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

Deliverable Number	AC095/SAG/W25/DS/P/10/1
Title of Deliverable	**D10 - Secure billing: First Implementation**
Nature of the Deliverable	Prototype
Contributing WPs	WP2.5
Contractual Date of Delivery	February 1997 (Y2M12)
Actual Date of Delivery	28 February 1997

Document Title	**Secure billing: First Implementation**
Document reference	ASPeCT/DOC/SAG/WP25/048/C
Issue status	Issue: final
Date of Completion	28 February 1997
Editor	Günther Horn

Abstract	This document provides a report on Deliverable D10 - the first implementation of the ASPeCT secure billing service.
Keywords	ACTS, ASPeCT, Value Added Services, security, billing

Table of Contents

1 Executive Summary

Deliverable D10 was produced by ASPeCT WP2.5. WP2.5 is concerned with the specification and implementation of procedures realizing a secure billing service for UMTS. Deliverable D10 is the first prototype implementation of such a secure billing service. This document is a **report on** the completion of **D10**, (i.e., it does **not constitute** deliverable D10). Please also note section 2 on limitations and restrictions pertaining to this deliverable.

The implementation of the first prototype follows the specification given in ASPeCT deliverable D07. The reader will notice a certain overlap of the material in this report on D10 with the material contained in D07. This overlap is intended for several reasons:

- This report on D10 specifies choices of options left open in D07.
- It provides more detail on the demonstration.
- It presents a concise summary of the specificiation of the first secure billing prototype.

The scenario for which our secure billing service is demonstrated is the following: We consider a mobile user using value-added information services in UMTS. The value-added service used in the demonstration is a service whereby a UMTS user can retrieve information from a remote server maintained by a value-added service provider. This remote information server is based on WorldWideWeb technology. The user has a Web client which can be used to browse information on the remote server. The information is given in hypertext format.

The ASPeCT contribution to this scenario is an ASPeCT-defined security protocol layer residing between the Web application and the communication stack which are both left unchanged. The security layer realizes mutual authentication and an incontestable charging (secure billing) service whereby neither the UMTS user nor the VASP can later deny that a specified amount of data was transferred at a specified charging rate. The service is based on a so-called micropayment scheme.

As a UMTS network is not yet available it has to be substituted by other appropriate means of communication. In the prototype delivered as D10 by the end of February 1997 the means of communication is an Ethernet link between the two computers representing the mobile user and the value-added service provider. In a second step, this link will be replaced by a GSM data connection. Finally, in the second demonstrator and trial due by the end of February 1998 the means of communication will be a UMTS testbed provided by the cooperating project EXODUS.

2 Important Note on Limitations and Restrictions

The public nature of this deliverable is restricted to the demonstration itself.

No general rights to the specifications or to the programmes and the libraries which constitute the demonstrator are given or implied.

Certain components may be

- proprietary,
- subject to non-disclosure agreements,
- claimed and acknowledged as background material,
- subject to governmental controls on export or re-export.

Specific enquiries or requests for clarification may be addressed, in the first instance, to the editor.

3 Document Control

3.1 Document History

Version A	4-2-97
Version B	21-2-97
Version C (final version)	27-2-97

3.2 Changes Forecast

Deliverable sent to EC	28-2-97

Any errors will be corrected. Necessary or desirable enhancement will be identified in the evaluation report D16.

3.3 Change Control

In conformance with the ASPeCT Quality Plan.

3.4 Document Cross References

[D07] Deliverable D07: Security Services: First Specification, Ref: AC095/RHUL/W23/DS/I/07/1

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

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

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

4 Abbreviation and Glossary of Terms

API	Applications Programming Interface
ASPeCT	Advanced Security for Personal Communications Technologies
CA	Certification Authority: an officially authorized or otherwise recognized issuer of certificates and certified key-material
Certificate	a collection of unforgeable information, *signed* by a *CA*, conveying trustworthy 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 Organization
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

5 Introduction

The remainder of this report on deliverable D10 - the first prototype implementation of a secure billing service for UMTS - is organized as follows:

Section 6 provides an **overview and background information** on the secure billing demonstrator. It details the scenario for which the secure billing service is demonstrated and explains assumptions made regarding the future development of technology and services. The relations of the entities involved in communications and in the billing process are described.

