Paper to the Second International distributed conference on Network Interoperability, 16-18 June 1997

Authentication for UMTS: Introduction and demonstration

Authors: Geneviève Vanneste, Johan Degraeve, Bart Franco (Siemens Atea, Herentals, Belgium)

Achim Müller (G&D, München, Germany)

keywords: UMTS, Security, smart card

1. Introduction

Currently, work is under way within ETSI to define a third generation mobile telecommunications system, known as the Universal Mobile Telecommunications System (UMTS), to be introduced in the early years of the 21st century. The main objective of UMTS is to offer a plethora of advanced mobile telecommunication services via a variety of public and private network operators in both outdoor and indoor environments. To allow a cost-effective introduction of UMTS, migration/evolution scenarios have been defined within ETSI, aiming at a smooth introduction of the new services and systems, starting from existing contemporary mobile and fixed telecommunication systems.[1]

The need for enhanced security features in UMTS, has led to the definition of specific security objectives. These objectives have been translated into security requirements, resulting in a classification of security features [2]. Mechanisms to realise the UMTS security features are currently under development. Secret key-based mechanisms, as well as public key-based mechanisms have been proposed for UMTS, providing mutual authentication, cipher key agreement for confidentiality, anonymity and non-repudiation.

To enable migration from GSM to UMTS a multi-application card is defined, containing a GSM SIM application and a preliminary UMTS UIM application. To achieve flexible introduction of new authentication mechanisms and algorithms, a framework for authentication has been introduced, with the ability to migrate smoothly from one mechanism to another.

In order to facilitate roaming in a network with a large number of Network Operators and Service Providers, it might be desirable (or even necessary) for roaming agreements to be set-up dynamically, as and when they are required.

A demonstrator, on a PC base, has been developed, demonstrating the authentication framework with a proposed public key and secret key based authentication mechanism for UMTS. At the user's side a smart card is available, providing the authentication functionality. The demonstrator will show the feasibility of a multiapplication card for GSM and UMTS. The card will contain both a GSM SIM application and a preliminary UMTS application. Various problems arise in supporting multiple applications on a single card : application selection, independence of application. A careful approach is required to ensure both adequate security and sufficient interoperability. The functionality of the terminal is put on a PC. At the network's side an Authentication centre (AC) has been developed. This AC can function in a visited network (with the NO) or in a home network (with the SP). The procedures for automatic roaming agreement are not demonstrated.

In a second demonstrator, the authentication framework will be demonstrated in an experimental UMTS environment with real users. It is our goal to connect the developed Authentication centre to a SCP (Service control point), by February 1998.

An object oriented design methodology has been followed. This has been most advantageous for the development of the finite state machines realising the authentication protocols.

The chosen validation criteria for the evaluation of the demonstrated procedures are presented.

This demonstrator has been developed in the ACTS project AC095, ASPeCT (Advanced security for Personal communications Technologies).

2. UMTS security mechanisms

In the following sections the currently discussed authentication framework for UMTS is presented, together with one of the proposed authentication mechanisms. The mechanism is based on public key crypto systems.

The interface to the UMTS UIM is described in detail.

2.1. The Authentication framework

The principle objective of the Authentication Framework [3] is to provide a flexible procedure for user-network authentication allowing a number of different mechanisms and algorithms to be incorporated, with the ability to migrate smoothly from one mechanism to another. This framework allows the authentication capabilities of SIMs, network operators (NOs) and service providers (SPs) to be taken into consideration for the selection of the mechanism to be used. A list of capability classes (including the mechanisms supported) will need to be maintained so that different entities (SIMs, NOs, SPs and TTPs) can permit the negotiation of the mechanisms to be used.

In order to facilitate roaming in a network with a large number of NOs and SPs, it might be desirable (or even necessary) for roaming agreements to be set-up dynamically, as and when they are required. In practice, the roaming agreement would be first requested as a result of an initial authentication request sent by the user/terminal to a network visited for the first time. A prerequisite of this procedure is that the SP and NO wishing to establish the agreement have authenticated each other.

NO-SP authentication will be carried out using a globally agreed mechanism in order to ensure that NOs and SPs world-wide have the capability to authenticate each other. Unlike the user-network authentication mechanism, flexibility to change mechanisms is not considered to be a crucial factor. Apart from being a prerequisite to a roaming agreement, NO-SP authentication will permit the SP to delegate user-network authentication to the NO. The SP would send authentication data to the NO in advance, permitting the NO to carry out authentication on behalf of the SP.

