payment scheme falls into the latter category.

Payment systems may also be classified according to their analogues in the world of traditional payments systems. The categories may overlap:

**credit-card based systems**: These systems emulate the transactions made today in credit-card payments. They are on-line, credit-based and use public-key cryptography (with the exception of the First Virtual proposal). A de facto standard protocol has emerged, the SET protocol [SET]. These systems have been largely designed to support high value payments (or macropayments) in the range of tens of dollars or above. However, they are generally not suited to low value payments (or micropayments) as the use of computationally expensive cryptographic mechanisms and - above all - the on-line authorisation for each payment may not be economical.

**micropayment systems:** These systems are designed to support payments in environments were users wish to make low value payments. Recently, many micropayment protocols have

been suggested [PayW, ikP, Netc, Pede, SVP, Payt, Mill, Netb]. The first four of these rely on the same cryptographic mechanism which was first proposed for payment applications in [Pede], but was proposed for use in authentication schemes earlier in [Lamp]. There, a signature value generated using a public key operation is spread over many other cryptographic values derived by much more efficient one-way functions.
An analogue in traditional systems may be the use of stamps. 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 over a larger number of payments.
Micropayment protocols are designed to be exceptionally efficient in the sense that the cost of the mechanism is small compared to the value of the payment. The exceptional efficiency required by micropayment protocols is achieved using less computationally intensive cryptographic operations wherever possible. Concerning the other criteria mentioned above, micropayment systems may exhibit quite different features: they may be on-line or off-line or a mixture of those, they may be credit-based or debit-based (some systems may have variants so that they can be used either way), they may use secret-key cryptography or public-key cryptography, they may be software-only or necessitate tamper-resistant hardware at the buyer and the vendor. It should be remarked that micropayment systems which are on-line for each payment lose many of the advantages that they gain through the use of efficient cryptographic mechanisms. The ASPeCT payment scheme falls into the category of micropayments. The cryptographic mechanism is based on [Pede]. The approach shows similarities to [ikP] in that the signature used for commitment to the target value of the tick chain is part of a protocol for mutual authentication between buyer and vendor. In a scenario where such mutual authentication is required the signature for the payment scheme then comes at no extra cost. In the case of [ikP], the protocol for mutual authentication is ikP for credit-card transactions, in the case of ASPeCT this protocol is the same as the user-network mutual authentication protocol for UMTS proposed to ETSI [ETS1]. In this way, payment for value added service (as realised in the first demonstrator) and payment for basic telecommunications services may be efficiently combined.
**Electronic purse systems:** These are typically off-line, debit-based, secret-key based systems. They require tamper-resistant hardware on the buyer's and the vendor's side which is their major disadvantage. They provide an efficient means for low-value payments for which reason they could be considered as micropayment systems.
**Electronic cash systems:** The salient feature of these systems is untraceability. The price to pay is the lower efficiency of the mechanisms involved. Nevertheless they are also considered for low-value payments.

**Confidentiality and integrity for HTTP transactions:** The security of electronic payment systems may also be supported by protocols which provide confidentiality and integrity for HTTP transactions. In this context SSL, the Secure Sockets Layer [SSL] which has become a de facto standard and SHTTP, the Secure Hypertext Transfer Protocol [SHTT] should also be mentioned.

### 6.1.2 Mobile specific aspects of the payment system:

The discussion above applied to payment systems for the Internet or the World-Wide Web, more specifically. This is the correct class of payment systems to look at as ASPeCT assumes (quite naturally) that the protocol used to retrieve the information from the value-added service provider is the HTTP protocol and that the information is structured in a compatible way with the formats used on the World-Wide Web. But only a small subclass of these payment systems will be suitable for use in a mobile environment.

The ASPeCT solution is targeted towards a UMTS environment. This is reflected in several ways:

First of all, the selected protocols 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 light-weight. The authentication protocol is identical to the one proposed to ETSI for UMTS user-network authentication.

Secondly, the latter fact implies that payment for value-added services (as realised in the first demonstrator) and payment for basic telecommunication services (as described in section 6.4 below) may be efficiently combined by integrating the initialisation of the payment process with the call set-up procedure in UMTS. This will not be realised in ASPeCT due to capacity constraints and due to the increased complexity of the combined application of the approach to basic and value-added services. Also, the application to value-added services seems more urgent as the potential for fraud is higher there.

Thirdly, the scenario is well-suited to the mobile environment as the roles needed in the payment scheme may be played by parties already active in today's mobile networks, namely mobile users, mobile (GSM, UMTS) service providers and value-added service providers (see subsection 6.1.3). Their existing business relationships, in particular the existing infrastructure for billing users, may also be used for the new payment scheme . No additional clearing network of financial institutions like banks or credit card organisations is needed.

Fourthly, the payment scheme works off-line which means that no additional signalling load is created.

### 6.1.3 Roles in the ASPeCT payment system

The roles can be seen from the diagram below:

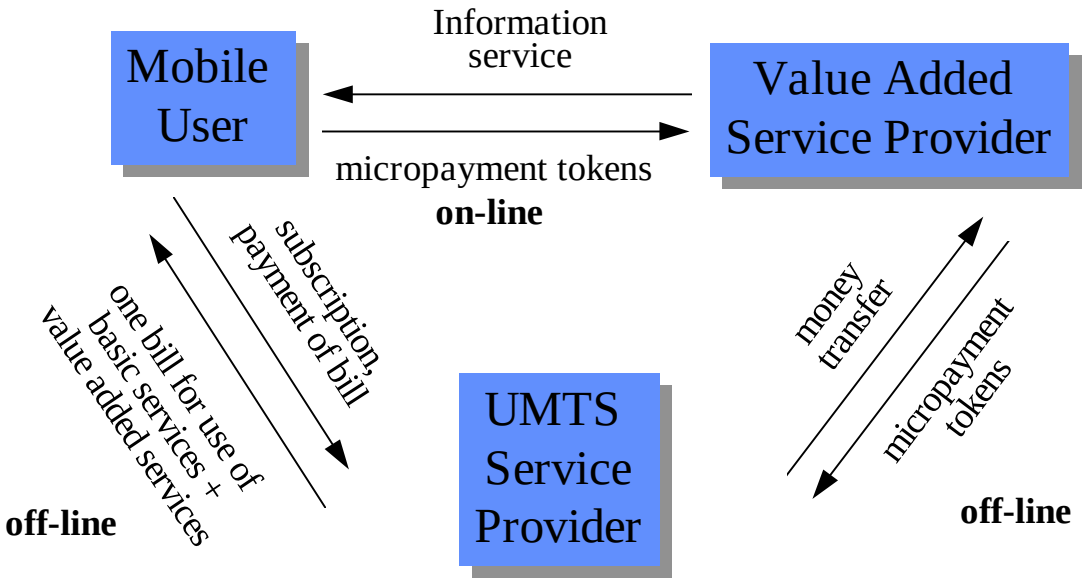



*Figure: Roles in ASPeCT payment system*

The only on-line connection is that between the mobile user (buyer) and the value-added service provider (vendor). The value-added service provider provides information to the user and sends charge requests. The user pays by sending micropayment tokens as described in the next subsection. The value-added service provider is able to check the validity of the micropayment tokens based on a certificate on the user's credentials issued by the UMTS service provider.

No previous contact between UMTS service provider and UMTS service provider is required as long as the former is satisfied by being able to successfully verify the certificate issued by the latter. However, it would be a major advantage if a previous business relationship existed as practical means for clearing the payment would then already be in place. This would typically be the case for the relationship between a UMTS service provider and an associated value-added service provider. Clearing of payments may take place periodically, e.g. daily or weekly.

The UMTS service provider plays the role of a broker: He provides the user with a means to pay electronically and vouches for the credit-worthiness of the user by issuing a certificate for him. A new certificate could be issued periodically, e.g. monthly. If a bill was not paid the old certificate would expire. The UMTS service provider bills the user and is paid by the user through established telecoms billing procedures. The UMTS service provider then forwards the due share to the value-added service provider .

It is seen as a major advantage of the integration of the implemented micropayment scheme in a telecommunications environment that the broker/banking infrastructure for billing the user and paying the vendor is already in place. This existing infrastructure can also handle the case where the call extends over several networks and hence several UMTS service providers are involved.

