The smartphone as a trusted platform

Rickard Öh Julien Lancry
Email: {ricoh905,julla181}@student.liu.se
Supervisor: Anna Vapen, {annva@ida.liu.se}
Project Report for Information Security, Second Course
Linköpings universitet, Sweden

Abstract

With the increased use of smartphones in security critical applications there is a need to implement security features to make the smartphone a trusted platform. This theoretical report aims to give the reader an overview of some of the existing solutions to make the smartphone a trusted platform. It will cover hardware based solutions like MTM and TrustZone but also Java Card which aims to make a smart card (SIM/USIM) more secure.

1. Introduction

The main objective with this report is to evaluate the possibility of using the smartphone as a trusted platform. We will explore different techniques to do this. Most of the work will be focused on hardware. But some software based solutions will be addressed.

This report will try to address the following topics:

- What is a smartphone?
- Why does a smartphone need to be a trusted platform?
- How could a smartphone be implemented as a trusted platform?
- How can the trusted mechanisms be attacked?

A smartphone is a mobile phone but with more advanced capabilities. A smartphone has more built-in processing power, memory and more advanced features like: complete operating system, built-in keyboard. In other words it is a miniature computer with phone capabilities.

The first smartphone called “Simon” was introduced by IBM and Bellsouth in 1994. Simon did not become a success due to the fact that it was very heavy and costly.

Today smartphones are very popular in business applications and with the constant growth of iPhone and Android smartphones the market is getting bigger and bigger. The mobile security market is expected to hit $889 million in 2011 and to hit $4 billion in 2014 [20] [21].

2. Background of smartphone security

Today smartphones are used in almost every way a regular computer is used. You can email, do e-commerce, surf the web, banking and have VPN connections. All these services sometimes require sensitive information to be stored, like: emails, keys, personal information and other sensitive data. These needs introduce new requirements on smartphone security. Of course some smartphone usage scenarios demand more security than others. This report will explain and evaluate different solutions on how to make a smartphone a trusted platform and some examples of how existing solutions can be attacked.

For instance, the e-commerce requires some tools to ensure a high level of security on the smartphone platform because handling critical data such as credit card number is a major concern. The user needs to trust the mobile device not to disclose the user’s private information.

3. Solution and Analysis

In this section, several ways of implementing trusted computing mechanisms on smartphone will be discussed in both hardware and software point of view.

3.1 Mobile Trusted Module (MTM)

The Trusted Computing Group (TCG) has defined a series of specifications regarding the use of trusted computing with mobile devices such as smartphones. It also has pointed a series of recommendations on how to implement MTM in a document called the "TCG Mobile Reference Architecture"[17].

The Mobile Phone Work Group (MPWG), the team inside TCG responsible for the mobile specifications, released the Reference Architecture specifications and the Mobile Trusted Module specification in June 2007. The specifications provide the core framework,

commands and control specifications needed to provide a TCG-based security building block solution in mobile phones.

The TCG Mobile Trusted Module specification has been developed with the particular needs and limitations associated with mobile devices in mind. Indeed, it is not conceivable to simply export the TPM module to the mobile environment due to special requirements in this topic. Nevertheless, MTM is a specific "branch" of the TPM general specification applied to the mobile industry. Its aim is the same as TPM and has many things in common with it. The explanation why the existing TPM was not enough for mobile platforms is because a mobile phone is often an embedded implementation and because protections for the regulatory parts (e.g. the network provider and the device manufacturer) of a mobile phone should never be turned off.

The general assumptions on MTM will first be outlined and then the main technical principles will be presented with all the specific concepts of MTM regarding TPM.

3.1.1 MTM general assumptions

As said before, the MTM specification is mainly TPM-based with some other specific "commands" to the use of mobile platforms. Precisely, MTM's origin lies in the TPM v1.2, but the mobile specification significantly differs from the original specification by removing certain non-revealing part regarding mobile platforms and by adding some others functions as[16]:

- The concept of *secure boot* is introduced. As many mobile devices are subject to regulatory approval, there is an obvious need for ensuring that the integrity of the platform (and especially the software part) has not been compromised. That uses mechanisms of measurement and validation to make sure the system will behave in the right way.
- The specification explicitly supports implementation of the MTM as functionality rather than as a physical implementation in hardware. Every part of the MTM specification can be implemented as a software-based solution.
- Moreover, the reference architecture takes into account the support of several parallel MTM instances in the same device. They are called “engines” and provide trusted services to their stakeholders. Some will be discretionary (MTM exposed to user applications) whereas e.g. the Device Manufacturer MTM by definition enforces security policy (mandatory access control).