It should be noted that the identity of the User is not released until the stage of user-network authentication. The rationale for this is that the identity of the User is immaterial until the stage of authentication is reached; it is only the identity of the Service Provider which is required up until the stage of authentication. Note also that the identity of the User is never necessarily required by the Network Operator, hence temporary identities are used to provide party anonymity of the User towards the Network Operator.

A further characteristic of the Authentication Framework is the use of an authentication Capability Class, which acts to identify the particular authentication mechanisms which are supported by the UIM of a User. Each respective authentication mechanism is identified by an unique identifier. The rationale for this is that visited Network Operators may immediately identify whether they can support a particular Capability Class; unknown authentication mechanisms would be defined by the respective Service Provider upon request from the Network Operator.

2.1.1. Operational Scenario

As an example the operational scenario (described in [3], but here some enhancements are included) is described where a user, not registered in the network, initiates authentication and no roaming agreement exists between the Network and the user's service provider.

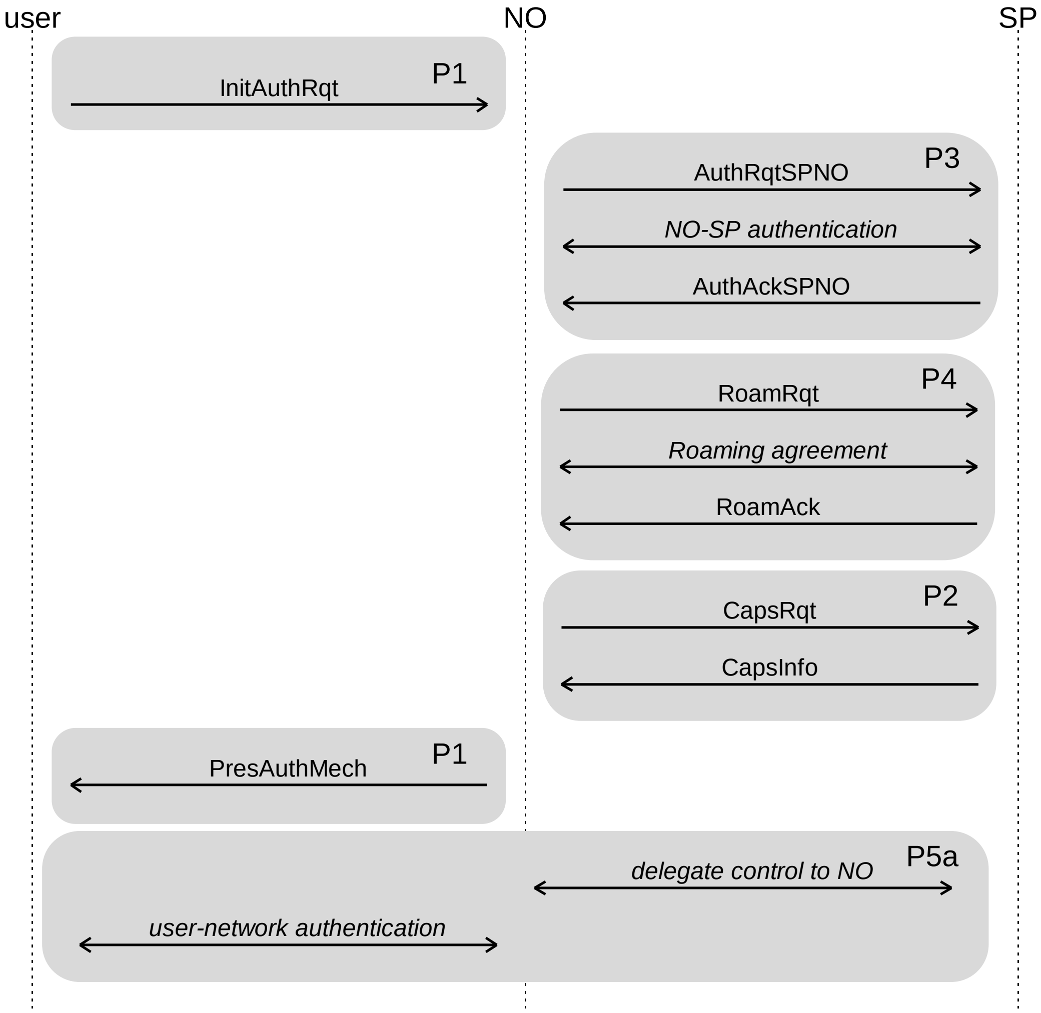


Figure 1.Operational scenario for ‘User not registered, no roaming agreement’