### 6.1.4 Overview of payment procedures realized in the first demonstrator

The secure billing demonstrator allows a user to establish a secure billing connection with a suitably enabled Web server. When the user first opens a TCP connection with the VASP, an authentication and initialisation of payment protocol is executed. As well as carrying out mutual authentication between the user and the VASP, this protocol lets the user commit to

certain number of ticks, *T*, and to a charging tariff, *ch_data*, which specifies the number of ticks that the VASP will charge for each byte of data it sends to the user.
The user commits to *T* ticks by signing a target value $a_T$ which is computed by carrying out *T* applications of a one-way function on a randomly chosen starting value α. The user also commits to the tariff by signing the string *ch_data*. The VASP can verify these commitments using the user's public key. The user now possesses a chain of one-way values $a_j$, or "ticks", which he can release successively, starting at the target value, in order to make payments. Tick payments can be verified by the VASP by applying a one-way function operation to each successive payment to show that it belongs to the chain associated with the digitally signed target value. The string *ch_data* can also be verified by the VASP to show that the user has agreed to the tariff.
The fact that the one-way function cannot be inverted means that the VASP cannot make money by generating a value $a_j$ which the user did not send.
The user's signature implicitly contains the identities of both user and VASP as well as fresh random data generated by both sides. This is important to prevent double spending (see below).
The VASP has to store only the user's signature and the tick $a_j$ with the lowest index. After the security session has been closed the VASP may submit $a_j$ together with the user's signature to the UMTS service provider for clearance. In practice, the VASP will wait a specified minimum period to collect more payment tokens from the user to benefit from reduced clearance costs by aggregating a number of user payments.
The UMTS service provider will check the user's signature and apply a one-way function to check the compatibility of $a_j$ with the $a_T$ in the signature in the same way as done by the VASP . In addition, in order to avoid double spending the UMTS service provider checks whether the same signature has been submitted on a previous occasion. If this is the case then he refuses payment. To be able to do this the UMTS service provider has to store records on the details of previous payments, at least for the period of validity of the user's signature key. (In fact the UMTS service provider will store those records for some time after the bill was sent out to the user so as to make the resolution of disputes possible.) If the checks are passed then the UMTS service provider will pay the VASP the equivalent of T-j ticks.
Procedures conducted at the end of the UMTS service provider are beyond the scope of ASPeCT WP 2.5.

In the following subsections of section 6.1 we address a number of issues which show the potential applications of the first demonstrator to a variety of scenarios and which may lead to possible enhancements for the second demonstrator.

### 6.1.5 Compatibility

The security enhanced Web clients and servers should be compliant with existing clients and servers such that users using the enhanced browser can still access existing servers, and users who access the enhanced server using existing browsers can still access free documents. The first demonstrator does not yet provide this functionality.
However, compatibility can be achieved with the current approach. Compatibility should be considered for realisation in the second demonstrator. The Graphical User Interface should provide informative dialog boxes to the user, where appropriate.

### 6.1.6 On-line use of Trusted Third Parties

The payment scheme implemented in the first demonstrator is an off-line scheme. This is quite appropriate for low value payments. There is no need for the VASP to perform an on-line credit check (e.g. with the user's UMTS service provider). Nevertheless, depending on the security policy, the VASP may want to perform such a credit check on certain occasions, e.g. when the sum of unredeemed payments exceeds a certain threshold.

The user and the VASP are issued certificates on their public keys. The provision of certificates, provided through a TTP infrastructure, provides for a scaleable solution, where users can make payments to a large number of different VASPs without having to establish a security association with each one in advance.
The user's certificate does not only permit the VASP to authenticate the user. It also serves as off-line authorisation of the user: It gives the VASP assurance that the UMTS service provider will redeem valid payment tokens sent by the user to the VASP. Note that certificates may become invalid or revoked for a number of reasons: The user is no longer given credit by the UMTS service provider, the private key of the user is compromised, the smart card of the user was stolen. To check for revoked certificates a VASP may periodically (e.g. daily) download revocation lists. This limits the period during which damage can be done using a revoked certificate. If the security level provided by this mechanisms is not considered sufficient then on-line access to a certificate server or TTP is required.

In the first demonstrator it is assumed that the necessary security information, including certificates on their own public keys, is distributed to the user and VASP in advance. They exchange their certificates in the course of the protocol. There is no way of checking whether a certificate has been revoked.
For the reasons mentioned above, it would be desirable to realise an on-line access to a certificate server in the second demonstrator.

TTPs could also support the escrow of a secret key, shared between the user and the VASP. This secret key could be used to support confidentiality of ASPeCT security protocol elements (e.g. user identity or charging data) exchanged between the user and the VASP. To provide confidentiality for HTTP transactions the de facto standard SSL (Secure Socket Layer) protocol, would be more appropriate.

The functions and protocols to be used are yet to be determined.

### 6.1.7 Charging principles

In the future, flexible charging may be based on various parameters, e.g. time, data volume or application specific parameters such as a URL (Uniform Resource Locator) specifying the location of a World-Wide Web page.
In the first demonstrator only one charging principle was implemented to demonstrate the idea: The user agrees a volume-based tariff when he first opens a TCP connection with the VASP. This tariff specifies the charge which will be made based on the volume of data sent over the TCP connection.
It would be desirable to associate payments with application level events which are under direct control of the user, and vary these payments according to the URL which is being

requested. In this way the user has more visibility over the payments which he makes. Charging for application level events has the advantage that the user can potentially know exactly what charges will be incurred for the actions which he takes.
An enhancement to the second demonstration and trial may be to assign charges to application level events such as requesting a particular URL. Of course such a scheme could make use of an underlying volume-based tariff, which changes depending on the URL and the expected amount of data being requested. However, this volume-based tariff need not be visible to the user.

### 6.1.8 Resubmitting requests

This subsection is closely related to the previous one. HTTP is a stateless protocol, so the server does not know if a resource has been successfully transferred to the client. As such it is necessary for the client to resubmit requests if a document has not been received correctly due to network partitions, for example. In the demonstration the user may pay for the same information twice if he resubmits a request.
A charging scheme based on URLs - as sketched in the previous section - will solve this problem by ensuring that if the URL has been previously accessed by the user, the server will not make a charge for downloading the information again. However, for some URL requests, particularly those which involve database queries, charges may be applied irrespective of whether the request has been previously submitted by that user.

### 6.1.9 Incremental payment for Web resources

In the demonstration there is a policy option for the VASP to ask for payments for information sent by the VASP to be incrementally released during transfer. This has the effect of reducing the value of individual payments, thereby reducing both the likelihood of fraud and the loss incurred due to fraud. Incremental payments are an improvement on schemes where only one payment is made, either before or after transfer. For instance, in scenarios where the user aborts transfer to obtain the most valuable information at the start of a document, the VASP still recovers some of the value of the information in proportion to the volume sent by the VASP. The VASP can also identify such a situation as potentially fraudulent, although it could have arisen due to random events such as network partitions, rather than malicious intent on the user's part. As such the VASP would only confirm this type of fraud if its occurrence was non random.
On the other hand, the VASP has also the option to sell each piece of information associated with a given URL as a complete item, and ask for one payment for this item. However, the price of the payment should be sufficiently low so that a loss could be tolerated - in line with the general idea behind micropayments. The user either receives and pays for the whole piece of information, or he received nothing and pays nothing. The appropriate time of payment is discussed in the next subsection.

### 6.1.10 Time of payment

Payment for a service can be made either before or after service utilisation. The demonstrator uses a post-payment model, where payments are released after a certain volume of information has been sent by the VASP. It would be easy, however, to configure the demonstrator in such a way that it realises a pre-payment model. Depending on which model is used, the risk of having to bear the cost of fraud lies with either the user or the VASP.

In the post-payment model the risk lies with the VASP, since the user could commit fraud by refusing to make a payment after the VASP has sent information, claiming that he did not receive the information. However, such fraud can be detected easily by the VASP, who can subsequently prevent the user from accessing services. Thus, as the value of individual payments decreases, it becomes increasingly unlikely that the user will risk committing fraud.
In the pre-payment model, if the user denies receiving the piece of information he ordered and paid for, then the VASP could resend it. The risk lies with the user here as the VASP could simply refuse to send the requested information or could send something different .

A solution where the risk for both the user and the VASP is eliminated cannot be provided by an off-line scheme. An on-line trusted intermediary is needed for this purpose (cf. e.g. the Netbill approach [Netb]).
However the basic ideology of micropayment schemes implicitly suggests that neither party should be overly concerned about the loss of a single payment, just as you do not worry about "the loss of a nickel in a candy machine". Systematic losses would be detected, and no more business would be done with the corresponding user or VASP.

### 6.1.11 Payment demands

In the demonstration the user only makes payments when he receives a demand from the VASP. It could be argued that sending payment demands is an unnecessary communications overhead. Instead, charging conditions laid out by the VASP during payment initialisation could specify when payments are due.
Sending a demand does have the advantage that the VASP can control exactly when payments should be made.
Which option will be chosen in the second demonstrator is a matter for further study.

### 6.1.12 Representation of ticks

In the demonstrator it is deliberately left open whether the tick payments represent real monetary value or whether they just represent acknowledgements that a chargeable event has occurred. It is felt that the decision about how to use the system should be left to those who operate it. In this way, the system is more flexible.
If each tick released represents the same monetary value, then the loss due to fraud is constant. However, if the ticks just represent acknowledgement of a chargeable event, then the loss due to fraud is variable. To limit the risk, there should be a specified maximum for the value of one tick.