All the important concepts of TPM are present in the MTM module and the MTM specification only adds some specificity compared to TPM. The first one is certainly the main specificity of MTM because it draws the ability for a mobile platform to be trusted and behaves like expected. The second one is also important because it means that the TCG Mobile Trusted Module specification should be just seen as general mechanisms for trusted computing on mobile platforms and not as a strong technical way of possible implementation in hardware or software. The device manufacturer is the only party that can decide whether functionality has to be implemented in hardware or software on the device. Eventually, the last point depicts the need of having secured trusted services (mandatory services) for roots critical functionalities whereas user functionalities should be discretionary and using the services provided by trusted "engines".

3.1.2 MTM in practice

The TCG Mobile Trusted Module specification [16], and the Mobile Reference Architecture [17], abstracts a trusted mobile platform as a set of trusted "engines". Those trusted "engines" are constructs that can manipulate data, provide evidence that they can be trusted to report the current state of the engine, and provide evidence about the current state of the engine.

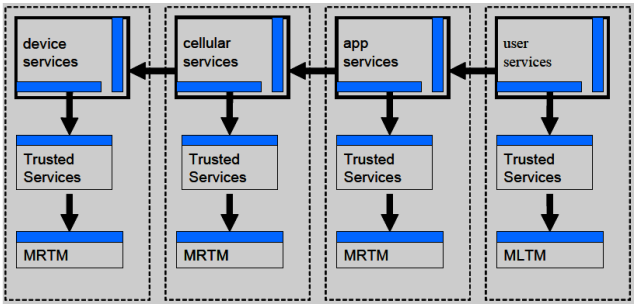

Figure 1. Example of a generalized mobile platform [16]

The MPWG gives an example of a generalized trusted mobile platform, shown in Figure 1, in its specification document. It contains multiple abstract engines, each acting on behalf of a different stakeholder. A stakeholder is an entity that has a role to play in the mobile platform and its life cycle.

The main stakeholders in a trusted platform according to the TCG consortium are [17]:

- **Users**, who use the mobile device physically, manipulate and store their data in the platform. There may be multiple *User* stakeholders in a

platform. An employee or a consumer may be a User stakeholder, for example.

- **Service Providers**, who provide services, consume in a platform such as applications which are a typical use for a smartphones (e.g., the AppStore by Apple or the AndroidMarket by Google). There may be multiple *Service Provider* stakeholders in a platform. Some examples of services are: corporate services for employees, content distribution services for consumers, address book, diary, etc.
- **Communications Carriers**, who are specialist *Service Providers* providing cellular radio access for the platform (e.g., the 3G mobile operator that provides the access to the phone network and nowadays to the Internet also). There may be multiple *Communications Carrier* stakeholders in a platform.
- The **Device Manufacturer**, who provides the internal communications within a platform and typically provides all the hardware resources within a platform. There is a single *Device Manufacturer* stakeholder in a platform as there is only one possible manufacturer for a device.

Getting back to the architecture depicted in Figure 1, the device engine provides basic platform resources, which may or could include a user interface, a radio transmitter and receiver, Random Number Generator, the IMEI, a SIM interface and other inputs/outputs interfaces within the mobile device. The device engine provides its services to an engine that provides cellular services. The cellular engine provides its services to an application engine, and the application engine provides its services to the user.

In each engine, conventional services have access to Trusted Services, which make measurements of the conventional services and store those measurements in a Mobile Trusted Module (MTM). The device, cellular, and application engines have a Mobile Remote-owner Trusted Module (MRTM), because those stakeholders do not have physical access to the phone and need a secure boot process to ensure that their engines do what is needed. The user engine has a Mobile Local-owner Trusted Module (MLTM), because the user does have physical access to the phone, and can load the software he wishes to execute. The MTMs can be trusted to report the current state of their engine, and provide evidence about the current state of the engine. The MRTM differs from the MLTM primarily in that the MRTM contains additional Protected Capabilities to support a secure boot process.

