Analyzing trusted elements in mobile devices

Kulkarni, S.N.

Award date:
2015

Technische Universiteit
Eindhoven
University of Technology

Department of Mathematics and Computer Science

Analyzing Trusted Elements in Mobile Devices

Master Thesis

Saurabh Kulkarni

Supervisor:
Dr ir L.A.M. (Berry) Schoenmakers

Eindhoven, November 2015

Abstract

Since last two decades, we have witnessed a significant trend from PC to mobile devices. The primary focus of this shift on mobile devices is making a device personal to the user, but, unfortunately, neglecting the trustworthiness of mobile devices. Mobile devices consist of many elements in hardware, software (firmware) and a combination of both. Some of these elements on mobile devices are trusted, while some untrusted. The problem with mobile devices is that both trusted and untrusted elements are executed at the same time to protect user's sensitive information (e.g., private keys, biometric data). So the question, Are the today's mobile devices trusted? And how much security is provided to users using these trusted elements, remains unanswered.
In this thesis, we define trusted element in the mobile devices and analyse the amount of trust these elements provide. Trusted element is a notion that provides security assurances as well as trust to the mobile devices. For instance, hardware-based elements like Trusted Platform Module (TPM) that is capable of supporting Trusted Execution Environment (TEE) provides a substantial amount of assistance for providing trust in mobile devices. Hence, we study various elements on different platforms to analyse the amount of security provided by these trusted elements.

Acknowledgement

This project would not have been possible without the support and guidance of my supervisors, colleagues, friends and family. I would like to extend my sincere gratitude towards all of them.

First of all, I offer my most heartfelt gratitude to the supervisor of this thesis, dr.ir. L.A.M. (Berry) Schoenmakers, who supported me with great patience, trust, flexibility and significant insights throughout this thesis and beyond and for whom working has been a demanding, educational and yet pleasant experience. His critical evaluation of my work kept me on my toes and helped me improve myself. My parents always reminded me the meaning of education that is *the process of discovering yourself.* During this research project, I was able to discover myself in the sense that gave me the more clarity to deal with new things. I am not sure whether all graduate students are given such opportunity to develop their individuality, self-sufficiency and passion for new learnings by being allowed to discover with such independence. For everything youve done for me, Prof. Berry, I thank you.

I would like to thank the entire Department of Information Security at TU/Eindhoven for guiding me during my masters program. Also, special gratitude to the members of my examination committee (dr. Jerry den Hartog and dr. Benne de Weger) for their input, valuable discussions and accessibility.

Lastly, I would like to thank all my friends in the Netherlands and India. Lunch, coffee breaks and conversations with my fellow students, friends were the best recreation I could have imagined during the intense and exhausting phase of reading the TPM, UEFI and other specifications and helped me greatly to stay focused and motivated.

Last but not least, I owe this project to my beloved mother (Mrs. Vidya Kulkarni) and my passionate father (Mr. Nitin Kulkarni) for their priceless encouragement and support. My parents raised me with a love of challenging myself and patiently encouraged me to continue in times when I felt low.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Technology advancements in mobile devices have emerged from the Personal Data Assistants (PDAs) to the ubiquitous and multifunctional smartphones [20]. Unfortunately, these advances in the mobile devices have been driven by market demand, mainly focusing on new features, applications but neglecting security perspective. As a result, mobile devices (especially smartphones) now face new security problems that are not found anywhere else [20].
There are also many types of research performed focusing on improving security in the mobile-devices. Due to complicated software architectures and dedicated hardware components, many of the solutions have merely resided in theory. Apart from those, genuinely interesting applications, software is introduced to secure mobile devices and also to increase its usability.
The core problem in mobile devices lies in providing trust to the users. Users expect that mobile devices should only behave the way its device owner wants it. But, the questions related to it, How much trust is provided by the mobile devices? And the amount of security provided to users using these trusted elements, remains unanswered. Therefore, the term "Trusted Computing" emerged to reshape the simple context of "Trust" in the domain of IT Security. The Trusted Computing Group (TCG) describes "Technical Trust" as: "An entity can be trusted if it always behaves in an expected manner for the intended purpose" [15]. When we break down the definition of technical trust further, we can identify certain functionality, so called Roots of Trust (RoTs), which typically consist of some minimal hardware and software that supports enabling of particular security feature [15]. But, for a trusted mobile device, it is just a foundation of security.

In this thesis, we define our notion of "trusted elements", to clarify it's meaning and applications on various platforms. Our idea of trusted element focuses only on mobile devices, and it aims to emphasise on various security platforms (that includes hardware, software and combination of both). In this thesis, one of our research questions is to study various trusted elements in mobile devices and to analyse how secure are they?

1.1 Motivation

The motivation behind this project is a need for transparency in the working of the mobile devices. Mobile devices are considered as a personal device with lots of facilities inside it. Our aim is to define a crystal clear line between personal data and organisational data. Hence, this project analyses various trusted elements in the mobile devices on different platforms.
Figure 1.1 explains the separation of personal data and organisational data. The line that separates the two types of data using trusted elements. As discussed in the previous section, RoTs is an assurance of trust which provides the foundation for security. Therefore, in this thesis we have analysed that trusted elements could be implemented on various platforms. For instance, "Trusted Platform Module" is a hardware-based trusted element that has a security foundation of RoTs. Implementation of Trusted Platform Module is also done on Firmware-based platform

Figure 1.1: Analyzing Trusted Elements

with more features and advancements.

1.2 Research Questions

The main research of this thesis is to analyse various trusted elements on different platforms. Furthermore, we define the notion of trusted elements to clarify the concept and it's requirement. Then, we analyse the trusted elements on the hardware platform (hardware-TPM, SIM Cards) and also we analyse firmware-based TPM in mobile devices using ARMs TrustZone technology. We also analyse biometric authentication system that is a combination of hardware and software platforms. For the practical implementation, we then analyse FIDO's UAF protocol that is web-based password less authentication system.

Therefore, the main research questions of this thesis include: Are the today's mobile devices trusted?. And If they are trusted, How much security is provided to users using these trusted elements?

To answer above mentioned main research questions, following points provide a concrete idea of overall research:

- Defining a notion of trusted element
- Analysing trusted elements on various platforms
- Practical implementations that use trusted elements
- Analysing how much security is provided using trusted elements

1.3 Outline of the thesis

The entire thesis is structured as follows:

Chapter 2: Defines trusted element and analyses the need for it. Moreover, the illustrations of trusted elements are briefly explained.

Chapter 3: Discusses the technical description about the Trusted Platform Module (TPM) in mobile devices. This chapter also provides the detailed explanation of TPM on firmware platform. Lastly, it also highlights on the applications of TPM called as "Roots of Trust (RoTs)".

Chapter 4: This chapter provides detailed explanation about the SIM cards and its limitations. It also elaborates, Universal Integrated Circuit Chip (UICC), and highlights on UICC as a trusted element.

Chapter 5: This chapter provides the comprehensive study about biometrics authentication system in mobile devices. Lastly, it provides description on vulnerabilities and countermeasures of biometric authentication.

Chapter 6: Explains the practical application using trusted elements. It also provides a step-by-step protocol architecture followed in this application.

Chapter 7: In the final chapter, we conclude our thesis with our overall analysis. This chapter provides the answers to all the research questions mentioned above.

Chapter 2

Trusted Elements

This chapter firstly defines trusted elements in mobile devices and the need towards it. Then, we briefly discuss few illustrations of trusted elements in various layers of mobile-device security.

2.1 Definition

Below, we introduce our notion of trusted elements that we will study in the context of mobile devices:

Definition. We define a **trusted element** as an isolated and/or tamper resistant component of a hardware or software platform, providing certain security guarantees and services.

The similar notion of secure elements (SE) by GlobalPlatform[1] is defined as follows: "A *secure element* (SE) is a tamper-resistant platform (typically a one chip secure microcontroller) capable of securely hosting applications and there confidential and cryptographic data (e.g. key management) in accordance with the rules and security requirements set forth by a set of well-identified trusted authorities."
Our notion of trusted element is intended to cover both secure element (SE) as defined by GlobalPlatform[2], A trusted environment in mobile devices also known as Trusted Execution Environment (TEE) and similar trusted components. There is a fundamental difference between above two definitions mentioned. Our definition is mainly focused on the mobile devices and it aims to emphasise on various security platforms (that includes hardware, software and biometrics-based). GlobalPlatform only focuses on three different form factors of SE: Universal Integrated Circuit Card (UICC), embedded SE and microSD.
Additionally, trusted elements are not only able to host the applications securely, but it also creates a secure communication layer towards the CPU. Unlike trusted elements, these features are not defined by Global Platform. According to Irish [28], Trusted Execution Environment's (TEE) are usually isolated parts of the device processor, while SEs are typically on another hardware chip. Recently on mobile devices, there are various layers of trusted elements.

Moreover, we need methods that unify trusted elements in the mobile devices. All users need mobile devices that can be trusted. RFC 4949 defines the *Trust* is "A feeling of certainty (sometimes based on inconclusive evidence) either (a) that the system will not fail or (b) that the system meets its specifications (i.e., the system does what it claims to do and does not perform unwanted functions)" [44]. Trust is an assurance in the ability of an entity to act dependably, securely, and reliably within a specified context. Hence, various mobile device manufacturers, researchers are finding a solution to create a trusted mobile device that is trusted in personal life or at entrepreneurial level.
In chapter 6, we study FIDO Alliance[3] which represents the application of trusted elements. FIDO

[1] https://www.globalplatform.org/mediaguideSE.asp
[2] https://www.globalplatform.org
[3] https://fidoalliance.org/

Alliance's UAF (Universal Authentication Framework) architecture is responsible for online-based authentication using password-less experience. UAF protocol also allows the application that combine multiple authentication mechanisms such as fingerprint + PIN and similar biometric features.
The following subsections briefly explains few illustrations of trusted elements that are discussed in this research project :

2.2 Illustrations of trusted elements

2.2.1 Trusted Platform Module (TPM)

The **T**rusted **P**latform **M**odule **(TPM)** is specified by the Trusted Computing Group TCG[4] which provides security mechanisms like remote attestation, protected storage areas, computation of cryptographic functions [9]. In simple words, TPM is tamper-resistant cryptographic security module that can provide secure storage for encrypted keys, sensitive information and also it can perform cryptographic operations.
The following are requirements for designing of the TPM:

- Private Keys (Cryptographic) cannot be stolen or given away.
- Inclusion of the malicious code can always be detected.
- Malicious code can be prevented from stealing/changing private keys.
- Encrypted Keys are not physically available to the thief.

There are two major versions of the TPM that are published by TCG i.e. TPM 1.2 specification [24] and TPM 2.0 specification [5]. After the publication of the TPM 1.2 specification, critics claimed that TCG specification takes control of the operation of the computer away from the user, thus leading to privacy issues. Hence, it will provide a shortcut for organisations to force users into the shift from competitive software and building Digital Rights Managements (DRM) into the computing platform.
In Chapter 3, we analyse in detail about implementation of TPM in mobile devices. Nevertheless, it is equally important to understand whether TPM or its security guarantees could be used in the mobile devices. Also, if they could be used then what requirements are needed for mobile devices. Depending on the security platform of a mobile device (like hardware-based or firmware-based) it's requirement changes.

2.2.2 Subscriber Identity Module (SIM) Cards

A Subscriber Identity Module (SIM) card is a smart card that securely stores the subscriber identifier and the associated key used to identify and authenticate to a mobile network. The main goal of SIM cards is to ensure the security and integrity of all kinds of personal data like any other secure element [17]. SIM cards are an advantageous component that is portable, tamper-resistant computer/platform used mainly in telecommunication industries worldwide. Since, SIM cards are also known as smart cards, they conform to ISO-7816-1 [38] standards regarding physical characteristics and electrical interface.
Following are the features of the SIM cards:

- **Portable Platform**: SIM cards are available in different shapes and unlike some of the other hardware based security modules like TPM; they are meant to be portable.
- **Processing Power**: Just like any other hardware, SIM cards can produce different processing capabilities and varying from an 8-bit microcontroller to greater than 32bit architectures.

[4] http://www.trustedcomputinggroup.org/about_tcg

- **Storage and memory**: The memory structure of the SIM cards usually includes RAM (Read Access Memory), ROM (Read Only Memory) and EPROM (Electrical Erasable Programmable Read-only Memory).
- **Multi Purpose Crypto-Processors**: SIM cards are also able to compute various individual and muti- purpose crypto-processors like RSA and rounds of symmetric cryptography.
- **Tamper Resistance**: SIM cards are filled with technology and investment in both hardware and software protection techniques of making it tamper resistance.
- **Secure Execution**: One of the most important features and the reason for the success of SIM cards are their ability to provide isolated, secure execution of applications.

In Chapter 4, we explain the details of SIM cards with its evolutionary version called as Universal Integrated Circuit Cards (UICC) and it's a difference with traditional SIM cards.

2.2.3 Biometrics in Mobile Devices

Biometrics is a method of user authentication that is based on users physical/behavioural characteristics what makes you who you are rather than simply assuming that the user owns the device that they have or carry. Biometrics in mobile devices provides both high security and high convenience. In recent biometrics in mobile devices, instead of typing a string of characters, user can directly read a displayed sequence of strings out loud, look into the camera and blink, or swipe the finger across an embedded sensor.
There are several biometric traits/characteristics that individuals carry that then can be used for identification or authentication purposes within mobile devices. The most common physical biometric characteristics are the eye, face, fingerprints, hand and whereas voice, signature, typing rhythm (keystroke) and gait are the most common behavioural biometric characteristics. But, many of these traits are hard or less secure to perform on the mobile devices. Each biometric is enabled through different, comparison matching algorithms, but it is important to verify whether the biometric is stored in the trusted elements of the mobile devices or not. Each of these algorithms performs differently based on how the thresholds are set, and the performance of different algorithms changes based on the quality of the initial enrolment. For instance, Touch-ID is Apple's fingerprint recognition system that uses "Secure Enclave" for processing, trusted storage and also the comparison of biometric templates in mobile devices. For face recognition, Google smartphones are dependent integrated solutions of biometrics. The report estimates that biometrics on mobile devices will generate about $8.3 billion worth of revenue by 2018 for the biometrics industry, not just for unlocking the device but to approve payments and as part of multi-factor authentication services [1].

In Chapter 5, we elaborated familiar types in biometrics, which are deployed on the mobile devices. And also we have described the comparison between various forms of biometrics based on the security, functionality and practicality.

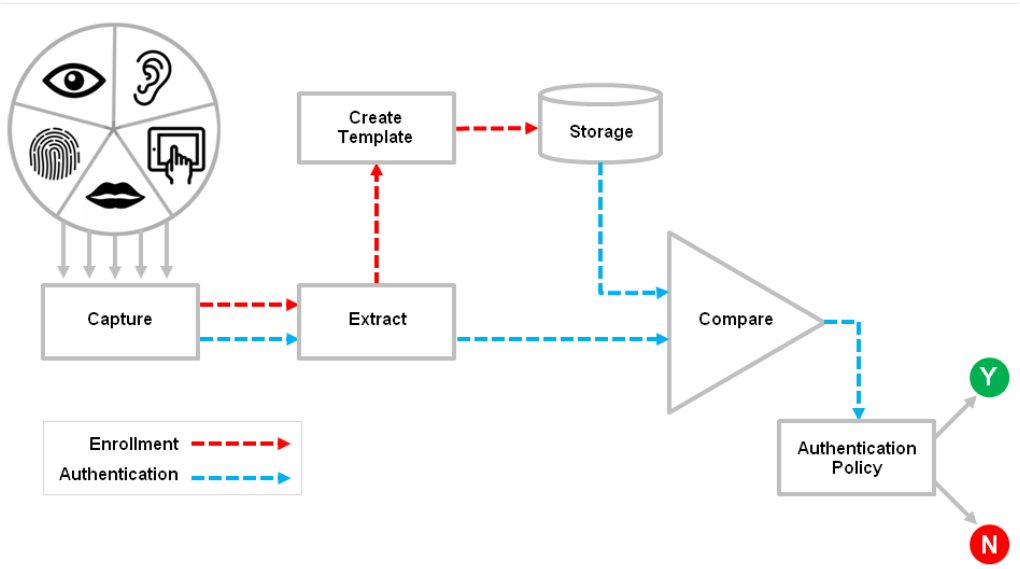

Figure 2.1: Basic Workflow of Biometrics

2.3 Summary

In this chapter, we have defined the term "trusted elements", that only applies to this thesis. Secondly, the need of this definition and all the relevant concepts is explained. Lastly, the brief introduction of few trust concepts are provided. As discussed above, these trusted elements are based on hardware and software or a combination of both. The detailed analysis of each trust concept is presented in the following chapters with relevant practical implementations.

