Thomas Zefferer, Arne Tauber, Bernd Zwattendorfer, Klaus Stranacher - "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, pp. 115-123

QUALIFIED PDF SIGNATURES ON MOBILE PHONES

Thomas Zefferer[1], Arne Tauber[1], Bernd Zwattendorfer[1], Klaus Stranacher[1]

PDF is one of the predominating document formats in e-Government processes. The electronic signing of PDF documents is crucial to ensure the security of e-Government processes. In Austria, a proprietary PDF signature format called PDF-AS has been introduced in order to meet given legal frameworks on national and European level. Fulfilling all requirements of qualified electronic signatures according to the EU Signature Directive, PDF-AS signatures are legally equivalent to handwritten signatures.

The growing popularity of smartphones raises the demand for mobile e-Government services. This includes the demand for mobile PDF signing capabilities. To satisfy this demand, we have developed a mobile PDF signing solution for current smartphone platforms. Following the PDF-AS signature format, our solution allows for the creation of PDF signatures that are legally equivalent to handwritten signatures. In this paper we discuss the architectural design of our solution and demonstrate its applicability by means of a prototypical implementation based on the popular Google Android platform.

1. Introduction

Electronic signatures are an important tool and key component of numerous e-Government solutions. Assuring data integrity and non-repudiation of origin, electronic signatures are the tool of choice to meet security requirements of e-Government applications. Similar to other European countries, also Austrian e-Government solutions heavily rely on electronic signatures. Within e-Government processes, Austrian citizens use national eID tokens such as personalized smart cards or mobile phones to create qualified electronic signatures according to the EU Signature Directive [1].

Most key concepts of the Austrian e-Government framework that has been discussed in [2] rely on XML based signature standards such as XMLDsig [3]. Besides XML, also PDF is a common document format in proceedings, e.g. to deliver official notifications. Hence, Austria has additionally introduced a proprietary PDF based signature format for national e-Government solutions[2]. This signature format is called *PDF-AS* and has been discussed in detail by Leitold et al. in [4]. PDF-AS relies on users' eID tokens as *Secure Signature Creation Device (SSCD)*. This way, PDF-AS signatures comply with the requirements of qualified electronic signatures and are legally equivalent to handwritten signatures.

During the past few years, mobile solutions have significantly gained importance and have become an emergent topic also in e-Government. The importance of mobile e-Government (m-Government) solutions has been discussed by Zefferer et al. in [5]. The authors conclude

[1] Institute for Applied Information Processing and Communications – Graz University of Technology, 8010 Graz, Inffeldgasse 16a, {thomas.zefferer|arne.tauber|bernd.zwattendorfer|klaus.stranacher}@iaik.tugraz.at

[2] Although reliance on existing PDF signature standards would have been beneficial, these standards have not been able to meet given legal requirements.

that smartphones seem to be especially suited to meet given requirements of mobile e-Government solutions. Unfortunately, due to the limited input and output capabilities of smartphones (small screen size, touch-based input, etc.) existing e-Government solutions are often difficult to use on smartphones. This applies also to existing solutions that allow creation of PDF-AS conformant signatures. None of them has been optimized for use on smartphones.

Given the importance of PDF-AS signatures in Austria, development of a user-friendly smartphone application that allows mobile users to sign arbitrary PDF files seems reasonable. Unfortunately, the requirement for integration of national eID tokens into the signature-creation process renders this task difficult. In this paper we propose a solution to this problem. Our solution allows mobile users to electronically sign arbitrary PDF documents on smartphones. The created signatures follow the PDF-AS specifications and hence comply with the requirements of qualified electronic signatures.

The remainder of the paper is organized as follows. In Section 2 we provide background information on the legal and technical framework our solution is based on. Section 3 describes the architectural design of our mobile PDF signing solution. Subsequently, implementation details are provided in Section 4. Finally, we give an outlook and draw conclusions summarizing the main facts and open issues.

2. Legal and Technical Background

Electronic signatures play an important role in most e-Government solutions as they represent the pendant to handwritten signatures in the digital world. The legal equivalence of qualified electronic signatures to handwritten signatures is defined by the Directive 1999/93/EC of the European Parliament and of the Council on a Community framework for electronic signatures [1]. To ensure a common legal basis on European level, all Member States of the European Union were required to transpose this directive into national law by July 19, 2001.

