An Example of Secure Java based Mobile Application

Milan Marković, Goran Đorđević

Abstract — **In this paper, a design and implementation of a JAVA application on mobile phones that securely connect to Web services is described. We considered a Web service scenario where mobile phone user produces a cryptographic signature in the JAVA application using the PKI SIM smart card. Data is encrypted using a crypto MIDlet or Xlet JAVA application installed on mobile phone with CLDC or CDC configurations. The user uses standard WSS signature mechanism (Web Service Security) 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 Java based mobile phone application. The mobile application processes the SOAP response and display the status to the mobile user. The work presented and examples described are carried out within the EU IST FP6 SWEB project (Secure, interoperable cross border m-services contributing towards a trustful European cooperation with the non-EU member Western Balkan countries) where possible mobile government scenarios are implemented. One example of the mobile government online services is particularly emphasized: sending m-residence certificate request and obtaining the m-residence electronic document as a municipality's response.**

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

I. INTRODUCTION

This paper mainly identifies the need for security in mobile communications, such as mentioned in [1]. It mainly presents a secure mobile framework that is based on widely used XML-based standards and technologies such as XML-Signature, XML-Encryption (XML-Security) and Web Services Security (WS-Security). It describes the main entities that participate in the communication, and illustrates its operation with a fully deployed mobile scenario.

Besides security aspects of the XML communication, a possible Federation ID system based on security token service is considered. The security token represents a collection of claims about an entity. Typically, this token is used to authenticate the sender or responder in a message transaction. In this work, SAML (Security Assertion Markup Language) tokens/assertions have a role of security tokens. Communication between JAVA mobile application, or the SOA-Based platform itself, and Security Token Service (STS) server is realized by using WS-Secured SOAP communication.

We also investigated possibility of using XML Key Management (XKMS) protocol [2] in the proposed e/m-government system. The XKMS protocol constitutes a lightweight front end for accessing PKI services, possibly used in the wireless communication where should be of primary benefits. It enables the integration of keys and certificates into mobile applications as well as the implementation of PKI X.509v3 digital certificate registration, revocation and update mechanisms.

The paper is orgnaized as follows. A description of the possible m-Governmental architecture is given in Chapter 2 while the possible m-Governmental scenario is considered in the Chapter 3. Chapter 4 is dedicated to the consideration about secure JAVA mobile application modules. Conclusions are given in Chapter 5.

II. IMPLEMENTATION ASPECTS

The proposed m-government model [1], [3] consists of:

- Mobile users (citizen, companies) who send some Web Services requests to e/m-government platform based on Web Services for a purpose of receiving some governmental documents (e.g. residence certificate, birth or marriage certificates, etc.). These users use secure JAVA mobile Web Service application for such a purpose.
- Fixed/Desktop users connecting to the proposed Web Service governmental platform through some desktop secure Web Service application (could be JAVA-based too). Since everything regarding functionality is similar and almost the same between the two client options, in the remaining part of the text we will consider only the secure JAVA Web Service mobile application and m-government solution model.
- Web Service endpoint implementation on the platform's side that implements a complete set of security features – the same set as implemented in the secure JAVA mobile and the secure desktop applications. Well processed requests with all security features positively verified, the Web Service platform's application proceeds to other application parts of the proposed SOA-Based platform, including the governmental Legacy system for issuing actual governmental certificates requested. In fact, the proposed platform

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).

could change completely the application platform of some governmental orgnaization or could server as the Web Service „add-on“ to the existing Legacy system implementation. In the latter case, the Legacy system will not be touched and only a corresponding Web Service interface should be developped in order to connect the proposed SOA-Based platform and the Legacy governmental system.