### 6.1.13 Optimisation of payment parameters

Increased flexibility in the use of the payment scheme could be achieved by adapting the use of some of the payment parameters. Some of the parameters which could be adapted would include:

- ***tick chain length***
  In the demonstration the length of the tick payment chain $T$ is fixed at $T = 2^{10}$. It would be desirable if the user could generate a variable length tick chain. If this approach was adopted then the length of the tick payment chain could not be assumed to be a system wide

parameter; it will change during each payment initialisation. However, for security reasons the admissible maximum value of *T* has to be a system-wide parameter. As such both the user and the VASP need to know the value of *T*.
This could be easily integrated in the current protocol.
The option could also be given to set the value of *T* to one, such that the user could make secure macropayments to the VASP. For macropayments the commitment would specify the amount to be paid rather than a chain of ticks. However, macropayments can also be made in the existing implementation by paying with many ticks at once.

- ***charging data field***
  In the demonstration the field specifies that each tick represents a chargeable event, namely the receipt by the user of a certain amount of data. More specifically, the charging data in the demonstration specifies a volume-based tariff which states that *x* ticks will be charged for each byte of data sent to the user.
  In the second demonstrator, the charging field could be used to provide more flexibility in the charging options. E.g. the charging data field could be used to indicate the monetary value represented by each tick. In this case, the charging data field could also contain currency information which allows the user and VASP to use different currencies.

### 6.1.14 Multiple tick chains

During initialisation of payment, the user in the demonstrator only generates one chain of tick payments for making payments to the VASP.
There are a number of possible variations of this basic idea which may be used to increase the efficiency or the flexibility of the scheme (see also the next subsection). The following two cases are examples where it might be useful to generate more than one chain of tick payments.
Firstly, if the ticks represent monetary value, then different chains could be used to represent different denominations. A scheme with more than one denomination will have increased flexibility. Whether it will be more efficient depends on the user's spending behaviour: it will be useful if the range of individual payment values made by the user is large, on the other hand it will be less efficient if the user always pays one single tick. The efficiency will also depend on the number of chains and the length of each chain (or the number and division of denominations). It should also be noted that separate signatures are required for each denomination, thus the cost of generating the extra signatures must not exceed the saving made through the use of the extra denomination.
Secondly, if the ticks represent confirmation that the user has received a certain service, then different ticks could be used to confirm different chargeable attributes. For example, the user might generate two tick chains, one for time, and another for volume. If the tariff is based on both time and volume, then the user can release a tick payment from one chain for each second of service utilisation, and a tick from the other chain for each byte of data sent / received.
Note that if a user has committed to more than one tick chain with a particular vendor, then the payment should specify which chain is relevant.

### 6.1.15 Multiple vendors

In the demonstration a user initialises the payment system with a particular VASP in order to obtain a chain of ticks which he can release successively in order to make credit-based payments to that provider.

In this model the user must generate and sign separate chains of ticks for each vendor he wants to make payments to. This requires a separate signature operation for each vendor.
A related scheme is "Paytree" [payt]. In this scheme more than one vendor could be paid based on a single commitment or signature. There are several variants of the "Paytree" idea. Some of them combine paytrees with paywords (i.e. tick payment chains) which seems quite attractive. Further study is needed, however.

## 6.2 Performance

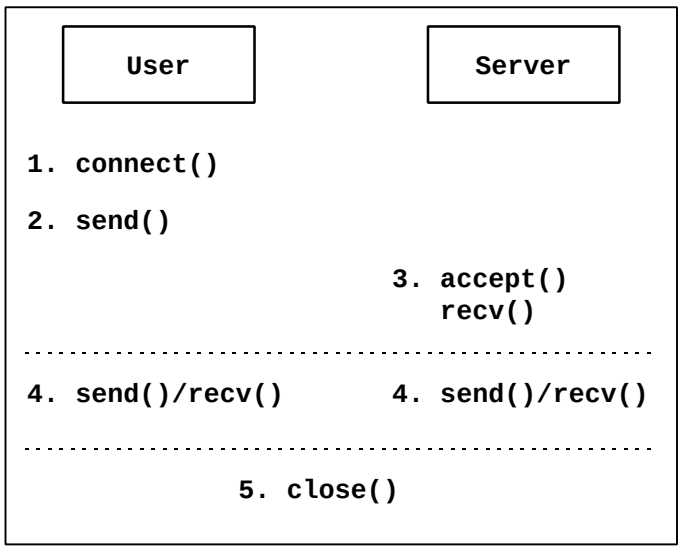
### 6.2.1 Introduction

In the secure billing demonstrator a mobile user retrieves information from a server over a mobile telecommunication system. The impact of the security measures on the information service is analysed in this section.

### 6.2.2 Information service without secure billing

The information service without secure billing is described in the following scheme:

0. The server waits for connection requests
1. A user initiates the connection request to the server
2. The user gets the notification that the server is waiting for requests
   Now the user can start sending application protocol requests to the server
3. The server is informed that a connection request is pending and accepts it
   Now the server can receive application protocol requests from the user
4. The user sends requests to the server and receives responses
   The server receives requests from the user and sends responses
5. User or server closes the connection



The first table shows the timing (in hours:minutes:seconds.milliseconds) for various requests for a small amount of data (only one send/receive call for information data):

| | 400 bytes | 783 bytes | 814 bytes | 455 bytes | 455 bytes |
|---|---|---|---|---|---|
| 1. connect() | 10:51:32.390 | 10:51:38.980 | 10:51:53.100 | 10:52:03.700 | 10:52:59.230 |
| 2. send() | 10:51:36.730 | 10:51:40.360 | 10:51:54.420 | 10:52:05.020 | 10:53:00.550 |
| 3. accept()/recv() | 10:51:38.710 | 10:51:42.280 | 10:51:55.680 | 10:52:06.230 | 10:53:01.810 |
| 4. send()/recv() | 1/1 | 1/1 | 1/1 | 1/1 | 1/1 |
| 5. close | 10:51:38.930 | 10:51:42.880 | 10:51:57.000 | 10:52:07.160 | 10:53:02.800 |
| total time | 6.540 | 3.900 | 3.900 | 3.460 | 3.570 |

The second table shows the timing for various requests for larger amounts of data:

| | 26354 bytes | 28902 bytes |
|---|---|---|
| 1. connect() | 10:52:07.380 | 10:53:02.960 |
| 2. send() | 10:52:08.700 | 10:53:04.390 |
| 3. accept()/recv() | 10:52:09.910 | 10:53:06.260 |
| 4. send()/recv() | 7/19 | 8/21 |
| 5. close | 10:52:51.920 | 10:53:43.940 |
| total time | 44.540 | 40.980 |
| connection | 2.530 | 3.300 |
| data transfer | 42.010 | 37.680 |

The number of receive calls does not match the number of send calls because the number of data packets to be transferred over the network can be changed by intermediate devices. In this table the total time is split up into the time for the connection set-up and the time for the data transfer (which was negligible for the table with the small amounts of data).

### 6.2.3 Information service with secure billing

The timing is changed when the secure billing service is used because some synchronisation points are introduced and the communication between the user and the server is blocked if certain conditions are not fulfilled. To a lesser extent the information service is slowed down because user/server resources are busy processing the secure billing protocol and so they are not always available for the information service.
The information service with secure billing is described in the following scheme:

0. The server waits for connection requests
1. A user initiates the connection request to the server
   **This request triggers the secure billing software to initiate a connection request to the secure billing software on the server side (a)**
2. The user gets the notification that the server is waiting for requests
   Now the user can start sending application protocol requests to the server
3. The server is informed that a connection request is pending and accepts it
   Now the server can receive application protocol requests from the user
4. The user sends requests to the server and receives responses
   The server receives requests from the user and sends responses
   **If the secure connection between the user and the server is not yet established, the secure billing software on the server side prevents information data from being sent to the user (a)**
   **Whenever the server has sent a response containing chargeable data the secure billing software on the server side sends a payment request to the secure billing software on the user side and if the charge is correct the user side sends a payment response (b)**
   **As long as the server waits for a payment no information data is sent to the user and**

**if the payment is not received within a certain time the connection between user and server is closed (c)**

5. User or server closes the connection

The first table shows the timing (in hours:minutes:seconds.milliseconds) for various requests for a small amount of data (only one send/receive call for information data):

| | 400 bytes | 783 bytes | 814 bytes | 455 bytes | 455 bytes |
|---|---|---|---|---|---|
| 1. connect() | 10:57:08.920 | 10:57:19.080 | 10:57:28.860 | 10:57:38.470 | 10:59:44.740 |
| 2. send() | 10:57:11.230 | 10:57:20.230 | 10:57:30.180 | 10:57:39.790 | 10:59:46.060 |
| 3. accept()/recv() | 10:57:12.320 | 10:57:21.660 | 10:57:32.320 | 10:57:41.220 | 10:59:47.380 |
| 4. send()/recv() | 1/1 | 1/1 | 1/1 | 1/1 | 1/1 |
| 5. close | 10:57:19.030 | 10:57:22.600 | 10:57:32.810 | 10:57:41.980 | 10:59:48.260 |
| total time | 10.110 | 3.520 | 3.950 | 3.510 | 3.520 |