The user sends an initial message to a NO - this will include the user’s service provider, authentication capability class, but not his identity nor his temporary identity. The NO does not have a roaming agreement with the SP so it initiates a procedure to establish one dynamically - if one cannot be established dynamically, then the request is refused. A procedure to establish a roaming agreement begins with the NO and SP authenticating each other. After authentication the NO and SP negotiate a roaming agreement which will involve each party digitally signing the agreement. Once an agreement has been established, the NO checks the authentication Capability Class of the User to establish if it is known. If it is known, the Network operator compares the associated authentication mechanisms with its own supported authentication mechanisms. If it is not known, the Network Operator sends the user’s authentication capability class to his SP. The SP will respond by providing the NO with the authentication capabilities of that particular authentication capability class - this will include the authentication mechanisms the user is capable of handling. The NO will then choose an authentication mechanism, from those of the User’s Capability Class, which is both supported by the Network Operator and by the User’s UIM. The NO then sends the identity of the prescribed mechanism to the user. The authentication mechanism for new registrations involving the SP, NO and user is initiated. Note, however, that the SP may choose to delegate the actual authentication to a Certification Authority (CA).

2.2. Public key Authentication mechanism

The ***Siemens protocol*** is a public key based authentication mechanism defined by Siemens AG . There exist three versions of the protocols. In the following sections only one version is described, the version allowing authentication of a user to a network, without the need that they share certificates of each other. This version is applied when 'New Registration' of a user in a network occurs. [4]

The goals of the protocol are the following:

- mutual explicit entity authentication of User and Network operator
- agreement between the user and the Network operator on a shared secret key K_S with mutual implicit key authentication
- mutual key confirmation of the User and the Network operator
- mutual assurance of key freshness
- non-repudiation by the User of data sent by the User to the Network operator and vice versa
- confidentiality of the identity IMUI of the User on the air interface
- exchange of certified public keys between U and N

The **data** used within the protocol:

AUTH$_N$: This value is calculated to authenticate the network operator (NO) to the user .

CertN : This is a valid certificate, issued by a certification authority CA, on a public key of the asymmetric signature system of the NO. It is available at the NO.

CertU : This is a valid certificate, issued by a certification authority CA, on a public key of the asymmetric signature system of the user. It is available at the user.

data1, data2 : Those are optional data fields, to illustrate the non-repudiation feature.

id$_{ca}$: : This is the identity of the Certification Authority.

id$_n$: This is the identity of the NO.

K$_S$: This is the session key .

g : generator g, known by the user, NO and SP, g is a generator of a finite group G with modulo p (p is a prime) in which the Discrete Logarithm Problem is hard .

s : This is the secret key agreement key for the NO. It is linked with g^s (the public key agreement key).

g^s : This is the public key agreement key of the NO.

IMUI : This is the International Mobile User Identity, uniquely identifying the mobile user.

PK_U : This is the public key of the user used to verify **signatures** from the user.

RND$_U$: This is random number generated by the user.

The **algorithms** used within the protocol:

h1 : This is a one way function. It is used to calculate the session key:

$$K_S = \mathbf{h1(}(g^{RNDU})^s \,||\, RND_N\mathbf{)}$$

h2 : This is a hash function and used to calculate $AUTH_N$.

$$AUTH_N = \mathbf{h2\ (}K_S\mathbf{)}$$

h3 : This is a hash function and used to calculate a hash value before signature calculation

Sig$_u$: This is a secret **signature** transformation owned by the user.

Ver$_u$: This is a verification algorithm corresponding the signature algorithm. This algorithm needs the public key (PK_U in this case) as input.