Section 7 is very technical. It gives a detailed **specification of the three security protocols** which in combination realize the secure billing service, namely

- the authentication and initialization of payment protocol;
- the re-initialization of payment protocol and
- the charge ticks protocol.

The non-security expert may want to skip that section - perhaps with the exception of the overview provided in subsection 7.1.

Section 8 describes the **architecture** of the secure billing demonstrator. It begins with a subsection on the protocol architecture. It is explained there how security functionality can be added without having to modify neither the application on top of the security layer nor the communication layer below. The second subsection describes the main functional modules which constitute the security layer. The third and fourth subsections describe the SW and HW platforms used.

The final section 9 is a **description of the demonstration**. After a short introduction it focusses on the demonstration described from an observer's point of view. In particular, it describes the Graphical User Interface and the interaction which is possible between the (human) user and the system.

6 Secure billing demonstrator - background and overview

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

Some of these advanced security features to be provided in UMTS will be made possible by the use of more powerful smart card technology not yet available for second generation systems. This technology - together with the use of suitably adapted security mechanisms - will make the use of so-called public-key techniques in mobile systems possible for the first time. Trusted Third Parties (TTPs) will provide the infrastructure for the use of these techniques. TTPs are dealt with in Work Package WP2.3 of ASPeCT. (For the description of the first TTP demonstrator see ASPeCT deliverable D09.) The services which TTPs provide are - among others - certification of public keys and support of key management for end-to-end security. These TTP services will be used in the first and in the second secure billing demonstrators respectively. TTPs also enable the provision of non-repudiation services based on digital signatures, opening the possibility of secure payment over future mobile networks. In the following, we describe how the first secure billing demonstrator realizes a payment system for value added services which makes use of the non-repudiation service.

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

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

included graphics, so, instead of numbers giving the prices of stock he may view charts of stock indices, instead of textual information on the nearest hotels he may view a street map of his surroundings indicating the location of the hotels, or he may view a coverage map of the mobile operator in whose domain he is roaming.
The charging for today’s VASs consists of a basic charge for the basic service and a premium for the added value. Both are based on the duration of the call. In the future, due to the greater variety of services offered more *flexible charging schemes* for the premium would be desirable. Flexibility relates to the parameters which determine the charge (in addition to the duration of the call, the charge may depend e.g. on the amount of data transferred, different tariffs may apply for different information items), to the variety of different possible tariffs and to the ease with which a certain tariff can be changed.

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

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

ASPeCT will demonstrate a proposed charging scheme for VASs in UMTS which satisfies the above requirements. The charging scheme which will be implemented in the first demonstrator is a credit-based payment scheme using *micropayments* according to Pedersen [Ped]. (It should be noted, though, that the same micropayment scheme may also be used on a pre-paid basis.) A similar scheme was recently published by Rivest and Shamir [Riv]. In the demonstrator, the user - acting as a Web client - will be able to retrieve WorldWideWeb pages from a VAS provider - acting as a Web server - over a mobile link.

We assume in our model that the user has a subscription with a UMTS service provider. 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 and the network operator). So, the communication relations differ from the relations in the billing process. In the following figure, we show the communication relations:

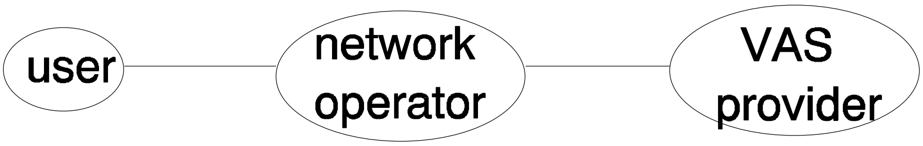


Figure 6.1 - Communication relations

The following figure depicts the relations among the participants in the billing process. The arrows show who pays whom. The subscriber enters into contractual relationship with the UMTS service provider (SP) on behalf of the user.

