An Example of Web Service Based Secure Mobile Application

Milan Marković, Goran Đorđević

Abstract — ***In this paper, a design and programming of JAVA applications on mobile phones that securely connect to Web services are described. We considered a Web service scenario where mobile phone user produces a cryptographic signature in the JAVA application using the smart card. Data is encrypted using a crypto Xlet JAVA application installed on mobile phone with CDC configuration. The user uses XML signature to wrap a cryptographic signature into the SOAP request and sends the request over to the remote Web service endpoint implementation. Web service performs request processing and sends SOAP response back to the WSA (Web Service API) framework. WSA processes the SOAP response and display the status to the mobile user. The example described is carried out within the EU IST (Secure, interoperable cross border m-services contributing towards a trustful European cooperation with the non-EU member Western Balkan countries) [1].***

Keywords — **Java mobile application, Mobile phone with CDC configuration, Smart card, SOAP protocol, SWEB, XML Signature, Xlet, Web service.**

I. INTRODUCTION

JJava 2 Micro Edition (J2ME) is a runtime environment for resource-constrained environments. J2ME includes specific virtual machines, configurations and profiles for various environments and needs. With an appropriate configuration and profile, J2ME applications could be executed within pagers, mobile phones, PDAs, set-top boxes and automobile navigation systems, just to mention some [2].

It defines configurations (hardware model configurations with supporting software) and profiles (supporting software APIs) that allow Java to be used on small and embedded devices.

The Java Specification Request 172 (JSR 172) specifies standardized client-side technology to enable J2ME applications to consume remote services on typical web services architectures.

JSR J2ME devices are only expected to consume Web services exposed by service endpoints. This scenario is depicted in Figure 1.

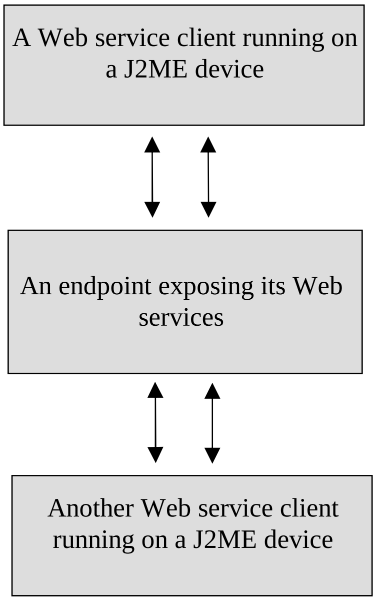
A Web service client running on a J2ME device
An endpoint exposing its Web services
Another Web service client running on a J2ME device

Figure 1: Web service consuming services exposed by a Web service endpoint

Web services are a good way to allow smaller devices and applications to use the processing power available on larger machine. Java 2 Micro Edition (J2ME) is a runtime environment for resource-constrained environments. Web Services APIs (WSA) for J2ME uses the idea of stub classes, so other technology components such as cryptography, XML signature and Java Card technology have to fit into WSA stub classes.

II. IMPLEMENTATION ASPECTS

In a process of development of JAVA Mobile Application we used J2ME developing environment. The J2ME is a runtime environment for resource-constrained environments. J2ME includes specific virtual machines, configurations and profiles for various environments and needs. With an appropriate configuration and profile, J2ME applications could be executed within pagers, mobile phones, PDAs, set-top boxes and automobile navigation systems, just to mention some [2].

Bouncy Castle APIs - In order to encrypt sensitive data we used Bouncy Castle Cryptography APIs. Bouncy Castle is an open source Java API for encrypting and decrypting data. There is a lightweight package that is suitable for MIDP applications where only a fraction of the API will be used at any one time.

Milan Marković, Banca Intesa ad Beograd, Milutina Milankovića 1c, 11070 Novi Beograd, Srbija (tel: 381-11-3770187; e-mail: mmarkovic@bancaintesabeograd.com).

Goran Đorđević, The Institute for Manufacturing banknotes and coins NBS, Pionirska 2, 11030 Beograd, Srbija (tel: 381-11-3691361; e-mail: goran.djordjevic@nbs.rs).

Obfuscation process - One problem inherent to most mobile devices is the limited amount of memory. As with most any library you use, only a small portion of the code is typically needed by your application. One common way to eliminate unused code, and at the same time make it more challenging to reverse engineer an application, is to use a Java obfuscator. We used open source obfuscator ProGuard.

Security and Trust Services API (SATSA) - Security and Trust Services API is a new API that provides additional security capabilities to the J2ME CLDC platform. It specifies a collection of APIs that provide security and trust services for J2ME CLDC by integrating a Security Element (SE).

