ABSTRACT

In today's world of advanced Internet, mobility is a key service. The use of smart phones, tablets and wearable devices is on the rise at fast pace. However, traditional software-based security solutions are unable to deliver the satisfactory level of protection and security assurances to the mobile device users especially in enterprise, government and military. In recent past, mobile industry has made efforts to standardize the security specifications. The standards from National Institute of Science and Technology (NIST) and Trusted computing Group (TCG); Trusted Platform Module (TPM) 1.2, Trusted Platform Module 2.0 and Mobile Trusted Module; have been analyzed and found that software based mobile security standards are unable to provide the basis of strong foundation required to implement mobile security as compared to the hardware rooted security mechanisms. Therefore, the standards for mobile devices need to be revised to overcome the inherit shortcomings. It was found that majority of the mobile security solutions were based on ARM System-on-Chip (SoC) that offer TrustZone security architecture with the vendors' specific solutions mounted on it. The integrated solution of ARM TrustZone does not comply with the available standards, and hence, several security vulnerabilities have also been reported. As all the hardware rooted security solutions are vendor specific, closed form and non-standardized a new mobile security model mobile Trusted Platform Module (mTPM) has been proposed. An effort has been done to comprehensively cover conceptual framework over existing standards and their corresponding implementation methodology. mTPM suggests the hardware rooted security implementation technique on the existing ARM TrustZone security technology while overcoming its shortcoming especially pertaining to lack of secure hardware peripherals including establishing the integrity of various roots of trust for processing, storage, entropy source, clock, and access to firmware. Fundamentally it could be considered as embedding a TPM hardware device in ARM SoC by suitable augmenting the existing architecture with additional hardware and software resources. It is hoped that the proposed mTPM model will provide a unified, vendor neutral and standardized security platform for the mobile device manufacturers. However, it is felt that the whole

concept should be subjected to physical testing and evaluation on a test bed through fabrication of prototype SoC.

DECLARATION

I hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification either at this institution or elsewhere.

_____

(Naveeda Ashraf)

ACKNOWLEDGMENTS

All praises to Allah for the strengths and His blessing in completing this thesis.

I would like to convey my gratitude to my supervisor, Dr. Rabia Latif, for her supervision and constant support. Her invaluable help of constructive comments and suggestions throughout the experimental and thesis works are major contributions to the success of this research.

I would also like to express my sincerest appreciation to Lec Waleed Bin Shahid and Lec Narmeen Shafqat for being an important part of my Research Supervisory Committee. Their scholarly guidance, assistance and knowledge have been meaningful for successful completion of my research.

I extend my deepest gratefulness to my father, Dr. Ashraf Masood, who provided me a platform and gave me the liberty to work in the area of my interest and extended his tremendous support prior to as well as during the course of this research. His technical guidance, encouragement, ideas and perspective were vital for completion of this tedious task. His support gave me confidence and helped me to understand about the subject matters deeply and inspired me towards my goals.

Last, but not the least, I am highly thankful to my mother (Mrs. Perveen Ashraf), my husband (Mr. Zeeshan Siddique) and my parents-in-law. They have always stood by my dreams and aspirations and have been a great source of inspiration for me. I would like to thank them for all their care, love and support through my times of stress and excitement.

Finally, I am grateful and thankful to Military College of Signals and National University of Sciences and Technology for providing me a chance to help achieve excellence by being associated with the prestigious institutions.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

API	Application Programming Interface
BYOD	Bring Your Own Device
BBS	Blum Blum Shub
DRM	Digital Management Rights
EH	Endorsement Hierarchy
IoT	Internet of Things
MDM	Mobile Device Management
MTM	Mobile Trusted Module
NIST	National Institute of Science and Technology
NH	Null Hierarchy
PEE	Policy Enforcement Engine
PH	Platform Hierarchy
ROT	Root of Trust
RNG	Random Number Generator
ROTI	Root of Trust of Integrity
ROTM	Root of Trust of Measurement
ROTR	Root of Trust of Reporting
ROTS	Root of Trust of Storage
ROTV	Root of Trust of Verification
RPMB	Replay Protected Memory Block
RTC	Real Time Clock
Sclk	Secure Clock
SCR	Secure Configuration Register
SES	Secure Entropy Source
SH	Storage Hierarchy
SMC	Secure Monitor Call
SoC	System on Chip
TBS	Trusted Base Services

TCG	Trusted Computing Group
TEE	Trusted Execution Environment
TPM	Trusted Platform Module
TRNG	True Random Number Generator
TZAPI	TrustZone Application Programming Interface

Chapter 1

INTRODUCTION

1.1 Introduction

Since the birth of mankind efforts had been made to invent devices which help in computing. The first known tool was abacus invented for the arithmetic tasks by the Babylon early in 2400 B.C. As the time passed technology enhanced and Charles Babbage invented the first mechanical computer in 19th century and originated the concept of programmable computer. In the first quarter of 20th century various sophisticated analog computers were made using direct mechanical or electrical model for computation. In 1944 the first electronic digital programmable computer was made named Colossus used by the Germans in World War 2. The concept of modern computing was given by Alan Turning in 1936. Since then technology advancements took place and vacuum tubes were replaced by transistors and then integrated circuits. With the invention of integrated circuits a revolution took place in the computing market and the first desktop computer was invented in 1971 named KENBAK-1. With the persistent miniaturization of computing resources and improvement in portable battery life, portable computers grew into popularity in last decade of 20th century. The need of portable computing encouraged the manufacturers to integrate the computing resources into cellular phones and now in different wearable and IoT.

With the increased pace in the development of computing devices the need to secure the computational data also increased. In 2003 Trusted Computing Group (TCG) took the first step to standardize the security implementation and gave the specifications for a Trusted Platform Module (TPM). In 2006 the computers were started to introduce the embedded TPM chips and built in security. Since then the standards as well as computing devices were modified with the time with enhanced security assurances to provide the security capabilities of confidentiality, integrity and availability. The question of security became complex when networking came into existence and it became even worse with the introduction of mobile computing.

As dependence on mobile technology is increasing, the employees tend to use personally-owned and organization-issued mobile devices simultaneously to utilize corporate data, resources and services for perform different activities. But unfortunately, mostly theses mobile devices especially personally owned are unable to provide strong security assurances to the organizations and end users. Besides the laptops and other such devices provide a hardware rooted security which lack in present mobile devices. Rooting and jail breaking are the common vulnerabilities present in mobile devices, which although provide the device users with greater flexibility and control over the devices but also bypass important security features and thereby introduce more threats and vulnerabilities. Enterprises have to accept these security risks present in the mobile devices because of several factors which include cost savings and employee desire for greater convenience.

The analysis of mobile attacks has cleared the importance of hardware based security. Some of the observations are based on the fact that security solutions are implemented most often in software. Also, the increasingly popular use of virtualization technologies to manage security in isolated environments or the software-based security offered through anti-virus or anti-theft applications are not able to prevent waves of advanced persistent attacks and thus security has to live underneath the operating system and be further assisted by the system hardware

With the increased utilization of smart connected devices mainly tablets and mobile phones, have fundamentally transformed our life styles where we now can access personal networks, bank accounts and business documents wherever and whenever required. In order to take full benefit of the richness and connectivity of these devices, there is a need to control the associated risks. This need activated to emerge one key platform Trusted Computing Group's Mobile Trusted Module with the other key platform Global Platform's Trusted Execution Environment. These two technologies work together in a unified manner called TPM MOBILE to provide security, peace of mind and enhanced services to its consumers. In 2012 NIST published SP 800-164 and took the step to standardize the basic requirements to harden the core of mobile devices. Hence all mobile devices should meet these standards as a primitive.

Although implementing good security hygiene and security tools provide efficiency and security and are a vital part of organization's security policy, even the best practices can be bypassed as users always remain the most commonly leveraged attack vector. Mobile industry is working to implement measures that "harden" the mobile devices, and embedding security into the core of mobile devices. Some of the leading solutions provided by different companies in the area of hardening include Qualcomm, Intel, Samsung Knox, Apple and Boeing. All the vendors provide their own closed form solutions based upon ARM Trust Zone security technology

The use of smart phones, tablets and wearable devices is on the rise at fast pace in enterprise, government and military. However, traditional software-based security solutions are unable to deliver the satisfactory level of protection and security assurances to the mobile device users. Higher level of security for such applications can only be ensured through hardware mechanisms. This research will focus on securing the mobile device through hardware rooted security and adaptation into such low power and resource constraint devices. Moreover, based on improved standards and practically feasible implementation technology, a hardware security solution will also be proposed for Android smart phones.

1.2 Problem Statement

In context to the above discussion the security of the mobile devices is not fool proof due to the following reasons:

- The software of mobile devices including its operating system is vulnerable to penetration by the attackers
- Hardware solutions proposed or implemented till date are non-compliant to standards, ad-hoc, vendor specific and closed form solutions.

1.3 Research Objective

The main objective of this thesis was**:**

- Analyze various Trusted Platform Module (TPM) standards and specifications available for security in desktop, laptops and other computing devices for adoption as primitives of MTM.

- Carry out a survey to analyze the extent of success of commercial implementation of MTM in android based mobile device by various vendors.
- Propose a hardware security solution based on improved standards and practically feasible implementation technology.

1.4 Research Methodology

Figure 1.1 illustrates the research methodology which will be adopted during the research work.

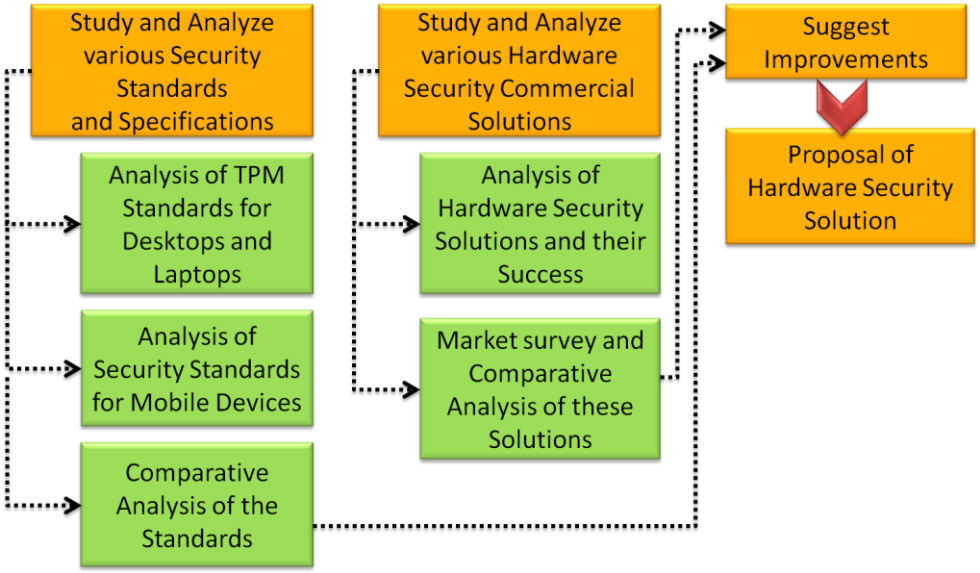

Figure 1.1: Research Methodology

The research will be started with the analysis of security standards available. This will include the analysis of the standards employed in static computing devices such as laptops and desktops. Afterwards the mobile security standards will be analyzed. This will include the understanding of NIST standards and the TCG standards. Afterwards a comparative analysis of all these standards will be carried out. Moreover the limitations present in standard will also be notified.

The second part of research will be to analyze the available commercial solutions. This will include the study and analysis of the hardware rooted security commercial solutions and the extent to which they comply with the standards. Afterwards a

comparative analysis of these solutions will be carried out and the limitations in their security implementation strategy will be notified.

The third part of research will be to propose improvements in the standards limitations. Also a practically implementable hardware security solution will be proposed based on the improved standards and specifications.

1.5 Significance for Industry and Military

Use of mobile devices by the military, diplomatic and other security agencies has increased rapidly since the last decade. In the absence of standardized solution, customized products with ad-hoc security system implementation are usually deployed. However, these devices are neither cost effective nor provide the necessary security assurances for military organizations and miss the requisite layer of security and ruggedized hardware to optimize it required for military purposes. Hence the proposed solution will discriminate itself from the available solutions by providing a standardized, vendor neutral, low cost and low power solution with reduced die size which are the main security constraints while incorporating hardware rooted security in mobile devices.

1.6 Thesis Contribution

It is said to the best of our knowledge that proposed model has not been published in any paper and is solely presented after our own research. Moreover the analysis and limitations of standards and ARM TrustZone mentioned during has not been done previously in any research studies.

The contribution of thesis in academics and industry will be as follows;

- In academics it will contribute towards the documented detailed analysis of the hardware rooted mobile security standards and their comparative analysis. Furthermore the documented analysis of the commercial solutions and their analysis will be a great add on and help for other researchers. Moreover the detailed analysis carried out for ARM TrustZone and its limitations will contribute towards boosting more research solutions for the industry.
- In industry it will contribute towards the analysis of different solutions and the extent to which they comply with the available standards. Moreover it will boost

the industry to modify the current standards based on the limitation notified during the thesis. The proposed solution will provide all the manufacturers with a unified and standardized security platform to build their secure mobile devices making the solutions vendor neutral. Moreover the proposed solution will facilitate the developers as the security solution is complete as a whole and is an open source solution.

- A survey paper on the topic "Analytical Study of Hardware-Rooted Security Standards and its Implementation Techniques in Mobile Devices", has also been accepted in 6th "International Conference On Identification, Information And Knowledge In The Internet Of Things (IIKI)", held in Qufu, China on 20-22 October217. IIKI 2017, is the sixth conference in the series which provides a dedicated forum for international experts to discuss current trends, challenges, and state-of-the-art solutions in the Internet of Things.
- A regular paper on the topic "Analytical Study of Hardware-Rooted Security Standards and its Implementation Techniques in Mobile Devices", has also been accepted in the 15th International Conference on Information Technology - New Generations (ITNG), being held in Nevada, Las Vegas, USA on 16-18 April 2018

1.7 Thesis Organization

The thesis is structured as follows:

- **Chapter 2** discusses different hardware rooted security standards available to the industry to develop secure mobile devices. The comparative analysis of these standards is carried out and the shortcomings present in them have been reported.
- **Chapter 3** discusses the key components of a TPM and its implementation techniques. Moreover different commercial solutions have been discussed in this chapter. The ARM TrustZone architecture is analyzed deeply and how different security components and mechanisms are deployed in it. Moreover the shortcomings analyzed in the ARM TrustZone are which are supported by the list of exploits of TrustZone gathered different internet sources.

- **Chapter 4** suggests the proposed hardware rooted security solution. It consists of two parts. First is the suggested modification of the limitations in the mobile standards. Second is the suggested security model which is based on the modifications in the shortcomings of ARM TrustZone security model. Some more enhancements to implement the modifications are also the part of this chapter.
- **Chapter 5** discusses the implementation mechanism and feasibility of the proposed hardware rooted security model mTPM. Moreover it also analysis the model for the compliance with the existing and modified standard and will carry out a comparative analysis of the security features inherit in ARM Trust Zone and proposed mTPM.
- **Conclusion** will summarize the whole research work and will notify the conclusions drawn during the thesis.
- **Future work** will end the thesis while highlighting the academic and industrial importance and more research directions in accordance to this thesis.

1.8 Conclusion

The objective and motivation to conduct the research on hardware rooted security in mobile devices has been described in this chapter. The research methodology developed during the research is also mentioned. Its importance for academics, industry and military has also been highlighted. At the end it describes the overall structural organization of the thesis.

Chapter 2

HARDWARE ROOTED SECURITY STANDARDS AND THEIR ANALYSIS

2.1 Introduction

In today's world of advanced internet, mobility is a key service. Therefore, mobile devices, such as smart phones and tablets, should support primary security objectives; confidentiality, integrity, and availability such that the mobile devices are secured against variety of the new and advanced threats. The laptops and other such devices provide a hardware-rooted security which is now expected to be available in the current mobile devices such as smart phones, tablets and various types of wearable.

To provide the baseline for hardware rooted security architecture in mobile devices some standards were developed. Hence in this chapter we will discuss different security standards developed for the mobile security purpose and their analysis will be carried out. Firstly we will discuss the standard developed by NIST. Afterwards the specifications given by TCG for laptops and mobile security platform will be highlighted. The TPM MOBILE security model will be discussed in detail. The chapter will end with the analysis of the available security model.

2.2 NIST Hardware Rooted Security Architecture

The previous discussion has cleared the importance of the requirement of hardware security in mobile devices. But many mobile devices are deficient in built-in secure hardware roots of trust. In 2012 NIST published SP 800-164 and took the step to standardize the basic requirements to harden the core of mobile devices. This section will in detail explain the NIST's hardware rooted security architecture. [3]

2.2.1 Trusted Security Components:

According to NIST the following three trusted core components should be developed in the mobile devices to form a hardware rooted security mobile device. Verification of the set of security components to provide security capabilities for personal

and bring your own device (BYOD) or company-issued device is required. These security components are:

1. **Roots of Trust (ROT):**

ROTs provide the foundation of trust assurance on the mobile devices. These ROTs can be deployed in hardware, software or firmware to provide the set of the trusted security critical functions. Hardware ROTs have more reliable behavior as compared to software ROTs and provide better immutability and smaller attack surfaces. Beside, software ROTs offer the advantage of fast deployment on diverse platforms. ROTs are trusted to perform such security-critical functions as software verification, cryptographic key protection, device integrity and device authentication, and behave in a trusted and predictable manner because their error cannot be identified. They are expected to discourage or prevent hackers from accessing the firmware when a mobile device is powered on. They also provide an evidence for hardware security foundation for Trusted Execution Environments (TEEs). Each ROT can be graded and evaluated according to the level of security it has to provide to the system. The devices should implement the following ROTs specified by NIST guidelines to provide the key security capabilities in the mobile devices:

- **Root of Trust for Storage (ROTS):**

  There should be a secure interface and repository to manage and store the cryptographic keys and other critical security parameters including the processing of policy details. It is preferred to implement ROTS in hardware as it should typically contain the cryptographic capabilities and keys which are used by the ROTS which are confined to its own logical boundary and not allowed to be accessed in plaintext by any other part of the system.

- **Root of Trust for Verification (ROTV):**

  A secure interface or engine should be present for digital signatures verification of all the applications and mechanisms and generate assertions according to the outcome. It also executes the algorithms of signature verification and accesses

the key store to verify the digital signatures. The keys may be stored in the internal memory of ROTV or it may request ROTS for the services.

- **Root of Trust for Integrity (ROTI):**

  NIST has identified this new ROT for integrity, which was not previously present in other standards. The device should provide an isolated secure interface, secure storage and integrity protection to store and manage the assertions. Tamper resistant locations are present to securely store the measurements and its assertions. The protected interface and the tamper resistant locations together form the ROTI.

- **Root of Trust for Reporting (ROTR):**

  This ROT provides a secure environment and interface to generate device integrity reports by identifying, managing and signing the assertions. It supplies the information after binding it cryptographically with its entity. ROTR provides the capability of integrity and non-repudiation of the device integrity reports.

- **Root of Trust for Measurement (ROTM):**

  ROTM offers trusted measurement functionality that is used by the assertions, attested by ROTR and protected via ROTI. It has the ability of reliable integrity measurements and establishes a ROT chain of transitive measurement components. The later the ROTM is called, the more the adversary gains the opportunity to weaken the measurement trust chain.

2. Application Programming Interface (API):

The API's expose the ROT's to the platforms so that OS and applications can have high level of security assurance. Mobile OS use the features offered by the ROTs to generate and store device integrity reports, measure and verify software and firmware, and protect locally stored authentication credentials, cryptographic keys and various sensitive data. This interface offers the professionals a set of security features which will secure their applications and protect their processing data. The layer of APIs reduces the burden on application developer to utilize the security trusted components of the ROTs and the OS without bothering about how to implement low level security features. Hence the APIs should be standardized for the mobile devices to bring all the developers on a

unified platform and to use these security capabilities across a broad range of devices. Applications request the APIs for the ROTs services of data protection through encryption, device integrity reports and to store or recover authentication credentials and other similar data.