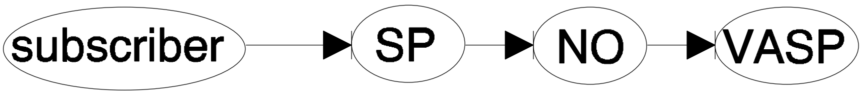


Figure 6.2 - Billing relations

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 SW 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. Here, we deal 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.

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

7 Protocols

7.1 Overview

The charging consists of two phases: In the *initialization phase*, the user and the VAS provider authenticate each other, and the user commits himself to a starting value for the micropayment scheme and a certain tariff by performing a digital signature on corresponding data. The authentication protocol is identical with one submitted to ETSI SMG for UMTS user-to-network authentication [ETS]. The starting value is the n-th iterate of a one-way function applied to a random value chosen by the user.

There are two versions of the authentication protocol, called versions A and B. For version A it is assumed that the involved entities are already in possession of the public keys of their communication partners. In version B, these public keys are provided as part of the protocol by the exchange of certificates.

In the *data transfer phase*, the user pays by releasing the pre-images of the starting value, so-called “ticks“ which represent unit charges. The value of one unit charge is agreed on in the initialization phase. The “ticks“ serve as proof to the VAS provider that the user incurred certain charges because only the user could have generated them. They are presented by the VAS provider to the user’s UMTS service provider (via the network operator) to clear the charges. The particular efficiency of the scheme stems from the fact that the user may commit himself to a large number of payments of unit charges with only one signature. Images of one-way functions are much less expensive to compute and to transmit than signatures.

7.2 Detailed description of the protocols

In the deliverable D07 [D07], a high level specification of the protocols was given where some options were left undefined and some questions were left open. This section gives a more detailed and complete specification of the protocol as implemented in the demonstrator.

7.2.1 Prerequisites on mechanisms and choice of cryptographic parameters

The protocols are executed between a user U and a VASP V. There is only one user and only one VASP.

For the **authentication and initialization of payment protocol**, we have the following prerequisites and choice of cryptographic parameters:

∗ There is an elliptic curve cryptosystem *E* over *GF(P)* whose parameters *P* (prime defining the field), *q* (size of the curve), g_x and g_y (coordinates of a generator g of the curve), a and b (coefficients of the defining equation) are configurable. As default values, the values from ISO CD14888-3, Annex C.2 are taken.

- There is a function f mapping E onto the numbers in the range [0..q-1]. It is $f(Z) = Z_x \bmod q$.
- V has secret and public key agreement keys v and g^v respectively where v is a number in the range [0..q-1] and gÎ E is as above.
- U possesses an asymmetric signature system with secret signature transformation Sig_U, secret key KU^- and public key KU^+. It is an AMV signature system based on the above elliptic curve E as described in ISO CD14888-3. $Sig_U(M)$ denotes only the appendix.
- U and V possess a symmetric encryption function where $\{ M \}_K$ is the encryption of message M with key K. This function is DES-CBC.
- U and V possess a random number generator. This RNG uses DES-OFB.
- U and V possess one-way function h2 and hash functions h1 and h3. The choices are as follows:
  h1 = h3 = RIPEMD-128, h2(x) = trunc(40, h3(x)) where trunc(n,y) returns the n least significant bits of y.

For **protocol version A** only: (This is the one implemented in the first demonstrator.)

- An authentic copy of the public key KU^+ of the asymmetric signature system of U is available at V. (It may have been distributed to V in an earlier run of protocol variant B.)
- An authentic copy of the public key agreement key g^v of V is available at U. (It may have been distributed to U in an earlier run of protocol variant B.)

For **protocol version B** only:

- There is a valid certificate *certU*, issued by a certification authority CA1, on the public key KA^+ of the asymmetric signature system of U, available at U.
- There is a valid certificate *certV*, issued by a certification authority CA2, on the public key agreement key g^v of V, available at V.
  CA1 and CA2 may coincide.
- U possesses the public key necessary to verify certificates issued by CA2.
- V possesses the public key necessary to verify certificates issued by CA1.