- External entities, such as: PKI server with XKMS server as a front end, STS server, UDDI and Time Stamping Authority (TSA). Functions of the proposed external entities are following:
  - **STS server** – is responsible for strong user authentication and authorization based on PKI X.509v3 electronic certificate issued to users. Communication between the STS server and user's JAVA mobile application is SOAP-based and secured by using WS-Security features. After the succesful user authentication and authorization, the STS server issues to the user a SAML token which will be subsequently used for the user authentication and authorization to the Web Service of the proposed m-government platform, or any other Web Service of othe governmental organizations – acting as the Federation ID or Single-Sign-On instrument. The SAML token is signed by the STS server and could consist of the user role for platform's user authentication and authorization and eventually of URL of the requested governmental Web Service, found by contacting the suitable UDDI server.
  - **PKI server** - is responsible for issuing PKI X.509v3 electronic certificates for all users/actors in the proposed m-governmental platform (users, civil servants, administrators, servers, platforms, etc.). Since some certificate processing functions could be too heavy for mobile users, the PKI services are exposed by the XKMS (XML Key Management Service) server which could register users, as well as locate or validate certificates on behalf of the mobile user. This is of particular interests in all processes that request signature verification on mobile user side.
  - **TSA (Time Stamping Authority) server** - is responsible for issuing time stamps for users requests as well as for platform responses (signed m-documents).

The secure JAVA mobile applicationcommunicates with all mentioned external entities, i.e. it has all security functions mentioned implemented, i.e. [4,5]:

- Secure JAVA mobile application sends Request for Security Tokens to the STS server by using WS-Secured (WS-Signature and WS-Encryption) SOAP communication.
- Secure JAVA mobile applications sends digitally signed (XML signature) request for mobile governmental document to the Web Service of the proposed platform by using WS-Encrypted SOAP communication.
- The sent request includes of the SAML token issued and signed by the STS server.
- The request is timestamped by sending a timestamp request and obtaining the corresponding timestamp response (digitally signed by the TSA).
- The secure JAVA mobile application also receives the signed and timestamped m-governmental document from the platform through WS-Encrypted communication and performs all necessary signature verifications and certificate validations (by help of the XKMS server) actions.

In the sequel, we will describe in more details a possible m-governmental scenario

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. POSSIBLE M-RESIDENCE SCENARIO

One example of the mobile government online services is particularly emphasized: sending m-residence certificate request and obtaining the m-residence electronic document as a municipality's response (mRCertificate m-Residence Certificate) (see Figure 1).

