Mobile Certificate Based Network Services

Göran Pulkkis and Farzan Yazdani
Arcada University of Applied Sciences, Helsinki, Finland

goran.pulkkis@arcada.fi farzan.yazdani@gmail.com

Abstract

The deployment of mobile certificates in Finland, the related Application Provider's Interface (API), and also the design, management, and use of a Mobile Signature Service (MSS) is described. The role of a mobile certificate in a Public Key Infrastructure (PKI) is outlined and some related research is presented. ETSI (European Telecommunications Standards Institute) standards for MSS implementations and FiCom (The Finnish Federation for Communications and Teleinformatics) application guidelines for these standards are presented. MSS implementation details are shown for user authentication and for electronic signing of plain text. A step-by-step user experience of an implemented MSS combining user authentication with electronic signing of plain text is presented in an Appendix A. A list of abbreviations and definitions is in Appendix B.

Keywords: mobile certificate, PKI, digital signature, authentication, SOAP, information security

Introduction

Electronic services are growing rapidly and the lack of security, compatibility and effectiveness is one of the reasons why new methods like mobile certificate will play a big role in the evolution. Vetuma is a public administration's joint service for citizen authentication and payment in Finland. Vetuma has introduced for citizens an additional method for authentication and payment with mobile certificates (FI: Vetuma, 2011).

The mobile certificate is based on the ETSI's MSS standards and has been developed in Finland in collaboration with mobile cellular network operators (Mobiilivarmenne, 2010). The MSS is used to give permission to a transaction (e.g. financial), which the user has initiated with his/her mobile device. Therefore the MSS is defined as a service in which the mobile signature process is coordinated or managed for the user and for the Application Provider (AP). The Mobile Signature Service Provider (MSSP) offers MSS systems for service providers (ETSI, 2003a; ETSI, 2003b).

In Web Service Interfaces the MSS is used for authentication or for signing text content or a digest of text content. Therefore MSS has a Web Service Interface, which MSSP provides and/or implements for an AP. In implementation of network services for mobile certificates, the concept of MSS must be thoroughly understood. (ETSI, 2003b)

The objective of this paper is to show how mobile certificate based solutions for IT services can be designed with FiCom application guideline (FiCom, 2012) for ETSI's MSS standards (ETSI, 2003a; ETSI, 2003b).

Background and Related Research

A mobile certificate is a recent Public Key Infrastructure (PKI) application. PKI is based on Public Key Cryptography discovered by Diffie & Hellman (1976) combined with certification of public keys as proposed by Kohnfelder (1978). An exhaustive presentation of the evolution and principles of PKI is found for example in (Gutmann, 2002). PKI is a basis for two-factor authentication and digital signature services.

The fundamental PKI component is often called “token”, which is a certified public/private key pair in Public Key Cryptography. A token is “soft” when the private key is stored in a file, which is usually encrypted. A token is “hard” when the private key is stored in hardware. Typical hard token examples are smartcards and cryptographic coprocessors in network servers. A hard token naturally provides far better protection against disclosure and tampering attacks than a soft token.

A typical hard token application is a smartcard implementation of an electronic identity (eID) card. A smartcard requires an attached smartcard reader and smartcard interfacing software in a computer using the eID. However, if the hard token is located on the SIM/USIM card of a mobile phone, then the eID can be used also in a computer without a hardware/software interface to a smartcard. A SIM card hosting a hard token is called a PKI SIM card. A PKI SIM card usually implements an eID, which is called “Mobile PKI” or “mobile certificate”. An assessment of Mobile PKI technology is published in (Oostdijk & Wegdam, 2009). Mobile PKI security is outlined in (Mobile, 2013). PKI SIM based digital signature solutions for e-banking are proposed in (Li, 2009).

Mobile Certificate in Finland

The mobile certificate developed by mobile phone operators in Finland is a hard eID token, which is used for authenticating the identity of a person and for approving a transaction or an agreement. Therefore it is used for granting access to electronic services and for digital signing of a text or a document (Mobiilivarmenne, 2011a). The private key of the mobile certificate is stored in the SIM card with the owner's personal information (Mobiilivarmenne, 2011c). The mobile operators have also signed a trust network agreement as Certification Authorities (CA) and agreed on a certification policy, which they manage and update (Mobiiliasiointivarmenne, 2011). The certification policy is an outcome of an agreement between mobile operators to guide decisions, achieve rational and common outcomes (Mobiilivarmenne, 2010).

Management of Cryptographic Keys

The mobile operators in Finland offer their subscribers mobile certificates, which are created according to the FiCom recommendation document (FiCom, 2012). The identity of a subscriber is checked before a mobile certificate is delivered or handed to the subscriber, when the subscriber makes a mobile certificate agreement with his/her mobile operator (Mobiilivarmenne, 2011a). The subscriber will get a certificate based Subscriber Identity Module (SIM) card, for which a Personal Identification Number (PIN) is included (Mobiilivarmenne, 2011b). The SIM card has been produced two pre-installed unique private keys, which cannot be read or tampered and can be used only by insertion of the correct PIN. The corresponding public keys are certified by the mobile operator together with subscriber personal information fetched from the Population Register of the Population Register Centre in Finland.

FiCom Recommendation

The FiCom recommendation issued by FiCom ry is an application guideline document for the ETSI MSS standards. The mobile certificate can be realized with the techniques, practices, limita-

tions and extensions implemented by various service providers in Finland. This is described in FiCom recommendation. The FiCom recommendation relies on ETSI TS 102 204, TR 102 206 and TS 102 207 standards and is based on the following techniques (FiCom, 2012):

- XML Schema Part1; Part 2
- Soap Version 1.2 Part 0: Primer; Part1: Messaging Framework; Part 2: Adjuncts
- XMLSignature
- WSDL 1.1
- PKCS#7
- Security Assertion Markup Language (SAML) v2.0.