Chapter 3

Mobile Trusted Platform Module

The following chapter provides detailed study regarding the TPM and it's type on different platforms. Firstly, we discuss the introduction to the most fundamental concepts and terminology about TPM, and it's implementation on hardware and firmware platforms. Secondly, we emphasise on the real-life applications (e.g., Dongle-based TPM) and lastly, we analyse various forms of Roots of Trust (RoTs). The primary goal of this chapter is to analyse the need of TPM in the mobile-devices and its practical implementations in today's mobile devices.

3.1 Trusted Computing in a Nutshell

The notion of Trusted Computing (TC) usually is used for a set of technologies specified by the Trusted Computing Group (TCG)[1]. TCG, the successor organisation of the Trusted Computing Platform Alliance (TCPA), is an industry consortium proclaiming itself an international standards group. It was founded in 2003 and incorporates many of the big players in hardware and software development, such as Microsoft, AMD, Intel, IBM and Cisco to name just a few [42]. According to TCG, "a not-for-profit organisation formed to develop, define and promote open, vendor-neutral, global industry standards [. . .] for interoperable, trusted computing platforms[2].

In Chapter 2 of trusted elements, we studied that TPM 1.2 and 2.0 specification mainly enforced on the Personal Computers (PC) due to the need of BIOS. But, there is also a need for a similar TPM hardware with trusted execution in mobile devices. So, TCG has published a specification about mobile-based TPM formerly known as Mobile Trusted Module (MTM) [8, 5]. We often see that systematised Personal Computers provides TPM security capabilities for a broad range of cases and applications. One of our research questions is to analyse if TPM could be implemented in mobile devices also?

Mobile TPM (MTPM) can be deployed on the mobile devices using two separate methods. Firstly, by using Hardware Based TPM, where we implement MTPM using a system-on-a-chip on the mobile devices. Secondly, implementation of MTPM using Firmware based method that is supported by ARMs Trustzone[3]. In the following sub-sections, we explain the need and implementations of the discussed two methods:

[1] http://www.trustedcomputinggroup.org
[2] http://www.trustedcomputinggroup.org/about_tcg
[3] http://www.arm.com/products/processors/technologies/trustzone/index.php

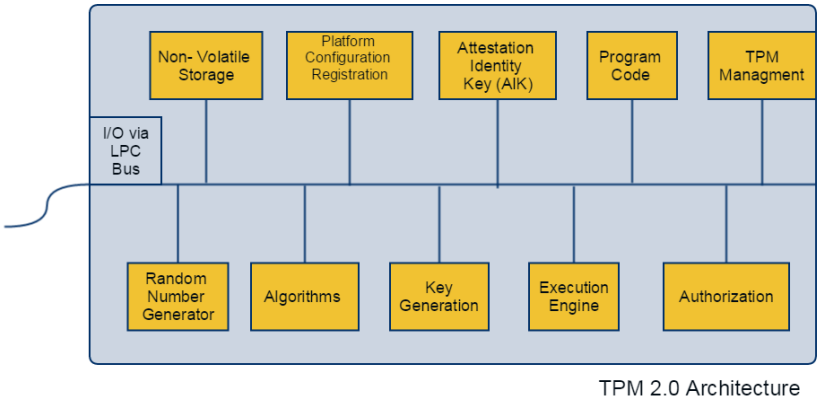

Figure 3.1: TPM 2.0 Architectural Diagram

3.2 Hardware Based Mobile-TPM

Hardware based MTPM, which is based on a traditional hardware chip of TPM. The MTPM includes a dedicated processor and storage resources that are physically distinct from the other resources of the device [36]. Preventing physical attacks against tampering is helpful. NIST proposed that mobile devices contain Roots of Trust (RoTs) security to provide a set of trusted functions and that "hardware RoTs are preferred over software RoTs due to their immutability, smaller attack surface and more predictable behaviour" [11].
In order to implement hardware-based MTPM, NIST considers following goals [11]:

- **Device Integrity**: Device Integrity is the assurance of the lack of corruption with the device's hardware, software, firmware platform. Device integrity is the necessary capability needed for the mobile device to become trusted. Device Integrity of the mobile device is a combination of RoTs with an established set of trusted components, extended into chains of trust and trusted agents. Any mobile device should be able to prove the evidence of device integrity to a third party using the form of device attestation [36] Device integrity is mainly used to detect attacks that are performed to deter the integrity of mobile devices (e.g., Jailbreaking, OS Root access).

- **Device Isolation**: Device Isolation means separation of processes from one another as well as separation of different data components from one another. In technical terms, restricting the flow of information from one entity to another by any means [43]. For instance, in Figure 1.1, the user would not like to want the organisation to be able to read and change the personal data and vice-versa on the mobile device. Threats that are considered in isolation of device are the data on mobile that can face malware ex-filtering sensitive data to the device, which leads to the loss of confidentiality.

- **Protected Storage**: TPM can store one or multiple Storage Root Keys (SRKs) that can only be used inside the shielded locations. The SRKs or keys derived from an SRK can be used to encrypt data to be stored outside of the TPM, such as hard disk. The only way to decrypt this data is to load it into the TPM since the key material used for encryption can only be used there. The major advantage of protected storage is that the shielded locations of the TPM are elongated so to verbalize providing means to encrypt bulk data.

The security parameters explained above are to be fulfilled with its security components. A particular RoTs or combination of RoTs can be implemented in various hardware and software (firmware) platforms, but it is important that these components are best in a group. In Section 3.5,

we will discuss RoTs with more details and analyse additional components of RoTs. Furthermore, it's important that all RoTs are implemented in a trusted hardware, software/firmware, etc. Following RoTs are the list of imperative features of TPM as explained by [11]:

- **Roots of Trust (RoTs)**: RoTs are the assurance of the trustworthiness to the mobile devices. NIST [11] suggests that roots of trust should also apply for integrity, isolation, and protected storage. Following are the additional components of trust explained by NIST:
  - **Roots of Trust for Verification** (RTV): RTV provides a protected interface and engine to verify digital signatures associated with software/firmware and create assertions based on the result [11].
  - **Roots of Trust for Integrity** (RTI): RTI provides protected storage, integrity protection, and a protected interface to store and manage assertions [11]. Section 3.5 gives a detailed explanation of this feature.

- **Application Programming Interface (API) for RoT's**: Features and capabilities that are provided by RoTs are required to be presented to the user in a better environment. Therefore, RoTs need to provide features by the operating system to the applications using Application Programming Interface (API) [11]. API provides security developers an open platform to choose various security features. API for RoTs would help developers protecting the data using different security, access control capabilities. Additionally, security developers do not need to program at low-level which is more error-prone to provide security features. The use of API's is simple and efficient for applying security capabilities to the applications. API also encourages the use of security features and concepts within the developers.

- **Policy Enforcement Engine**: There have to be varied secure mechanisms for the Device Owner to manage different policy agreements that would be required to establish access with multiple Information Owners [11]. Policy Enforcement Engine enforces policies on the device with the avail of other device components and enables the processing and management of policies on both the device and in the Information Owners environments. This mechanism also provides Information Owners, ability to express the control they require over their information. Furthermore, Policy Enforcement Engine interacts with all of the Information Owners and translates the desired requisites for storing and sharing their information into the appropriate device, network configurations and policies [11].
  According to [11], Policy Enforcement Engine needs to be trusted element to implement the Information Owner's requirements correctly and to prevent one Information Owner's requirements from adversely affecting another Information Owner [11].

3.2.1 Implementation of Hardware-Based Mobile-TPM

As described in above subsection, the goal of hardware based MTPM is to implement TPM 2.0 specification in the mobile devices. Also, the main challenge in this implementation is to provide efficiently all the security capabilities to the mobile devices. So, there is a need for precise specification of the requirements that are necessary for its implementation. Researchers like TCG[4], NIST have explained some of the necessary requirements needed for the implementation [11].

The TCG specification does not describe implementation details. Instead, TPM 2.0 specification by design defines the characteristics of a TPM in a mobile architecture, not how to implement a TPM on a particular mobile device. The implementation details and decisions related to MTPM are left to a TPM manufacturer.
Boot Sequence of the device is paused if a non-approved software being executed is detected. For instance, if Operating System is jailbroken or has detected root access, then the TPM secures the system by halting the boot process to prevent integrity of the system.

[4] http://www.trustedcomputinggroup.org/

3.2.2 Limitations of Hardware Based Mobile-TPM

As explained in the previous section, Hardware Based MTPM was designed for better security, and even also it's theoretical contribution in making mobile devices is appreciated by researchers and mobile vendors. But, there are also some limitations that make the implementation questionable. Following are few major limitations of Hardware Based MTPM:

- Implementation of Hardware Based MPM in mobile devices like smartphones needs a dedicated hardware that can make the device weigh much more than the normal mobile device.
- According to [47], for embedding a traditional chip inside a motherboard for any computing the system increases the Bill Of Materials (BOM) cost by around 1$ to 2$ per system.
- Another common problem faced with hardware-based MTPM is regarding the computational power of the system especially with the mobile devices that are resources constrained. Therefore, embedding costlier MTPM chips and decreasing computational power would eventually lead to the exclusion of the devices in the manufacturing process [47].

3.3 Firmware Based Mobile-TPM

Firmware Based MTPM is an another implementation option that based on Trusted Execution Environments (TEE) [47]. In this implementation, isolation of software (firmware) is enforced by the hardware like processors that are already embedded in the mobile devices. This mechanism also enables processing of sensitive data outside the main operating environment as well as isolated system memory. Hardware architecture of the mobile device provides a communication of RoTs to the main processor. For instance, if the Firmware-TPM uses ARMs TrustZone then it helps to split system-on-chip in a Secure world or Normal World.

3.3.1 Features of Firmware-TPM

According to [47], ARMs TrustZone facilitates the implementation of a RoTs, making it highly secure against software attacks. After the instantiation of Firmware-TPM, it then uses an existing ARM-based architectures and extensions related to it.

The need of dividing the operating system is that hardware restricts the access to secure world resources such as ROM, RAM, and non-volatile storage. If the use of "Secure World" is active, the software's executing on CPU has a different view as compared to the "Normal World" OS. Following sub-section provides more details about the features and implementation of Firmware-Based MTPM.

As discussed above, firmware-based implementation is a combination of hardware, software and also some extensions to support the applications. In conceptual Firmware-TPM, the hardware of a system is responsible for making the partitions in "Secure World" and "Normal World". Various software in the systems uses the extensions for designing, which are then executed for providing Isolated Execution Environment (IEE). But importantly, conceptual Firmware-TPM only provides "Isolated" execution environment but not "Trusted" execution environment. Thus, the RoTs is used for providing a way to establish trust in the execution environment. Hence, only an isolated execution environment equipped with a RoTs is a real Trusted Execution Environment (TEE) [52]. Therefore, conceptual Firmware-TPM also is less likely to be implemented. Eventually, manufacturers or mobile vendors can use their version of Firmware-TPM using Trusted Execution Environment (TEE) to make it a trusted device (e.g., ARMs TrustZone with Firmware-TPM). The following figure explains the architecture of Firmware-TPM using ARMs TrustZone:

3.3.2 Architecture of Firmware-TPM using ARM's TrustZone

The architecture consists of hardware components, software mechanisms that rely on this hardware using the help of some specific extensions. Similarly, there are two Operating Systems which

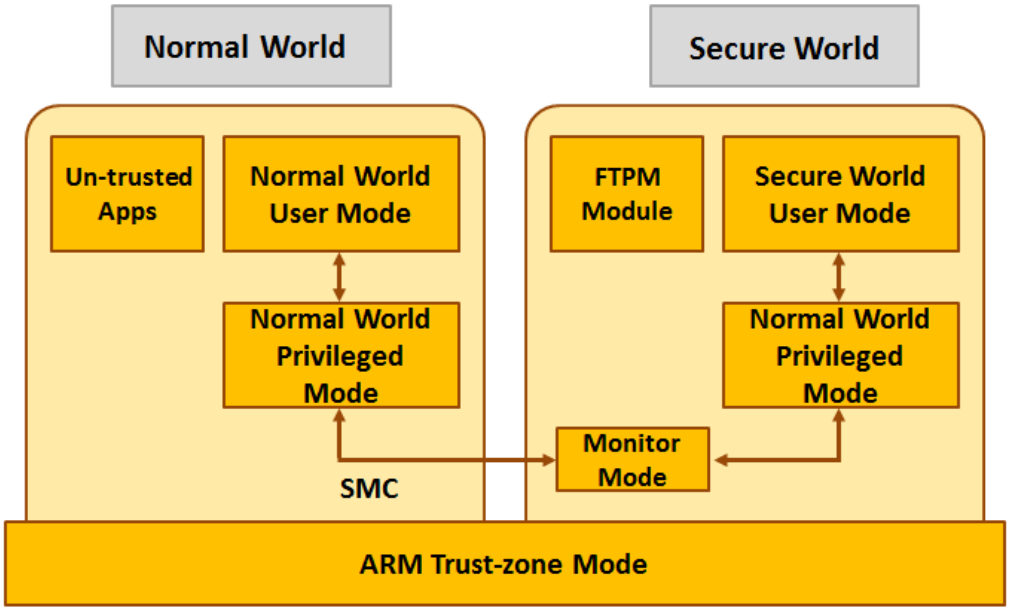

Figure 3.2: Architecture of Firmware-TPM using ARM's TrustZone

execute in this system, as shown in Figure 3.3. One of the major components of this Operating System also called as "Secure World", where the goal is to keep the secure world small and only capable of performing security relevant tasks. Consequently, the more complicated the Secure World is, the more likely it could contain software security flaws that can be utilized to compromise the system. The "Secure World" is considered as part of the Trusted Computing Base (TCB) of the system.

On the other hand, the other component of this Operating System is called as "Normal World" also known as "Rich OS". This component is responsible for implementing all sorts of constraints the user faces using the system. These components may include Graphical User Interface (GUI), Network Interface Card (NIC), Graphics Card for high-end applications. The normal world Operating System can also be considered as an Operating System, which is mass-produced such as Windows, Linux, OS X or Android. The user is only given permission to work with the Normal World OS, which is not part of the TCB. Firmware-Based TPM is installed in FTPM Module Interface of the "Secure World" during system initialization, before the ARM processor switches to "Normal World" operating mode, thereby enabling the use of ARM's Module TrustZone extensions [47].

Switching the mode of operation between two worlds, i.e., Secure world and Normal World is done using the mechanism called as "Secure Monitor Call" (SMC). This mechanism is responsible for passing the instructions from Secure World to Normal World and vice versa. The Monitor Mode software is responsible for providing a robust gatekeeper that manages the switches between the Secure and Non-secure processor states [2]. In most designs, it's functionality will be similar to a traditional Operating System context switch, ensuring that state of the world that the processor is leaving is safely saved, and the state of the world the processor is switching to is correctly restored [2].

3.3.3 Security Features and Boot Sequence

The aim of the ARMs TrustZone is to enable a device from robust security solution. F-TPM also ensures that it can preserve integrity and confidentiality of its code and data from all other

software running on the system. But, only the apps that deal with the security of the mobile device or the OS are executed in the Secure World.

The following Figure 3.3 shows the high-level description of the Boot Sequence within a mobile device of Firmware-TPM using ARMs TrustZone.