3. The Policy Enforcement Engine (PEE):

PEE is generally the part of mobile OS. It imposes policies on the device in accordance with other device components and allows maintenance, management and processing of policies on both the Information Owner's and device environments. The PEE offers the Information Owners with the capability of control over their information. The PEE makes the policies based on the Information Owners requirements and enforce them while sharing information and storing within the device and across network. In the case of an un-resolvable conflict this engine notifies the device owner and enforces a default policy which denies the unauthorized access of data until the error is resolved.

2.2.2 Key Security Capabilities:

In order to meet the NIST criterion the mobile devices should implement the following three key data security capabilities:

1. Device Integrity:

Device integrity refers to the nonexistence of corruption in the hardware, software or firmware of a device. A mobile device provides an evidence of secure execution and device integrity if its configurations can be shown to be in a trusted state. Counterfeited devices will not connect, store data or run applications.

2. Isolation:

Isolation refers to the ability of the system to keep different data components, applications and mechanisms separate from each other and hence control the flow of information from one process to another. In mobile devices isolation is required between different layers of architectural contexts and assurance that no applications interfere in the process of other application. The isolation mechanisms make use of the assertions generated by the ROTs to establish the required secure environment.

3. Storage Protection:

Storage protection refers to preserve the confidentiality and integrity of data at rest and in transit and upon access revocation. Protected storage primarily depends on encryption algorithms used to authenticate credentials of authorized users for integrity and protection of data and the associated keys. The most prominent risks offered while establishing secure storage include exposure of the secure keys in an insecure domain or insecure storage of those keys. The keys must also be stored securely during and after processing of encryption or decryption algorithms, integrity protection and digital verification algorithms. It should be standardized practice of the application developers to request the services of RTS to use and store the encryption keys. Protected storage and key protection mechanisms collectively ensure secure authorization for retrieval of keys and minimize exposure of keys to unsecure area. Protected storage also provides confidentiality and integrity of data by cryptographic means and controlling the access of unauthorized users, processes or devices through logical or physical means. Physical protection includes restricted access of keys by authorized entities to permitted operations. Logical protections include access policies implemented in firmware or software permitting authorized access.

2.2.3 Mobile Device Security Architecture:

Similar to other computing devices, mobile device architecture comprises of a stack of hardware, software and firmware. The higher level of the stacks depend on the lower levels for various services and interfaces and consider them as trustworthy as they have limited access of those levels. . Figure 2.1 describes the architecture of mobile devices. [3]

1. Hardware Layer:

This is the lower most layer of the architecture and includes the hardware components of the device. These components are either deployed by the chipset or device manufacturer of the mobile device. The security components operating at this level are vital and serve as the foundation of trust for higher levels of the stack.

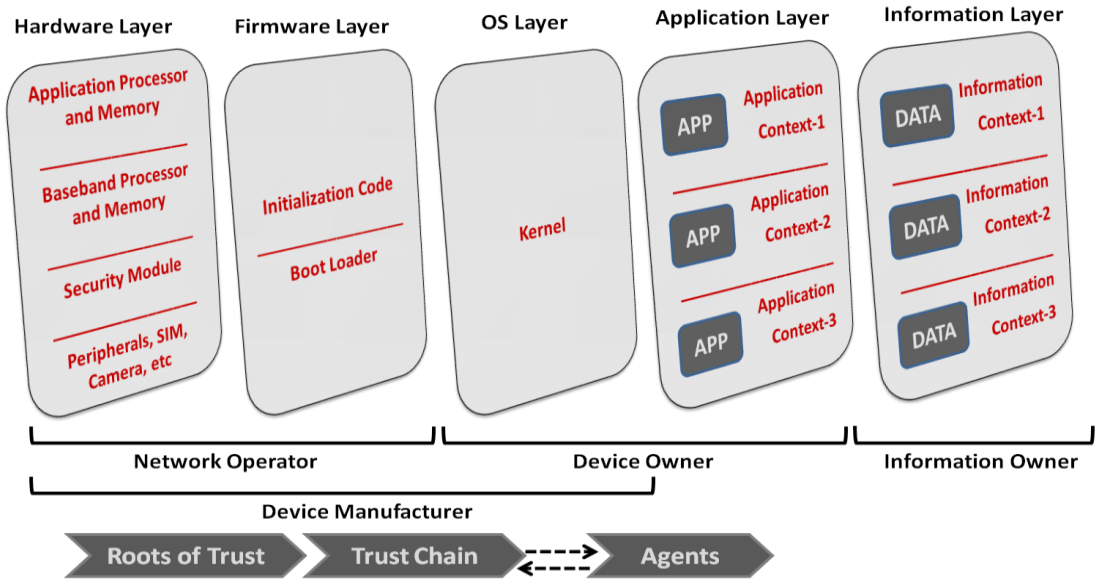

Figure2.1: NIST's Mobile Device Architecture

2. Firmware Layer:

Firmware is a secure set of code that communicates with the mobile device hardware. Firmware is typically written by the device manufacturer. It includes the secure boot code that initializes the firmware and the hardware components and boots the OS. This layer may also contain some of the ROTs supporting the hardware ROT's to provide the security capabilities.

3. OS Layer:

The OS Layer includes the OS kernel, PEE, and system service components along with their configuration data. The operating system is typically approved by the device manufacturer and controlled by the Device Owner. The components of this layer create a secure environment to establish a secure communication between the interfaces and device hardware and to implement security policies. The OS kernel offers application isolation, in collaboration with device integrity and protected storage incorporated in hardware and firmware of the device.

4. Application Layer:

Application Context is responsible for executing the applications. The OS kernel responsible for providing application isolation prevents applications from accessing other

applications container and modifying their user data, configuration data or code. This application isolation capability allows the devices to execute trusted and un-trusted applications simultaneously while at the same time providing the application isolation assurance to the users.

5. Information Layer:

Each Information Context is managed and controlled by its own information owner. Information owners configure their own security requirements and policies of accessing, processing and storing their data on the device. Information owners should not have access permission to modify or temper OS or applications components without the permission of device owner. But information owners should be assured of the secure functionality of the device and that the device platform's security capabilities always run in a trustworthy state while access to information owner's data is managed by their own configured policies.

2.3 TPM MOBILE Specifications

With the increased utilization of diverse connected devices mainly mobile phones and tablets, have fundamentally transformed our life styles where we now can access personal networks, bank accounts and business documents wherever and whenever required. To take full benefit of the richness and connectivity of these devices, there is a need to control the associated risks. This need activated to emerge two key platform security technologies; Global Platform's Trusted Execution; Environment (TEE) and Trusted Computing Group's (TCG) Mobile Trusted Module (MTM). These two technologies work together in a unified manner called TPM MOBILE to provide security and improved services to its consumers.

2.3.1 TPM Specifications:

TPM is an international standard for a secure crypto-processor, which is a dedicated hardware designed for secure processing in the devices. Since 2006 laptops and desktops are manufactured with the in build TPM chips. TCG in 2003 developed the first version of standard of TPM known as TPMv.1.2. Some of its salient features include strong cryptographic algorithms for hashing, authentication and authorization prevalent at

time of standardization such as SHA-1, RNG, RSA and HMAC. Moreover a single storage hierarchy model was presented to store data, keys and different mobile platforms. Later as the time passed, some of the specifications became obsolete and a need arose to modify the standard. Hence in 2012, TPM 2.0 was published which addressed many of the same use cases of TPM v1.2 and provided many same features with enhanced security capabilities to provide a high level of security assurance for desktop and laptops. TPM 2.0 is not backward compatible to TPM v1.2. TPM v2.0 provides stronger cryptographic algorithms than TPM v1.2 and also discarded the obsolete algorithms which were supported previously. Moreover it provides a three level hierarchy model for platform, storage and endorsement. TPM 2.0 in contrast to TPM v1.2 provides several documents as the reference implementation describing the behavior of codebase thereby ensuring a uniform behavior. The Table 2.1 shows the major differences in specifications of both the policies. Now TPM 2.0 is considered as the internal and accepted standard for hardware security in devices. [25]

Table 2.1: Major difference in Specifications of TPM v1.2 and TPM v2.0

Specification	**TPM v1.2/MTM**	**TPM v2.0**
Algorithms	DES, RSA, SHA-1	AES, RSA, P256, SHA-1, SHA-256
Cryptographic Primitive	RNG, SHA-1	RNG, SHA-1, SHA-256
Hierarchy	One (Storage)	Three (Storage, Platform, Endorsement)
Root Keys	One	Various keys and Algorithms per hierarchy
Authorization	HMAC, PCR, locality, physical presence	HMAC, Password, Policy
NV RAM	Only Unstructured data	Unstructured Data, Counter, Bitmap, Extend

2.3.2 Mobile Trusted Module (MTM):

MTM is security architecture with its origin lying in the TPM v1.2 and approved by TCG for use in mobile devices. It is anticipated to provide the same security and protocol interoperability as desktops and laptops, but with some enhancements for mobile devices... In 2008 TCG gave the specifications for MTM which were derived from TPM v1.2 with some changes for the mobile platform. The main changes introduced in the MTM that make it dissimilar from the TPM v 1.2 specifications are: [27,28]

1) The idea of secure boot is initiated. This means that the boot sequence is not only calculated, but also stopped when tempered software is detected. This enhances the integrity of mobile systems and is an important building block for security services or for those which focus on regulatory approvals.
2) The TCG mobile specification allows the MTM to be explicitly implemented not only in hardware but also in alternative implementations such as software or firmware. The MTM is considered as a functionality which makes it possible for device manufacturers to implement it as a privilege to their existing architectures.
3) It supports to run several parallel MTM instances of multiple stakeholders on the same device while still fulfilling the TCG specifications.

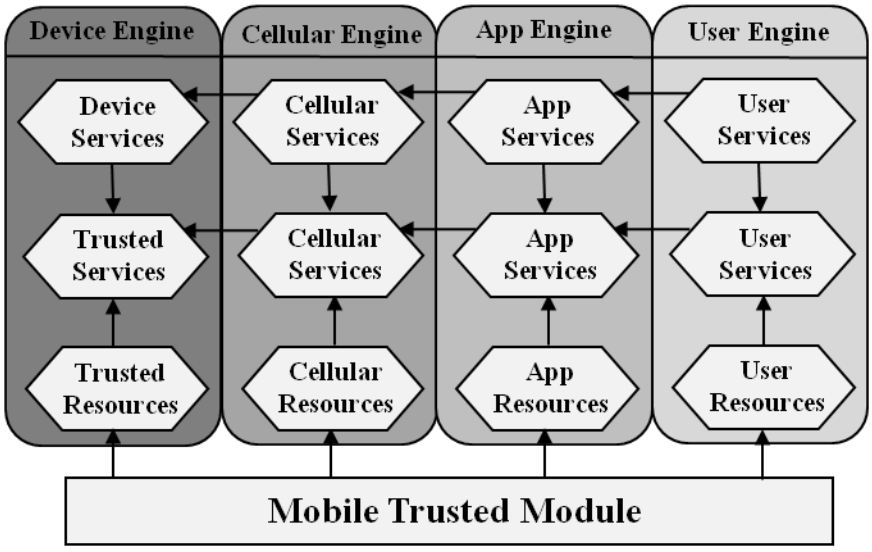

Figure2.2: Generic Architecture of MTM

The MTM specifications are dynamic and scalable allowing multiple MTMs called engines interlocked with each other and under the control of different stakeholders. Stakeholders include device manufacturers, mobile network operators, application providers and the users; as shown in figure 2.2. Ideally in a mobile platform a single MTM hardware should be accessed by different engines with each engine as a notion of its own trusted services. Each mobile platform engine should support:

1) Functionality to implement trusted and non-trusted services related to different stakeholders.
2) Self-test to find out the trustworthiness of its own state.
3) Secure storage of cryptographic keys; such as endorsement key, attestation identification keys and migration key.

2.3.3 The GlobalPlatform TEE:

The GlobalPlatform TEE categorizes a consistent isolation environment for SoC's for executing sensitive data, code and resources separately from the main OS environment. This isolation is possible due to the hardware architecture and the boot process utilizes hardware ROTs embedded in the SoC to make it robust against software and different probing attacks. Moreover prior to execution the applications running in the TEE are cryptographically verified, leading to high integrity assurance. Also it can be used as a distinct security coprocessor. It provides a trusted 'bridge' between the user and other security technologies such as Secure Element access control on one side and secured user interface on the other side.

The functionality of the Trusted Applications is used by the main OS and applications via a standardized 'Client API' which run normally in their own environment. Trusted Applications are implemented in the Internal API which provides assurance of secure access to resources, different cryptographic algorithms and storage regardless of the underlying SoC hardware.

2.4 TPM MOBILE Security Model

TPM Mobile security model unifies the hardware security architecture proposed by the MTM model and GlobalPlatform TEE. The security of the TPM MOBILE starts

with the boot process. The hardware ROT which mainly is an integrity key embedded on the processor starts its boot security. During the later stages of the boot, applications are verified cryptographically to make sure that authorized software is running on the device as shown in the figure 2.3

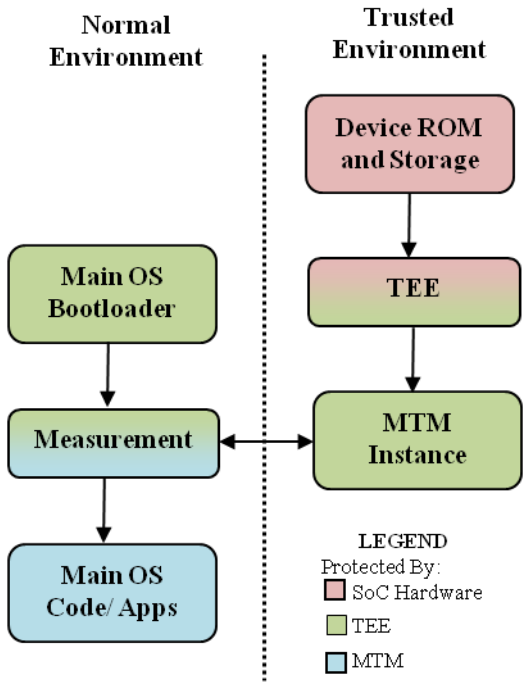

Figure2.3: TPM MOBILE Boot Process

After the secure boot the main OS can be accessed at any time which runs in a secure environment of the TEE, protected by the strong hardware mechanisms from the calling process. Hence the device security is ensured at all times during the boot chain process and the secure boot process of the TEE completes before handing over to the TPM MOBILE instance to provide protected boot services to the main OS. Therefore, the TEE provides the mandatory security bridge between the TCG-based main OS security model and the device's base security mechanisms simultaneously with minimal changes in the software design.

The security model as illustrated above works due to the chain of trust from one component to the next is ensured and cryptographically protected. While the specific implementation details differ, they must comply with the TEE specification and TPM-MOBILE deployed running on them, ensuring trustworthy protection and portability

across mobile devices. Figure 2.4 shows the protection hierarchy trust chain for the TPM MOBILE data and operation.

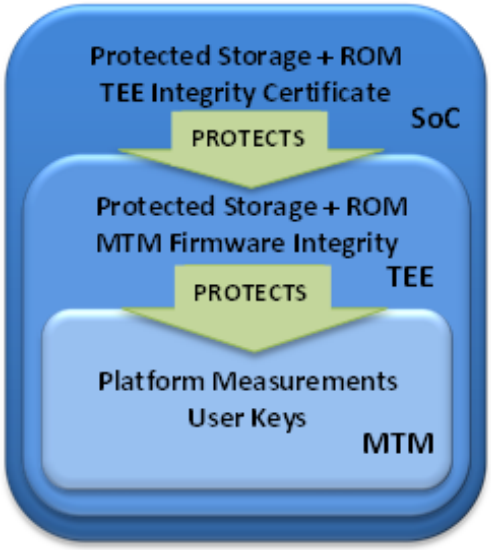

Figure2.4: Mobile Device Protection Hierarchy

Hence the three hardware security capabilities illustrated by NIST are achieved as follows:

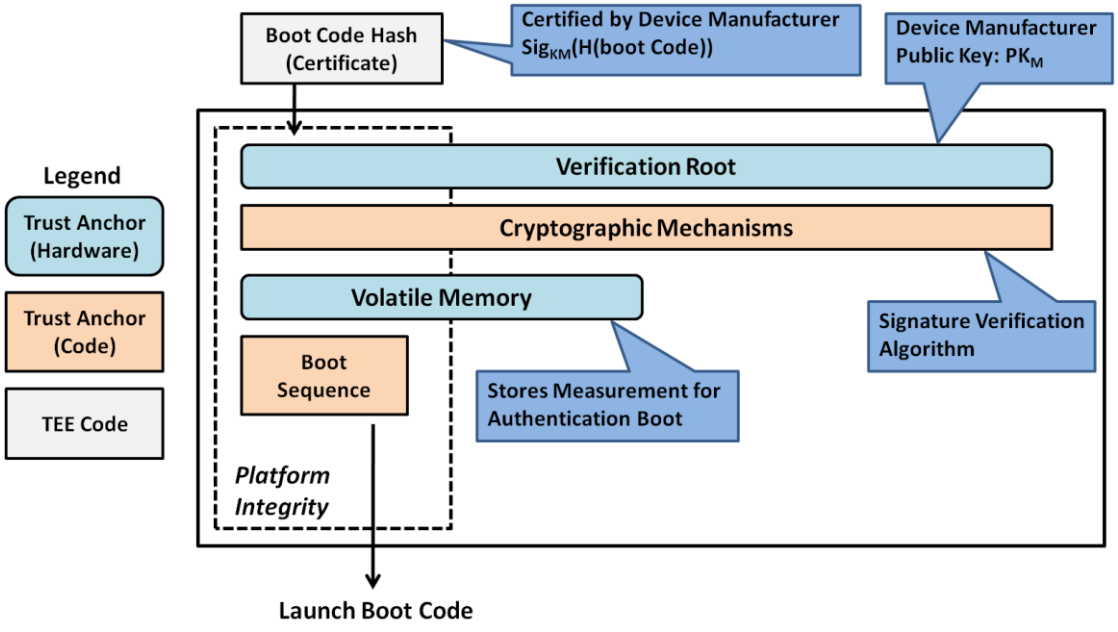

Figure2.5: Platform Integrity in TPM Mobile Security Model

1) Platform Integrity:

This will be achieved through the mechanism of secure boot. The device will run a secure boot code present in the hardware using root and device manufacturer keys and will calculate the boot hash using stored and trusted cryptographic algorithms. This calculated has will be checked with the hash certificated stored by the device and verified. If verification passes the system will boot otherwise in case of failure the system will stopped. Figure 2.5 shows the phenomenon.

2) Secure Storage:

This is achieved by having the trusted hardware its dedicated volatile and non-volatile memory which can be isolated from the insecure memory portion. Moreover no confidential data is sent in plain text to the insecure memory and is encrypted using the device key and cipher text is stored on the insecure memory. The figure 2.6 shows the phenomenon.

