Towards Mobile Government: Verification of Electronic Signatures on Smartphones

Thomas Zefferer, Fabian Golser, and Thomas Lenz

Institute for Applied Information Processing and Communications
Graz University of Technology
Inffeldgasse 16a, 8010 Graz, Austria
thomas.zefferer@iaik.tugraz.at, fabian.golser@student.tugraz.at, thomas.lenz@iaik.tugraz.at

Abstract. Electronic signatures are a crucial concept for transactional e-government services. Beside the secure creation of electronic signatures, the reliable verification of electronically signed documents is of special importance. Various tools, which allow verification of electronic signatures, have been introduced during the past years. However, most of these tools have been tailored to the requirements of classical end-user devices such as desktop computers or laptops and cannot be conveniently used on smartphones. This is problematic, since smartphones and related mobile end-user devices are gradually replacing classical end-user devices. To overcome this issue, we present a signature-verification solution for smartphones in this paper. The presented solution is based on a platform-agnostic architectural design, which can be applied on arbitrary smartphone platforms such as Google Android or Apple iOS. The practical applicability of the proposed solution has been evaluated by means of a concrete implementation. This implementation shows that the presented solution provides convenient means to verify electronically signed documents on smartphones and hence paves the way for the realization of transactional e-government services on mobile end-user devices.

Keywords: Electronic signatures, Mobile Government, Smartphones, Signature verification.

1 Introduction

Electronic signatures are an important cryptographic concept for transactional e-government solutions [6]. Electronic signatures are based on asymmetric cryptographic methods and algorithms such as RSA [1] or ECDSA [2]. These cryptographic algorithms are usually applied together with a public-key infrastructure (PKI), which is used to unambiguously link a signer’s cryptographic key to his or her identity by means of electronic certificates. This way, the cryptographic concepts of electronic signatures and PKIs can be used to reliably assure data integrity and non-repudiation or origin. These properties make electronic signatures especially suitable for the reali-

adfa, p. 1, 2011.

zation of transactional e-government solutions that require a digital alternative to hand-written signatures.

In Europe, the importance of electronic signatures and related concepts has been recognized by legislative bodies of the European Union. In particular, the use of electronic signatures has been defined and regulated in the Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures (hereinafter referred to as EU Signature Directive) [3]. The EU Signature Directive defines different types of electronic signatures. For e-government use cases, especially qualified electronic signatures are of relevance, as they are defined to be legally equivalent to hand-written signatures by the EU Signature Directive. This way, qualified electronic signatures are perfectly suitable for transactional electronic procedures and help to avoid media breaks by rendering the print-out of documents and the application of hand-written signatures unnecessary [6].

During the past years, electronic signatures have become an integral component and key concept of various e-government services and solutions all over the world [11]. This includes solutions for the creation of electronic signatures as well as for the validation of an electronic signature. Most of these solutions have been mainly designed for classical end-user devices such as desktop computers and laptops. However, during the past years, smartphones and tablet computers have emancipated from these classical end-user devices and are nowadays frequently used to access information and services. Governments and public administrations are required to face this recent development and to provide e-government services and applications also for mobile end-user devices [16].

Appropriate concepts and solutions to implement electronic signature based procedures on mobile end-user devices have already been introduced and discussed in literature. For instance, a smartphone app for Google Android[1] that allows users to electronically sign arbitrary PDF documents on their mobile devices has been introduced in [4]. Together with other similar solutions, this work has shown that electronic signature based solutions on smartphones are basically feasible. Interestingly, most of the proposed solutions focus on the creation of electronic signatures on smartphones, but do not provide appropriate means to verify electronic signatures on mobile end-user devices. A smartphone user, who received an electronically signed document, has therefore no opportunity to conveniently and reliably verify the obtained signature on his or her smartphone. Due to the lack of appropriate signature-verification tools on smartphones, smartphone users are not able to employ the key advantage of electronic signatures compared to hand-written signatures, i.e. their unambiguous verifiability.

