MASTER

# Securely accessing a web service using a mobile station

Tiel van, G.J.L.C.M.

*Award date:*
2008

TECHNISCHE UNIVERSITEIT EINDHOVEN
Department of Mathematics and Computer Science

MASTER'S THESIS

# Securely accessing a Web Service using a Mobile Station

by
ing. G.J.L.C.M. van Tiel

Supervisors:
dr. B.M.M. (Benne) de Weger (TU/e)
ir.A.A.M. (Ton) van Opstal (Ericsson)

Eindhoven March 2008

## Preface

This master's thesis concludes my studies at the Technische Universiteit Eindhoven at the department of Mathematics and Computer Science. The research I have done to conclude my studies would not have been possible without the help of a number of people.

The project was carried out at Ericsson in the Customizations division. Special thanks go to Ton van Opstal and Frank Jansen who both took care of my daily supervision.

I would like to thank the Technische Universiteit Eindhoven for their support. Thanks to my supervisor Benne de Weger for providing constructive input. Also thanks to Jerry den Hartog and Igor Radovanovic for taking place in my assessment committee.

I would like to thank my father and my girlfriend, Lydia, for their continuous support.

Finally, a word of appreciation for my friends and my colleagues at Ericsson for their support during my project.

Frits van Tiel
Eindhoven, March 2008

## Abstract

In this thesis we investigate the problem of securely accessing a Web Service from a Mobile Station. Current Web Services mostly offer low or no security by only requiring username/password authentication. This is prone to phishing and requires the User to remember a vast amount of usernames and passwords. Web Services which require more secure authentication, such as online banking and government services online often rely on additional hardware devices. These hardware devices are Web Service specific and thus a User may need to carry multiple devices. Carrying these devices is inconvenient for a User who wishes to be mobile. Additionally current Web Services do not offer non-repudiation of a User's actions. Therefore we are unsatisfied with current solutions.

We propose to use the capabilities of the Smart Card, also know as Subscriber Identity Module(SIM) Card, in the Mobile Station of the GSM Network for authenticating a User to a Web Service. Using these capabilities we can also achieve non-repudiation of a User's actions. By combining the capability of the Smart Card to sign a security token with the identity metasystem[10] envisioned by Microsoft we achieve authentication. Because the Smart Card is a tamper resistant device and it requires a PIN to operate we achieve two-factor authentication. To achieve non-repudiation we combine the signing capability of the Smart Card with the Digital Signature Service(DSS) protocol[20].

The proposed solution is feasible because it leverages existing hardware, software and protocols and doesn't require additional hardware. Current two-factor authentication systems require an additional hardware device, therefore our solution is also more user friendly. The adaptation of our solution to the identity metasystem presents the User with a consistent interface across multiple devices because the identity metasystem is also available on the PC.

Our contribution to the solution is fourfold. The first is the Identity Selector on the Mobile Station. The Identity Selector supplies security tokens to the Service Requester. The Identity Selector is accessible as a local service, so the Service Requester could be a browser, Java ME Midlet or native application. The Identity Selector is responsible for obtaining security tokens from the Identity Provider. We extend the definition of the Identity Selector as given in the Identity Metasystem by also making it responsible for assistance in the DSS protocol. The Digital Signature Service defines an interface to process Digital Signatures for Web Services.

Our second contribution is the Identity Provider on the Mobile Station. This entity is responsible for supplying the Identity Selector with security tokens. The Identity Provider communicates with the Smart Card using SATSA-PKI[13]. By the extended definition we have given the Identity Provider it is also responsible for supplying the Identity Selector with Digital Signatures through the DSS protocol. By keeping the Identity Provider and Identity Selector as separated entities we have the possibility to deploy them on different devices in the future.

The third contribution is the Certificate Manager. The Certificate Manager is not defined in the identity metasystem. We define it as an extension. The Certificate Manager is responsible for supplying the Smart Card with the certificates required to make Digital Signatures. The Certificate Manager obtains these certificates from the Certification Authority[27]. To mutually authenticate the User and Certification Authority the GSM SIM authentication mechanisms are combined with EAP-SIM[24]. The Certificate Manager uses SATSA-APDU[13] to communicate with the SIM in this authentication. To store the certificate on the Smart Card the Certificate Manager uses SATSA-PKI.

As the fourth contribution to the solution we have defined a security token profile for use in the identity metasystem. Several security token profiles have been defined. Only the X.509 Certificate Token Profile[23] can contain a Digital Signature. Unfortunately the Digital Signature format used in this security token profile is incompatible with those supplied by SATSA-PKI. Therefore we define our own security token profile containing a Digital Signature produced by SATSA-PKI.

The above described concepts have been proven with a proof of concept. The proof of concept runs on an emulated Mobile Station with an emulated Smart Card. Also the GSM Network Operator involved in the SIM authentication is emulated. A limited Certificate Authority and Relying Party have been implemented to illustrate implementation on this side is also possible. The proof of concept does not fully implement all four contributions but shows the feasibility of the most important facets.

Future work in this area could be extending this solution to be used on any device with a Smart Card reader. Taking this a step further would be investigating the possibilities of using the Mobile Station and Smart Card as an Identity Provider for an Identity Selector running on another device.

## Contents

## List of Figures

## List of Tables

## Abbreviations

| | |
|---|---|
| AuC | Authentication Centre |
| CDMA | Code division multiple access |
| CHV | Card Holder Verification |
| CMS | Cryptographic Message Syntax |
| CSR | Certificate Signing Request |
| DSS | Digital Signature Services |
| EAP | Extensible Authentication Protocol |
| GSM | Global System for Mobile Communications |
| GUI | Graphical User Interface |
| ICC | Integrated Circuit Card |
| JSR | Java Specification Requests |
| IMSI | International Mobile Subscriber Identity |
| MAC | Message Authentication Code |
| OCSP | Online Certificate Status Protocol |
| PIN | Personal Identification Number |
| PKI | Public Key Infrastructure |
| PUK | Personal Unblocking Key |
| RUIM | Removable User Identity Module |
| SAML | Security Assertion Markup Language |
| SIM | Subscriber Identity Module |
| TLS | Transport Layer Security |
| UICC | Universal Integrated Circuit Card |
| UMTS | Universal Mobile Telecommunications System |

## Definitions

Adopted from Electronic Authentication Guideline[19]

| | |
|---|---|
| Assertion | A statement from a verifier to a Relying Party that contains Digital Identity information about a party. Assertions may also contain verified attributes. Assertions may be digitally signed objects or they may be obtained from a trusted source by a secure protocol. |
| Certification Authority | A trusted entity that issues and revokes public key certificates. |
| Claimant | A party whose identity is to be verified using an authentication protocol |
| Credential | An object that authoritatively binds an identity (and optionally, additional attributes) to a token possessed and controlled by a person. |
| Data integrity | The property that data has not been altered by an unauthorized entity. |
| Digital Signature | An asymmetric key operation where the private key is used to digitally sign an electronic document and the public key is used to verify the signature. Digital signatures provide authentication and integrity protection. |
| PIN | A password consisting only of decimal digits. |
| Registration Authority | A trusted entity that establishes and vouches for the Digital Identity of a entity to a Certification Authority. The Registration Authority may be an integral part of a Certification Authority, or it may be independent of a Certification Authority, but it has a relationship to the Certification Authority. |

Adopted from the Identity Metasystem[10]

| | |
|---|---|
| Digital Identity | A set of claims made by one party about another party |
| Identity Provider | A network entity providing the digital identity claims used by a Relying Party |
| Identity Selector | A software component available to the Service Requester through which the User controls and dispatches her Digital Identities |
| Relying Party | A network entity providing a Web Service, and relying upon Digital Identity |
| Security Token | A token containing claims made about or by and identity |
| Service Requester | A software component acting on behalf of a party who wants to obtain a Web Service through a digital network |
| User | A party which is a human being or system who wants to get access to a Web Service |
| Web Service | A service accessible over a IP network, provided by a Relying Party |

Adopted from GSM 01.04[1]

| | |
|---|---|
| GSM Network Operator | An administration or its licensed operator(s) which provides a GSM network and its Mobile Services |
| Mobile Equipment | A Mobile Station without the SIM |
| Mobile Service | A radio communication service between mobile and land stations, or between Mobile Stations. |
| Mobile Station | Equipment intended to access a set of Mobile Services while in motion or during halts at unspecified points |
| Subscriber | An individual or entity who, or which, obtains a Mobile Service from a GSM Network Operator |

# 1 Introduction

## 1.1 Motivation

The world of mobile communications is expanding from voice communications to other area's. People are using their Mobile Stations to access other services like SMS, email and internet.

More and more services are provided over the internet and with internet coming to the Mobile Station, these Web Services are accessible everywhere. Several parties, such as banks, web-shops and governments, provide Web Services.

Access to these Web Services brings new demands and threats to the Mobile Station and parties providing these Web Services. For example the Web Services must be protected against unauthorized access and eavesdropping. We call the parties who provide these Web Services Relying Parties, because they rely on security measures. There are some existing solutions and other solutions have been proposed, but none of them fulfill all our requirements.

## 1.2 Problem definition

A solution has to be found in which a Mobile Station can securely access a Web Service. This access must be mutually authenticated, it must be resistent to eavesdropping and must guarantee data integrity. In addition a User must not be able to repudiate his involvement in a transaction. Even with these strict requirements the solution must be privacy- and most important user-friendly.

## 1.3 Objectives

The objective is to propose an architecture which provides a solution to this problem. The architecture will be shown to work with a proof of concept. We provide an evaluation of the architecture and the proof of concept. The final objective is to estimate which problems can be expected when implementing the solution and touch upon some further research in this area.

## 1.4 Thesis structure

In section 2 all relevant background material is summarized and where necessary put into context. The section could be skipped by readers familiar with the subject. The thesis continues by formalizing the requirements and presenting use cases in section 3. This section also describes existing solutions and why they are insufficient. The result of the literature study and evaluation of the requirements is the basis of the solution which we propose. The solution is presented in section 4 by means of an architecture. Section 5 presents the solution in a more concrete way by a design, and evaluation of the proposed solution with regard to the requirements. The solution is partly implemented resulting in a proof of concept which is discussed and evaluated in section 6. We conclude this thesis with our Conclusions and suggestions for future work in section 7.

# 2 Background

## 2.1 Authentication

Authentication is the act of establishing the validity of an item, meaning that claims made about or by the item are shown to be true. When looking at Information Security we are interested in authentication of:

- Digital Identity: is a claimant indeed who he says he is.
- Data: is the data un-tampered.

The authentication part of this thesis focuses on the authentication of a Digital Identity. The following subsections summarize relevant information from the Electronic Authentication Guideline[19] by the National Institute of Standards and Technology(NIST).

### 2.1.1 Authentication Factors

Authentication depends on one or more authentication factors. Factors of authenfig:identitymetasystemtication of humans can be divided into three classes:

- Something you have: a security token or ID card
- Something you know: a password or PIN
- Something you are: a fingerprint or other biometric identifier

### 2.1.2 Authentication Tokens

To proves the validity of a claim, tokens are used in authentication. Each type of token incorporates one or more of the authentication factors. Tokens that provide a higher level of assurance incorporate two or more factors. There are four kinds of tokens.

#### Hard token

A hardware device that contains a protected cryptographic key. Authentication is accomplished by proving possession of the device and control of the key. Hard tokens shall:

- require the entry of a password or a biometric to activate the authentication key;
- not be able to export authentication keys;

#### Soft token

A cryptographic key that is typically stored on disk or some other media. Authentication is accomplished by proving possession and control of the key. The soft token key shall be encrypted under a key derived from some activation data. Typically, this activation data will be a password known only to the User, so a password is required to activate the token. Each authentication shall require entry of the password or other activation data and the unencrypted copy of the authentication key shall be erased after each authentication.

#### One-time password device token

A hardware device that generates one time passwords for use in authentication. The device may or may not have some kind of integral entry pad, an integral biometric (e.g., fingerprint) reader or a direct computer interface (e.g., USB port). The passwords shall be generated by using a block cipher or hash algorithm to combine a symmetric key stored on a hardware device with a nonce to generate a one-time password. The nonce may be a date and time, a counter generated on the device, or a challenge from the verifier (if the device has an entry capability). The one-time password typically is displayed on the device and manually input to the verifier as a password (direct electronic input from the device to a computer is also allowed). The one-time password must have a limited lifetime, on the order of minutes, although the shorter the better.

#### Password token

A secret that a claimant memorizes and uses to authenticate his or her identity. Passwords are typically character strings, however some systems use a number of images that the claimant memorizes and must identify when presented along with other similar images.

### 2.1.3 Authentication Modes

When authenticating a claimant there are three different modes.

#### Individual

A personal identifier of the claimant is verified in this mode. This leads to the identification of an individual. An example of a personal identifier is a social security number.

#### Identity

A non-personal identifier of the claimant is verified in this mode. This leads to the identification of an Digital Identity only and is not directly linkable to an individual. An email address is an example of such an identifier.

#### Attribute

A attribute of a claimant is verified in this mode. This leads to the identification of some properties of an individual, not his Digital Identity or the individual behind the Digital Identity. Attributes that could be verified are gender and age, for example.

### 2.1.4 Registration and Identity Proofing

In the registration process an applicant undergoes identity proofing by a trusted Registration Authority (RA). If the RA is able to verify the applicants identity, the applicant is given a token and issued a credential as needed to bind that token to the Digital Identity or some related attribute. The applicant may now use the token as a claimant in an authentication protocol.

### 2.1.5 Assurance Levels

Four assurance levels are defined, numbered 1 to 4. Level 4 provides the highest level of authentication assurance, while Level 1 provides the least assurance. The technical requirements for authentication mechanisms (tokens, protocols and security protections) are stated in this section.

#### Level 1

Although there is no identity proofing requirement at this level, the authentication mechanism provides some assurance that the same claimant is accessing the protected transaction or data. It allows a wide range of available authentication technologies to be employed and allows any of the token methods of Levels 2, 3, or 4. Successful authentication requires that the claimant proves through a secure authentication protocol that he or she controls the token.

#### Level 2

Level 2 provides single factor remote network authentication. At Level 2, identity proofing requirements are introduced, requiring presentation of identifying materials or information. A wide range of available authentication technologies can be employed at Level 2. It allows any of the token methods of Levels 3 or 4, as well as passwords and PINs. Successful authentication requires that the claimant prove through a secure authentication protocol that he or she controls the token. Eavesdropper, replay, and on-line guessing attacks are prevented.

#### Level 3

Level 3 provides multi-factor remote network authentication. At this level, identity proofing procedures require verification of identifying materials and information. Level 3 authentication is based on proof of possession of a key or a one-time password through a cryptographic protocol. Level 3 authentication requires cryptographic strength mechanisms that protect the primary authentication token (secret key, private key or one-time password) against compromise by the protocol threats including: eavesdropper, replay, on-line guessing, verifier impersonation and man-in-the-middle attacks. A minimum of two authentication factors is required. In addition to the key, the User must employ a password or biometric to activate the key. Three kinds of tokens may be used: soft cryptographic tokens, hard cryptographic tokens and one-time password device tokens.

#### Level 4

Level 4 is intended to provide the highest practical remote network authentication assurance. Level 4 authentication is based on proof of possession of a key through a cryptographic protocol. Level 4 is similar to Level 3 except that only hard cryptographic tokens are allowed. A minimum of two authentication factors is required.

## 2.2 Identity Management

More and more Web Services require authentication, most of these Web Services have their own authentication mechanism. This requires Users to have a different Digital Identity and token for each Web Service. As a result Users have a hard time remembering their different passwords, an tend to reuse their passwords which is a security threat.

In an attempt to restrict the amount of Digital Identities a number of initiatives have emerged to manage Digital Identities using identity management. In general all these initiatives provide a single sign on service. A claimant authenticates itself to an authentication credential provider which provides the claimant with an assertion. The claimant can then present this assertion to other parties requiring authentication.

### 2.2.1 Identity Metasystem

Microsoft has introduced the identity metasystem. This identity metasystem supports any Digital Identity system and offers consistent User control of Digital Identities. In this subsection we summarize the identity metasystem defined in [10].

Any Digital Identity system may have its own underlying technology, to reason about them the identity metasystem has defined three distinct roles:

- User - The entity which has one or more Digital Identities. Each Digital Identity can contain different information
- Identity Provider - The provider of a Digital Identity. Each provider may provide different levels of assurance that the User is who he claims to be, and may represent this Digital Identity in a different way.
- Relying Party - The application that relies on the Digital Identity, for example for authentication, or the determination of the User's age.

A User might rely on a Service Requester that supports an Identity Selector, to access a Relying Party. The User might also be able to choose from a group of Identity Providers to provide the Digital Identity it wants to present to the Relying Party. The interaction between these parties involves the following phases which are illustrated in 1.