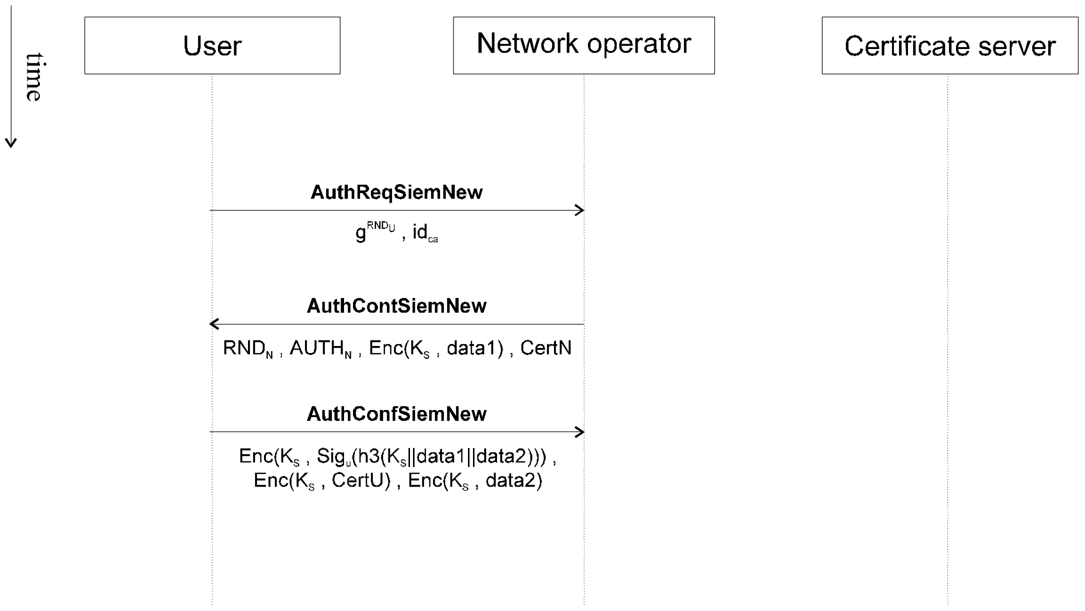
Enc: This is a symmetric encryption algorithm. **Enc**(K,data**)** means that data is encrypted with key K.

Dec: This is a symmetric decryption algorithm. It corresponds **Enc().**

The following list is **required** (by user and/or NO):

- the user needs the id_{no} ;
- both the user and NO possess the generator g;
- the NO has secret and public key agreement keys s and g^s ;
- the NO has a valid certificate CertN;
- the user has a signature transformation Sig_u ;
- the user has a valid certificate CertU;

The message flow corresponding to the authentication protocol:


time
User
Network operator
Certificate server
AuthReqSiemNew
g^{RND_U} , id_{ca}
AuthContSiemNew
RND_N , $AUTH_N$, $Enc(K_S , data1)$, CertN
AuthConfSiemNew
$Enc(K_S , Sig_u(h3(K_S||data1||data2)))$,
$Enc(K_S , CertU)$, $Enc(K_S , data2)$

The certificate server is not interrogated on-line during this version of the protocol.

2.3. *The UMTS UIM*

Until now, the discussion of the authentication mechanisms (in particular the Siemens protocol) has been concentrated on messages exchanged between logical entities: the user, the network operator and the service provider. This is not sufficient to describe the interface between a UMTS mobile terminal and a UMTS UIM. The message flow across that interface is different for at least three reasons:

- the communication structure is fixed: the mobile terminal sends a card command to the UIM (together with some data) and gets data back (card response).
- the UIM cannot initiate a communication: this is not envisaged for the UMTS. For example, a framework like the SIM Toolkit in GSM does not yet exist.
- the message length is limited to 255 bytes. Longer messages can only be sent with some chaining mechanism.

The aim of this section is to give an introduction into the design process for UIM card commands - how to map the message flow across the user / network operator interface as described before to some functionality of security related card commands necessary to support that message flow.

During authentication set-up, the network operator sends a message called *InitAuthRqt* to the user's service provider in case of a new registration. The content of this message is the *authentication capability class* of the user and the *identity* of its service provider. Because the user is not yet known to the network, it is up to the UMTS mobile terminal to read out this information from the UIM plugged into it. The first two card commands issued are therefore:

- ReadBinary(File_ID[EF_{SPID}])
- ReadBinary(File_ID[EF_{ACCL}])

The elementary files EF_{ACCL} and EF_{SPID} store the respective data and are integrated into a directory DF_{UMTS}.

The service provider informs the network operator about the user's authentication capabilities. Based on this information, the network operator makes a choice among a list of mechanisms supported by the UIM and sets that mechanism for the rest of the current session with the card command

- SetAuthMechanism(AuthMechID)

The UIM responds with Yes or No depending on the network operator's choice.