In Austria, the Signature Act [6] defines the legal basis for the use of electronic signatures in Austria and fulfils all requirements of the EU Signature Directive. Together with the Austrian e-Government Act [7], the Signature Act represents the legal basis for e-Government solutions in Austria. All technical implementations that require qualified identification and authentication, e.g. in terms of qualified electronic signatures, need to comply with these national laws and legal requirements.

The Austrian e-Government Act defines the so called *Citizen Card* as key concept and core component of Austrian e-Government applications. The Citizen Card allows citizens to authenticate at remote e-Government services and to create qualified electronic signatures that are legally equivalent to handwritten signatures according to the EU Signature Directive. The Citizen Card is a technology-neutral and token-independent concept. Currently, Austrian citizens can use personalized smart cards or their personal mobile phones as Citizen Card in order to authenticate at remote services or to create electronic signatures. The mobile phone based solution is called *Mobile Phone Signature* and has been introduced and discussed by Orthacker et al. in [8].

The Mobile Phone Signature is implemented as a server based service. A hardware security module (HSM) being connected to this service acts as SSCD. The HSM securely stores the private signature keys of all citizens. Access to these keys is protected by means of a strong authentication scheme relying on the factors knowledge and possession. The factor

knowledge is covered by a secure password that has to be entered by users through a Web form in order to start the signature creation process. The factor possession is covered by the user's mobile phone. Possession of this token is verified by sending a one-time password – a transaction number (TAN) – to the mobile phone via SMS. Users have to enter this TAN through the Web form provided by the Mobile Phone Signature. Only if both password and TAN are provided correctly, access to the user's private key is granted and the signature creation process can be completed successfully.

E-Government applications can access Citizen Card functionality through a standardized interface irrespective of the underlying technology (e.g. smart card, mobile phone, etc.). This XML based interface is called *Security Layer* and has been discussed by Leitold et al. in [9]. Through this interface, e-Government applications can request citizens to create XML based signatures using their personal Citizen Card.

However, in many e-Government scenarios, PDF is currently the predominating document format. Hence, appropriate means to electronically sign PDF files are mandatory. Unfortunately, it turned out that existing PDF signature formats are often not able to comply with given legal requirements. Therefore, a proprietary PDF signature format called PDF-AS has been defined in Austria [10]. According to the PDF-AS specifications, a visual signature block is added to the signed PDF document. This signature block represents the pendant to a handwritten signature and contains information on the signatory, the signature date, and some technical information related to the signing process.

A Web application that implements the PDF-AS specification is available in Austria as open source module [11]. The *PDF-AS Web-Application* allows users to upload and sign arbitrary PDF files. Additionally, the functionality of the PDF-AS Web-Application can also be accessed by means of a servlet based interface. Through this interface, third party applications can hand over files to be signed and receive successfully signed PDF documents.

The PDF-AS Web-Application is perfectly suitable for Web based e-Government applications. Unfortunately, using this Web application on smartphones raises several challenges. Of course, the PDF-AS Web-Application can theoretically be used by smartphones users to electronically sign PDF documents. However, limited input and output capabilities of smartphones often complicate the use of Web based interfaces.

To cope with this situation, we have developed a smartphone app that allows smartphone users to sign arbitrary PDF documents using qualified electronic signatures. Our solution relies on both the Austrian Mobile Phone Signature and the PDF-AS Web-Application. The architectural design of our solution is discussed in the following section.

3. Architectural Design

The proposed solution allows Austrian citizens to sign arbitrary PDF documents with their smartphones. The created signatures follow the PDF-AS specifications and hence represent qualified electronic signatures. In the following, the basic building blocks and the general process flow of our solution is described in more detail.

3.1 Basic Building Blocks

From the user's perspective, the entire signature creation process is carried out on the smartphone. To improve usability, users have to interact with a smartphone app only and do

not need to directly communicate with external components. Behind the scenes, several distributed components are actually involved in the PDF signature creation process.