The MRTM consist itself of a subset of the TMP v1.2 plus a set of new mobile-specific commands designed to support the requirements set by the MTM specification. In addition, a Root-of-Trust-for-Verification (RTV) and Root-of-Trust-for-Measurement (RTM) module would be the first executable running in the runtime environment.

3.1.3 MTM use case examples

In 2005, when the work on mobile trusted computing started in the Trusted Computing Group, several basic use cases were identified among the potential applications of trusted computing in the mobile environment.

The aim was also to demonstrate how every stakeholder in the system (e.g., the device manufacturer, the user, etc.) could benefits from a Mobile Trusted Platform.

These use cases are presented as follows by the TCG consortium [15]:

- *Platform Integrity*, which is a basic requirement for many others use cases and not only for mobile applications. For example, secure boot mechanisms are parts of the Platform Integrity.
- *Device Authentication*, which is a core feature of cellular systems both for the device and the subscriber.
- *Robust DRM Implementation*, which enables the protection of private contents such as films, music, books, etc.
- *SIMLock / Device Personalization*, which is about to ensure that a mobile device remains locked to a specific network and is widely used by mobile network operator. But it is still possible to unlock the (U)SIM in an authorized manner.
- *Secure Software Download*, which is not particularly linked to mobile devices but is critical in the fact that the integrity of the device software has to be kept in any circumstances.
- *Secure Channel between Device and UICC* (Universal Integrated Circuit Card), which provides a secure interface between the device on the one side and the UMTS subscriber identity module on the other side.
- *Mobile Ticketing*, which is a specific application to the mobile environment. The TCG Mobile Phone Work Group distinguishes between two kinds of Mobile Ticketing: the first one where the actual ticket is just a record on a server and the second one where the ticket is actually an object stored in the device.
- *Mobile Payment*, which is not really the big boom expected a few years ago. But with the emergence of smartphones and regarding the increasing number of people buying those devices, we should

probably begin to see more of these applications in real life that take advantage of the smartphones' power.

- *Software Use*
- *Prove Platform and/or Application Integrity to End User*, which is a main characteristic of trusted computing in general and so applies for mobile trusted platform.
- *User Data Protection and Privacy*, which is about how to preserve users' data confidentiality and integrity. Especially critical in business uses of trusted computing with mobile platform.

3.1.4 MTM market implementation

So far, there has not been any market implementation of the MTM specification and Reference Architecture. At time of writing, we have just seen a MRTM (Mobile Remote-Owner Trusted Module) emulator [19], developed by Nokia Research Center, which enables to test different configurations of MTM. This emulator can be found on the Internet, free of charge using.

But what is important and must be noticed is that this Mobile Trusted Module specification makes it possible for device manufacturers to add the MTM as an add-on to already deployed, proprietary security solutions. The manufacturer is the only one to choose an implementation of the different main concepts of MTM. The TCG consortium draws several opportunities to do it [18]:

- By using a specialized MTM chip (derived from a TPM chip but with mobile constraints built-in).
- By using a TPMv1.1 or TPMv1.2 chip and some extra layer in software to implement the extra commands required by the MTM specification.
- By using another hardware chip bounded to the platform and running an MTM application amongst others (typically parallel running MTMs).
- By running a software MTM in a virtualized engine with the visualization environment protected by an underlying hardware MTM.
- By running a purely software-based MTM in a CPU chip that, in this case, provide the hardware base.

3.1.5 MTM's resistance against attacks

Despite there is so far not any implementation in a real product of the Mobile Trusted Module specification, it has to be notice that one former engineer from the US Army succeeded a few months ago to tamper a Trusted Platform Module (TPM) chip. But the task took him more than six months to achieve and he destroyed more than 50 chips (made by Infineon) before succeeding his attack on the hardware [22].

A lot of mechanisms of defense have been built into the chip (hardware) to hardly complicate the task for attackers to tamper the whole system but there is always still a minor risk of succeed and so MTM is not a 100% safe solution for trusted computing.

Eventually, the TCG consortium defines a series of "Preventative Methods" in its Mobile Reference Architecture specifications document. It covers both hardware-based protection capabilities and software-based measures on how to improve the safety of MTM in a mobile platform.