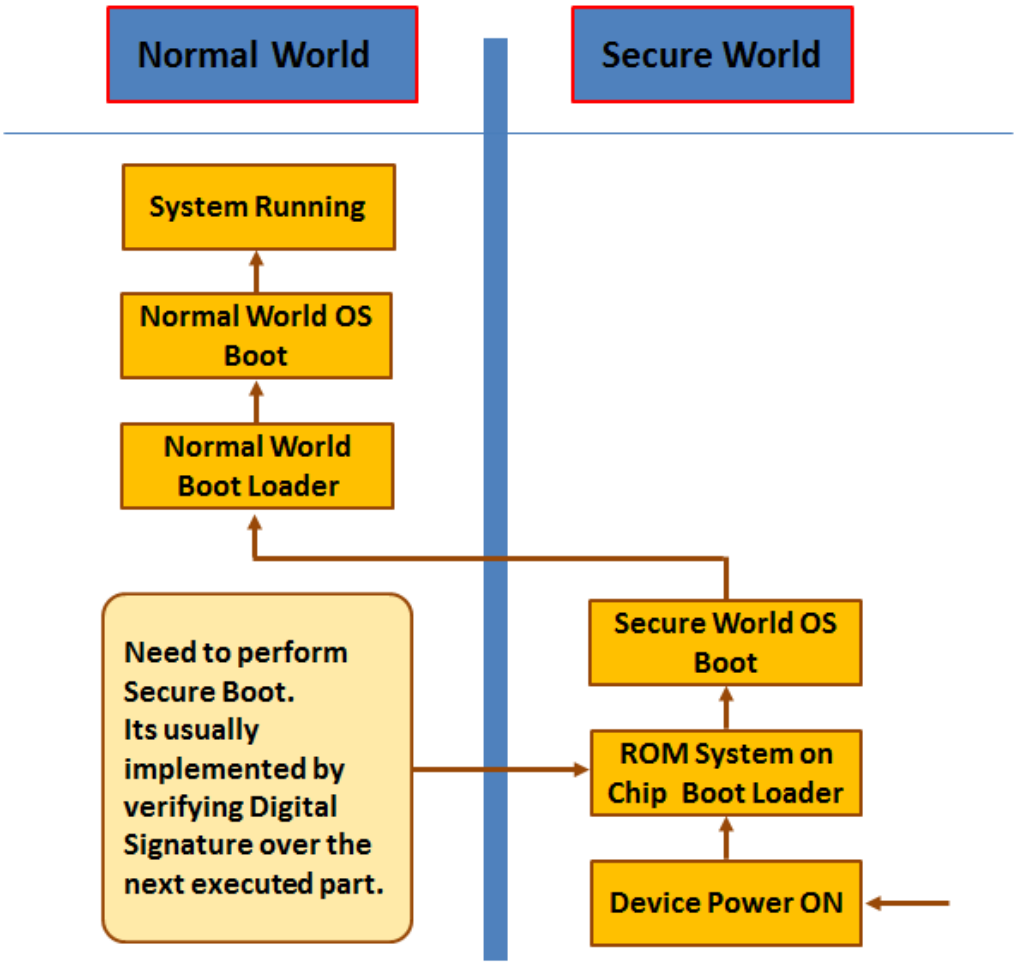

Figure 3.3: Boot Sequence of Firmware-TPM using ARM's TrustZone

- **Stage 1**: During First Stage, the user just needs to Power ON the device. It would help the user to make the secure boot.
- **Stage 2**: After the device is Powered ON, the bootloader gets loaded using Read Only Memory (ROM) and also helps in making the secure boot. In this stage, Bootloader is the ROM system-on-chip bootloader where it tries to load the secure world into the mobile device from any device component. For instance, Network Interface Card (NIC), memory card, flash memory.
- **Stage 3**: In this stage, Secure World is formed and also the device manufacturer digitally signs it before entering the market. In the second stage, bootloader then verifies this signature.
- **Stage 4**: In the Normal World, verification of the signature is done using the public key of the mobile device that is usually stored in the ROM's on the SoC. helps to ensure the integrity of the mobile device.

- **Stage 5**: After the verification of the keys in ROM then the Normal World OS is booted using the Secure World in which the normal world bootloader is executed in the normal world itself. This stage is only is executed when the Secure World verifies all the keys and boots up completely.

As discussed above, this process is explained in the abstract and only taken the security perspective of this boot sequence. This process may vary from manufacturer to manufacturer, as some would add or modify the some security features or might implement this process differently.

3.3.4 Advantages over Hardware-Based Mobile-TPM

There are some fundamental differences between hardware and firmware-TPM that make the firmware MTPM easy to implement. Following are features of the Firmware-TPM make the implementation in mobile devices easier as compared to hardware-based:

1. No need of Dedicated hardware, unlike Hardware-Based mobile TPM that not only reduces the Bill Of Materials (BOM) but also it ensures less overhead of memory and computational performance.

2. Implementation of firmware is done using Trusted Execution Environment (TEE) that provides the trusted component for the device and hence more a mobile device is more secure than before.

3. Firmware-TPM also provides a software interface to the security extensions functionality integral to various processors. Consequently, it provides more flexibility as the functionality of Secure World is defined by software.

3.4 Dongle-based TPM

The Pico[5] was developed in 2011. Pico is designed to make all passwords obsolete. The goal of Pico is not to use passwords, but instead the user carries with them a small token (the Pico) that stores the cryptographic authentication keys they need to prove their identity to the services he/she wishes to use [46]. Pico has several advantages. Firstly, the TPM is no longer tied to a single phone, but could be used by all of the users devices concurrently, sharing his sessions between them. However, an external device is an extra thing to carry, as well as more expensive and less ubiquitous and than an internal chip [28].

One interesting aspect of the above devices is that they may be able to provide secure communication between trusted element and user. An implementation might involve the SE/TEE taking direct control of the hardware/screen. Trusted paths are useful in determining whether requests originated from the user or malware [28].

[5] http://mypico.org/

Table 3.1: Security Capabilities of the Pico [46], Table 1.

Security Features	**Explanation**
Memoryless	Users should not have to memorize any secrets
Scalable	Scalable to thousands of apps
Secure	At least as secure as passwords
Additional requirements if token-based:	
Loss-Resistant	If token lost, user can regain access to services
Theft-Resistant	If token stolen, thief cant impersonate user
Benefits promised by Pico in addition to the above:	
(Usability-related)	
Works-For-All	Works for all credentials, not just web passwords
From-Anywhere	The user can authenticate from any client
No-Search	The user doesnt have to select the correct credentials
No-Typing	The user no longer has to type the password
Continuous Authentication	is continuous, not just at session start
(Security-related)	
No-Weak	The user cannot choose a weak password
No-Reuse	The user cannot reuse credentials with different apps
No-Phishing	Phishing (app impersonation) is impossible
No-Eavesdropping Network	eavesdropping is impossible
No-Keylogging	Keylogging is impossible
No-Surfing	Shoulder surfing is impossible
No-Linkage	Different credentials from same user cannot be linked

3.5 Roots of Trust (RoTs)

Roots of Trust (RoTs) are the combination of hardware/software components that are inherently called as trusted. The RoTs provide the assurance of trust between all the components with or within mobile devices in case of the Trusted Computing. As discussed in the previous section, the RoTs are the foundation of any mobile device. Traditional mobile device faces lots of challenges that includes a greater risk of physical attacks, power and space constraints, data protection. Therefore, the design of RoTs must ensure that the mobile devices contain firmware boot protections, secure measurement of firmware and hardware, secure storage, device authentication, application, and data isolation. Following subsections elaborate all these concepts briefly. In short, requirements of RoTs are that they must be trusted by design, compressed in size and self-protected by hardware or software that is implemented in hardware. Following are the types of RoTs that are applied in the mobile devices:

3.5.1 Roots of Trust for Storage (RTS)

According to TCG, RTS is a computing engine capable of maintaining an accurate summary of values of integrity digests and the sequence of digests[6]. In simple words, RTS is a trusted entity that provides confidentiality and integrity for stored information in TPM without interference leakage. It uses RSA encryption to protect data and ensures that data can only be accessed if the platform is in a known state (Sealing).
For instance, let's consider the following example:

Sample Core Question - "Are users Cryptographic keys kept in a Secret vault ? ".
Answer -

- Component of TPM is also called as "Roots of Trust for Storage" (RTS).

[6]http://www.trustedcomputinggroup.org/developers/glossary

- It does not store the cryptographic keys in the secret vault directly, but it uses one key to store and protect the other keys in the secret vault.
- Hence, it's called as Roots of Trust for Storage (RTS).

3.5.2 Roots of Trust for Measurement (RTM)

As discussed in 3.5, Roots of Trust for Measurement (RTM) has been defined by the TCG as "RTM that relies on the piece of RTM code that is neither stored on TPM nor encrypted or sealed using it [43]". Therefore, it is considered as immutable. Firstly, after the mobile device is booted, the BIOS is given control of the mobile device, followed by the bootloader, Operating System loader, and finally the Operating System (e.g., Android, Mac, Windows, etc.).
Following are two ways in which mobile devices capture the integrity measurements -

Static Roots of Trust for Measurement (SRTM)

SRTM takes place at system boot. The first process executed at boot level is called the Core Root of Trust for Measurements (CRTM). Block of BIOS boot will measure the BIOS and send the value (hashed) to the TPM's component location called as Platform Configurations Register (PCR) '0' before executing it[7].

Dynamic Roots of Trust for Measurement (DRTM)

Technically, it creates a secure/clean state, and it will report (provide measurement hashes in PCRs) on a piece of code someone wants to execute (also called as Measured Launched Environment - MLE). Generally, MLE could also be an Operating Systems [43].

3.5.3 Roots of Trust for Reporting (RTR)

All the TPMs have an Endorsement Key (EK) that is a signing key wherein its public key is certified by a Trusted Third Party (TTP) (e.g., TPM manufacturer, mobile vendor) [43]. Additionally, due to privacy reasons, EK only obtains a key certificate from a Certificate Authority (CA) for an Attestation Identity Key (AIK), which TPM is responsible for generating itself [43, 11].
For example, consider the following scenario:
Core Question - "Is this mobile device in a secure state"?
Answer -

- We check that which authority is responsible for knowing the secure state of the mobile device (i.e., RTM)
- Lastly, We check that which authority in TPM would present us the results provided by RTM (i.e., RTR)

Therefore, in this case, TPM is RTR, which presents user the results that provided to it by RTM.

3.5.4 Roots of Trust for Integrity (RTI)

According to NIST specification [11], they have introduced two new components of RoTs to protect mobile devices, namely Roots of Trust for Integrity (RTI) and Roots of Trust for Verification (RTV). RTI provides a limited, compressed protected interface for accessing and modifying storage locations. Together the tamper evident locations and protected interface collectively form the Root of Trust for Integrity [11].

[7] https://security.stackexchange.com/questions/39329/how-does-the-tpm-perform-integrity-measurements-on-a-system

3.5.5 Roots of Trust for Verification (RTV)

As discussed in Section 3.2, RTV is responsible for providing a protected interface and engine for verifying digital signatures associated with software/firmware of the mobile device as well as to create assertions based on those results [11, 43]. Implementation of RTV starts with executing the signature verification algorithm as well as accessing a key store that includes the public key that is needed to verify a signature. This key store may be stored internally in RTV, or it may rely to protect and maintain the key store on RTS.

3.6 Summary

The focus of this chapter was the analysis of the TPM for mobile devices. Moreover, many concepts related to TPM are also mentioned in detail. The main goal of this chapter was to analyse the hardware and software (firmware) based TPM that is also a component of trusted element. Additionally, a practical implementation of TPM in an unusual way is described in Section 3.4.
To the end of this chapter, we explained in detail the required concepts used in the implementation of TPM, called as RoTs. Nevertheless, an important point that hardware TPM only provides an isolation execution environment but whereas firmware TPM adds a trust layer (RoTs) to develop Trusted Execution Environment (TEE).

Chapter 4

Subscriber Identity Module (SIM Cards)

4.1 Introduction

In Section 2.2.2, we studied that "Subscriber Identity Module" (SIM) card is a smart card that securely stores the subscriber identifier and the associated key used to identify and authenticate to a mobile network. An SIM is a removable smart card based on an embedded Integrated Circuit Chip (ICC). In modern mobile phones, the smart card itself is called Universal IC Card (UICC), and SIM is one of the applications running in the UICC. The SIM is a hardware-based module that runs on an Integrated Circuit Card (ICC) type of Smart Card. Important security features of SIM card includes tamper-resistance, efficient, secure and availability on portable the platform. Smart cards that contain a microcontroller chip are sometimes called chip cards to distinguish them from cards that offer either memory storage only, or memory storage and non-programmable logic as they look quite similar from physical characteristics.

In this chapter, we analyse the main components of SIM/UICC from the security perspective. As discussed in Section 2.2.2, the primary goal of SIM cards is to ensure the security and integrity of all kinds of personal data like any other trusted element. Moreover, SIM Cards should securely store confidential data on SIM Cards so that it's prevented from eavesdropping by an attacker due to the usage of cryptographic algorithms and security protocols. Data is stored, protected, and kept secret. The card body that holds the micro-controller, the chip hardware, the operating system and the applications are four components that are responsible for the security of SIM Card.

4.2 Architecture

There are various types of SIM Cards like Java, .Net Version, etc. Current SIM architecture provides a more flexible environment in comparison of previous SIM design for handling application on the SIM. The new design of SIM, enables downloading of the application via OTA (Over-the-Air) and also enables interoperability across card manufacturers for loading of Java-based applets onto the SIM card from any source [50]. The SIM card in the Java-based architecture runs a separate Operating System from the device's Operating System. This isolation between Operating System provides better security for communication between the SIM and the device, and also allows the SIM card to act as a hardware firewall between the mobile device and the information on the SIM memory [50].

Java SIM Cards provide following characteristics to ensure the security to the mobile devices:

1. **Portable Platform**: SIM Cards come in different shapes which also changes it hard-

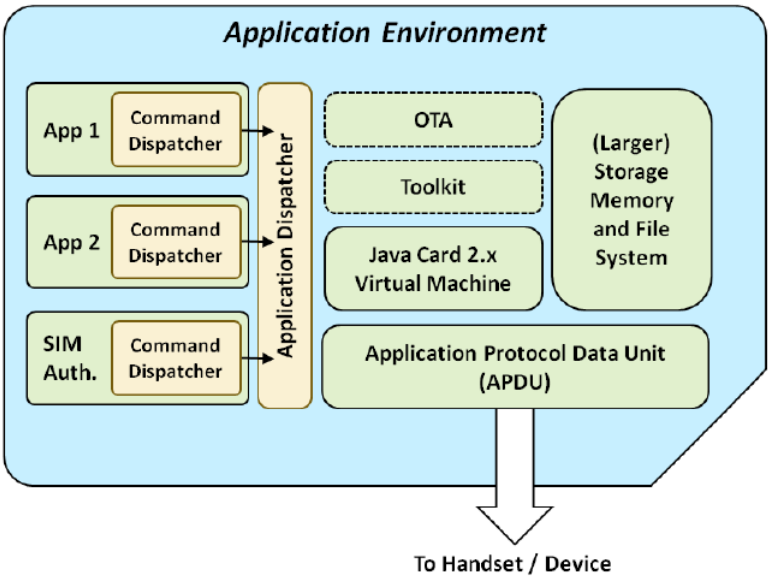

Figure 4.1: Java Card SIM Architecture [50]

ware/software specifications, also SIM Cards are meant to be portable. So, it's an advantage for the usability of such a technology but could also be a vulnerability for security. In this section, we will analyse whether characteristics of SIM Cards are sufficient to define it as a trusted element.

2. **Processing Power**: Just like any other hardware, SIM Cards come in various processing capabilities and variants from an 8-bit microcontroller to greater than 32-bit architectures. The processing capabilities depends on the application and users because of their need and specification. Processing power does play a vital role in building a trusted element because more the processing power more security could be provided on SIM Cards.

3. **Storage Capacities**: As with processing power the storage capacities of SIM also vary depending on the users and vendor's requirements. Still newer chip production technologies have taken the storage capacity up to a GigaByte (GB), a mass majority of smart cards in the industry as of this writing contain a few hundred kilobytes of persistent memory [19]. The memory structure of SIM Card is usually divided into three different storage types:

   - **RAM**: Random Access Memory, which is volatile and mutable. Contents of RAM are not preserved when power to the card is removed.
   - **ROM**: Read Only Memory is persistent and non-mutable. ROM is mostly used to load with the program code that would never be modified (e.g., the Card Operating System). This loading is done during the manufacturing of the card. ROM contains the cards operating system routines, persistent data, and permanent user applications. Contents of ROM are preserved when power to the card is removed.
   - **EPROM**: Electrical Erasable Programmable Read-only Memory (EPROM) is a persistent and mutable memory used for data storage on the card. Content in EPROM is preserved when power to the card is removed. It is usually meant to load programmable code and updates later in the smart card lifecycle after post issuance.