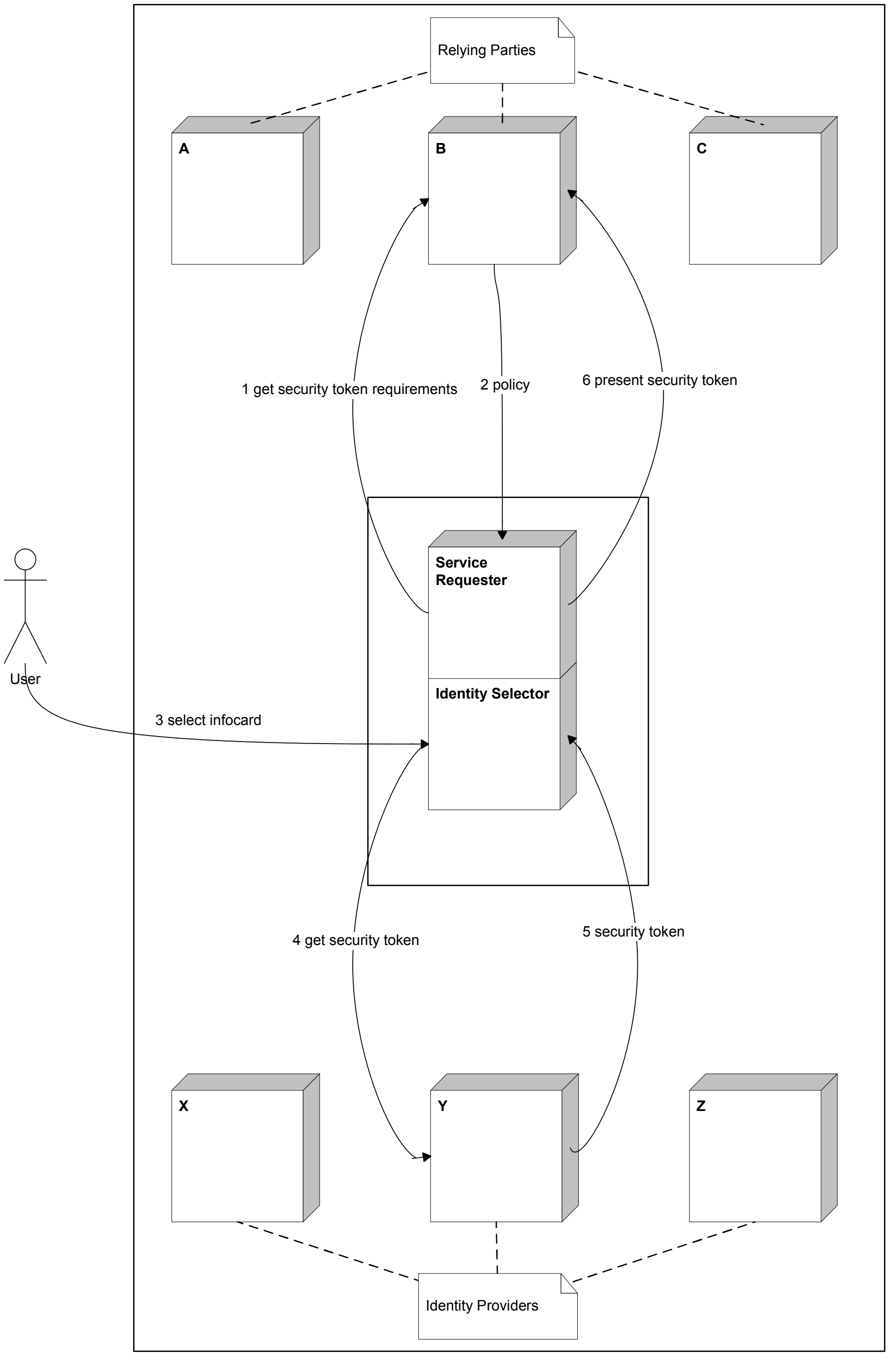

Figure 1: Identity Metasystem

1. The Service Requester requests the security token requirements from the Relying Party that the User wishes to access. This information is contained in the Relying Party's policy, and it includes things such as what security token formats the Relying Party will accept, and exactly what information those tokens must contain.
2. Once the Service Requester has the security token requirements, the Service Requester passes this information to the Identity Selector, asking it to request a token from the appropriate Identity Provider.
3. After this security token has been received, the Identity Selector gives the security token to the Service Requester, which passes the security token on to the Relying Party. The Relying Party can then use this security token to authenticate the User.

This illustrates the Identity Selector and shows the identity metasystem is independent of the format of the security token that is requested. Because of this the identity metasystem can work with any Digital Identity system, using any type of security token. All of the exchanges defined by the identity metasystem are defined using open published protocols. For example a Relying Party's policy is described using WS-SecurityPolicy, the policy is retrieved using WS-MetadataExchange, the security token is acquired using WS-Trust, and that security token is conveyed to the Relying Party using WS-Security. In the case the Service Requester is a browser the policy and security token are expressed in HTML.

### Identity Selector

The Identity Selector is responsible for the communication with the Service Requester and Identity Provider, but also with the User. It presents a interface for working with Digital Identities to the User. Implementations of Identity Selectors are available for platforms such as Windows, Mac and Linux.

### Information Cards

From the point of view of a User, an information card is the visual representation of a Digital Identity that he or she sees on his or her screen. To the metasystem, however, an information card is actually an stored XML document.

Every information card is created by some Identity Provider. A User must acquire the appropriate cards in some way, such as through the Identity Provider's website. How this is done is defined by each identity providerthere's no mandated way to acquire information cards.

The contents of an information card help Users intelligently choose a Digital Identity. They also allow the metasystem to match a card to a Relying Party's requirements, and to acquire an appropriate security token from the Identity Provider that issued this card. The card does not contain sensitive data about this identity. For example, an information card created by a credit card company would not contain the user's credit card number. While this kind of sensitive information might appear as a claim in a security token created by an identity provider, it is always stored at the identity provider's system.

## 2.3 Signing and Non-repudiation

### 2.3.1 Definition

A Digital Signature is a cryptographic means to verify the origin of a document, the Digital Identity of a sender, the time or date a document was sent, and so on. The Digital Signature of a document is a piece of information based on both the document and some signer's information.

One usage of a Digital Signature is for non-repudiation. Non-repudiation is a means to guarantee that a party involved in a transaction cannot deny his involvement. The Digital Signature is proof of the involvement of the party.

For verifying the origin and non-repudiation it is of importance that the Digital Signature can be verified by a third party.

### 2.3.2 Public key cryptography

Digital Signatures can be created by using public key cryptography. In this case the Digital Signature of a document is a piece of information based on both the document and the signer's private key. It is typically created through the use of a hash function and a private signing function. Such a Digital Signature can also be used for non-repudiation purposes if it is created under certain conditions.

### 2.3.3 Issues

To use public key cryptography for signing and non-repudiation there are certain security related issues.

#### Key Pair Generation

The private key should only be known to the signer, so the signer should be the one to generate the private key.

#### Storage of Private Key and Certificates

The private key must be stored in such a way it cannot be copied or accessed without authorization. To use a Digital Signature in a legal case it might be required that the private key is stored in a tamper proof hardware device. It also might be required that the Digital Signature is generated by such a device.s

### 2.3.4 Public Key Infrastructure

In cryptography, a Public Key Infrastructure (PKI) is an arrangement that binds public keys with respective Digital Identities by means of a Certification Authority. The Digital Identities must be unique for each Certification Authority. For each User, the Digital Identity, the public key, their binding, validity conditions and other attributes are made unforgeable in public key certificates issued by the Certificate Authority.

#### X.509

X.509 is an standard for Public Key Infrastructure. X.509 specifies, amongst other things, standard formats for public key certificates and a certification path validation algorithm.

X.509 also includes standards for certificate revocation list (CRL) implementations. Another way of checking a certificate's validity is the Online Certificate Status Protocol (OCSP). The OCSP is an Internet protocol used for obtaining the revocation status of an X.509 digital certificate. Messages communicated via OCSP are usually communicated over HTTP. The "request/response" nature of these messages leads to OCSP servers being termed OCSP responders.

In X.509 certificates the User's Digital Identity in the certificate is called the Subject.

### 2.3.5 Digital Signature Services

The Digital Signature Services (DSS) specifications[20] describe two XML-based request/response protocols: a signing protocol and a verifying protocol. For our thesis we are only interested in the first. Through the signing protocol a client can send documents to a server and receive in return a Digital Signature on the documents. The DSS protocols allow Digital Signatures in may formats and encapsulate them in a DSS Signature Object.

## 2.4 GSM

### 2.4.1 Introduction

GSM is the most widely used cellular standard worldwide. The GSM Association estimates that 82% of the global mobile market uses the standard. Its wide availability makes international roaming very common between GSM Network Operators, enabling Subscribers to use their phones in many parts of the world. GSM differs from its predecessors in that both signaling and speech channels are digital, and so it is considered a second generation (2G) mobile phone system. Data communication was built into the system by the 3rd Generation Partnership Project (3GPP).

The GSM network is designed to be as secure as its landline equivalent, Public Switched Telephone Network (PSTN). This security is beneficial for both the Subscriber and the GSM Network Operator.

### GSM Network Operator

The security is beneficial to the GSM Network Operator because authentication of the Subscriber enables billing, identifying abuse, preventing fraud and un-authenticated network usage.

### Subscriber

The Subscriber benefits from the security because encryption of the air traffic achieves confidentiality and the use of temporary identifiers prevents tracking and linking of the Subscriber and thus adds privacy.

## 2.4.2 Technical details

The GSM network (Figure 2) can be divided into three broad parts:

- The Subscriber carries the Mobile Station
- The Base Station Subsystem(BSS) controls the radio link with the Mobile Station
- The Network and Switching Subsystem (NSS) performs the switching of calls between the mobile users and other mobile and fixed network users, also known as GSM core network

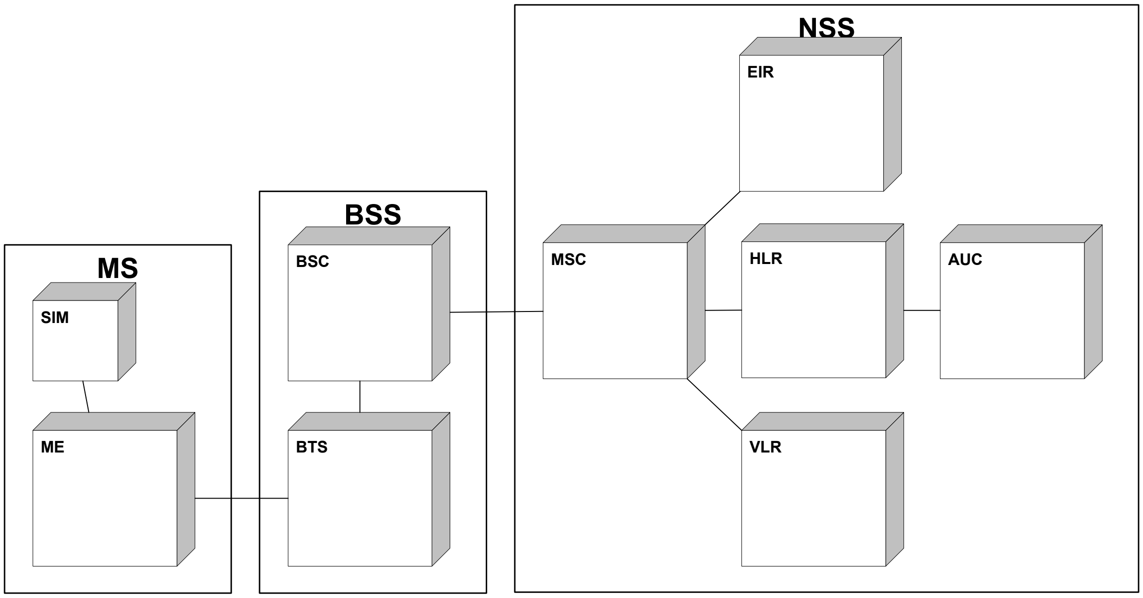

Figure 2: GSM Network Architecture

#### Mobile Station

The Mobile Station consists of the Mobile Equipment, i.e. the handset, and a Smart Card called the Subscriber Identity Module(SIM) card. The SIM provides personal mobility, so that the Subscriber can have access to subscribed services irrespective of a specific terminal. By inserting the SIM card into another handset, the Subscriber is able to access subscribed services from that handset.

The Mobile Equipment is uniquely identified by the International Mobile Equipment Identity(IMEI). The SIM contains the International Mobile Subscriber Identity(IMSI) used to identify the Subscriber to the GSM network, a secret key for authentication and other information. The SIM may be protected against unauthorized use by a password or Personal Identification Number(PIN).

#### Base Station Subsystem

The Base Station Subsystem is composed of two parts, the Base Transceiver Station(BTS) and the Base Station Controller(BSC).

The BTS houses the radio transceivers that define a cell and handles the radio-link protocols with the Mobile Station. In a large urban area, there will potentially be a large number of BTSs deployed. The BSC manages the radio resources for one or more BTSs. It is the connection between the Mobile Station and the Mobile Switching Centre(MSC).

#### Network and Switching Subsystem

The central component of the Network and Switching Subsystem is the MSC. This acts like a normal switching node of the PSTN or Integrated Services Digital Network (ISDN) and connects the mobile signal to these fixed networks.

The Home Location Register (HLR) is a central database that contains details of each Subscriber that is authorized to use the GSM network.

The Authentication Centre (AuC) is a function to authenticate each SIM that attempts to connect to the GSM network. Once the authentication is successful, the HLR is allowed to manage the SIM and provide subscribed services. An encryption key is also generated that is subsequently used to encrypt all air traffic between the Mobile Station and the GSM core network.

The Visitor Location Register (VLR) is a temporary database containing the Subscribers who have roamed into the particular area which it serves. Each BSS in the network is served by exactly one VLR, hence a Subscriber cannot be present in more than one VLR at a time.

### 2.4.3 Security

As we briefly explained in the introduction GSM has several security areas.

#### Subscriber authentication

Before the Subscriber can access any of the Mobile Services, the Subscriber must authenticate to the GSM network. We focus on this security area and we will discus further details in 2.4.4.

#### Data confidentiality

Both the signaling and Subscriber data are transmitted encrypted over the radio interface. The encryption key is agreed upon during the Subscriber authentication.

#### Privacy

The IMSI is only transmitted over the air when absolutely necessary, otherwise a temporary identifier(TIMSI) is used. This temporary identifier is assigned after Subscriber authentication and transmitted in encrypted form to the Mobile Station.

### 2.4.4 Authentication Components

The authentication of the Subscriber in the GSM network has our main interest. The authentication of the Subscriber involves the SIM, Mobile Equipment and GSM network.

#### SIM

A Smart Card is a Integrated Circuit Card (ICC), containing an operating system, file system and stand alone applications.

In the 2G technical specification[11] the SIM is defined as an ICC, so it is both a physical(SIM card) and a logical entity(SIM). In 3G the Universal Integrated Circuit Card (UICC) is introduced, the UICC may contain a SIM and USIM application, so the physical and logical entities become separated.

The SIM stores information of three types. Subscriber related information such as IMSI, authentication key, PIN, language preferences and personal phone numbers. It also stores network related information such as service phone numbers and advice service charges. It also stores temporary information such as the encryption key and TIMSI. The authentication key is denoted as $K_i$ and the encryption key is denoted as $K_c$.

Next to storing information the SIM also implements several algorithms. $A3$ for generating responses to challenges and $A8$ for generating encryption keys from the same challenges.

The SIM may also contain some custom applications provided by the GSM Network Operator or a third party.

International Mobile Subscriber Identity

The IMSI is only used to identify the Subscriber, it contains the mobile country code and is followed by the mobile network code. The rest of the digits are the mobile subscriber identification number which uniquely identifies the Subscriber within the GSM Network Operator's customers base.

Subscriber authentication key

The key $K_i$ is used for authentication of the Subscriber by the GSM Network Operator. The security of GSM depends upon this shared secret between the AuC and the SIM. The $K_i$ is securely put into the SIM during manufacture and it is also securely replicated into the AuC. The $K_i$ can be any format and length. This $\mathrm{K}_i$ is never transmitted between AuC and SIM, but it is used with the IMSI and a random generated number to produce a challenge/response for identification purposes and the generation of encryption key $K_c$ for use in over the air communications.

Authentication algorithm

Algorithm $A3$[2] is a one-way function and is implemented by the SIM and by the AuC. It is used in the challenge-response mechanism of the Subscriber authentication.

Cipher key generation algorithm

The $A8$[2] algorithm is another one-way function used in establishing the encryption key $K_c$ used for encrypting air traffic. The encryption key $K_c$ is used by the algorithm $A_5$ which is of no further relevance during authentication.

PIN

The PIN is also commonly referred to as Card Holder Verification (CHV). It is an 4 to 8 digit code required by the SIM to authenticate the Subscriber. The Subscriber is able to remove this requirement.

### Mobile Equipment

The Mobile Equipment contains one algorithm $A5$[2], this algorithm is used to encrypt and decrypt data send and received over the air interface. The key $K_c$ is used by this algorithm.

### GSM Network

The MSC is the network node which authenticates the Subscriber. To authenticate the Subscriber it uses triplets generated by the AuC, these triplets are stored in the HLR or VLR. The triplets consist of the encryption key $K_c$, a random challenge $RAND$ and a response $SRES$ which is dependent on $RAND$. The MSC is responsible for comparing the response $SRES$ with the response returned by the Mobile Station $SRES'$.

## 2.4.5 Authentication schemes

### Subscriber - SIM

Before the SIM can authenticate to the network the Subscriber must first authenticate to the SIM. This authentication is done by entering the PIN on the Mobile Equipment which relays the PIN to the SIM.

This authentication is a protection against the use of stolen cards. If the verification of the PIN fails, two more attempts to enter the correct PIN may be made. After three attempts the PIN and the SIM become locked. The SIM can only be unlocked by entering the Unblock CHV, this is an 8 digit key also known as Personal Unblocking Key (PUK). The SIM becomes permanently blocked if the PUK is wrongly entered 10 times.

The Subscriber is allowed to change his PIN or completely remove the protection. The PIN is only known to the SIM and Subscriber, the PUK is also known by the GSM Network Operator.

### SIM - MSC

After the Subscriber has authenticated to the SIM the SIM authenticates to the MSC.