In case of the Siemens protocol, it is necessary that the network knows which certification authorities are supported by the UIM:

- which CA issued the certificate on the user's public key agreement and signature verification key ?
- which CA is accepted by the UIM - can the UIM verify certificates issued by a particular CA ?

In general, it may be possible that the UIM splits this information and stores it in different files. For the rest of this section, we assume that one CA satisfies both purposes. The next card command issued by the mobile terminal is therefore:

- IdCA = ReadBinary(File_ID[EF_{CA}])

The mobile terminal sends this identity transparently to the network operator which looks for a certificate on its public key agreement key issued by that particular CA. If the NO can provide such a certificate, it sends it to the mobile terminal which stores it in some file EF_{CertN} and verifies its validity:

- Status = VerifyCertificate(File_ID[EF_{CertN}])

The status value is either Yes or No depending on a positive verification of the appended signature on it by the UIM.

The No's certificate contains (among other things) information on an elliptic curve and some rational point *g* on it which defines a cyclic group over that curve acceptable for cryptographic purposes. Armed with this information, it is possible for the UIM to compute powers of *g* for any random integral exponent. This is the next step in the Siemens protocol and achieved by the card command

- P = GetChallenge( )

The UIM computes a random number *RND* and sends g^{RND} back as the response data for GetChallenge.

The challenge P produced by the UIM is transparently forwarded to the network by the mobile terminal. With this challenge and its private key agreement key, the network can compute a session key K_S shared with the UIM and an authentication token $AUTH_N$ which is a hash code of the session key:

- $AUTH_N$ = MutualAuthenticate(M)

This card command uses the message M sent by the network to the mobile terminal as a response to the challenge P:

$$M = RND_N \parallel AUTH_N \parallel Enc(K_S,data1)$$

The AUTH token computed by the UIM covers

$$Enc(K_S,Sig(K_S,data1)) \parallel Enc(K_S,CertU)$$

according to the protocol. This is unfortunately not immediately possible because of the length restriction for the card's response. The encrypted user certificate is therefore read out with a second card command which is the last one for the Siemens protocol:

- EncryptedCertificate = ReadCertificate( )

It may look surprising at first sight how the whole protocol changes by breaking down the message flow across a different interface. But this also shows that design of card commands is different from defining messages flowing across a network.

3. Description of the demonstrator

The demonstrator has been developed within the ACTS project ASPeCT (Advanced security for Personal Communications technologies and services). The aim of the demonstrator is to show a migratory path for security features. After a study in depth on the migration problem, a multi-application card for GSM and UMTS is proposed as the migratory path to introduce new and enhanced security features. The UMTS authentication framework is implemented in combination with a public key based authentication mechanism (see section 2.2). A version of the demonstration without smart card is also available for a secret key based authentication mechanism.

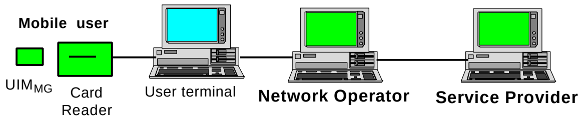
3.1. Overview

Within the demo, three logical entities (roles) are involved:

- The User: he is authorised by a subscriber to make use of the telecommunication services, the subscriber subscribed to by the service provider.

- The network operator: provides the network capabilities necessary for the support of the services or set of services offered to the users.
- Service Provider : has overall responsibility for the provision of a service or set of services to users associated with a subscription and for negotiating the network capabilities associated with that service or set of services with network operators.

These roles have been mapped upon the following physical architecture:



Description of the demonstrated features:

New registration, roaming agreement exists:

The user is not registered with the NO, the NO and user's SP do have a roaming agreement. The NO has no security related data for the user.

The user wants to register and sends a registration request to the network. The network recognises that it has a roaming agreement for the user's service provider. The NO will receive the authentication capabilities from the registered user by the SP. An authentication mechanism will be negotiated and executed between the NO and the user.

Authentication of registered user:

The user is registered with the NO, the NO has security related data for the user.

The NO or the user can initiate the authentication by sending the appropriate message. The NO has all necessary data of the user, it has a roaming agreement with the SP and knows the users authentication capabilities. An authentication mechanism will be executed between the NO and the user.

The authentication mechanism implemented on the smart card (UIM) and in the network is the one described in section 2.2.