Mobile Signature

A mobile signature can be implemented in a variety of ways with the use of capabilities of a mobile device (SIM/UICC infrastructure) and mobile network infrastructures. The mobile signature can be used in devices such as the mobile telephone, tablet computer, PDA, Laptop PC and remote telemetry unit with integral or external smartcards using a mobile network as a communications channel. (ETSI, 2003a)

The mobile signature can be enhanced by improving the protection against certain potential threats with the additional utilities such as "time-stamping" (i.e. describes the date and time at a given moment) and can be used to clarify that the transaction was made on behalf of the citizen's request on a specific time. The mobile signature is used in services were the user has initiated and wants to grant the permission to proceed with a transaction. The service may be initiated through the Internet, voice-call, interactive voice response systems and other electronic communications channels. Therefore face-to-face services are also possible. Some mobile signature technologies are shown in Figure 1. (ETSI, 2003a)

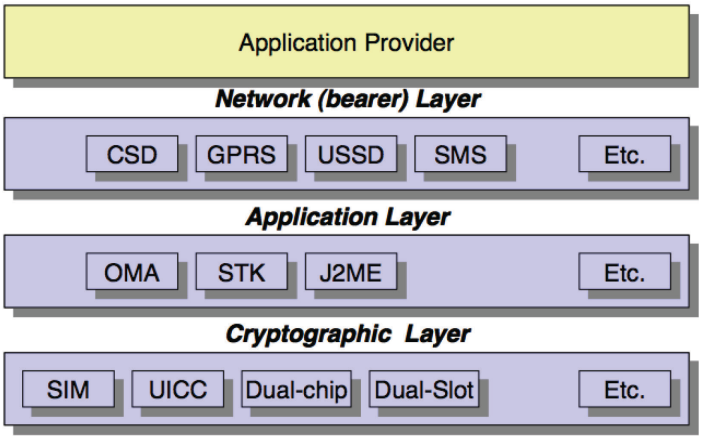

Figure 1: Modular Approach to Mobile Signature. (ETSI, 2003a)

Mobile Signature Design

The mobile signature functionality can be included in Server Side and Smart Card based implementations, which use any mobile network communication such as USSD, SMS, Circuit Switched, and GPRS as a communication channel. Even protocols like WAP can be used for signing, which then can be achieved by installing a suitable application in the mobile device that can be addressed by an application or service being used by the user. (ETSI, 2003a)

A server side signature creation is achieved with a signature "proxy" or "gateway" and the mobile signature is an encryption of an appropriate code such as Message Authentication Code (MAC). The signature is created by the server whenever the user has entered the PIN-code using the mobile device keypad. The smartcard based signature creation is achieved by a crypto processor, which can be implemented on a smartcard such as SIM-card and Universal Integrated Circuit Card (UICC). Therefore mobile operators have the role of "Smartcard Issuer". The difference between the server side and the smart card based signatures is that the server side signature validity cannot provide as high degree of confidence as the smart card based signature. A mobile smartcard implementation is shown in Figure 2 (ETSI, 2003a).

Figure 2: Typical mobile smartcard implementation. (ETSI, 2003a)

Mobile Signature Service (MSS)

The MSS can be defined as a service for users and application providers, where the mobile signature process is coordinated or managed (ETSI, 2003a). The MSS is provided from MSSP to service providers. The role of MSSP is to execute registration and certification procedures. A MSS has a Web Service Interface, which MSSP provides and/or implements as a Mobile Signature Web Service between end users and AP. (ETSI, 2003b)

Mobile Signature Service Provider (MSSP)

A MSSP distributes Mobile Signature Services to service providers with a high level of security. To provide such services there are only few procedures to execute for registration and certifica-

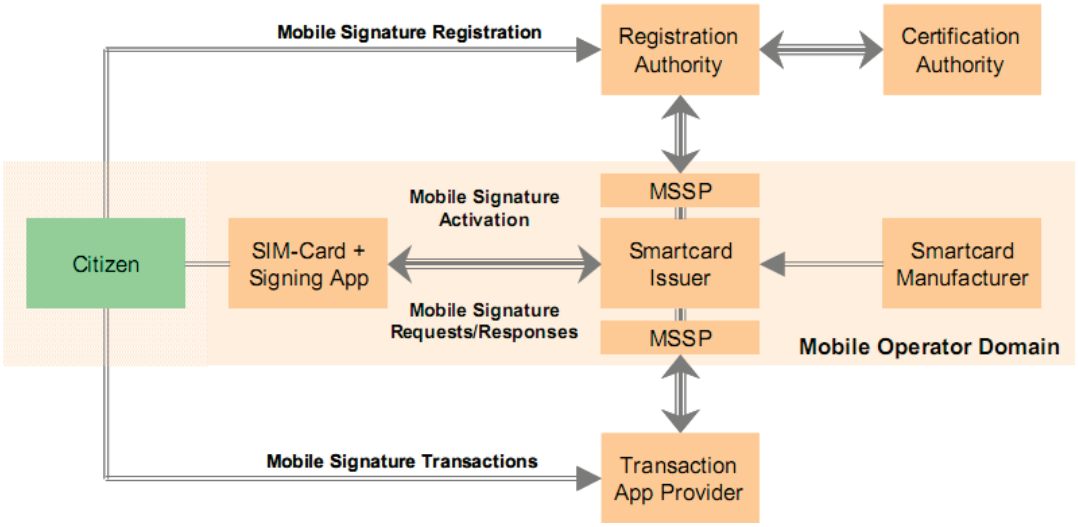

Figure 3: Mobile signature roles (ETSI, 2003a)

tion. The mobile PKI registration process needs four entities to be implemented: the Certification Authority (CA), the Registration Authority (RA), the MSSP, and the end user. A MSSP may possess a CA and optionally a RA. Mobile signature roles are shown in Figure 3. In Finland a MSSP acts as a CA and as a RA. (ETSI, 2003a; Mobiiliasiointivarmenne, 2011)