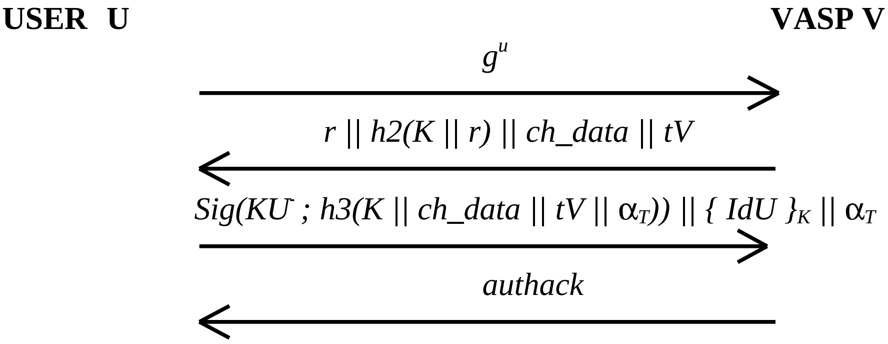
For the **payment protocols**, we have the following additional prerequisites:

- There is a public system parameter T which gives the maximum number of ticks to which the user can commit himself by one signature. T is at most equal to 2^{16}. For performance reasons, T may be set to a lower value. Default: $T = 2^{10}$.
- There is a length-preserving one-way function $F: \{0,1\}^n$ ® $\{0,1\}^n$, where n is a public system parameter. The choice for the first demonstrator is $n = 64$ and $F(x) = trunc(64, h(x))$ where h equals RIPEMD-128.

7.2.2 Authentication and initialization of payment protocol

For the sake of simplicity, only the information flow in the error-free case is shown below.

Protocol variant A:


USER U
VASP V
g^u
$r \,||\, h2(K \,||\, r) \,||\, ch\_data \,||\, tV$
$Sig(KU^- ; h3(K \,||\, ch\_data \,||\, tV \,||\, \alpha_T)) \,||\, \{ IdU \}_K \,||\, \alpha_T$
authack

Detailed description of protocol variant A:

operations at U:

1. generate random number u
2. compute g^u
3. **send message** U -> V: authreq: g^u

operations at V:

4. generate random number r
5. compute K := h1(f($(g^u)^v$)||r)
6. compute h2(K, r)
7. send message V -> U: authcont: r|| h2(K, r) || ch_data|| tV
   / ch_data contains information on the tariff to be applied. tV is the UTC time of V. */*

operations at U:

8. compute K := *h1(f($(g^v)^u$)||r)*
9. compute h2(K, r)
10. display ch_data on the screen in separate window, display warning on the screen if the difference tV-tU (tU = UTC time of U) is greater than a pre-defined delta_t. authcont_check = O.K. if received h2(K, r) equals h2(K, r) computed in step 9 and if ch_data is confirmed by human user via GUI (can be switched off).
11. compute $\{ IdU \}_K$, generate random a_0 and compute $a_T = F^T(a_0)$.
    */*IdU = 64 bit string*/*
12. compute Sig(KU^-; h3(K|| ch_data|| tV|| α_T))

13. **send message** U -> V: authresp: Sig(KU$^-$; h3(K|| ch_data|| tV|| α_T)) || { IdU }$_K$ || α_T

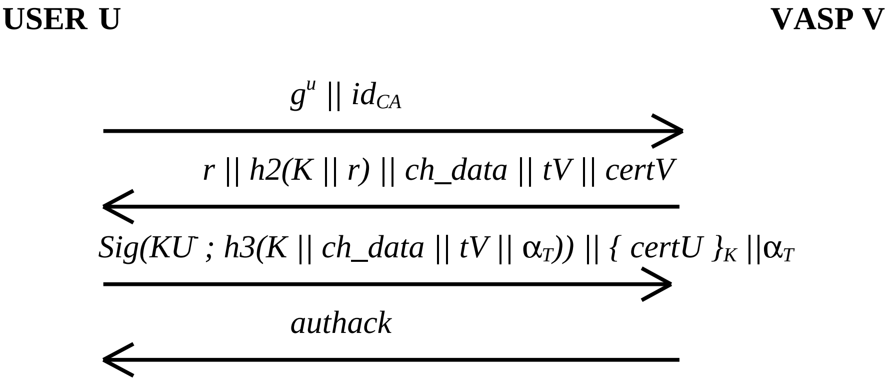
operations at V:

