Reliable electronic certification on mobile devices

Nuno Alvarez Fernandes*
nuno.alvarez@tecnico.ulisboa.pt

Abstract

Nowadays many documents are still signed in a handwritten way, being highly susceptible to forgery. Digital signatures address this vulnerability by providing a cryptographically secure way to do it. They provide a secure and reliable way to sign digital documents, thereby improving the security of the three key services stipulated by the handwritten signatures: i) Authentication: the signer is who he or she claims to be; ii) Integrity: the data has not been modified or tampered with since the signature was applied; iii) Non-repudiation: an irrefutable proof of signature. Furthermore, this type of signatures can also be performed remotely.

With the exponential growth in the use of mobile devices in everyday life, there is an increasing availability of mobile technologies, giving rise to new applications that take advantage of such devices to improve the way users perform their daily tasks. The work herein proposed aims to facilitate the signing process of digital documents on mobile devices by creating a viable and trusted certification system that uses mobile devices, eliminating the need for external readers, and increasing the users' flexibility. Specifically, it consists of a simple and intuitive mobile application that enables users to digitally sign electronic documents on their devices, using a private signature key stored in a smart card (in this case cased on a micro SD card) inserted in the device, thus allowing to provide qualified digital signatures. All private material can be transferred from one device to another simply by moving the secure micro SD card. Thus, by using a hardware-based security technique, the developed solution provides a protected environment for the user credentials which is never exposed, and cannot be compromised.

Keywords: Security; Qualified Digital Signatures; Mobility; Smart Cards

1 Introduction

Over the years, many cases of forgery have occurred with handwritten signatures. Digital signatures provide a way to perform this process digitally. While serving the same purpose as a handwritten signature, a digital signature uses digital keys (asymmetric cryptography) instead of using pen and paper, strengthening the three security properties previously mentioned – authentication, data integrity and non-repudiation. If properly created, they can have the legal equivalent of handwritten signatures, according to the European Union (EU) directive for electronic signatures [1].

With more and more users using mobile devices, such as laptops, smartphones and tablet computers, mobile computing has become one of the main sources of processing, through which most users accomplish their daily electronic tasks.

There are several different solutions to provide digital signatures on mobile devices, based on asymmetric cryptography and capable of providing qualified signatures. The existing solutions are divided into: those that only make use of the Subscriber Identity Module card of the mobile phones; those that use both the SIM card and the middleware of the devices to aid in the execution of the signature functions; and those that use the SIM card and high level services to provide digital signatures. However, all these solutions use the SIM card as the security element that stores the sensitive data used in the

*Instituto Superior Técnico

signing processes. The main problem with this is that the SIM card is dependent on the mobile operator, requiring the user to have a SIM card that has a usable cryptographic token, which is hard to find.

The work herein presented differs from the existing solutions by considering the use of a mobile device to perform the digital signing of documents, interacting with a "separate" smart card contained in that same device, ensuring that even if the device is attacked, the security of the signature remains intact.

Section 2 covers the state of the art, analyzing the concept of digital signatures, and the different protocols, standards and mobile signature solutions that currently exist. It also examines the mobile operating systems and smart cards. Section 3 describes the proposed solution, taking into account the state of the art. Section 3 details the proposed solution, describing the technologies selected to implement it. Section 4 evaluates the proposed solution. Section 5 concludes this paper with some final remarks.

2 State of the Art

Section 2.1 presents the digital signatures, and describes the different protocols that can be used to perform them. It also describes the standards and the existing mobile signature implementations. Section 2.2 presents the available mobile operating systems in the market. Section 2.3 introduces the smart cards, describing its technologies, formats, and communication protocols.

2.1 Digital Signatures

Over the last years the number of e-government, e-commerce and business information systems has been increased. Ensuring the security of such systems is progressively demanding, because malicious entities are constantly devising new ways to impersonate someone and forge documents.

2.1.1 Algorithms and Protocols

There are several solutions that implement the signing process by using public key cryptography [2]. Although these solutions have the same purpose, they differ in some aspects, such as the storage of the private keys, the type of information on which they are based on and the type of technology they use. One of these digital signature systems is based on smart-cards, in which smart cards store all the necessary cryptographic information of the user (e.g., keys and certificates).