3.2 ARM TrustZone

ARM is a technology company best known for its processors but also develops software development tools, systems and platforms, system-on-a-chip infrastructure. ARM is very dominant in the smart phone/mobile phone processor industry, for example, 98% of the mobile phones sold in 2007 used at least one ARM processor [11]. Several of today's most popular smart phones uses an ARM processor, like the HTC Hero and Apple iPhone/iPhone 3GS [11].

TrustZone is a security technology that enables the applications core to switch between two states, the "normal world" and the "secure world". TrustZone is not implemented on a separate chip, that the MTM can be, but built-in inside the ARM processor. Advantages with a built-in technology are that it does not require extra space inside the phone, and might also be cheaper [14]. The "normal world" and the "secure world" are implemented running each "world" on a separate virtual processor (that are run on the physical processor).

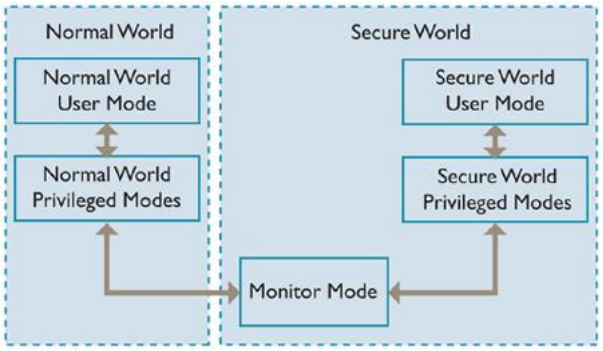

Figure 2. ARM TrustZone, showing the states, normal world and secure world [10].

The two virtual processors execute in a time-sliced fashion, context switching through a new core mode called monitor mode (Figure 2) when changing the currently running virtual processor. The entry to monitor

can be triggered by software executing a dedicated instruction, the Secure Monitor Call (SMC) instruction, or by a subset of exception mechanisms. The normal world processor can only access non-secure system resources and secure world processor can access both secure and non-secure system resources [13]. In order to provide strong isolation between the two worlds, the processor provides an additional status bit called non-secure (NS) bit, which determines in which world the program executes.

The booting of a secure system is a critical point, if the boot can't be trusted neither can the system. Many attackers try to attack the system when it is powered down, and for example replacing the bootstrap code. If a system boots without first checking that the boot code is authentic, the system is vulnerable [13]. TrustZone support verifiable boostrapping and secure non-volatile memory [13].

Some advantages with using TrustZone instead of a MTM (hardware-based MTM) are:

- An MTM would probably cause additional costs [14].
- TrustZone's crypto engine is directly attached to the CPU, which means that cryptographic operations can be carried out faster and more secure. In a MTM-based system the communication is done using an external bus [14].

3.3 (Universal) Subscriber Identity Module cards (USIM/SIM)

This section will explain USIM/SIM cards how they are used and how they can be attacked. A technology called Java Card will also be explained. It provides a secure environment for applications that run on smart cards and other devices with very limited memory and processing capabilities [3]. Most USIM/SIM cards today are based on Java Card [7].

A SIM-card is a subscriber identity module application on a smart card which is used to identify a subscriber on mobile telephony devices. A smart card is any pocket-sized card with embedded integrated circuits which can process data, although today most USIM/SIM cards are 25x15 mm. SIM is the application that runs on SIM card. SIM cards are used in GSM networks and USIM cards are used in UMTS (Universal Mobile Telecommunications Systems) networks. UMTS is built on top of GSM, and is the network commonly referred to as 3G in Sweden. The dominant smart card technology today runs on 4 to 8 Kb of RAM and 32 to 62 Kb of EEPROM, using a slow 8-bit processor [7].

3.3.1 Data

SIM/USIM cards store network specific information to authenticate and identify subscribers on the network. Most important ones [1][2]:

- ICC-ID (Integrated Circuit Card ID)
- IMSI (International Mobile Subscriber Identity), unique number associated with all GSM and UMTS network mobile phone users.
- Authentication key (Ki), a permanent subscriber authentication secret key.

The key Ki has a length of 128 bits and is stored within the USIM/SIM for use in: authentication of the subscriber identity to the network, data confidentiality over the radio interface, file access conditions and GSM conversion functions (only for USIM).