14. decrypt { IdU }$_K$
15. retrieve KU$^-$. If Sig(KU$^-$; h3(K|| ch_data|| tV|| α_T)) verifies set authresp_check = O.K.
16. store Sig(KU$^-$; h3(K|| ch_data|| tV|| α_T)), IdU, K, ch_data, tV, α_T.
17. set j ← T, tck_cnt ← 0, $\alpha \leftarrow \alpha_T$
/ Initialization of parameters of tick payment protocol. */*
18. **send message** V -> U: authack: { }
*/*message contains no user data*/*

operations at U:

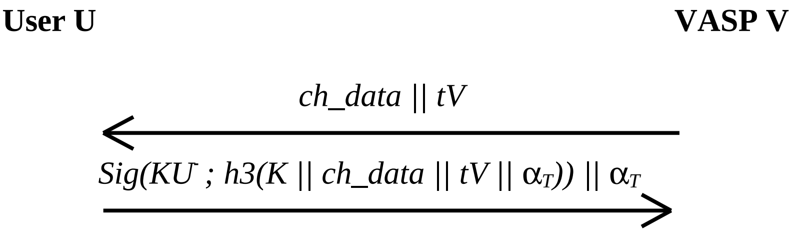
19. set j ← T, tick_total_U ← 0
/ Initialization of parameters of tick payment protocol. */*

Protocol variant B:


USER U
VASP V
g^u || id_{CA}
r || $h2(K$ || $r)$ || ch_data || tV || $certV$
$Sig(KU^- ; h3(K$ || ch_data || tV || $\alpha_T))$ || { $certU$ }$_K$ ||α_T
authack

Detailed description of protocol variant B: straightforward from above description and protocol A.

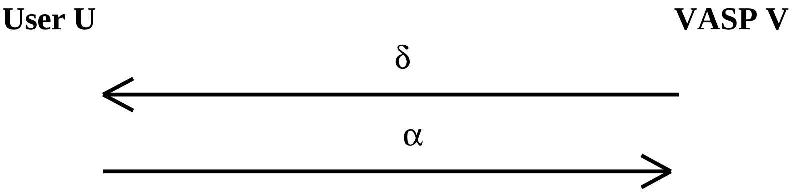
7.2.3 Re-initialization of payment protocol


User U
VASP V
$ch\_data \,||\, tV$
$Sig(KU^- ; h3(K \,||\, ch\_data \,||\, tV \,||\, \alpha_T)) \,||\, \alpha_T$

The detailed description of the protocol is straightforward from the description of protocol A, except perhaps for the following:
Reinitreq_check (U): display ch_data on the screen in separate window if ch_data is different from previously signed ch_data, display warning on the screen if the difference tV-tU (tU = UTC time of U) is greater than a pre-defined delta_t.
authcont_check = O.K. if (ch_data is equal to previously signed ch_data OR ch_data is confirmed by human user via GUI), (the latter can be switched off).
Reinitresp_check (V): check signature

At the end:
at U: Set j ← T, tick_total_U ← 0 /* *Initialization of parameters of tick payment protocol at U. */*
at V: Store transcript Sig(KU$^-$; h3(K|| ch_data|| tV|| α_T)), IdU, K, ch_data, tV, α_T . /* *Do not overwrite previously stored transcripts.*/*
at V: Set j ← T, tck_cnt ← 0, $\alpha \leftarrow \alpha_T$ /* *Re-initialization of parameters of tick payment protocol at V.*/*

7.2.4 Charge ticks protocol



Detailed description of the charge ticks protocol:

Initialization of paramenters is done at the end of the authentication and the re-initialization protocols respectively, see there.