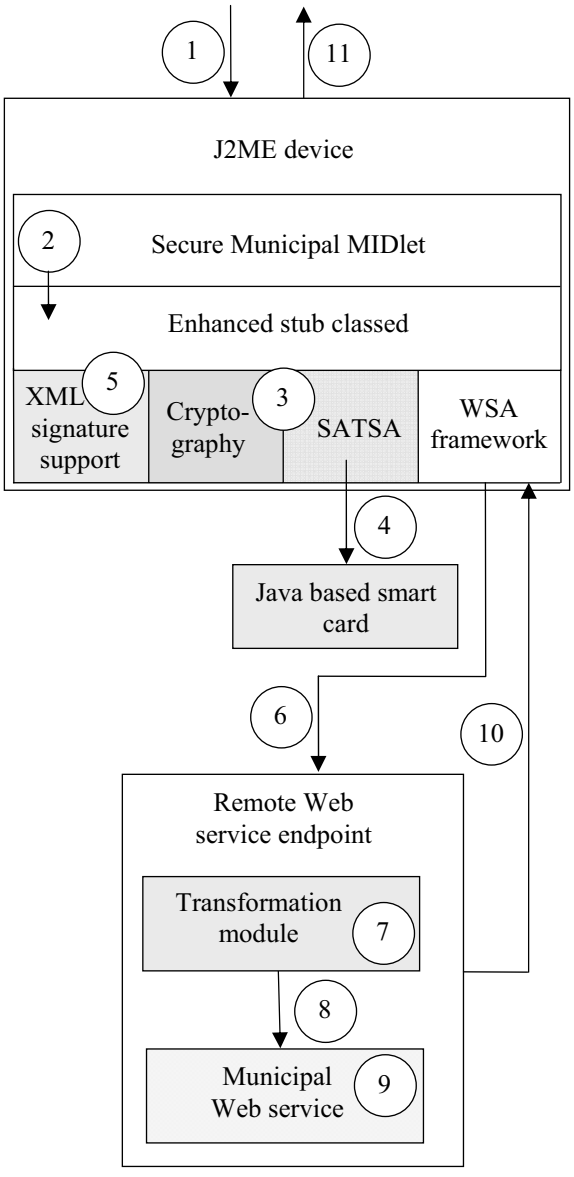

Figure 1: Security components for wireless access to Web services

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 order to communicate with a Web Service endpoint of the municipality A, a J2ME device uses the Web Services APIs (WSA) for J2ME. The WSA uses the idea of stub classes, so other technology components such as cryptography, XML signature and Java based PKI smart card technology have to fit into WSA stub classes [6]. The Java based PKI smart card technology provides a means of securely storing confidential information, private and secret keys. You can use the Java based smart card technology for two basic purposes:

- to securely store a cryptographic key;
- to implement signature calculation algorithm.

Suppose User_A is a Java Card user, who wants to produce a cryptographic signature using the Java Card. He will need to provide a PIN code while accessing his Java Card. The PIN code will work only User_A's Java Card. Therefore, if a hacker steals his Java Card, the hacker will also need to know PIN code for that specific Java Card.

In order to communicate with a Java Card, a J2ME device will need an API called Security and Trust Services API (SATSA). Another important technology for developing secure web service is XML signature. You can use XML signature to wrap a cryptographic signature within an XML message.

A possible secure mobile web service scenario include following steps to WSA security (see Fig. 1):

1. Suppose User_A, a municipal resident, wants to access the municipal document exchange service to request for issuance of residence certificate. User_A's J2ME cell phone has a secure Municipal MIDlet running. So he will invoke the MIDlet.
2. The MIDlet hands over the request to a set of enhanced stub classes that use four technologies: XML signature, Cryptography, SATSA/Java Card, and WSA to author a secure reserve SOAP request that wraps user authentication data.
3. Enhanced stub classes use cryptography and SATSA to fetch all cryptographic support required by the secure SOAP request.
4. SATSA, in turn uses a Java Card application to compute cryptographic signature value over User_A's SOAP request.
5. Next, enhanced stub classes use the XML signature support to author a complete XML signature and wrap the signature in the SOAP request.
6. Enhanced stub classes use WSA framework to send the request over to the remote Web service endpoint implementation.
7. The remote Web service implementation will need to transform the incoming SOAP request before processing. A transformation module hosted in the remote Web service will do the job.
8. The transformation module will hand over the request to the actual Web service implementation.
9. The Web service will perform request processing.
10. The web service will send certification document issuance status in the form of a SOAP response back to the WSA framework.
11. The WSA will process the SOAP response, extract certification status and display the same to the User_A.

The same scenario is used for implementation other m-services like m-Invoice [9].

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.

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.

Locate and validate certificates by using corresponding functions of the XKMS protocol and communicates with the XKMS server via SOAP communication [8].

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.

We elaborated a m-government framework which is based on secure JAVA mobile application, PKI certificates, SOA-based platform, XML-security, WS-Security, SAML, Time Stamping and XKMS.

In order to implement the abovementioned functionalities, the mobile JAVA application communicates with following external entities:

- STS (Security Token Service) server,
- XKMS (Xml Key Management Service) server,
- Time Stamping Authority (TSA) server,
- m-Governmental (SWEB) platform – Interaction Tier Manager of the municipality.

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,

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

[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,BiH.

[9] Journal Article: Cuno, S., Glickman, Y., Hoepner, P., Karantjias, T., Marković, M., Schmidt, M. (2008). The Architecture of an Interoperable and Secure eGovernment Platform Which Provides Mobile Services, Collaboration and the Knowledge Economy: Issues, Applications, Case Studies, *Paul Cunningham and Miriam Cunningham (Eds)*, *IOS Press*, Amsterdam ISBN 978-1-58603-924-0, pp. 278-256