4. **Encryption and security**: Most SIM Cards have customised encryption/signature (hash) verification hardware because the main CPU could be less efficient (for code running on a virtual machine). Usually, SIM supports DES /Triple-DES, RC4 but also more recently AES,

Table 4.1: Comparison of various Secure Element (SE) [37]

Criteria SE Alternatives	Security	Re-usability	Standardization	Total
Baseband Processor	+	NA	NA	NA
Embedded Hardware	++	NA	+	++
SMC (Secure Memory Card)	+	++	+	++++
UICC	++	+	++	++++++

RSA, DSA, COMP128 and Public Key Infrastructure (PKI) are added to increase it's security. PKI private keys never leave the card. A hardware random generator is often present to provide more secure encryption key generation [19]. Above all, the cryptography policy of SIM Cards could be vendor-specific depending on the implementation and specification details.

5. **BaseBand Processor**: Due to the need of high computational power for controlling communications and computational functions in mobile devices, the BaseBand Processor is executed. BaseBand Processor is responsible for handling a cell phone connectivity and also managing the application operations. Use of providing a high level of security, the SE can be hosted by the secure memory of the BaseBand. Thus, the handset architecture does not have to be modified [10, 19].

6. **Secure Memory Card**: A portable Secure Memory Card (SMC) is made up of memory, embedded smartcard element, and smartcard controller. In other words, it is a combination of a memory card (e.g., MMC, SD, etc.) and a smart card [40]. Therefore, it's considered that the SMC is responsible for providing the same high level of security as a smart card. Furthermore, it is compliant with most of the particular standard, interfaces and environment for smart cards (e.g., GlobalPlatform, ISO 7816, Javacard, etc.). Although, SMC are portable and have better storage capacity, the SMC can host a large number of applications in it and does not need to be reissued whenever the user purchases new mobile device [40]. One of the advantages is that the SMC could be embedded into any mobile device that supports NFC technology.

7. **Hardware Element**: In the embedded hardware case, GlobalPlatform's SE is a smartcard-embedded onto the mobile and cannot be removed. This feature prevents the attacker from doing physical attacks on these smart cards. Hence, SE also provides the same high-level security in mobile devices like the one supported by a smart card. Nevertheless, this hardware element is embedded in the mobile devices during the manufacturing stage, which must be personalized after the device is delivered to the end user, and thus implies the design of a new secured personalization process and indirectly an increase of the cellphones price for the customer [40].

Table 4.1 shows the comparison of various SE. Therefore, different distribution of the SE alternatives and its relevant criteria are established. It appears that the BaseBand Processor based SE alternative is neither appropriate for a short-term nor a mid-term implementation.

For hardware SE alternative, if the lack of standardization is striking, it could be considered as an attractive mid-term alternative, especially for banks that are willing to find an independent alternative (i.e., alternative which does not oblige them to share a secure area with the Mobile Network Operator (MNO)) [40]. However, before that, a personalization process also plays a vital role in evaluation various SE. Hence, personalisation process would have to be designed, a communication protocol to get standardized and the chip industry would have to be more involved in the development of such hardware [40].

According to Reveilhac et al. [40], SE based UICC alternative is clearly the most advanced solution and is the best for short-term alternative. Although being portable, UICC is secured, well-tried and also it's complaint with smart-card and SE standards.

Following explanation is an illustration of UICC cards: The simplified model of owner relationship on the UICC is where the MNO owns the whole chip and all applications on it. In other words, UICC is implemented only as an SE and it provides authentication to the telephone network [10]. Obviously there can be more parties than the MNO, who can use the UICC as an SE. But even if they were allowed by the MNO to place their applications on the UICC securely, would they consider it secure? The reason for doubt is that if they sent their applications with the corresponding secret keys in it to the UICC, the MNO would be able to read the keys and potentially have access to all communication in clear text even if it was encrypted [10]. To solve such an issue, there are new security models that generate both security and revenue to all stakeholders, since only favoring one of them would harm the cooperation needed to implement such a complex solution [18, 40].

4.3 Limitations of SIM Cards

In the previous section, we discussed that SIM Cards contains the characteristics as that of trusted element. Also, the fact that many companies are trying to develop the new solution that could add more features than SIM Cards. But, SIM Cards have few limitations that restrict to define it as trusted element. Firstly, trusted elements like TPM is bound to a platform (hardware/firmware) but a smart card is "portable". Hence, a smart card is more vulnerable to attack. And importantly, the TPM also acts as the Root of Trust for Measurement (RTM) for a particular platform but SIM Cards lack in providing such functionalities.

We can witness the basic functionalities differences between the SIM Card and a trusted element as TPM. So, there was a need to for developing an efficient, more secure, tamper-resistant element that could fulfill the capabilities of a trusted element. In the following section, we explain UICC (Universal Integrated Circuit Chip) which is responsible for ensuring the confidentiality, integrity and security of all kinds of user's personal data, and it typically carries capacity of holding a few hundred kilobytes.

4.4 Universal Integrated Circuit Chip (UICC)

Universal Integrated Circuit Card (UICC) also known as SIM Cards for modern mobile devices. UICC comprises of tiny computers with CPU, memory, I/O system and a file system. They are affordable (ranging from a few cents to a couple of dollars depending on the configuration), tamper resistant and also the SE, as defined by GlobalPlatform[1]. The SIM card was from the beginning a removable card and today we recognize SIM functionality on removable Universal Integrated Circuit Cards (UICC) in several physical form factors (e.g. 1FF, 2FF, and 3FF) [10]. On modern UICCs, the basic SIM functionality (identification and authentication) is merely one of the several applications running on the UICC operating system [10].
The UICC has proven itself to be robust against attacks and is well suited to hosting sensitive applications and data. This functionality of UICC provides a trusted environment for both programs and data [10]. Java Card versions with a GlobalPlatform SE can facilitate isolation between Security Domains (SDs) and allows for loading and installation of applets (programs) to these security domains [10].

Figure 4.2 explains the architecture of UICC, as explained by [33]. In this architecture is an overview of how the UICC and mobile device are connected to each other:

1. CPU/Processor: It is responsible for processing in UICC the applications.

[1] https://www.globalplatform.org/mediaguideSE.asp

2. RAM: As discussed in Section 4.2, RAM is a memory for storing/hosting applications securely. Also, responsible for reading and writing the applications.

3. ROM: As discussed in Section 4.2, ROM is used as a memory for storing user data. This only allows reading of the data.

4. Interface: A User interface that could be connected to the screen, keyboard, microphone, speaker.

5. A Communication interface between UICC and the Mobile device

6. Non-Volatile Memory: Storage of applications is transferred in the Non-volatile memory with the ability to securely store the encryption keys or certificates.

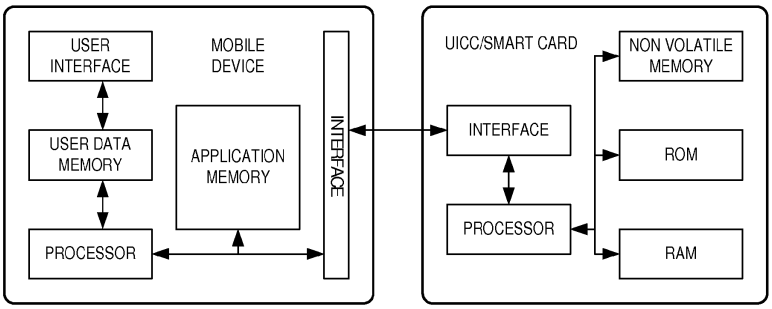

Figure 4.2: UICC/SmartCard enabled architecture for mobile devices [33]

According to GlobalPlatform, The primary goal of the UICC is to ensure the security and integrity of the card's components for the life of the card. These components are:

- The runtime environment;
- The Issuer Security Domain;
- Supplementary Security Domains;
- The Applications.

To ensure card's security and integrity, the GlobalPlatform is designed to support a range of secure mechanisms for:

1. Data Integrity;
2. Resource Availability;
3. Confidentiality;
4. Authentication.

The choice of security policy and cryptography is assumed to be industry and product specific, also differs from vendor to vendor.

4.5 Advantages of UICC

UICC is implemented mainly in all mobile telecom networks and its also a type of smart card technology. Smaller in size than a traditional SIM card, it contains a computer, or microprocessor, its own data storage and software. It is an evolution of the SIM used to identify subscribers in GSM (Global System for Mobile Communications) networks. Unlike SIM Cards, Mobile devices can use encryption keys from the UICC Card to encrypt/decrypt data stored in the memory of the device, this data could be user's personal data.

However, a significant advantage of the UICC over the SIM is that it can have multiple applications on it. In UICC, multiple applications are stored, and their processes are also executed on the same device, it is essential to be able to build trusted applications and store their associated credentials in a trusted environment. The MNO have identified the UICC as the current most appropriate SE because of the following advantages that it offers to the marketplace: Universal, portable, dynamic remote management, standardised and long lifecycle.

4.6 UICC as a Trusted Element

As discussed above, UICC Card is a new generation of smart card and also one of the trusted element that could provide a portable security to mobile devices. The main goal of UICC Cards is that it provides better security i.e., Confidentiality, Integrity, Availability for the mobile devices. UICCs are not only used for basic SIM functionality (identification and authentication) but also for running applications on the UICC Operating System [10].

Hence, above mentioned goals of the UICC Cards/Smart Cards could be completed with following requirements:

- **Trusted Element**: As a Trusted Element, UICC Card provides better security and is included in software and tamper resistant hardware that provides a filtered access to applications stored directly on the smart card. It also provides a trusted communication between UICC card and mobile device interface. But above all, UICC Card are only part of a larger card system involving multiple parties and off-card components, the GlobalPlatform[2] also relies upon non-cryptographic, procedural means of protection, such as code testing and verification, physical security, and secure key handling.

- **Tamper Resistance**: Tamper-resistant microprocessors are used to store and process private or sensitive information, such as private keys, personal information. As discussed above, UICC card has its own CPU, ROM and is capable of processing and even has its own storage areas. So in order to protect UICC Cards, its important to ensure that an attacker should be restricted from retrieving or modifying any information.

- **Data Security**: Data Security in UICC is more advanced than SIM Cards. Many UICC cards are comprised of a processor, a communication interface that connects UICC to the mobile device, RAM, ROM and non-volatile memory, as shown in Figure 4.2. UICC is capable of storing the user's personal information, private cryptographic keys. UICC does not generate the certificate and/or private cryptographic keys, but rather they are generated in more Trusted Environment (e.g., software, machinery) [33]. The Keys and Certificates then would be transmitted to the UICC using encrypted transport protocol within the mobile device. Also, UICC Cards have separate memory for hosting applications securely. According to [33], The mobile device will receive data messages of various types, and request keys or certificates from the UICC which would then be stored in an encrypted form using a key/certificate generated. Also, the user will be able to use encryption keys and certificates via a GUI. The user is able to pickup a single or multiple files from the mobile device and can choose to encrypt the files using the private key present in UICC [33].

[2] https://www.globalplatform.org/

4.7 Summary

SIM Cards are the traditional smart cards with the fewer security capabilities on the mobile devices. Recently, SIM Cards using the more recent Advanced Encryption Standard (AES) or Triple DES standards are capable of providing better encryption standards [10]. But as the technology matures, several flaws in the cryptography of the SIM Cards are detected. For instance, the use of DES (Data Encryption Standard) has vulnerabilities that could give an attacker the root access to SIM Card[3]. To overcome the limitations of traditional SIM Cards, UICC (Universal Integrated Circuit Chip) is developed, a new generation of SIM Cards. As discussed above, UICC is SE as defined by GlobalPlatform to ensure more security for ensuring Data Integrity, Availability, Confidentiality and Authentication [10]. Unlike traditional SIM Cards, UICC cards do not store any user specific data like personal contacts but it stores cryptographic keys, personal identification (in an encrypted form), etc. [33]

Therefore, this chapter explains what are better security alternatives for hardware-based platforms in mobile devices. SIM Cards is capable of considering SE, but its new version is more enhanced with security capabilities and secure communication to MNO. According to GlobalPlatform, UICC cards are removable, tamper-resistant secure modules that are capable of securely hosting applications and confidential and cryptographic data (e.g. key management) by the rules and security requirements set forth by a set of well-identified trusted authorities[4].

[3]https://srlabs.de/rooting-sim-cards/
[4]https://www.globalplatform.org/mediaguideSE.asp

Chapter 5

Biometrics in Mobile Devices

Biometrics identifies people by measuring certain aspect of individual anatomy or physiology (such as your hand geometry or fingerprint), some deeply fundamental skill, or other behavioral characteristic (like a handwritten signature), or something that is a combination of the two (e.g., voice) [1]. As discussed in Section 2.2.3, Biometrics is a method of user authentication through "what he/she is" rather than just assuming that the user owns a device that they have or carry. Biometric based authentication are becoming the base of a broad range of more secure identification and personal verification solutions. As the level of security breaches and transaction fraud increases, the need for highly secure identification and personal verification technologies is becoming apparent. According to Savvides [4], following are the current security problems with passwords:

1. Repetition of using passwords for multiple accounts makes it more vulnerable.
2. Users build passwords of things that are clearer (birthdays, wife/girlfriends name, pets, etc.). Thus, it makes easier to crack passwords.
3. Creating strong passwords (mainly meaningless to users) makes it hard for memorising, and also really hard to memorize such passwords for multiple accounts.

Biometrics (in mobile devices) is a user authentication technology such as the face, finger, hand, iris, and speaker recognition that are commercially available in mobile devices. We believe that the era of using biometric authentication for mobile devices is imminent [49]. Many mobile users are now accustomed to talking to small mobile devices, and seeing themselves through the device camera. As the quality of sensors and processing power of mobile devices improves, mobile biometric authentication has become a realistic proposed [49, 1]. To improve the mobile device security, the main reason to use biometrics in the mobile device is to protect the data on the device and to provide secure yet convenient access to the device and to the network it may be connected to [49, 29].

In this thesis, we view biometrics authentication as a trusted element. The reason is that there are many types of research and biometrics applications that consider security aspect of biometrics [12]. Applications of biometric authentication consist of trusted components, devices by well-known companies. But, there are still few applications of biometrics that have been able to provide better security and trust to the user. One of the reasons behind this is the less-trusted deployment of all trusted elements. Hence, one of our research questions is, how secure is this trusted element (biometrics authentication)? And does it mitigate all the threats involved during its deployment?.

Furthermore, we have divided a biometric system into two parts: Integrated Biometrics and Non-Integrated Biometrics as shown in Figure 5.2.Integrated Biometrics system is a user-authentication system that performs all its process within a mobile device. In short, the capturing, feature extraction, storage, comparison and authentication policy of user's biometric data is done within a

mobile device. The main advantage of this system is more security is provided because biometric data always resides in the mobile-devices. In Non-Integrated biometrics, the user's biometric data goes to remote servers for storage purposes. For instance, banking applications are providing voice biometrics as an option for authentication[1]. Therefore, user's biometric data is stored on the application servers. It is important that the software/application is trusted, as security and privacy of a user is upon them.

During deployment of biometrics as a trusted element, it also depends on how communication happens within a mobile device (for integrated and non-integrated biometrics) or with a mobile device. Many researchers claim that non-integrated mobile biometric authentication the application (software) itself has to be trusted [49, 29]. Also, its communication with the server or cloud has to be trusted to avoid any attacks like man-in-the-middle attacks. For instance, if an impostor can access a users biometric information, then impostor can replay this information to a matching algorithm used for user authentication, and be accepted as a valid user, given that the matching algorithm is not able to recognize the origin of the biometric information [12].

According to [29], any human physiological and behavioral characteristic can be used as a biometric feature as long as it satisfies the following requirements:

- **Universality**: each person should have the characteristic.
- **Distinctiveness**: any two individuals should be sufficiently different regarding the characteristic.
- **Permanence**: The characteristic should be sufficiently invariant (on the comparison criterion) over a period.
- **Collectability**: The characteristic can be measured quantitatively.