To close this gap, we present a signature-verification solution for smartphones in this paper. This solution is tailored to the special requirements and properties of current smartphones and related mobile end-user devices. Considering the current heterogeneous ecosystem of different smartphone platforms and mobile operating systems, we first introduce a platform-agnostic architectural design for the proposed solution.

[1] http://www.android.com/

We evaluate the applicability and practicability of this platform-agnostic architectural design by means of a concrete implementation for the Google Android platform and show that our solution is basically ready for productive operation.

2 Related Work

The importance of electronic signatures for e-government is evident and has been discussed extensively in scientific work such as [6]. Their relevance becomes also evident when analyzing e-government infrastructures and solutions of different countries [5][14]. In most cases, electronic signatures are a key concept used to reliably authenticate users, to protect the integrity of data in online processes, and to obtain written consent from users in electronic procedures.

Besides e-government, electronic signatures can actually also be useful in other fields of application from the corporate and the private sector. For instance, experience has shown that companies frequently make use of electronic signatures e.g. to sign invoices that are electronically sent to customers. Similarly, electronic signatures are increasingly used also by private persons e.g. to sign contracts in electronic form. In this context, especially PDF signatures have recently gained importance. Beside the well-known PDF signature format introduced by the company Adobe[2], solutions that allow private, public, and corporate users to create qualified electronic signatures on PDF documents are available in several countries [6].

With the growing importance and increasing spread of electronic signature based solutions, also the need for and the importance of appropriate signature-verification tools has increased. Such tools are crucial as they allow receivers of electronically signed documents to verify the validity of the obtained document's signature. During the past years, different verification tools for electronic signatures have been introduced. For instance, a publicly available Web based signature-verification tool called WebNotarius[3] has been provided by Unizeto Technologies SA[4]. WebNotarius supports the verification of different document and signature formats including PCKS#7 [7], CMS [8], S/MIME [9], and XMLDSig [10]. The German company signagate[5] provides a similar Web based signature-verification tool. In contrast to WebNotarius, this tool is however limited to the PDF file format. Web based signature-verification tools for the verification of signed XML and PDF files are also provided by the two companies ascertia[6] and SecuredSigning[7].

Another powerful signature-verification tool has been introduced by Lenz et al. [11]. Similar to the above-mentioned solutions, also the tool proposed by Lenz et al. follows a Web based approach and allows for the verification of different document and signature formats. However, access to this tool's functionality is not limited to the

[2] http://www.adobe.com/products/acrobat/electronic-signatures-e-signatures.html

[3] http://www.webnotarius.eu

[4] http://www.unizeto.pl/

[5] http://www.signagate.de/

[6] http://www.ascertia.com/

[7] http://www.securedsigning.com/

Web interface. Additionally, the tool features a web-service interface that can be used by external entities to communicate with the tool and to access its functionality programmatically. This way, external entities such as Web applications can send signed documents, which should be verified, to the signature-verification tool and retrieve results of the conducted verification process.

This brief survey on existing signature-verification tools shows that most existing solutions currently follow a Web based approach and that these tools are mainly tailored to the needs of classical end-user devices such as desktop PCs and laptops. These solutions allow users to upload signed documents to a central Web application through a Web based interface and display verification results in the used Web browser. Even though Web based interfaces can also be accessed from smartphones, this is actually less practicable due to the limited input and output capabilities of smartphones. For smartphones, a dedicated app that allows the verification of electronic signatures would be beneficial, since smartphones apps can be tailored to the special input and output capabilities of smartphones. We propose a general architectural design for signature-verification tools for smartphones in the next section.

3 Architectural Design

The verification of electronic signatures is a complex task that involves the application of cryptographic methods to technically verify the validity of a given signature and the communication with external PKI entities to determine the validity of used signing certificates. Even though the computational power of smartphones is constantly increasing, it is usually reasonable to outsource complex operations to server components in order to speed-up processes and to save smartphone resources at the same time.