The second table shows the timing for various requests for larger amounts of data:

| | 26354 bytes | 28902 bytes |
|---|---|---|
| 1. connect() | 10:57:42.420 | 10:59:48.590 |
| 2. send() | 10:57:43.800 | 10:59:50.070 |
| 3. accept()/recv() | 10:57:45.880 | 10:59:52.100 |
| 4. send()/recv() | 7/20 | 8/22 |
| 5. close | 10:58:39.050 | 11:00:48.020 |
| total time | 56.630 | 59.430 |
| connection | 3.460 | 3.510 |
| data transfer | 53.170 | 55.920 |

When the user connects to the server, the authentication and initialisation of payment protocol for the secure billing is executed. This happened for the request for the 400 bytes in the first table. The authentication protocol was started at 10:57:10.400 and the secure connection was established at 10:57:18.590, which is a period of 8.190 seconds. However this protocol runs partly in parallel with the information service, so only a part of this time really delayed the information service. Compared to the first request for the information service without secure billing there is a difference of only 3.570 seconds. For the next requests in the first table the secure connection is already established. For every request in the first table the actual charge protocol can be ignored, as it did not delay the information service.
For the requests in the second table there is a delay caused by the secure billing protocols. The data transfer for the information service is accomplished with several data packets to be sent over the network and between two successive packets a charge protocol run is executed and delays the sending of the second packet. The third table shows the timing for some charge protocol runs:

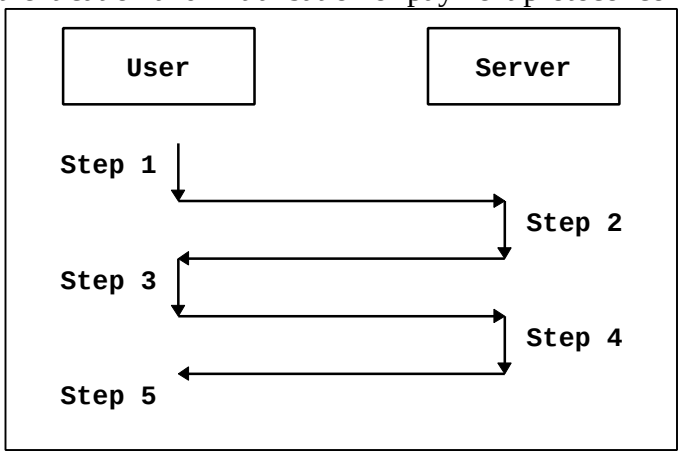
| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| start server | 10:57:45.940 | 10:57:54.620 | 10:58:01.100 | 10:58:15.980 | 10:58:22.790 |
| start user | 10:57:52.480 | 10:57:59.340 | 10:58:06.590 | 10:58:20.820 | 10:58:27.630 |
| end server | 10:57:54.560 | 10:58:01.100 | 10:58:15.980 | 10:58:22.790 | 10:58:29.330 |
| total time | 8.620 | 6.480 | 14.880 | 6.810 | 6.540 |
| real delay | 2.080 | 1.760 | 9.390 | 1.970 | 1.700 |

The real delay is the difference between the time when the user starts the charge protocol and the time when the server accepts the payment. When the server starts the charge protocol, the

user has not yet received the data the server charges. Only when the user has received this data does the charge protocol continue.

### 6.2.4 Performance of the secure billing protocols

The authentication and initialisation of payment protocol consists of several steps:



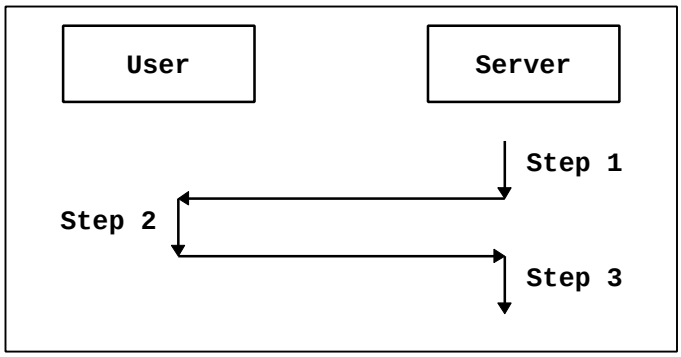

The first table shows the timing for these steps:

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| start | 10:57:10.400 | 10:57:13.200 | 10:57:12.980 | 10:57:18.640 | 10:57:18.590 |
| end | 10:57:11.230 | 10:57:13.530 | 10:57:16.660 | 10:57:19.140 | |
| total time | 0.830 | 0.330 | 3.680 | 0.500 | |

There is user input in step 3, which took 0.880 seconds. Excluding this user input the accumulated time for all the steps is 4.460 seconds, while the whole time for the protocol is 7.310 seconds. Step 3 includes the generation of a digital signature with a smart card including the initialisation of the smart card terminal. The signature generation on the smart card takes about 0.990 seconds and including the handling of the smart card terminal it takes 2.480 seconds.

During the data transfer from the server to the user one run of the charge protocol is executed for every packet of data sent. The protocol timing was already analysed in the former section, the time for the steps necessary at the user and at the server is negligible compared to the total protocol time. The total time for the (three) steps of the charge protocol is in the range between 0.050 seconds and 0.250 seconds.

When the amount of ticks agreed on in the authentication and initialisation of payment protocol is spent, the re-initialisation of payment protocol is executed. This protocol consists of only three steps:




| | 1 | 2 | 3 |
|---|---|---|---|
| start | 11:00:29.890 | 11:00:34.890 | 11:00:39.610 |
| end | 11:00:29.890 | 11:00:37.640 | 11:00:40.110 |
| total time | 0.000 | 2.750 | 0.500 |

The accumulated time for all the steps is 3.250 seconds, while the whole time for the protocol is 10.220 seconds. The protocol is much faster than the authentication and initialisation of payment protocol, but the timing shown here was influenced by some other network activities.

### 6.2.5 Conclusion

The impact of the secure billing protocol on the performance of the information service is within an acceptable range.
However there are two ways to improve the overall performance :

1. To improve the performance of the smart card handling and signature generation
2. To change the billing strategy of the server side in the following way:
   - when a packet of data is sent, ask for the payment (as before)
   - when **two** packets of data are not yet paid, block the next (the **third**) packet
   In this way the payment protocol steps would not be synchronisation points for the data flow of the information service.

## 6.3 Security aspects and smart cards

### 6.3.1 Security features provided by the demonstrator

In this subsection we discuss the security features provided by the security protocols implemented in the demonstrator. (The treatment is not intended to be rigorous). The benefits of the secure environment provided by the smart card we use are discussed further below.

There are three cryptographic protocols:

1. the authentication and initialisation of payment protocol;
2. the charge ticks protocol.
3. the re-initialisation of payment protocol;

The first protocol provides:

- mutual authentication of user and service provider;
- agreement of a session key;
- confidentiality of the user identity with respect to third parties;

- exchange of public key certificates (version B);
- non-repudiation by the user of charging related data sent by the user to the service provider.

An important feature in the context of billing is the non-repudiation of charging related data. In conjunction with the charge ticks protocol it provides **incontestable charging**: The user cannot deny later that he was the one who sent the payment tokens for the service use in question. (To be more precise: that the user's smart card was involved in the generation of the payment tokens.) On the other hand, the service provider cannot later claim that the user used the service and sent corresponding payment tokens if in fact he did not. The characterisation of the service used may be included in the charging related data. In the demonstrator we limit ourselves to characterise the service used by the time it was requested. Additionally, one could include the address of a subtree in a server in the charging related data to characterise the service used, provided the same tariff applies to the documents in the subtree.

The second protocol provides for repeated payments whose validity can be checked by the service provider by securely linking them to the non-repudiable charging related data received in a previous run of the first protocol.

The third protocol may simply be seen as an abbreviated version of the first protocol in that it provides a new non-repudiable set of charging related data in an efficient way, whenever necessary (i.e. when the user has run out of payment tokens).

A remark regarding the session key agreed in the first protocol is appropriate here:
In the demonstration the session key is only used to provide confidentiality of the user identity. The session key may also be used to provide confidentiality and/or integrity for all the messages exchanged in the security protocol, in particular for the charging related data which are not protected in our demonstration.
If the user has to pay before receiving the service (the requested document) then integrity protection of requests can prevent the subtle type of fraud (whose relevance will depend on the precise circumstances) described in [ikP, 4.4]. In the ASPeCT demonstrator the user pays after receiving the service (cf. section 6.1.8 above).
In addition, the session key may be useful if a brokerage TTP is introduced (cf. [ikP, 5]).