5.1 Fingerprint

Fingerprint recognition is an automated process of identifying or confirming the individual based on the comparison of two given fingerprints. Fingerprint recognition is by far the most used biometric solution for authentication on computerized systems[2]. The reasons for fingerprint recognition being so popular are the ease of acquiring, use and acceptance by users as compared to other biometrics. Furthermore, there are several (ten fingers) sources of this biometric on each [48].

A fingerprint is the pattern of ridges and valleys on the surface of a fingertip. There are different levels of information when representing fingerprint:
Level 1 (Global): There are three basic patterns of fingerprint ridges: arch, loop, and whorl, as shown in Figure 5.1.
Level 2 (Local): The major minutia features of the fingerprint are ridge ending, bifurcation and short ridge. The representation of a fingerprint by their minutiae is not only the type and position of the feature but also the direction and the angle of the ridge, the distance between two continuous ridges.
Level 3 (Fine): Ridges details as the width, shape, inholes, etc.

Following sub-sections explains the popular deployment of fingerprints authentication in detail.

[1] http://www.nuance.com/for-business/customer-service-solutions/voice-biometrics/index.htm
[2] http://www.biometric-solutions.com/solutions/index.php?story=fingerprint_recognition

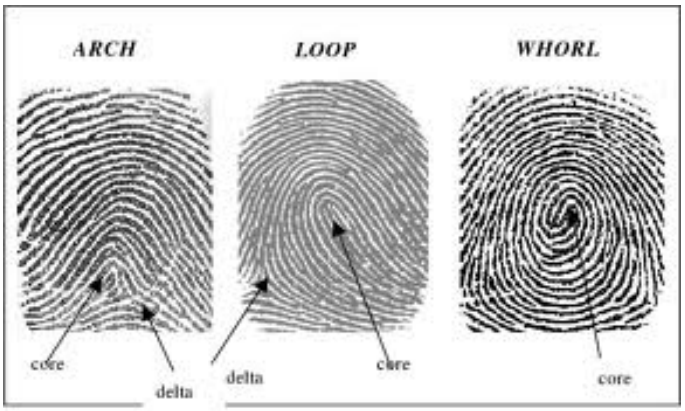

Figure 5.1: Structure of Arch-Loop-Whorl

5.1.1 Fingerprint Architecture with TrustZone

This section explains in detail about working of Fingerprint systems in ARMs TrustZone architecture. Due to the lack of full specification of fingerprint authentication in ARMs TrustZone, the following description is based on the discussions and blogs available on the internet. As discussed in Chapter 3, TrustZone separates the normal world (the normal user/kernel mode) and the secure world (Trusted Execution Environments, or TEE) by creating additional operating modes, known as the Secure mode and the Monitor mode. The Secure mode has the same capabilities with the normal world while operating in separate memory space. Critical security of the TrustZone is installed in the secure world during system initialization, before the ARM processor switches to normal world operating mode, thereby enabling the use of ARMs Module TrustZone extensions [47]. The mode of operations could be switched between two worlds namely, secure world and the normal world using the mechanism called as "Secure Monitor Call" (SMC). The role of the Monitor Mode software in the architecture is to provide a robust guard that is responsible for switches between the secure and non-secure processor states.

Fingerprint authentication works in a similar TrustZone Architecture. Fingerprint authentication leverages the ARM TrustZone technology to ensure the trust of data from camera/accelerometer. This data is then encrypted and stored in a secure world of TrustZone architecture. Also for the attackers with the root privilege in legacy OS would not be able to compromise the integrity and freshness of the collected data. Figure 5.2 depicts the enhanced fingerprint authentication framework utilizing TrustZones protection. In this design, the fingerprint sensor driver, fingerprint recognition algorithm, and the fingerprint data are all isolated in the secure world, so the fingerprint authentication framework remains secure even though normal world kernel is compromised.
Samsung also implements the similar architecture like ARMs TrustZone[3] in its modern mobile devices. According to [6], Samsung's Galaxy S5 does not store fingerprint image. Instead, it stores the mathematical representation of the user's biometric (which prevents reverse engineering). This representation is then stored in the secure part of the semiconductor architecture that cannot be accessed physically or by any external third parties, remote access. This security measures to secure the fingerprint authentication system is well-protected. But, there are still attacks like spoofing for which Samsung has not discussed any countermeasures. For instance, companies like FireEye has demonstrated that Samsung's Galaxy S5 fingerprint scanner could be hacked by collecting the data which enters through the sensor and before reaching to the trusted zone for encryption (mathematical representation)[4].

[3]http://www.androidauthority.com/arms-built-security-might-just-get-rid-password-397924/
[4]http://bgr.com/2015/04/22/samsung-galaxy-s5-fingerprint-scanner-hack/

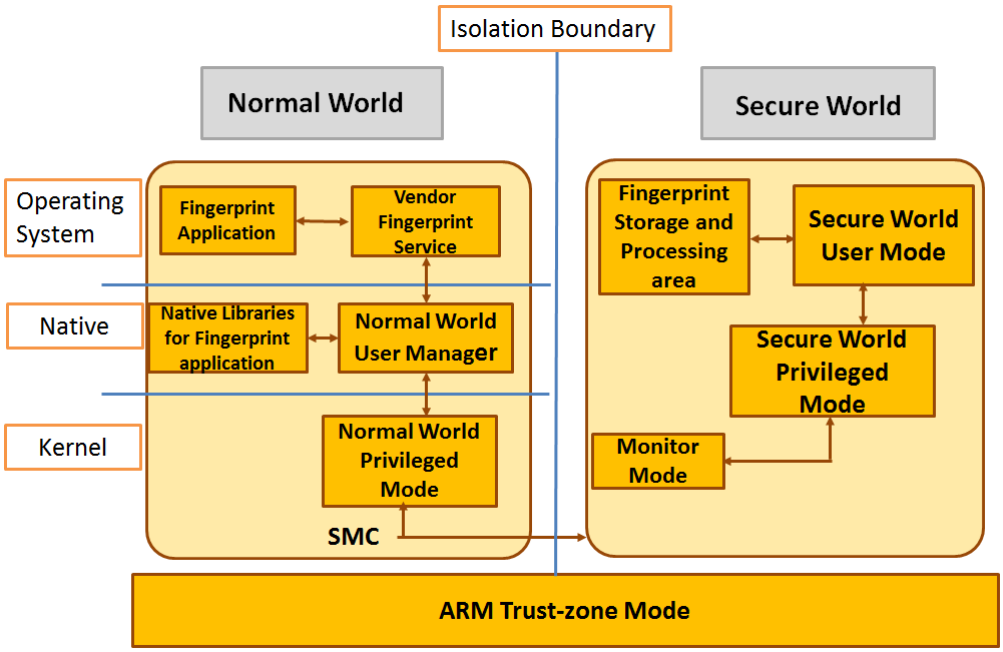

Figure 5.2: Fingerprint Verification with TrustZone Architecture

5.1.2 Secure Enclave

The Secure Enclave is a coprocessor for Apple's A-7 series processor. Secure enclave has a secure boot that is separated from the application processor. Thus, the reflection of communication between the secure enclave and the application processor is highly encapsulated. The primary functionalities of secure enclave include key management, processing cryptographic operations and maintaining the data integrity [13]. Each secure enclave is embedded with Unique ID (UID) during the fabrication process from the manufacturer. Other parts of the system cannot access UID, neither does Apple [7]. UID is used to encrypt the Secure Enclave's memory space and data of files stored in the file system [7].

Generally speaking, the secure enclave is like a vault where user's information is stored, and this information cannot be accessed without the user's Touch ID [**?**]. Also, the fingerprint will be saved after it has been encrypted. The Secure Enclave also decrypts and processes the fingerprints that are received from the Touch ID, verifying if the coming fingerprints are compared with the templates. The application processor forwards the fingerprints data to the Secure Enclave [7, 13]. The reason behind this is that biometric data is encrypted (like Fingerprint) with a session key between the Secure Enclave and the Touch ID, so that the application processor cannot read it.

5.1.3 Working of Touch ID

Touch ID is a biometric authentication sensor based on a high definition fingerprint scanner embedded into home button on iPhones and iPads [13]. This sensor is used to allow users to unlock their devices by simply touching the home button. Even though, Touch ID is used to unlock a device without a password; users are still required to set passwords on their devices, before being able to use Touch ID [13]. Touch ID sensor is developed by a laser cut sapphire crystal as well as a capacitive touch sensor. Hence, it is capable of taking a high-resolution image of user's fingerprint and intelligently analyze it to provide accurate readings from any angle. Only Touch ID sensor is responsible for recognizing the touch of a finger, so that the sensor only gets activated when needed, preserving battery life [23].
Cherapau et al. [13] introduced the workflow of Touch ID. Unlocking the device using Touch ID

and password is explained in the following steps for simplification.
Following steps are involved in unlocking the iPhone with Touch ID enabled from the Figure 5.3.

- To unlock the device, the user has two options: they can either type in their passcode (1) or can use the Touch ID (2).
- If a user selects Touch ID as a way of Authentication (2), it authenticates the user by matching his/her fingerprint with saved fingerprints (templates) (3).
- If the authentication is successful, the sensor releases the Temporary Encryption Key (TEK) to the secure enclave (4). This allows decrypting class keys and sending them to the crypto-chip[5] (7).
- If the user fails to authenticate for five times with Touch ID (2) or does not unlock device for 48 hours, the Touch ID sensor flushes the TEK, which leaves typing in the passcode (1) as the only option for unlocking the device.
- Without Touch ID, the user can type his/her passcode, which is sent to the secure enclave. The combination of the device key (6) and password (5) are used to decrypt class keys and send them to the crypto chip (7).

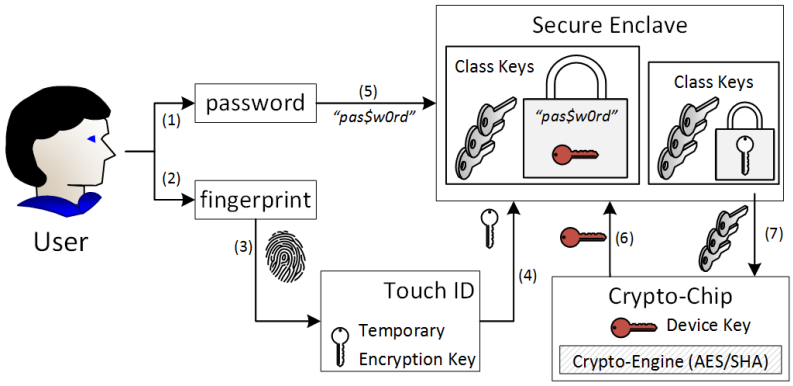

Figure 5.3: Unlocking iPhone with Touch ID or Passcode [13]

Security of iPhone's Touch ID has always been an issue within researchers. As discussed above, fingerprints in Touch ID are encrypted in the secure enclave to prevent attacks like reverse engineering. The biometrics hacking team of the Chaos Computer Club (CCC)[6] has successfully bypassed Apple's Touch ID biometric security using easy everyday means. The fingerprint of the phone user, that was photographed from a glass surface, was sufficient to create a fake finger that could unlock an iPhone 5s secured with Touch ID[7].

5.2 Voice Authentication

Voice is a combination of physiological and behavioral biometrics. These features of an individuals voice are based on the shape and size of the defined characteristics (e.g., vocal tracts, mouth, nasal cavities, and lips) that are used in the synthesis of the sound. Physiological characteristics of the voice are invariant for an individual, but the behavioral part of the speech of a person changes over the period (e.g., medical conditions, emotional state, age factor). Voice of an individual is also not very distinctive. A text-dependent voice authentication system is based on the utterance

[5]When the user locks the device, the class encryption keys are wrapped by a random Temporary Encryption Key (TEK)
[6]http://www.ccc.de/
[7]http://www.ccc.de/en/updates/2013/ccc-breaks-apple-touchid

of a fixed predetermined sequence of string. One of the main advantages of voice authentication system deploying on the mobile devices is that the cost of developing this system is cheaper as compared to other factors of authentication.

The most significant difference between voice biometrics and another biometrics is that voice biometrics is the only biometrics available in the commercial market that can process audio information. Most other biometrics are image-based (e.g., face, iris). Furthermore, the most commercial voice biometrics systems are designed for use with virtually any standard telephone that are commercially available. The ability to work with standard telephone equipment makes it possible to support broad-based deployments of voice authentication systems in a variety of settings. In contrast, most other biometrics require proprietary hardware, such as the vendors fingerprint sensor or iris-scanning equipment.

Voice authentication (in mobile devices) is based on voice-to-print authentication, where complex technology transforms voice into text. The need of voice biometrics in mobile devices is because the online based authentication systems. Voice biometrics (in mobile devices) requires a microphone, which is also available on mobile devices nowadays. Many researchers claim that voice biometrics can replace the currently used methods of authentication, such as PINs, passwords, or account names. But voice authentication (in mobile devices) will be a parallel technique for finger-scan technology as many people view finger scanning as a higher authentication form.

5.2.1 Enrollment and Verification Process of Voice Biometrics

Voice verification, also known as speaker recognition, determines the identity of the speaker. During the first stage of voice authentication system which consists of the enrollment process. The enrollment process requires an individual to say a set of string, typically a numeric value, in sequence and this process is repeated several times. Using a feature extraction process that is responsible for extracting the feature points in an individual's voice sample. After this process, a template is created from this input using feature extraction model, which defines the characteristic of the voice [30]. The template is then stored in a database depending on the type of platform (i.e., integrated or non-integrated).

After enrollment process, the user can authenticate himself to the system. In this process, the user is prompted to speak into an embedded microphone and vocalize a random sequence of digits, as they appear in the display. While many PDAs incorporate a built-in sound card and microphone, they typically lack the processing power (i.e., floating point hardware) to perform the needed calculations quickly enough. According to Jansen [30], the main reason for this is that voice-modeling algorithms rely heavily on floating point arithmetic, whose execution must be emulated in software. Other drawbacks to this type of solution include environmental sounds, individual speaker variability in pronunciation (e.g., for the number 12, saying one-two versus twelve), the significant amount of time needed for enrollment compared to other biometric mechanisms, and the larger size templates that are needed [30].

One of the advantages of voice authentication system is that it provides better security as compared to other biometrics. During the verification process, this system not only compares the voice prints of the user but also deploys the speech recognition algorithms to check whether the particular challenge has been said by the user (liveness detection of voice).

5.3 Other Biometrics Recognition

5.3.1 Face Recognition

According to [3], Facial Recognition is defined as follows: "Facial recognition is the automatic processing of digital images that contain the faces of individuals for the purpose of identification, authentication/verification or categorisation of those individuals."

Facial recognition is also considered as one of the popular forms of Biometrics. As the technology

develops, various electronic forms of facial recognition are being developed. Recently, many mobile manufacturers have developed various facial recognition applications for mobile device authentication (e.g., Google, HTC, iPhone, Motorola, etc.).

Ijiri et al. [27] discussed the general face recognition technique that consists of several stages where many design, deployment and security decisions must be taken:

- Feature extraction can be performed in many ways. The set of relevant features must be previously defined.
- Learning algorithm decisions condition the way the features are analyzed to obtain user patterns.
- Similarity measures: the suitability of the measures depends on the pattern structure.
- Similarity thresholds can be experimentally defined, according to the real environment of system usage.

According to [3], the process of facial recognition itself is comprised of some discrete subprocesses:

1. Face Detection and Capturing: The process of capturing the face of an individual and converting to a digital form (the digital image). For instance, the embedded camera in mobile device captures the face when it detects the face of a person from a certain distance (varies from mobile manufacturers). An image capturing of a face could be performed by static camera or video system that can generate images of better quality, such as web camera [16].
2. Normalization: The process to smooth variations across detected facial regions. For instance, converting to a standard size, rotating or aligning color distributions of a facial image captured.
3. Feature extraction: The processing of isolating and outputting repeatable and distinctive readings from the digital image of an individual. The facial features often used are those who change petite over the ages, as the upper ridges of the eye sockets or the areas around the cheekbones, sides of the mouth, nose shape and the relative position of these features about each other [16].
4. Enrollment: If this is the first time an individual has encountered the facial recognition system the image and reference template may be stored in a database as a record for an authentication process.
5. Comparison: The process of measuring the similarity scores between a set of features (the sample) with the template in the system. The main purposes of the comparison process are to identify and authentication. A third purpose of comparison is to categorize the template that is the process of feature extraction from an image of an individual to classify that an individual in one or several broad categories (e.g. age, gender, a colour of clothes, etc.) [3]. It is not necessary for a categorisation system to have an enrollment process [3].