Notation:
at U: /* *T - j is the number of ticks already sent by U. tick_total_U is the number of ticks which has to be paid by U - according to the count of U. δ is the number of ticks whose payment is requested by V in the current run of the tick payment protocol.*/*

at V: */* T - j is the number of ticks requested by V. tck_cnt is the number of ticks correctly received by V. a is the last tick received by V.*/*

Operations at V:

If $j \geq \delta$ the charge ticks protocol is run else the re-initialization of payment protocol is run (cf. FSM description).

1. Set $\alpha' \leftarrow \alpha$
2. Set $j \leftarrow j - \delta$
3. Send message V -> U: ticks: δ

Operations at U:

4. If $(j \geq \delta)$ and $(tick\_total\_U + j - T \geq \delta)$ then chtickreq_check = OK
   */*On the user side, j < dshould not occur in a correct protocol run because the* VASP also checks $j \geq \delta$.*/
5. Set $j \leftarrow j - \delta$
6. Set $\alpha \leftarrow F^{j} (\alpha0)$
7. Send message U → V: chtickresp: α

Operations at V:

8. If $\alpha' = F^{\delta} (\alpha)$ then chtickresp_check = OK
9. Set $tck\_cnt \leftarrow tck\_cnt + \delta$
10. Store α, tck_cnt
    / a, tck_cnt should be continuously stored, not only at the end of the session, otherwise the paid ticks are lost when the program crashes. Previous values are overwritten.*/*

8 Architecture of secure billing demonstrator

The first demonstrator is built in two versions. The first version has been completed by the end of February 1997, the second version will be completed for demonstration at the IS&N conference in May 1997.

8.1 Protocol architecture

Regarding the protocol architecture, it is particularly worth emphasizing that our concept permits the use of existing applications (WorldWideWeb client and server) as well as of existing communication stacks (TCP/IP). Most applications useful in our context are based on a standardized interface, namely sockets, or, in a Windows environment, more specifically Windows sockets. There are three protocol layers (cf. Figure below): An application layer, a communication layer and a security layer in between, realizing the security protocols as described in section 7. The security layer provides Windows sockets to the application layer. There is no need to modify the application and there is no need for a security interface extending Windows sockets. The security layer uses Windows Sockets to access the communication stack. In this way, the security layer is independent from particular applications and is transparent for the application. This is seen as a major advantage of our approach.

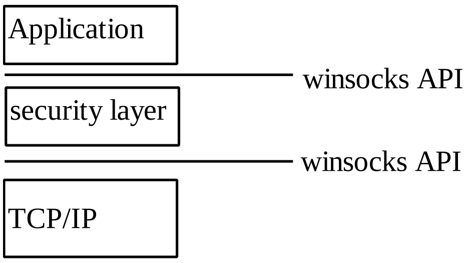


Figure 8.1 - Protocol stack

At the highest level, the application layer will implement the functionality of the Web server on the VASP's site, and of the Web client on the user's site. The server and client will communicate at the application level using the hypertext transfer protocol (HTTP). In practice the Web client will request information by specifying a uniform resource locator (URL), which will uniquely identify the file to send.

At the lowest level, the transport system layer will be implemented as a TCP/IP stack. The Web server and client will communicate at the transport system level using the transmission control protocol (TCP). The transport system layer provides a DLL which will provide a Winsocks interface to the layer directly above it.

The ASPeCT security layer will exist between the application layer and the transport system layer. As such, it must have a Winsocks interface at the upper and lower levels. The Web server and client will communicate at the security control level using ASPeCT-developed secure billing protocols.

The ASPeCT security layer will additionally interface with separate software modules which implement the secure billing demonstrator as outlined in subsection 9.2.

8.2 Architecture of the ASPeCT security layer

The ASPeCT security layer consists of the following principle modules:

The finite state machines (FSMs) module
This module implements the secure billing protocols as specified in section 7.