### 6.3.2 Security level

The security level is scaleable by adapting the lengths of the cryptographic parameters. The currently implemented security level is characterised by the use of single DES for encryption and random number generation, RIPEMD-128 for hashing and 128-bit elliptic curves for key agreement and signatures. This is still sufficient today for the kind of low-value transactions for which the payment scheme is intended. It will be inadequate in a few years time. The system is designed in such a way that the algorithms can be easily replaced. (e.g. RIPEMD-128 by RIPEMD-160 etc.)
(A fine point regarding the signature system: For the signature system to be replaced with a signature system providing message recovery, the third message of the protocol would need to be encrypted to protect the user's identity, cf. [ETS1]. However, the choice of such a signature system would make the protocol less efficient. Nevertheless, if it is a requirement that the signature system be replaced then the third message of the protocol could be encrypted .)

### 6.3.3 Security analysis

A formally rigorous analysis of the security of a system such as our demonstrator is infeasible. However, there are a number of ways of increasing the confidence in the security of the system:
Firstly, the algorithms we use have all been discussed in the literature for some time and are believed to be secure by the cryptographic community.
Secondly, the authentication protocol has been formally analysed by means of the automated tool AUTLOG [franc] employing a logic of authentication which is a variant of the BAN logic. The analysis was presented to ETSI SMG in [ETS3].
Thirdly, within the project a thorough analysis was undertaken of the security of tick payments, the use of one-way functions for tick payments and the appropriated choice of algorithms and parameters. In particular, an upper bound for the maximum number of ticks which can be guaranteed by a single signature has been determined. It was also found that the use of hash functions, as opposed to mere one-way functions, was not required.
Fourthly, what may further increase the confidence in the security of the tick payment scheme which we implemented is the fact that four research groups have proposed a cryptographic mechanism for micropayments which is identical to the one used in the tick payment scheme, three of these groups apparently independently [PayW, Netc, ikP, Pede].

### 6.3.4 Smart cards

Smart cards - or more generally tamper-resistant hardware - may be used in electronic payment systems for two different reasons:

- to provide protection to the participants in the system against a user (customer) or a service provider (merchant) tampering with their own terminals to commit fraud at the expense of other participants. This is the case e.g. in electronic purse payment systems.
- to provide protection to a participant in the system against a third party tampering with the participant's terminal to commit fraud at the expense of this participant. This is the case e.g. for the credit-based tick payments scheme implemented in the ASPeCT demonstrator.

This implies that the ASPeCT system could be securely run as a pure software system if the environment of the user's terminal was deemed secure. This, however, may not be assumed in general. The security of the protocol relies on a trustworthy implementation, which protects certain cryptographic information. In order to achieve a higher degree of security, parts of the security functionality should be implemented on a separate trusted, tamper-resistant security module or smart card.
The first demonstrator features a smart card which serves as a key storage and signature generator.
It is for further study what the function split between the terminal and the smart card should be for the second demonstrator. A shift of additional functions may increase the security of the system. It is clear, however, that only a smart card with an interface to the human user (at least a rudimentary display and keyboard) can provide complete security if the terminal cannot be trusted.

## 6.4 Applicability

The ASPeCT micropayments protocol provides a secure and efficient means of allowing users to make a series of low value payments to a vendor. In the first demonstration this vendor is a value-added service provider, who is paid by the user for providing him with Web documents. However, the scheme could also be applied more generally to other payments in UMTS. In this section, we consider the application of the mechanism to the payment of basic telecommunication services.

We consider how this approach may be used to help provide an accurate and incontestable charging scheme for basic telecommunication services in UMTS. Other approaches to the provision of incontestable charging services have been investigated. These include a scheme which involves the generation of digitally signed charge tokens which are released by users during a chargeable service utilisation [Puet].

### 6.4.1 Charging schemes for basic telecommunication services in UMTS

An effective charging scheme for basic telecommunication services is an essential requirement in operating a network, not only for increasing revenue, but as a method of introducing feedback and control. In UMTS charging is likely to play an important role in helping to optimise the use of bandwidth on the air interface. For instance, users may be encouraged to shape their traffic, such that the overall performance of the radio subsystem is enhanced. After each user has minimised their own charges[1], the radio subsystem should be left operating at an efficient point.

Tariffs should reflect each user's relative network usage. Thus, the effective network usage of a bearer service must be measured as part of the charging procedure. A tariff is then applied to the measurements in order to calculate the charge due. In UMTS a wide range of attributes may be used to measure the effective network usage of a basic service. These attributes include:

- connection mode (connection / connectionless)
- connection end-point identifiers[2]
- symmetric attribute (unidirectional / bi-directional symmetric / bi-directional asymmetric)
- communication configuration (point-to-point / point-to-multipoint / broadcast)
- peak bit rate
- mean bit rate
- delay variation tolerance
- maximum transfer delay
- bit error rate
- error characteristics (uniform / bursty)

If the tariff is based on many of these attributes, then complex network measurements and calculations may need to be carried out, thus increasing the cost of implementing the charging scheme. Complex tariffs may also appear confusing to the user. However, it is often acceptable to assume that tariffs are based on two relatively simple measurements: the volume of information transferred and the duration of the service.

---

[1] which may be done automatically by the user's terminal

[2] This would include A-number and B-number for a regular telephony call. In this way charges based on distance and network interconnection could be applied.

Note that the tariff may be based on other factors apart from effective network usage. Some of the other factors that may be taken into consideration include:

- commercial/marketing issues (subscriptions, discounts, inclusive call charges, minimum call charges, unitisation, etc.)
- charges which are dependant on B-number (free calls, premium rate calls, etc.)
- the effect of other users (e.g. congestion)

The requirements for security and charging in UMTS are being studied by ETSI SMG [ETS2] while the GSM MoU are considering the network operator's requirements on charging and billing in third generation systems, including UMTS [MoU].

### 6.4.2 Secure charging for basic services

The use of more complex charging schemes in UMTS will increase the requirement for incontestable charging, whereby parties involved in the charging scheme should have the assurance that justified claims relating to charges can be proved and that unjustified claims cannot be made. For instance, from the user's point of view, it is more difficult to judge whether charges made are justified, when tariffs involving both volume and duration measurements are used. As a result the user may need to be assured of the charges by trusting a secure payment scheme implemented in his mobile terminal (or more specifically in his UIM or smart card).

In second generation mobile telecommunication systems charges for basic services are based on measurements made by the network operator. Thus, the user must trust the network operator to generate accurate and reliable bills. A more acceptable approach for third generation systems such as UMTS may be to involve the user in the generation of charging information.

The requirements for generating accurate and incontestable charging information in UMTS are detailed in a draft ETSI SMG Technical Report on security and charging [ETS2]. These requirements indicate that certain information, which may be used to calculate the charge, should be provided to the network operator by the user. The requirements also indicate that the source and integrity of this information should be verifiable by both the network operator and the service provider. Typically this information will include measurements which are used as a basis for calculating the charge.

The charging information generated by the user may consist of tokens which represent real monetary value, or it may just contain an acknowledgement that a chargeable event has occurred. In order to prevent fraudulent abuse, the value of individual acknowledgements or payments must be kept small. Thus, the user may have to make a series of incremental acknowledgements or payments during a particular service utilisation, rather than a single relatively large acknowledgement or payment at the start / end of service utilisation.

The use of incremental acknowledgements or payments has the effect of reducing both the likelihood of fraud and the loss incurred due to fraud. Depending on whether a pre-payment or post-payment model is adopted, the risk of having to bear the cost of fraud will rest with one of the two parties[3]. In the post-payment model, the network operator accepts the risk. For instance, the user may fail to send an acknowledgement or payment by maliciously terminating the service before sending the token and could claim that the disconnection was due to a network problem.

[3] see Section 6.1.10

The ASPeCT micropayments protocol could provide a secure and efficient means of allowing users to pay for basic telecommunication services received from network operators. The protocol was designed to be exceptionally efficient such that the cost of the mechanism is small compared to the value of individual payments.
The protocol could allow a user to establish a secure billing relationship with a network operator. For example, when the user requests service from the network operator, an initialisation of payment protocol could be executed as part of the user-network authentication procedure. This could allow the user to commit a chain consisting of a certain number of ticks, *T*, and a charging tariff, *ch_data*, which would specify the number of ticks that the network will charge for each chargeable event during service utilisation.
During service utilisation the user would release the ticks from the chain as appropriate to either acknowledge or pay for each chargeable event as it occurs. Using this scheme the network operator, or indeed any other party with an authentic copy of the user's public signature verification key, would be able to verify each acknowledgement or payment. Thus, the scheme provides an incontestable charging service.