Following explanation of the Client-Server architecture is emphasized on non-integrated biometrics system. In non-integrated biometric authentication, the user is dependent on the software/plug-in to secure its biometric data. Hence, non-integrated biometric solutions (face and voice) will have to make use of trusted software measures to protect the template and the biometric processing.

Client - Server Architecture

Recently, many banking applications, online shopping applications use facial recognition as an option for user authentication. Client-Server architecture is considered very usable for the users but could be more challenging for security. If the deployment of such software is not secure, then it might lead to malicious behavior either on user's mobile devices or the applications. Many online vendors are still researching for providing better security for facial recognition to the mobile devices. Other advantages of a client-server architecture include the quick execution of the technology updates and the ability to share biometric models across multiple devices via a centrally hosted database[8]. All input samples are converted into a mathematical representation (encrypted form) for assurance of security and privacy before being transmitted across the network. But, one of the disadvantages of this non-integrated architecture is that the client requires a secure connection to the server, so this architecture is unsuitable for the use cases requiring biometric face authentication in remote or rural areas with poor network coverage[9].

KeyLemon provides an online authentication mechanism using biometric facial recognition. The website claims that its responsible for ensuring a transparent, anti-spoofing and secure online authentication mechanism. The architecture of the Client and Server authentication is explained below:

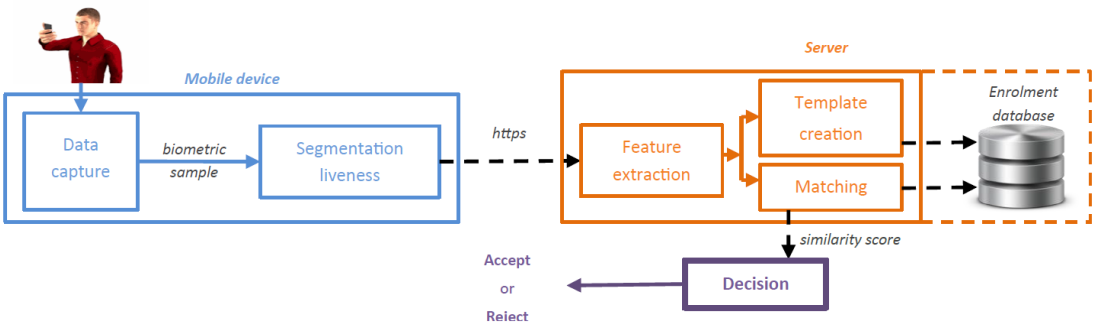

Figure 5.4: Client Server Architecture of Face Recognition

- **Step 1**: Firstly, user is connected to the server. And then user captures their facial image.
- **Step 2**: The captured image is then forwarded as a raw biometric sample for a segmentation liveliness. This process happens in the mobile device depending on the OS; Native face recognition libraries check whether the user is legitimate or fake using algorithms.
- **Step 3**: When the native libraries approve that the user is legitimate then the biometric sample of the user is forwarded to the server side using HTTPS link. This protocol helps for making a secure communication between client and sever, thus securing user's biometric sample.
- **Step 4**: At the server side,biometric sample first executes feature extraction mechanisms where lots of facial characteristics are extracted based on which the user gets a unique identity. This process is explained briefly in Section 5.3.1.
- **Step 5**: After the feature extraction process, user's biometric sample is either stored in the encrypted form in the Servers (e.g., cloud) or it's used for the comparison purpose. In the matching system, similarity score decides whether to grant access or not. Moreover, the similarity scores indicate the degree of fit between the features and the compared template. In some cases, the features may take the same form as the stored template. For verification, a single specific claim of subject enrollment would lead to a single similarity score [1].

[8]https://www.keylemon.com/oasis
[9]https://www.keylemon.com/oasis

Also, there are other factors that are responsible for an efficient use of face recognition system. For instance, A High Resolution (HR) camera is often used to capture the photographs, but there are few researches, as [26], in which infrared cameras are used. Socolinsky and Selinger [45] analyzed face recognition performance using visible and thermal infrared photographs. Other proposals also use the mobile phone camera to build the database. Many researchers have developed their own way for facial recognition system in mobile devices. But, many of these schemes are tried on the standardized databases images. Therefore, many implementations fail in the real-life scenarios where situations are unpredictable.

Attending to the face detection approach, we can find two different types of techniques:

- Skin color segmentation techniques.
- Simple feature extraction techniques.

Most of the recent works are based on the Viola-Jones algorithm [1], which has become the main reference for face detection techniques since 2001 for mobile devices. This approach for visual object detection uses the simple feature extraction method, and it is capable of achieving high face detection results in the mobile devices.

5.3.2 Iris Recognition

Iris recognition is a particular type of biometric authentication system that can be used to identify an individual reliably by analyzing the patterns found in the iris. The iris is a reliable form of authentication because of the uniqueness found in its pattern. Even though, there is a genetic influence, especially on the iris color, the development of iris is done through folding of the tissue membrane and then its leads to degeneration (to create the pupil opening) which finally results into a random and unique iris [32]. One of the advantages of iris authentication system is there is huge variability of the pattern between individuals, i.e., that large databases can be searched without finding any false matches [14]. This states that iris can be used to identify individuals rather than just confirming their given identity. Among the biometrics as mentioned above, iris is known for its inherent invariance and accuracy, though only a few works have explored this topic on mobile devices

Automated iris recognition in mobile devices using following three main tasks: first we must locate the iris through a camera. Iris detection is an integrated mechanism in many mobile devices. For checking the liveness of the iris, system relies on an infrared LED and an infrared camera to scan user's iris. Secondly, it is required to encode the iris information into a format that is amenable to calculation and computation, (For instance, a binary string). Finally, the data must be able to store and to load and compare these encoding. Also, it's more important to understand if the data is stored in a trusted element. Japanese mobile manufacturers Fujitsu[10] and NTT DoCoMo[11] have introduced a smartphone in markets which has an integrated iris scanner[12].

As discussed above, iris recognition is as unique identifier as fingerprint or the veins in user's palm, both identifying methods which have been used widely until now. Many researchers claim that iris recognition is more secure than other compared forms of biometric recognition. To register user's iris pattern on the authentication system, all user needs to do is stare at the camera for around 15-20 seconds.

Advantages of Iris Recognition

Iris recognition is most often used for security authentication purpose (e.g., like facial recognition).

- Iris recognition technology is currently being deployed in mobile devices where extracting the iris code based on Adaptive Gabor Filter in which the Gabor filters operating parameters depends on the amount of blurring and sunlight in captured image [31].

[10] http://www.fujitsu.com/global/
[11] https://www.nttdocomo.co.jp/english/
[12] `https://www.nttdocomo.co.jp/product/smart_phone/f04g/index.html`

- The technology has also been used to prevent unauthorized access to personal computers and mobile devices.
- A small, portable iris-scanning mobile device is available for consumer use, bypassing the need for cumbersome password entry.
- Iris recognition applications are also available for the iPhones and other smartphones.
- The potential exists for iris recognition technology to replace most current forms of physical access-based identification (in mobile devices). This states anything that requires a password, personal identification number (PIN), or a key [25]. Another advantage of iris recognition over another biometric system is that unlike those physical methods of identification, an iris cannot be stolen. Iris recognition technology clarifies the problems of both password management and fraud [25].
- Glasses or contact lenses do not interfere with the operation of iris recognition technology. Very few surgical procedures involve the iris, in which case re-enrollment in the database is necessary.
- The reason behind this is that, no two irises are identical - neither even between identical twins, nor even between the left and right eye of the same person. Iris recognition technology is also accurate for comparison because it uses more than 240 points of reference in an iris pattern, whereas fingerprints use about 60 comparison points.
- It is an internal organ that is well protected against damage and wear by a highly transparent and sensitive membrane (the cornea). Thus distinguishing iris recognition from fingerprints, which can be difficult to recognize after years of certain types of manual labor.
- The iris is mostly flat, and its geometric configuration is only controlled by two complementary muscles that control the diameter of the pupil. This makes the iris shape far more predictable than, for instance, that of the face.

5.4 Vulnerabilities and Countermeasures of Biometrics Authentication

As discussed in the previous sections, biometrics (in mobile devices) aim at improving security capabilities using hardware and software platforms. But like any other security system, biometric authentication also have many vulnerabilities that are susceptible to threats. Designing secure biometric authentication systems can be a challenging process, and it is imperative to evaluate the performance and security of biometric system for finding the possible vulnerabilities and countermeasures. Biometric is a user authentication system. So, it's important to understand that security of a biometric system is equally dependent on technology and user. For better security in biometric systems, technology and user should work together.

Following are two significant attacks that are described briefly while considering the vulnerabilities in biometric authentication.

Hill-Climbing Attack: The term hill-climbing designates an attack in which the similarity score given by the matcher is used to iteratively modify a synthetically generated template, or group of templates until the verification threshold is reached [22]. According to [35], the hill-climbing attack may be performed by an application that sends random templates to the system, which are perturbed iteratively. Soutar proposed that the input image is conveniently modified until the desired matching score is attained [35]. The template format of the comparison process must be known to the attacker and also the input image size. Note that the image size is easy to obtain in general as it is usually made public by the sensor vendors (e.g., face, fingerprint verification) [35].

Trojan Horse Attack: Attacks like Trojan horse are found in the feature extractor process. In this attack, the impostor can replace the feature extractor module with a Trojan horse. The

objective of this attack is to have malware that appears to perform a desirable function for the authorized user, but instead it shows some other features that are commanded by an impostor. Usually, Trojan horses can be controlled remotely, and the effect on the biometric systems is also efficient enough for interrupting the process [34].

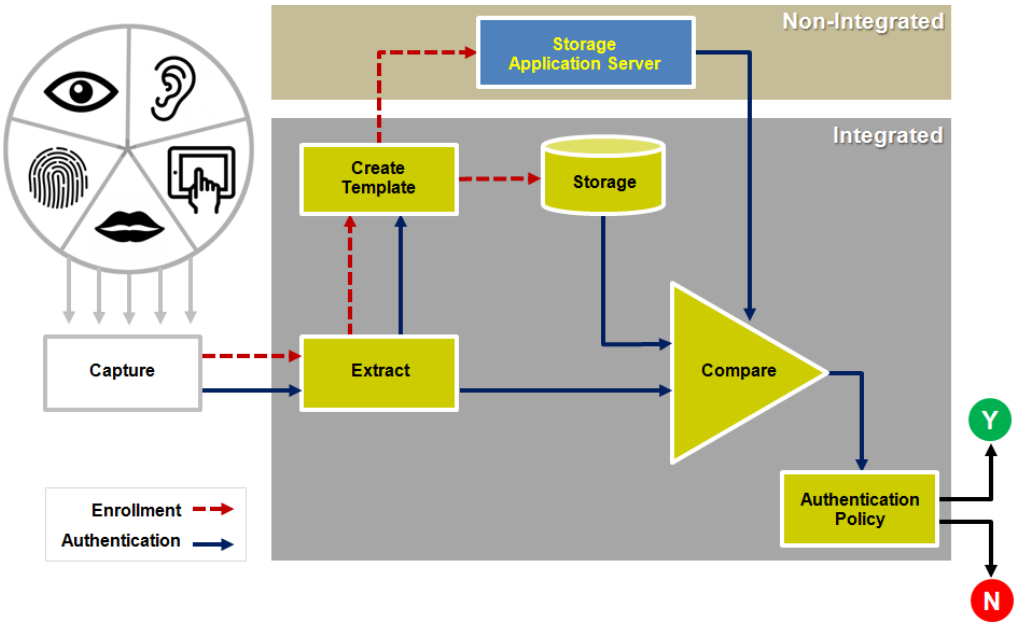

Figure 5.5: Risk Analysis of Biometrics Process

This section represents the detailed vulnerabilities and countermeasures analysis on every biometrics process. These vulnerabilities are intended to either bypass the security measures or to intercept the working of an authentication system. In Table 5.1, we represent the vulnerabilities and countermeasures vectors in a tabular format. In this thesis, vulnerability analysis of biometrics is also primarily categorized into two parts: Integrated and Non-Integrated biometric systems.

Table 5.1: Vulnerabilities and Countermeasures of Biometric Authentication

Platform	Stages	Vulnerabilities/Threats	Countermeasures
Integrated- Biometrics	Capture	* Spoofing * Malicious reader * Tamper attacks * Brute-force attack	- Tamper Resistance - Liveness detection - Better mobile security architecture
	Feature Extraction	* Insertion of malicious data * Main-in-the-Middle attack * Trojan Horse attack * Hill-Climbing attack	- Challenge-Response - Trusted environment
	Create Template	* Template attack * Modify template feature vector * Covert recognition	- Strong Security and Privacy policy - Trusted environment
	Storage	* Template retrieval * Template replacement and/or modification	- Cryptographic policies - Secure storage - Strong Access Controls
	Compare	* Match override / False Match * Attacks on comparison algorithms	- Trusted environment - Strong and secure algorithms
	Authentication Policy	* Modify Access Rights * Hill Climbing * Match Override	- Fined access control policies - Secure channel - Coarse scores
Non-Integrated Biometrics	Server - Side Vulnerabilities	* Template replacement * Impostor stealing templates * Non-removal of user's biometrics even after un-subscribe	- Challenge-Response authentication - Transparency with user - Encrypted templates
	Communication link threats	* Between Server and Client (like, MitM, Replay attacks) * Between capturing sensor and feature extractor process. (like, MitM, Replay attacks) * Eavesdropping attacks	- Communication over secure channel - Mutually Authenticate user using Symmetric or Asymmetric keys

Integrated Biometrics As discussed in Section 2.2.3, integrated biometrics is a user-authentication system that performs its process within a mobile device. One distinct feature about integrated biometrics is the user's biometric data resides on a mobile device. Deployment of the secure biometric system plays a vital role in securing user's biometric data. Therefore, the following are significant vulnerabilities and attacks possible on the integrated biometric system with corresponding countermeasures.

- **Capture**: Traditional biometric systems in mobile devices are vulnerable to spoofing attacks. For instance, the artificial finger could be created through commercially available silicon or gelatin, that can deceive a biometric fingerprint sensor. Pictures, speech synthesis tools can be used to deceive face and voice recognition systems in mobile devices. One of the major problems with biometric data is that once it's been stolen it cannot be changed quickly. An attacker can also perform Brute-force attack on the system. The motive behind this attack is to launch a dictionary of fingerprints on the authentication system. The possibility to access the system using brute-force attack depends on the comparison accuracy of the authentication system itself.
  Hence, possible countermeasures include the use of tamper-resistant hardware (e.g., sensors, microphone, embedded-camera) and also sensor should be able to detect if it's a real person or not. Additionally, using secure mobile device architecture helps mitigating more vulnerabilities on biometric sensors.

- **Feature Extraction**: Feature extraction is the process wherein input sample is processed which is captured by the sensor. In both integrated and non-integrated biometric systems, this process is performed on the user's mobile device. So, there is a risk of insertion of malicious data from the impostor. Additionally, attacks like man-in-the-middle where impostor could try to eavesdrop on the communication channel between sensor and feature extractor to steal the biometric sample. One of the indirect ways to attack the feature extraction process is the hill-climbing attack. As discussed above, this attack is a modified version of brute force attack, but hill-climbing attack uses the feedback provided by the device (e.g., number similarity scores matched). The feature extractor can be attacked using a Trojan horse attack, it then produces feature sets that are preselected by the impostor. Mitigating these vulnerabilities and attacks, there should be a trusted environment where such process are more secure (for, e.g., Deployment in TrustZone or secure enclave architectures).

- **Create Template**: Input sample that is processed during feature extraction it's then converted into a compressed image called "Template". Traditionally, the creation of biometric templates is performed on the mobile device without any protection. But modern mobile devices contain templates that are a mathematical representation of user's biometric sample, as discussed in Section 5.1.3.