Mobile Signature Messaging Modes

There are two messaging modes, "synchronous" and "asynchronous", for achieving a mobile signature. In Finland, the concept "MSSP" generally refers to the system utilizing signature service roaming, formed by all operators. When trying to invoke the end-user mobile device for confirmation of a transaction, some steps may take time, such as determination of network connection etc. Therefore the best transaction processing mode for MSSP is "asynchronous". A multi request-response protocol is needed from the MSSP to get the "asynchronous" mode function properly. (ETSI, 2003b)

The messaging communication modes have following service messages: signature request, related signature response, status request, and related status response. Additional and optional service messages are receipt request and related receipt response. An AP uses messaging modes to request and receive related response from the AE. A signature request message requests a signature from an AE, which then requests the signature from the user through the HMSSP. The AE acknowledges the signature request message with a signature response message to the AP. The status request message mode is for inquiring from an HMSSP the completion of the previously submitted service request. The HMSSP reports the status of the signature event in the status response message. The receipt request message asks for an acknowledgment of the success or failure of the event to the user. The receipt response message is for acquiring the requested acknowledgment. The messaging modes use many HTTP events, which follow each other and are established in the AP's system. The messaging modes are useful for obtaining reference information about asynchronous communication at an early stage and therefore give high service reliability. (FiCom. 2012)

"Asynchronous Client-Server" Mode

Message exchange in the "asynchronous client-server" mode consists according to (FiCom, 2012) of the following steps shown in Figure 4:

1. The user establishes a connection to AP's server. The user wants to access a service or sign an electronic text or document. Therefore the AP server asks the user for information such as mobile phone number and spam prevention code (SPC).
2. The AP server displays the event number in the current business channel that identifies the signature event and guides the user to look on the device (mobile phone).
3. The AP server can now request a signature from an Acquiring Entity (AE), with which the Web Service Interface has been integrated. There is a mutual authentication between the AP server and the AE. The request can also include additional information of the user's identity, which the AP server may ask from Home Mobile Signature Service Provider (HMSSP).
4. When the signature request is received an MSSP event number is generated, which the AE returns to the AP server. A description of the event is indicated in the response message.
5. The AE makes sure that the information, which was requested, is in accordance with the service agreement. Afterwards the AE sends the message to the user's HMSSP, which is the user's operator. The HMSSP invokes the user's mobile phone for signature request if the user's SPC is correct and the user allows the signature service (incl. value-added services) to be requested by the AP server.

6. The user gets a signature request, which also shows the same event number that was shown in the AP server interface and therefore the user has to ensure that both correspond to each other. Afterwards the user can sign the event by inputting the given PIN code. The result is a digital signature, which is delivered to the HMSSP.
7. The HMSSP creates a PKCS#7 message by compiling a digitally signed message from the digital signature. This message is then attached as a part of to the signature response (MSS_StatusResp). The value-added service, which was requested from the AP server will also be processed and attached to the digitally signed message.
8. After the AP server has sent the signature request to the AE to be processed, the AP server asks for the completed signature response at specified intervals (MSS_StatusReq).
9. The HMSSP verifies the digital signature and reports the status of the signature request in the status response (MSS_StatusResp) to the AP server's (MSS_StatusReq). The completion of the signature is delivered as part of the status response.
10. The AP server links between the User identified in the signature response (a digital signature) and the one with the AP server's own database. When the AP server has identified the User, the result of the process is called authenticated. Therefore the AP server's interface can be changed to an authenticated interface.
11. When the AP server has been assured of User's identity, it can send a receipt of the completion of the event to the User through the same channel, which was used during the signature request, but using the receipt request instead (MSS_ReceiptReq).
12. The receipt message request is delivered to User.
13. The response of the receipt is delivered the same way as the signature response (MSS_ReceiptResp).

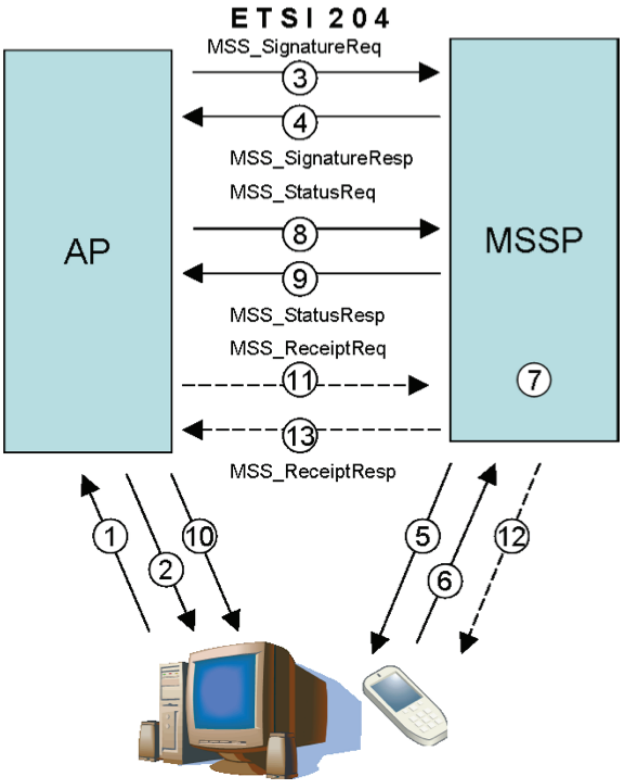

Figure 4: Signature Method – "Asynchronous Client-Server" Mode. (FiCom, 2012)

"Synchronous Client-Server" Mode

The only difference in "Synchronous Client-Server" compared to the "Asynchronous Client-Server" messaging mode is that the connection is established until the signature response is acquired from HMSSP to the AP server and therefore the MSSP sends the signature response to the AP server. This messaging mode is not recommended by FiCom, because of it consumes system resources of the AP server and the MSSP unnecessarily (FiCom, 2012), but ETSI declares that this messaging mode has to be supported by the MSSP. (ETSI, 2003b).