In order for the MSC to authenticate the SIM it needs a triplet generated by the Subscribers home AuC. The SIM contacts the MSC and identifies itself by sending the IMSI, with the IMSI the MSC can determine the home network of the Subscriber. The MSC obtains a triplet in one of the following ways:

- The Subscriber is authenticating to a home network MSC, the MSC contacts the HLR and requests a triplet. If the HLR has an unused triplet it returns the triplet. Otherwise the HLR contacts the AuC, which generates one ore more triplets, and returns them to the HLR which in turn returns one triplet to the MSC.
- The Subscriber is not authenticating to a home network MSC, the MSC contacts the VLR and requests a triplet. If the VLR has an unused triplet it returns the triplet. Otherwise the MSC contacts the Subscribers home network HLR and asks it to send a triplet.

Once the MSC has a triplet consisting of $SRES$, $RAND$ and $K_c$ the MSC forwards the $RAND$ to the Mobile Station. The SIM calculates its response $SRES'$ using $A3$ and the

Mobile Equipment sends the response back to the MSC. The MSC compares $SRES$ from the triplet with the $SRES'$ from the response. If these are equal the SIM is authenticated to the network.

The implementation of $A3$ is GSM Network Operator specific, this means that each GSM Network Operator may implement its own algorithm. Since this algorithm is used in the SIM and in the home AuC this has no influence on roaming.

### 2.4.6 Considerations

In practice the algorithms $A3$ and $A8$ are combined in a function called COMP128. The first version of this algorithm is proven to be flawed. It is possible to retrieve the key $K_i$ by a chosen plaintext attack, such an attack requires physical access to the SIM. COMP128 v.1 is not considered safe anymore, however some GSM Network Operators may still be using this algorithm even though alternatives are available.

The queries between MSC, HLR, VLR and AuC are handled by Mobile Application Part (MAP). MAP doesn't use any cryptographic mechanisms, so all information is carried in plaintext. This includes the GSM triplets, thus it is basically possible for an intruder who has a physical access to the signalling link and any intermediate operator to read or modify them.

The implementation of the algorithms $A3$ and $A8$ may be GSM Network Operator specific, but the input parameter $RAND$ is fixed at 128 bits. Since the algorithm $A5$ is fixed, its input $K_c$, is also fixed. This 64 bits $K_c$ is the output of $A8$ and so also the output size of $A8$ is fixed. The $SRES$ is fixed at 32 bits. The size of the encryption key $K_c$ and response $SRES$ might be considered too small with regard to the currently available computing power.

During authentication the SIM authenticates to the GSM network, however the GSM network does not authenticate to the SIM. This leads to the possibility of a man in the middle attack. This party would be able to violate the privacy of the Subscriber by intercepting the IMSI, the party would also be able to intercept any signalling and data.

The possibility of removing the PIN protection from the SIM might be a security risk as this reduces the two factor authentication to single factor authentication.

Researchers have showed how $A5$ could be compromised in theory, however even if a practical attack was possible this would not have any influence on the security of the authentication schemes.

## 2.5 EAP-SIM

### 2.5.1 Extensible Authentication Protocol

The Extensible Authentication Protocol (EAP), is a universal authentication framework frequently used in wireless networks and point-to-point connections. It is defined in RFC 3748 [17].

EAP is an authentication framework, not a specific authentication mechanism. The EAP provides some common functions and a negotiation protocol for the desired authentication mechanism. Such mechanisms are called EAP methods and there are currently about 40 different methods.

EAP is not a protocol; instead it only defines message formats. Each protocol that uses EAP defines a way to encapsulate EAP messages within that protocol's messages.

### 2.5.2 EAP-SIM

EAP-SIM is an authentication mechanism for the EAP framework. It is defined in RFC 4186[24].

The EAP-SIM mechanism specifies enhancements to GSM authentication and key agreement whereby multiple authentication triplets can be combined to create authentication responses and session keys of greater strength than the individual GSM triplets. The mechanism also includes network authentication, user anonymity support, result indications, and a fast re-authentication procedure.

#### Overview

An authenticator is an entity that requires authentication from the peer.

A peer is an entity that is being authenticated by an authenticator.

The authenticator initiates the authentication by sending a EAP-Request/Identity to the peer. The peer responds with an EAP-Response/Identity including the IMSI.

Following this response the authenticator sends a EAP/Request/SIM/Start including a list of all EAP-SIM versions it supports, $AT\_VERSION\_LIST$. The peer chooses a EAP-SIM version, $AT\_VERSION$, which both the peer and the authenticator support, generates a random number $NONCE\_MT$ and responds to the packet with the EAP-Responce/SIM/Start packet. This packet includes the $NONCE\_MT$ and $AT\_VERSION$.

After receiving the EAP-Response/SIM/Start the authenticator obtains $n$ GSM triplets, where $2 <= n <= 3$. Using the triplets the authenticator sends an EAP-Request/Challenge which contains the $n$ $RAND$s and a Message Authentication Code (MAC) $AT\_MAC$ over these $RAND$s. The peer runs its own authentication algorithms with the $RAND$s as input. Calculates its own $AT\_MAC$ over the output of the authentication algorithms and compares this with the received $AT\_MAC$. If the MACs do not match, the authentication exchange terminates. Otherwise the peer responds with the EAP-Response/SIM/Challenge, containing $AT\_MAC$ which is a MAC over the peer's $n$ $SRES$ response values.

The authenticator computes a MAC over the $n$ received $SRES$ values and compares this MAC with the $AT\_MAC$ received from the peer. If they are equal the authenticator sends an EAP-Success.

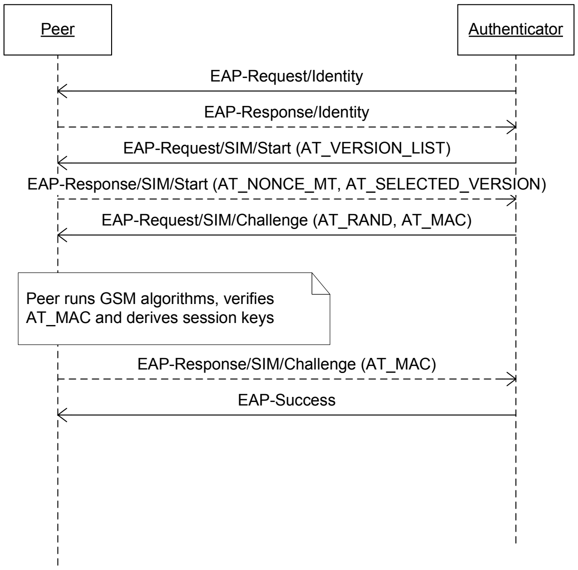
Peer
Authenticator
EAP-Request/Identity
EAP-Response/Identity
EAP-Request/SIM/Start (AT_VERSION_LIST)
EAP-Response/SIM/Start (AT_NONCE_MT, AT_SELECTED_VERSION)
EAP-Request/SIM/Challenge (AT_RAND, AT_MAC)
Peer runs GSM algorithms, verifies AT_MAC and derives session keys
EAP-Response/SIM/Challenge (AT_MAC)
EAP-Success

Figure 3: EAP-SIM

### Key Generation

On EAP-SIM authentication as described above, a Master Key ($MK$) is derived from the underlying GSM encryption values $K_c$, the nonce $NONCE\_MT$, and other relevant information as follows.

$$MK=\text{SHA1}(IMSI|n{*}K_c|NONCE\_MT|AT\_VERSION\_LIST|AT\_VERSION)$$

The $MK$ is used to calculate the MAC value over the EAP-SIM messages, this enables mutual authentication between the parties. When the peer retrieves an EAP-SIM message from the authenticator, covered by the MAC, it can compute its own version of the MAC and compare with the MAC included in the EAP-SIM message. I.e. the peer is able to verify that the EAP-SIM message retrieved is not a replay because the master key contains $NONCE\_MT$, and that the authenticator possesses valid GSM triplets, since the $K_c$s are concatenated in the master key.

### Security considerations

Several security considerations are discussed in [24], we summarize the most relevant ones for this thesis.

#### $A3$ and $A8$ Algorithms

The GSM $A3$ and $A8$ algorithms are used in EAP-SIM. The operation of these functions falls completely within the domain of an individual GSM Network Operator, and therefore, the functions are specified by each GSM Network Operator rather than being fully standardized. As mentioned earlier in 2.4.6, those algorithms may not be safe, therefore the EAP-SIM authentication will also be unsafe in this case.

#### Mutual Authentication and Triplet Exposure

EAP-SIM provides mutual authentication. The peer believes that the GSM Network is authentic because the network can calculate a correct $AT\_MAC$ value in the EAP-Request/SIM/Challenge packet. To calculate $AT\_MAC$ it is sufficient to know the $RAND$ and $K_c$ values from the GSM triplets used in the authentication. Because the network selects the $RAND$ challenges and the triplets, an attacker that knows $n$ GSM triplets for the Subscriber is able to impersonate a valid GSM Network to the peer. In other words, the security of EAP-SIM is based on the secrecy of $K_c$ keys, which are considered secret intermediate results in the EAP-SIM cryptographic calculations. Given physical access to the SIM card, it is easy to obtain any number of GSM triplets.

### 2.5.3 SOAP

EAP typically runs directly over data link layers such as Point-to-Point Protocol (PPP) or IEEE 802, without requiring IP.

However because EAP is not a protocol but a message format it can be encapsulated to be run over any protocol. Since our main concern is accessing services over the internet, an encapsulation within IP would be desirable. Because we are accessing Web Services it might be even more appropriate to encapsulate the EAP messages in HTTP, which also leverages some potential problems with firewall and proxy mechanisms.

In 'An Experimental Study of a Business Domain Independent Application Level and Internet Access Authentication and Authorization Concept'[28] the concept of utilizing SOAP as the transport medium for EAP messages is introduced. SOAP makes use of an internet application layer protocol as a transport protocol. HTTP as such a protocol has gained wide acceptance for transporting SOAP.

## 2.6 Java ME

### 2.6.1 Introduction

Java Micro Edition (Java ME) is a specification of a subset of the Java platform aimed at providing a certified collection of Java APIs for the development of software for small, resource-constrained devices such as cell phones, PDAs and set-top boxes.

Java ME has become a popular option for creating games and applications for cell phones, as they can be emulated on a PC during the development stage and easily uploaded to the phone. Java ME devices implement a profile, profiles are subsets of configurations.

#### Connected Limited Device Configuration

The Connected Limited Device Configuration(CLDC) is aimed at Mobile Equipment, such as cell phones, it contains a strict subset of the Java class libraries, and is the minimal needed for a Java virtual machine to operate.

A configuration provides the most basic set of libraries and virtual-machine features that must be present in each implementation of a Java ME environment. When coupled with one or more profiles, the CLDC gives developers a solid Java platform for creating applications for consumer and embedded devices.

#### Mobile Information Device Profile

Designed for cell phones, the Mobile Information Device Profile boasts GUI API, and MIDP 2.0 includes a basic 2D gaming API. Applications written for this profile are called MIDlets. Almost all new cell phones come with a MIDP implementation, and it is now the de facto standard for downloadable cell phone games.

#### Optional Java Specification Requests

Mobile Equipment may implement additional Java Specification Requests(JSR) to provide extra functionality on the handset.

### 2.6.2 Security Domains

Accessing certain method calls and APIs from MIDlets has some restrictions. It is possible that in those cases the user will get prompted for confirmation to allow the certain method call or the access might be blocked altogether, which will result in an SecurityException to be thrown.

#### Security domains

MIDP 2.0 specification defines 4 security domains in which the MIDlet can be installed:

- third party protection domain (untrusted 3rd party)
- identified third party protection domain (trusted 3rd party)
- operator protection domain
- manufacturer protection domain

An application may be signed, by signing an application it will be placed in a certain security domain. The security domain is determined by the identity of the signer. For example, if the application is signed by the GSM Network Operator it is placed in the operator protection domain. Applications in the operator protection domain will be allowed to use all method calls without confirmation and security exceptions.

### 2.6.3 SATSA

The Security and Trust Services API(SATSA) specification defines optional packages for Java ME. The specification has been produced in response to Java Specification Request 177 (JSR-177). This JSR specifies a collection of APIs that provides security and trust services by integrating a Security Element (SE). A SE, a component in a Java ME device, provides the following benefits:

- Secure storage to protect sensitive data, such as the user's private keys, public key (root) certificates, service credentials, personal information, and so on.
- Cryptographic operations to support payment protocols, data integrity, and data confidentiality.

- A secure execution environment to deploy custom security features. Java ME applications would rely on these features to handle many value-added services, such as user identification and authentication, banking, payment, loyalty applications, and so on.

A SE can be in a variety of forms. Smart Cards are commonly used to implement a SE. They are widely deployed in wireless phones, such as SIM cards in GSM phones, UICC cards in 3G phones, and RUIM cards in CDMA phones.

The API in JSR-177 is defined in four optional packages that can be implemented independently. The two packages in which we are interested are SATSA-APDU and SATSA-PKI.

### SATSA-APDU

The SATSA-APDU optional package defines an API to support communication with Smart Card applications using the Application Protocol Data Unit (APDU) protocol.

An APDU is a short message represented by bytes. SATSA-APDU enables an application to exchange APDU messages with a Smart Card card application.

In SATSA-APDU, messages are either commands or responses. An application can use SATSA-APDU to send commands to a Smart Card application and receive responses. An application could send a command message to request the IMSI and receive the IMSI in the response message.

MIDP 2.0 limits access to these API's to the operator and manufacturer domain.

### SATSA-PKI

SATSA-PKI optional package defines an API to support application level Digital Signature signing and basic user credential management(for example, for certificates).

It allows management of certificates stored on a device (Smart Card) and provides the process for getting the appropriate keys stored on the device for encryption. It enables generation of certificate requests and to locally register user credentials. The user credentials are used in conjunction with other parameters to compute formatted Digital Signatures.

When deployed on the MIDP 2.0 platform, the package also includes the interface Certificate (for providing a common interface to (X.509) certificates) and CertificateException (for errors occurring with Certificate), which are defined in the MIDP 2.0 API.

The package allows generation of application-level Digital Signatures using certificates stored on the device (Certificate Store) that conform to the Cryptographic Message Syntax (CMS) [26] format.

The CMS is the IETF's standard for cryptographic protected messages. It can be used to digitally sign, digest, authenticate or encrypt any form of digital data. CMS is based on the syntax of PKCS#7 [29].

### 2.6.4 Push Registry

In MIDP 1.0 there is only one way to launch a MIDlet, manual activation by the user. The MIDP 2.0 specification adds two new mechanisms to launch a MIDlet: in response to an incoming connection or at a scheduled time. These mechanisms are called Push Registry.

### 2.6.5 Web Services

The Web Services API specification defines two optional packages for Java ME. The specification has been produced in response to JSR-172.

The Web Services package facilitates access to remote SOAP / XML based web services from CDC and CLDC based profiles. The XML Parsing package is the second package in this API, it adds XML Parsing support to the platform.

# 3 Analysis

## 3.1 Requirements

In order to propose a solution the exact requirements for this solution have to be known. We have formulated the following requirements.

### 3.1.1 Functional Requirements

The functional requirements describe what the solution should be able to do.

#### Internet Protocol

The proposed solution shall be able to operate on top of IP. The services requiring secured access are Web Services, the protocol used on the web is IP, so the solution must be usable over IP.

#### Compatibility

The proposed solution shall be independent of the GSM Network Operator and Mobile Equipment manufacturer. A primary goal is to develop a solution which is usable by a wide audience, this means it can't be bound to a single manufacturer or be unavailable when roaming between networks.

#### Authentication

Our requirement for authentication of the User is similar to Level 4 as we discuss in 2.1.5. The Relying Party shall be able to authenticate the User by multi-factor authentication, the User shall be able to authenticate the Relying Party. The access to the Web Service must be secured, therefore it shall be possible to verify the Digital Identity of the User. Also the User will need to be able to verify the Digital Identity of the Relying Party to prevent a man in the middle attack. It shall be possible to prevent the authentication of a Digital Identity with a certain credential, this prevents the use of a lost or stolen credential for verification. To prevent a denial of service attack only a trusted party and the User shall be able to revoke the credential. The number of successive times the User may retry to authenticate after authentication with a false credential shall be limited.

#### Integrity

The integrity of the data sent between Service Requester and Relying Party shall be guaranteed. This assures both parties that no-one has modified, inserted or deleted data.

### Non-repudiation

It shall be possible to complete a transaction in such a way that the User cannot repudiate he has agreed on the transaction. With this property a User can be held accountable for his actions. A third party is able to verify transactions made between the User and Relying Party, this verification should be possible at any point in time after the transaction has taken place.

### Privacy

The identity of the User accessing a service shall remain confidential. Only the Relying Party and Service Requester shall be able to learn the identity. This adds to the privacy of the User, because no other party can see which User uses a particular Relying Party.

It shall be possible to perform multiple transactions provided by different Relying Parties without the Relying Parties being able to link the transactions. If multiple Relying Parties collude they will not be able to identify if a User has made transactions with both Relying Parties, this reduces the possibility of data-mining and thus increases privacy.

It shall also be possible to perform a transaction with a Relying Party without the Relying Party being able to link the transaction to a natural person. The privacy of the User is guarded because the Relying Party cannot resolve the Digital Identity to a natural person.

### Confidentiality

The confidentiality of the data sent between Service Requester and Relying Party shall be guaranteed. An eavesdropper will not be able to listen in on a transaction between the User and a Relying Party.

### Authorization

The Relying Party shall be able to authorize the User to access a Web Service. It shall be possible to de-authorize a User. The Relying Party must be able to differentiate between different Users, and therefore will be able to personalize its Web Services, or deny service to abusers or Users with expired contracts.