Figure 1 illustrates the general architecture of our solution. Basically, three components are involved in the signature creation process. The *Smartphone App* implements the user interface and communicates with involved distributed components. Computationally intensive operations are implemented by a server based *PDF Signature Service*. Finally, the *Austrian Mobile Phone Signature* acts as secure signature creation device. In the following, these three components and their interactions are described in more detail.

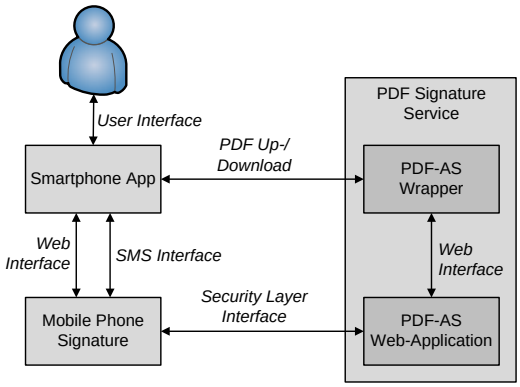

Figure 1: The architectural design of our solution is based on three core components.

3.1.1 Smartphone App

The Smartphone App is the key component of our solution. It implements a user-friendly interface that allows users to start signature creation processes. Selected PDF files that have been passed to the Smartphone App are uploaded to the central PDF Signature Service for further processing. The Smartphone App implements an appropriate interface to communicate with the PDF Signature Service.

The Smartphone App also implements an interface to the Austrian Mobile Phone Signature. In order to authorize the signature creation process, user credentials (phone number and secret password) as well as the mobile TAN need to be supplied by the user. Hence, the Smartphone App must be able to receive the mobile TAN that is delivered by SMS and to provide the Mobile Phone Signature with the required credentials.

3.1.2 PDF Signature Service

This component receives PDF documents from the Smartphone App and prepares them for the PDF signature creation process. The PDF Signature Service consists of two internal components, the *PDF-AS Web-Application* and the *PDF-AS Wrapper*.

The PDF-AS Web-Application is used to add electronic signatures to given PDF files according to the PDF-AS specifications. While all required processing of the PDF file is carried out by the PDF-AS Web-Application, the signature itself is created by the Austrian Mobile Phone Signature acting as the user's national eID. Hence, the data to be signed has to be sent to the Mobile Phone Signature. The data exchange between the PDF-AS Web-Application and the Mobile Phone Signature relies on the standardized Security Layer interface that has been discussed in Section 2.

After completion of the signature creation process, the Mobile Phone Signature returns the computed signature value to the PDF-AS Web-Application. The PDF-AS Web-Application adds a signature block to the PDF file using the obtained signature value. The signed PDF file is finally returned to the Smartphone App.

Although the PDF-AS Web-Application provides several possibilities for its invocation, invocation out of a smartphone app is still no trivial task. To ease the communication between the Smartphone App and the PDF-AS Web-Application, the PDF-AS Wrapper acts as intermediary between these two components.

On the one hand, the PDF-AS Wrapper provides an interface to the Smartphone App that allows for an easy exchange of PDF files. On the other hand, the PDF-AS Wrapper implements the PDF-AS Web-Application's Servlet based interface to hand over obtained PDF files to be signed and to receive successfully signed PDF documents.

3.1.3 Austrian Mobile Phone Signature

The Mobile Phone Signature computes electronic signatures of the data provided by the PDF-AS Web-Application through the Security Layer interface. All signatures are created with the help of the user's personal signing key that is securely kept inside the Mobile Phone Signature's HSM.

To authorize the usage of this key, the user has to prove knowledge of a secret password and of a TAN that is sent to the user via SMS. All these data has to be entered by the user through a Web form provided by the Mobile Phone Signature. The Smartphone-App communicates with the Mobile Phone Signature through this Web form to provide the required credentials.

3.2 General Process Flow

The general process flow of our solution consists of the following steps. First, the user passes a PDF file to the Smartphone App. After that, the PDF file to be signed is uploaded to the PDF-AS Wrapper. The PDF-AS Wrapper receives the PDF file and hands it over to the PDF-AS Web-Application by invoking its Servlet based interface.