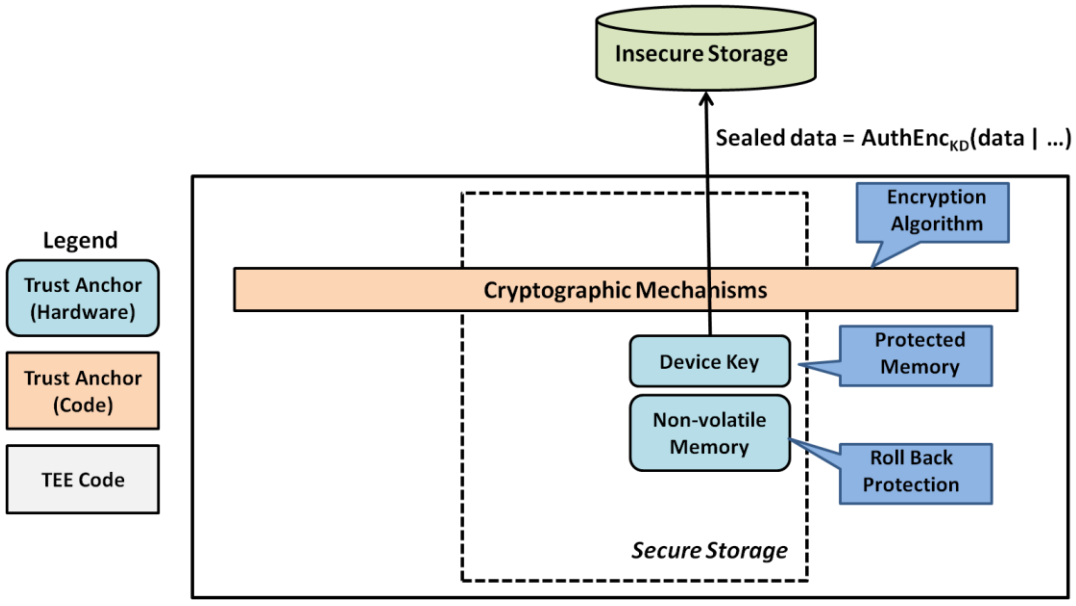

Figure2.6: Secure Storage in TPM Mobile Security Model

3) Isolated Execution:

Trusted application run within the trusted hardware location. All the client applications before launch are authenticated using the application's public key. Afterwards an isolated and dedicated memory is allocated to the application to execute and to prevent data exchange among different applications. Moreover whenever secure

services are requested by the applications they are executed in the secure hardware location. The figure 2.7 illustrates the isolated execution components involved.

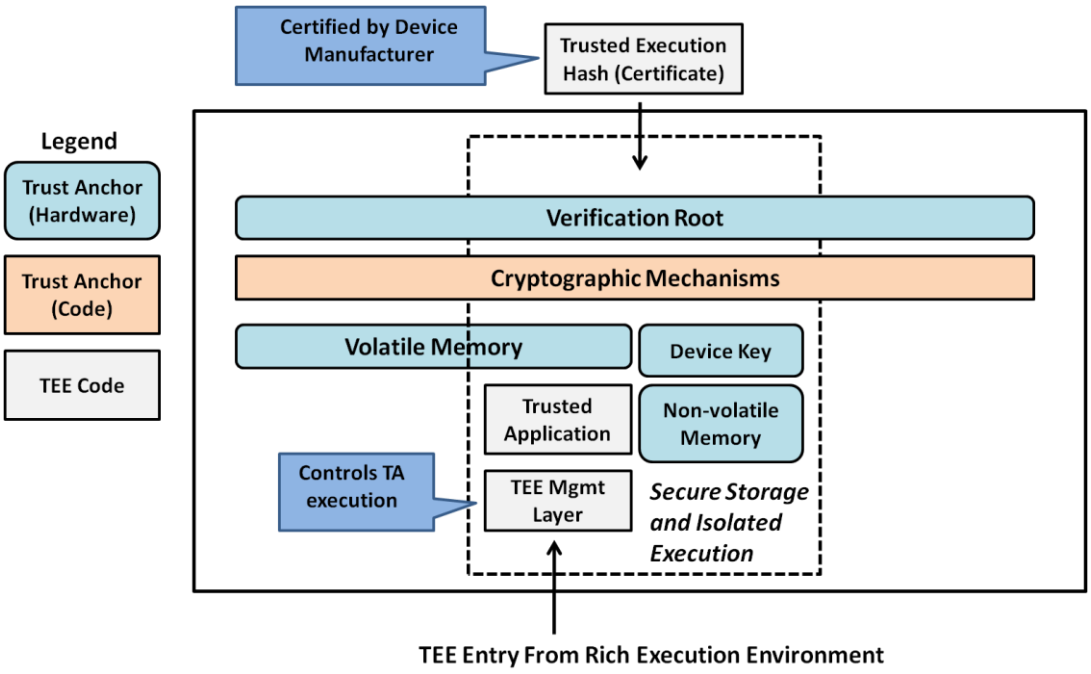

Figure 2.7: Isolation Execution in TPM Mobile Security Model

The figure 2.8 represents the components involved in the five hardware roots of trust specified by the NIST in order to develop a secure hardware system while implementing the security capabilities;

2.5 Analysis of Standards

In the above section we have analysis two sets of standards; one is the NIST hardware rooted standard and other is the TCG's TPM Mobile standard.

2.5.1 Comparative Analysis of Standards:

The major difference between the above mentioned two sets of standards is that NIST provides the key components and capabilities required for the manufactures to develop a secure mobile device whereas TPM Mobile also describes how these key components can be used to develop a secure device. Moreover it also gives the TEE architecture to use these hardware rooted security components on the upper layer in a trustworthy manner. NIST has introduced a new hardware root of trust named ROT for

integrity for integrity checking of the measurements and processes taking place in the device.

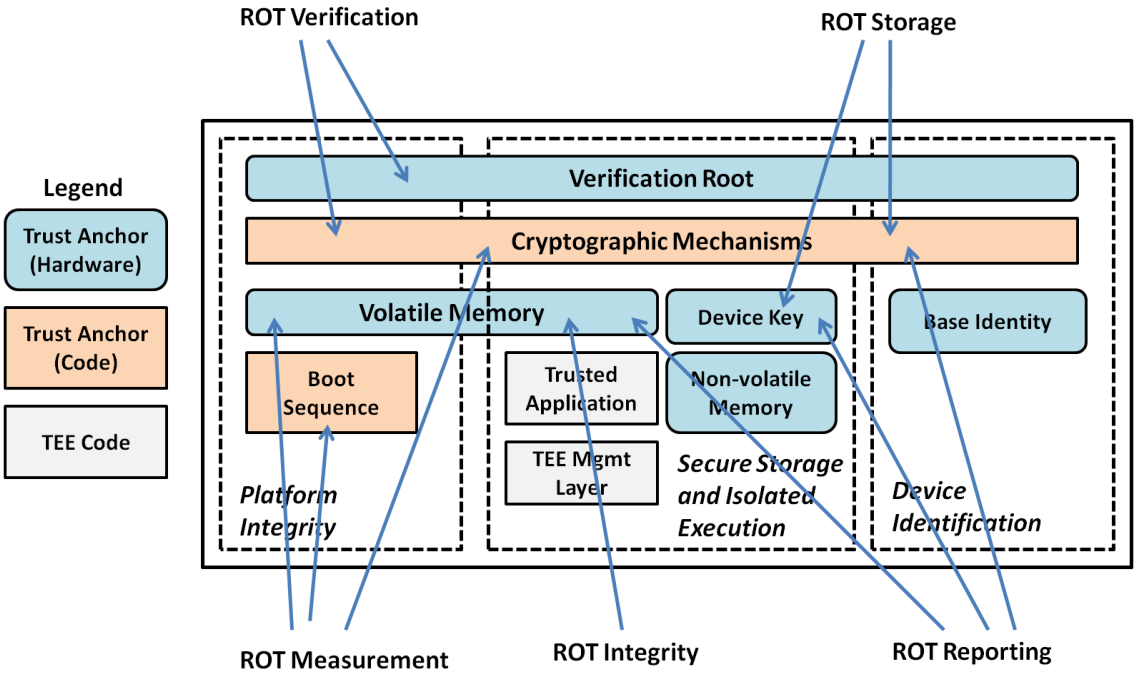

Figure2.8: Mapping of ROT's For Security Capabilities

While analyzing the standards provided by TCG, TPM 2.0 is the internationally accepted and implemented security standard. TPM 2.0 is the enhanced security standard of TPM v1.2 but it is not backward compatible to it. The major differences in both the standards are illustrated in table 2.1. Furthermore TCG developed the standard for mobile devices called MTM which was derived from TPM v1.2. It used the specifications provided in TPM version with some changes for the mobile environment which included the multi-stack holder environment and allowing to run multiple MTM instances on the same device. The concept of secure boot was introduced. Moreover the compulsion to implement TPM security components in hardware was removed. Developers have been provided flexibility to implement these security features in hardware, software or firmware upon their own requirements and design. The question of how to provide theses security facilities to the OS and application layer was resolved by Global Platform TEE which provided the specifications and secure boot chain mechanism of TEE to develop a

secure trusted environment for the mobile devices. Together the MTM and TEE provide a complete security standard TPM MOBILE for the mobile devices.

2.5.2 Analysis of MTM Specifications:

In computing systems mutual trust among peer systems is established through attestation process for integrity assurance. Secure attestation is ensured by cryptographically protected hardware that is resistant to software attacks. TCG published the improvement in specifications for trusted computing on mobile devices and TPM is considered to be the root of trust and enables secure attestation by providing secure cryptographic primitives for signatures and hashes. However, the similarities of MTM and TPM have raised many implementation concerns and respective challenges. The following are some of the salient observations.

1) MTM provides relatively weak security policies of the time as it is derived from TPM v1.2. TPM v2.0 promises enhanced security policies and have proven to be a better standard in high computing devices such as desktops and laptops. For example, MTM specifies DES and SHA1 as encryption and hashing algorithms whereas they are now obsolete and better security algorithms such as AES, SHA256, etc are present and added as a standard in the later version TPM 2.0. Moreover, MTM mandates single storage hierarchy model which is unsuitable for the mobile environment having multiple stalk holder hierarchy. Hence a modified version of MTM should be presented which must be comparable to TPM v2.0 providing enhanced and up to date features and specifications.

2) TCG enlightens the functionality aspects of MTM while not focusing on the implementation technique required in developing such modules. This aspect has been left over for the manufactures to define their implementations strategies by themselves which does not makes one manufacture’s model compatible with the other one. Moreover these solutions are closed form solutions. Hence there is a need to modify the standard to incorporate the reference implementation techniques and to bring all the manufacturers on a unified platform.

3) TCG specifies a separate deployment of TPM functionality in an isolated module which may be unable to yield the desired trade-off between cost, security and

performance. Mobile devices are now providing more computing resources and performance but have a serious constraint of device size and power management. High security requires high computing resources and physical area utilizing more power resources and hence making the device more costly. On the other hand software implementation of MTM will not be able to meet the security challenges. Therefore there is a need to standardize a suitable implementation technique for the mobile device environment and to yield the desired trade-off between cost, power, security and performance.

4) The algorithms defined for security in MTM support cryptographic algorithms which require large computation power and resources, and hence are less suitable choices for low computing resource constraint processors. e.g. SHA-1 as a hashing algorithm and RSA as a public key algorithm require more computing resources and high power consumption. Suitable algorithms with less computing resources should be suggested for the mobile computing environments.

5) The implementation technique of the cryptographic algorithms does not specify cryptographic mode of operation. A specific cryptographic mode of operation that is resilient to channel errors should be suggested in the policy for the implementation purposes.

6) The last concern is related to the robust implementation of cryptographic primitives. Typically, cryptographic co-processors occupy large silicon area and have poor flexibility. On the contrary, a co-design approach of hardware and software allows algorithm flexibility to be achieved at relatively less hardware cost and smaller surface area.

The modified standard should mitigate the limitations of TCG specifications discussed above while providing a new concept for the implementation of the MTM security services. Suggestions for the modified version will be discussed in the last chapter where the solution implementation technique for the mobile devices will be proposed.

2.6 Selection of Cryptographic Algorithms

The selection of cryptographic algorithms for the security of embedded systems is a critical and vital element in strengthening their secure architecture. Both TPM v2.0 and MTM have provided conventional cryptographic algorithms for the purpose of encryption decryption, hashing, digital signatures etc. For example TPM v2.0 proposes AES for symmetric ciphering and deciphering, RSA for asymmetric ciphering and deciphering and SHA-256 for hashing functionality. The proposed cryptographic algorithms are popular for their cryptographic strength and also standardized by NIST and NSA as one of the best secure crypto algorithms. But these algorithms are suitable for devices embedded with high power processors meant for excessive computing such as laptops, desktops, tablets and smart phones. On the contrary these algorithms fail to meet the power, processing and memory constraint environment of various wearable and IoTs which include Bluetooth, NFC, RFID and smart card systems. Therefore a new branch of cryptography was introduced named as lightweight cryptography. Lightweight ciphers were developed for such resource constraint devices providing the same comparable security as conventional crypto algorithms but utilize less power and memory due to smaller key size, smaller block size, less number of rounds and relatively simpler design architecture.

2.6.1 Lightweight Block Ciphers:

Some of the lightweight block algorithms implemented widely and known for their high strength and throughput are listed below in table 2.2 along with their features. [11,13,14]

The cipher with the maximum throughput and minimum memory usage is considered to be the better cipher. From the above listed ciphers the best cipher having the maximum through put is Simon/Speck with 855 Kb/s of throughput. After that PRINCE and mCrypton seem to provide a better through put of 533 Kb/s and 482 Kb/s respectively. Figure 2.9 demonstrates their relative through put graph. Figure 2.10 and 2.11 illustrates the graphs of memory usage on RAM and RAM. It depends on the design criterion of the developer that whether it uses more ROM space for the algorithm code and states storage or provides an appreciable amount of RAM for processing. From the

algorithms listed DESLX uses the maximum of ROM space but Present utilizes maximum of RAM Space. Observing both the metrics simultaneously, Speck/Simon uses minimum of Ram space and no RAM is util36ed during processing. It stores its intermediate states and processing data in registers. Hence Speck/Simon provides an integrated solution if maximum throughput and minimum storage utilization.

Table2.2: List of Lightweight Block Cryptographic Algorithms and their Performance Metrics

S.No.	Lightweight Algorithm	Key Size	Block Size	No. of Rounds	Throughput (Kb/s @ 100kHz)	Power Consumed (µW)/bit	Memory Utilized (bytes)	
							RAM	ROM
1	DESLX	184	64	16	44.4	1.6	112	16816
2	HIGHT	128	64	32	188.2	-	18	3130
3	mCrypton	64 96 128	64	12	482.3	-	18 20 24	2726 2834 3108
4	Piccolo	80 128	64	25 31	237.04 193.9	4.42 2.78	79 91	2434 2510
5	Present	80 128	64	31	 200	2.78 3.67	142 142	4814 4964
6	PRINCE	128	64	12	533.3	5.8		
7	SEA	96	96	93	103	3.218	24	2804
8	SIMECK	64 96 128	32 48 64	32 36 44	88.9 120 133.3	0.606 0.875 1.162	-	-
9	SPECK/ SIMON	64 72/96 96/128 128/192/256	32 48 64 128	32 36 42/44 68/69/72	 855	3.98 3.32 3.65 4.20	0 0 0 0	324 556 602 1108
10	XTEA	128	64	64	57.1	19.5	11	1394

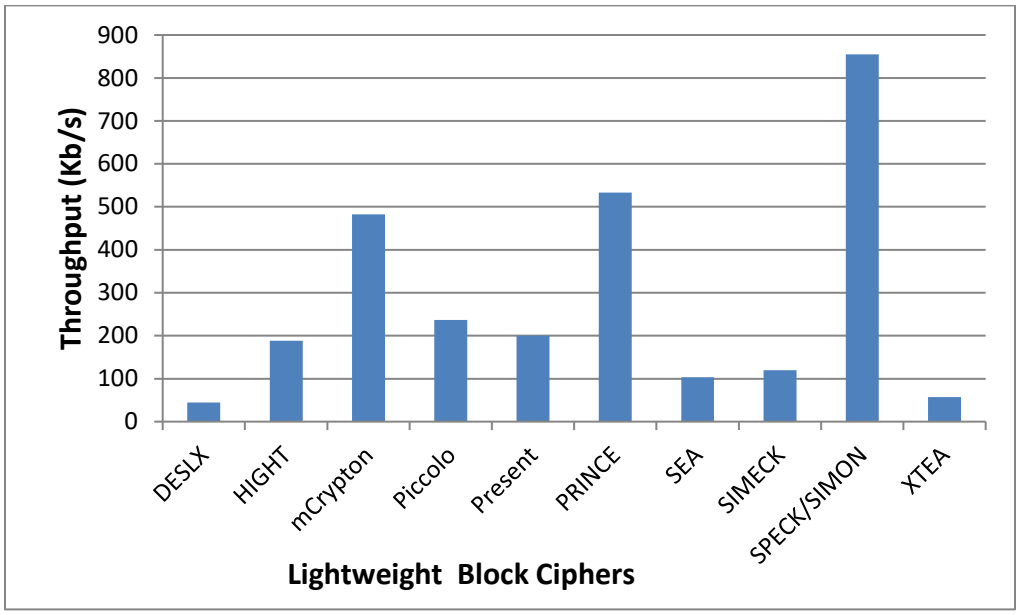

Figure 2.9: Comparative Analysis of Throughput of Popular Lightweight Block Cipher

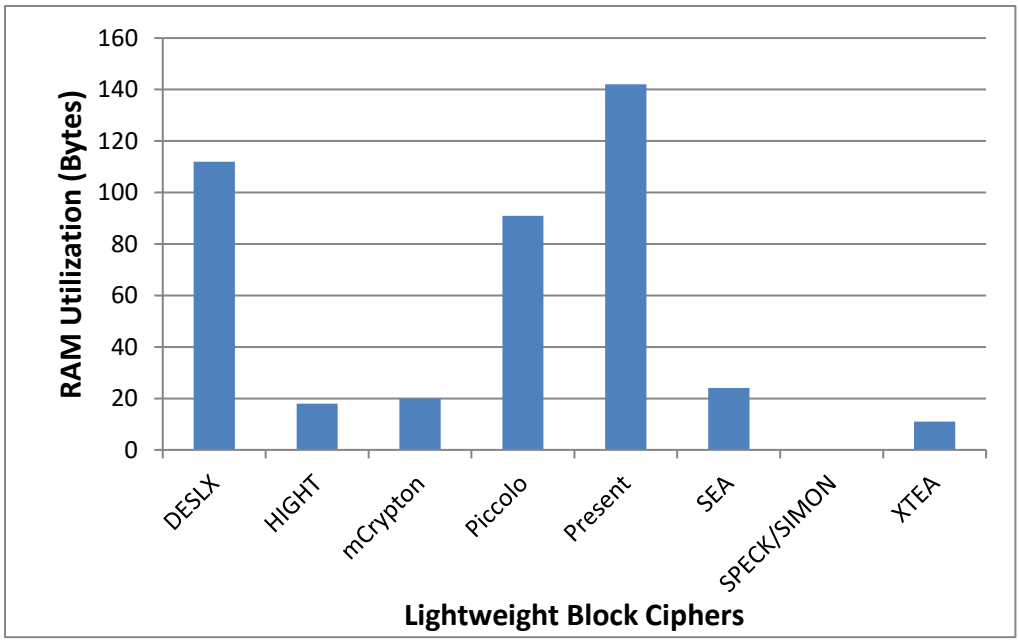

Figure 2.10: Comparative Analysis of RAM Utilization of Popular Lightweight Block Cipher

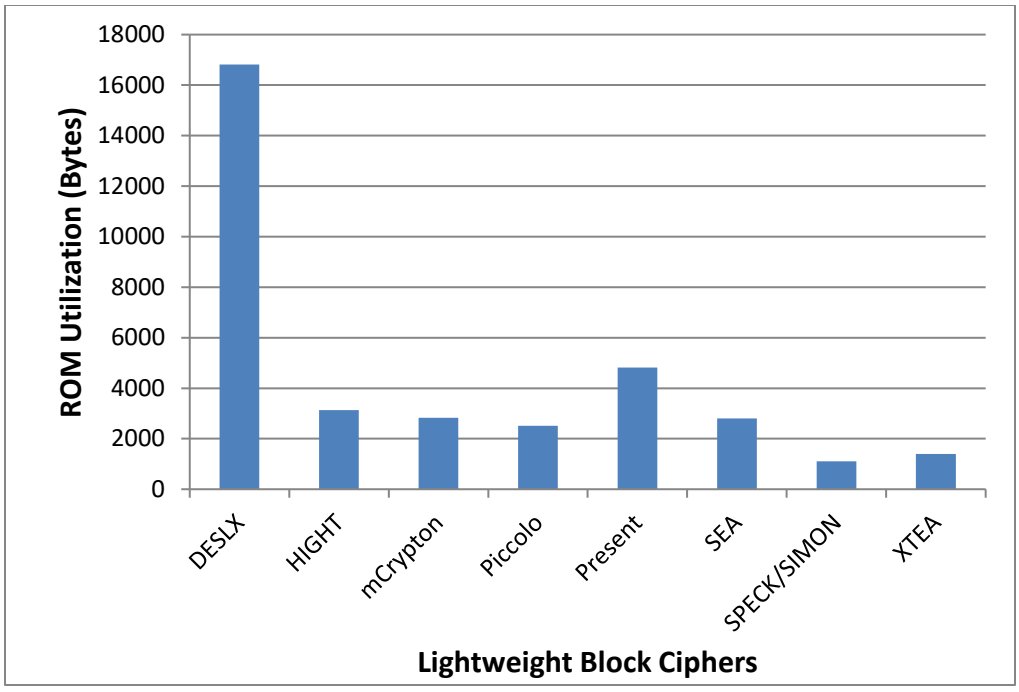

Figure 2.11: Comparative Analysis of ROM Utilization of Popular Lightweight Block Cipher

Out of the above listed light weight ciphers NIST has recommended DESL, SEA, TEA, SIMON and SPECK. Piccolo algorithm provides the best results of throughput and relative hardware size and is prioritized when implementing the algorithms in hardware. Simon and Speck are the algorithms made by NSA but have not publically released them. SPECK targeted for hardware and SIMON targeted for software implementation have proven to be among the best algorithms for resource constraint devices.