Application Provider's Interface

The message interface between AP and AE is created in the form of MSS service messages, which are described in (ETSI, 2003b). The AP sends a service request to the AE and receives a related service response. The service request of AP is a HTTP POST Request and the AE's response message is a HTTP Response. The MSS Service messages are included in SOAP envelopes, which are transmitted as HTTP messages.

SOAP Envelope

A SOAP envelope contains a header element (env:Header) and a content element (env:Body). (FiCom, 2012) The Header element is optional. It is useful for implementing XML signatures, which the FiCom recommendation does not address. The Body element is compulsory and one of the following message types is attached to it:

- MSS_SignatureReq (operation: MSS_Signature)
- MSS_SignatureResp (operation: MSS_Signature)
- MSS_StatusReq (operation: MSS_Status)
- MSS_StatusResp (operation: MSS_Status)
- MSS_ReceiptReq (operation: MSS_Receipt)
- MSS_ReceiptResp (operation: MSS_Receipt)

Each message type includes further attributes and specific sub-elements. The Web Service Description Language (WSDL) 1.1 specifies that the message element is covered inside the element specifying the name of the "operation". (FiCom, 2012)

Namespaces

The element specifying the SOAP Envelope reserves its own namespace:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
```

The element specifying the message type of the content element reserves namespaces for the ETSI MSS standard specifications and if necessary also for XML signature specifications and for value added services of signature requests specified by (FiCom, 2012):