Asymmetric Key Cryptography

Asymmetric key cryptography, also known as public-key cryptography, refers to a cryptographic technique which uses a pair of keys – one private and the other public - that share a mathematical relation. The public key is used to verify digital signatures whereas the private key is used to create the signatures. One of the most widely used digital signature algorithms today is RSA. When using this type of cryptographic system, besides the need to keep the private key a secret, it is also necessary to be cautious with the origin of the public key. Trusting this key is a delicate procedure and represents a major concern in order to assure the correct association between this key and its owner. If this association is corrupted, it could lead to falsely signed documents. Typically, this concern is addressed through the use of a Public-Key Infrastructure (PKI), in which one or more third parties, named Certificate Authorities (CAs), certify the ownership of the key pairs, meaning they bind the public keys with the respective user identities.

Smart Card-based Protocol

A smart card (also called chip card or Integrated Circuit Card (ICC)) is a portable computing device which contains programmable memory and offers some tamper-resistance capabilities. Over the years,

companies and end-users have given increasing importance to public-key cryptography, because they see it as a very reliable technology to ensure the security of their applications [3]. Today's cryptographic smart cards can perform on-chip key generation and store the private key to avoid the risk of having more than one copy of it. They can also provide a secure computing environment for private key operations. This type of cards can be seen as a security token able to provide secure digital signatures. The main weakness of this solution is the high risk of loss, theft (leaving the scheme's security reduced to that of the PIN system) or breaking of the smart cards.

2.1.2 Standards

There are several signature formats defined which can be divided into three classes: Cryptographic Message Syntax (CMS) based signature formats; Extensible Markup Language (XML) based signature formats; and Portable Document Format (PDF) based signature formats. Only the XML based formats are analyzed hereinafter, since they are the ones that are used by this thesis.

XMLDSig: XMLDSig, is defined by the World Wide Web Consortium (W3C) Recommendation, in the XML Signature Syntax and Processing, and defines an XML syntax for digital signatures. It comes from the need for native XML security services, in order to ensure the security of transactions that make use of XML. It is intended to assure the integrity, message authentication and/or signer authentication for data of any type. One advantage of XMLDSig is the possibility of signing only certain parts of an XML document. Therefore, an XML document can possess several signatures, created at different times, by different entities and associated to different data. Also, an XML Signature is flexible because is operates on the XML Information Set, which allows to work on subsets of the data and consequently to have different ways to bind the signature and signed information.

XAdES: XAdES is a standard which comprises a set of extensions to XMLDSig signed data. Since XMLDSig lacks certain important features of extended electronic signature, the most evident of which is Long Term Validity, XAdES specifies a number of additional profiles for use with advanced electronic signature in the meaning of EU Directive 1999/93/EC [1]. Since digitally-signed documents may be used or archived for long periods of time, it must be possible to prove signature validity at any time in the future in spite of the technological advances. Accordingly, any later attempts by the signer or the verifying party to repudiate the validity of the signature will be futile. This is an important benefit, because even if the cryptographic algorithms responsible for the signature are broken, the electronically signed documents can keep their validity for long periods. There are six profiles defined by XAdES: XAdES – basic form, XAdES-T (Timestamp), XAdES-C (Complete), XAdES-X (eXtended), XAdES-X-L (eXtended Longterm) and XAdES-A (Archival). The XAdES specifications provide an XML solution.

2.1.3 SIM Card-supported mobile signature solutions

There are a variety of mobile technologies and infrastructures that support digital signatures by using the SIM card (smart card) as the security token, thus providing qualified signatures. Some use only the SIM card while others use the SIM card and the middleware of the mobile device. There are also those that, besides using the SIM card, include high level services that are independent of the mobile device's technology. One solution beloging to each category is presented.

SIM card-based: One solution is the WIM (Wireless Application Protocol (WAP) Identity Module), a security module that is implemented in the SIM card, and that has tamper-resistant capabilities, with

the main purpose of being used by WAP applications. It provides secure storage for cryptographic credentials (certificates, keys and authentication objects (PINs)) as well as processing capabilities. Thus, it allows the execution of cryptographic operations, providing the WAP applications with security services and allowing the use of digital signatures. The WIM functionality is implemented as an independent application on the SIM card. This way, the key pairs are created inside the card, the signature processes are done by the WIM app and the keys never leave the card, causing the SIM card to be considered as a Secure Signature Creation Device. For these reasons, the digital signatures generated by the WIM app can be classified as qualified signatures [4]. Presently, the level of usage of WAP is very low, since all modern mobile devices support full HTML browsing and do not use any WAP markup, like WML.