3.3.2 Authentication and key agreement procedure (USIM)

The mechanism achieves mutual authentication by the user and the network showing knowledge of a long-term pre-shared secret key Ki which is shared between and available only to the USIM and the AuC (Authentication Centre) located in the user's HE (home environment). The USIM and HE also keep track of sequence number counters that must be within a range, to avoid replay attacks. When the mobile equipment (ME) boots it asks the USIM for the IMSI and passes this to the network requesting access and authentication. The authentication process is done using the parameters RAND (nonce) and AUTN (authentication token), these parameters are sent to the USIM and if they are acceptable, it produces a response RES which is sent back to the network. The network compares RES with XRES (calculated by the network) if they match the network considers the authentication and key agreement successful. The USIM computes the CK and IK which will be used in the confidentiality and integrity algorithms [1]. This section only described the USIM authentication and key agreement procedure. The SIM procedure is fairly the same but uses, for example, different algorithms to compute keys.

3.3.3 Attacks

Smart cards can be attacked in a number of ways for example, timing attacks, power analysis attacks and probing attacks. The most crucial attack against a SIM card would be to somehow retrieve the shared secret key.

3.3.3.1 COMP128 attack (GSM)

This attack deals with the secret COMP128 algorithm which is/was used in GSM Networks for authentication

purposes. COMP128 is a keyed hash function. The algorithm has been derived from some leaked pages of the secret standard and with help of reverse engineering.

The attack considers the two algorithms:

- A3: Algorithm used to authenticate the MS to the network.
- A8: Algorithm used to generate the encryption key [2].

The attack makes it possible to extract the secret 128-bit key Ki. Because the whole GSM-specification (not the crypto algorithms) is publicly available, it is possible to clone the SIM card, and use it to spy on the owner or make free calls. In 1998 the first version of this attack needed an average of 150,000 challenges (took 8 hours) to extract the key [4]. The GSM responded to this by making a new algorithm, COMP128-2 (also secret). But some providers stayed with COMP128-1and just set a maximum on queries that the SIM card would accept (lower than 150 000). Two years later a group of IBM researchers and the Swiss Federal Institute of Technology published a paper called Partitioning Attack [5] where they extracted the secret key by sending just 8 challenges (!) to the SIM. This method was based on DPA (differential power analysis). Both these methods required physical access to the card but the second method could copy a SIM card in just 2 seconds! A big factor to why these attacks were made possible was the GSM security design process: it was conducted in secrecy. COMP128-2 and COMP128-3 are secret algorithms and COMP128-4 is based on a public standard, AES. The COMP128 attacks show that security by obscurity simply does not work.

3.4 Java Card

The Java Card platform was designed and developed from the beginning to enhance the security of smart cards, and the security issue is still the most important one [7]. The Java Card technology combines a subset of the Java programming language with a runtime environment optimized for smart cards and similar small-memory embedded devices [6]. It gives the user the opportunity to program the devices and make them application specific. USIM/SIM cards and ATM cards are often based on Java Card [7].

Developers can build and test programs using standard software development tools and environments, then convert them into a form that can be installed onto a Java Card technology-enabled device. Application software for the Java Card platform is called an applet.

In Java Card objects are stored in persistence memory since RAM is very scarce on smart cards.

The current Java Card version is 3.0 [6]. It is separated in two different editions: Java Card Classic Edition and Java Card Connected Edition.

3.4.1 Security and portability

Another important feature of the Java Card technology is portability. As in Java, the Java Card applet runs in a virtual machine (Java Card VM). This combined with a well defined runtime library means that a Java Card applet can run on many different vendor specific smart cards (USIM/SIM cards), independently of the underlying operating system of the smart card.

As stated before, security is the main objective in Java Card. Security features in Java Card:

- Data encapsulation. Java Card applets are executed inside the Java Card VM and the specific per applet data is isolated from the underlying OS and hardware.
- Applet firewall. The applet firewall prevents the objects that were created by one applet from being used by another applet. But applications can share objects witch each other using Shared Interface Objects (SIO), which configures the applet firewall to allow this.
- Cryptography. Commonly used encryption algorithms like: AES (CBC, ECB), DES (CBC, ECB), KOREAN SEED (CBC, ECB), Elliptic curves and RSA are supported. Also other services like signing, key generation and key exchange are supported.
- Atomicity and transaction. A Java Card guarantees the integrity of the data. Either the whole operation is performed or not at all [8][6][7][9].