### Flexibility

A Mobile Station shall be able to run multiple Service Requesters concurrently. These Service Requesters might be acting on behalf of different Users. This allows the User to use multiple Web Services at the same time or for multiple Users to use the same Mobile Station at the same time.

A User may use Service Requesters on different Mobile Station's concurrently. Using multiple phones at the same time adds more flexibility for the User.

### 3.1.2 Quality Requirements

The quality requirements describe what properties the solution must satisfy.

#### Number of passwords

The number of passwords a User has to remember shall be limited. It is hard for a User to remember many passwords: a User is prone to forgetting and interchanging them. Therefore the Users may reuse them which creates a security risk.

#### Password input

The number of times the User has to input a password per transaction shall be limited. The usability of the solution is greatly reduced if the passwords are asked too often. However not asking them at the right time creates a security risk.

#### User interaction

The User interaction shall be limited and clear. Usability is an important requirement: the solution becomes less usable if too many, irrelevant or vague questions are asked.

#### Trusted third party

The introduction of a new trusted third party shall be avoided. Introducing a trusted third party is costly and raises new trust issues.

#### Hardware

The Mobile Equipment shall not require new hardware components. The introduction of new Mobile Equipment hardware means there will be less devices on which the solution will be available and may add considerable costs.

#### Software

If additional software is required on the Mobile Equipment or Smart Card, this software shall be able to be installed after purchase. The times new software shall be needed will be limited.

#### Availability

The availability of the access to the Web Service by a Mobile Station shall be high. Users and businesses may come to rely on the Web Services provided by the Relying Party, so the Web Services should be available at any time.

## 3.2 Use Cases

With these requirements we can formalize the following use cases. A graphical representation of these use cases is show in figure 4.

### 3.2.1 User

#### Obtaining a credential for a Digital Identity

Before the User can authenticate to the Relying Party he must first obtain credentials by which the Relying Party can verify the Users Digital Identity.

#### Removing a credential

When the User decides not to use a particular credential anymore it may remove the credential to prevent abuse by another party.

#### Reporting loss of a credential

If the User loses possession or control of one of its credentials he must take steps to avoid usage of the credential in question.

#### Authenticating a Relying Party

The User authenticates the Relying Party when the User wants to use a Web Service.

#### Authenticating to Relying Party

The User authenticates to the Relying Party when the User wants to use a Web Service.

#### Committing to a transaction

The User commits to a transaction.

### 3.2.2 Relying Party

#### Authorizing a User

The Relying Party authorizes a User to access a Web Service.

#### De-authorizing a User

The Relying Party de-authorizes a User to access a Web Service.

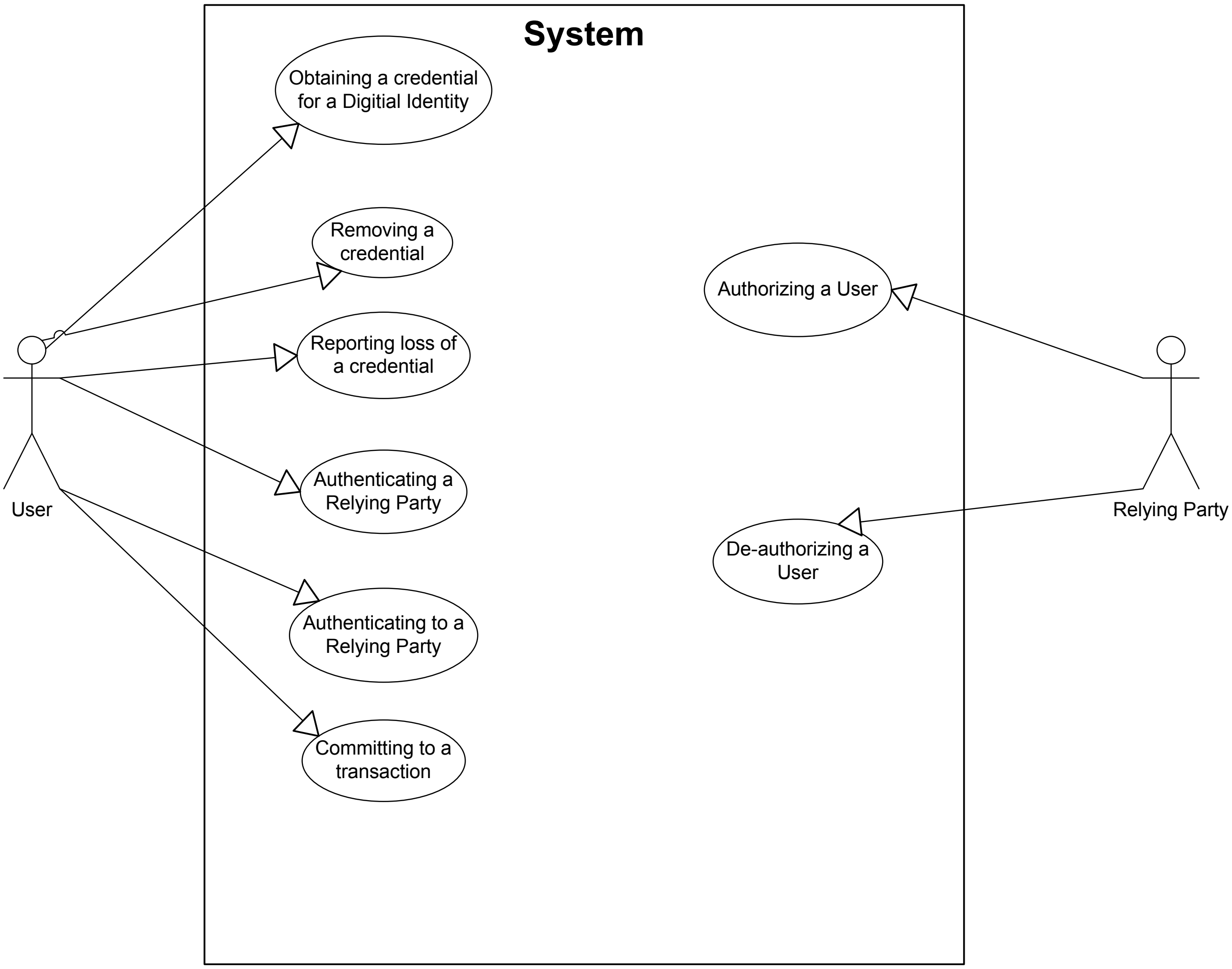
System
Obtaining a credential for a Digitial Identity
Removing a credential
Reporting loss of a credential
Authenticating a Relying Party
Authenticating to a Relying Party
Committing to a transaction
Authorizing a User
De-authorizing a User
User
Relying Party

Figure 4: Use Cases

## 3.3 Existing Solutions

### Password

Many Web Services today are protected with a password, this solution in general has several drawbacks. We already touched upon the problem of remembering multiple passwords but also phishing is a risk. Phishing is an attempt to acquire sensitive information, such as usernames and passwords, by masquerading as a trustworthy entity. The limited keyboard features on many Mobile Stations add another device specific problem to this solution. Furthermore since the password is a shared secret between the User and a Relying Party there is no way achieve non-repudiation.

### One-Time Password

A stronger authentication method for authentication is the use of a One Time Password(OTP). Such a password can be generated by a hardware device. This requires the User to carry an additional device, potentially one device per Relying Party.

An application on the Smart Card present in the Mobile Station could be used to generate the OTP. An alternative is letting the Relying Party generate the OTP and sending it to the User out of band. This requires an additional communication channel such as SMS. In both cases the password has to be entered on the handset with a (poor) keyboard. Again the OTP is a shared secret and thus isn't suitable for non-repudiation.

### Public Key Infrastructure

Wireless PKI

For the Wireless Application Protocol (WAP) a Wireless PKI[4] (WPKI) is defined. WAP also has defined its own Transport Layer Security (TLS) called WTLS[7]. With WTLS User authentication is possible because of the support of client side certificates. The security functions and information needed for identification and authentication are provided by the Wireless Identity Module (WIM). The WIM is defined as an Smart Card application, it can be implemented on a separate Smart Card or next to the SIM on a ICC.

WAP also defines the Wireless Markup Language[6] (WML) which has its own Wireless Markup Language Script[3] (WMLScript). This client-side scripting language defines a cryptographic library [8] to create a Digital Signature *Crypto.signText().* This Digital Signature could be used for non-repudiation. The certificates and private keys used to create the Digital Signatures are stored in the Wireless Identity Module[5] (WIM) a application on a Smart Card.

As promising as the features look several problems stopped WAP from getting widely accepted. The main reason for this is that most of the wireless protocols like WML and WTLS were not compatible with their widely used non-wireless companions HTML and TLS. In the original design WTLS even was seriously flawed by not offering end to end security. With the incompatible WML also WMLScript didn't take off. As additional problem *Crypto.signText()* was supposed to show the text to be signed, however this interface was different in every browser which was confusing to the User.

Because WAP didn't take off the WPKI isn't usable at this point. If WAP had taken of this would have been a possible solution. Both authentication and non-repudiation could have been offered by client side certificates. The inconsistent user interface problem could have been overcome by cooperation of GSM Manufacturers.

PKI Client

Several initiatives have risen to deploy PKI on the mobile. Finland has deployed its Citizen Certificate which can be attached to the SIM card of a Mobile Station [12]. Also has recently deployed the "Mobile Signature" WPKI service for Telefónica Móviles in Spain. This solution is offered to the enterprizes customers of Telefónica Móviles in order for them to generate transactions to be signed by the employees and the collaborators of these companies. Users are receiving and signing the transactions with the SIM.

Both these solutions are solutions in a restricted domain where the PKI is customized to the situation. This results in custom ways of provisioning the certificates, limited number of Digital Identities(certificates) and limited usage of the certificates(authentication or signing only). The solutions are particularly unsuitable for our purposes because certificates are distributed physically which is to bothersome for Users and using the same certificate for authentication and signing is a security risk.

### EAP-SIM Authentication

Over the last years several ways to authenticate using the SIM have been proposed. Most of them attempt to leverage authentication of a pc by using a Mobile Station for example [31] and [32]. In [33] and [25] the idea of using EAP-SIM for authentication to Web Service from the Mobile Station is further researched and tested with a proof of concept.

These papers conclude that strong authentication using EAP-SIM is possible, however this solution is not sufficient for our goals. Using EAP-SIM does not provide Digital Signatures an therefore can not provide non-repudiation. Using this method for authentication does not allow multiple Digital Identities for unlinkability.

Table 1: Comparison of existing solutions

| Solution | User authentication | non-repudiation | unlinkability |
|---|---|---|---|
| Password | weak | no | yes |
| OTP | strong | no | yes |
| WPKI | strong | yes | yes |
| PKI Client | strong(n/a with non-repudiation) | yes(n/a with authentication) | yes |
| EAP-SIM | strong | no | no |

# 4 Architecture

The purpose of this section is to describe the proposed architecture.

## 4.1 Identity Metasystem

The architecture is an extension to the identity metasystem proposed by Microsoft.

The identity metasystem is a way to manage several Digital Identities from different identity management systems. Information about each Digital Identity is stored in a information card. The metasystem is supports many different types of Digital Identities and can easily be extended to support an additional kind of Digital Identity.

There are Identity Selectors available for Windows, Macintosh an Linux so a Identity Selector for the Mobile Station is a logical step. By integrating our solution in that Identity Selector, the platform for our solution can be easily expanded.

The Identity Selector described in the identity metasystem is a convenient way for the User to select his Digital Identity. By integrating into this Identity Selector we reuse this usability.

### CMS Token Profile

The identity metasystem works by exchanging security tokens with Relying Parties and Identity Providers. The currently supported tokens are SAML tokens as defined by OASIS in [22]. A SAML token contains assertions about the User, made and signed by the Identity Provider.

Our solution requires Digital Signatures made by the User. SAML tokens can not provide this. The OASIS standard defines three more token profiles, two of them are also not suitable for our solution because they fail to provide digital signatures.

The third is the X.509 Certificate Token Profile. This token contains a X.509 certificate or a reference to such a certificate. This token may also contain some additional data. The token may contain a Digital Signature in the XML-Signature format [21]. Unfortunately the SATSA-PKI package is only able to generate Digital Signatures that conform to the Cryptographic Message Syntax(CMS) format as specified in RFC 2630[26]. The CMS format is incompatible with the XML-Signature format and therefore also the X.509 Certificate Token Profile is unsuitable. The main reason the formats are incompatible is that in XML signature, the actual input to the signature algorithm is a XML structure which contains the references and hashes of the signed content. The actual contents are not input to the signature algorithm. Similarly, in PKCS#7, the actual input for signing is usually a ASN.1 structure.

As a result of the lack of a suitable token profile we have defined our own token profile, the CMS Token Profile. A token conforming to this profile contains only a CMS signature. This CMS signature must contain both the content and the certificate.

The reason for not using TLS client side certificates as alternative to security tokens in this case is twofold. The first is the lack of support for client side certificates on many of the Mobile Stations. The second is the lack of privacy TLS offers for certificate exchange. The exchange of certificates in TLS takes place before encryption is enabled. This means the User and Relying Party certificate are being sent in plaintext. Sending the Relying Party certificate is not a big deal as anyone who connects to the Relying Party will be able to see the certificate. Revealing the User's Digital Identity to a third party however violates our privacy requirement. It is possible to first set-up encryption and then do the certificate exchange, this is however not standard in TLS and requires one of the sides to request re-negotiation. This would require changes in either all Mobile Equipment HTTP clients or all Relying Party HTTP servers.

## 4.2 Credential Management

The creation of a CMS signature requires a X.509 certificate and the accompanying private key.

With SATSA the Mobile Station is able to generate and store a User's public and private key. From these keys the Mobile Station is able to create a Certificate Signing Request(CSR). For the Mobile Station to obtain an X.509 certificate this CSR must be processed by a Certification Authority.

In order for the Certification Authority to sign the X.509 certificate contained in the CSR it must verify the identity of the User who requested the certificate. Our solution achieves this authentication to the Certification Authority trough the use of EAP-SIM. In A Unified Authentication Solution for Mobile Services[25] EAP-SIM is used for strong authentication to a Relying Party. We propose to use it to authenticate to the Certification Authority.

After receiving the certificate from the Certification Authority the Mobile Station can create an information card for the use of the new credential in the identity metasystem.

# 5 Design

## 5.1 Use Cases

With this architecture our use cases can be refined.

Figure 5 illustrates the new set of use cases for the User. In the proposed architecture four additional use cases have been defined, these use cases are specific to the chosen solution.

### 5.1.1 Authenticating a Certification Authority

The User authenticates the Certification Authority before any of the other use cases.

### 5.1.2 Authenticating to a Certification Authority

After authenticating the Certification Authority the User authenticates to the Certification Authority.

### 5.1.3 Registering additional Subscriber

By registering an additional Subscriber the User can access his Digital Identities from another subscription.

### 5.1.4 Removing Subscriber

By removing a Subscriber the User prevents management of his Digital Identities from this subscription.

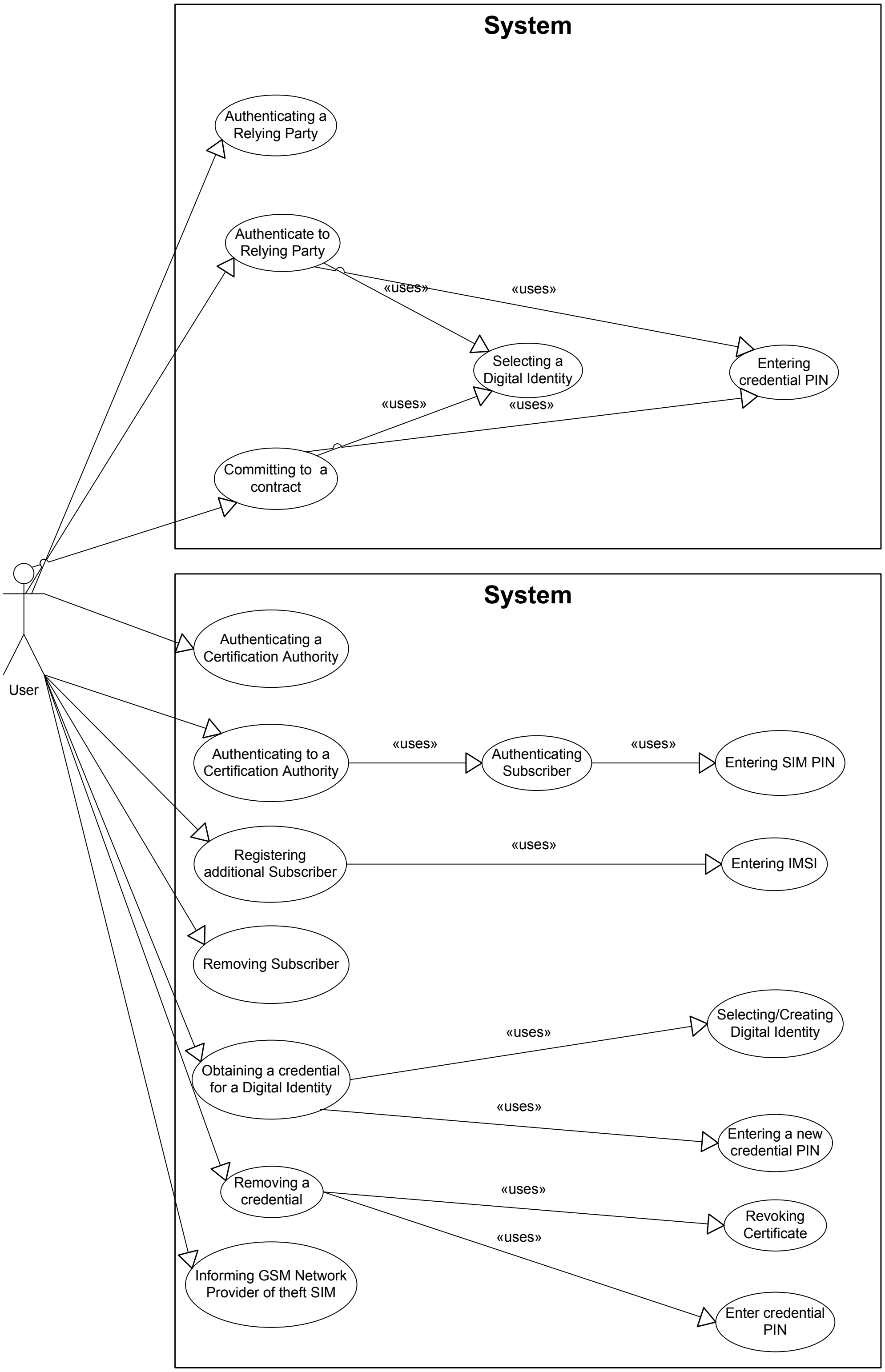
System
Authenticating a Relying Party
Authenticate to Relying Party
«uses»
«uses»
Selecting a Digital Identity
Entering credential PIN
«uses»
«uses»
Committing to a contract
System
User
Authenticating a Certification Authority
Authenticating to a Certification Authority
«uses»
Authenticating Subscriber
«uses»
Entering SIM PIN
Registering additional Subscriber
«uses»
Entering IMSI
Removing Subscriber
Selecting/Creating Digital Identity
«uses»
Obtaining a credential for a Digital Identity
«uses»
Entering a new credential PIN
Removing a credential
«uses»
Revoking Certificate
«uses»
Informing GSM Network Provider of theft SIM
Enter credential PIN