```
<MSS_ReceiptReq xmlns=http://uri.etsi.org/TS102204/v1.1.2#
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:fi="http://mss.ficom.fi/TS102204/v1.0.0#" ...>
```

Several examples of each MSS service message are presented in (FiCom, 2012).

Error Handling

On error during of an event, a SOAP FAULT message is returned to the Application Provider. The SOAP FAULT message contains a status code of the error.

Implementation of Mobile Certificate Based Services

The FiCom recommendation and the Laverca Software Development Kit (SDK) are described as well as how these are utilized to implement proper MSS functionality.

Overview of FiCom Recommendation

To understand the potential of how a MSS works in practice there is need of familiarization techniques used in (FiCom, 2012) and ETSI's MSS standards (ETSI, 2003a; ETSI, 2003b)-

The currently supported messaging modes are asynchronous client – server and synchronous client – server. The synchronous messaging mode is not recommended. Strong mutual authentication and encryption is used in messaging between all entities.

An agreement between the AP and AE is needed to get access to a MSS. Once an agreement has been made with the AE, following information will be provided:

- AP name (Application Providers name): The AP name is displayed on the terminal device
- AP_ID (Application Provider ID) & AP_PWD (Application Provider Password): The AP_ID and AP_PWD are used as an addition to strong mutual authentication and encryption between AP and AE

The FiCom recommendation is in brief:

- Currently supported message formats are MSS_SignatureReq, MSS_SignatureResp, MSS_StatusReq, MSS_StatusResp, MSS_RecepitReq, MSS_ReceiptResp.
- The process of the user registration is an internal matter for each HMSSP.
- The MSS Service registration messages are not addressed
- XML signed service messages are currently not supported.
- The MSISDN is used for finding the user and HMSSP.
- A user can be also found with the UserIdentifier whose mandatory postfix additionally identifies HMSSP, but is currently not supported.
- The supported character sets in service requests are GSM, UTF-8 and UCS2.
- The supported character sets on a terminal device are GSM 03.38 and UCS2.
- Only the UTF-8 characters are available and included in the GSM 03.38 character set.
- The HMSSP can provide six different signature services:
    - Authentication
    - Anonymous Authentication (currently not supported)
    - Signature of plain text content
    - Signature of digest content (currently not supported)
    - Issuing consent
    - Operator service for authentication
- Each HMSSP signature service is a separate signature profile, which indicates the desired service.
- The users can prohibit signature profiles through his/her own mobile subscription.
- AdditionalServices is an added value service used as an expansion for the MSS standard:
    - Mobile phone spam prevention (SPC)
    - An event ID
    - AE validation (currently not supported)
    - User's language preference (currently not supported)
    - PersonIdentity service
- The format of signature requests is standardized.

- The user's certificate is supplemented with a digital signature format base64-encoded PKCS#7 or PKCS#1 (currently not supported).
- It is mandatory to synchronize the system clock for AE, RE, and HMSSP with the NTP service. It is also recommended for AP.

Laverca SDK

The Laverca SDK is a Mobile Signature Service Application Programming Interface (MSSAPI), which is an Open Source implementation of ETSI TS 102 204 client software in Java. In addition to the ETSI standard it also supports the FiCom recommendation functionality and the developer does not need to be familiar with the asynchronous client – server messaging mode operating principle. (Laverca, 2012)

Network Service

In order to implement mobile certificate based network services, the following is needed:

- Web server with a platform (Linux, Windows)
- Certificate (server certificate)
- Agreement with the AE
- Secure communication between all parties.

A server that acts as a web server with installed operating system (Linux or Windows) is required. The web server can be installed using open source software or license based web server software such as Apache or Microsoft IIS (Internet Information Services). A server certificate is needed for using the mobile certificate service for authentication and for encryption of the communication between AP and the AE server (mutual authentication). In addition, the server certificate is used for encryption when users visit the web page. The certificate must be installed on the server side.

An agreement with the operator is needed for setting up a mobile certificate based service. Once an agreement has been made with the operator additional authentication information will be provided (Lacerca, 2012):

- AP name (Application Providers name)
- AP_ID (Application Provider ID)
- AP_PWD (Application Provider Password)
- MSS_Signature URI (Operator's Signature request Uniform Resource Identifier)
- MSS_Status URI (Operator's Status request Uniform Resource Identifier)
- MSS_Receipt URI (Operator's Receipt request Uniform Resource Identifier)-

The additional authentication information is used for establishing a connection to the operator's server and using the appropriate MSS-service URI for sending a SOAP envelope as an HTTP message. Data communication between the parties is encrypted and also secured by mutual authentication. To implement the mobile certificate services according to FiCom recommendations for data communication the MSS standard must be used. (Laverca, 2012)

A MSS service message sent to the operator is based on a HTTP POST request and the related response from the operator is based on the HTTP response. The communication method that is used can be "asynchronous client-server" or "synchronous client-server" (not recommended). (FiCom, 2012)