Handheld and SIM card-based: One solution is The Security And Trust Services API (SATSA), which is a specification for Java Platform ME, also known as Java Specification Request (JSR)-177, that enables Java ME applications (Java ME MIDlets) to communicate with a security element, such as a SIM card [4]. This specification allows the security element to perform different security processes on Java ME MIDlets like electronic signatures. SATSA provides some advantages such as the MIDlets' portability, allowing them to be transported between different mobile devices, and the ability to perform signature processes that follow the standards by using a WIM application, resulting in qualified signatures. The main weakness is the low number of deployments by device manufacturers.

Services-based: One solution is the Mobile Signature Service (MSS), a specification defined by the ETSI designed to simplify the development of mobile signature-based solutions for the mobile application/service providers, and is comprised of: end-user, smart card issuer (mobile operator), Registration Authority, CA, Mobile Signature Service Provider (MSSP), roaming MSSP and Application Provider. In this type of systems, the APs do not need to supply the end-users with any kind of software. If the AP have a transaction that requires an electronic signature, they send a request to the MSSP with certain data related with the transaction, and this one invokes the user's signing application, obtaining the signature of the provided data. Its main advantage is that APs do not need to develop specific signature solutions for the different mobile devices. That is performed by the MSSP. Thus, the MNO does not allow any third party to access the user's SIM card. In contrast, its main problem is that the AP is obligated to set up an agreement with an MSSP capable of delivering the aforementioned services.

2.2 Mobile Operating Systems

Nowadays, there are many MOSs occupying the market with different proportions. [5] shows the market share of the various mobile platforms in terms of online usage, with Android devices browsing the Web more than iOS devices or any other MOS. Given this, and the fact that the Android platform is the one used in this work, Android is the only one analyzed.

Android: Android is the world's most popular mobile OS and is currently developed by Google. Its main implementations are focused on smartphones and tablet computers. This MOS is defined as an open-source platform, enabling developers to use third-party tools (libraries) to create or optimize their applications. The open nature of Android is a significant advantage. Also, the market share of this MOS is another an advantage compared to all others. In terms of security, Android runs its applications in full isolation (apps are "sandboxed") and therefore does not allow them to interact with sensitive data or system files/resources, or to modify or collect information stored by other apps [6]. Yet, within this isolation policy, apps can exceptionally examine each other's programming logic (but not the private data) and have shared access to the device's removable storage, like the SD card. A serious problem of

Android is hardware and software fragmentation, making it harder for developers to create new apps.

2.3 Smart Cards

Smart processor cards are the ones considered herein, as they can store and process information, and can also perform some on-card functions such as digital signing of data. To be considered secure, smart card systems must be tamper-resistant, meaning that the data contained therein cannot be obtained successfully through physical attacks. To achieve this, most smart cards are comprised of anti-physical tampering mechanisms and cryptographic functions, like ciphering and hashing, which are implemented by hardware specific-processors in such a way that they are able to prevent side-channel attacks [7].

A well known smart card OS is the Java Card. Its technology enables small applications (applets) to be run securely on a wide range of smart cards. Any smart card supporting Java Card technology, regardless of its vendor and underlying hardware, can run applets developed with Java Card technology.

Currently there are different types of formats used in different solutions [8]. A specific format considered in this thesis is the Micro SD. This is a flash memory card in which a smart card can be embedded, and can thus be deployed with the strength of smart card chips. Most Android-based devices support them through a micro SD slot. To communicate with the smart cards, the messages used follow a specific format defined in ISO/IEC 7816-4, the APDU. When the master of the communication sends a command to the smart card, the message must be in the command APDU format.

3 Proposed Solution

As seen in the related work, particularly in Section 2.1.3, there are several solutions that deliver qualified digital signatures via mobile devices. These solutions use the SIM card of those devices (which is a smart card) as the security element performing the necessary cryptographic tasks. The problem is that these cards are provided by MNOs and there are very few that are capable of offering cards with a usable cryptographic token, leaving the user with two unsatisfactory options: acquiring a dual SIM device (which has a low commercial presence in the mobile industry), with a common mobile SIM card and a SIM card which includes the cryptographic features; or owning two different devices, each having a SIM card implementing a different function.

A possible solution is the use of a “separate” smart card that seamlessly integrates the mobile devices as the security element, such as a mobile security card. For this a micro SD card which contains an integrated smart card element can be used. This solution has several advantages: it does not depend on the mobile operator, allowing the user to take advantage of cryptographic features on any mobile device covered by any mobile operator; there is a wider availability of such cards in the market; and it has a stronger portability concept, as it is easier to remove/insert a micro SD card from/in mobile devices.