2.6.2 Lightweight Hash Functions:

Table 2.3 shows the list of lightweight hash functions implemented widely along with their performance metrics [15,16]. QUARK, PHOTON, DM-PRESENT and SPONGENT are the hashing functions standardized by NIST. As shown from figure 2.12 among the listed lightweight hashing functions QUARK seems to fulfill the tradeoff of high throughput, less power consumption and minimum memory usage. Whereas PHOTON and SPONGENT provide a wide range of digest size options for implementation. The analysis carried out related to the lightweight algorithms will be used in the last chapter were the suggested solution will be presented.

Table2.3: List of Lightweight Hashing Functions and their Performance Metric

S.No.	Lightweight Algorithm	Digest Size	Rate	Internal State Size	Throughput (Kb/s @ 100kHz)	Power Consumed (µW)/bit	Memory Utilized (bytes)	
							RAM	ROM
1	ARMADILLO	80	48	256	109	44	112	16816
		128	64	384	1000	-	-	-
		160	80	480	100	-	-	-
		192	96	576	100	-	-	-
		256	128	768	100	-	-	-
2	DM-PRESENT	64	80	64	242.42	6.28	18	3130
		64	128	64	387.88	7.49		
3	Lesamnta-LW	256	128	256	125.55	-	-	-
4	PHOTON	80	16	100	2.82	1.59		
		128	16	144	1.61	2.29		
		160	36	196	2.70	2.74	60	598
		224	32	256	1.86	4.01		
		256	32	288	3.21	4.55	96	364
5	QUARK	136	8	136	1.47	2.44		
		176	16	176	2.27	3.10	42	974
		256	32	256	3.13	4.35	60	1106
6	GLUON	128	8	136	12.12	-	-	-
		160	16	176	32	-	-	-
		224	32	256	58.18	-	-	-
7	SPONGENT	80	8	88	0.81	1.57	-	-
		128	8	136	0.34	2.20	-	-
		160	16	176	0.40	2.85	66	598
		224	16	240	0.22	3.74	-	-
		256	16	272	0.17	4.21	101	364

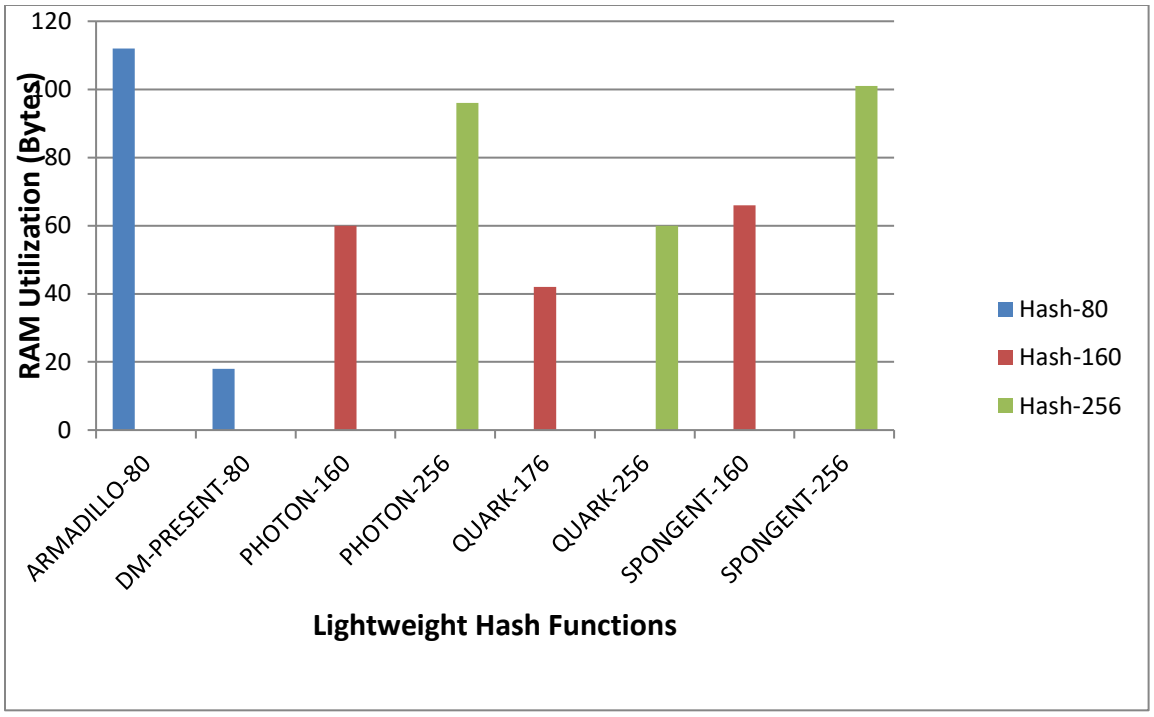

Figure 2.12: Comparative Analysis of RAM Utilization of lightweight Hash Functions

2.7 Conclusion

In this chapter we have highlighted various hardware rooted mobile security standards available for the mobile manufacturers to develop secure mobile systems. The relative comparison has also been carried out and it can be concluded that NIST lists down the components and capabilities required in developing a secure mobile system whereas TPM Mobile also provides the TEE architecture to use these hardware rooted components efficiently on the upper layers of the mobile device architecture. Analysis of all these standards and their comparison was carried out. Moreover some of the shortcomings analyzed in MTM standard were also discussed. At the end of the chapter concept of lightweight cryptography was introduced and different lightweight block ciphers and hashing algorithms are listed. There relative comparison with respect to throughput, power consumption and memory (RAM and ROM) was carried out. The next chapter will focus on the TPM implementation techniques and the commercial security solutions available in the market.

Chapter 3

MOBILE SECURITY IMPLEMENTATION SOLUTIONS AND THEIR ANALYSIS

3.1 Introduction

In the last chapter we highlighted the hardware rooted security standards present for the mobile devices. Also we analyzed these standards and after their comparison listed some of the shortcomings present in the MTM specification. In this chapter we would discuss the key components of a TPM and its implementation techniques for mobile devices. Afterwards we will discuss the different security solutions developed by the industry and will analyze them for the extent to which they comply with the standards. As most of the solutions are dependent on the underlying ARM TrustZone technology, hence an in depth analysis will be carried out of ARM TrustZone. Moreover its shortcomings will also be highlighted.

3.2 Internal Components of a TPM

TPM is the basic component in the trusted computing devices which offers a hardware root of trust to ensure OS and applications integrity. The TPM is basically a hardware chip embedded with the basic necessary security features like generation of random numbers, cryptographic operations execution, secure storage of vital data and secret keys; as shown in the figure 3.1. TCG gave the specifications compulsory for the TPM in its version 1.2 as the first standard in 2003 followed by its enhanced security version 2.0 in 2008 which is now internationally accepted and implemented in static computing devices. TPM comprises of the following trusted components embedded into a single SoC. [23,24]

3.2.1 Secured Input and Output:

The data transfer on the communication bus takes place after encoding or decoding according to the protocol specified by the programmer. TCG has not specified

any design or structure of this input output port in any of its specifications i.e. neither in TPM v1.2 nor in TPM v2.0 and have left this to the design of the platform manufacturers.

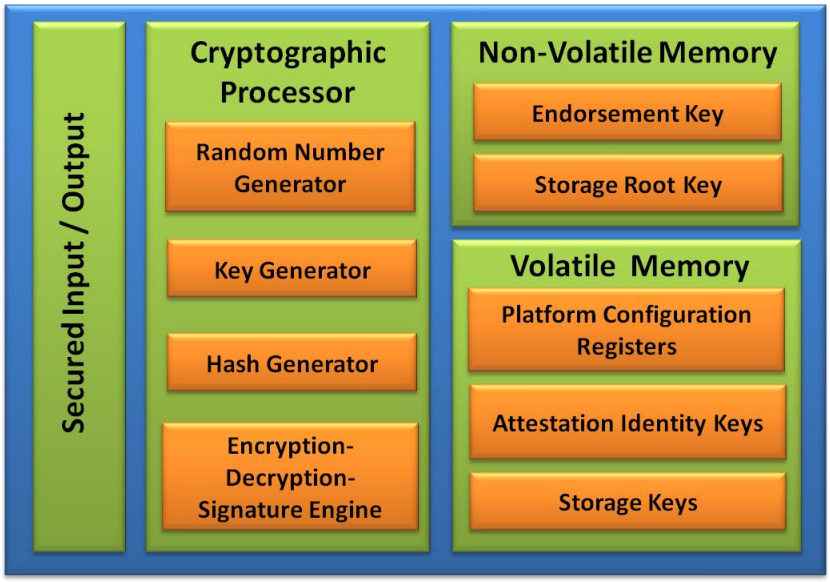

Figure 3.1: Internal Component of a TPM Chip

3.2.2 Cryptographic Processor:

The cryptographic processor should be a dedicated hardware meant to perform cryptographic operations. It performs the following cryptographic operations:

- **Encryption/Decryption and Signature Generator:**

  The engine performs symmetric encryption and decryption and also generates signatures using it. In TPM v 2.0 AES and algorithms equivalent with it are specified as a standard.

- **Random Number Generator:**

  TPM uses RNG for random-nonce and key generation. It also generates random sequences required for the digital signatures. TCG as not imposed any restriction regarding the implementation of the RNG and have left the design decision on the programmers.

- **Key Generator:**

  It provides the functionality to generate symmetric and asymmetric keys using RNG capable of generating random sequences. The key generation mechanism has been standardized and the process must comply with its preliminary tests. But TCG has not specified any compulsion on the key generation process.

- **Hash Generator:**

  The Hash Generator uses HMAC implementation for authentication and authorization purposes. TCG has specified SHA-1 and SHA 256 as the hashing algorithms to support integrity measurement generation.

- **Non-volatile and Volatile Memory**

  Volatile memory is used to store temporary data items including the temporary state keys and data generated during the signing or decryption operations. The non-volatile memory is used to store persistent data items that relate to a TPM's identity (including permanent keys) and associated state data.

3.3 TPM Implementation Techniques

TPM functionality can be implemented in three different ways in the embedded system; as illustrated in the figure 3.2. Each of these three implementation methods has their own pros and cons with respect to areas of interest such as cost, security and flexibility. [28,31]

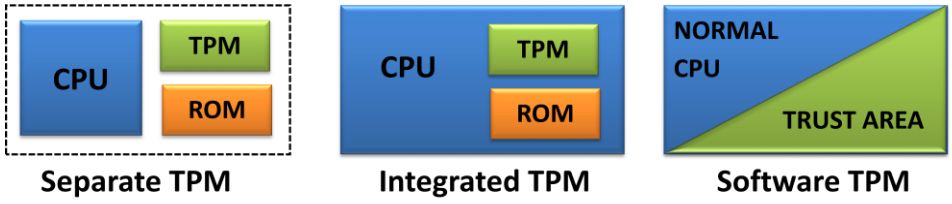

Figure 3.2: Options for Implementing TPM Functionality

3.3.1 Separately Mounted TPM:

The first method is to mount a discrete TPM chip on the mother board interconnected with the processor via a bus, used for data communication. This approach is a concrete example of complaint TPM specifications and is widely deployed in today's static systems for trusted computing in desktop computers. Examples of separately mounted TPM include IBM's Secure-Blue technology or Texas Instruments' M-Shield security technology. A discrete TPM chip soldered on the motherboard increases manufacturing cost, size, and weight in embedded systems. This is the major problem for devices with low power constraint resources like mobile phones. Also interfacing a TPM chip at board-level increases the security threats especially when a device is operating in a hostile environment. Therefore the cost, size, and security constraints of embedded devices require different integration technique of hardware-supported security functionality.

3.3.2 Software TPM:

The second method is a software-TPM which executes as an isolated secure environment of a general-purpose processor. In software-TPM, malicious and un-trusted applications run on the same processor where the TPM operations are executed. Hence no discrete boundary is present between the TPM functionality and the rest of the components. Hence secure implementation of shielded locations cannot be realized in a software-TPM. Moreover the software providing the TPM functionality cannot protect itself against tampering and other malicious activities.

3.3.3 Integrated TPM:

The third method is an on-chip deployment of the TPM module so to make a single SoC acting as secure processor. Hence a single chip provides the functionality of both TPM as well as general purpose computing concurrently as trusted computing is embedded with processor core and memory. An alternate idea is to integrate security features directly into the processor core through micro-architectural enhancements. Hence this provides the advantages of the software TPM implementation discussed above, most particularly reduced cost and size. Moreover it provides a better protection

against tampering and other physical probing attacks. Potentially malicious applications cannot access critical data including secret keys which are stored inside the TPM.

3.4 Mobile Market Share Analysis

With the increased utilization of smart connected devices principally smart phones and tablets, have fundamentally transformed our life styles where we now can access personal networks, bank accounts and business documents wherever and whenever required. According to statista nearly 85% of the market is captured by the android smart phones as shown in the figure 3.3.

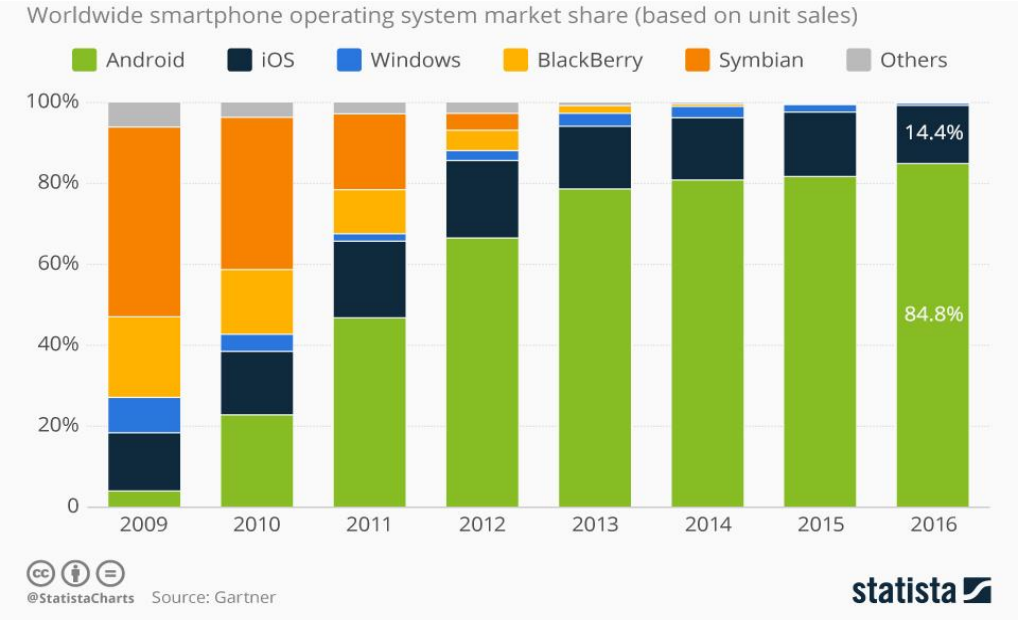

Figure 3.3: Yearly Smartphone OS Market Share

The chipsets distribution being used in these 85% of android phones is shown in figure 3.4 and figure 3.5. From the graphs it can concluded that Qualcomm, Samsung and MediaTek capture almost 90% of the market.

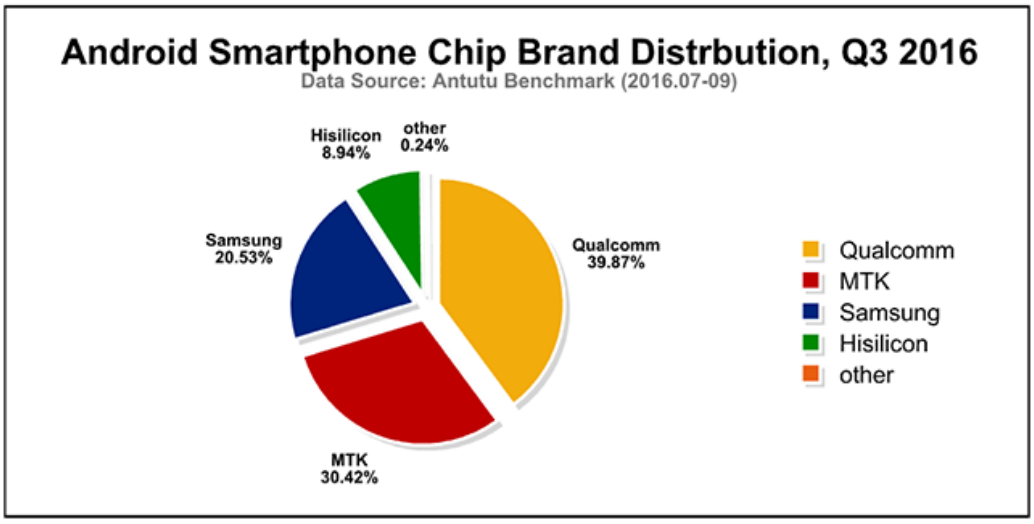

Figure 3.4: Smartphone Chipset Market Share

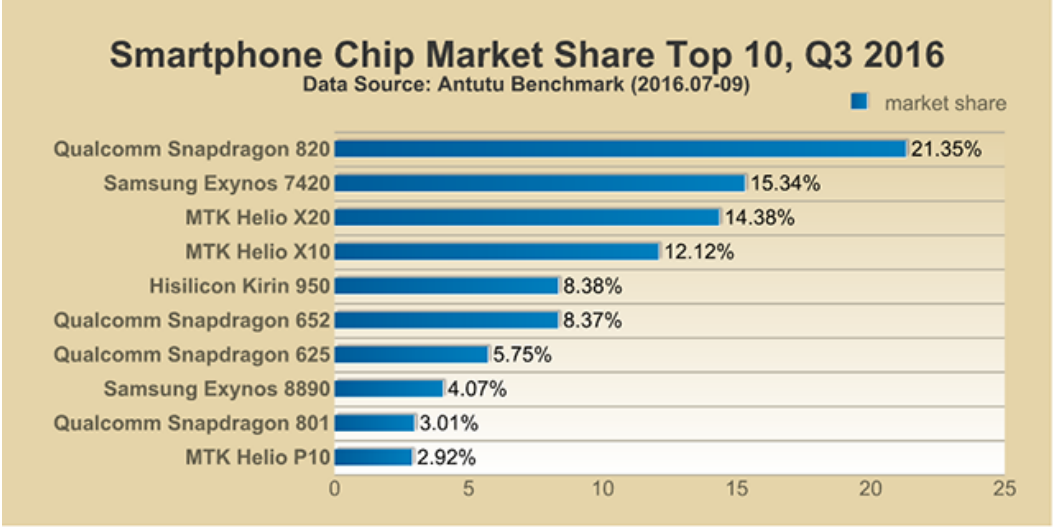

Figure 3.5: Android Smartphone Chipset Market Share

3.5 Contemporary Implementation Solutions

Some of the leading solutions provided by different manufacturers in the area of hardening mobile devices are highlighted in this section. We will specially be interested in analyzing the security solutions provided by Qualcomm, Samsung and MediaTek as they capture most of the market. But other solutions discussed are also equally important and with hold a significance standing in the industry.