The signature-verification tool for smartphones that we present in this paper follows this approach and relies on functionality provided by a central server component. This is illustrated in Figure 1. A smartphone app provides the user means to verify arbitrary signed documents. Basically, the app allows the user to choose documents, which should be verified, and displays results of the verification process. However, the app does not implement the signature-verification process itself, but accesses a central server component for this purpose. This way, the smartphone app can be kept lightweight. Furthermore, additional functionality can be added to the signature-verification process easily without requiring users to update their local smartphone apps.

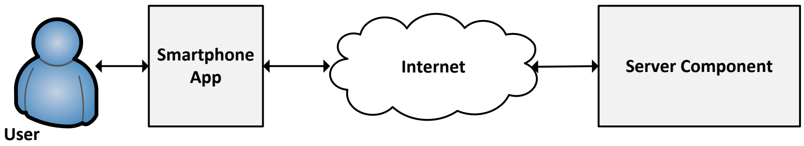


Fig. 1. General architecture of the proposed signature-verification solution for smartphones

According to the general architecture shown in Figure 1, the proposed solution consists of a server component and a smartphone app. We propose and discuss the architectural designs of these two core components in the following subsections.

3.1 Server Component

While the smartphone app is kept lightweight, the server component implements most functionality required to verify electronic signatures and electronically signed documents. This includes the determination of the format of the provided document, the verification of the provided document's signature(s), and the verification of the used signing certificate's validity. From these requirements, the architectural design shown in Figure 2 can be derived.

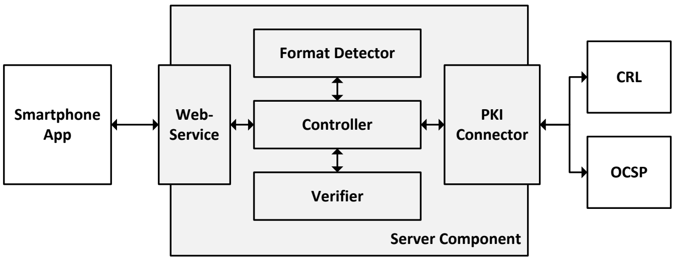


Fig. 2. Architectural design of the server component

In general, the server component consists of the following core components.

- *Controller:* The Controller represents the central component of the server component. It controls the entire process flow that starts with the reception of a signed file to be verified and finally leads to the verification of the provided document.
- *Web Service*: The Web Service represents the interface to the smartphone app. The smartphone app can use the provided Web Service to hand over documents to be verified and to retrieve verification results. Reliance on a Web service based interface guarantees that the functionality of the server component can be accessed by arbitrary external components.
- *Format Detector*: The Format Detector implements the first step of the verification process. Any document or file to be verified is sent to the Format Detector first, in order to determine the provided document's format. Based on the result of the format-detection process, the appropriate verification module is selected by the Controller.
- *Verifier*: The Verifier checks the cryptographic validity of the provided document's signature(s). The Verifier implements appropriate verification modules for each supported document format. The correct module is selected by taking into account the result of the format-detection process. The selected verification module verifies the cryptographic validity of the provided document's signature.
- *PKI Connector*: The PKI Connector verifies the validity of the used signing certificate by accessing appropriate certificate revocation lists (CRL) or external entities

implementing the online certificate status protocol (OCSP). This way, the PKI Connector represents the interface to external public-key infrastructures.

Following the architectural design outlined in Figure 2, the server component encapsulates all functionality that is required to verify electronically signed documents. Access to this functionality is provided through a Web-service interface. This way, the server component can be used by arbitrary external entities including smartphone apps. The architectural design of a smartphone app that uses signature-verification functionality provided through this Web-service interface is presented in the following subsection.

3.2 Smartphone App

As shown in Figure 1, a smartphone app represents the second core component of the proposed signature-verification solution for smartphones. The smartphone app basically takes over two core tasks. First, it implements a graphical user interface (GUI). Through this GUI, the user can select signed files, which should be verified, and define various parameters related to the verification process. Additionally, the GUI is used to display verification results. Second, the smartphone app communicates with the server component through the provided Web-service interface in order to transmit signed documents and to retrieve the corresponding verification results.