The SE is a hardware or a software component in a J2ME device. It provides the following features:

- Secure storage to protect sensitive data.
- Cryptographic operations.

The support for cryptographic smart cards is of particular interest to developers writing J2ME applications for smart phones. Keys and certificates can be stored on the smart card and data can be signed without the private key ever leaving the card. High-end smart cards are temper resistant and provide authentication schemes, such as requiring a PIN or a password before access to the smart card is granted. This way security is dependent on the smart card not being compromised. Private keys do not have to be stored on diverse insecure clients, enabling vendors to focus on keeping the smart card secure from physical tempering and, just as important, smart card API exploitation[4].

This client application comprises of following functionalities:

- Graphical User Interface (GUI) for presenting business functionalities to the end user,
- Business (core) functionalities of the application – SWEB functionality: m-residence certificate.
- Security functionalities,
- Communication.

The SWEB secure JAVA mobile application objects and community are represented on Figure 2.

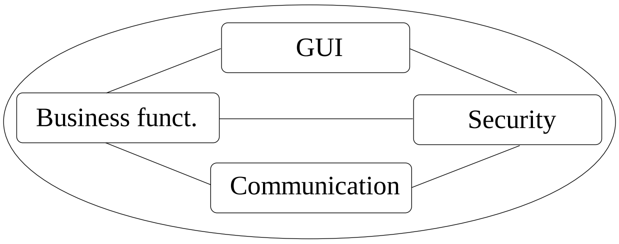

Figure 2: JAVA mobile application community

III. SWEB ARCHITECTURE

SWEB uses defines an SWEB community, consisting of [1]:

- Citizens,
- Civil servants,
- Administrators.

Depending on the scenario, it might be necessary to introduce some other roles, like delegates of either citizens or civil servants and several levels of administration here, but it is assumed that for the functional purpose of the SWEB system those roles doesn´t matter as they usually don´t influence the platform processes directly, but using itself delegates which are actually belonging to one of the groups mentioned.

Citizens are the primary users SWEB targets. Using a mobile device, they access the system, initiate requests or receive notifications pushed by the platform. Citizen delegates are handled like civil servants, as they can´t access the system for someone else, due to the nature of the SWEB authentication mechanism. Instead they are forced to get help by a civil servant, actually initiating the request.

Civil servants are the right hand of the SWEB platform. Where SWEB is only able to check requests for security constraints, Civil Servants may approve or decline requests on a semantic legal level that is elusive by computer systems. They are also necessary when it comes to delegate requests by other citizens or civil servants from other municipalities.

Administrators are those responsible for administration of the community as a whole or the platform and the involved community members in detail. In SWEB there are administrators that are actually handling the technical maintenance and administrators that are able to hand out certificates to civil servants.

Those three roles are actually directly mapped to system roles, when it comes to the technical realization. While administrators are mainly used for PKI administration and security certificate handling, civil servants and citizens are roles that need to be integrated into the platform logic directly to distinguish between them, when it comes to access control, authorization and to business logic decisions. Therefore the decision to use SAML assertions with integrated roles came naturally [7].

By using SAML together with WS-Security it was a small step to extend the server-server communications to use this technology as well. For that reason, internally there was a fourth role defined. The role of each server is important as it is necessary to be defined for intercommunication between the various SWEB platforms. However, although the communication is established between two servers, the documents delivered are meant to be assured and signed by civil servants, to assure responsibility by a human being.

IV. SWEB WEB SERVICE DESCRIPTION

Principal Residence case

A citizen of city A needs a certification for his principal residence in city A. He will contact the municipality of city A for that.

In this process, he sends a request to this municipality first. The municipality creates his mRCertificate. He gets a final notification message and can pick up his mRCertificate afterwards [8].

In a more detailed view, there are three system objects belonging to the municipality. It is the SWEB Platform, the local IT Infrastructure (legacy system) and the civil servant as the human actor. The citizen sends his request to the SWEB platform, which in return first sends a notification back about the incoming request and afterwards forwarding the request to the civil servant for approval. After this, the request is send to the legacy system, where the mRCertificate is created.

After that, the civil servant has to approve this mRCertificate. Furthermore, there is a final notification send to the mobile to inform the citizen that he can pick up his mRCertificate. Finally, the mRCertificate needs to be retrieved by the citizen using the document retrieval service described before (See Figure 3).