### 6.4.3 Application of micropayments to basic telecommunication services in UMTS

When applied to the payment of basic service, ticks could be used in different ways depending on the charging scheme. For a simple tariff based on duration, a tick could be released every second during service utilisation. For more complex tariffs, ticks could be released according to a combination of the volume of information transferred and the duration. In the more complex case separate tick chains could be used to confirm each separate chargeable attribute. For example, ticks from one chain could be released to pay for each second of service utilisation, while ticks from another chain could be used to pay for each byte of data sent / received[4].
The initialisation of tick chains could be incorporated into the authentication protocol which is run between the user and the network operator. Indeed, payment initialisation in the first demonstrator has been incorporated into a protocol which was proposed for UMTS user-network mutual authentication [ETS1].
Payment initialisation relies on the ability of the user to generate digital signatures on information and for the network operator to be able to verify these signatures. Thus it is envisaged that tick payment schemes could be incorporated more seamlessly into authentication protocols which are based on public key techniques, rather than those based on secret key techniques. Note that, in the ASPeCT protocol, the payment information to be digitally signed is concatenated with authentication information. This means that the payment signature and verification are combined and so separate cryptographic operations are not required.
Although payment initialisation could be integrated into call establishment procedures, the actual payments must be transferred during service utilisation. Implementation in UMTS would require payments (and optionally demands for payment) to be sent in an appropriate signalling channel. The requirements on signalling load will depend on the rate of payments, and the size of each individual payment. Both of these parameters will depend of the type of service being paid for and the level of security required.
During payment initialisation, the user commits to a finite number of ticks. Thus the user may need to reinitialise the payment scheme during service utilisation if he runs out of ticks. Since

[4] see Section 6.1.14.1

reinitialisation involves generating another digital signature, it would be desirable if reinitialisation during a service utilisation was avoided. However, as the number of ticks in each commitment increases, the effort to compute and/or store the tick chain also increases. Therefore, in practice, the length of the tick chain generated will be a trade off between this additional effort required, and the cost of carrying out reinitialisation during service utilisation. There is a maximum length for the tick chain for security reasons also, but this maximum length will be sufficient for practical purposes.
In the credit-based approach the network will present information, collected as part of the payment scheme, in a bill to the user via his service provider. This bill will contain the user's identification, the signed commitment, the last tick payment received from the user, and the number of ticks consumed by the user from the commitment. It will be possible for the authenticity and integrity of the bill to be verified by the service provider, the user or an arbitrator, who may become involved in the case of a dispute.
In the case where charging is shared between more than one of the participating parties it should be possible for any of the parties to be able to generate and send ticks during service utilisation. For example, it should be possible both the called and the calling party in a basic telephony call to be able to generate and send ticks to their respective network operators in the case that the charge for the call is split between both parties.

### 6.4.4 Alternative payment models

In the above we assume a credit-based model, where users receive credit from service providers and use this credit with network operators for the provision of basic telecommunication services. In this model, the network operators aggregate the credit-based payments made by users, and bill the service providers in order to recover the credit. The service providers then bill the users to clear their debits.
It is likely that other methods of payment will exist in UMTS. These may include pre-payment with no subscription, or debit with an electronic purse. It is possible that a micropayments scheme could be used in both of these schemes. It is believed that the tick payment scheme could be easily modified to be used in a debit-based scheme. However, the exact protocols and mechanisms required are for further study.

### 6.4.5 Payment for connectionless bearer services

The application of micropayments detailed in Section 6.4.3 assumes a connection orientated service, where there is an establishment and release phase, and where information is guaranteed to be delivered in the same order that it was sent. However, UMTS bearer services may involve packet based connectionless transmission, where information can be transmitted directly with no guarantee of ordered delivery and without the need for an establishment phase at the start of transmission or a release phase at the end of transmission.
In packet based services it may not be acceptable to simply carry out authentication between the user and the network at the start of the service. Instead, individual packets may need to be authenticated. The simplest way of providing packet authentication would be to generate and append a message authentication code (MAC) to each packet using a secret key shared between the user and the network. However, although this allows the recipient to verify the authenticity of both the source and the contents of a packet, it does not allow anyone else to verify the authenticity of the packet. Thus, this mechanism cannot be used to provide an incontestable charging service. For an incontestable charging service the network needs to

generate evidence of receiving packets from users which can be verified by the user, the user's service provider or an arbitrator.
Incontestable charging could be achieved using a public key digital signature algorithm. In this way every originator could digitally sign each packet before it is transmitted, and every recipient could verify the signature before accepting and acknowledging the packet. The digital signatures could then be presented in a bill to the user's service provider. Using this mechanism the user, the user's service provider or an arbitrator could verify the bill, providing that they have an authentic copy of the user's public key.
Implementing such a mechanism would involve a considerable amount of cryptographic processing to be carried out for every packet sent or received by any given node. Moreover, the bill would consist of every signed packet sent by the user! A more efficient approach may be to introduce ticks, which could be used to spread one digital signature value across many one-way values.
To implement such a scheme the user could generate a chain of ticks before transmitting the first packet to the network. The user digitally signs the chain of ticks and sends it to the network as part of the first packet. Then, for subsequent packets, the user appends ticks from the chain to either acknowledge or pay for the transfer of each packet. Using this scheme the network operator, or indeed any other party with an authentic copy of the user's public signature verification key, would be able to verify each acknowledgement or payment. Thus, the scheme would provide an incontestable charging service.
Note however that the one-way values cannot be used to authenticate each packet. Instead, packet authentication could be achieved by appending a MAC to each packet. The MAC would be calculated on the packet contents, after the one-way value had been added.
Note also that depending on the security policy, micropayments need not be appended to each individual packet. Instead, it may be sufficient for the user to periodically send a number of ticks to the network operator, corresponding to the number of packets sent or received since the last payment.

### 6.5 Architecture

For the architecture of the secure billing demonstrator there were two choices, to integrate it with the application software or to develop an additional security layer between the application and the communication stack. (For other security services - e.g. integrity checks - there is also the possibility to integrate it with a layer in the communication stack. This did not seem reasonable for secure billing) For the existing demonstrator the solution with an additional layer between the application and the communication stack was chosen because of the main advantage that the application software need not be changed. The demonstrator uses a standard Web client and a standard Web server without any enhancements. Furthermore the secure billing software realised in the first demonstrator is not even restricted to the World-Wide Web, as the application protocol (HTTP) is not evaluated. With the existing software one could also realise secure billing for other services, e.g. for ftp-services.
For an integrated solution there are two main arguments:
1. An integrated solution would use only one communication channel
2. Payments may be associated with application events rather than underlying transport events

For a transport service which is circuit-switched the first argument might be important but for a packet based service the number of simultaneous communication channels between two entities should not have any consequences as long as the bandwidth of the underlying transport medium is sufficient.

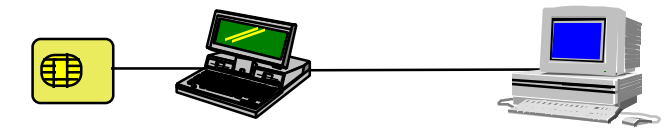
The payment is of course always associated with transport events because the user pays for the information he retrieves from the server or for the time he is connected to the server and this is always apparent in the transport layer. If the payment for the retrieved information depends on the information in a way which is only visible in the application protocol, then the security layer must interpret the application protocol in the transport data packets being sent/received to determine the correct payment. This approach will be investigated in the second demonstrator.

### 6.6 Test environment

The objective of the secure billing activities in ASPeCT is to develop and validate a new approach to securely paying for services delivered over a mobile network. The demonstrator shows the implementation of a payment scheme for mobile value added information services. These were chosen because, on the one hand, the risk of fraud is particularly high for value added services already today and, on the other hand, the availability of such services based on hypertext and graphics is expected to become a driving force for UMTS because they cannot be offered by current mobile systems in a satisfactory way.

As UMTS is not yet available, this implies, however, that the environment in which our payment can be demonstrated is not optimally suited for the type of service for which the payment system is intended. This raises the question what we can learn from our demonstrators developed in a number of test environments for the use of the system in the environment, namely UMTS, in which it will be eventually used.

The first demonstrator comes in two versions which differ in the communication infrastructure they use (cf. [D10]):
In both versions there is a laptop computer representing the user and a desktop computer representing the Value Added Service provider.
In the first version, the two computers are connected via Ethernet. This is shown in the next figure:



☐ ***Smart Card*** ☐ ***Laptop PC*** ☐ ***Desktop PC***

Mobile User Value Added Service Provider

*Figure - Physical configuration of version 1*

In the second version, the laptop computer is connected via a modem card to a GSM terminal. The desktop computer is connected to the Siemens intranet. The GSM terminal connects to a gateway to the Siemens intranet via a GSM data service. This is shown in the next figure:

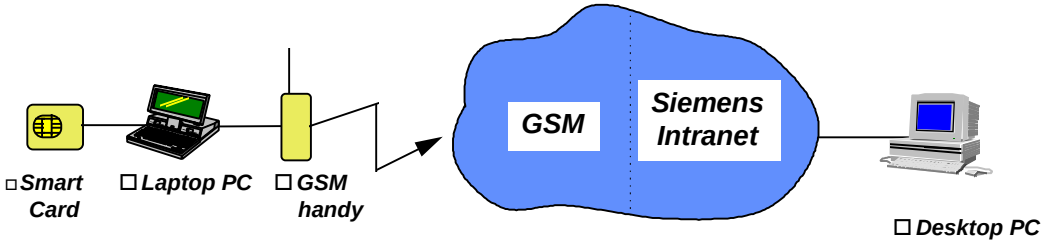



*Figure - Physical configuration of version 2*

The second demonstrator will be able to use the UMTS testbed provided by EXODUS which will alleviate some of the restrictions imposed by the two environments in which the first demonstrator is shown.