3.2. Selection of algorithms

The algorithms chosen for the mechanism described in section 2.2 are:

h1, h2, h3 : RIPEMD-128

Sig_u, Ver_u : AMV signature, based on elliptic curves, .

Enc, Dec: DES-CBC

finite group G : elliptic curve, one point is represented by 40 bits.

3.3. The UIM realisation

The UIM implemented for the demonstration supports both the GSM and the UMTS. It will be a multifunctional smart card with at least two applications on it. For the demonstration, the UIM supports the SIM functionality as specified for GSM Phase 2 and the security functionality defined by the Siemens authentication protocol.

From an observer's point of view, the UIM can be used as a GSM SIM by setting up a phone call with a GSM handset that supports standard features. This shows that the card is compatible with the GSM Phase 2 functionality according to the most recent version of the GSM 11.11 specification.

The UIM functionality is demonstrated by plugging the smart card into an intelligent card reader that communicates with a PC that simulates the UMTS network. The access to the security functionality is controlled by the mobile terminal implementation in that PC.

It will also be possible to directly access data stored in the UIM that is not security relevant with the card reader's display and keyboard. This does not touch the message flow between PC and UIM and allows a convenient interface to user data stored in it.

3.4. The Network realisation

The demonstrator we have to build for the ASPeCT project consists of several entities exchanging messages to each other. Each entity acts like a finite state machine. It receives an event (a communication message over TCP/IP, serial link or message queue or a user message from the Graphical User Interface) and responses to that event by taking some actions like calculating an algorithm and sending a message. Both communication between entities in the same application (via a message queue) as well as communication between entities in different applications (via TCP/IP) are possible.

In the demonstrator, all entities are represented by finite state machines. The design of these finite state machines is based on a state pattern, and implemented in C++ [5]. We use this state pattern for the following reasons :

- An object's behaviour depends on its state, and it must change its behaviour at run-time depending on that state.
- Operations have large, multipart conditional statements that depend on the object's state. This state is usually represented by one or more enumerated constants. Often, several operations will contain this same conditional structure. The state pattern puts each branch of the conditional in a separate class. This lets you treat the object's state as an object in its own right that can vary independently from other objects.

The intent of the state pattern is to allow an object to alter its behaviour when its internal state changes. The object will appear to change its class.

The key idea in this pattern is to introduce an abstract class (TFSM_State) to represent the states of all entities in the demo. This abstract class declares an interface common to all classes that represent different operational states. The subclasses of this abstract class implement the state-specific behaviour.

The class TFSM maintains a state object (an instance of a subclass of TFSM_State) that represents the current state. The class TFSM delegates all state-specific requests to this state object. TFSM uses its TFSM_State subclass instance to perform operations particular to the state.

Whenever the state changes, the TFSM object changes the state object it uses.

An ASN.1 shareware tool is used to produce the necessary C++ routines for BER encoding and decoding of the messages exchanged in our demo. ASN.1 is a notation for describing data structures. It is an abstract representation because it does not specify how data is represented in a local computer nor does it specify how data is represented when they are communicated between systems. The tool we use is called SNACC and is freely available via the internet.

It was agreed between the ASPeCT partners to use the ACRYL library from Siemens ZT IK 3 for the provision of basic cryptographic functions. Following functions are provided by ACRYL, which stands for Advanced CRYptographic Library :

- Random number generation based on DES-OFB and triple DES-OFB
- Hash functions RIPEMD-128 and RIPEMD-160
- RSA signature generation and verification
- AMV signature generation and verification based on an elliptic curve over GF(p)
- Encryption with DES-CBC and triple DES-CBC
- Exponentiation in GF(p)
- Exponentiation in an elliptic curve over GF(p)
- Key generation for RSA, DES and elliptic curves

3.5. Evaluation of the demonstrator

Performance measurements have been done, to calculate the total delay introduced by running the authentication framework in combination with the public key authentication mechanism. More results are shown in the demonstration.

4. Conclusion

The restrictions of the used environment, being just a prototype of some entities, composing a real network, restrict the value of the made measurements.

However it gives a first indication on the impact and feasibility of having multi-application cards and authentication based on public key mechanisms.

The demonstrator is a good basis for the realisation of the authentication mechanism in an UMTS experimental environment. By the end of

February 1998 this demonstrator will be ported and enhanced on the trial network of the ACTS EXODUS network.