MSS Implementations

The following software and development kits are needed for the MSS implementations described in this chapter:

- Eclipse Integrated Development Environment (IDE) framework V3.7.1 indigo
- Java Development Kit Standard Edition (JDK SE) V1.7.0_03
- Laverca V1.01 SDK
- Apache ant V1.8.3
- Bash script
- Java Server Page
- Ubuntu server V11.10
- Apache Tomcat V7.0.26

Eclipse is needed for development of Java applications with the utilization of Laverca SDK. The Java JDK SE and Apache Ant are needed for compilation of Java applications. The bash script was used for running Java application.

MSS implementations user authentication and electronic text signing were designed with Java Server Page (JSP), which makes it possible for software developers to create dynamically generated web pages based on HTML, XML or other types of document. JSP uses the Java programming language. Whenever the JSP page is visited for the first time, the server will compile the JSP file and then run it. Each test case executes a bash script within the JSP. Then the bash script runs a Java application.

As a MSS implementation server the operating system Linux (Ubuntu server V11.10) and the Apache Tomcat v7.0.26 were installed. Apache Tomcat supports web applications.

MSS implementation of user authentication and electronic text signing is described in this chapter. The step-by-step user experience of a combined user authentication and text signing MSS implementation is presented in Appendix A.

User Authentication

With preinstalled correctly configured software, authentication can be achieved. To implement this MSS, three JSP pages following each other were needed. The JSP page contains an HTML form, which redirects the user to the second JSP page when the user has inserted a mobile phone number and a SPC.

The second JSP page uses the request.getParameter(“name”) command for reading the attributes mobile phone number and the SPC, which the user inserted in the first JSP page. To be able to use the user input in the third JSP page, there is a need to execute the session.setAttribute(“key”, value) command for both attributes.

In the second JSP page a sequence number is generated with the System.currentTimeMillis() command and this number is used as an application provider transport ID (APtransID). There is also an event ID (eventID) shown to the user on the JSP page. The application provider can create the eventID differently. The eventID was created by shortening the APtransID to the last 4 digits and adding a letter before the first digit. There is also a need of an “if statement”, which will redirect the user to the third JSP page if the user has inputted the mobile phone number and SPC with the correct syntax.

The third JSP page uses objects in the Java classes InputStreamReader and BufferedReader for reading line by line the system output and for getting user information such as the user’s name,

electronic service ID or other related information of the user. This information is asked for in a signature request.

The InputStreamReader class is used within the try catch method. The try catch method uses the session attributes from the second JSP page and executes the command. The command can either run directly the Java application or use a bash script that executes a Java application. Both use the mobile phone number, the SPC, the eventID and the APtransID as parameters. Due to security reasons, a bash script is used for hiding the path of the classes and the Java file.

The Java application takes 4 arguments, which are the ones used by the bash script.

To be able to authenticate the user, access to the operator's mobile certificate service is needed. This is achieved by acquiring a server certificate, AP_ID, AP_PWD, MSSP Signature Uniform Resource Identifier (URI), MSSP Status URI, and MSSP Receipt URI.

The server certificate is used for mutual authentication and encryption. The SSL/TLS support is based on the Java Virtual Machine's standard support. The "keystore" and "truststore" files are used by the Java Virtual Machine. Both are Java keystore files. The "keystore" file holds a private key and a server certificate. The private key and server certificate are created when an agreement with the operator is made. The "truststore" file holds the operator's server certificate, which is public. The AP_ID and AP_PWD are used as an additional authentication for a signature request.

The locations of the Java "keystore" and "truststore" files are defined in the Apache XML configuration with the specified password for accessing these Java keystores. The MSSP Signature Uniform Resource Identifier (URI), MSSP Status URI and MSSP Receipt URI are also defined in Apache XML configuration file, which the Java application will use when making the appropriate request for signature, status or receipt.

Electronic Text Signing

This MSS implementation was created in the same way user authentication and it uses authentication test as a template. Changed parts are described in this subchapter.

The first JSP page can be created as stand-alone or be combined it with the user authentication implementation. If created as stand-alone, the first JSP page contains a form with input fields for mobile phone number, SPC and text area. The additional text area is attached to the form. If combined with the user authentication implementation, the first JSP gets the values of mobile phone number and SPC from the third JSP page of the user authentication implementation. The mobile number phone and SPC can be obtained using the session.setAttribute("key", value) command on the third JSP page of the user authentication implementation. The additional text area is also attached to the form. This means that there are five JSP pages following each other. The text area is used for signing and allows a maximum length of 160 characters.