Figure 5: User Use Cases

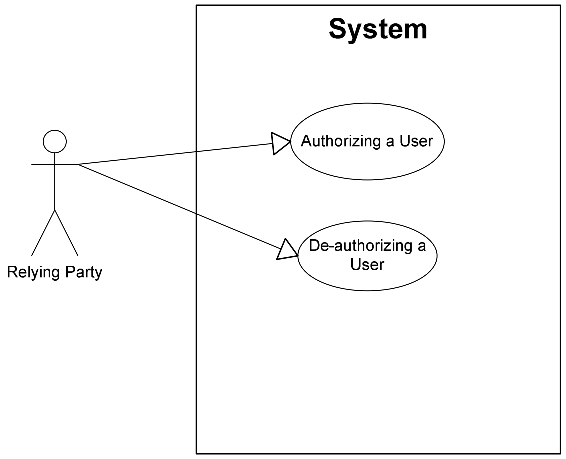

Figure 6: Relying Party Use Cases

The choice of our solution has no influence on the required use cases for the Relying Party as we show in figure 6.

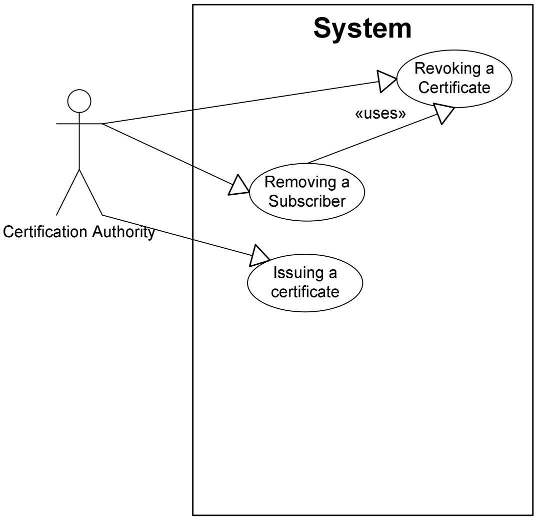

Figure 7: Certification Authority Use Cases

Since the role of the Certification Authority is specific to our solution the use case diagram 7 contains new use cases.

### 5.1.5 Issuing a certificate

Before a User can authenticate he must be issued a certificate for a Digital Identity. The Certificate Authority issues the certificate.

### 5.1.6 Revoking a certificate

The Certification Authority revokes a certificate to prevent misuse of the certificate.

### 5.1.7 Removing a Subscriber

The Certification Authority is notified by the GSM Network Operator that a SIM is lost or stolen. If this is the case the Certification Authority prevents this Subscriber from authenticating and revokes all certificates issued to this Subscriber.

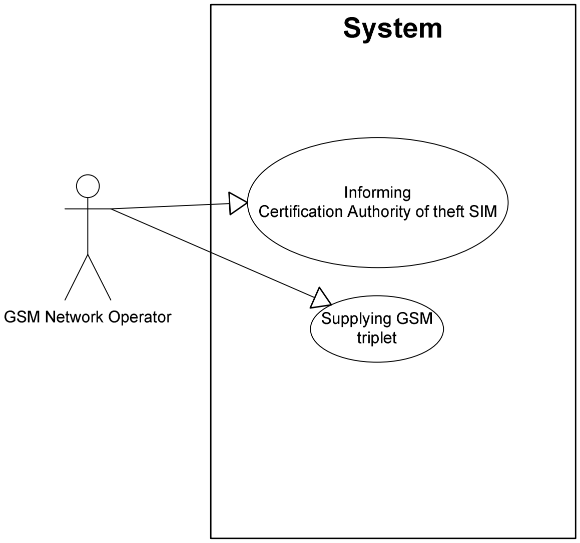

Figure 8: GSM Network Operator

The role of GSM Network Operator is also solution specific which means the use cases in 8 are also new.

### 5.1.8 Informing Certification Authority of theft SIM

When the GSM Network Operator is informed by the Subscriber it's SIM is stolen it notifies the Certification Authority of this fact.

### 5.1.9 Supplying GSM triplet

When asked by the Certification Authority the GSM Network Operator returns a GSM triplet.

## 5.2 CMS Token

The CMS Token Profile defines a CMS token. The CMS token is designed around the DSS Signature Object, which's schema can be found in appendix A. A proposal for the CMS token schema is included in appendix B.

## 5.3 Components

All components named here can also be found in the deployment diagram of the provisioning phase provided by figure 9, and the deployment diagram picturing the secure access of a Web Service which is given in figure 10.

### 5.3.1 Identity Selector

The Identity Selector is an important component of the identity metasystem. It is responsible for interpreting the token requirements the Service Requester has received and presenting the User a choice from the applicable Digital Identities. Then it obtains a security token for the Digital Identity and returns it to the Service Requester which in turn presents it to the Relying Party.

In the identity metasystem design the Identity Selector obtains its security tokens from a external Identity Provider. The CMS tokens required by our solution requires are generated on the Mobile Station, therefore our Identity Selector obtains its security token from an internal Identity Provider.

#### Java ME

The Identity Selector is shall be implemented as a Java class and accessible via a MIDlet. This assures the availability on a large number of operating systems and Mobile Equipment. The MIDlet will be activated using Push Registry so it can be started whenever a Requesting Party accesses a specific port.

#### Signature Service

In the identity metasystem the Identity Selector is responsible for selecting the Digital Identity which needs to be authenticated. In our solution the Identity Selector is also used for selecting the Digital Identity to be used in a Digital Signature for non-repudiation. Therefore it is also responsible for interpreting the signature requirements the Service Requester receives from the Relying Party. After obtaining the DSS Signature Object from the Identity Provider it is responsible for forwarding it to the Service Requester. These are new features for the Identity Selector that aren't specified in Identity Selector Interoperability Profile[30]. A new OBJECT tag and XHTML syntax will have to be defined. This allows communication between the Relying Party and Service Requester about the requirements if the Service Requester is a browser. The signature requirements should be specified just like the token requirements already defined in the identity metasystem. They should contain which Identity Providers are allowed and what kind of signature formats.

### 5.3.2 Identity Provider

The Identity Provider is responsible for providing a security token to the Identity Selector.

#### Java ME

The Identity Provider will be a Java class and will be accessible via same MIDlet as the Identity Selector. Using SATSA-PKI the class is able to obtain a CMS signature from the Security Element. When asked to supply an security token the Identity Provider uses the Java ME Web Services package to create a CMS token from the CMS signature. When it is asked to supply a Digital Signature for non-repudiation it uses the same package to wrap the CMS signature in a DSS Signature Object.

#### Certificates

The certificates used for the creation of CMS signatures are X.509 certificates[27]. Each certificate can be used for either authentication or non-repudiation. Certificates used for authentication produce CMS signatures used to create a CMS token. CMS signatures produced by certificates for non-repudiation are used to create a DSS Signature Object.

The validity period of the certificates is Certification Authority dependent but should not be too short nor too long. This would cause additional work for the User and additional strain on the Mobile Station, or a security risk. We suggest a validity period ranging from 6 months up to a year. This is no longer than the life expectancy of a subscription and thus prevents the risk of losing an unused SIM with credential. It is also long enough not to be bothersome for the User when he needs to reacquire a credential.

### 5.3.3 Certificate Manager

The Certificate Manager communicates with the Certification Authority. It obtains X.509 certificates and the information cards which represent these certificates in the Identity Selector. The certificates are stored on a Security Element. The Certificate Manager obtains the CSR and CMS signatures from the Security Element.

#### Authentication

The Certificate Manager authenticates the Subscriber to the Certification Authority using EAP-SIM. The Subscriber is identified by its IMSI.

The Certification Authority authenticates to the Certificate Manager using a X.509 certificate in TLS. The Certification Authority uses a short lived certificate. The time frame in which a short lived certificate can be abused when stolen is very short, therefore the need to check for revocation is mitigated. Checking for revocation would put a strain on the Mobile Equipment.

#### Subscribers

It may be allowed that one User is associated with multiple Subscribers by the Certification Authority. This allows a User to manage his Digital Identities from multiple subscriptions. This also allows him to migrate from one subscription to another subscription without losing his Digital Identities.

#### Identities

A User may have multiple Digital Identities for privacy reasons. For each of these Digital Identities a User may request a certificate by issuing a CSR. A Digital Identity is represented as the Subject field in a X.509 certificate. Multiple certificates with the same Digital Identity may be issued to the same User. This allows a User use the same Digital Identity on multiple Smart Cards. For each certificate the User is allowed revoke it or remove it from the Security Element, the Security Element's implementation may or may not remove the associated private key. The user is also allowed to obtain the information card belonging to the certificate.

#### Java ME

The Certificate Manager will also be implemented in Java ME and is accessible via a MIDlet. The Certificate Manager needs access to the SIM to be able to perform EAP-SIM authentication. The Certificate Manager also needs access to the Security Element to generate CSRs and store credentials. Both can be provided by using the SATSA-APDU and SATSA-PKI packages from JSR 177.

### 5.3.4 Relying Party

A Relying Party can be implemented in any language, as long as it is able to request and process the CMS token. If the Relying Party offers its service by web site it needs to be able to express its token and signature requirements in HTML and process the token from the HTML post. Otherwise the Relying Party needs to use the required WS-* and DSS protocols for this.

#### Authentication

The Relying Party authenticates to the Service Requester using a short lived server side X.509 certificate.This authentication is performed by the Transport Layer Security(TLS) protocol underneath HTTP. Using a short lived certificate removes the need of checking for revocation on the Mobile Station. Revocation checking puts a strain on the Mobile Station because it requires storage of a CRL or execution of the OCSP which requires a round trip to the OCSP-responder. This round trip to the OCSP-responder is a risk to the User's privacy because it allows linking by the OCSP responder. This linking can be based on the User's IP address, it could be prevented by using a HTTP-Proxy.

The Relying Party authenticates the Service Requester by sending him his token requirements and processing the resulting CMS token. In the token requirements the Relying Party specifies it expects a CMS token. A nonce is included in the token requirements. This nonce is randomly generated by the Relying Party and is to be used only once.

The resulting CMS token includes a CMS signature consisting of a Digital Signature, the signed content and the certificate associated with the private key by which the Digital Signature is made. The signed contend is the nonce and an identifier of the Relying Party. The Relying Party checks the certificate for revocation. The Relying Party shall use a CRL for this check, only when the Relying Party requires high assurance of the certificate's validity the Online Certificate Status Protocol(OCSP) shall be used. The usage of this protocol may violate the privacy of the User, because it enables the OCSP responder to correlate the User with the Relying Party. Again the use of a HTTP-proxy could prevent this.

If the certificate is valid and not revoked, the Relying Party checks the correctness of the Digital Signature. Finally the Relying Party verifies the content contains the correct nonce. The content must also contain an identifier of the Relying Party the CMS token is intended for. The nonce prevents a replay attack and the use of pre-generated Digital Signatures. The identifier prevents a man in the middle attack. In such an attack one Relying Party imposes as the User by forwarding CMS tokens from the User to another Relying Party.

### Non-repudiation

Non-repudiation of a User's transaction is achieved by receiving a Digital Signature from the User. The Relying Party sends its signature requirements to the Service Requester and processes the DSS Signature Object which is returned. In the signature requirements the Relying Party specifies it expects a CMS signature and the text to be signed.

The resulting DSS Signature Object includes a CMS signature consisting of a Digital Signature, the signed content and the certificate associated with the private key by which the Digital Signature is made. The Relying Party checks the certificate for revocation. Only when the Relying Party requires high assurance of the certificate's validity the Online Certificate Status Protocol(OCSP) shall be used for this.

If the certificate is valid and not revoked the Relying Party checks the correctness of the Digital Signature. The Relying Party verifies the content is the text to be signed. The Relying Party stores the CMS signature and the certificate chain to validate the certificate, as evidence of the transaction. The Relying Party should also store the timestamp at which the CMS signature is received. In addition it could require the time to be included in the text to be signed.

### 5.3.5 Service Requester

The Service Requester can be a custom application or the Service Requester could be a browser. It can be implemented in any language available on the Mobile Station.

#### Authentication

The Service Requester authenticates to the Relying Party by presenting a CMS token.

The Service Requester authenticates the Relying Party with the X.509 certificate exchanged by the TLS protocol underneath HTTP.

### 5.3.6 Certification Authority

The Certification Authority is an entity already existing in the network. The Certification Authority must be extended with a database in which certificates are linked to the User and Subscriber(s). It must be extended to be able to support EAP-SIM. This requires a relationship with a number of GSM Network Operators.

We estimate the impact on the Certification Authority minimal. There are already multiple authentication methods used by Certification Authorities and the already have to do administration of the issued certificates. A Certification Authority could start with a small group of GSM Network Operators to provide Certificates to their Subscribers and gradually extend their service to other GSM Network Operators.

#### Authentication

The Certification Authority authenticates to the Certificate Manager by using a short lived server side X.509 certificate exchanged by the TLS protocol carrying HTTP.

The Certificate Manager authenticates to the Certification Authority using EAP-SIM, the Certification Authority checks if the IMSI it receives is not blacklisted.

### 5.3.7 GSM Network Operator

The GSM Network Operator is an already existing entity in the network. The GSM Network Operator will deliver the new service of assisting in EAP-SIM authentication. It is necessary to setup relationship with all Certification Authorities it assists with EAP-SIM in order to notify them in case of SIM card theft.