5. References

[1] ETR 050101 Special Mobile Group (SMG): Universal Mobile Telecommunications System (UMTS) objectives and overview version 2.1.0

[2]ETR 050901 Special Mobile Group (SMG): Security principles for the Universal Mobile Telecommunications (UMTS) version 3.0.0

[3] AC095/Atea/WP21/DS/P/05/1 Migration Scenarios, ACTS ASPeCT deliverable

[4] AC095/Atea/WP21/DS/P/02/1 Initial report on security requirements, ACTS ASPeCT deliverable

[5] Design Patterns, Elements of Reusable Object-Oriented Software, ISBN 0-201-63361-2

6. Abbreviations

AC	Authentication Centre
ACTS	Advanced communications technologies and services
AMV	the Agnew-Mullin-Vanstone equation
ASPeCT	Advanced security for Personal Communications Technologies
CA	Certification Authority
ETR	ETSI technical report
ETSI	European telecommunications standards institute
FSM	Finite State Machine
GSM	Global system for mobile communications
NO	Network operator
SIM	Subscriber Identification module
SP	Service Provider
UIM	user identity module
UMTS	Universal Mobile Telecommunication System

7. Authors

Geneviève Vanneste received the computer science engineering Degree at the Katholieke Universiteit Leuven in Belgium in 1989. After 1 year being employed as a trainee at the ENC of IBM in Heidelberg Germany, she started at ATEA in the mobile development department. In 1990-1991, she was involved in the specification of a secure TMN architecture for the GSM-AC. She remained responsible for the technical part of the realisation of that architecture in the AC until 1994. Since May 1993 she attends ETSI SMG5, focusing on security for UMTS and the alignment with FPLMTS. Since the start of ETSI SMG 10, June 1995, she actively contributes to the GSM and UMTS standardisation process. In 1994-1995 she attended TR46, standardising PCS in the US, focusing on security and interworking. In 1995-1996 she regularly attended a subgroup of T1P1, standardising a multi-application smart card, UIM, in the US for PCS. Since September 1995 she participates in the EC funded ACTS project, ASPeCT. She co-ordinates the work done in the work package studying migration towards UMTS security, a demonstrator of the proposed UMTS authentication framework is prepared. She participates in the development of security concepts for the Siemens mobile switch.

Johan Degraeve graduated as a civil engineer in electronics at the Rijksuniversiteit Gent in Belgium, in 1989. After serving the Belgian Army he joined Siemens Atea where he started in the software development department for mobilophony, where he is still active now. In this department he participated for two years in the development of the GSM Authentication Centre. After that he participated three years in the overall co-ordination of the software development and test for different country-specific D900 realisations. D900 is the Siemens GSM MSC/VLR/HLR/AuC realisation. Since September 1995 he participates in the EC funded ACTS project ASPeCT, which studies the security requirements for the third generation mobile telecommunication systems. He also attends the ETSI SMG10 meetings, where security for GSM and UMTS are standardised.

Bart Franco received the electronics engineering Degree at the Katholieke Industriële Hogeschool der Kempen Geel in Belgium, in 1989. After serving the army, he joined Siemens Atea in 1990. He was engaged in the B900 GSM project, as a member of the call processing team. It was this team that made Belgian's first GSM calls, in coöperation with Proximus, Belgacom Mobile. Later on he joined the C450 mobile team, again as a member of the call processing team, developing a number of new features. From 1996 on, he is participating in the ASPeCT project, which studies the feasibility and acceptability of new and advanced security features in existing and future personal communications networks, based on demonstrations and trails.

Achim Müller studied Computer Science, Economics and Mathematics at the University of the Saarland in Saarbrücken from 1990 until 1995. In 1994, he was admitted to the 'Studienstiftung des Deutschen Volkes' and received the computer science engineering degree in 1995. From 1995 until 1996, he joined the 'Graduiertenkolleg Informatik' funded by the 'Deutsche Forschungsgemeinschaft'. In 1996, he started as a systems engineer on a staff position in the mobile communications department of Giesecke & Devrient in Munich and participates in the EC funded ACTS project ASPeCT. His work concentrated on the specification of a UMTS UIM and on migration of security with respect to multifunctional telecommunication smart cards. He is currently also involved in a security project focusing on prepaid solutions for GSM.

Contact address:
Geneviève Vanneste,
p82586@vnet.atea.be
Siemens Atea
Atealaan 34
B-2200 Herentals
Belgium