The second JSP page saves the user's inputted text in the text area to a text file with the format ISO-8859-1. The text file will be used within the Java application.

The third JSP page will execute a bash script, which executes the Java application in the same way as in the user authentication implementation.

Discussion and Evaluation

The contribution of this paper is to show how IT services using mobile certificates can be designed with FiCom application guideline (FiCom, 2012) for ETSI's MSS standards (ETSI, 2003a; ETSI, 2003b). A MSS solution combining user authentication with electronic signing of plain text has been implemented and successfully tested on Internet. Direct access from a smartphone to the

Subscriber Identity Module (SIM) could however simplify mobile certificate based solutions for IT services.

Conclusions

There are still certain things that might be done differently or maybe in more detail. More consideration of security might be required for protection against future threats. Due to technology advancements and other ways of implementing MSS, the described MSS implementations are possibly not the best solutions. Future research could use the Interactive Voice Response (IVR) and other communication channels as new methods for providing mobile certificate based services.

References

Diffie, W. & Hellman, M. (1976). New Directions in Cryptography. *IEEE Transactions on Information Theory, 22* (6), 644-654

ETSI. (2003a) TR 102 203 V1.1.1. Retrieved December 1, 2012 from http://docbox.etsi.org/EC_Files/EC_Files/tr_102203v010101p.pdf

ETSI. (2003b), TS 102 204 V1.1.4. Retrieved December 1, 2012 from http://docbox.etsi.org/EC_Files/EC_Files/ts_102204v010104p.pdf

FI: Vetuma eID and ePayment service to be updated following rapid growth (2011), Retrieved December 1, 2012 from http://www.epractice.eu/en/news/5310055

FiCom. (2012), FiCom's (The Finnish Federation for Telecommunications and Teleinformatics) application guideline for ETSI's MSS standards: V2.1. Retrieved December 1, 2012 from http://www.mobiilivarmenne.fi/documents/MSS_FiCom_Implementation_guideline_2.1.pdf.

Gutmann, P. (2002). PKI: It's Not Dead, Just Resting. *Computer, 35* (8), 41-49

Kohnfelder, L. (1978). Towards a Practical Public-Key Cryptosystem. BSc Thesis. Department of Electrical Engineering, MIT, Cambridge, Massachusetts, USA. Retrieved March 10, 2013 from http://groups.csail.mit.edu/cis/theses/kohnfelder-bs.pdf

Laverca (2012). Retrieved December 1, 2012 from https://sourceforge.net/projects/laverca/files/1.01/

Li, Q. (2009). *The E-bank digital signature solution based on PKI-SIM cards*. Proceedings of the IEEE International Conference on Communications Technology and Applications ICCTA '09. USA: IEEE Press, 900-902

Mobiiliasiointivarmenne. (2011), Varmennepolitiikka Operaattoreiden mobiiliasiointivarmenteita varten: V1.1. (In Finnish). Retrieved December 1, 2012 from http://www.mobiilivarmenne.fi/documents/Mobiiliasiointivarmenne-Varmennepolitiikka.pdf

Mobiilivarmenne. (2010), Mobile certification launch in Finland. Retrieved December 1, 2012 from http://www.mobiilivarmenne.fi/en/bulletin/mobile-verification-launch-in-finland

Mobiilivarmenne. (2011a). Mobile certificate (Mobiilivarmenne). Retrieved December 1, 2012 from http://www.mobiilivarmenne.fi/en/

Mobiilivarmenne. (2011b), FAQ. Retrieved December 1, 2012 from http://www.mobiilivarmenne.fi/en/faq/

Mobiilivarmenne. (2011c). Mobile Certificate in wide use in Finland. Retrieved December 1, 2012 from http://www.mobiilivarmenne.fi/en/bulletin/mobile-certificate-in-wide-use-in-finland

Mobile PKI. (2013). Mobile PKI Security. White Paper. Nexus. Retrieved March 11, 2013 from http://www.nexussafe.com/Global/pdf/white papers/Nexus White Paper-Mobile PKI-EN.pdf

Oostdijk, M. & Wegdam, M. (2009). Mobile PKI A technology scouting for security and use of mobile authentication technologies. SURFnet. Retrieved March 10, 2013 from http://www.terena.org/news/community/download.php?news_id=2528

Appendix A: User Experience of an Implemented MSS Combining User Authentication and Text Signing

A web page for insertion of a mobile phone number and a SPC is shown in Figure A1. After insertion of a mobile phone number and a correct related SPC, mouse clicking on the "Logga in/login" button results in

- that the web browser switches to web page showing the inserted mobile phone number and a created event ID (see Figure A2), and
- that the user's mobile phone receives an pop-up SMS message containing the same event ID (see Figure A3).

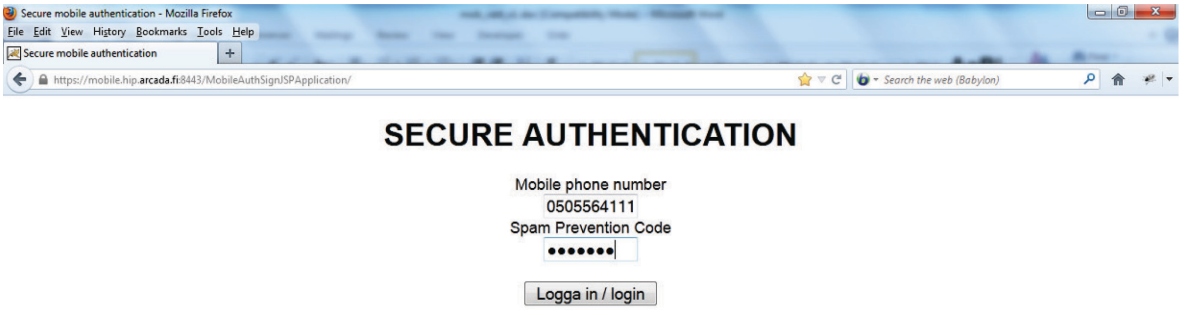

Figure A1: Web page for insertion of a mobile phone number and a SPC.

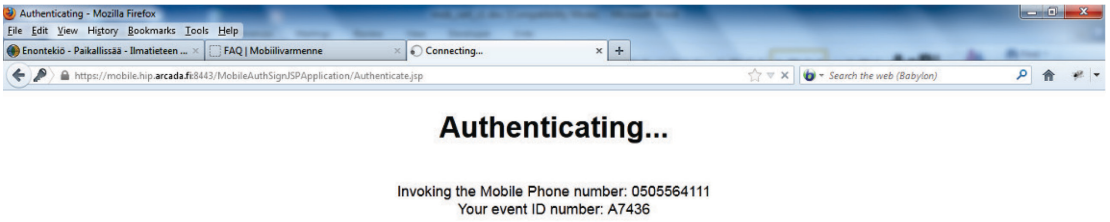

Figure A2: Web page showing an inserted mobile phone number and a created event ID.

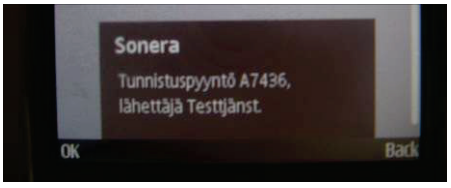

Figure A3: A received pop-up SMS message in the user's mobile phone showing the same event ID as the web page in Figure A2.

After OK acknowledgement of the received pop-up SMS message, the user's mobile phone receives a second pop-op SMS message informing in Finnish about the user data (name, electronic communication code, age and gender), which will be transmitted over the network connection (see Figure A4).

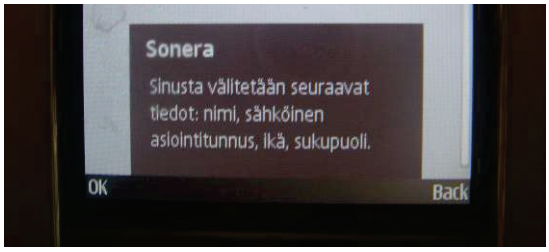

Figure A4: A received pop-up SMS message in the user's mobile phone showing in Finnish the user data, which will be transmitted over the network connection.

After OK acknowledgement of the received second pop-up SMS message, the user's mobile phone receives a third pop-op SMS message for insertion of the PIN code for the private mobile certificate key stored in the user's SIM/USIM card (see Figure A5).

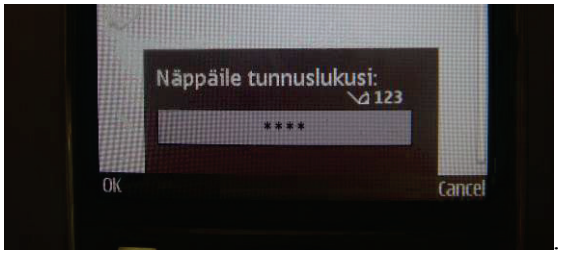