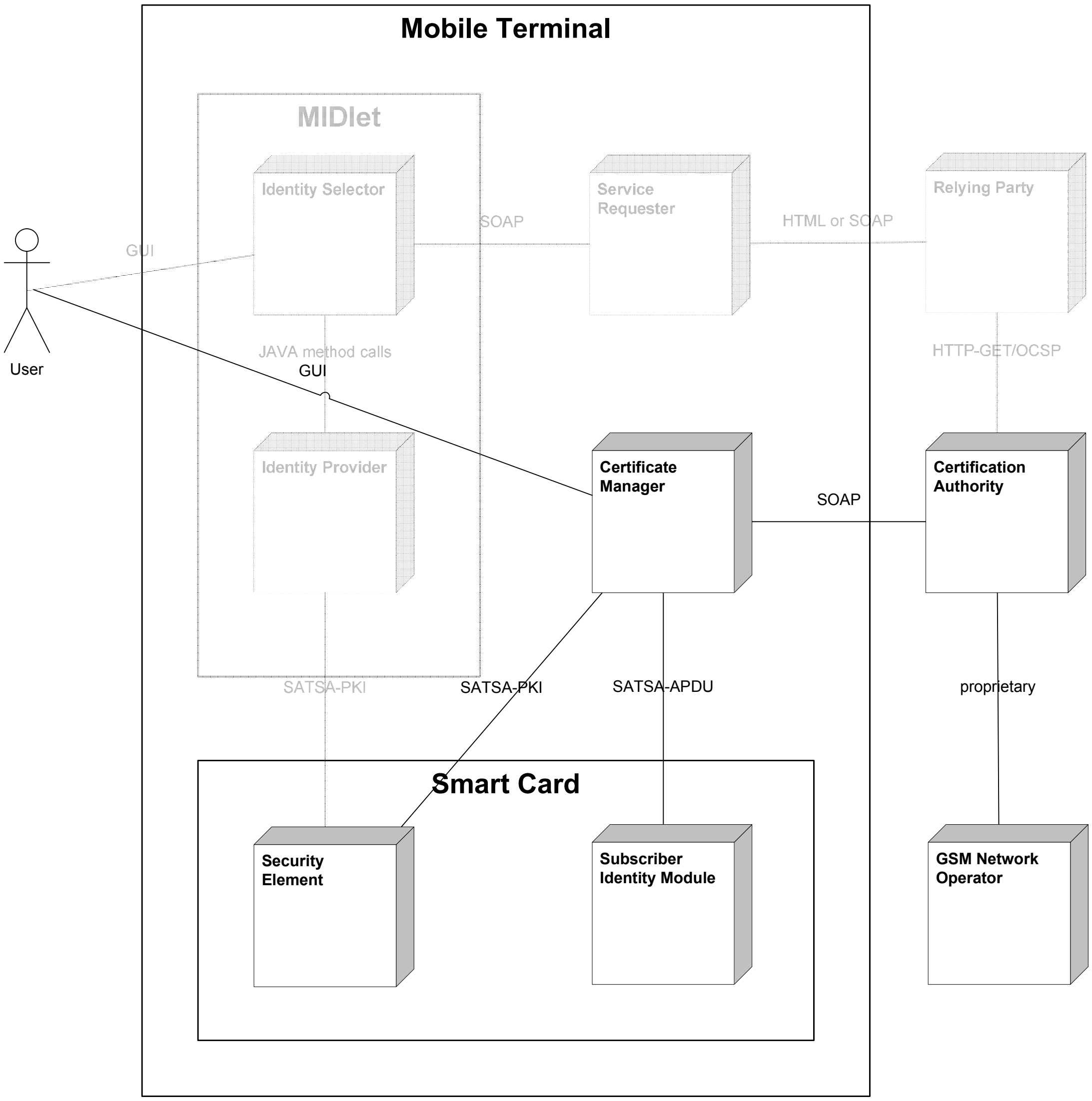

Figure 9: Deployment diagram for provisioning

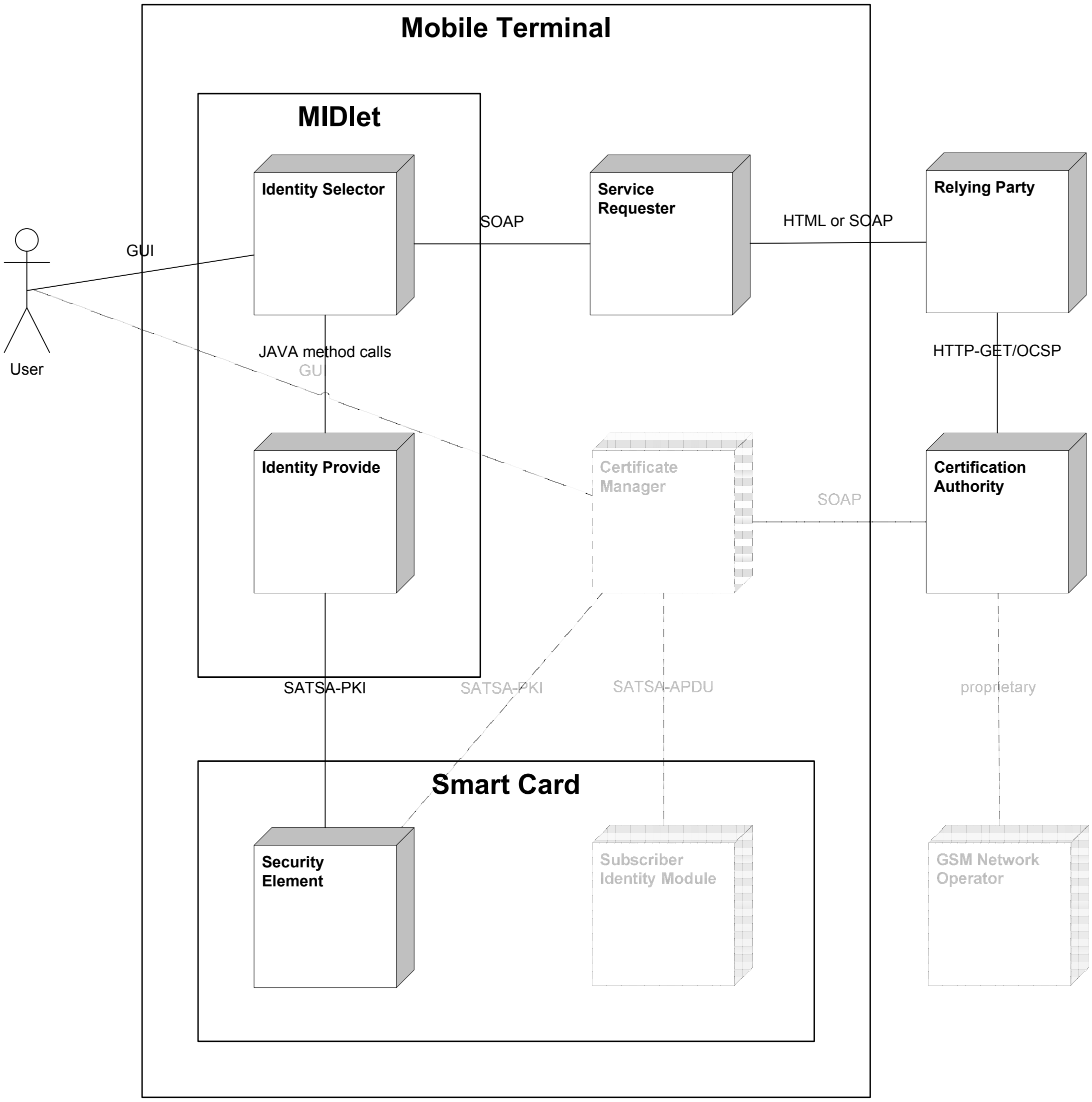

Figure 10: Deployment diagram for securely accessing a Web Service

## 5.4 Interfaces

The interfaces described here can be seen in figure 9 and 10. The usage of the interfaces can be seen in the sequence diagrams. Figure 11 shows a sequence diagram picturing the provisioning. A sequence diagram of securely accessing a Web Service is shown in 12.

### 5.4.1 Service Requester - Relying Party

#### Browser

In case the Service Requester is a browser the interface to exchange a CMS token between a Relying Party is described in the Identity Selector Interoperability Profile[30]. The use of the identity metasystem for DSS Signature Objects in a web browser is not standardized. Our intention is to contribute to standardization by using an OBJECT tag and XHTML syntax similar to the syntax for token exchange.

#### Non-browser

If the Service Requester is not implemented as a browser, SOAP is used as an interface for both token and signature exchange. The specific messages for token exchange are defined by WS-* protocols and the messages for signature services are standardized by the DSS protocols.

Both the exchange of SOAP messages and HTML will be done over HTTP using TLS with server side certificates. Using TLS provides server side authentication, and provides integrity and confidentiality of the communication between Service Requester and Relying Party.

### 5.4.2 Service Requester - Identity Selector

The Service Requester forwards any token and signature requests to the Identity Selector. This interface is also used to return the CMS token or DSS Signature Object to the Service Requester. The Identity Selector is activated by sending data to a TCP port, this is possible by using Push Registry. We propose a SOAP interface over HTTP (optionally with TLS) on this port. This way we are prepared for separate deployment of the Identity Selector and the Service Requester.

### 5.4.3 Identity Selector - Identity Provider

The Identity Selector forwards requests to the Identity Provider an receives the responses from the Identity Provider. In our setting it is not necessary to be able to deploy the Identity Selector and Identity Provider separately. Therefore in our design they will be implemented in the same MIDlet and the interface will be provided by class methods.

### 5.4.4 Identity Provider - Security Element

CMS signatures used for authentication and signing are generated by the Security Element. Therefore the Identity Provider needs a Security Element interface. This interface will be implemented using the SATSA-PKI API.

### 5.4.5 Certification Authority - Certificate Manager

The Certificate Manager needs to communicate with the Certification Authority, this communication will be realized by a SOAP interface running over HTTP using TLS. This provides server side authentication, integrity and confidentiality.

### 5.4.6 Certificate Manager - Security Element

In order for the Certificate Manager to generate CSRs and store certificates the Certificate Manager also needs a Security Element interface. This interface will be realized by using the SATSA-PKI package.

### 5.4.7 Certificate Manager - SIM

To be able to perform EAP-SIM authentication the Certificate Manager needs to communicate with the SIM. Therefore the Certificate Manager needs a Smart Card interface, the SATSA-APDU package will provide this interface.

### 5.4.8 Certification Authority - GSM Network Operator

The Certification Authority needs to obtain the triplets for EAP-SIM authentication from the GSM Network Operator. The interface may be any protocol both parties agree on.

Figure 11 shows a sequence diagram picturing the provisioning. A sequence diagram of securly accessing a Web Service is shown in 12.

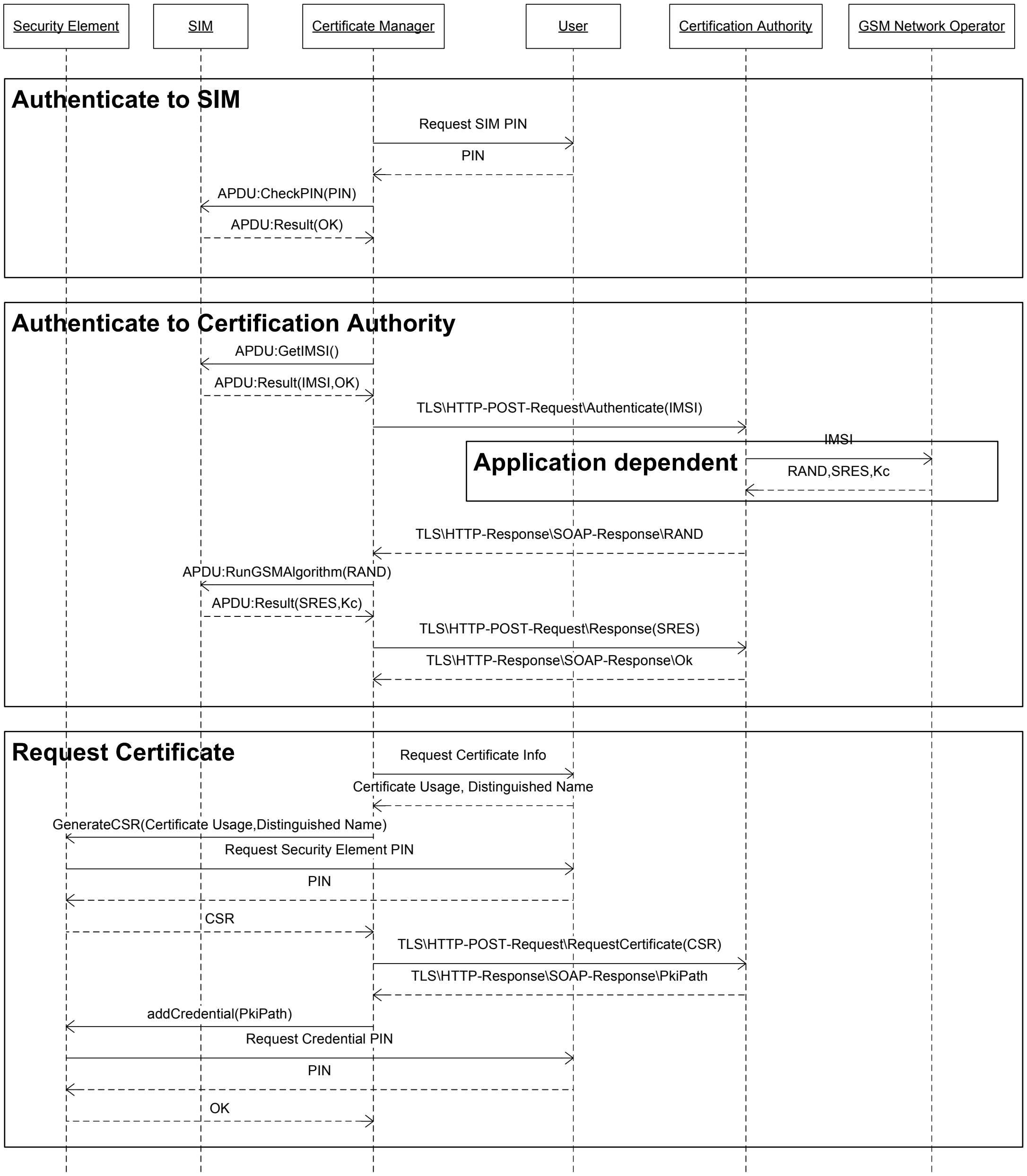
Security Element
SIM
Certificate Manager
User
Certification Authority
GSM Network Operator
Authenticate to SIM
Request SIM PIN
PIN
APDU:CheckPIN(PIN)
APDU:Result(OK)
Authenticate to Certification Authority
APDU:GetIMSI()
APDU:Result(IMSI,OK)
TLS\HTTP-POST-Request\Authenticate(IMSI)
IMSI
Application dependent
RAND,SRES,Kc
TLS\HTTP-Response\SOAP-Response\RAND
APDU:RunGSMAlgorithm(RAND)
APDU:Result(SRES,Kc)
TLS\HTTP-POST-Request\Response(SRES)
TLS\HTTP-Response\SOAP-Response\Ok
Request Certificate
Request Certificate Info
Certificate Usage, Distinguished Name
GenerateCSR(Certificate Usage,Distinguished Name)
Request Security Element PIN
PIN
CSR
TLS\HTTP-POST-Request\RequestCertificate(CSR)
TLS\HTTP-Response\SOAP-Response\PkiPath
addCredential(PkiPath)
Request Credential PIN
PIN
OK

Figure 11: Sequence diagram for provisioning

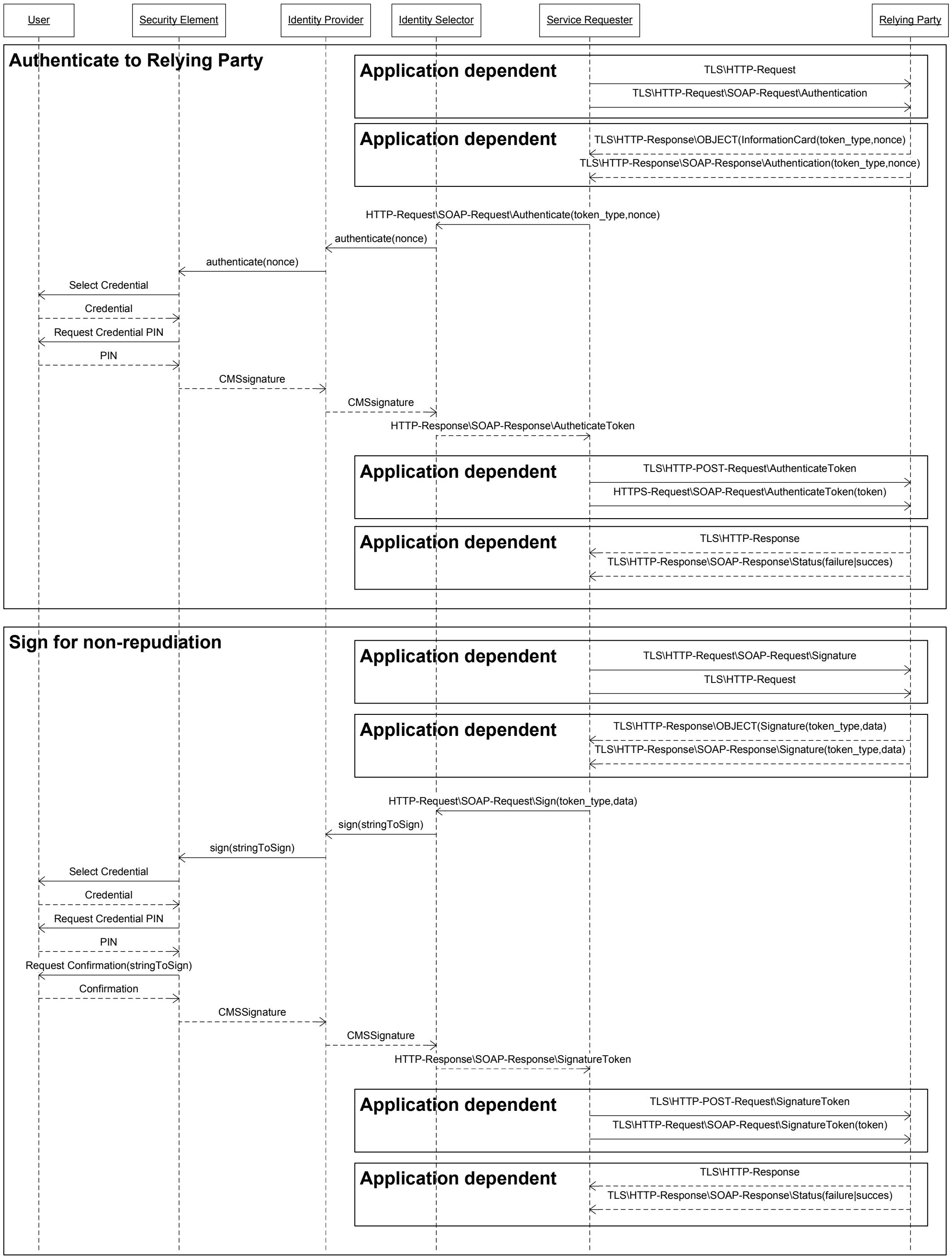
User
Security Element
Identity Provider
Identity Selector
Service Requester
Relying Party
Authenticate to Relying Party
Application dependent
TLS\HTTP-Request
TLS\HTTP-Request\SOAP-Request\Authentication
Application dependent
TLS\HTTP-Response\OBJECT(InformationCard(token_type,nonce)
TLS\HTTP-Response\SOAP-Response\Authentication(token_type,nonce)
HTTP-Request\SOAP-Request\Authenticate(token_type,nonce)
authenticate(nonce)
authenticate(nonce)
Select Credential
Credential
Request Credential PIN
PIN
CMSsignature
CMSsignature
HTTP-Response\SOAP-Response\AutheticateToken
Application dependent
TLS\HTTP-POST-Request\AuthenticateToken
HTTPS-Request\SOAP-Request\AuthenticateToken(token)
Application dependent
TLS\HTTP-Response
TLS\HTTP-Response\SOAP-Response\Status(failure|succes)
Sign for non-repudiation
Application dependent
TLS\HTTP-Request\SOAP-Request\Signature
TLS\HTTP-Request
Application dependent
TLS\HTTP-Response\OBJECT(Signature(token_type,data)
TLS\HTTP-Response\SOAP-Response\Signature(token_type,data)
HTTP-Request\SOAP-Request\Sign(token_type,data)
sign(stringToSign)
sign(stringToSign)
Select Credential
Credential
Request Credential PIN
PIN
Request Confirmation(stringToSign)
Confirmation
CMSSignature
CMSSignature
HTTP-Response\SOAP-Response\SignatureToken
Application dependent
TLS\HTTP-POST-Request\SignatureToken
TLS\HTTP-Request\SOAP-Request\SignatureToken(token)
Application dependent
TLS\HTTP-Response
TLS\HTTP-Response\SOAP-Response\Status(failure|succes)

Figure 12: Sequence diagram for securely accessing a Web Service

## 5.5 Evaluation

To evaluate our design we match it against our requirements. For each requirement we will explain if and how our architecture and design fulfills it.