  It is observed that one fingerprint can only contain a limited number of minutia points, usually 20-40. If attackers acquire minutiae information successfully, it's possible to reconstruct the template [51]. If the user is not aware of the acquisition of his/her biometric data, the application is defined as covert. This is one of the vital risks involved in a biometric system, regarding the privacy issue. For mitigating vulnerabilities on template creation process, a better security policy should be developed which includes cryptographic policies to secure the templates and user's biometric data.

- **Storage**: This is the database where the biometric templates are stored. In this attack, the attacker endangers the security of the database storage where all the templates are stored. Possible attacks on the database can be performed by exploiting the vulnerability in the database software. Alternatively, the attacker can add new templates, modify existing templates or delete templates. Hill [41] described a way to create an image of a fingerprint based

on the information contained within the stored template (reverse engineering).
Possible countermeasures to protect the stored templates is to encrypt them. Alternatively, using better cryptographic policy where all templates are digitally signed can also be secure. Developing strong and fine-grained access control policies also play a vital role in securing the database storage.

- **Compare**: Authenticating the user against the stored template, the corresponding template is retrieved from the database and comparison is performed against the template derived from a newly acquired input signal [39]. Possible attacks on the comparison vector are that the comparison vector is attacked and corrupted so that it produces preselected similarity scores in the biometric system. Impostors can try to attack the algorithms during comparison process through modification of variables, etc. Mitigating attacks on comparison vector, it's important to have this process in a trusted environment which allows to operate secure and robust algorithms.

- **Authentication Policy**: Authentication policy is a final stage that determines whether the access is granted or not. According to [39], the attack can be realized with a Hill-Climbing on the feature extractor, the comparison algorithm or the authentication policy of the system, acting as a manipulator of each components output. Alternatively, final match decision can be overridden by the hacker; then the authentication system could be disabled.
Mitigating attacks on the authentication policy includes fine-grained access controls, application of secured channel between comparison vector and authentication policy [39].

Non-Integrated Biometrics

As discussed in Section 2.2.3, non-integrated biometrics is a user-authentication biometric system that uses a client-server based authentication. One of the important characteristics of such system is that it communicates over the internet channel and user's biometric data is stored on a centralized server. Hence, following points emphasize on the vulnerabilities and attacks related to it with possible countermeasures.

- **Server-side vulnerabilities**: In the server-side biometric authentication system, there are several vulnerabilities like template replacement where an attacker can eavesdrop on the server's database. For example, if a user terminates relationship with a certain company, how can a user be sure that his/her biometric data is removed from their records?. Hence, there should be a transparency of between user (client) and server side.
Possible countermeasures for this stage includes using a strong and secure servers that could be trusted enough to store user's biometric data. For better interaction and secure biometric system, it's efficient to implement challenge-response authentication system. Use of encrypted templates on biometrics's server side.

- **Communication link threats**: If the user's biometric data is sent in clear text or protected weakly, an eavesdropper could eavesdrop by listening in on communications between the server and client side. Such risk occurs if the server communicates with the client over the public networks, instead of encrypted and secure channel. The same risk applies if the communication happens with the capturing sensor and feature extractor. In this communication channel, attacks like MitM, replay attacks are possible.
Possible countermeasures for such vulnerabilities on communication link are to encrypt and secure the communication links. Also, the user should mutually authenticate themselves using symmetric or asymmetric keys. More security policies should be developed for server and client side to improve the non-integrated biometric system.

5.5 Summary

In this chapter, we have described the important part of the "trusted elements" in mobile devices. Biometrics is key factor in user-authentication that are recently developed mobile devices. Even with the technology, security of biometrics in mobile devices has been a topic of conspiracy. The important point in this chapter is about the types of biometric systems analyzed. Also, we have categorized biometric system in two forms: Integrated and non-integrated. Many researchers have their way of protecting biometrics in mobile devices. As discussed above, attacks on biometric system are either to steal the user's biometric data or to deter the system process.

Lastly, we performed a detailed risk analysis on the biometric systems that are also primarily categorized in integrated and non-integrated. The main goal of risk analysis is to understand the level of risk on every process in biometric systems as well as to analyze possible countermeasures.

Chapter 6

Authentication using FIDO Alliance

FIDO (Fast ID Online) Alliance[1] is a set of technology-agnostic security specifications for strong authentication in a computer-based and mobile-based specifications. The FIDO Alliance is divided into two main specifications: UAF (Universal Authentication Framework) and U2F (Universal 2nd Factor (U2F)). UAF is an industry standard for password-less authentication mechanism. It uses various types of biometrics (like, fingerprint, face, iris or human voice) as a mode of authentication. The first implementation on the smartphone of FIDO's UAF, which supports to meet up with the built-in device fingerprint scanner at payment service PayPal to authenticate is Samsung Galaxy S5[2]. U2F specification focuses on two-way factor authentication primarily using USB or NFC Devices. These technologies are based on similar security technology found in smart cards[3].
The purpose of this chapter is to analyse and understand the application of trusted elements. As discussed in the previous chapters, we studied the notion of trusted elements, with the need towards it. Hence, this chapter will analyse all the implementation techniques of trusted elements and also potential risks involved in it. For instance, the attack possible on Samsung S5's fingerprint was demonstrated. Additionally, FireEye plans to demonstrate hack that intercepts biometric data before it hits devices secure zone at upcoming RSA security conference[4].

The FIDO Alliance has a goal of changing the nature of web-based authentication by developing specifications that construct an open, scalable, interoperable set of policies [21]. FIDO remains a foundation for passwords to securely authenticate users while using online services. This new standards/specifications for (security) devices and browser plug-ins will enhance any website or cloud application to interface with a broad variety of existing and future enabled FIDO devices that will ensure better security [21]. This security is being ensured using more complex concepts that aim to provide trusted environment for online payment systems in the mobile devices. The security of the FIDO Alliance only ensures if the mobile device contains trusted element.

This chapter aims to study in detail about FIDO Alliance in Password-less authentication (UAF) but not second-factor based authentication (U2F) because this thesis contains biometrics as a trusted elements. According to FIDO's UAF architecture, following are the goals that need to be accomplished to address today's strong authentication issues and develop a smoothly-functioning low-friction ecosystem, a comprehensive, open, multi-vendor solution architecture is needed that encompasses [21]:

- User devices, whether personally acquired, enterprise-issued, or an enterprise trend as BYOD, and the device's potential operating environment. For instance, Home, Office, etc.

[1] https://fidoalliance.org/
[2] http://www.samsung.com/us/support/owners/product/SM-G900AZDAATT
[3] http://www.forbes.com/sites/amadoudiallo/2013/11/30/google-wants-to-make-your-passwords-obsolete/
[4] http://www.theguardian.com/technology/2015/apr/23/samsung-investigating-fingerprint-hack-galaxy-s5

- Authenticators which consists implementation of security tokens or authentication tokens.
- Relying on party applications and their deployment environments
- Meeting the needs of both end users and relying on parties
- Strong focus on both browser and native app-based end user experience

This solution architecture in UAF FIDO Alliance must feature following characteristics:

- FIDO UAF Authenticator discovery,
- Attestation, and provisioning
- Cross-platform strong authentication protocols leveraging FIDO
- UAF Authenticators
- A uniform cross-platform authenticator API
- Simple mechanisms for Relying Party integration.

The following section gives a step-by-step overview of the protocol architecture in UAF. The core UAF protocol is a conversation between UAF Client (mobile device) and UAF Server (server station, clouds) hosted by FIDO Alliance.

Following four steps are comprised of UAF protocol conceptually [21]:

1. **UAF Registration**: UAF allows the relying party to register a FIDO Authenticator with the user's account at the relying party. A FIDO's UAF client can only be able to register existing authenticators by that policy. Figure 6.2 refers to the users registering to UAF architecture with secure communication between server and client. A Relying Party can transparently detect when a user begins interacting with them while possessing an initialized FIDO UAF Authenticator [21]. In this initial introduction phase, the website will prompt the user when it detects the system authenticator that stores the password [21].

   Following are steps involved during the registration process of UAF authentication:

   (a) **Step 1**: During the process of online payment using mobile devices, the first step consists of initiating the application of online-payment on the mobile device. For instance, PayPal[5].

   (b) **Step 2**: After Step 1, user needs to fill the correct credentials for the online payment system and request for "Legacy Authentication". The online-payment system issues these credentials.

   (c) **Step 3**: After the request for the legacy authentication process and providing correct login credentials. Then in step 3, the server responds with the Registration request using the Authenticator[6] and also a FIDO device policy (consists of Terms and Regulations of FIDO Alliance).

   (d) **Step 4**: This step is a task of verifying a user and generation of new Key Pair for challenge-response authentication with server. According to UAF architecture, in this step user registers with biometrics authentication in the application that is connected to an online payment system. After the registration of the user's biometrics, the application then creates a new key pair and sent to server for challenge-response process.

[5] https://www.paypal.com

[6] Authenticator: Could be a device that is capable of storing the user's biometrics in a trusted environment. e.g., USB

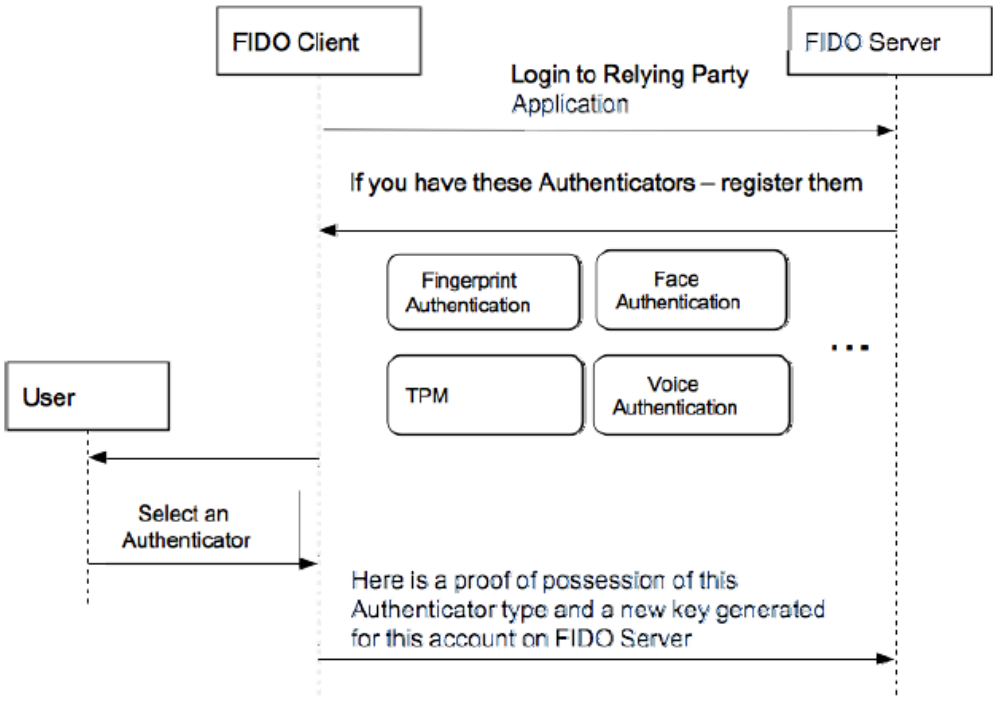

Figure 6.1: UAF Registration Protocol Flow [21]

(e) **Step 5**: After sending a challenge of Key Registration data to the server that includes following:

- Hash (Final Challenge)
- AAID (Authenticator Attestation ID)
- Public key
- KeyID
- Registration Counter
- Signature Counter
- Signature (Attestation key)

(f) **Step 6**: If implementations of all above steps are done correctly, then the server returns the success response with the confirmation of secure web-based online system.

2. **Authentication Process**: After execution of UAF Registration, the FIDO's UAF Authenticator will be subsequently employed whenever the user authenticates with the website (and the authenticator is present) [21]. Some authenticators may sample the use of biometric data such as a face image, fingerprint, voice-print or iris-scan. The use of biometric data depends on the integrated or non-integrated biometric installed on the mobile device. Following steps are involved in the authentication process of UAF protocol [21].

- **Step 1**: After the user selects goods or services on an online shop, then he/she needs to CheckOut[7] to purchase the product(s) or services.
- **Step 2**: During the CheckOut process, user needs to select FIDO Alliance's method of payment (e.g., PayPal). Choosing this payment method, would initiate an Authentication process to the Relying Party.

[7]A point at which goods/services are paid for, in a supermarket or similar store.

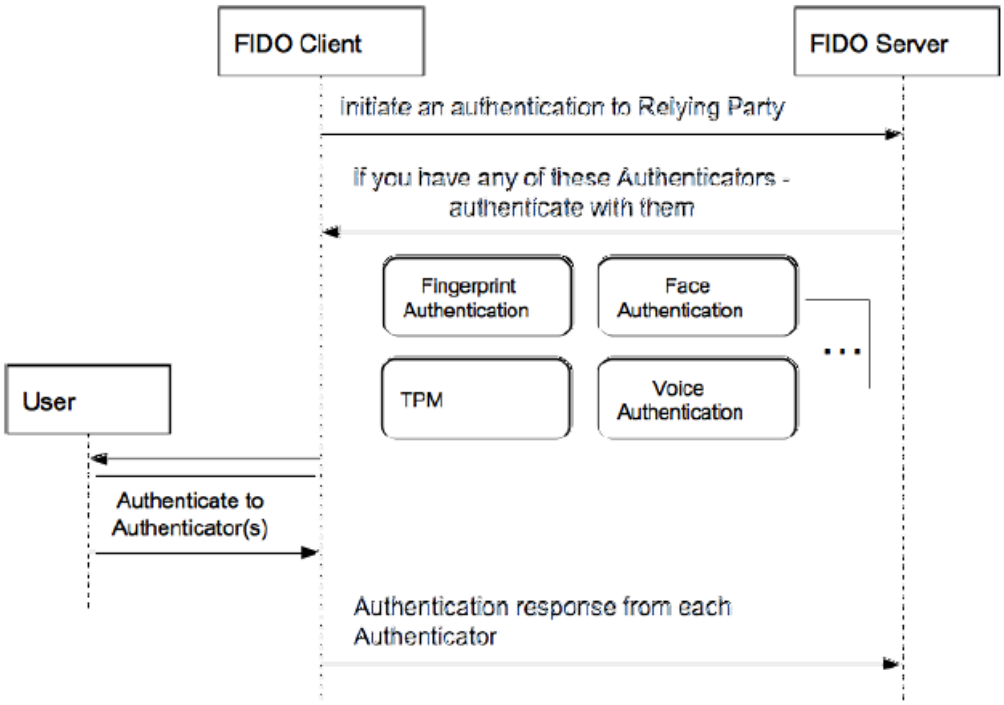

Figure 6.2: UAF Authentication Protocol Flow [21]

- **Step 3**: In this step, Relying party's server would send the Authentication request with a challenge. As discussed above, this is the first step in the Authentication process.
- **Step 4**: After the Challenge has been sent to the user's mobile device, user verifies his identity using any biometric method selected during the registration process as well as signs the challenge.
- **Step 5**: Hence, user's mobile device sends a Response with a Signed Challenge, Authenticator random, Signature Counter, Signature. If the valid response is validated correctly by the server, then the server returns the success command.

6.1 Implementation of FIDO's UAF Architecture

As discussed in the previous section, FIDO's UAF architecture is one of the secure web-based application procedure in the mobile devices. But, it's equally important to understand the implementation of UAF Architecture in mobile devices to ensure its security and compatibility. The FIDO protocol assumes that individual authenticator implementations will succumb over time to attacks; sensors will get spoofed, security hardware will get broken as attacks become more sophisticated [21]. The role of FIDO is to limit the impact of such attacks. FIDO also assumes that authenticators with superior security characteristics or more practical methods of authentication will come onto the market, and hence there will be a need to minimize the effort required to allow a relying party to adopt those new authenticators. FireEye demonstrated that it was possible to spoof the fingerprint sensor in Samsung's Galaxy S5[8]. The attack required physical control of the users device, and the ability to produce a sufficiently good dummy fingerprint that could fool the fingerprint sensor. Its important to realize what this means: unlike passwords, where a credential can be stolen

[8]http://www.planetbiometrics.com/article-details/i/2961/desc/fireeye-claims-samsung-galaxy-s5-fingerprint-vulnerability/

from one site (due to a server-side breach), and can be used by a remote hacker on any device to attack another site (if the user re-used the password across sites), even this failure mode shows that the FIDO approach results in an attack vector that is neither remotely exploitable, nor scalable across a large number of users. A successful attack on a device only gives a user the ability to attack a single user, unlike a password database breach, where every user is compromised.