Figure A5: A received pop-up SMS message in the user's mobile phone asking for insertion of a PIN code for the private mobile certificate key stored in the user's SIM/USIM card.

After insertion of the correct PIN code and OK acknowledgement of the received third pop-up SMS message the user is authenticated for a web page in which the user can insert text to be signed with his/her mobile certificate (see Figure A6).

Figure A6: A web page in which the user can insert text to be signed with his/her mobile certificate.

After insertion of the text to be signed and mouse clicking on the "Sign" button results in

- the web browser switches to web page showing the mobile phone number of the user and a created event ID (see Figure A7)
- the users mobile phone receives an pop-up SMS message containing the same event ID (see Figure A8)

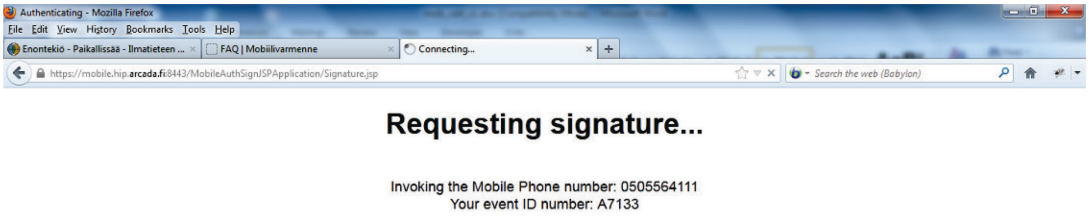

Figure A7: Web page showing the mobile phone number of the user and a created event ID.

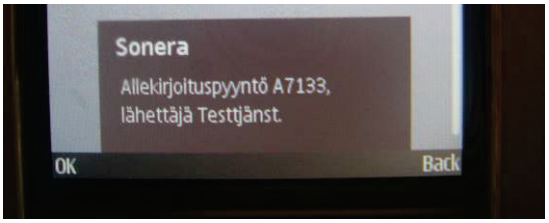

Figure A8: A received pop-up SMS message in the user's mobile phone showing the same event ID as the web page in Figure 10.

After OK acknowledgement of the received pop-up SMS message, the user's mobile phone receives another pop-op SMS message informing in Finnish about the user data (name, electronic communication code, age and gender), which will be transmitted over the network connection (see Figure A9)

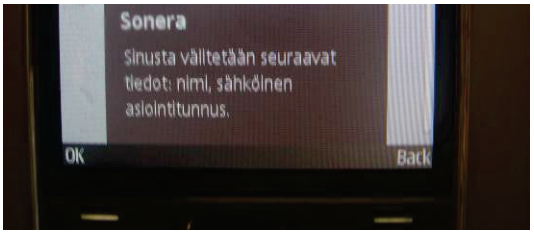

Figure A9: A received pop-up SMS message in the user's mobile phone showing in Finnish the user data, which will be transmitted over the network connection.

After OK acknowledgement of the received second pop-up SMS message, the user's mobile phone receives a another pop-op SMS message showing the inserted text to be signed (see Figure A10).

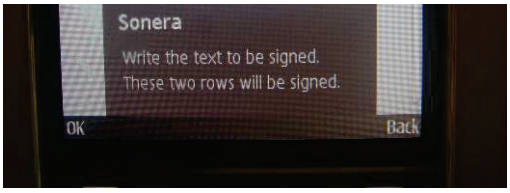

Figure A10. A received pop-up SMS message showing the inserted text to be signed.

After OK acknowledgement of the received second pop-up SMS message, the user's mobile phone receives a third pop-op SMS message for insertion of the PIN code for the private mobile certificate key stored in the user's SIM/USIM card (see Figure A11).

After insertion of the correct PIN code and OK acknowledgement of the received pop-up SMS message the user is authenticated for a web page in which is shown the insert text to be signed together with the base64 encoded signature (see Figure A12).

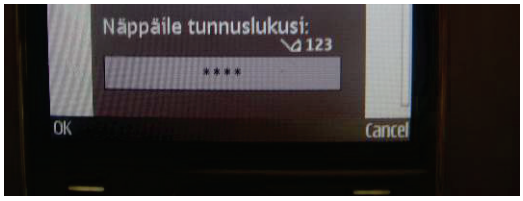

Figure A11. A received pop-up SMS message in the user's mobile phone asking for insertion of a PIN code for the private mobile certificate key stored in the user's SIM/USIM card.

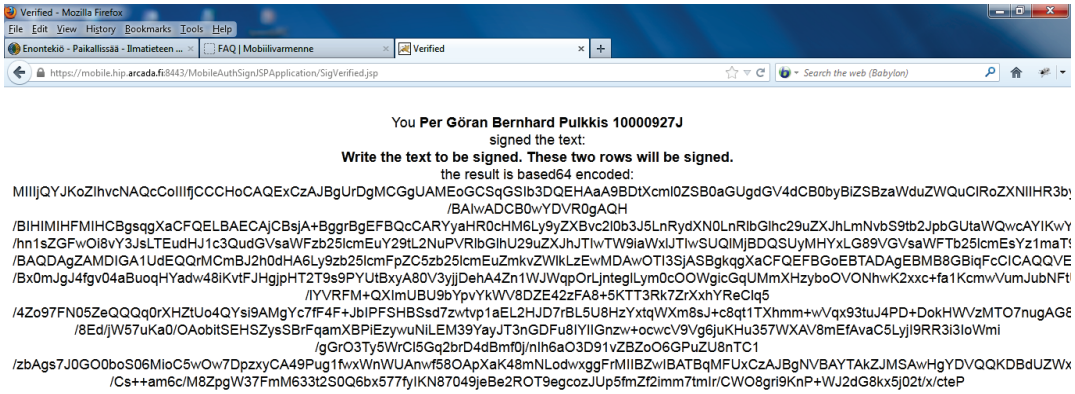

Figure A12. A web page showing the insert text to be signed together with the base64 encoded signature.

Appendix B: Abbreviations and Definitions

AE	Acquire Entity, the AE offers a Web Service Interface to an AP for a mobile signature service complying with the FiCom recommendation.
AP	Application Provider, the AP needs the user's signature and is AE's customer.
AP_ID	Application Provider's contact information in MSSP systems
AP_PWD	Application Provider's password in AE's system
CA	Certificate Authorities
ETSI	European Telecommunications Standards Institute, is an European Standards Organization which produces globally applicable ICT standards for. These standards are for mobile, radio, fixed, converged, broadcast and Internet Technologies.
FiCom	The Finnish Federation for Communications and Teleinformatics, is a Finnish Information and Communications Technology (ICT) sector trustee and cooperation organization. FiCom members are companies such as telecom, Internet and cable operators.
FTP	File Transport Protocol
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
HMSSP	Home Mobile Signature Service Provider, the user's home operator
ICT	Information and communications Technology
IIS	Internet Information Service
ISO	International Organization for Standardization
IVR	Interactive Voice Response

JSP	Java Server Page
MAC	Message Authentication Code
MSISDN	Mobile Subscriber Integrated Services Digital Network Number
MSS	Mobile Signature Service
MSSP	Mobile Signature Service Provider, MSSP provides HMSSP services to users and potentially AE services to AP and/or AEs.
MSSPAPI	Mobile Signature Service Provider Application Programming Interface
NTP	Network Time Protocol
PC	Personal Computer
PDA	Personal Digital Assistant
PIN	Personal Identification Number
PKCS	Public Key Cryptography Standards
RE	Routing Entity, A RE routes traffic between an AE and a HMSSP. A RE can be a component of AE or HMSSP systems or a separate system of a TTP.
SAML	Security Assertion Markup Language
SIM	Subscriber Identity Module
SMS	Short Message Service
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
TTP	Trusted Third Party
UCS2	Universal Character Set – 2-bit
UICC	Universal Integrated Circuit Card
USIM	Universal Subscriber Identity Module
USSD	Unstructured Supplementary Service Data is a GSM communication technology that is used to send text between a mobile phone and an application program in the network
UTF-8	Universal Character Set (UCS) Transformation Format – 8-bit
VPN	Virtual Private Network
WSDL	Web Services Description Language
XML	Extensible Markup Language

Biographies

Göran PULKKIS received in 1983 his doctoral degree at Helsinki University of Technology and is presently researcher in computer science and engineering at Arcada University of Applied Sciences, Helsinki, Finland. His current research interests include network security and applied cryptography.

Farzan YAZDANI received in 2012 his Bachelor of Engineering degree in Information Technology at Arcada University of Applied Sciences and has worked as a research assistant in a project called ‘Elli’, in which his task was to develop healthcare ICT services.