3.4.2 Classic Edition

The Java Card 3.0 Classic Edition is based on an evolution of the Java Card Platform 2.2.2 and is backward compatible with. It contains all the features described in the previous sections. It solely supports applet-based applications [6].

3.4.3 Connected Edition

The Java Card 3.0 Connected Edition features an enhanced runtime environment and a new virtual machine. It includes several new features:

- Extended applets (Figure 3). Similar to classic applets but can use all the new APIs like java.lang, java.util, GCF (Generic Connection Framework) and Threads (multithreading not possible in 2.2.2).

- Servlet applications (Figure 3). Based on Servlet 2.5 API. Communicate using standard HTTP/HTTPS protocol.
- Automatic garbage collection

Figure 3 shows the architecture of Java Card 3.0 Connected Edition.

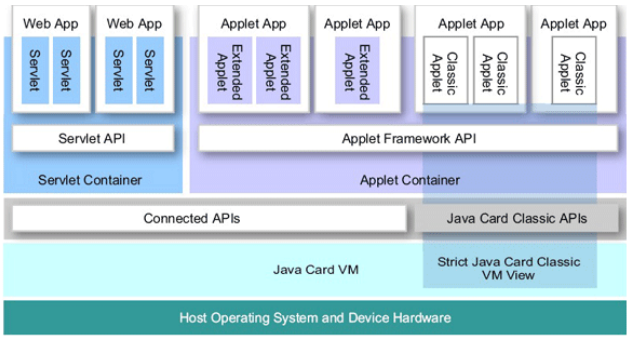

Figure 3. Java Card 3.0 (Connected Edition) architecture [7].

The servlet API supports HTTP/HTTPS which means support for HTML, REST, SOAP. This means that you can run your own web server on SIM card!

The Connected Edition introduces a lot of nice features which could enhance the security of a Java Card SIM card but also weaken it.

Advantages:

- Secure local-storage (like keys, user data).
- Secure remote application management
- Threading, new APIs more advanced SIM applications.
- Garbage collection, memory leakage could threaten the availability.

Disadvantages:

- Threading, could introduce security risks.
- Increased connectivity, introduces many more attack opportunities.

Java Card 3 needs high-end smart cards and cannot be run on the SIM cards used today. It needs approximately 24 Kb of volatile and 128 Kb of non-volatile memory and a fast 32-bit processor [7].

4. Conclusion

A smartphone is a mobile phone with advanced capabilities. Since smartphones are used more and more in critical business and everyday life applications, they require more built-in security mechanisms. In this report we have investigated several methods to make a smart phone a trusted platform.

MTM is a general concept that enables trusted computing either by hardware-based solutions or by software-based solutions. The specification has been available since 2007 but no commercial implementations exist today. We think there are several reasons for this; the market might not be ready for or understand the need for trusted computing in smartphones. Implementing the MTM might not be as profitable to implement as some other features. Other similar technologies might be simpler to implement, for example TrustZone.

TrustZone is a built-in technology in some specific models of the ARM processor. TrustZone enables the applications core to switch between two states, the "normal world" and the "secure world". TrustZone can for example be found in iPhone, but we have not found any information about the extent it is used.

The report also covers Java Card which from the beginning was designed and developed to enhance the security of smart cards, for example SIM/USIM cards. The Java Card technology combines a subset of the Java programming language with a runtime environment optimized for smart cards and similar small-memory embedded devices. The introduction of Java Card 3 proposes new and exciting functionalities of SIM/USIM and ordinary smartcards. Unfortunately we do not think the new features of Java Card 3 will be seen in SIM/USIM cards, due to the increased security risks it causes.

We have not been able to find that many attacks on the technologies we have described. The reason for this is probably because they are new and in some cases not yet implemented in commercial applications.

The mobile security market will grow rapidly in the coming years and is expected to hit $4 billion in 2014. We think the mobile/smartphone security market has a bright future and that we will see many new and exciting technologies.

References

[1] 3GPP Specification. 3GPP TS 31.102 Characteristics of the Universal Subscriber Identity Module (USIM) application. http://pda.etsi.org/exchangefolder/ts_131102v080800p.pdf. Accessed 2010-03-15. (Requires login)

[2] 3GPP Specification. 3GPP TS 11.11. Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) Interface. http://pda.etsi.org/exchangefolder/ts_100977v081400p.pdf. Accessed 2010-03-15. (Requires login)

[3] Java Cards Technology. Oracle/Sun Developer Network. http://java.sun.com/javacard. Accessed 2010-03-15.