The security control module
The security control module analyzes Windows sockets calls from the Web application in order to trigger certain events in the secure billing protocols. It also stops transmission of further messages from the server if payment is not duely received from the user.

The cryptographic functions module
It consists mainly of the Siemens cryptographic library ACRYL which provides a uniform Application Programming Interface to access the cryptographic functions.

The tracer module
This module traces the events of other modules (if specified there) and displays them in a window of the GUI.

The secure billing application Graphical User Interface (GUI).
It consists of two parts. The first part displays the tracer messages, the second part provides an interface to manage the security parameters of the system.

8.3 Hardware configuration

Both versions of the first demonstration are based on laptop PCs representing the UMTS user and the VAS provider. The user’s smart card will be attached via a card reader to the user’s laptop PC. The two versions differ in the way in which connectivity is provided between the user’s laptop PC and the VAS provider’s laptop PC. In the first version, connectivity is provided by an Ethernet link. This is depicted in the following figure 8.2:

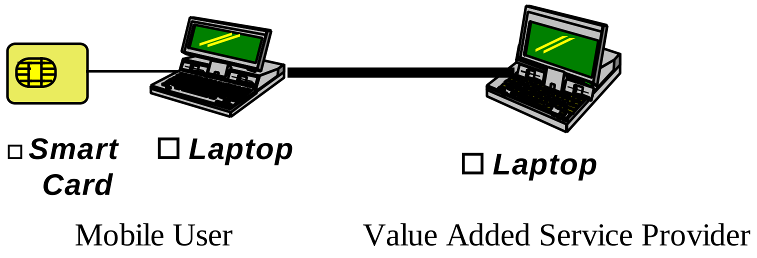


Figure 8.2 - Physical configuration of version 1

In the second version, connectivity will be provided by GSM data connections. This is depicted in the following figure 8.3:

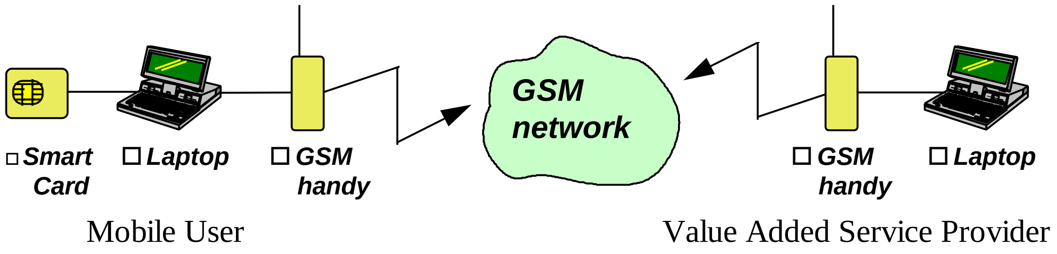


Figure 8.3 - Physical configuration of version 2

The network operator is not represented in the demonstration set-up.

When the demonstration is run over a GSM connection (version 2), the smart card attached to the PC is different from the SIM card needed to operate the GSM handy. In the future, it is envisaged that - together with the advent of mobile communicators with integrated mobile terminal functionality - there will be multi-application smart cards which integrate SIM or UIM functionality with e. g. payment functions such as the ones needed for secure billing.

The smart card chip is the Siemens SLE44CR80CS cryptocontroller which features a crypto co-processor for modular arithmetics which is needed to carry out public-key based cryptographic mechanisms efficiently.

The card reader is attached to the laptop via the PCMCIA slot.

8.4 Software configuration

The Windows 3.11 operating system is used on both demonstration laptop PCs. The smart card chip runs a proprietary Siemens card operating system.

A TCP/IP based communications sub-system provides a means to interface the two PCs. The communications sub-system provides a network programming interface based on the Berkeley Sockets paradigm. This interface is the Windows sockets 1.1 application programming interface. The applications on the user and VASP side are compatible with this interface (see section 8.1 above).

The Windows Web server HTTPd V1.4c for Windows 3.1, will be used on the VASP laptop PC.
The Web client residing on the user laptop PC is Microsoft Internet Explorer in the current version 2.01.