Implemented security functionalities for the SWEB platform are following:

1. WS-Secured SOAP communication with end users according to the scenarios.
2. Signature verification of signed and timestamped requests, SAML token and e/minvoices as well as validations of certificates from all parties.
3. Create UserProfile from Civil Servant's X.509v3 certificate.
4. Timestamping documents signed by Civil Servants.
5. Signing and timestamping cross-border mRCertificate request that should be sent to the other municipality.
6. Requesting and receiving SAML token for Civil Servant and for the SWEB platform.
7. WS-Secured SOAP communication with the Interaction Tier Manager of another SWEB platform according to the cross-border scenario.
8. Locate and validate certificates by using corresponding functions of the XKMS protocol and communicates with the XKMS server via SOAP communication.

Secondary Residence case

A citizen of city A needs a certification for his secondary residence in city B. In this scenario, the citizen contacts his municipality in city A. Now, there is a communication between the two municipalities, as city B is the municipality responsible for creating this mRCertificate. The citizen gets his notification and mRCertificate through the municipality of City A (see Figure 4).

As in the previous scenario, after the citizen is notified of the incoming request, the request is delivered to the civil servant of city A. He approves the request and sends it to the municipality of city B. There, the request has to be approved from the civil servant of city B. Afterwards the request will be processed by the legacy system of city B where the mRCertificate is created. The mRCertificate will be approved again by the civil servant of city B. Now, the mRCertificate goes to the municipality of city A where it is approved by the civil servant. Finally, the notification message is send and the citizen can fetch his mRCertificate.