The PDF-AS Web-Application starts the PDF signing process by assembling an appropriate Security Layer conformant signature request. This request is sent to the Mobile Phone Signature. To authorize the signature creation process, the user's mobile phone number and secret password have to be provided first. To simplify this procedure, the mobile phone number can be automatically extracted by the Smartphone App from the citizen's mobile phone. In this case, only provision of the password by the user is required.

Both mobile phone number and password are transmitted to the Mobile Phone Signature for verification. If verification succeeds, a one-time password (TAN) is sent to the user's mobile phone via SMS. This TAN has to be retransmitted to the Mobile Phone Signature. The Smartphone App automatically extracts the TAN from the received SMS and transmits it to the Mobile Phone Signature. In this case, the user does not even recognize the reception of an SMS from the Mobile Phone Signature. This decreases required user interactions and increases usability. Users just need to enter their signature password and signature creation is seamlessly started.

After provision of the TAN to the Mobile Phone Signature, the signing process is initiated. The result of the signing process is returned to the PDF-AS Web-Application, which

incorporates the generated signature into the PDF document. Subsequently, the signed PDF document is transferred back to the user's smartphone application via the PDF-AS Wrapper. Finally, the user can view and inspect the signed PDF document.

4. Implementation

We have evaluated the practical applicability of our solution by means of a prototypical implementation. This implementation relies on the two components PDF-AS Web-Application and Mobile Phone Signature that have been introduced in Section 2. Additionally, we have implemented a PDF-AS Wrapper and an appropriate Smartphone-App according to the above mentioned architectural design. The implementations of these two components are described in the following.

4.1 PDF-AS Wrapper Implementation

The PDF-AS Wrapper component acts as intermediary between the Smartphone-App and the PDF-AS Web-Application. We have implemented the PDF-AS Wrapper by means of a Web application. Its functionality has been implemented using the Java Servlet API and the Apache Struts framework. The PDF-AS Wrapper can be run in a single Servlet container (e.g. Apache Tomcat) together with the PDF-AS Web-Application.

The implementation of the PDF-AS Wrapper has been kept as minimalistic as possible. As this component is completely transparent to the user, it does not feature any user interface. Its only two interfaces are intended to exchange PDF documents with the Smartphone-App on the one, and with the PDF-AS Web-Application on the other side.

4.2 Smartphone-App Implementation

In contrast to the PDF-AS Wrapper, the developed implementation of the Smartphone-App is substantially more complex. Relevant implementation details are provided in the following.

In general, our solution can be realized on all current smartphone platforms. As a first step, we have implemented our solution for the Android 2.2+ platform. To meet the stated requirements and to realize an intuitive user experience, the developed Android-App implementation consists of different components as illustrated in Figure 2.

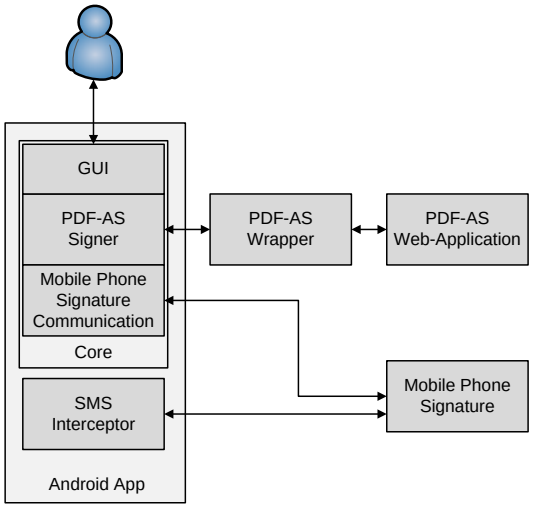

Figure 2: The developed Android app consists of several internal components.

In general, the developed Android app consists of a *Core* component and an *SMS Interceptor*. The Core component is the core module handling the interaction with the user via the graphical user interface (GUI) and the communications with the external components PDF-AS Wrapper and the Mobile Phone Signature.

Upon installation, the app is registered as handler for PDF documents. This means that if PDF documents are accessed from other applications, e.g. an e-mail client or a file explorer, the user can choose to open the PDF document with the developed app. If multiple applications on the phone are registered as PDF handlers (e.g. the Adobe Reader app), the Android OS shows a dialog to select the desired handler. In our case the app with the display name "PDF HandySignature"[3] has to be selected.