The solution herein consists of a mobile certification system based on a smart micro SD card. The private keys used in the signature processes are securely stored inside the card and can only be used by means of user authentication, minimizing the risk of being used by attackers or malicious applications, as they would need to gain access to the device and authenticate as the legitimate user.

Section 3.1 provides an overview of the proposed solution and of its main components, including their roles for carrying out the desired features. Section 3.2 presents the trust model, describing the assumptions taken by the proposed solution with respect to the main components. Following, Section 3.3 presents the proposed solution with further detail by describing its architecture and selected technologies.

3.1 Overview

The goal of the proposed solution is to enable users to digitally sign any electronic document in their own mobile devices using secure cryptographic keys. To achieve this, a viable and reliable signing system is proposed. In order to assure its functionalities, the proposed solution (depicted in Figure 1) is comprised of two main components, the certification application installed on the user's mobile device and the signing module running on the smart card. The first handles all communication with the smart card via APDU messages, sending all user requests to the card and receiving the respective responses. The second component fulfills the user requests, performing the necessary cryptographic operations, and provides a secure environment to store sensitive data, such as the private keys.

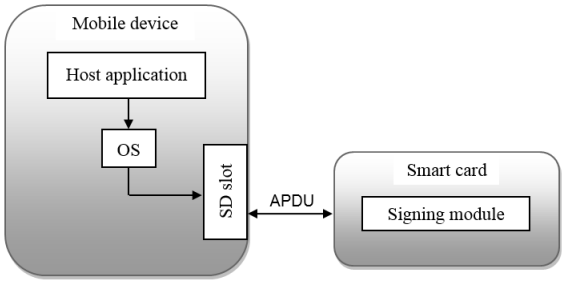

Figure 1: Certification system overview

3.2 Trust Model

The main components identified in Section 3.1 are executed over different environments. In order to better understand how these components work, it is important to take certain assumptions about them.

Mobile app's assumptions:
1. The mobile device's OS is trustworthy – The OS where the app is installed is reliable and not in control of a malicious user;
2. The user inserts the proper smart card in the mobile device;
3. The mobile device only sends data to the smart card if authorized by the user;
4. The correct hash is provided to the smart card - The hash of the document initially chosen by the user is indeed delivered to the card, and not replaced or tampered with by a malicious entity.

Signing module's assumptions:
1. The smart card has tamper-resistant capabilities;
2. The data stored within the signing module cannot be accessed by any other applet contained in the smart card;
3. The smart card is managed only by the trusted entity responsible for the creation of the private data in the card, such as the signature keys or public-key certificates;
4. The private keys are securely stored – The keys used to sign documents never leave the card;
5. The card owner is the only one who knows the PIN value to access the smart card – If the PIN value is used to access the card, then the authenticated user is the expected user.

3.3 Architecture

The architecture is designed in a way that it enables the integration of the certification system with different systems or applications. Thus, the goal is to create an interoperable solution that has a low-cost to the end user. To achieve this, the proposed solution makes use of a smart card with a secure signing module, and a middleware that ensures the proper communication between the card and the end-user application, as is depicted in Figure 2.

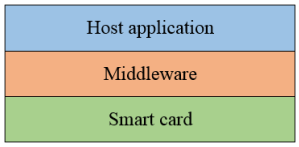

Figure 2: Proposed solution architecture

The Smart card layer provides a secure storage of the private data inherent to the signature processes and the execution of all cryptographic operations. The Middleware layer builds and maintains the communication between the card and the user's application, "speaking" the language of both and performing the respective translations through the use of APIs. The Host application layer is the one that interacts directly with the users and receives all their requests.

Selected technologies: For the development of the smart card behavior, the technology chosen was the Java Card. It enables a certain level of abstraction over the smart card implementation, making it possible for the applications to be independent of the card's underlying hardware. Also, Java Card implements a firewall in order to keep the applets from accessing data that does not belong to them, thus ensuring data security.

The middleware is further divided into two components: the Host application API and the Smart card library. First, to implement the application API the selected interface standard was the PKCS#11. This is a well known and widely used API standard intended for cryptographic devices associated with a single user, such as a smart card, and supported on several operating systems. Regarding the smart card library, this is the component that communicates directly with the smart card. It is provided by the card vendor (GO-Trust) and is named GTSDUpi. For the application to be able to communicate with the card, a PKCS#11 module was required. A PKCS#11 module is a software library with a defined API that allows the access to a cryptographic hardware. This module was developed with the help of the IAIK PKCS1#11 Wrapper for Java, a library for the Java platform that enables the access to PKCS1#11 modules from within Java.