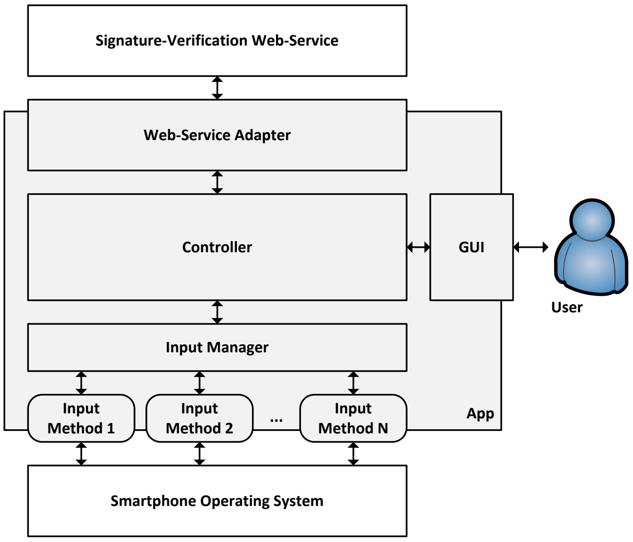


Fig. 3. Architectural design of the smartphone app

Based on these two core tasks, we have developed an appropriate architecture for the smartphone app. This architecture has been designed such that it is applicable on arbitrary smartphone platforms and not restricted to a certain platform such as Apple iOS[8]

[8] http://www.apple.com/ios/

or Google Android[9]. The resulting platform-agnostic architecture is shown in Figure 3.

Similar to the server component, also the smartphone app is composed of several core components that implement the app's functionality. These components are introduced in the following in more detail.

- *Controller:* The Controller represents the central element of the smartphone app. This component implements the app's business logic and controls other components and building blocks of the app.
- *Graphical User Interface (GUI):* The GUI represents the interface to the user. It allows the user to select arbitrary signed files for verification and displays obtained verification results.
- *Input Manager:* The Input Manager represents the main interface to the underlying smartphone platform and its mobile operating system. The Input Manager is used to retrieve files to be verified from the operating system. Different smartphone platforms provide apps different methods to retrieve files from the operating system. For instance, direct access to the smartphone's file system is available on Google Android devices, but forbidden on smartphones running the Apple iOS operating system. To cope with this situation, the Input Manager follows a modular approach. This is also illustrated in Figure 3. Depending on the particular smartphone platform, the Input Manager supports different input methods, which allow the retrieval of signed files from the underlying operating system.
- *Web-Service Adapter:* The Web-Service Adapter implements the communication to the Web-service interface provided by the server based signature-verification tool. This way, the Web-Service Adapter basically represents the gateway to the signature-verification functionality. The app's Controller component uses the Web-Service Adapter to send signed files selected by the user through the GUI and retrieved from the smartphone's operating system by the Input Manager to the server based signature-verification tool and to retrieve the results of the signature-verification process.

The app's architecture and its core components have been designed in a platform-agnostic way. Hence, this architecture can be used for appropriate signature-verification solutions for all current smartphone platforms. We have assessed the applicability and practicability of the proposed architecture by means of a concrete implementation for the Google Android platform. Details of this implementation are provided in the next section.

4 Evaluation

In order to assess and evaluate the practical applicability of the proposed architecture, we have developed a signature-verification solution for Google Android smartphones according to the proposed architectural design. The Google Android platform has

[9] http://www.android.com

been chosen, as this platform is currently the world market leader and can be assumed to remain one of the most important platforms in the future [15].

According to the architectural design discussed in Section 3, our implementation consists of a server component and a smartphone app. Further implementation details of these two components are presented in the following subsections.

4.1 Server Component

For the server component, our implementation relies on the Web based signature-verification tool introduced by Lenz et al. in [11]. As this tool already provides an appropriate Web-service interface, it is perfectly suitable to act as server component for our implementation.