3.5.1 ARM TrustZone:

ARM launched Trust-Zone in 2003. ARM Trust-Zone is a SoC-based approach that offers the security for a TEE running beside the main OS. Applications referred to as Trusted Applications run on the Trust-Zone-protected TEE. Trust-Zone technology is incorporated tightly the ARM processors. The security is embedded in the mobile processor SoC and Trust-Zone system intellectual property blocks and accessed through and AMBI bus thus enabling the security of the system peripherals such as keyboard, cryptographic blocks, secure memory, and screen from diverse software attack. TrustZone uses system-wide hardware security virtualization technique to create an isolated environment for trusted applications. [19]

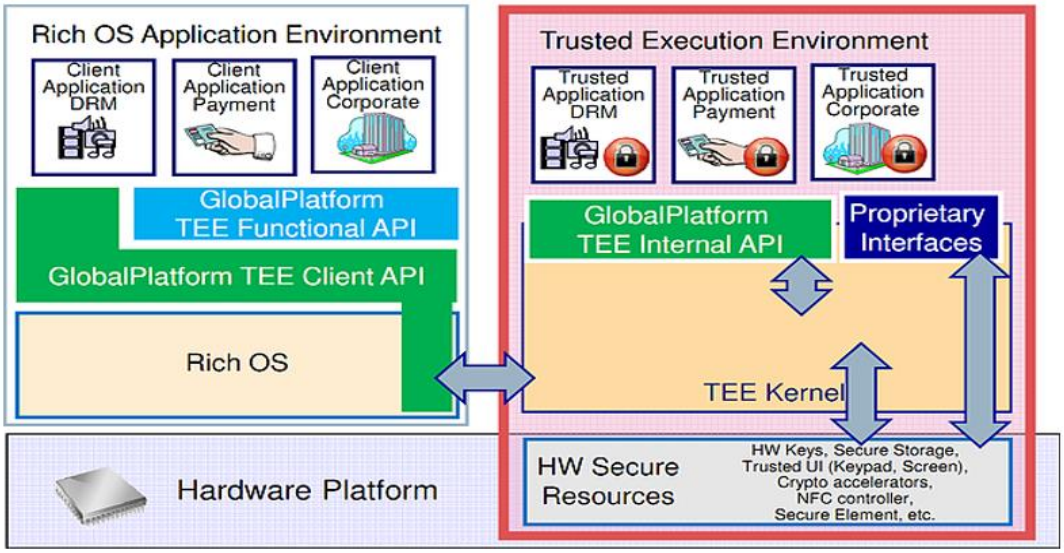

Figure 3.6: ARM Trust-Zone Environment

Smartphone vendors utilize TrustZone technology with their own closed form solutions to provide security to their customers. The manner in which Trust-Zone like architecture is implemented is decided by SoC manufacturers such as Qualcomm, Samsung and MediaTek. As a result, Trust-Zone implementations may vary from vendor to vendor

3.5.2 Qualcomm:

Qualcomm is a licensee of ARM technology and its solution referred to as Snapdragon Security Solutions; offers the security enabled in its Snapdragon family of processors. Its security solution is based on three pillars of security: [33]

1. **Secure MSM:**

   All Snapdragon processors are enabled with the feature of Secure MSM. Secure MSM is composed of three components which are congruent with the features enabled by ARM's TrustZone. The first component is Secure Boot. Secure Boot code is embedded in the SoC's ROM and is the system's ROT responsible for authenticating the code executed on the device. The second component is its own TEE. TEE is the internal OS code independent of the device OS. It utilizes ARM's TrustZone technology for trusted execution of code and to prevent, isolate, or monitor possible damage different malwares. The third component is a hardware-based cryptographic accelerator. The high-speed cryptographic accelerators are able to handle multiple data channels, while preserving context separations.

2. **Studio-Access Technology:**

   Studio-Access Technology provides security on the digital rights management (DRM) controls of the system. The assurance of the trusted execution of DRM permits the premium content owners to expand content distribution and at the same time preserve the content rights. Its TEE is named as Content Protection Zone which is responsible to isolate multiple video streams and to store DRM keys in secure storage.

3. **Enterprise and BYOD Security:**

   Snapdragon-based solutions provide the APIs that mobile device management (MDM) vendors can use for the BYOD security solutions as Qualcomm itself does not provide an end-to-end enterprise solution.

3.5.3 Samsung:

Samsung offers its security solution by the name of KNOX. Samsung KNOX was launched in 2013 and continues to develop to protect enterprises and their data, while providing employees with the productivity needs of BYOD. KNOX is a suite of products and claims to provide device & data security, easy enrollment, container usability and cloud-based mobile device control. Samsung is also an ARM licensee, and uses TrustZone technology to support embedded security. Its trusted environment called ARM TrustZone-based Integrity Measurement Architecture continuously monitors the integrity of the Linux kernel. Samsung KNOX creates a separate secure area on the device for enterprise and corporate applications and data. These applications and data are isolated from applications outside the container. Thus KNOX provides a complete MDM cloud based solution for the BYOD scenarios. [35,36]

3.5.4 MediaTek:

MediaTek is a Taiwanian company and have been developing mobile SoCs since 2009. It has not created its own Smartphone and only markets the chipsets to other Smartphone vendors for use. It has raised prominence over the last few years due to its lower cost products and multi-core CPU design. MediaTek uses the same ARM CPU core designs as Qualcomm and Samsung but MediaTek was one of the first to adopt ARM's big.LITTLE architecture and developed the first heterogeneous multiprocessing technology (HMT) SoC MT813 named CorePilot. MediaTek marketed the first true octa-core mobile CPU with this HMT technology to achieve flexibility in task allocation to the individual cores along with peak performance and to achieve power efficiency in these power constraint devices.

It also uses ARM Trust-Zone features to provide security to its users with its own modified closed-source TEE to communicate with the hardware. Among the developers community, MediaTek doesn't have a good standing as many consumers remain skeptical about the company's hardware due to its refusal to share source-code. The lack of source-code prevents third party patches for any security or hardware issues left unfixed by the company [38]

3.5.5 Intel:

Similar to ARM, Intel has taken the approach of embedding security features into SoC. While ARM includes Trust-Zone as part of its IP cores that integrated into a single chip, Intel instead embeds security in its processors as a separate IP block known as Intel trusted execution environment. This IP block offers a separate environment for security mechanisms with its own microcontroller core, memory and OS. Hence Intel offers a consistent security across all its processors through this embedded IP block. But this too is specific to only Intel processors which comprise less than 8% of the mobile market share. [33]

3.5.6 Boeing:

Boeing had recently launched its Black Smartphone designed to handle the ultra-secure needs of the U.S. defense and security communities**.** Boeing security system is referred as PureSecure architecture. Similar ARM's TrustZone, PureSecure is built upon layers of trust from embedded hardware and OS policy controls. Also physical security of the phone is important and therefore the phone is sealed; and any tampering or attempted disassembly of the phone would destroy the device and trigger the wipe functions to clear the device contents. Boeing Black is compatible with leading MDM systems and solutions.[33]

3.5.7 Apple:

Apple is considered to be one of the most secure solutions of the industry. It withholds a unique position in the industry when it comes to hardware/software integration as Apple designs its own chips and its own operating system. Apple has designed security into its products from the silicon up. iOS devices with an A7 or later processor (so the iPhone 5S and newer), also have a Secure Enclave processor (SEP) which is also has its own secure boot process. The Secure Enclave is interesting because like the secure boot process, it is separate from iOS. This is done because this separation makes it harder to attack. [42]

System security is designed so that both software and hardware are secure across all core components of every iOS device. This includes the boot-up process, software

updates, and Secure Enclave. This architecture is central to security in iOS, and never gets in the way of device usability. The tight integration of hardware and software on iOS devices ensures that each component of the system is trusted, and validates the system as a whole. From initial boot-up to iOS software updates to third-party apps, each step is analyzed and vetted to help ensure that the hardware and software are performing optimally together and using resources properly.

1. **Secure boot chain:**

   Each step of the startup process contains components that are cryptographically signed by Apple to ensure integrity and that proceed only after verifying the chain of trust. This includes the boot loaders, kernel, kernel extensions, and baseband firmware. This secure boot chain helps ensure that the lowest levels of software aren't tampered with.

2. **System Software Authorization:**

   Apple regularly releases software updates to address emerging security concerns and also provide new features; these updates are provided for all supported devices simultaneously. Users receive iOS update notifications on the device and through iTunes, and updates are delivered wirelessly, encouraging rapid adoption of the latest security fixes.

3. **Secure Enclave:**

   The Secure Enclave is a coprocessor fabricated in the Apple S2, Apple A7, and later A-series processors. It uses encrypted memory and includes a hardware random number generator. The Secure Enclave provides all cryptographic operations for Data Protection key management and maintains the integrity of Data Protection even if the kernel has been compromised. Communication between the Secure Enclave and the application processor is isolated to an interrupt-driven mailbox and shared memory data buffers.

3.6 Analysis of Market Implementation Solutions

All the contemporary solutions discussed above are implemented as an integrated TPM module implementation technique using virtualization and mostly rely on ARM TrustZone technology for security. Qualcomm, Samsung, MediaTek and Huawei use ARM TrustZone technology to develop their own closed form solutions to provide security to the end users. As a result, Trust-Zone implementations may vary from vendor to vendor. This comprises almost 99% of the mobile market share. Therefore, the solutions available are ad-hoc, vendor specific and closed form solutions. Moreover, as the available solutions are closed-form and are not available to the application developers or higher layers of mobile architecture hierarchy, hence, there is a need for a unified solution which can be implemented on all the mobile devices without major modifications and available to all the vendors and application developers.

A hardware implementation of the TPM into a dedicated hardware chip complying with the TPM v2.0 is accepted worldwide and had been deployed by the manufacturers in static computing devices since 2006. But the deployment of this dedicated chip in mobile devices arose many complications in which cost, size and power consumption constraints are a main concern. As 99 % of the solutions are based on ARM TrustZone technology therefore an integrated TPM implementation based on TrustZone architecture will be a better solution. Hence, our proposed solution which will be provided in the next chapter will also use the integrated solution of ARM TrustZone with some modifications. Hence it is important to understand and analyze the ARM TrustZone security architecture and its limitations.

3.7 ARM TrustZone Architecture

The ARM TrustZone is an integrated security architecture aiming to offer a security framework enabling the device to that enables a device to defend itself from the imposed threats. TrustZone technology provides the infrastructure foundation of security allowing the SoC designers to implement their own design functions and security environment using it. The main security goal of TrustZone is to provide a programmable environment that permits confidentiality and integrity of diverse set of security functions.

The embedded security is achieved by partitioning all of the SoC's hardware and software resources into two worlds - the secure world for the security subsystem, and the Normal world for everything else. In order to ensure isolation of the secure and non-secure world, hardware logic has been embedded in the TrustZone-enabled AMBA3 AXI bus fabric which creates a strong security hardware perimeter between the two worlds. ARM has enabled some further security extensions of TrustZone in some of its processor cores which enable a single physical processor core to safely and efficiently execute code from both the Normal world and the secure world in a time-sliced fashion. [33]

3.7.1 Processor Architecture:

The following families of the ARM processors are designed to implement the additional security extensions of the Trust zone architecture:

- Cortex-A9 MPCore processors
- Cortex-A9 processors
- ARM1176JZ (F)- processors
- Cortex-A8 processors

In the above processors each physical core is designed to provide two virtual cores, secure and non-secure, and a switching mechanism known as monitor mode. The integration of these two worlds is made possible by the value of the Non-Secure (NS) bit which is originated indirectly from the mode of the virtual processor and sent on the main system bus to access instructions or data. The non-secure world has an open access to the non secure system resources but is restricted to access the secure services whereas secure virtual processor has open access to all the resources. This is illustrated in figure 3.7[19]

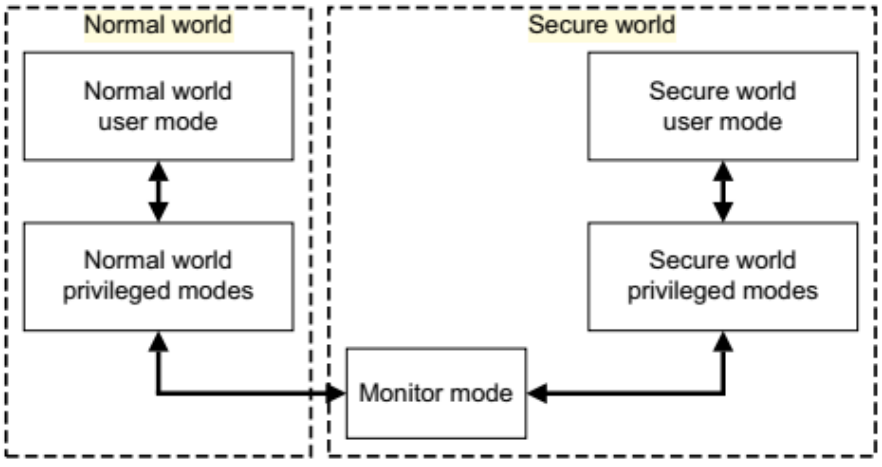

Figure 3.7: ARM TrustZone Virtual Modes Implementing Security Extensitons

The two virtual processors of a single physical processor context switch between the two worlds in a time-sliced fashion through monitor mode which changes the currently running virtual processor mode. The monitor mode is the part of the secure world and the mechanisms by which monitor mode switches the processor from Normal mode are tightly controlled. The monitor mode is triggered by a dedicated instruction set called Secure Monitor Call (SMC) instruction, or by the hardware exception mechanisms which include interrupt requests, fast interrupt requests, external pre-fetch abort or data abort exceptions.

The monitor mode software is designed and implemented by the SoC developers. Its main functions is to store the state of the current world before switching and restoring the state of the mode it has switched to start the processing from where the world stopped previously. As indicated previously the mode of the processor is indicated by the NS-bit which resides in the Secure Configuration Register (SCR). This bit is set to 1 for Normal world and set to 0 for the secure world. It is recommended that the monitor mode should only be allowed to modify the NS bit as it stores the current world state securely before to the other world. Even if secure world software sets the NS-bit the processor will switch to the Normal mode and the system will become vulnerable as the pipelined data of secure world will now be visible to the normal world and not secured by the monitor mode. The normal world software should also not be able to access or modify the contents of the SCR. This is illustrated in figure 3.8

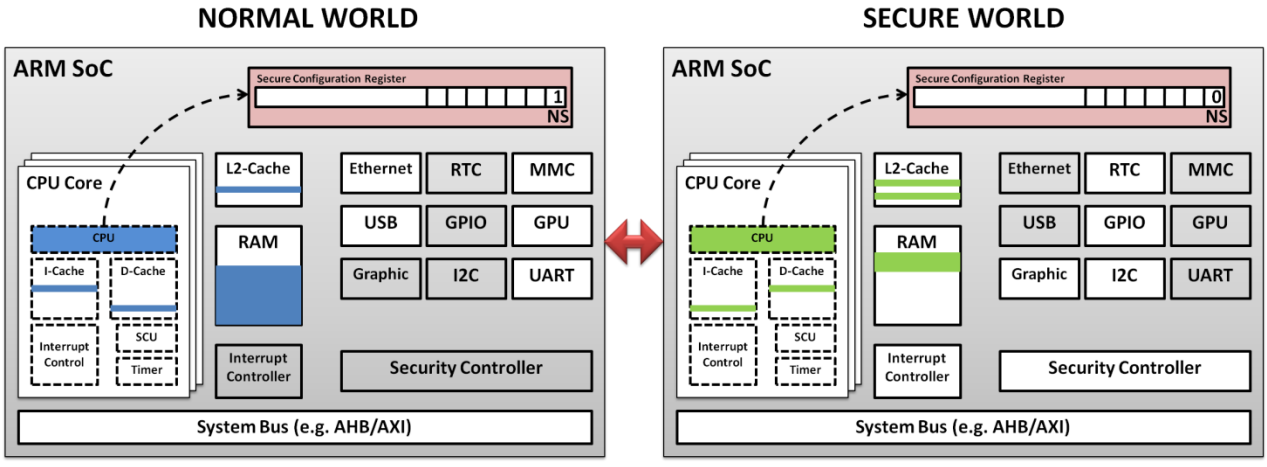

Figure 3.8: Switching Mechanism from Normal World to secure world.

3.7.2 Memory Architecture:

The ARM architecture provides a 32 bit addressing architecture with two possible design configurations. In the first design a 32-bit physical address space is dedicated for secure world processing and 32-bit physical address space for non secure world processing. In the second design the hardware supports memory address space aliasing and the same memory space is aliased between the two worlds and provides two distinct memory locations in the address map. NS bit is the 33rd address bit which indicates the processor mode and the rights of the processor for the provided address location. Hence the secure mode can access all the memory space whereas while the NS bit is 1 and the processor are in the non-secure mode than it can access only the configured non secure address space.

3.7.3 Software Architecture:

A dedicated secure world OS is a complex but powerful design. It can simulate concurrent execution of multiple independent secure world applications, run-time download of new security applications, and secure world tasks that are completely independent of the Normal world environment. One of the advantages of a design based on operating system principles is the use of the processor MMU to separate the secure world memory space into multiple user-space sandboxes. Provided that the secure world kernel software is correctly implemented, security tasks from independent stakeholders can execute at the same time without needing to trust each other. The kernel design can enforce the logical isolation of secure tasks from each other, preventing one secure task from tampering with the memory space of another.

3.7.4 Booting a Secure System:

The most important and vulnerable instance during the life cycle of a secure system is a boot time. This is the instance at which the attackers attempt to break the software while the system is powered down. The attackers either try to modify the booting software or boot the system through image in the flash and if the device without verifications boots the attacker code the system becomes vulnerable. Hence according to the standards a chain of trust is required right from the booting of system established by the ROT that cannot be easily tempered. This is known as a secure boot sequence.

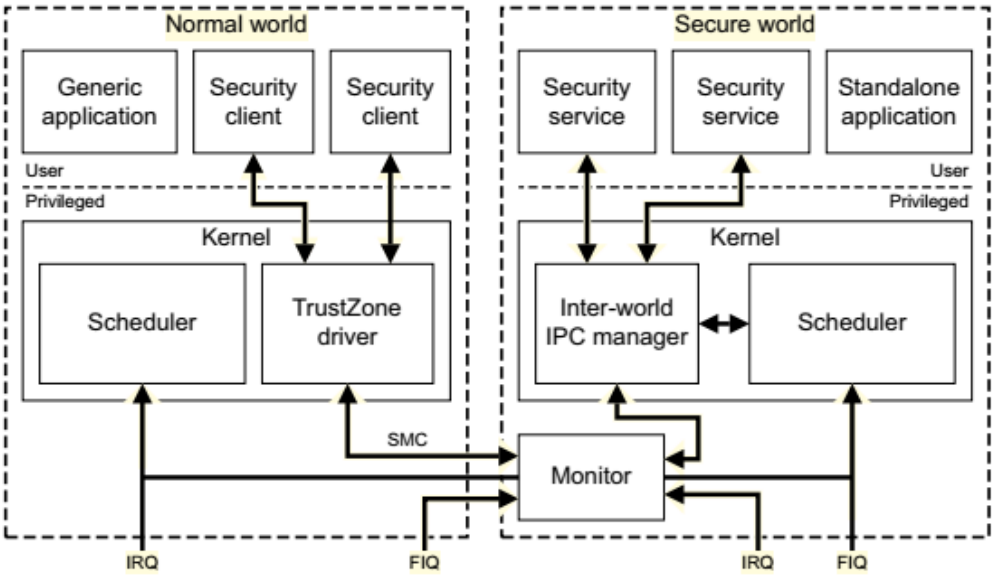

Figure 3.9: Software Architecture of ARM TrustZone

- **Secure Boot Sequence:**

The secure boot sequence of the TrustZone-enabled processor initiates the system in the secure world upon powering up the device. Therefore all the security related checks and configurations are done in the secure mode and the handed over to the normal world for any modifications in the system running.