To provide an appropriate implementation of the chosen signature format (XAdES), a library was selected - JDigiDoc. This is a Java library designed for applications that handle digital signatures and their verification, using smart cards or other supported cryptographic tokens.

The host application is further divided into two apps: the Client app and the Registration Officer app. The programming language selected to develop both apps was Java, capable of running on multiple mobile platforms such as Windows and Android. The Client app has a GUI, built using the Swing technology, designed to guide the user while performing the cryptographic operations. The Officer app presents a command-line interface in which the user enters the desired command and associated parameters in order to create, delete or import a particular object.

4 Solution Evaluation

A relevant part of the evaluation of the proposed solution is the **security analysis**. Starting with the Java Card system: since it provides a secure environment for its applications to run, the user's sensitive information is securely protected against logical attacks targeting the card. It is also protected against physical attacks due to the smart card's tamper-resistant capabilities. Even if the user loses the smart card, or the card is stolen, a malicious person cannot have access to the information contained within the card, as this data is kept inside Java applets which ensure that the private data never leaves the card.

The signature applet is responsible for assuring the proper storage, management and use of the sensitive information. The operations that require the use of this information (e.g., a private signature key), such as the digital signing, always require the local authentication of the user. This authentication is managed by the applet, forcing the user to perform an authentication with the correct PIN value. This way, the system is able to ensure the user's presence at the time of the digital signing process.

It is also important to consider the vulnerable point present in the connection between the smart card and the certification app, in particular regarding the digital signing operation of the solution. This operation involves the selection of a document by the user and consequent hash, both made in the certification app. The sending of the hash to the card is vulnerable in the sense that it does not assure unequivocally that the hash does not belong to another document. There could be the case where an attacker could find a way to tamper with or replace the user's hash with his own without the user noticing, causing him to sign a malicious hash. Thus, providing the correct hash to the card is assured by the certification app, which is an assumption required to ensure the proper functioning of the solution.

In relation to the certification application, it is tasked with providing to users all the operations that the proposed solution can perform, and presenting their results. It runs on an OS, such as Windows or Android, which is also a vulnerable point, and subject to attacks. If a malicious user were to gain control of the user's OS without him noticing, the user requests could be tampered with, possibly leading to the discovery of sensitive data, such as the authentication PIN. Thus, the OS must be trustworthy, which is also a necessary assumption to make sure that the solution can operate securely. Neverthless, the private keys stored on the smart card can never be retrieved by the attacker.

It is also necessary to analyze the behavior of the solution and how it conducts its most commonly used operations, especially when communicating with the smart card. To perform this evaluation, **performance tests** were carried out on the client app using Process Explorer. The tests were accomplished over an Intel Core i7-4700HQ CPU at 2.40 GHz, with 8 GBs of memory, and with the Microsoft Windows 8.1 as OS. The tests (depicted in Figure 4) consist on performing several operations, which involve making requests to the smart card for writing or creating data, and for reading data.

The first graph represents the evolution of the CPU usage in percentage terms. It is seen that this usage reaches its peak on the digital signature operation, with approximately 10%. This is a not a high value, yet it is the operation that causes the most CPU consumption. This is because the operation is comprised of many tasks: it performs the hash of the file selected by the user and sends it to the smart card to be signed. Then, upon the reception of the signed hash, the application requests to the card the export of the public-key certificate associated with the used signature key, so as to incorporate its information on the signed file, thus finalizing the signing process. The second graph shows the behavior of the amount of memory requested by the application to the system. Also, it is the digital signature operation that requires the largest amount of the resource, approximately 10 MB, because of the reasons previously stated. None of the operations requires more than 10 MB of memory. The total memory

requested by the application throughout its cycle is approximately 58 MB. The third graph depicts the input-output operations that happen through the file system where the application runs, namely the creation, writing, reading and/or deleting of data. The operation corresponding to the verification of the signed file stands out due to the reading of the signed file by the application, and the huge amount of validation tests that it performs, searching for any validation warnings or errors that the signed file may have.

To have a practical evaluation of the solution, **usability tests** were performed with twenty users (most of them belonging to PDM&FC). These tests were performed on both applications to achieve a thorough evaluation of the system. The users performed several pre-defined tasks with the smart card physically attached to a laptop. A summary of the test results is depicted in Figure 3. These results show that the majority of the users felt that the solution is easy to use, thus providing a good user experience, and that is has a strong security level.