But nevertheless, the two versions of the first demonstrator already provided very valuable information:
The Ethernet environment allowed us to verify that the software developed by ASPeCT works and that the integrated demonstrator provides the specified functionality. In particular, the correct functioning of the communication between the two PCs could be shown. The Ethernet environment is also very well suited for the type of service we demonstrate as it provides high-bandwidth, low-latency, low-error rate connectivity. It proved also useful for measuring the execution times of the ASPeCT-developed software.
The GSM environment, on the other hand, is more realistic for the demonstration of mobile services, of course, but it is not ideally suited for the type of service as the bit rate it provides is too low. In addition, it is circuit-switched while a packet radio service would be better suited for the type of application, especially from a billing point of view. Nevertheless, the demonstration works fine also in this environment provided that the files retrieved are not too large.
The UMTS testbed for the second demonstrator will provide DECT access on the mobile side thus providing a bit rate of up to 32 kbit/s which will considerably reduce the bandwidth restriction experienced with GSM. The trial environment for our demonstrator will then be quite close to a real UMTS environment.

## 6.7 User-friendliness

### 6.7.1 Quality of Service

#### *6.7.1.1 General*

Among many definitions, it can be generally assumed that the Quality of Service is determined by the user's perception on the degree by which the service meets, or surpasses, the need it is designed for. It may also be defined by the level of overall user satisfaction regarding the provided service.

QoS is defined in [ITU1] as follows: "The collective effect of service performance which determine the degree of satisfaction of a user of the service". This definition of QoS is a wide

one, encompassing many areas. The QoS parameters, as user satisfaction are subjective in nature, depending on individual perception and expectations.

As derived from [ITU2], the user-oriented QoS parameters provide a valuable framework for design, but they are not necessarily usable in specifying performance requirements. Similarly, the performance parameters ultimately determine the user-observed QoS but they do not necessarily describe that quality in a way that is meaningful to users.

#### *6.7.1.2 User-perceived QoS*

The parameters related to the QoS of the demonstration, from the user's viewpoint can be:

- **usability of the security features**
- **acceptability of the security features**
- **user-perceived stability of the demonstration**
- **user-perceived performance of the demonstrated protocols**
- **user friendliness of the GUI / GUI operability**
- **overall user satisfaction of the demonstration**

### 6.7.2 Achieved goals

#### *6.7.2.1 Usability of the security features*

The security functions are triggered when the user commences the communication. He is not involved in the protocols execution. He is simply informed about them by the GUI.

#### *6.7.2.2 Acceptability of the security features*

The existence of the security features is a very positive feature for the user because it fulfils the need for secure and incontestable charging for value-added services. The degree of acceptability naturally depends on the impact of the security functions on the transaction speed.

#### *6.7.2.3 User-perceived stability of the demonstration*

The demonstration is quite flexible regarding intentional or unintentional misuse by the users.

#### *6.7.2.4 User-perceived performance of the demonstrated protocols*

There is no user-perceived impact on the transaction speed or in the general communication performance induced by the security layer.

#### *6.7.2.5 User friendliness of the GUI / GUI operability*

The GUI operability can be defined as its ability to be successfully and easily operated by a user. This goal is fully achieved in the integrated demonstration, where the protocols are executed automatically, while the user can be informed about the security-related message exchange by the tracer windows.

#### *6.7.2.6 Overall user satisfaction of the demonstration*

The user impression is that the demonstration succeeded in achieving its goal: to present the proposed security function and prove its efficiency in a friendly and perceivable way.

## 6.8 Appearance of demonstration

### 6.8.1 First demonstration's Graphical User Interface

The first WP 2.5 demonstration presents a procedure for establishing security and integrity of billing, in the transfer of value-added information from a VAS provider to a User. The network operator, that provides the communication link in real environments, only collects the basic charge and is not involved in the secure billing procedure for the premium charge that corresponds to the VAS. Thus, two entities are involved, the User and the VASP. The User has a web client that retrieves World-Wide web pages from a VAS provider. Three protocols are demonstrated:

- **authentication and initialisation of payment**: the two entities make sure of each other's identity, agree on the payment parameters and commit themselves to them
- **data transfer and charging**: after the successful completion of the initial protocol, the web server commences the transfer of data, as long as their value does not exceed the amount previously agreed
- **re-initialisation of payment protocol**: this protocol runs if a new commitment needs to be made by the User, regarding the value of a certain piece of information

The protocols execution is triggered by accessing the VASP's web page, through the web browser window. The procedure can be viewed through two "tracer" windows that display the messages sent and received by the two interacting entities.
This is the minimum amount of information that the demonstration user needs to know, in order to understand the interactions he sees on the screen:

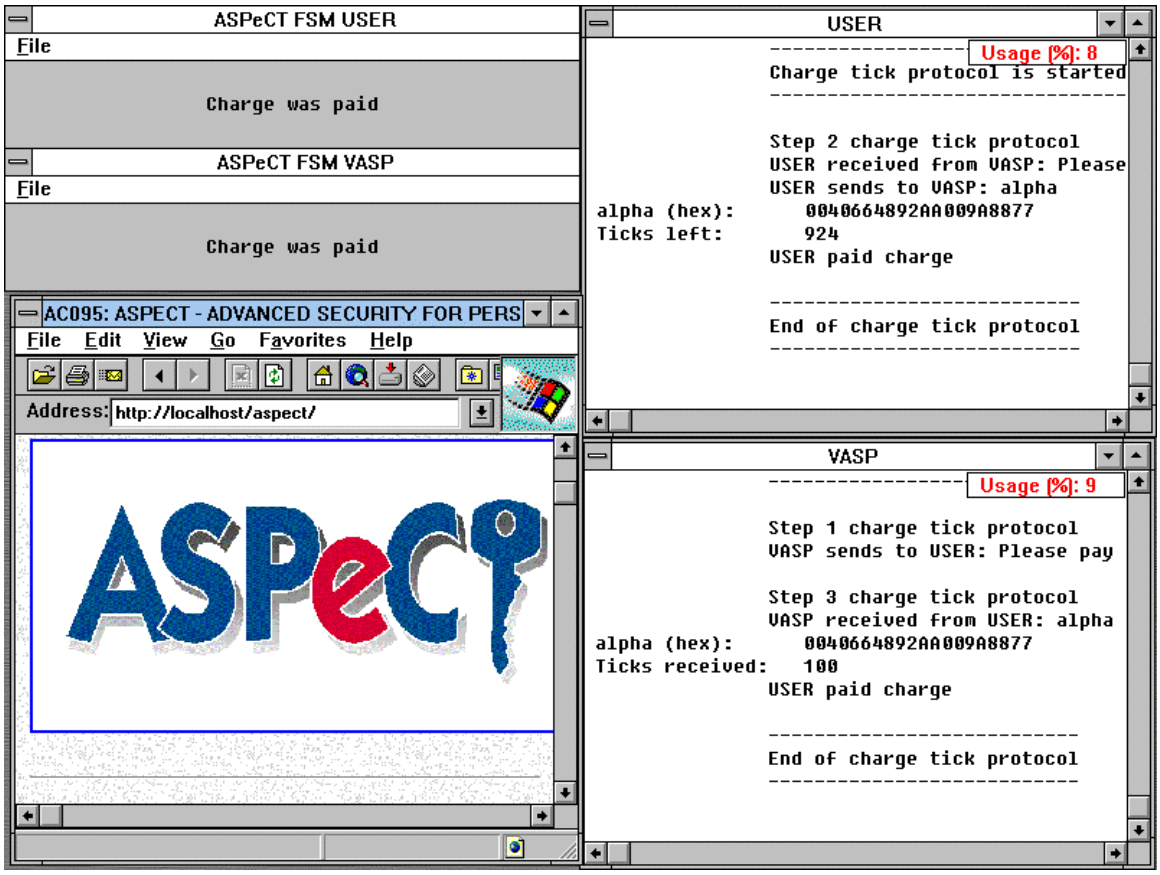



*Figure 1 : An example screen of the secure billing first demonstration*

Additionally, the user who wishes to involve himself more in the demonstration, may try a number of available options, as far as the configuration and the protocols execution are concerned.

At first, he has the option to run the applications in one or two PCs.

There is also the possibility to observe the process step-by-step, in order to have an elaborate perception of the protocols, instead of selecting the overall view of the execution of the secure billing scenario without his interference.