Details of the implementation of the server component are provided in [11]. For our smartphone based signature-verification solution, the design of the server component's Web-service interface is of special importance. The provided Web service uses the SOAP protocol [12] to transmit data in the form of XML based SOAP messages between the server component and external entities. HTTP [13] is used on the underlying layer and acts as carrier for exchanged SOAP messages.

SOAP requests being sent to the server component need to comply with a well-defined XML schema that is shown below. While the element *Document* is mandatory and contains the signed document to be verified, the element *FileID* is optional and can be used to identify the signed file.

```
<xsd:element name="VerifyDocumentRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="Document"
                      type="xsd:base64Binary"/>
         <xsd:element name="FileID" type="xsd:token"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Upon reception of a schema compliant request, the server component starts the signature-verification process and verifies all electronic signatures of the received document. Results of the verification process are collected and assembled into an XML based verification report. This verification report is finally electronically signed by the server component in order to guarantee its authenticity and integrity. The signed verification report is embedded into a SOAP message that complies with the XML schema shown below. This SOAP message represents the response that is finally returned to the calling smartphone app.

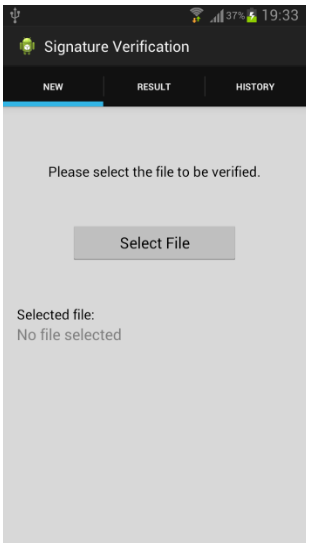
```
<xsd:element name="VerifyDocumentResponse">
   <xsd:complexType>
      <xsd:sequence>
```

```
         <xsd:element name="VerificationReport"
                      type="tns:VerificationReportType"/>
         <xsd:element name="Signature"
                      type="dsig:SignatureType"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

The provided Web-service interface allows external entities to easily access the server component’s signature-verification functionality. For more details on the implemented signature-verification process itself, the interested reader is referred to [11].

4.2 Smartphone App

The smartphone app represents the second basic building block of the architectural design proposed in Section 3. According to the proposed design shown in Figure 3, the smartphone app needs to implement an appropriate GUI as well as appropriate means to select documents to be verified. Finally, the smartphone app also needs to implement means to communicate with the Web service provided by the server component.



Please select the file to be verified.

Selected file:
No file selected

Fig. 4. File selection

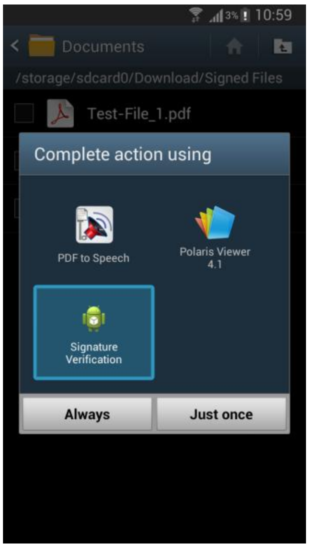
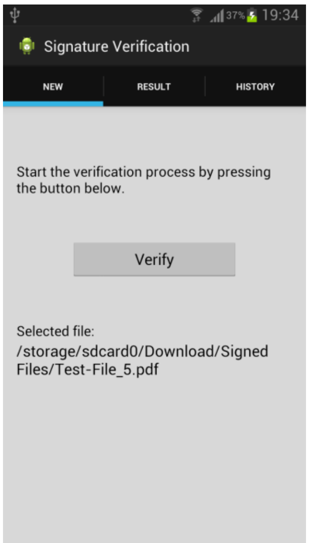


Fig. 5. File handler

Our implementation relies on the Google Android platform to realize a smartphone application that is able to meet these requirements. Considering the special capabilities and specifics of this smartphone platform, our implementation of the smartphone app allows users to select documents for verification in two different ways. First, user can use the GUI shown in Figure 4 to open a file-selection dialogue and to choose the document to be verified from the smartphone's file system. Second, the smartphone app also registers a file handler for supported file formats in the operating system. This way, the smartphone app can be easily selected from a dialogue that appears when the user attempts to open one of the supported file types. This is illustrated in Figure 5.