[4] Marc Briceno, Ian Goldberg, and David Wagner: GSM Cloning, http://www.isaac.cs.berkeley.edu/isaac/gsm-faq.html Accessed 2010-03-11.
[5] Josyula R. Rao, Pankaj Rohatgi Helmut Scherzer and Stephane Tinguely. Partitioning Attacks: Or How to Rapidly Clone Some GSM Cards (2002). http://www.research.ibm.com/intsec/gsm.ps. Accessed 2010-03-19.
[6] Java Card 3.0.1 Platform Specification. Oracle/Sun Developer Network. http://java.sun.com/javacard/3.0.1/specs.jsp. Accessed 2010-03-20.
[7] Java Card 3: Classic Functionality Gets a Connectivity Boost. Oracle/Sun Developer Network. http://java.sun.com/developer/technicalArticles/javacard/javacard3/. Accessed 2010-03-20.
[8] Wikipedia. Search word: Java Card. http://en.wikipedia.org/wiki/Java_Card. Accessed 2010-03-20.
[9] Java Card 2.2.2 Platform Specification. Oracle/Sun Developer Network. http://java.sun.com/javacard/specs.html. Accessed 2010-03-20.
[10] ARM TrustZone® technology. Arm. http://www.arm.com/products/processors/technologies/trustzone.php?tab=Hardware+Architecture. Accessed 2010-04-10.
[11] Wikipedia. Search word: ARM Architecture. http://en.wikipedia.org/wiki/ARM_architecture. Accessed 2010-04-10.
[12] TDDD17 Information Security (Linköping University course). Lecture: System security 3: Trusted computing. Ben Smeets. Ericsson Research Lund / Lund Universitet.
[13] ARM TrustZone Security Whitepaper. ARM. http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf. Accessed 2010-04-16.
[14] Liqun Chen (Editor), Yi Mu (Editor), Willy Susilo (Editor). Springer; 1 edition. Information Security Practice and Experience: 4th International Conference (2008). ISBN: 3540791035
[15] Trusted Computing Group, Mobile Phone Work Group Use Case Scenarios, Specification Version 2.7, 2005. http://www.trustedcomputinggroup.org/files/temp/6443B207-1D09-3519-AD3180491A6DF1F5/MPWG%20Selected_Mobile_Phone_Use_Case_Analyses_v1.pdf. Accessed 2010-03-11
[16] Trusted Computing Group, TCG Mobile Trusted Module Specification, Version 1.0 Revision 6, June 2008 http://www.trustedcomputinggroup.org/files/resource_files/87852F33-1D09-3519-AD0C0F141CC6B10D/Revision_6-tcg-mobile-trusted-module-1_0.pdf. Accessed 2010-03-11.
[17] Trusted Computing Group, TCG Mobile Reference Architecture, Specification Version 1.0, 2007. http://www.trustedcomputinggroup.org/files/temp/644597BE-1D09-3519-AD5ADDAFA0B539D2/MPWG%20tcg-mobile-reference-architecture-1.pdf. Accessed 2010-03-11.
[18] Trusted Computing Group, Mobile Trusted Module Specification Technical Overview FAQ http://www.trustedcomputinggroup.org/files/resource_files/879DB14E-1D09-3519-AD3FC64EEB271FEF/MTM_Specification_Technical_FAQ_062007.pdf. Accessed 2010-03-28.
[19] Nokia Research Center, MTM Emulator http://mtm.nrsec.com/. Accessed 2010-04-10.
[20] Mobile Security Market to Exceed $4 Billion by 2014. ABI Research. http://www.abiresearch.com/press/1583-Mobile+Security+Market+to+Exceed+$4+Billion+by+2014. Accessed 2010-04-18.
[21] Smartphone adoption to create new revenue opportunities for m-security market? http://www.infogrok.com/index.php/prediction-technology/smartphone-adoption-to-create-new-revenue-opportunities-for-m-security-market.html. Accessed 2010-04-18.
[22] Security Chip That Does Encryption in PCs Hacked http://abcnews.go.com/Technology/wirestory?id=9780148&page=1. Accessed 2010-04-16.