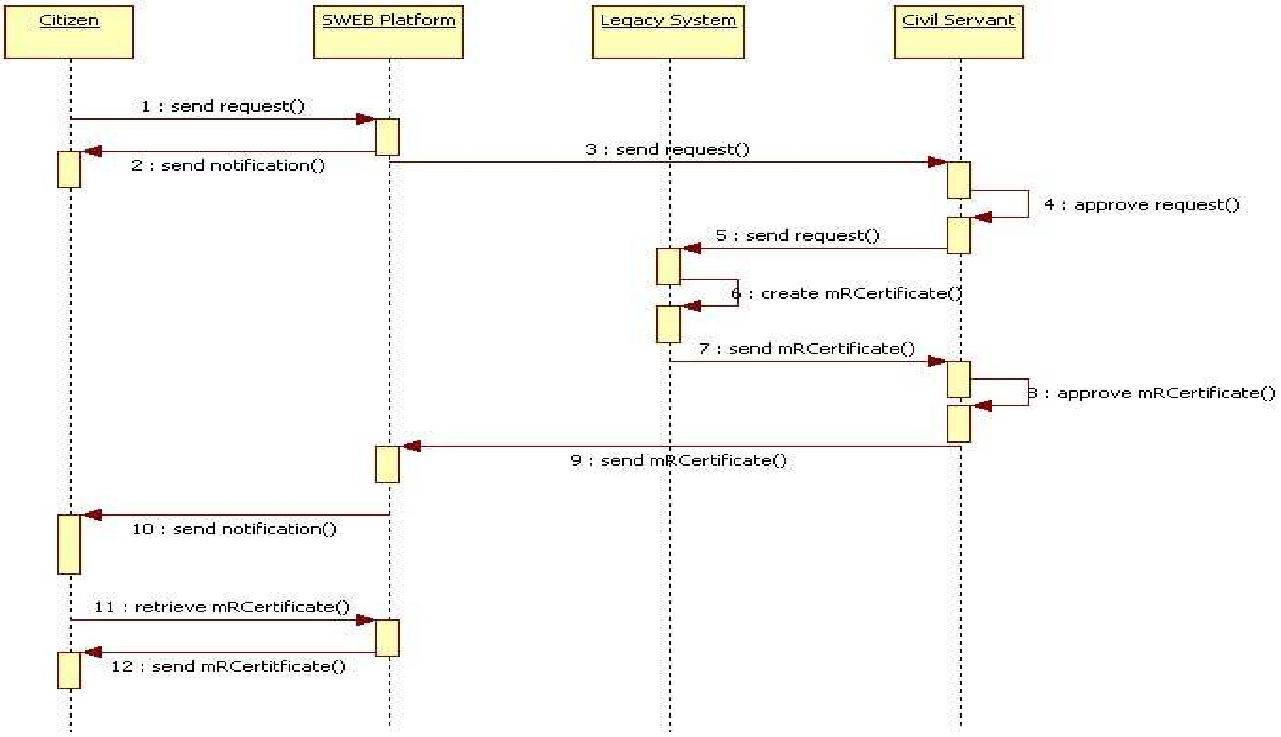

Figure 3: Principal Residence's scenario details

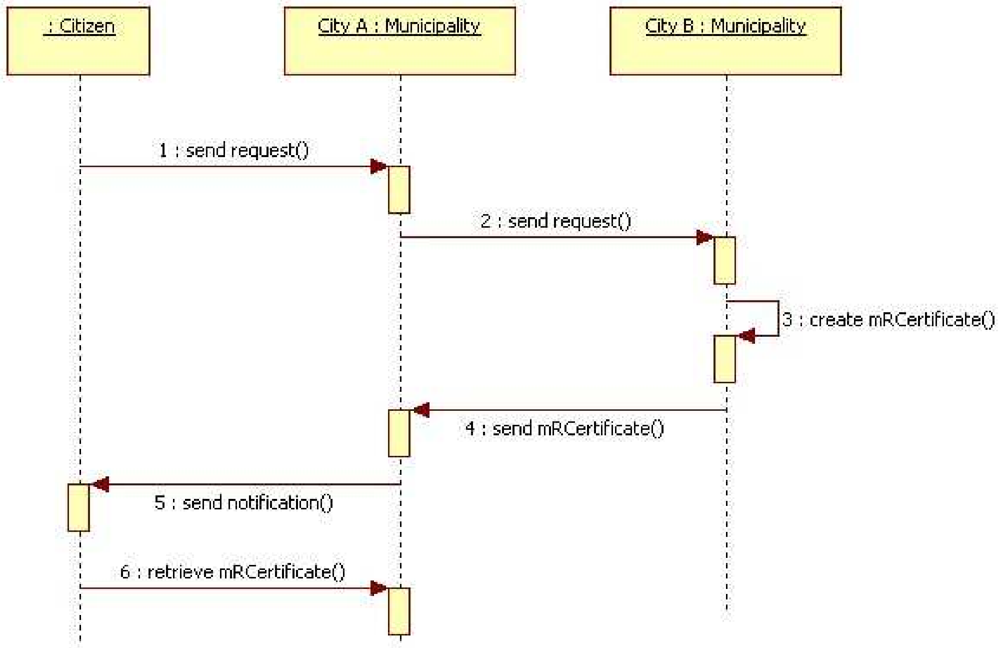

Figure 4: Secondary Residence's scenario details

V. CONCLUSION

This work is related to the consideration of some possible SOA-based m-government online systems, i.e. about secure mobile communication between citizens and companies with the small and medium governmental organizations, such as municipalities. In fact, we elaborated a secure m-government framework which is based on secure JAVA mobile application. We described the two SWEB Web service cases: principal residence and secondary residence. In both anayzed cases citizen required his (her) residence certificates using mobile phone with installed secure SWEB mobile midlet application.

We considered the scenario where private asymmetric keys and digital certificates stored on the smart card and data can be signed without the private key ever leaving the card.

ACKNOWLEDGEMENT

This work is being carried out in the context of the IST international cooperation project SWEB (044979). This paper is based on the work performed within the context of this project and the authors would like to acknowledge all SWEB partners.

DISCLAIMER

This reseacrh outlined in this paper has been undertaken with the financial assistance of the European Community. The views expressed herein are those of SWEB Consortium and can therefore is no way be taken to reflect the official opinion of the European Commision. The information in this document is provided as is and no guarantee or warranty is given to state that the information is fit for any particular purpose. The user therefore uses the information at their sole risk and liability.

LITERATURE

[1] SWEB Project Homepage, http://www.sweb-project.org.

[2] Introduction to J2ME Web Services, C. Enrique Ortiz, http://developers.sun.com/techtopics/mobility/apis/articles/wsa/.

[3] MIDP 2.0: SATSA-APDU API Developer's Guide, version 1.0, February 2nd, 2007. Forum Nokia, Handbook. Mill Valley: University Science, 2007.

[4] Building a secure SOAP client for J2ME, Part 1: Exploring Web Services APIs (WSA) for J2ME", Bilal Siddiqui, 16 Jun 2006, http://www-128.ibm.com/developerworks/edu/

[5] "Understanding the Web Services Subset API for Java ME", C. Enrique Ortiz, March 2006, http://developers.sun.com/techtopics/mobility/midp/articles/webservices.

[6] MIDP 2.0: SATSA-APDU API Developer's Guide, version 1.0, February 2nd, 2007. Forum Nokia, *Handbook.* Mill Valley, CA: University Science, 2007.

[7] Spyridon Papastergiou, Athanasios Karantjias, Despina Polemi, and Milan Marković, "A Secure Mobile Framework for m-Services", The Third International Conference on Internet and Web Applications and Services, ICIW 2008, June 8-13, 2008 - Athens, Greece.

[8] M.Marković, G.Đorđević, "Java based secure mobile web service scenario," INFOTECH 2009, March, 2009, Jahorina, Republic Srpska, Bosnia and Herzegovina.