### 5.5.1 Functional Requirements

The functional requirements describe what the solution should be able to do.

#### Internet Protocol

All external interfaces are designed on SOAP, SOAP runs on HTTP and therefore our solution is fully usable over any network using IP, including those which require NAT and HTTP proxies.

#### Compatibility

The solution is applicable to Mobile Equipment from any manufacturer as long as it supports Java ME with the Connected Limited Device Configuration and MIDP 2.0 and the required JSR's. The required JSRs are bundled in the Mobile Service Architecture JSR 248. Even a Mobile Equipment manufacturer which doesn't comply to this could design his own Identity Selector, Identity Provider and Certificate Manager. As long as the external interfaces, tokens and signature format are adopted the solution would still be compatible.

Only the use of the Certificate Manager is dependent on the GSM Network Operator, therefore roaming is of no influence when using a Web Service. Also during use of the Certificate Manager roaming is of no influence since the triplets used for authentication to the Certificate Authority are always provided by the Subscriber's home network MSC.

#### Authentication

The authentication of the Relying Party is provided by server side certificates in TLS. Using short lived certificates limits the use of a compromised certificate.

Authentication of the User is achieved by the CMS token. The generation of the token is based on a hardware token, the Smart Card which requires the PIN, therefore achieving level 4 authentication. X.509 V3 certificates can be used to provide extensions. Such an extension could be defined by the Certification Authority to provide the IMSI in the certificate. This would then allow Subscriber authentication. The validity of the User certificate should be checked by the Relying Party using a CRL, this prevents the use of a stolen or lost certificate.

### Integrity and Confidentiality

Data Integrity and confidentiality is provided by using TLS for all external interfaces.

### Non-repudiation

Non-repudiation is achieved by requiring a DSS Signature Object from the Service Requester. This Digital Signature is made by the Security Element and requires a PIN by the User. The Relying Party itself can verify the Digital Signature and validity of the User certificate. The DSS Signature Object can be stored and used as proof which any third party can verify.

The validity of this Digital Signature in a legal case may differ from country to county. This thesis does not make any assumptions about this.

### Privacy

The User certificate, containing the Digital Identity, is never sent over the network in plain text, instead CMS tokens are sent over a HTTP over TLS connection. The usage of a CRL in favor of a OCSP prevents the Certification Authority from learning which User uses which Relying Party. This maximizes the privacy for the User.

A User may have as many certificates as he likes. Each of these certificates may contain a different Digital Identity. Only the Certification Authority can link these certificates to the same User. By using a certificate per Digital Identity the User can prevent the linking of transactions with different Relying Parties. A User could even prevent linking of multiple transactions with the same Relying Party by using certificates with different Digital Identities.

The User certificate is not directly linked to a natural person. It is only linked to a Subscriber, and only the Certification Authority can link these two together. If the subscription is not prepaid the GSM Network Operator is able to link the subscription to a natural person. The Relying Party thus will be unable to resolve the Digital Identity to a natural person, which is beneficial to the privacy of the User.

By design our architectures does not leak additional information. By using IP it is inevitable to leak certain information such as the User's and Relying Party's IP address and other IP traffic details. Other means such as a HTTP-proxy or Tor[16] network can be used to prevent this.

### Authorization

The Relying Party can identify a User by the Digital Identity in his certificate. This means he can authorize and de-authorize a Digital Identity to access a Web Service. A Certification Authority will never issue a certificate with the same Digital Identity to a different applicant. Therefore the Relying Party will be able to differentiate between different Digital Identities. If a Certification Authority issues certificates which contain statements about the Digital

Identity such as age or address, this would allow the Relying Party to authorize the User based on it's attributes rather then its Digital Identity itself.

#### Flexibility

The Identity Selector and Identity Provider are stateless and do not keep track of Service Requesters. Therefore multiple Service Requesters may run simultaneously acting on behalf of different Users, adding flexibility.

The User may be associated with multiple Subscriptions by the Certification Authority. This way a User may request a certificate with the same Digital Identity on a different Mobile Station giving him more flexibility.

### 5.5.2 Quality Requirements

The quality requirements describe what properties the solution must satisfy.

#### Number of passwords

A credential PIN is associated with each credential stored in the Security Element. The User itself may decide how many credentials he wishes to use and what PIN to use. A SIM PIN is also required to access the Certificate Manager. The number of PIN codes is therefore fairly limited.

The PIN can be the same for all credentials and the SIM. It is however suggested to use a different PIN for the SIM and to have a disjunct PIN for authentication and non-repudiation credentials.

#### Password input

Each time a User authenticates to a Web Service provided by a Relying Party a PIN is required. During the established session the PIN is not needed again. Only when a Relying Party requires a Digital Signature as a confirmation of a transaction a PIN will be required again.

While using the Certificate Manager a PIN is required for authentication to the Certification Authority and once for each certificate a User wants to store or remove.

#### User interaction

When accessing a Web Service the Identity Selector needs to know which certificate to use and the credential PIN code associated with this certificate. This is a strict limitation on the User interaction.

The Certificate Manager requires more interaction when requesting, storing and deleting certificates. Since the use of the Certificate Manager is bound to be infrequent this is no real limitation.

### Thrusted third party

The Certification Authority and GSM Network Operator are already known as trusted parties. Therefore we have not introduced any new trusted third parties. Although we extend the trust we already have in the GSM Network Operator.

### Hardware

Existing Mobile Equipment already has a Smart Card reader. No further hardware is required in the Mobile Equipment.

### Software

The Identity Provider and Identity Selector software are not yet present on the Mobile Equipment. They are designed as a java MIDlet which can be installed at any point in time or provided via a firmware update. If the web browser is to be used as a Service Requester the web browser will require an update. This update is needed so the web browser understands the HTML object and XHTML tags and is able to contact the Identity Selector.

The Security Element needed on the Smart Card can be deployed later through the use of the an Over The Air(OTA) SIM Application Tookit(SAT). For GSM Network Operators not supporting this the Smart Card must be returned and exchanged for a new one.

### Availability

The validation and generation of a CMS token is independent of the Certification Authority and the GSM Network Operator. Therefore the availability is only dependent on the Mobile Station and the Relying Party, this is no additional constraint. Only when the Relying Party wishes to use an OCSP the availability is influenced by the availability of the Certification Authority.

During the use of the Certificate Manager both the GSM Network Operator and the Certification Authority are required in addition to the Mobile Station. This use is very infrequent so it will have a low impact on overall availability.

# 6 Implementation

This section is dedicated to the development of the proof of concept. The proof of concept only verifies the facets of our solution which are new and unproven.

The proof of concept has not been verified on a real Mobile Station or real Smart Card. The reason for this is the fact that we were unable to obtain a Smart Card with a SIM and a Security Element, and with the right permissions. Instead we used a PC, with a fully emulated environment.

Figure 13 and 14 depict the deployment diagrams of the proof of concept.

## 6.1 Service Requester/Identity Selector/Identity Provider

The Service Requester, Identity Selector and Identity Provider are combined in a single entity for the proof of concept. This entity has been implemented in a MIDlet. The MIDlet is run in the Java ME Virtual Machine provided by the SUN Java Wireless Toolkit[15].

The choice for implementing three entities in one reduces the amount of interfaces needed to be implemented. These interfaces were designed with Push Registry and SOAP. Both technologies have been used on the Mobile Station before and sample code is available in the SUN Java Wireless Toolkit, therefore these interfaces aren't a part of our proof of concept. Also a Mobile HTTP server had to be implemented for one of these interfaces. Work in this area has been done by Nokia[14], and it is not the focus of our research so this also isn't part of our proof of concept.

As a further reduction in work the interface between this entity and the Relying Party is not using SOAP messages but is directly implemented over HTTP. Unfortunately we haven't been able to test this with HTTP over TLS because the emulated environment has problems with self signed server side certificates.

The entity implemented for the proof of concept is able to obtain the token requirements including nonce from the Service Requester over HTTP. After obtaining the requirements it is able to present the available certificates to the User, request the PIN for the certificate and generate a CMS signature. The entity then sends the CMS signature to the Relying Party, it however does not build a CMS token around the CMS signature since this is trivial.

The entity is able to obtain the text to be signed over HTTP from a Relying Party. The entity can present the User with the appropriate certificates and request the PIN for the selected certificate. The resulting CMS signature isn't packaged in the DSS Signature Object because again this is trivial. The obtained CMS signature is send back to the Relying Party over HTTP.

## 6.2 Relying Party

The Relying Party is implemented in a Servlet. This Servlet runs on an Apache Tomcat web server. The web server runs on the same host as the SUN Java Wireless Toolkit.

The Relying Party is able to send the token requirements, including a random generated nonce, to the Service Requester. The CMS signature the Service Requester returns can be validated, the certificate can be extracted and content of the CMS signature can be compared to the nonce. The Relying Party uses a session to keep track of the issued nonce and also stores authentication information in the same session. After the User is authenticated the Digital Identity from the X.509 certificate is also stored in the session.

Sending the text to be signed is also implemented in the Relying Party. When generating this text to be signed it includes the time and a random nonce to prevent replay attacks. It is able to verify the CMS signature the Service Requester returns. As a proof the Relying Party stores all CMS signatures. Therefore even if the User returns later with a different certificate with the same Digital Identity it is able to retrieve the transactions the User has agreed upon.

## 6.3 Certificate Manager

The Certificate Manager is implemented as a MIDlet an runs in the SUN Java Wireless Toolkit.

All communication is done directly on HTTP instead of implementing the SOAP layer. Instead of using EAP-SIM it uses the GSM authentication. EAP-SIM over HTTP has already been proven in 'A Unified Authentication Solution for Mobile Services'[25]. Using GSM authentication proves we can access the SIM and is more easily understood then EAP-SIM.

The Certificate Manager is able to request the $IMSI$ from the SIM. The Certificate Manager sends the $IMSI$ to the Certification Authority and receives a $RAND$. The Certificate Manager relays the $RAND$ to the SIM and receives the $SRES$ and $K_C$ from the SIM. The Certificate Manager is then able to send the $SRES$ contained in the triplet to the Certification Authority and therefore completing a GSM authentication.

Once authenticated the Certificate Manager can obtain a CSR from the Security Element. The User can specify the Subject for the certificate and the certificate usage. The Certificate Manager can send the CSR to the Certification Authority and receive the corresponding PkiPath, an ASN.1 DER encoded sequence of certificates. The certificate will be stored on the Security Element requiring the credentials PIN.

## 6.4 Certification Authority

The Certification Authority is implemented as a Servlet. This Servlet runs on an Apache Tomcat web server. The web server runs on the same host as the SUN Java Wireless Toolkit.

The Certification Authority is able to authenticate the Service Requester. After receiving the $IMSI$ it forwards the $IMSI$ to the GSM Network Operator and receives a triplet. This communication with the Certification Authority is implemented directly on top of HTTP.

After receiving the triplet the Certification Authority forwards the $RAND$ from the triplet to the Certificate Manager. When the Certificate Manager returns its $SRES$ the Certification Authority compares it with the $SRES$ from the triplet. All communication with the Certificate Manager is done directly on top of HTTP again leaving the SOAP layer out. The Certification Authority uses sessions to keep track of the Certificate Managers by their $IMSI$.

When the Certificate Manager sends a CSR the Certification Authority checks if the Certificate Manager is authenticated. If the Subject in the certificate is not registered to another User the Certification Authority generates a certificate using OpenSSL and returns the PkiPath to the Certificate Manager. The Certification Authority keeps track of all certificates registered to an User. The proof of concept Certification Authority does not implement certificate revocation, re-downloading of certificates or registering multiple Subscribers to a single User. This is due to a lack of time and relevance for our proof of concept.

## 6.5 GSM Network Operator

The GSM Network Operator is implemented as a Servlet. This Servlet runs on a Apache Tomcat web server. The web server runs on the same host as the SUN Java Wireless Toolkit.

The GSM Network Operator has only one function which is returning a triplet on receipt of an $IMSI$. The GSM Network Operator has a list of each $IMSI$ with a $K_i$. By looking up the $K_i$ from the $IMSI$ it can generate a triplet consisting of $RAND$, $SRES$ and $K_c$. The GSM Network Operator returns this triplet to the Certification Authority. The messages are implemented directly on top of HTTP.

## 6.6 SIM

The SIM is implemented as an Applet implemented in the Javacard Framework. The Javacard is simulated in the Java Card 2.2.1 C Reference Implementation Simulator[9].

Unable to obtain a Smart Card with a SIM and Security Element with the right access rights, we had to implement our own SIM. This SIM is loaded into a virtual Smart Card which already contains a Security Element.

The SIM implements methods to validate the PIN, retrieve the $IMSI$ and run a combined $A3$ and $A8$ algorithms.

## 6.7 Security Element

The Security Element is implemented as an Applet in the Javacard Framework.

The Security Element used is part of the SUN Java Wireless Toolkit. This particular Security Element is limited to keys of 512 bits. This is too short for current security standards but of no influence to our proof of concept. Real life Security Elements offer longer keys.

## 6.8 Evaluation

The implementation did not reveal any problems with our architecture or design. However other problems have arisen and we will describe the reason for and possible solutions to these problems in this evaluation.

The biggest problem with our proof of concept has been the lack of a suitable Smart Card. The only way to obtain such a card is from a GSM Network Operator willing to cooperate in such a project. In our case the time to achieve such cooperation was too short. As a result we had to implement our own SIM and run the proof of concept in an emulated environment.

A second problem has been the lack of a GSM Network Operator code signing certificate. Without this certificate we were unable to sign our code to the GSM Network Operator domain. This resulted in additional warnings in the proof of concept which makes it hard to illustrate the user-friendliness of the concept. Again this has been due to the lack of cooperation with a GSM Network Operator.

Further issues are related to the emulated environment. Due to the lack of support for self signed certificates we could not test our proof of concept with a Relying Party or Certification Authority with HTTP over TLS. We have tested HTTP over TLS with certificates from known Certificate Authorities. These tests were successful, including the tests for validity checking.

The capability to remove a certificate from the Security Element has not been tested in any environment. We have tried this in the emulated environment but have not succeeded, this is a point for further investigation.