Figure 3.10 shows the schematic diagram of the secure boot sequence which takes place in TrustZone. After the device is powered on the system will start executing a ROM-based boot-loader which is responsible for initializing critical peripherals such as memory controllers. Afterwards the device boot-loader residing in the external non-volatile memory or flash is executed and all the secure world OS initializations are carried out. Once the secure world configurations are initiated securely the boot sequence passes the control to the normal world boot-loader. The normal world OS is booted and at this point the system is considered to be running.

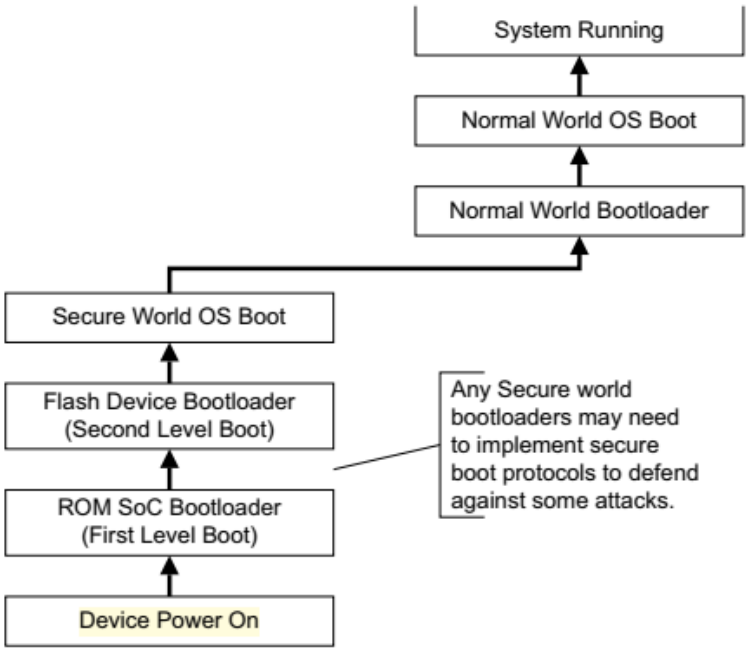

Figure 3.10: A typical boot sequence of an ARM TrustZone based processor

- **Secure Boot:**

  The secure boot mechanism is related to the cryptographic verifications at each stage of the boot sequence. Secure boot ensures that each hardware and software of the device are loaded and executed after cryptographic verification of integrity so as to restrict the unauthorized execution of any malicious or tempered flash images of the software. The secure boot mechanism implements a chain of trust and verifies the authenticity of each component before execution. The key ownership of the chain can change at each progressive stage of the boot sequence. For example the device manufacturer's public key may authenticate the first boot loader, the secure world OS may be authenticated with a secondary public key related to its binary and the normal world OS may use its own residing public key to authenticate its OS and the applications that it loads.

3.7.5 TrustZone API:

ARM TrustZone have developed their own standardized software API, called the TrustZone API (TZAPI), for the development of security solutions. TZAPI provides a trusted interface between the client applications running in the Normal mode and the

secure world trusted services and functions. The secure functions (encryption, signatures, integrity checks, etc) are only accessed via the monitor mode and not accessible to any other operational software of device [20,21]. Figure 3.11 shows the ARM Trust Zone access mechanism. If a client application or OS requires secure services, it requests the TZ driver via a TZAPI. The TZDriver sends an appropriate SMC call to the monitor mode. The monitor mode switches the processor to the secure mode and the requested operation is carried out. After the secure operation is completed the monitor mode transmits the results to the TZAPI driver and switches all the processors back to normal world.

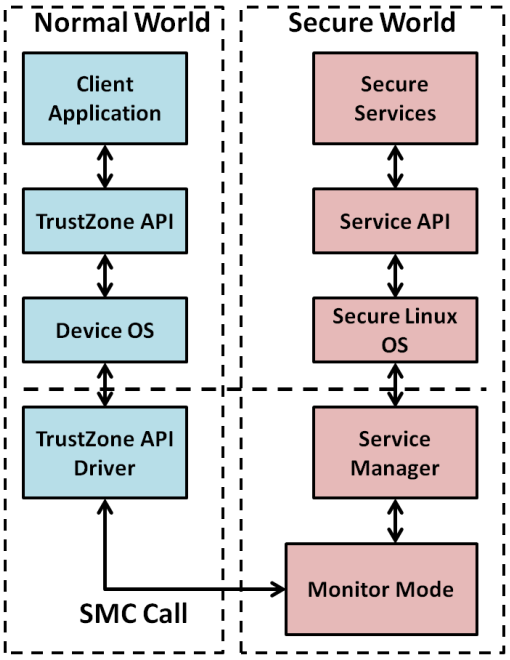

Figure 3.11: ARM TrustZone Access Mechanism

TZAPI is basically a communication interface between the client applications and the secure services through command requests. This interface provides a mechanism for the authenticating the applications, verify their installed services, and permits run-time download of new security services. Although the TrustZone API is targeted at systems using a TrustZone-enabled processor, and tries to take advantage of the available hardware features such as World-shared memory, it is designed to be portable to almost any implementation of a secure environment.

3.8 Android Exploits Database

A list of android exploits has been presented which lists the exploits only related to ARM TrustZone. Moreover a short description of the exploits with the CVE code and Score is also mentioned for reference (CVE vulnerability data is taken from National Vulnerability Database feeds provided by NIST and other data sources including exploit-db and Metasploit modules). Only high CVSS score exploits have been listed which support our analysis on the shortcomings present in ARM TrustZone.

Table 3.1: List of ARM TrustZone Exploits on Android Platform

S.No	CVE Code	CVSS Score	Exploit Platform	Vulnerability Type	Exploit Description
1	2015-8999	9.3	Google Android 7.1.1 and earlier	Overflow	In TrustZone buffer overflow vulnerability can potentially occur in all Android releases from CAF using the Linux kernel while loading an ELF file.
2	2015-9005	9.3	Google Android 7.1.1 and earlier	Overflow	In TrustZone in all Android releases from CAF using the Linux kernel, an Integer Overflow to Buffer Overflow vulnerability could potentially exist.
3	2015-8998	9.3	Google Android 7.1.1 and earlier	Overflow	In TrustZone integer overflow vulnerability can potentially occur in all Android releases from CAF using the Linux kernel.
4	2016-2431	9.3	Google Android 6.0.1 and earlier	Gain privileges	The Qualcomm TrustZone component in Android on Nexus 5, Nexus 6, Nexus 7, and Android One devices allows attackers to gain privileges via a crafted application, aka internal bug.
5	2013-3051	6.2	Motorola Android 4.1.2, Razor, Atrix, Qualcomm MSM8960	Admin	The TrustZone kernel, when used in conjunction with a certain chipset does not verify the association between a certain physical-address argument and a memory region, which allows local users to unlock the boot loader by using kernel mode to perform crafted SMC operations
6	2016-8763	9.3	Huawei P9, P9 Lite, P8, P8 Lite	Gain privileges	The TrustZone driver in Huawei phones has an improper resource release vulnerability, which allows attackers to cause a system restart or privilege elevation.
7	2014-9948	9.3	Google Android	Gain Information	In TrustZone in all Android releases from CAF using the Linux kernel, Improper Validation of Array Index vulnerability potentially exist.
8	2015-9002	9.3	Google	DRM	In TrustZone an out-of-range pointer offset

			Android 7.1.1 and earlier		vulnerability can potentially occur in a DRM routine in all Android releases from CAF using the Linux kernel.
9	2015-6647	9.3	Google Android 6 and earlier	Gain privileges	The Widevine QSEE TrustZone application in Android allows attackers to gain privileges via a crafted application that leverages QSEECOM access
10	2014-9945	9.3	Android	Authorization	In TrustZone in all Android releases from CAF using the Linux kernel, Improper Authorization vulnerability could potentially exist.
11	2014-9935	9.3	Android	Overflow	In TrustZone an integer overflow vulnerability leading to a buffer overflow could potentially occur in a DRM routine in all Android releases from CAF using the Linux kernel.
12	2015-8996	7.6	Android	Time-of-Check Time-of-Use Race	In TrustZone a time-of-check time-of-use race condition could potentially exist in a QFPROM routine in all Android releases from CAF using the Linux kernel.
13	2016-1029	9.3	Android	Time-of-Check Time-of-Use Race	In TrustZone in all Android releases from CAF using the Linux kernel, a Time-of-Check Time-of-Use Race Condition vulnerability could potentially exist.
14	2015-9007	9.3	Android	Gain Information	In TrustZone in all Android releases from CAF using the Linux kernel, Double Free vulnerability could potentially exist.
15	2015-9003	9.3	Android	Gain Information	In TrustZone a cryptographic issue can potentially occur in all Android releases from CAF using the Linux kernel.
16	2015-6639	9.3	Google Android 6.0 and earlier	Gain privileges	The Widevine QSEE TrustZone application in Android allows attackers to gain privileges via a crafted application that leverages QSEECOM access
17	2016-0803	10	Google Android 6.0 and earlier	DOS, Code Overflow, Memory Corruption	libstagefright in mediaserver in Android allows remote attackers to execute arbitrary code or cause a DOS (memory corruption) via a crafted media file that triggers a large memory allocation in the SoftMPEG4Encoder or SoftVPXEncoder component
18	2016-0801	8.3	Google Android 6.0 and earlier Apple 9.0. Mac 10.0	DOS, Code Overflow, Memory corruption	The Broadcom Wi-Fi driver in the kernel in Android 4 allows remote attackers to execute arbitrary code or cause a denial of service (memory corruption) via crafted wireless control message packets
19	2015-9073	10	Qualcomm	Overflow	In all Qualcomm products with Android releases from CAF using the Linux kernel, an untrusted pointer dereference can occur

					in a TrustZone syscall.
20	2015-9072	10	Qualcomm	Gain Information	In all Qualcomm products with Android releases from CAF using the Linux kernel, an untrusted pointer dereference can occur in a TrustZone syscall.
21	2015-9071	10	Qualcomm	Overflow	In all Qualcomm products with Android releases from CAF using the Linux kernel, a buffer over-read vulnerability exists in a TrustZone syscall.
22	1015-9070	10	Qualcomm	Gain Information	In all Qualcomm products with Android releases from CAF using the Linux kernel, the Secure File System can become corrupted.
23	1015-9068	10	Qualcomm	Gain Information	In all Qualcomm products with Android releases from CAF using the Linux kernel, an argument to a mink syscall is not properly validated.
24	1015-9067	10	Qualcomm	Gain Information	In all Qualcomm products with Android releases from CAF using the Linux kernel, a potential compiler optimization of memset() is addressed.
25	1015-9066	10	Qualcomm	Overflow	In all Qualcomm products with Android releases from CAF using the Linux kernel, a buffer overflow vulnerability exists in an Inter-RAT procedure.
26	1015-9065	10	Qualcomm	Gain Information	In all Qualcomm products with Android releases from CAF using the Linux kernel, a UE can respond to a UEInformationRequest before Access Stratum security is established.
27	1015-9070	10	Qualcomm	Gain Information	In all Qualcomm products with Android releases from CAF using the Linux kernel, the UE can send IMEI or IMEISV to the network on a network request before NAS security has been activated.
28	1015-9070	10	Qualcomm	Overflow	In all Qualcomm products with Android releases from CAF using the Linux kernel, a buffer overflow vulnerability exists in a procedure involving a remote UIM client.
29	1015-9070	10	Qualcomm	Overflow	In all Qualcomm products with Android releases from CAF using the Linux kernel, an integer overflow to buffer overflow vulnerability exists when loading an ELF file.
30	1015-9070	10	Qualcomm	DRM	In all Qualcomm products with Android releases from CAF using the Linux kernel, playReady DRM failed to check a length potentially leading to unauthorized access to secure memory.

The list of Qualcomm exploits range from CVE-1015-9033 to CVE-1015-9073. Some are listed in the table. All exploits listed are in one way or other related to the shortcomings mentioned in the next section. The overflow exploits especially indicate the non secure storage of data in the secure worlds. Moreover gaining high privileges or access control in the system also indicates bypassing of the authentication mechanisms and weak ROT implementation. Qualcomm takes almost 40% of the market share (discussed in the previous chapter) and its security solution based on TrustZone is used widely by different vendors. The exploits listed related to it are independent of the OS being used or MDM solutions programmed on the higher layers of the mobile devices. Hence the vulnerabilities present in TrustZone need to be addressed to make the mobile systems secure.

3.9 Shortcomings of ARM TrustZone

More than 99% of the mobile market is held by ARM processors and use TrustZone for their security solution implementations. Although the ARM TrustZone documentation explains the mechanism to securely configure the processor, memory and I/O devices while processing in different virtualized environments, some observations are made across all the vendor solutions and products while analyzing their core. These are listed below:

3.9.1 Absence of Secure Storage:

One of the extremely valuable functions of a TPM is its ability to seal a private key under the hash of the code using it. This means that one can create a private key which can only be read by a piece of code that hashes to a certain value. TrustZone in itself does not provide any way to store the secret data. So a key can be created in the secure world but cannot be stored securely. Similarly due to single memory distribution between the two worlds, the secure data of secure world which should not be accessible to the normal world can be captured while operating in the normal world. The problem of absence of secure storage has arisen because TrustZone specification doesn’t provide any mechanism to implement secure storage. As ARM TrustZone does not provide secure storage, which is the basic and essential capability required to build a secure hardware

rooted system according to the standards. Hence it does not comply with the standards of NIST and MTM and so neither do the 90% of the security solutions available in the market as their core comprises of ARM TrustZone.

3.9.2 Absence of Secure Clock:

Mostly all the secure systems inherit a secure clock. Although TrustZone provides a mechanism to protect memory, interrupts, ARM peripheral bus and other system buses but it fails to guarantee the secure transmission of data on its peripherals and even when they can be programmed by the controller while operating in the normal world. Malicious codes can be used to program the peripheral insecure.

3.9.3 Lack of Secure Entropy and Persistent Counters:

Most trusted systems make use of cryptography. However, the TrustZone specification is silent on offering a secure entropy source or a monotonically increasing persistent counter. As a result, most SoCs lack an entropy pool that can only be read from the secure world, and a counter that can persist across reboots and cannot be incremented by the normal world.

3.9.4 Security Provided Through Virtualization Technique:

Each of the physical processor cores in the ARM TrustZone processor designs provide two virtual cores, Secure and Non-Secure (Normal), and a monitor mode to toggle between them. This permits trivial incorporation of the virtual processors into the system security mechanisms; the secure virtual processor can see all the resources whereas the non-secure virtual processor can only use un-trusted system resources. Therefore the TrustZone architecture is software based and does not contains the security advantages of a dedicated hardware TPM chip. Moreover, although ARM offers virtualization extensions, it is not mandatory for the vendor to apply theses security extensions. As a result, many ARM-based SoC smart phones lack this security virtualization support and only operate in the normal world.

3.10 COMPLIANCE OF TRUSTZONE WITH STANDARDS

ARM TrustZone provides an open source security solution architecture which the device manufacturers can utilize in their own solutions. Trust Zone provides an integrated

solution using time sliced virtualization technique. It provides a hardware virtualized separation for both the worlds operation but the cryptographic algorithms implementation and the firmware design is left to the developers choice. Therefore the cryptographic algorithmic utilization strength and compliance depends on the mobile device manufacturer's solution such as Qualcomm, Samsung etc. Hence the four roots of trust which depends on the cryptographic strength; ROTI, ROTV, ROTM and ROTR are partially dependent on the device manufacturer. As far as their hardware rooted side security is concerned all the 5 ROTs do not comply to the NIST and modified MTM (TPM v2.0) as TrustZone lacks secure storage and is unable to protect secure world data from disclosure. TrustZone exploits related to Overflow, authentication bypass, and authorization bypass prove the lack of ROTs. The API developed by TrustZone complies with the standard. It creates a secure mechanism to transport the ROT to the OS and Applications and provides a secure boot mechanism to boot the system.

3.11 Conclusion

In this chapter we have discussed the internal key components of a TPM and their implementation techniques with the TCG specifications related to them. It was concluded that integrated TPM techniques provides a better solution fulfilling the tradeoff between cost, size, power consumption and performance metrics of resource constraint devices. ARM TrustZone is a widely deployed security architecture of most of the industrial mobile security solutions. But at the same time ARM TrustZone does not complies with the security standards unable to provide secure ROTs. A list of exploits related to the TrustZone vulnerabilities was also presented. Hence all the available market mobile security solutions are non-standardized, ad-hoc, vendor specific and closed form solutions. Hence a suitable solution is required to overcome the limitations in standards and ARM TrustZone. In the next chapter the suggested hardware rooted mobile security solution will be presented which will provide a unified platform to the industry for standardized and open source mobile security solutions.

Chapter 4

mTPM: PROPOSED SECURITY MODEL FOR MOBILE DEVICES

4.1 Introduction

In the preceding chapters, existing security standards including NIST and MTM were described and analyzed in detail. Thereafter, the popular implementation methodology of MTM commonly known as Arm TrustZone has also been described and analyzed. Analyzing both aspects, it was concluded that currently prevalent security system based on MTM implemented through TrustZone Technology provided in ARM processors is inadequate to provide requisite security in mobile devices. In this chapter, a possible solution to the shortcoming of this security system has been proposed. In view of popularity and suitability of ARM TrustZone Technology, the proposed solution has been developed around it so that minimum changes may be required both in hardware and associated software segment. The solution with be described in three distinct parts namely conceptual framework, hardware implementation and software interface. However, before describing the solution, it has been considered appropriate to revisit the conclusions of analytical results of the existing technology especially with a focus on shortcoming that have particularly been addressed in the proposed solution.

4.2 Review of Conclusions of Analytical Results

1) The motivation of this research is based on the fact that the current software based security solutions in mobile devices are unable to provide adequate assurance to the users' data and applications. **Some limitations observed in the MTM standard indicate that it provides relatively inferior security features as compared to TPM v2.0.** Hence **a hardware rooted security solution is considered essentially desirable for mobile devices which lies in the core of these devices and can be trusted upon for the security capabilities**.

2) Different hardware rooted security standards for the mobile security provided the functional aspects of the security and kept the implementation aspects open for the developer. Therefore, the **industrial solutions implementing these standards are not standardized on a unified methodology**.
3) The discussion on TPM implementation techniques concluded that **integrated TPM implementation technique would be the best option for deploying security in the mobile device environment having low cost, small size and low power consumption**.
4) As ARM has captured almost all of the market, ARM TrustZone architecture was described and analyzed in detail. ARM TrustZone uses hardware virtualization technique to implement security and shares the processor, memory and other hardware essentials between secure world and normal world of operation. Although ARM TrustZone technology follows integrated implementation methodology, it has been found with certain vulnerabilities that have been reportedly exploited to crack the security system. **Hence it was concluded that ARM TrustZone provides relatively lower security than a dedicated hardware TPM chip deployed in laptops and desktops.**
5) Various commercial implementation solutions were discussed which included those from Qualcomm, Samsung, Intel, MediaTek and Boeing. After their analysis it was concluded that nearly 99% of the market uses ARM processors for manufacturing the core of their mobile devices and implement security solutions based on ARM TrustZone technology with their own closed form implementations. Hence all the commercial solutions available are adhoc, vendor specific and closed form solutions.

Therefore, a standardized mobile security solution is required which provides the same level of security as a dedicated TPM chip and which complies with the standards and provides a unified implementation methodology for the developers to develop a uniform and open source security solution for the entire array of mobile devices.

4.3 Proposed Security Model - mTPM

In this section the proposed solution will be described. An effort has been made to offer a standardized mobile security that should address the limitation of the vendor

specific existing solutions both from architectural as well as implementation perspectives. From architectural perspective, the proposed solution is a combination of MTM and TPM v2.0. Whereas from implementation perspective, it is built around ARM TrustZone duly coupled with TPM philosophy, wherever applicable, to provide reliable ROT components. Although to achieve the ultimate security objectives, certain hardware upgrades in ARM hardware architecture have been suggested, the solution has been kept backward compatible with existing hardware, of course, with known vulnerabilities and constraints.