Figure 3: Usability tests summary

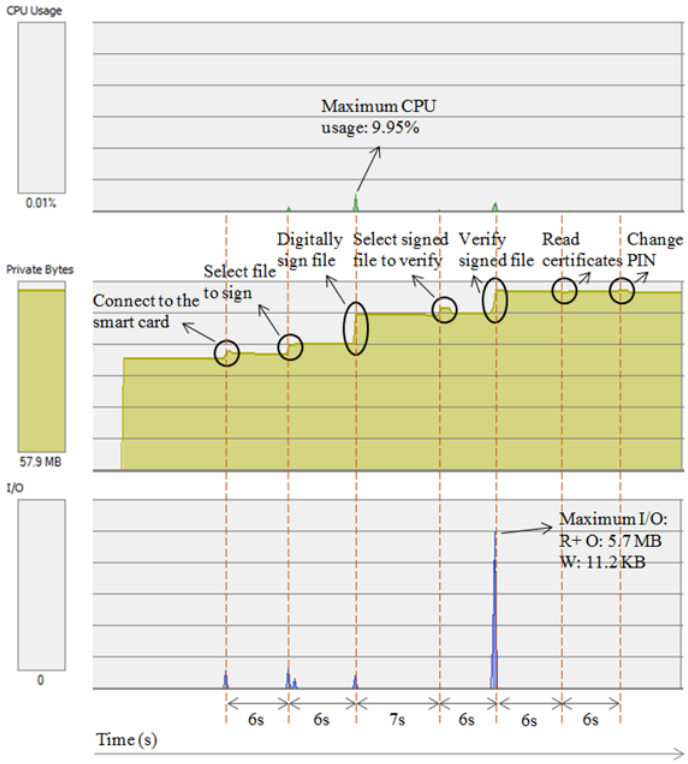

Figure 4: Solution performance tests

5 Conclusion

The strong presence of mobile technologies in the surrounding environments has an obvious influence on today's digital solutions. There are various signature mechanisms that use mobile devices, which are divided into: those using only the SIM card of the mobile phone, those that use both the SIM card and the device's middleware, and those using the SIM card and high-level services. Each has its own limitations, but they all share a common drawback: using the SIM card as the security element. Such card is dependent on an operator, forcing the user to acquire a SIM card that has a usable cryptographic token, which mainly is not something that is easy to find. The proposed solution is an electronic certification system that provides qualified digital signatures using a secure element inserted in a mobile device, eliminating the need for external readers. The considered secure element is a smart card on a micro SD card, eliminating the previous drawbacks, as it is independent from any operator and has a higher availability in the market. The user's sensitive data is stored inside the card, which has tamper-resistant properties and secure computing capabilities. To access the sensitive data, the user has to authenticate himself. Online attacks (e.g., brute force attacks) are prevented by blocking the card once the maximum number of authentication attempts is reached. Thus, even if the mobile device is attacked, the security of the digital signatures remains intact. Another advantage is its interoperability, providing its interface via the PKCS#11 standard. Any application that complies with this commonly used standard can use the developed system without having to know its details when communicating with it.

References

[1] The European Parliament and the Concil 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. *Official Journal of the European Communities*, L 12:12–20, 2000.

[2] Florian Nentwich, Engin Kirda, and Christopher Kruegel. Practical security aspects of digital signature systems. *Technical University Vienna. Technical*, 2006.

[3] Helena Handschuh and Pascal Paillier. Smart Card Crypto-Coprocessors for Public-Key Cryptography. In Jean-Jacques Quisquater and Bruce Schneier, editors, *CARDIS*, volume 1820 of *Lecture Notes in Computer Science*, pages 372–379. Springer, 1998.

[4] Antonio Ruiz-Martínez, Daniel Sánchez-Martínez, María Martínez-Montesinos, and Antonio F. Gómez-Skarmeta. A Survey of Electronic Signature Solutions in Mobile Devices. *J. Theor. Appl. Electron. Commer. Res.*, 2(3):94–109, December 2007.

[5] Net Applications. Mobile/Tablet Operating System Market Share. `http://www.netmarketshare.com/`. Online; accessed 20-November-2014.

[6] Google. Android security overview. `https://source.android.com/devices/tech/security/`. Online; accessed 21-November-2014.

[7] Keith Mayes and Konstantinos Markantonakis. *Smart Cards, Tokens, Security and Applications*. 1 edition.

[8] ISO/IEC 7810:2003 Identification cards – Physical characteristics. [Online]. Available: `http://www.iso.org/`.