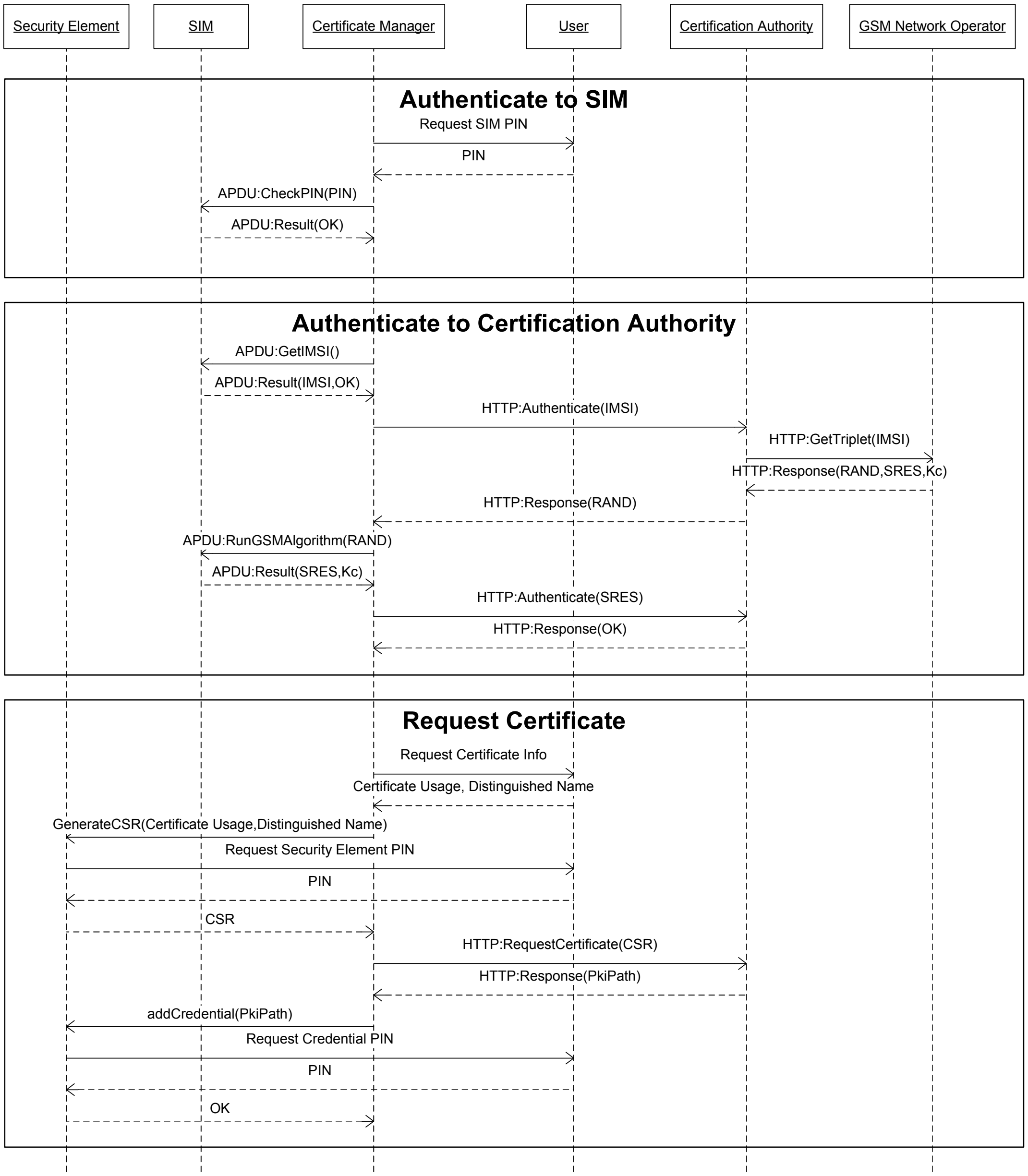

Figure 13: Sequence Diagram for the proof of concept accessing a Certification Authority

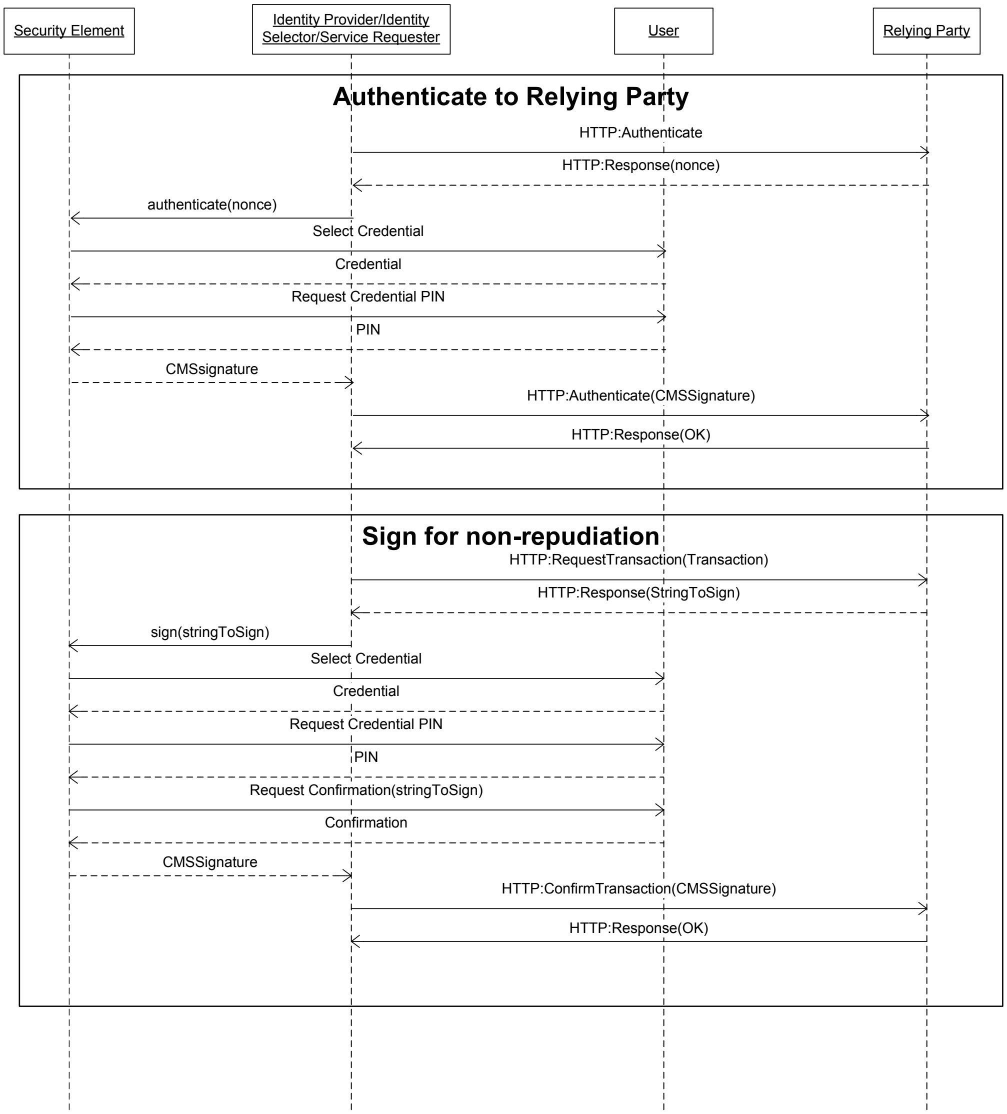

Figure 14: Sequence Diagram for the proof of concept accessing a Web Service

# 7 Conclusions

The focus of this thesis was to investigate the problem of securely accessing a Web Service from a Mobile Station. This means we have performed a literature study the topics of authentication, non-repudiation, Digital Identities, Digital Signatures and identity management. Effort has gone into investigation of existing protocols and systems. A large part of the master's project has been spent on architecture and design and a smaller portion has gone into the implementation of a proof of concept.

## 7.1 Achievements and results

The proposed architecture combines the already defined identity metasystem with EAP-SIM and existing PKI to achieve two-factor mutual authentication. This authentication fulfills the requirements of the highest security level defined by NIST. The architecture also fulfills the requirement of non-repudiation of a User's actions. Furthermore we provide privacy for the User.

The architecture is fully compliant with the defined requirements. It is based on existing standards, systems and protocols. It uses existing hardware and trusted parties. Currently deployed username/password solutions are less user friendly and our solution requires no additional hardware device like currently deployed strong authentication methods do.

The proof of concept is an investigation into the technical feasibility of the solution. This shows the solution is feasible and has great potential. The major problem with the proof of concept is the lack of deployment in a real environment. This was impossible because we did not have a Smart Card with a SIM, Security Element and the right permissions. Another problem was the lack of a code signing certificate which caused the proof of concept to throw warnings which it otherwise would not throw. We have however used SATSA-PKI to prove we can produce CMS signatures. We have also showed these CMS signatures can be used by a Relying Party for authentication and non-repudiation. SATSA-APDU has been used to illustrate that SIM authentication is possible and therefore EAP-SIM can be used to authenticate to a Certification Authority.

The security architecture we present is a innovative solution which allows secure use of a Web Service from a Mobile Station. Current authentication systems are bothersome and inconvenient to use and they rarely offer strong authentication. Our solution offers strong authentication and allows non-repudiation. In addition it is user-friendly and is designed with privacy in mind. The solution doesn't require additional investments in hardware and thus is inexpensive. The solution only requires investments of the GSM Network Operators and the Certificate Authorities. They are required to cooperate and extend their services to support EAP-SIM. With only his PIN a User can access his banks, online email, use corporate services and communicate with the government.

## 7.2 Critical review

The lack of a suitable Smart Card and code singing certificate has kept us from demonstrating the proof of concept on a Mobile Station. Our focus has been on the critical parts of the architecture. Therefore we have left existing parts of the architecture out of the prototype. The critical parts have been the SATSA-APDU an SATSA-PKI interfaces on the Mobile Station and the HTTP interface to both Relying Party and Certificate Manager.

Most of the research has gone into existing solutions, protocols and standards. The combination of existing technologies is critical to our solution because this makes it widely acceptable and ready to deploy.

The security of the proposed architecture depends on EAP-SIM and therefore on SIM authentication. This means that if vulnerabilities are found in the SIM or SIM authentication this will have immediate impact on our solution. In case of critical problems with SIM authentication GSM Network Operators will need to move to a more secure platform. Newer generations of mobile networks such as UMTS and CDMA have already implemented stronger authentication, using AKA algorithms. Authentication over EAP is also available using AKA algorithms with EAP-AKA[18]. We can easily add EAP-AKA authentication to our solution and therefore evolve with the mobile networks.

The solution relies on the SATSA-APDU and SATSA-PKI packages to communicate with the SIM and the Security Element. These packages are part of the Java ME Virtual Machine on the Mobile Equipment. If the Java ME Virtual Machine is compromised it may be possible to intercept the SIM or credential PIN. The same holds if the operating system beneath the Java ME Virtual Machine is compromised. Such a compromise is already a problem for it could lead to the use of Mobile Services without the Subscribers consent. So its in the GSM Network Operators and therefore the Mobile Equipments manufacturers' interest to keep the Mobile Equipment secure.

## 7.3 Future work

Future work must start with validation of the proof of concept on a real Mobile Station with a real Smart Card. If this is successful the CMS Token Profile and HTML structures used for the transport of DSS Signature Objects have to be standardized. With these definitions the extended definitions of a Identity Selector and Identity Provider is the next step. The Identity Selector should support all currently available tokens used within the identity metasystem and the CMS token. Both the Identity Selector and Identity Provider roles should be extended to support signature services. To develop the Certificate Manager we have to define how EAP messages are encapsulated in SOAP, some work on this has already been done in [28]. With this definition the Certificate Manager can be defined and the Certification Authority can be extended to use EAP-SIM.

The next step in the evolution of this architecture is making it available to other devices. This could be done in several ways. The first could be extending the architecture to any device with a Smart Card reader. With the right interfaces available the architecture can be implemented on any device with a Smart Card reader. This would allow secure access to Web Services authentication from laptops, desktops and other devices. Another angle to achieve availability of the architecture on other devices is separating the Identity Provider and Identity Selector role. This would make it possible to use the Mobile Station as Identity Provider for an Identity Selector deployed on another device. In this situation no Smart Card reader is required in the device.

The investigation of EAP-AKA as a replacement of EAP-SIM in this architecture is a logical follow-up study. As for the compromise of Mobile Equipment, researching how the Mobile Equipment can qualify as a Trusted Computing Base would be a good follow-up.

Looking at the Relying Party side of things we have shown that strong authentication and non-repudiation are achievable goals when providing Web Services. From the User's perspective we have brought a user-friendly architecture with great control of privacy. The GSM Network Operator now has a way to make access to the internet more attractive from the Mobile Station. Summarized we have taken an important next step in securely accessing a Web Service from a Mobile Station.

## References

[1] GTS 01.04: European digital cellular telecommunications system (phase 1); abbreviations and acronyms (gsm 01.04). `http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=4604`, 1995.

[2] ETS 300 534: Digital cellular telecommunications system (phase 2) (gsm);security related network functions (gsm 03.20 version 4.4.1). `http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=5403`, August 1997.

[3] Wmlscript specification. `http://www.openmobilealliance.org/tech/affiliates/wap/wap-193-wmlscript-20001025-a.pdf`, 2000.

[4] Wireless application protocol public key infrastructure definition. `http://www.openmobilealliance.org/tech/affiliates/wap/wap-217-wpki-20010424-a.pdf`, 2001.

[5] Wireless identity module. `http://www.openmobilealliance.org/tech/affiliates/wap/wap-260-wim-20010712-a.pdf`, 2001.

[6] Wireless markup language. `http://www.openmobilealliance.org/tech/affiliates/wap/wap-238-wml-20010911-a.pdf`, 2001.

[7] Wireless transport layer security. `http://www.openmobilealliance.org/tech/affiliates/wap/wap-261-wtls-20010406-a.pdf`, 2001.

[8] Wmlscript crypto library. `http://www.openmobilealliance.org/tech/affiliates/wap/wap-161-wmlsscriptcrypto-20010620-a.pdf`, 2001.

[9] Java card development kit. `http://java.sun.com/products/javacard/dev_kit.html`, 2003.

[10] Microsoft's vision for an identity metasystem. `http://msdn2.microsoft.com/ms996422.aspx`, 2005.

[11] Sim/usim internal and external interworking aspects. `http://www.3gpp.org/ftp/Specs/archive/31_series/31.900/31900-710.zip`, 2006.

[12] Fineid. `http://www.fineid.fi`, 2007.

[13] JSR 177 security and trust services api (satsa). `http://jcp.org/en/jsr/detail?id=177`, 2007.

[14] Mobile web server. `http://research.nokia.com/research/projects/mobile-web-server/`, 2007.

[15] Sun java wireless toolkit for cldc. `http://java.sun.com/products/sjwtoolkit/`, 2008.

[16] Tor. `http://www.torproject.org/index.html.en`, 2008.

[17] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and E. H. Levkowetz. RFC 3748: Extensible authentication protocol (eap). `http://tools.ietf.org/html/rfc3748`, 2004.

[18] J. Arkko and H. Haverinen. RFC 4187: Extensible authentication protocol method for 3rd generation authentication and key agreement (eap-aka). http://www.ietf.org/rfc/rfc4187.txt, 2006.

[19] W. E. Burr, D. F. Dodson, and W. T. Polk. Electronic authentication guideline. http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf, 2006.

[20] S. Drees. Digital signature service core protocols, elements, and bindings version 1.0. http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.pdf, 2007.

[21] D. Eastlake, J. Reagle, and D. Solo. RFC 3275: (extensible markup language) xml-signature syntax and processing. http://tools.ietf.org/html/rfc3275, 2002.

[22] P. Hallam-Baker, C. Kaler, R. Monzillo, and A. Nadalin. Web services security saml token profile. http://www.oasis-open.org/committees/download.php/1048/WSS-SAML-06.pdf, 2003.

[23] P. Hallam-Baker, C. Kaler, R. Monzillo, and A. Nadalin. Web services security x.509 certificate token profile. http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0.pdf, 2004.

[24] H. Haverinen and J. Salowey. RFC 4186: Extensible authentication protocol method for global system for mobile communications (gsm) subscriber identity modules (eap-sim). http://tools.ietf.org/html/rfc4186, 2006.

[25] H. Holje, I. Jrstad, and T. van Do. A unified authentication solution for mobile services. In *1st ERCIM Workshop on eMobility*, pages 131–142, 2007. ISBN: 978-972-95988-9-0.

[26] R. Housley. RFC 3852: Cryptographic message syntax. http://tools.ietf.org/html/rfc3852, 2004.

[27] R. Housley, W. Polk, W. Ford, and D. Solo. RFC 3280: Internet x.509 public key infrastructure certificate management protocol (cmp). http://tools.ietf.org/html/rfc3280, 2005.

[28] R. Huber and N. Jordan. An experimental study of a business domain independent application level and internet access authentication and authorization concept. *Mobile Business, 2005. ICMB 2005. International Conference on*, pages 35–41, July 2005. ISBN: 0-7695-2367-6.

[29] B. Kaliski. RFC 2315: Pkcs #7: Cryptographic message syntax version 1.5. http://tools.ietf.org/html/rfc2315, 1999.

[30] A. Nanda. Identity selector interoperability profile. http://download.microsoft.com/download/1/1/a/11ac6505-e4c0-4e05-987c-6f1d31855cd2/identity-selector-interop-profile-v1.pdf, 2007.

[31] A. Pashalidis and C. Mitchell. Using gsm/umts for single sign-on. *Mobile Future and Symposium on Trends in Communications, 2003. SympoTIC '03. Joint First Workshop on*, pages 138–145, October 2003. ISBN: 0-7803-7993-4.

[32] M. Schuba, V. Gerstenberger, and P. Lahaije. Internet id - flexible re-use of mobile phone authentication security for service access. In *IEEE Communications Magazine*, volume 42, September 2004.

[33] A. Wangensteen, L. Lunde, I. Jrstad, and D. van Thanh. A generic authentication system based on sim. *icisp*, 0:24, 2006. ISBN: 0-7695-2649-7.

# Appendices

## A SignatureObject

The following schema fragment defines the <SignatureObject> and <Base64Signature> elements and is taken from [20].

```
<xs:element name="SignatureObject">
  <xs:complexType>
    <xs:sequence>
      <xs:choice>
        <xs:element ref="ds:Signature"/>
        <xs:element ref="dss:Timestamp"/>
        <xs:element ref="dss:Base64Signature"/>
        <xs:element ref="dss:SignaturePtr"/>
        <xs:elementname="Other"type="dss:AnyType"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="SchemaRefs" type="xs:IDREFS" use="optional"/>
  </xs:complexType>
</xs:element>

<xs:element name="Base64Signature">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:base64Binary">
        <xs:attribute name="Type" type="xs:anyURI"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

Our solution is based on CSM signatures which are Base64Signatures. The <Base64Signature> contains a base64 encoding of some non-XML signature, such as a PGP or CMS [26] signature. The type of signature is specified by its Type attribute.

An example of a <SignatureObject> to be used in our framework is given below.

```
<dss:SignatureObject>
    <dss:Base64Signature Type="urn:ietf:rfc:3852">
        a345f...0ea34
    </dss:Base64Signature>
</dss:SignatureObject>
```

## B CMSToken

The following schema is a proposal for the <CMSToken> in the CMS Token Profile.

```
<xs:element name="CMSToken">
    <xs:element ref="dss:SignatureObject"/>
</xs:element>
```

An example of a <CMSToken> to be used in our framework is given below.

```
<xyz:CMSToken>
    <dss:SignatureObject>
        <dss:Base64Signature Type="urn:ietf:rfc:3852">
            a345f...0ea34
        </dss:Base64Signature>
    </dss:SignatureObject>
</xyz:CMSToken>
```