The demonstration software was written in ANSI C.

The second version of the demonstration will also make use of TTP services provided by the ASPeCT TTP demonstrator. In general, both the user and the VASP will have their own, possibly different, TTP. However, in the demonstration only one TTP PC may be used, which will represent the TTP of both the user and the VASP. The user and VASP PCs will communicate with the TTP PC(s) off-line using diskettes.

9 Description of Demonstration

9.1 Introduction

The value-added service (VAS) used in the demonstration will be a service whereby a UMTS user can retrieve information from a remote server. The value-added service provider (VASP) application will be based on Web infrastructure, where the remote information server is a Web server, and the user has a Web client, which can be used to browse information on the remote server.

In a typical scenario the UMTS user connects to the Web server on his UMTS terminal. He then uses his Web client to access information on the VASP's Web server. The Web client presents the UMTS user with hypertext information which he may wish to browse on-line, or save locally for viewing off-line. Secure billing is based on the amount of data transferred to the user over the connection and the value of the information transferred as specified by an agreed charging rate. A secure payment protocol provides an incontestable charging service whereby neither the UMTS user nor the VASP can later deny that a specified amount of data was transferred at a specified charging rate.

9.2 Description of the demonstration from the observer's point of view

9.2.1 Authentication and initialisation of payment

After each entity has been initialised with personalised security information, the VASP can offer a secure billing service to users who wish to browse its Web server. To initiate secure billing the user will execute the authentication and initialisation of payment protocol with the VASP, as described in section 7. This execution will be triggered by using the WEB browser, there is no need for the user to execute it explicitly.

Each entity will first have to obtain a certificate of the other entity's public key. Variant B of the authentication and initialisation of payment protocol is executed if certificates have not been exchanged previously, whereas Variant A is executed if the user and VASP have obtained certificates in a previous run of Variant B.

Part of the authentication and initialisation of payment protocol will involve the VASP supplying the variable *ch_data*, which will be used to specify the charging rate for the subsequent payment protocol.

Optionally, the value of *ch_data* may be displayed to the user in a Window of the Graphical User Interface, and the execution of the protocol is only continued when

the user manually confirms that he agrees with the tariff given by *ch_data*. This feature can be switched off by setting the corresponding security parameter.
The result of the successful execution of the authentication and initialisation of payment protocol will be mutual authentication between the user and the VASP, an agreement of a session key and an initialisation of the tick payment scheme.
The session key may be used for end-to-end confidentiality or integrity services. However, only the key management for these services will be demonstrated here.

Error conditions will be flagged to the user as appropriate via the GUI of the secure billing application.

9.2.2 Data transfer and charging

After the authentication and initialisation of payment protocol has been completed, the VASP will commence the transfer of the appropriate HTTP response message associated with the first HTTP request. During the data transfer the VASP will continually ask the user to release a payment parameter for a number of payment ticks. Providing the user responds appropriately, the data transfer will continue. The secure billing application cansuspend data transfer if a commitment to pay was not received.

9.2.3 Re-initialisation of payment protocol

If the total number of ticks required in the payment protocol exceeds the value of the system parameter *T*, then a commitment to a new series of tick payments has to be given. In this case the re-initialisation of payment protocol is used.
The protocol offers the possibility to agree a new charging rate between the user and the VASP and will commit the user to another series of tick payments.

9.2.4 Tracing

Throughout all interactions, the observer will be able to monitor the message flows using the tracer. The tracer will provide a detailed analysis of the execution of the protocol and will allow recording of the protocol messages, so they can be viewed off-line. The tracer will be an optional feature which can be turned on for demonstration and debugging purposes.

9.2.5 The secure billing application GUI

The user or VASP will interact with the secure billing application via its GUI in order to supply and receive management information relating to the service being offered. For example, the UMTS user can get accurate and reliable information about how much he has been charged for the service received so far. In this respect

the GUI will supply the user (and VASP) with evidence generated by the payment protocol which will confirm the accuracy of the charging information.
At the end of the call, the GUI will present the user or VASP with a summary of the relevant data collected during the call.