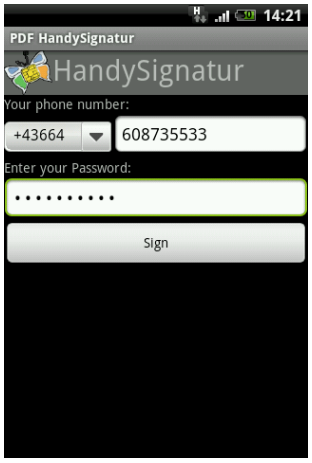

Figure 3: The GUI is used to collect required user credentials.

Upon selection of our app, its *GUI* component is opened. The implemented user interface is shown in Figure 3. This is the first and only step of the entire signature creation process that requires user interaction. The shown GUI provides the user various form fields to enter the mobile phone number and the secret password of the Mobile Phone Signature. If the mobile network operator allows reading out the mobile phone number, the app automatically fills out the first form field. For security reasons, the user's secret password must not be saved and has to be entered each time the app is used.

In a next step the app's *PDF-AS Signer* component is called, which submits the PDF document to the remote PDF-AS Wrapper. As soon as the PDF document has been uploaded, the app's *Mobile Phone Signature Communication* component is called, which initiates communication with the external Mobile Phone Signature component. To sign the PDF document, the Mobile Phone Signature fetches the signature request from the PDF-AS Web-Application and requests the user's secret password from the app's Mobile Phone Signature Communication component.

Communication between the user and the Mobile Phone Signature is usually carried out in a Web browser. For a maximized user experience, the whole HTTP(s) communication with the Mobile Phone Signature is automatically handled by the app's Mobile Phone Signature Communication component using the Android HTTP client API[4]. This means that no user

[3] Note that *"PDF HandySignatur"* is the German name for the developed Android app.
[4] The Android HTTP client API is based on the Apache HTTP client API

input is required and all communication with the PDF-AS Wrapper and Mobile Phone Signature can be carried out in the background.

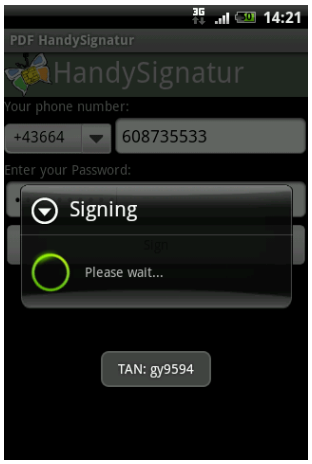

Figure 4: Incoming SMS messages are intercepted and contained TANs are extracted by the SMS Interceptor component

As soon as the signature password has been submitted to the Mobile Phone Signature, a TAN is sent to the mobile phone via SMS. This is where the *SMS Interceptor* component (cf. Figure 2) comes into play. The SMS Interceptor component is registered within the Android OS as SMS handler with the highest priority. This way, it receives incoming SMS messages prior to all other applications (e.g. the standard SMS app). The SMS Interceptor checks whether an incoming SMS originates from the Mobile Phone Signature. If this is the case, it extracts the TAN and communicates it to the app's PDF-AS Signer component. Figure 4 illustrates this step.

After extraction of the TAN, the SMS message is discarded. This way, other apps (e.g. standard SMS app) do not notice and cannot disturb the execution of the PDF signing process.

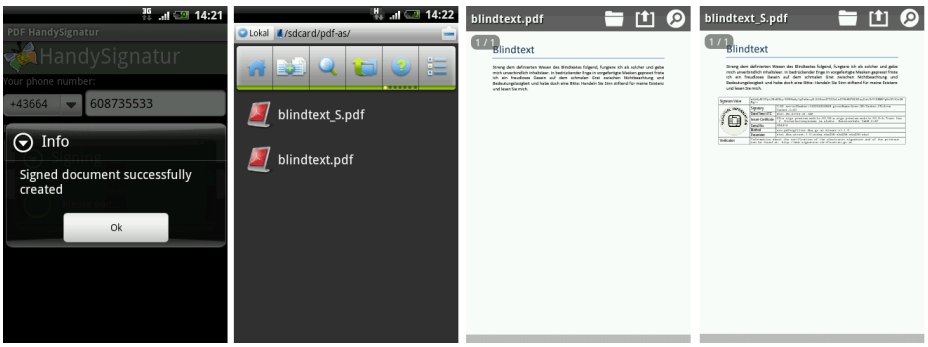

Figure 5: After completion of the PDF signing process, the signed PDF file is stored on the local file system.