The proposed mTPM has been designed to provide a TEE that acts as a basic OS for the secure world for the provision of ROT as per TPM v2.0. By leveraging the time-sliced isolation feature of ARM TrustZone coupled with additional hardware assisted security components, mTPM will provide superior secure execution environment. Using this concept, mTPM offers two fundamental security guarantees:

- **Confidentiality:** The whole execution of the mTPM (including its secret variables and internal execution state) is hidden from the rest of the system. Only the mTPM's inputs and outputs, but no intermediate states, are observable.
- **Integrity:** The system cannot affect the behavior of the mTPM. Because, it has been ensured that the mTPM's commands are executed correctly according to the TPM 2.0 specifications.

4.3.1 Suggested Modifications in Standard and mTPM:

As described earlier, TCG specifications in MTM discussed in the Chapter 2 specify obsolete cryptographic primitives and their respective ROTs and have left open for the implementers. Therefore, the proposed model will specify all of them so that to be standardized for the entire industry. Accordingly, the proposed model gives following specifications:

1. The Proposed Model will implement all the specifications of TPM v2.0 (due to enhanced security requirements) with the desired modifications for mobile platform. The MTM standard should also be compatible to TPM v2.0 to bring all the TPM manufacturers at a unified platform.

2. TPM specifies an isolated monolithic implementation of all cryptographic functions with built-in storage and processing. However, the same is not practical in case of mobile devices due to size, cost, and power consumption perspectives. Therefore, it is proposed that the security functions are integrated into a dedicated processor core of the main processor. This will allow a flexible, cost effective and low power consumption implementation. However, in order to achieve the same degree of security, all ROTs must be implemented in hardware elements with strict red and black isolation. The details of this aspect will be covered in next sub-section.
3. With the advent of multi-core processors in smart phones, they are not considered as computing resource constraint devices. Therefore, they are expected to process traditional cryptographic primitive function meant for high performance computers. However, these smart phones are also used to become part of IoT network and interact with low-power low-performance sensors and actuators. Since these sensors and actuators can only have lightweight cryptographic primitives, the smart phones should also have compatible lightweight cryptographic primitives. Therefore, mTPM proposes lightweight cryptographic primitive along with traditional cryptographic primitives as recommended by TPM v.2.0. As described in Chapter 2, a special study was conducted for selection of appropriate cryptographic primitives for mTPM. Specifically, following additional lightweight cryptographic primitives are proposed:
    a. Symmetric Cryptographic Algorithms: SIMON/SPECK, PRESENT
    b. Asymmetric Cryptographic Algorithms: ECC
    c. Hashing Algorithms: QUARK, SPONGENT
4. Similar to requirement for lightweight cryptographic primitives, there is also a requirement of suitable cryptographic mode of operation. After a literature survey of comparative analyses, it was deduced the “Counter Mode” is most appropriate for resource constraint devices due to several reason. The biggest advantage of the counter mode over most block cipher modes is the possibility to pre-compute key stream for all cipher output blocks in parallel. Because it is possible to parallelize both encryption and decryption, the counter mode achieves a very high throughput especially for streaming data that is very common in today’s Internet connected

devices. In case Authenticated Encryption scheme is desired, many standardized scheme especially GCM provide suitable option.

5) There are other aspects of TPM v.2.0 required to be made complaint from functional perspective. However, since they pertain to hardware implementation such as Random Number Generator (RNG), secure memory for attestation and authorization, secure clock etc., they will be discussed in the following sub-section containing implementation aspects of proposed mTPM.

4.3.2 Proposed Implementation Solution for mTPM:

The discussion on TPM implementation techniques concluded that integrated TPM implementation technique would be the best option for deploying security in the mobile device environment having low cost, small size and low power consumption. As ARM TrustZone technology follows integrated implementation methodology, it has been found to be a suitable choice as the foundation for the proposed mTPM solution. Moreover, as ARM has captured almost all of the market. It was considered to a best choice as minimum changes will be required to adopt the proposed mTPM. Since ARM TrustZone used hardware virtualization technique to implement security but it shares the processor, memory and other hardware essentials between secure world and normal world of operation. Therefore, it has been found with certain vulnerabilities that have been reportedly exploited to crack the security system. The proposed mTPM actually works around ARM TrustZone such that to alleviate its shortcoming and make it conceptually compliant to TPM 2.0

Proposed implementation technique comprises of the security implementation changes in the integrated technology of ARM TrustZone and the additional security enhancements required to support the integrated technique in compliance to TPM v2.0.

1. Dedication of Security Processor:

In the TPM implementation section three different TPM implementations techniques were discussed and a conclusion was drawn that an integrated TPM is a preferred solution technique for mobile devices due to hardware incorporated security, low computing resource processors, less surface area and power consumption constraints of the embedded systems. Hence the proposed mTPM

implementation model is an integrated TPM like Trust-Zone but different from it by implementation aspect. The primary difference is that ARM TrustZone transforms the main processor into two processors by time multiplexing it into two execution environments of Secure and Normal world. Each core of the processor switched its execution mode depending on selection of "World" the processor's operating mode. Whereas the proposed mTPM model dedicates a single core i.e., Core 0 out of the multi-core processors for the Secure World and all the remaining cores for normal for permanently without switching their role at anytime. This arrangement has several advantages:

a. The dedicated core for TPM services truly comply the TPM v2.0 requirements as Core 0 will never perform any other functions (for Normal World).

b. The integrated TPM processing element provide superior security as compared to isolated hardware device as the bus for communication between Secure and Normal worlds is inside the main processor and inaccessible for interception externally.

c. The dedicated core is a programmable device and provides more programmable user flexibility (instruction set) then a hardware TPM chip. This will provide us with the flexibility of selecting and altering different cryptographic algorithms embedded in the core for security purposes and updated later on.

d. It will also not increase the die size of the SoC as no separate module is being integrated with the processor.

e. It will overall decrease power consumption as the core will operate only when cryptographic and mTPM services will be needed.

2. Memory Storage:

ARM TrustZone provides no guidelines as to how to manage the memory as ROM and RAM are both physically shared between Secure and Normal worlds. The use of TrustZone is not entirely opaque to the non-secure side because hidden physical resources appear as holes in the physical address space. The unavailability of secure storage reduces the usefulness of TrustZone as trusted technology for secure

world computing. Especially, unavailability of memory for cryptographic variable is a serious shortcoming. Although monitor kernel defines Secure and Normal world ROM and RAM allocation in run-time, the same are actually physically shared. Effectively, this shortcoming has been exploited the most as amply described in the last chapter. Keeping this aspect in view, mTPM architecture requires following arrangement:

a. A dedicated "Secure Memory" for storing cryptographic keys, Random Data Pool, Application level security parameters, and intermediate stage data under processing should be provided.

b. Secondly, there are command mechanisms in ARM TrustZone (Monitor Kernel in SE Linux) for allocating static (permanent) allocation of ROM and RAM to a particular processing core. mTPM recommendations included this aspect to be configure for Core 0 to prevent any chance of exposure of secure world data to Normal world.

c. In addition to this an OTP storage is required to program Encryption Keys for Application and OS provider. This storage should be fusable after write to prevent read back at a later stage.

d. An additional optional arrangement could be done to store sensitive data duly protected by cryptography in external memory controller such as eMMC. This storage provides a replay protected memory block (RPMB) partition. Like its name suggests, RPMB is a mechanism for storing data in an authenticated and replay-protected manner.

3. Secure Entropy Source:

TPM specifications require an Entropy Source (a pool of Random Numbers) generated by True Random Number Generator (TRNG). It is used to draw cryptographic variable/keys. However, ARM SoC has generally ignored this requirement out rightly. Since this is an essential requirement for secure processing, mTPM has included a Secure Entropy Source (SES) in it. An SES consists of a

TRNG and a Secure Memory for its storage. The requirement of Secure Memory as been defined earlier but the source of random number is defined as under:

a. A hardware TRNG is to be included in the ARM SoC which should be accessible in secure world processing only. The data generated by TRNG should be stored in Secure Memory as defined earlier.

b. In case, TRNG is not available, then Random Numbers may be generated by sampling an analog (audio or RF) signal lines while the signal is not present. However, the same may not have the requisite randomness property. To achieve this, it is recommended to mix this data with deterministic but cryptographically secure random number generator such as Blum-Blum-Shub (BBS) Generator. The analog signal sampled data duly tested for basic randomness tests is to be XOR bit-by-bit with BBS Generator that will be seeded from segment of the same analog signal sampled data. The resultant data may be stored in Secure Memory dedicated to secure world processing.

4. Secure Clock:

Similar to TPM, mTPM also requires a hardware Secure Clock (Sclk) that is accessible to Secure world processing only for configuration. An Sclk is required to perform time bound service refusal or time bound authorization in by secure world. The Sclk should be hardware temper proof and accessible to Normal world only through monitor kernel for Read only operation. This clock makes Non-Volatile entries and never rolls backward. In case, Secure Clock is not provided, the monitor should use tradition Real Time Clock (RTC) that should be available to both Secure and Normal worlds. However, in this case it should only be used for time bound lockout but not time bound authorization.

5. Resource Allocation and Availability:

Hiding memory and internal peripherals devices from the non-secure world is one of the main features of TrustZone. However, TrustZone does not define which segments of memory and peripherals are protected by this mechanism. Furthermore, access to central devices (such as the system control registers) cannot be transparently

emulated. This is left entirely in the hands of the SoC vendors. SoC vendors lock down the firmware and do not share it for re-configurability at monitor mode layer. In mTPM all such parameter will be available for re-configuration at OS level for flexibility of hardware allocation especially for virtualization in security processes.

- **Cryptographic Key Hierarchy:**

  Just like TPM 2.0, mTPM provides four hierarchies of Encryption Keys (for authorization/signing/attestation) and SRK (for encryption) namely EH, SH, PH and NH for greater flexibility. These four hierarchies are intended to be used by platform manufacturers and the Storage and Endorsement hierarchies, and the Null hierarchy will be used by OS's and OS-present applications. This arrangement will encourage the vendors to make firmware/boot loader controllers accessible to OS providers and end-user applications.

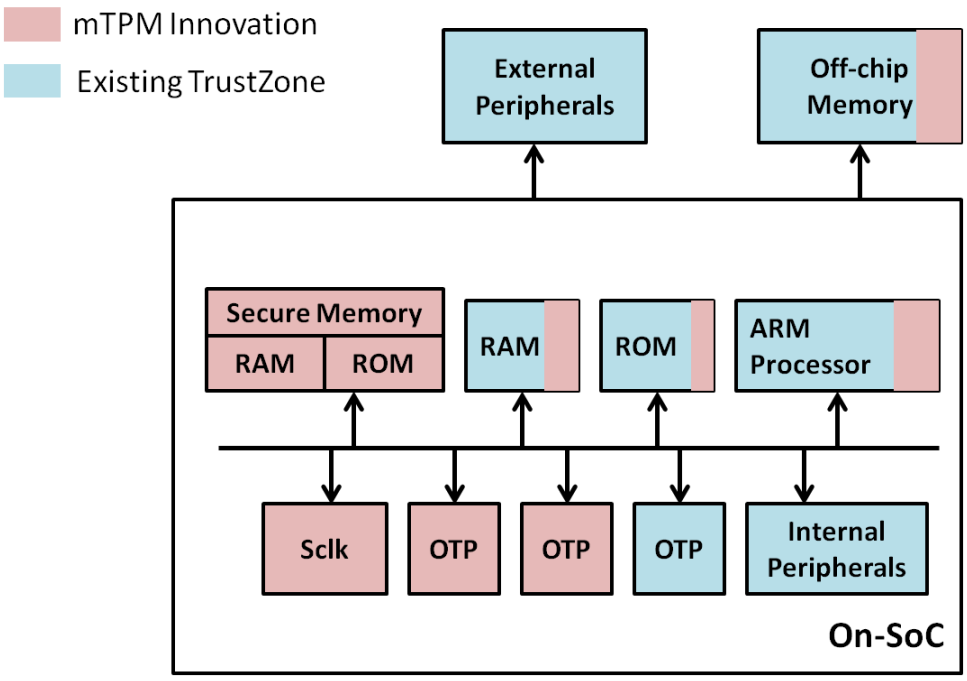

Figure 4.1: mTPM and TrustZone Combined SoC Components

Figure 4.1 shows the mTPM additive aspects in the SoC fabrication. The above mentioned aspects are only the salient ones that are essentially required for upgrading in the ARM TrustZone architecture and make mTPM conformable to TPM 2.0

specifications. Overall the enhancements cover several inter-related aspects to provide the comprehensive TEE.

Programming the single core as a secure processing core is one single aspect of developing a secure foundation for a mobile system. At the same time it is also necessary that the core should incorporate all the hardware roots of trust and fulfill the concept of protected capabilities and shielded locations and should guarantee that no security-critical information is leaked to the un-trusted parts of the system or applications. To achieve this objective according to TPM v2.0 it is required that the dedicated core should be physically isolated from the logical separation architecture of the multi-core processors. Moreover as the functional and physical requirements of the cryptographic processors are different from the general purpose processors, hence the architecture of the core should be modified accordingly. A secure crypto-processor

- Accelerates the cryptographic process i.e. encryption, decryption, hashing, signatures, etc
- Detects and protects against tempering i.e. the processor is temper-proof
- Contains the intrusion detection capabilities and protects the data disclosure. This could be achieved through hardware firewall behind all Secure Memory elements and internal peripherals.
- Consists of secure I/O ports i.e. the I/O ports are separate for input (red signals) and output (black signals) assuring that no sensitive information leaks from the processing segment.
- Contains clear segregation in processing of data i.e. data of sensitive or classified plain-text information (red signals) and encrypted information, or cipher-text (black signals) should be processes separately.
- Contains its separate and segregated memory i.e. a separate RAM for non volatile data at run (it also contains round keys and each round data) and a separate ROM (to store device keys, verification keys, certificates, etc).

Hence a software-flexible hardware solution can be achieved by isolating a single core out of the multi-core processors and designing it for the hardware ROT capabilities.

Figure 4.2 shows suggested model implementation diagrammatically. As depicted in the figure dedicated memory area and core will be used for secure processing instead of using the same memory and cores for secure and non secure services in a time sliced manner.

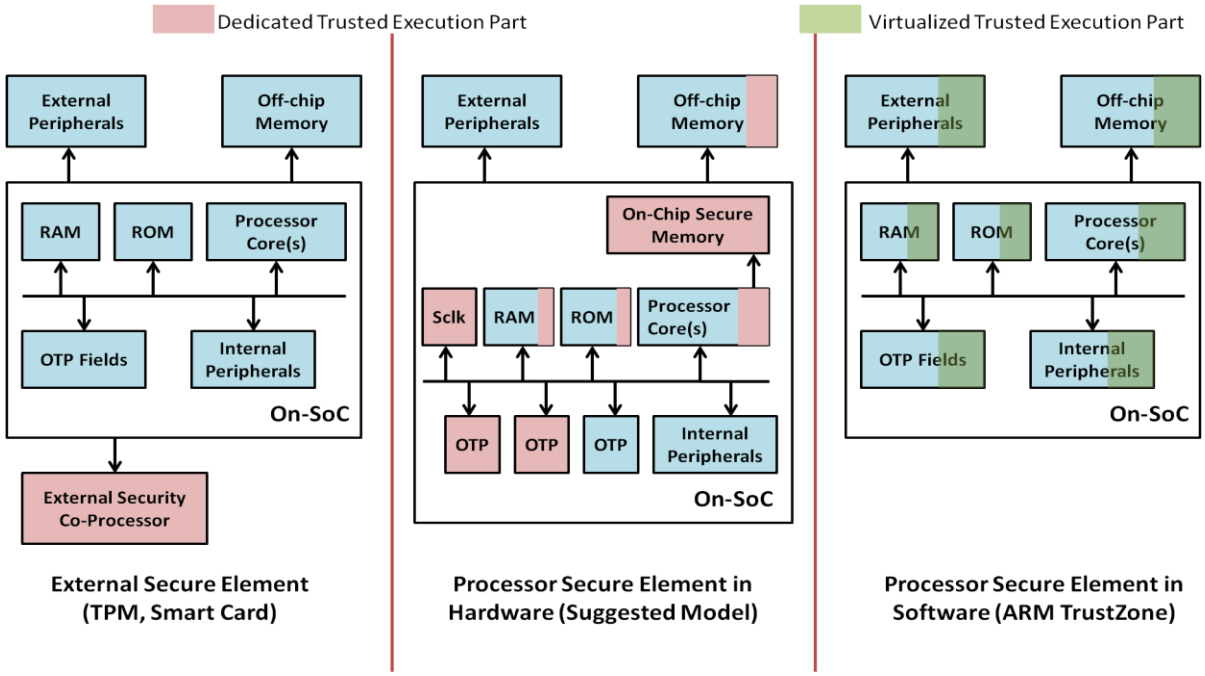

Figure 4.2: TEE Hardware Realization Alternatives

4.4 Accessing Secure Resources from OS and Applications:

The functions provided by the secure core (encryption, signatures, integrity checks, etc) should only be accessed via the monitor mode and not accessible to any other operational software of device. Figure 4.3 shows the left figure shows the ARM Trust Zone access mechanism in mobile phones and the right figure shows the TPM access mechanisms in laptops and desktops. As the security mechanism deployed in laptops and desktops is hard to break hence its security mechanism will be used in our model. In TrustZone client application or OS requiring secure services requests the TZ driver via a TZAPI for the services. The TZDriver sends an appropriate SMC call to the monitor mode. The monitor mode switches the processor to the secure mode and the requested operation is carried out. The monitor mode transmits the results to the TZAPI driver and switches the processor back to normal world. Whereas in case of TPM hardware, The TPM Base Services (TBS) are responsible to formulate and carry the commands of the

user mode applications and OS via TPM driver to and from the TPM. All the other operations keep on functions on the normal CPU of the device.