In the former option, of the **menu driven demonstration**, the user executes a number of commands, acting as the User and the VASP sequentially, from the menu of the corresponding screens, "ASPeCT FSM USER" and "ASPeCT FSM VASP".

The latter option is the **integrated demonstration**, where the user needs only to contact the web server. The authentication and initialisation phase then starts, followed by the billing protocol. The user can verify the correct execution of the protocols from both sides, afterwards, through the examination of the tracer windows. This option provides the possibility to estimate the efficiency of the executed protocols, by observing potential delays introduced by the security functions.

Throughout all interactions, the observer will be able to monitor the message flows using the tracers. The tracers provide a broad analysis of the execution of the protocols and confirm the accuracy of the charging scheme implementation. There are two tracers, one for the User, one for the VASP, in order to present more convincing evidence of the protocol execution and facilitate the viewers. Both contain detailed information about the parameters exchanged and the amount charged for the value-added service, while the protocols are running.

In addition, at any time, the demonstration user may:

- re-initialise the payment protocol (only from the VASP side)
- change the (symbolic) name of the VASP (only from the User side)
- configure and resize the windows of the web server, browser, User tracer and VASP tracer
- close the connection between the two entities (from either side)

### 6.8.2 The observer's view

#### *6.8.2.1 Menu driven demonstration*

The objective of the menu driven demonstration is to enable better understanding of the security protocols running and their relations, while the demonstration user executes them one-by-one through the GUI. Possible error cases may likewise be tested, in order to evaluate the demonstrator's robustness. For example, the user may choose to execute the protocols in the reverse order. Additionally, he may refuse to commit to a payment and watch the secure billing application suspend the data transfer.
The tracer windows provide detailed information of the security process. They present all the protocol parameters and their values, including headers that indicate which protocol is running, key values, signature checking. They also present indicative messages in error situations, caused by incorrect user actions. Thus, the GUI provides information to the demonstration user on the impact of his actions to the secure billing exchange.

#### *6.8.2.2 Integrated demonstration*

The protocols execution is now triggered by using the web browser, there is no need for the user to execute them explicitly. The user is only prompted to decide whether he agrees or not on the charging rate of the payment protocol, through the relevant dialog box. If he does not agree, the authentication procedure stops and the connection closes. In that case, the User tracer indicates that the charging rate was not accepted and subsequently, a message rejecting the authentication was sent.
The tracer windows provide detailed information of the security process. They present all the protocol parameters and their values, including headers that indicate which protocol is running, key values, signature checking.

### 6.8.3 Suggestions for enhancement

The current implementation requires a minimum of related background from the demonstration user. The reason for that is that the information provided by the tracer is mainly addressed to the "qualified" users who are already aware of the protocol flow and its parameters. In that respect, the GUI could provide some additional information to users outside of ASPeCT. For example an indicative Monitor window could be introduced to provide visual representation of the message exchange, addressed to the less involved users, while the Tracer window displays specific information, intended for users more familiar with the protocols and the scope of the demonstration.
The tracers should also be enhanced to support options, for example to save or print the presented messages, so that they can be viewed off-line.

Messages could be displayed when an improper selection is made. For example, the user may not execute the protocols in the right order and expect, to a point, to be guided back to the correct sequence by the GUI, through explanatory error messages.

The user should be aided by the Graphical User Interface to access all information relevant for him with respect to the payment system used. In particular, it should be possible for him to monitor how much was spent with which value-added service provider.
This information is currently only available in the tracer which is used for test purposes is not suited for the information of the user, or it is not available at all.
On the other hand, care should be taken that the user is not overburdened with information he does not want or need. To give an example: On the user side, a decision needs to be taken on whether or not user's smart card is to sign a new set of charging data whenever the charging scheme is initialised. The user may be given two options: In the first option, a window is popping up each time and the (human) user is asked for explicit confirmation. This is what is realised in the first demonstrator. In the second option, the user's charging policy (e.g. price limits) may be stored in the user's terminal or smart card and the decision may be taken automatically. The user would only be prompted in cases not covered by the stored policy.
It is desirable that the second demonstrator contains sufficient options for the user to access the information he needs - and only that information.

# 7 Summary of suggested enhancements

In this section we summarise the areas in which new functions could be defined to enhance the first demonstrator.

Please note that this is no requirement specification of the second demonstrator. A decision about which functions will be implemented in the second demonstrator can only be taken after the feasibility of the realisation of the proposals and - very importantly - the effort required have been studied in depth.

The following additional functionality has been suggested for further study:

- **compatibility**
  existing browser should be able to communicate with ASPeCT servers and vice versa;

- **on-line use of Trusted Third Parties**
  an on-line component should be added to access a certificate server in real-time so as to be able to authorise a user on-line if required by the security policy;

- **additional functionality to increase of the flexibility of the charging model**
  this includes introduction of url-based charging and the flexible handling of resubmitted requests;

- **optimisation of payment and protocol parameters;**

- **investigation of further enhancements to the tick payment scheme;**

- **further elaboration of the Graphical User Interface.**

# 8 References

[BoKn] A. Bosselaers, L. Knudsen: Electronic Payment Systems: Properties and Options, ASPeCT/DOC/KUL/017/WP25/A, Mar 1996.

[CAFE] J.-P- Boly et.al., The ESPRIT project CAFE: High Security Digital Payment Systems, ESORICS '94, LNCS 875, Springer Verlag, Berlin 1994, pp 217-238.

[D10] ASPeCT D10, Secure Billing: First Implementation, AC095/SAG/W25/DS/P/10/1, February 1997.

[ecas] ecash home page, http://www.digicash.com/ecash/ecash-home.html

[ETS1] ETSI/STC/SMG SG DOC 73/95, A public key based protocol for UMTS providing mutual authentication and key agreement, Siemens AG, 1995.

[ETS2] ETSI TS UMTS 22.15, Special Mobile Group : Universal Mobile Telecommunications System (UMTS) ; Service aspects ; Security and charging, Version 1.0.0, April 1997.

[ETS3] ETSI/STC/SMG SG DOC /95, A formal analysis of the Public-key protocol for UMTS proposed by Siemens. Siemens AG, Sep 1995.

[Firs] First Virtual Holdings Inc., http://www.fv.com/.

[franc] V. Kessler, G. Wedel, AUTLOG - An advanced logic of Authentication. Proc. IEEE Computer Security Foundations Workshop, Franconia 1994, pp90-99.

[gang] http://ganges.cs.tcd.ie/mepeirce/Project/oninternet.html

[ikP] R. Hauser, M. Steiner, M. Waidner, Micro-payments based on iKP, Proc. Securicom Conference, Paris, 1996.

[ITU1] ITU-T Recommendation E.800 (04/94), "Terms and definitions related to quality of service and network performance including dependability"

[ITU2] ITU-T Recommendation I.350 (03/93), "General aspects of quality of service and network performance in digital networks, including ISDNs"

[Lamp] L. Lamport, Password Authentication with insecure communication. Communications of the ACM, 24 (11), 770-771, Nov 1981.

[Mill] The Millicent protocol for inexpensive electronic commerce, http://www.research.digital.com:80/SRC/millicent/

[MoU] GSM MoU PRD TG.24, Requirements for charging, billing, accounting, tariffing, Version 0.1.0, 3 January 1997.

[Netb] The Netbill electronic commerce project, http://www.ini.cmu/NETBILL/home.html

[Netc] Anderson R et al, NetCard - A practical electronic cash system. http://www.cl.cam.ac.uk:80/users/rja14/

[Payt] Jutla C S, Yung M, Paytree : "Amortized-signature" for flexible micropayments, Proceedings of Second USENIX Association Workshop on Electronic Commerce, pp 213-221.

[PayW] Rivest R L, Shamir A, PayWord and MicroMint: Two simple micropayment schemes, Cryptobytes, vol 2 no 1 pp7-11, 1996. http://theory.lcs.mit.edu/~rivest/

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

[Puet] Pütz S, Secure billing - incontestable charging, Proceeding of the Second IFIP TC6/TC11 International Conference on Communications and Multimedia

Security, September 23-24 1996, Essen, Germany; pp 208-221, ISBN 0-412-78120-4.

[SET] Secure Electronic Transactions (SET). http://www.mastercard.com/set

[SHTT] The Secure Hypertext Transfer Protocol, July 1995. http://www.eit.com/creations/s-http/draft-ieft-wts-shttp-00.txt

[SSL] The Secure Sockets Layer (SSL) Protocol, Version 1.0. http://home.netscape.com/newsref/std/

[SVP] Stern J, Vaudenay S, SVP a flexible micropayment scheme, To appear in the proceedings of the Financial Cryptography '97 Conference. http://www.dmi.ens.fr/~vaudenay/

[Tan1] Tang L, A set of protocols for micropayments in distributed systems, Proceedings of First USENIX workshop on electronic commerce, July 1995.

[Tan2] Tang L, Low S, Chrg-http: A tool for micropayments on the World Wide Web, Proceedings of Sixth USENIX Security Symposium, pp123-129.