As soon as the TAN has been submitted to the Mobile Phone Signature, the computed signature value is returned by the Mobile Phone Signature to the PDF-AS Web-Application. The PDF-AS Web-Application finally adds the signature block to the PDF document and returns the singed file to the PDF-AS Signer component via the PDF-AS Wrapper. This is illustrated in Figure 5. The signed PDF document is saved with the extension "_S" ("S" for signed) parallel to the unsigned PDF document. The figure also illustrates the original unsigned file and the signed PDF document, which now contains an additional visible signature block.

5. Conclusions

In this paper we have introduced a mobile solution for the creation of qualified PDF signatures using the Austrian Citizen Card concept. It allows for the creation of PDF signatures that are equivalent to handwritten signatures according to the EU Signature Directive. The proposed solution has been prototypically implemented for the Google Android platform. The developed Android app provides Austrian citizens a user-friendly mobile alternative to electronically sign arbitrary PDF documents with their Citizen Card.

Our solution is based on the PDF-AS Web-Application and the Austrian Mobile Phone Signature, which both represent core components of the Austrian e-Government infrastructure. Acting as SSCD, the security of the Mobile Phone Signature is crucial. Amongst other factors, the security of the Mobile Phone Signature is also based on the use of two different communication channels. In the typical scenario, users communicate with the Mobile Phone Signature via a Web browser to provide required credentials. The SMS based delivery of the TAN represents the second communication channel.

Our solution combines these two communication channels. All required communication with the Mobile Phone Signature is implemented by the developed Android App. We are currently evaluating if and to what degree this potentially decreases the security of our solution. Integration of results of this evaluation process into future developments of our solution is regarded as future work.

References

[1] 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. Official Journal of the European Communities. (1999).

[2] Posch, R., Leitold, H.: Weissbuch Bürgerkarte. Austrian Federal Ministry for Public Services and Sports; Federal IT-Coordination (in German). (2001).

[3] Eastlake, D., Reagle, J., Solo, D.: XML Signature Syntax and Processing (Second Edition), W3C Recommendation, http://www.w3.org/TR/xmldsig-core/ (accessed 2012-03-19).

[4] Leitold, H., Posch, R., Rössler, T.: Media-break resistant eSignatures in eGovernment – an Austrian experience. Emerging Challenges for Security, Privacy, and Trust - 24th IFIP SEC. (2009).

[5] Zefferer, T., Teufl, P.: Opportunities and Forthcoming Challenges of Smartphone based mGovernment Services. European Journal of ePractice, Volume 13. (2011).

[6] Bundesgesetzblatt für die Republik Österreich BGBl. I Nr. 190/1999. The Austrian Signature Act. (1999).

[7] Bundesgesetzblatt für die Republik Österreich BGBl. I Nr. 10/2004. The Austrian E-Government Act. (2004).

[8] Orthacker, C., Centner, M., Kittl, C.: Qualified Mobile Server Signature. Proceedings of the 25th TC 11 International Information Security Conference SEC 2010. (2010)

[9] Leitold H., Hollosi, A., Posch, R.: Security Architecture of the Austrian Citizen Card Concept. In ACSAC '02: Proceedings of the 18th Annual Computer Security Applications Conference, page 391, Washington, DC, USA, 2002. IEEE Computer Society. (2002).

[10] Leitold, H., Posch, R., Rössler, T.: Media-break resistant eSignatures in eGovernment: an Austrian experience. In Javier Lopez Dimitris Gritzalis, 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).

[11] EGov-Labs: PDF-AS Project Information. http://egovlabs.gv.at/projects/pdf-as/, (accessed 2012-03-19).