6.2 Advantages of FIDO's UAF Architecture

FIDO is meant to boost interoperability among strong authentication devices and eliminate the need for multiple usernames and passwords [21]. The specifications define an open, scalable, interoperable set of mechanisms that supplant reliance on passwords to securely authenticate users of online services [21]. Security devices and browser plug-ins will allow any website or cloud application to interface with a broad variety of existing and future FIDO-enabled devices that the user has for online security, wrapping in biometrics, TPM and other technologies.

The protocols, which are based on public key cryptography, are categorized into two user experiences that support a wide range of scenarios. The UAF protocol enables the user to register a UAF-enabled device with a FIDO-ready server or website, authenticate their identity on their device with a fingerprint or PIN, for example, and log in to the server using a secure public key.

6.3 Summary

The goal of FIDO is that it enables the relying party to adopt risk and business appropriate authenticators, and adapt their selections over time [21]. If a particular implementation gets spoofed or hacked, then the relying party can change their server side policy, and disallow its use. If the vendor issues an update that addresses the problem, the relying party can require use of the new version of the authenticator. If new authenticators come on the market, they can choose to use them without having to replace their authentication infrastructure, rewrite business logic, or application code.

The FIDO protocol assumes that individual authenticator implementations will succumb over time to attacks; sensors will get spoofed, security hardware will get broken as attacks become more sophisticated. The role of FIDO is to limit the impact of such attacks. FIDO also assumes that authenticators with superior security characteristics or more usable methods of authentication will come onto the market, and hence there will be a need to minimize the effort required to allow a relying party to adopt those new authenticators.

Chapter 7

Conclusions

We have put forward the notion of trusted elements which should cover concepts like secure elements, Trusted Execution Environment (TEE) among other things. In this thesis, we have studied various candidates for trusted elements, ranging from Trusted Platform Module (TPM), Subscriber Identity Module (SIM) till Biometric authentication process. One of the advantage of trusted elements includes that it can host the applications securely, and it's also capable of creating a trusted communication layers towards the CPU in the mobile devices.

Firstly, we have described the Trusted Platform Module (TPM) in the mobile devices, also called Mobile-TPM (MTPM). The implementation of TPM is categorized into two parts: Hardware-based MTPM and Firmware-based MTPM. Hardware-based TPM is a primary implementation method to provide RoTs. As discussed in the previous chapters, TPM is capable of providing a substantial amount of assistance for the trust in mobile devices. But, Hardware-based MTPM only provides Isolated Execution Environment (IEE) but not TEE. It also has several shortcomings that are related to deployment costs and computational overheads. We also described the different components and services provided by a TPM for creating a trusted platform (both hardware and firmware). Alternatively, Firmware-Based MTPM provides TEE using ARMs TrustZone Technology. One of the advantages is that deployment of Firmware-MTPM can be done in less-costly methods and also it also provides TEE.

Secondly, we then explained Subscriber Identity Module (SIM) cards that are a subset of smart cards. SIM Card is a tamper-resistant smart card that is responsible for providing confidentiality, integrity and other security capabilities. But recently, several security flaws have been detected in SIM Cards. For instance, an attacker can gain root access to the SIM Cards due to the flaws in SIM's cryptographic policies. This leads to an untrusted communication between SIM and the mobile device. Alternatively, UICC is a new generation of SIM card, which ensures more security for data integrity, availability, confidentiality and authentication towards the mobile devices. Unlike traditional SIM Cards, UICC stores cryptographic keys, personal identification (in an encrypted form). Therefore, we consider UICC as a trusted element.

Furthermore, we described biometric authentication in mobile devices. We consider biometrics as a trusted element. Therefore, we examined that how secure is this trusted element (biometric authentication). We studied various types of biometrics such as fingerprint, facial, voice and iris recognition as these are most commonly deployed on mobile devices. We then categorized biometric authentication in two parts: Integrated and non-integrated. Integrated is considered more secure as all processes are executed on the mobile device, and also user's biometric data resides on a mobile device. In the non-integrated biometric system, usually the user's biometric data is stored in application server at a remote location. Therefore, the application, communication channel and server itself have to be trusted. Vul-

nerabilities can be detected in both integrated and non-integrated biometric system with corresponding countermeasures, as discussed in Section 5.4. Many mobile manufacturers prefer to Therefore, biometric authentication system could be a trusted element depending on its implementation in mobile devices.

Finally, we then describe a practical implementation of FIDO's UAF (User Authentication Framework) protocol. UAF is an industry standard for password-less authentication mechanism that implements trusted elements in its architecture. It uses various types of biometrics (like, Fingerprint, face, iris or human voice) as a mode of authentication and other trusted elements like TPM to store the user's biometric data. The reason is that its not just important to secure the biometric templates, but it's also important to be able to perform comparison operations against those templates in a secure area. One of the primary goals of FIDO's architecture is to reduce the impact of such attacks on the mobile devices. We assume that trusted elements in mobile devices are secure to analyse which other elements are secure? As discussed in Chapter 6, FireEye demonstrated that it was possible to spoof the fingerprint sensor in Samsung's Galaxy S5 due to poorly protected sensors in mobile devices. Therefore, weakest link in the authentication process should be primarily secured (e.g., sensors).

7.1 Future Work

Firstly, the risk analysis is performed on the qualitative research basis of biometric authentication. As we discussed in previous section, finding the weakest element in any process is an important factor. Therefore, for identifying weakest element in the system, the better quantitative risk analysis is required.

For future work, further analysis is required to find which other trusted elements could be added in the mobile devices to make it more secure. Also, trusted elements that are recently introduced in smart television (e.g., PayTV cards, biometric authentication) should perform the security analysis.

Bibliography

[1] Biometrics intro. https://www.pcas-project.eu/images/Deliverables/PCAS-D3.1.pdf. Accessed: 2015-09-28. 7, 27, 34, 35

[2] Building a Secure System using TrustZone Technology. http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf. Accessed: 2015-09-28. 13

[3] Face-Recognition. http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2012/wp192_en.pdf. Accessed: 2015-09-30. 32, 33

[4] Introduction to Biometric and Applications. http://users.ece.cmu.edu/~jzhu/class/18200/F06/L10A_Savvides_Biometrics.pdf. Accessed: 2015-10-18. 27

[5] TCG TPM specification 2.0, 2013. http://www.trustedcomputinggroup.org/resources/trusted_platform_module_specifications_in_public_review. Accessed: 2015-07-24. 6, 9

[6] Samsung's response. http://www.franken.senate.gov/files/letter/140718SamsungResponse.pdf, 2014. Accessed: 2015-10-25. 29

[7] Biometrics Touch-ID. http://www.apple.com/business/docs/iOS_Security_Guide.pdf, 2015. Accessed: 2015-09-28. 30

[8] TCG Mobile Reference Architecture Version 1.0 Revision 1. https://www.trustedcomputinggroup.org/specs/mobilephone/tcg-mobile-reference-architecture-1.0.pdf,12, June 2007. Accessed: 2015-07-24. 9

[9] Raja Naeem Akram, Konstantinos Markantonakis, and Keith Mayes. An Introduction to the Trusted Platform Module and Mobile Trusted Module. In *Secure Smart Embedded Devices, Platforms and Applications*, pages 71–93. Springer, 2014. 6

[10] Anders Andersen and Arne Munch-Ellingsen. Mobile device security: the role of NFC, UICC and secure elements. *Norsk informasjonssikkerhetskonferanse (NISK)*, 7. 21, 22, 24, 25

[11] Lily Chen, Joshua Franklin, and Andrew Regenscheid. Guidelines on Hardware-Rooted Security in Mobile Devices (draft). *NIST Special Publication*, 800:164, 2012. 10, 11, 17, 18

[12] Liqun Chen, Siani Pearson, and Athanasios Vamvakas. A Trusted Biometric System. *HP Laboratories Technical Report HPL-2002-185. July 15th*, 2002. 27, 28

[13] Ivan Cherapau, Ildar Muslukhov, Nalin Asanka, and Konstantin Beznosov. On the Impact of Touch ID on iPhone Passcodes. In *Symposium on Usable Privacy and Security (SOUPS)*, 2015. ixix, 30, 31

[14] PP Chitte, JG Rana, RR Bhambare, VA More, RA Kadu, and MR Bendre. IRIS recognition system using ICA, PCA, Daugman's Rubber Sheet Model Together. *International Journal of Computer Technology and Electronics Engineering*, 2(1):16–23, 2012. 35

[15] Sansar Choinyambuu. A Root of Trust for Measurement, Mitigating the lying endpoint problem of TNC. 2011. 1

[16] Nathan L Clarke, Steven M Furnell, and Paul L Reynolds. Biometric authentication for mobile devices. In *Proceeding of the 3rd Australian Information Warfare and Security Conference*, pages 61–69, 2002. 33

[17] Kurt Dietrich and Johannes Winter. Implementation aspects of mobile and embedded trusted computing. In *Trusted Computing*, pages 29–44. Springer, 2009. 6

[18] Lasse Edlund. *Secure and confidential application on UICC*. Skolan för datavetenskap och kommunikation, Kungliga Tekniska högskolan. 22

[19] Peter Edsbäcker. SIM cards for cellular networks: An introduction to SIM card application development. 2011. 20, 21

[20] Jan-Erik Ekberg et al. Securing Software Architectures for Trusted Processor Environments. 2013. 1

[21] UAF FIDO. README: GUIDE TO DOCS: FIDO UAF Review Draft Spec Set. ixix, ixix, 43, 44, 45, 46, 47

[22] Javier Galbally, Julian Fierrez, Javier Ortega-Garcia, Chris McCool, and Sebastien Marcel. Hill-climbing attack to an eigenface-based face verification system. In *Biometrics, Identity and Security (BIdS), 2009 International Conference on*, pages 1–6. IEEE, 2009. 36

[23] Ming Gao, Xihong Hu, Bo Cao, and Dianxin Li. Fingerprint sensors in mobile devices. In *Industrial Electronics and Applications (ICIEA), 2014 IEEE 9th Conference on*, pages 1437–1440. IEEE, 2014. 30

[24] TRUSTED COMPUTING GROUP et al. TPM main, part 1 design principles. *Specification Version*, 1:2003–2011, 2007. 6

[25] Rajeev Gupta and Ashok Kumar. Noisy Iris recognition & Its Importance. *Journal of Ultra Scientist of Physical Sciences International Journal of Physical Sciences*, 25(2):229–234, 2013. 36

[26] Song-Yi Han, Hyun-Ae Park, Dal-Ho Cho, Kang Park, and Sangyoun Lee. Face recognition based on near-infrared light using mobile phone. *Adaptive and Natural Computing Algorithms*, pages 440–448, 2007. 35

[27] Y. Ijiri, M. Sakuragi, and Shihong Lao. Security Management for Mobile Devices by Face Recognition. In *Mobile Data Management, 2006. MDM 2006. 7th International Conference on*, pages 49–49, May 2006. 33

[28] Henry Irish and Trinity Hall. Using the SIM as a Trusted Element to Secure the Mobile Web. 2013. 5, 15

[29] Anil K Jain, Arun Ross, and Salil Prabhakar. An introduction to Biometric recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(1):4–20, 2004. 27, 28

[30] Wayne Jansen. Authenticating users on handheld devices. In *Proceedings of the Canadian Information Technology Security Symposium*, pages 1–12, 2003. 32

[31] Dae Sik Jeong, Hyun-Ae Park, Kang Ryoung Park, and Jaihie Kim. Iris recognition in mobile phone based on adaptive gabor filter. In *Advances in Biometrics*, pages 457–463. Springer, 2005. 35

[32] Tsuyoshi Kawaguchi and Mohamed Rizon. Iris detection using intensity and edge information. *Pattern Recognition*, 36(2):549–562, 2003. 35

[33] P. Kristoffersen. Secure storage of user data in UICC and Smart Card enabled devices, November 4 2009. EP Patent App. EP20,080,103,770. ixix, 22, 23, 24, 25

[34] Mrs U Latha and K Rameshkumar. A Study on Attacks and Security Against Fingerprint Template Database. *International Journal of Emerging Trends Technology in Computer Science (IJETTCS)*, 2, 2013. 37

[35] Marcos Martinez-Diaz, J Fierrez-Aguilar, Fernando Alonso-Fernandez, Javier Ortega-García, and JA Siguenza. Hill-climbing and brute-force attacks on biometric systems: A case study in match-on-card fingerprint verification. In *Carnahan Conferences Security Technology, Proceedings 2006 40th Annual IEEE International*, pages 151–159. IEEE, 2006. 36

[36] Kathleen N McGill. Trusted mobile devices: Requirements for a mobile trusted platform module. *Johns Hopkins APL Technical Digest*, 32(2):544, 2013. 10

[37] Pardis Pourghomi and Gheorghita Ghinea. Cloud-based NFC Mobile Payments. 2013. xixi, 21

[38] Wolfgang Rankl and Wolfgang Effing. *Smart card handbook*. John Wiley & Sons, 2010. 6

[39] Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM systems Journal*, 40(3):614–634, 2001. 40

[40] Marie Reveilhac and Marc Pasquet. Promising secure element alternatives for NFC technology. In *Near Field Communication, 2009. NFC'09. First International Workshop on*, pages 75–80. IEEE, 2009. 21, 22

[41] Arun A Ross, Jidnya Shah, and Anil K Jain. Toward reconstructing fingerprints from minutiae points. In *Defense and Security*, pages 68–80. International Society for Optics and Photonics, 2005. 39

[42] Thomas Rossow. TPM 2.0, UEFI and their Impact on Security and Users Freedom. 2013. 9

[43] Paul E Sevinç, Mario Strasser, and David Basin. Securing the distribution and storage of secrets with trusted platform modules. In *Information Security Theory and Practices. Smart Cards, Mobile and Ubiquitous Computing Systems*, pages 53–66. Springer, 2007. 10, 17, 18

[44] R Shirey. RFC 4949–internet security glossary, 2007. 5

[45] Diego A Socolinsky and Andrea Selinger. A comparative analysis of face recognition performance with visible and thermal infrared imagery. Technical report, DTIC Document, 2002. 35

[46] Frank Stajano. Pico: No more passwords! In *Security Protocols XIX*, pages 49–81. Springer, 2011. xixi, 15, 16

[47] Stefan Thom, Jeremiah Cox, David Linsley, Magnus Nystrom, Himanshu Raj, David Robinson, Stefan Saroiu, Rob Spiger, and Alastair Wolman. Firmware-based trusted platform module for arm processor architectures and trustzone security extensions, February 12 2013. US Patent 8,375,221. 12, 13, 29

[48] Stephen J Tipton, Daniel J White II, Christopher Sershon, and Young B Choi. iOS Security and Privacy: Authentication Methods, Permissions, and Potential Pitfalls with Touch ID. 28

[49] Shari Trewin, Cal Swart, Larry Koved, Jacquelyn Martino, Kapil Singh, and Shay Ben-David. Biometric authentication on a mobile device: a study of user effort, error and task disruption. In *Proceedings of the 28th Annual Computer Security Applications Conference*, pages 159–168. ACM, 2012. 27, 28

[50] Elaheh Vahidian. Evolution of the SIM to eSIM. 2013. ixix, 19, 20

[51] Kai Xi, Tohari Ahmad, Fengling Han, and Jiankun Hu. A fingerprint based bio-cryptographic security protocol designed for client/server authentication in mobile computing environment. *Security and Communication Networks*, 4(5):487–499, 2011. 39

[52] Shijun Zhao, Qianying Zhang, Guangyao Hu, Yu Qin, and Dengguo Feng. Providing Root of Trust for ARM TrustZone using on-chip SRAM. In *Proceedings of the 4th International Workshop on Trustworthy Embedded Devices*, pages 25–36. ACM, 2014. 12