Start the verification process by pressing the button below.

Selected file:
/storage/sdcard0/Download/Signed Files/Test-File_5.pdf

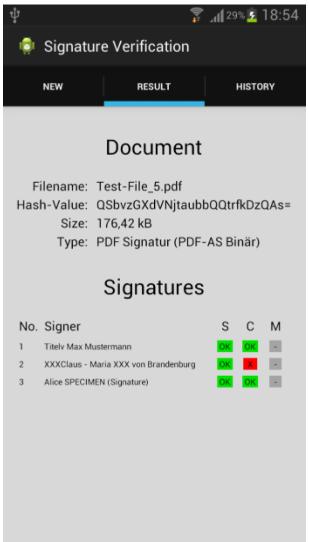
Fig. 6. Start of signature-verification process

When the document to be verified has been selected using one of the two supported methods, the signature-verification process can be started using the Send File button as shown in Figure 6.

After touching this button, the selected document is sent to the server component's Web-service interface using a schema compliant SOAP request. Upon completion of the signature-verification process, the verification result is returned to the smartphone application. The smartphone application evaluates the obtained verification results and displays them to the user as shown in Figure 7.

For each verified document, the smartphone app displays related information such as the filename, the hash value of the document, its size, and the type of the detected signature. Furthermore, verification results of all signatures that have been found in the document are presented to the user. As shown in Figure 7, for each detected signature, the verification result of the signature (S), the signing certificate (C), and the manifest (M) are shown. This way, the validity of the found signatures becomes apparent immediately.

As an additional feature, the implemented smartphone app provides the user with a history of recently verified documents. This is illustrated in Figure 8. Users can review their recent verification results by clicking on the respective file. Entries in this history list can be deleted using the recycle-bin icon located next to the file name.



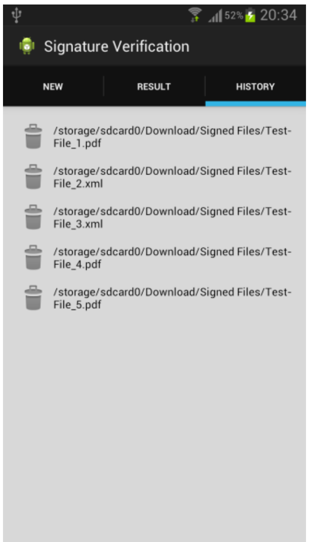
Document

Filename: Test-File_5.pdf
Hash-Value: QSbvzGXdVNjtaubbQQtrfkDzQAs=
Size: 176,42 kB
Type: PDF Signatur (PDF-AS Binär)

Signatures

No.	Signer	S	C	M
1	Titelv Max Mustermann	OK	OK	-
2	XXXClaus - Maria XXX von Brandenburg	OK	X	-
3	Alice SPECIMEN (Signature)	OK	OK	-

Fig. 7. Result illustration



/storage/sdcard0/Download/Signed Files/Test-File_1.pdf
/storage/sdcard0/Download/Signed Files/Test-File_2.xml
/storage/sdcard0/Download/Signed Files/Test-File_3.xml
/storage/sdcard0/Download/Signed Files/Test-File_4.pdf
/storage/sdcard0/Download/Signed Files/Test-File_5.pdf

Fig. 8. Result history

5 Conclusions

In this paper we have proposed a signature-verification solution for smartphones. This solution tackles the problem that most currently available signature-verification tools have been designed for classical end-user devices such as desktop PCs or laptops and hence lack an appropriate level of usability on smartphones and related mobile devices. The architecture of the proposed signature-verification solution has been designed in a platform-agnostic way in order to assure that the solution is applicable on arbitrary smartphone platforms.

The general applicability and practicability of the proposed solution and of the presented architectural design has been successfully evaluated by means of a concrete implementation for the Google Android platform. This implementation shows that the proposed solution is capable to provide easy and usable means to verify electronically signed documents on smartphones. Although the implemented Android application is already fully functional, it is still in a prototypical state. We are currently working on several improvements in order to prepare our solution for publication and distribution in the Google Play store. Similar implementations of the proposed solution on other smartphone platforms such as Apple iOS or BlackBerry are also regarded as future work.

Summarizing, the solution presented in this paper provides users the opportunity to conveniently verify electronically signed documents on smartphones and related mobile end-user devices. This way, the presented solution represents a significant step towards the mobile processing of transactional mobile procedures and helps to pave the way for future mobile government solutions.

References

1. Jonsson, J. and Kaliski, B.: Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1, RFC 3447, RFC Editor, United States, 2003.
2. National Institute of Standards and Technology (NIST): FIPS-186-2: Digital Signature Standard (DSS), January 2000.
3. The European Parliament and the Council of the European Union: Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures, 2000, [Online]. Available: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2000:013:0012:0020:EN:PDF
4. Zefferer, T., Tauber, A., Zwattendorfer, B. and Stranacher K.: Qualified PDF signatures on mobile phones, Electronic Government and Electronic Participation - Joint Proceedings of Ongoing Research and Projects of IFIP EGOV and IFIP ePart, 2012.
5. Leitold H., Hollosi A., P. R.: Security Architecture of the Austrian Citizen Card Concept, In Proceedings of 18th Annual Computer Security Applications Conference (ACSAC’2002), Las Vegas, 9-13 December 2002. pp. 391-400, IEEE Computer Society, ISBN 0-7695-1828-1, ISSN 1063-9527., pages 391–400, 2002.
6. Leitold, H., Posch, R., and Rössler, T.: Mediabreak resistant eSignatures in eGovernment-An Austrian experience. In Dimitris Gritzalis, J. L., editor, Emerging Challenges for Security, Privacy, and Trust - 24th IFIP SEC, volume IFIP AICT 297 of IFIP Advances in Information and Communication Technologies, pages 109 – 118. Springer, 2009.
7. RSA Laboratories: PKCS#7: Cryptographic Message Syntax Standard, RSA Laboratories, 1993.
8. Housley, R.: Cryptographic Message Syntax (CMS), RFC 5652 RFC Editor, United States, 2009, [Online]. Available: http://www.ietf.org/rfc/rfc5652.txt
9. Ramsdell, B. and Turner, S.: Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification, RFC 5751, RFC Editor, United States, 2010.
10. World Wide Web Consortium: XML Signature Syntax and Processing (Second Edition), W3C, 2008b, [Online]. Available: http://www.w3.org/TR/xmldsig-core/
11. Lenz T., Stranacher K., Zefferer T.: Towards a Modular Architecture for Adaptable Signature-Verification Tools, 9th International Conference on Web Information Systems and Technologies, 2013.
12. Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J.-J. and Nielsen, H. F.: Soap version 1.2 part 1: Messaging framework, W3C, 2007.
13. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T.: Hypertext transfer protocol – http/1.1, RFC 2616, RFC Editor, United States, 1999, [Online]. Available: http://www.ietf.org/rfc/rfc2616.txt.
14. Posch, K.C., Posch, R., Tauber, A., Zefferer, T., and Zwattendorfer, B.: Secure and Privacy-preserving eGovernment – Best Practice Austria. In: Rainbow of Computer Science, Springer, 2011.
15. Jones, C.: Android Solidifies Smartphone Market Share. Forbes, 2013, [Online]. Available: http://www.forbes.com/sites/chuckjones/2013/02/13/android-solidifies-smartphone-market-share/.
16. Zefferer, T., Teufl, P.: Opportunities and Forthcoming Challenges of Smartphone-Based m-Government Services. In: European Journal of e-Practice, Megatrends in E-Government, 2011. [Online]. Available: http://www.epractice.eu/files/European%20Journal%20epractice%20Volume%2013%20-%2004%20-%20Megatrends%20in%20eGovernment.pdf