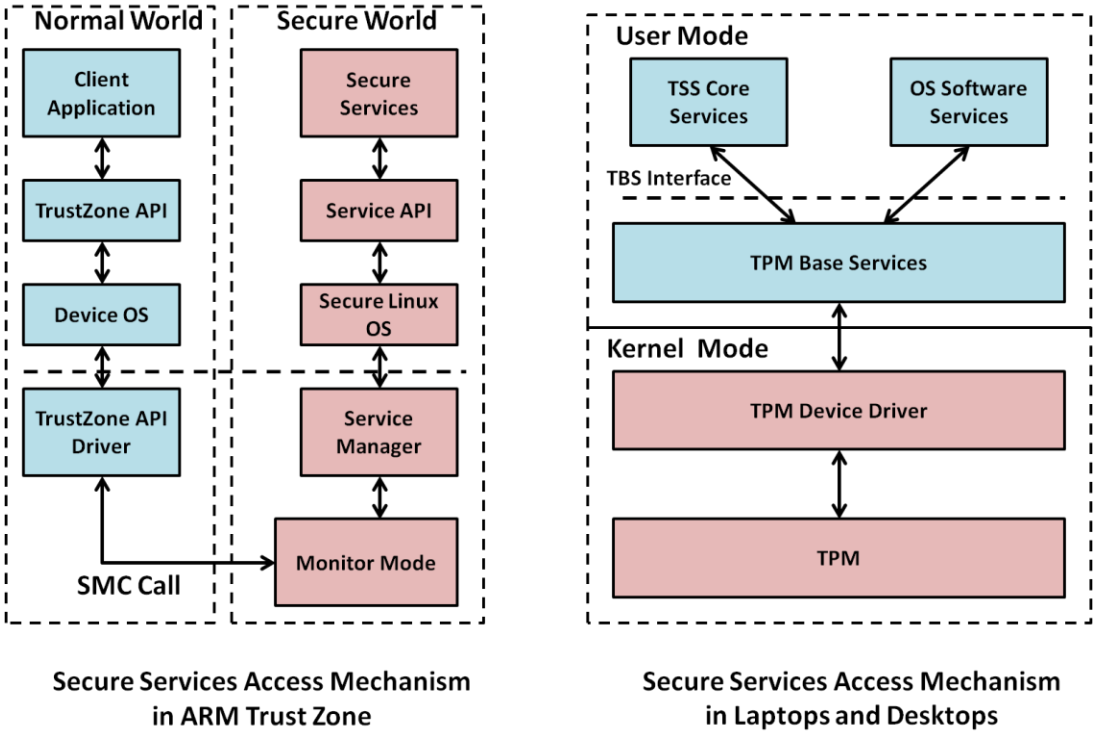

Figure 4.3: Secure Services Access Mechanism

In case of multi-core architecture of TrustZone, in order to process the secure world command the monitor mode switches the required number of cores to the secure world and all the other cores remain idle and stop functioning even in the normal worlds. This is because the memories and cache of both the world are shared and if other cores will operate in normal world data sharing is possible. After the secure operation is completed the monitor mode transmits the results to the TZAPI driver and switches all the processors back to normal world. This is illustrated in figure 4.4

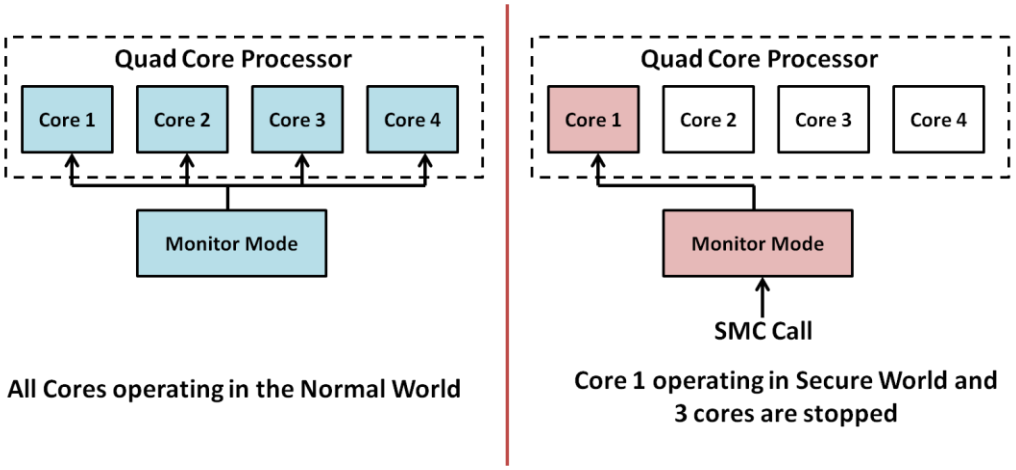

Figure 4.4: ARM TrustZone Secure Service Execution Mechanism

In order to implement static computing TPM access mechanism some changes will be required in the Trust Zone access mechanism which will also complement our modified integrated model. In the proposed security model all the upper layer API's including TZAPI will function previously. This will provide us with the advantage that OS and application developers will not have to modify their programs for the mTPM. Only now the functionality programmed in the monitor mode (present in the firmware) will change. Previously monitor mode switched the processor between the two worlds depending upon the operation requested. Now the monitor mode will redirect the secure services request towards the secure world processing core "core 0" (core 1 in figure 4.5) and normal world operation towards all other processors which may work normally even during the secure operations. Figure 4.5 illustrates this phenomenon.

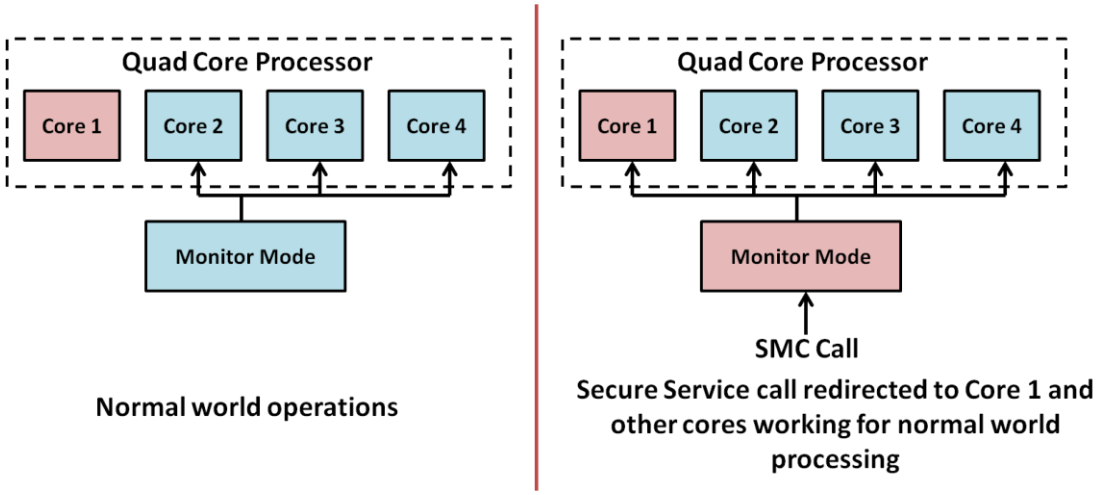

Figure 4.5: Proposed Model Security Operation Access Mechanism

4.5 Conclusion

In this chapter a new security model for mobile device namely mTPM has been presented. As it has been quite elaborated, mTPM provides all the security provisions of TPM 2.0 for mobile devices including but not limited to resource constraint wearable, IoT sensors and actuators. In the next chapter the practical implementation feasibility of the proposed security model mTPM will be discussed and the model will be analyzed for the compliance with the standards.

Chapter 5

PROPOSED mTPM: DISCUSSION AND ANALYSIS

5.1 Introduction

In the previous chapter a new mobile security model – mTPM was proposed. mTPM is a mobile security model which targets almost all the smart phones of the industry which include Qualcomm, Samsung, MediaTek, and Huawei. Nearly 99% of these industrial solutions are based on ARM Trust Zone architecture. Therefore in the proposed mTPM model the limitations in the existing and dominant hardware solution i.e. ARM TrustZone are removed to make the solution backward compatible to the existing technology. Moreover the new model not only complies with the existing standards but also suggests modifications in the mobile standards and also implements them in its model. In this chapter we will analyze the mTPM model and its implementation feasibility. A comparative analysis of ARM TrustZone solution and the proposed mTPM solution will also be carried out. Before analyzing the proposed model, a summary of the key specifications of mTPM is given below.

5.2 Summary of the Architectural Specifications of mTPM:

The following are the listed summarized specifications of hardware rooted proposed mobile security model mTPM;

- A single core of the multi-core processor will be programmed for secure processes and normal processing will run on rest of the cores.
- The core will be programmed similar to a crypto processor also providing the programmability of light weight crypto algorithms in suitable modes of operation (counter mode or GCM).
- The core will consist of an on-chip separate memory (ROM and RAM) only accessible to and via the secure core.

- For backward compatibility of the hardware solution, eMMC module deployed in the mobile devices can be used as a separate and dedicated memory till a single SoC with on chip memory and other hardware components is manufactured and marketed.
- A dedicated section from on-chip shared memory (RAM, ROM and OTP) will also be allocated to the secure core for data communication between secure and non-secure cores.
- A dedicated section from off-chip memory will also be allocated to the secure core.
- A Sclk will be fabricated on chip and dedicated to the secure core to ensure data loss prevention from the secure core.
- OTP field will be used to store the keys of the 4 level hierarchies i.e. EH, SH, PH and NH. These keys will be used for authentication and other cryptographic functions on their hierarchy levels.
- Monitor mode functionality and internal OS commands will be modified and programmed according to the proposed mTPM model which will be elaborated in the coming section of this chapter.

5.3 mTPM -Proposed Model Implementation Feasibility

In the last chapter a security model mTPM was proposed. The extent to which it is feasible to implement on hardware will be discussed in this section.

5.3.1 Implementation on Multi-core Processors:

The security model proposed is applicable only on multi-core processor architecture as we are aiming to dedicate a core for secure processing. More the number of cores in a mobile devices the more feasible it will be to implement the proposed model in the device. In January 2011 LG took the initiative to market its mobile phone with a dual core processor named LG Optimus 2X. Since then the market of mobile phones changed it approach of research and vendors started developing phones with multi core characteristics. Multi-core processing not only increased the performance criteria of computing but also made a great difference in power consumption issues. Processing with more and more cores have now become a trend and a mobile device characteristic.

Figure 5.1 shows the market share of different multi-core processors in quarter-3 of 2016 which clearly demonstrates that 97% of the market inherits 4 or more cores in the mobile phones. As majority number of mobile phones exhibit more than 4 cores and almost more than 60% possesses 8 cores. As the model specifies to isolate a core foe secure processing then the number of cores available for normal processing will decrease. More the number of cores available in a mobile device less will be the effect on the performance of the system. As most of the market possesses more than 8 cores hence the security model can be implemented on most of the mobile devices.

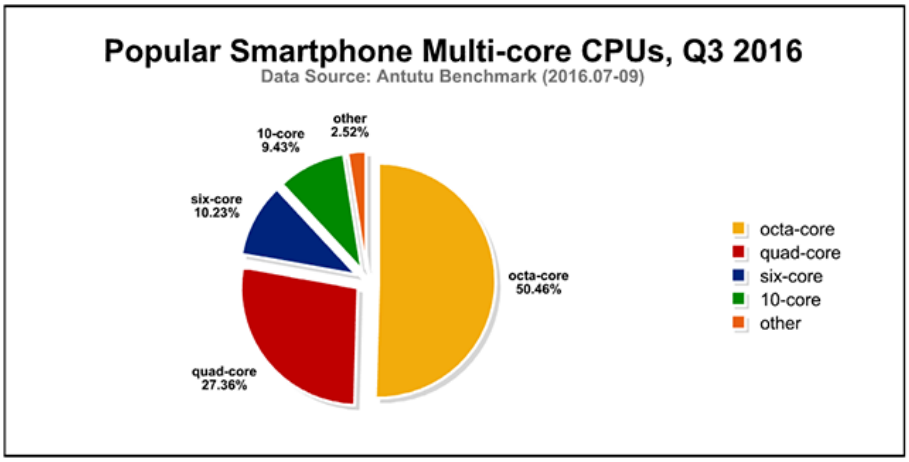

Figure 5.1: Market Share of Multi-core Processors

5.3.2 Dedicating Secure Functions to a Single Core:

Is it possible to dedicate a single core for specialized tasks? The answer to this question is, YES it is possible to separate the cores of the multi-core processors while operating. This has become possible due to heterogeneous multiprocessing technology developed by MediaTek in 2013 for programming the cores and data processing. In its first true octa core processor every core could be programmed independently and used simultaneously with flexible utilization. Today most of the multi-core processors of the mobile devices operate on heterogeneous multiprocessing technology. The OS which runs on the firmware is SE Linux. One way to assign the secure processing tasks to the first core is by using taskset tool. The following steps have to be taken;

- "Taskset" is an inbuilt tool of util-linux package. If it is not present in the linux package install the tool.
- In order to reserve the CPU core '0' for secure processing and disallow any other process or program to run on this core the following command should be added in the kernel boot-loader during boot or GRUB configuration file. **"Isolcpus=<0>".** This command means isolate cpu core number for any processes. Now core zero is reserved and no processes will run on the core except specified.
- To assign a specific task to a specific core the following command are used;

$ taskset –p <CORE-MASK> <PID>

Or **$ taskset –cp <CORE-LIST> <PID>**

Here pid refers to the program id. For example, in order to assign a process with id 9030 to core 0 the command will be

$ taskset –p 0x01 9030

Or **$ taskset –cp 0 9030**

The lowest bit in a hexadecimal core bitmask corresponds to core ID 0, the second lowest bit from the right to core ID 1, the third lowest bit to core ID 2, etc. So for example, a "0x11" represents CPU core 0 and 4. Now only process 9030 will run on core 0.

Hence using taskset all the secure functions will be assigned to the secure core and all other processes will keep on running on the other cores. As a proof of concept, this idea has already been implemented by LG for high quality audio operations. LG launched its G-Series mobile phones in 2009 having the characteristic of high fidelity sound system. It embedded this characteristic into this series of its mobile phones by dedicating a single core of the snapdragon series for high fidelity sound system operations designed to produce high quality audio. The audio quality and performance of this series is comparable to home theater systems and is used and renowned worldwide.

5.3.3 Dedicating a Secure Memory to Secure Functions:

The emerging standard for easily binding processes to processors on Linux-based supercomputers is “numactl”. It can operate on a coarser-grained basis (i.e., CPU sockets rather than individual CPU cores) than taskset (only CPU cores) because it is aware of the processor topology and how the CPU cores map to CPU sockets. Using numactl is typically easier–after all, the common goal is to confine a process to a NUMA pool (or “cpu node”) rather than specific CPU cores. To that end, numactl also lets you bind a processor’s memory locality to prevent processes from having to jump across NUMA pools or memory nodes.

If we want to bind a specific process of simulation to one processor socket with taskset without knowing its PID then the following command will be used

$ taskset –cp 0 simulation.x

The same operation can be carried out using numactl as follows

$ numactl --cpunodebind=0 simulation.x

Now if we want to restrict the “simulation.x” memory use to the NUMA pool associated with cpu node ‘0’ then the following command will be used;

$ numactl --cpunodebind=0 --membind=0 simulation.x

numactl also lets you supply specific cores (like taskset) with the“–physcpubind or –C”. An alternative syntax to numactl -C will be

$ numactl --C 0 -m 0 simulation.x

By using the above set of commands we can dedicate processes to the dedicated memory locations. Therefore it is practically feasible to isolate a core of the multi-core processor for secure processing and dedicate specific tasks, processes and memory to the required core.

5.3.4 Percentage Usage of a Core in a Multi-core Processor Architecture:

From the above section it has been concluded that it is possible to isolate a core and dedicate a memory to that core for secure functions. But to get an idea about the performance degradation of the CPU after implementing mTPM, a set of experiments

were carried out. These experiments will reveal the percentage usage time of each core in present mobile devices during execution of different processes. For this purpose Samsung Note 5 was used. Its core processor is Exynos 720 which is an octa-core processor. To run the tests on the CPU an open source tool named Workload Automation tool was used. It is developed by ARM to run the tests on CPU of Android and Linux devices. The software supports linux kernel internal tracer known as ftrace. The following experiments were carried out.

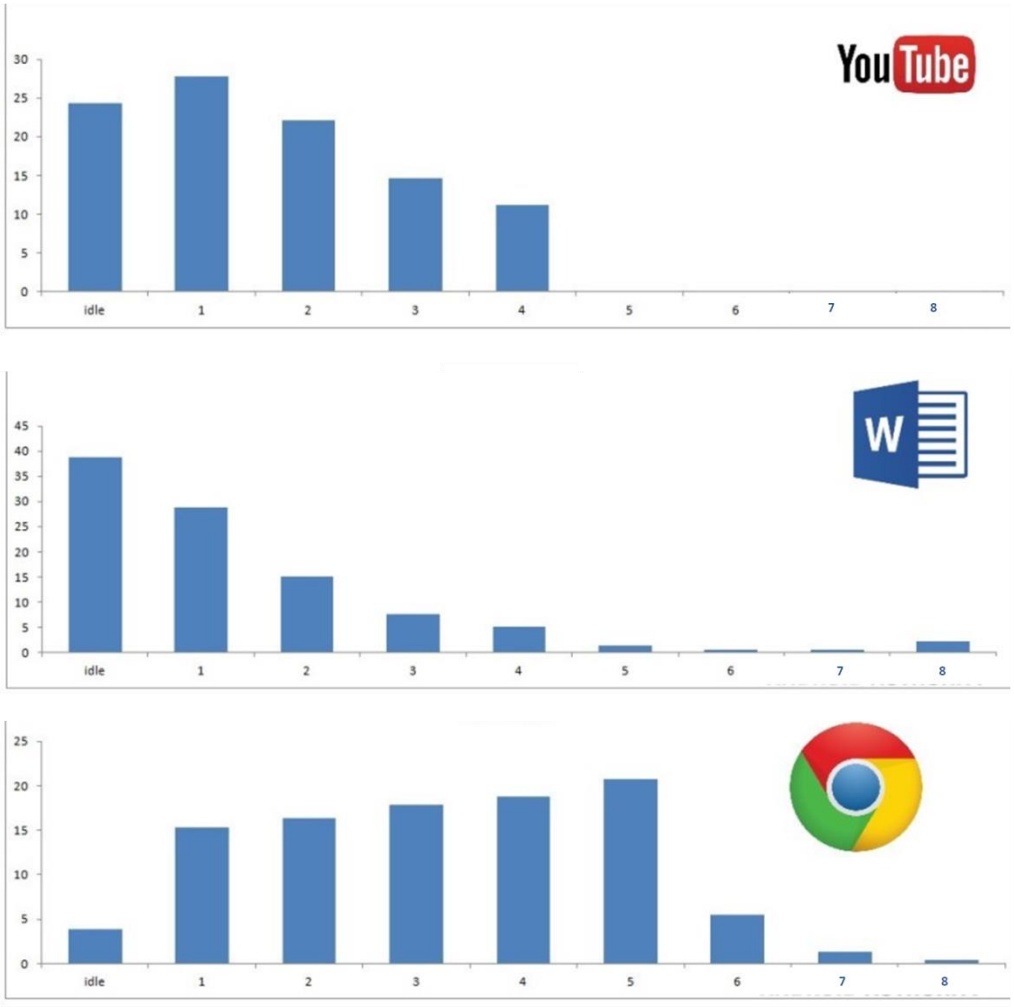

Figure 5.2: Percentage of time the number of cores is being used in Processing
(Top:You Tube Streaming, Middle: Word Documentation, Bottom: Web Browsing on chrome)

Figure 5.2 shows three graphs. The graphs represent percentage time all the cores are used in 90sec. The lower most graph depicts the percentage usage time of cores while web

browsing on a facebook site using chrome browser. For less than 4% of the time the whole CPU is idle, for 15% of the time 1 core is being used and so on. What is interesting is that for over 20% of the time 5 cores are being used in parallel. Also around 1% of the time all 8 cores are being used.

The central graph shows the graph while working on a MS Word document. The graphs clearly depict 45% of the all the cores are idle and less than 5% of the time all eight cores are being used for processing.

The top most graph depicts the percentage of time the cores are used in data capturing while streaming a 720p video on YouTube over Wi-Fi. In parallelization only 4 cores are being at most and almost 25% of the time all 8 cores are idle.

It can be concluded from experiments carried out that not all the cores of an octa-core processor are being used at a time and mostly the cores are idle. As shown from the experiments all eight cores are used only 5% of the time and if the processes of eighth core are shifted to other seven cores, it will not make a greater performance difference on the CPU usage. More the number of the cores less will be the performance degradation. Hence it is possible to implement the mTPM model as it does not cause major effect on the performance of the whole processor.

5.4 mTPM Compliance with Standards

The proposed model complies with all the security components and capabilities described in the standard of NIST and MTM as well as with the modified standard. This is illustrated below;

- **ROTS:** It has been proved through the exploits (mentioned in the previous chapter) that ARM TrustZone lacks ROTS and is unable to protect the secure world data from normal world access. In the proposed mTPM model a dedicated memory (ROM and RAM) should be embedded with the processor accessible only to the secure core. Or as an alternative eMMC module will be used as a secure storage area of the secure processor. This dedicated memory will provide a secure repository for the cryptographic keys and other security parameters and will fulfill the requirement

of ROTS. Moreover in TPM v2.0 dedicated memory is the primary component of the TPM which will be satisfied now in the proposed model.

- **ROTV:** The verification algorithms suggested used in digital signatures will be lightweight algorithms and will run in the secure core. The dedicated memory will be used to process and store data and no red data will be transmitted out of the core. The keys and other certificates required for verification will be fetched from ROTS embedded in the dedicated secure memory.
- **ROTI:** The isolated and temper proof locations required to store and processes measurements and assertions will be provided by the secure core and its private memory. No measurements or assertion record will be available outside the core. And as the processor SoC is considered to be temper proof hence they will fully comply with the NIST standard for ROTI component implementation.
- **ROTR:** The integrity of the results and reports and non repudiation will be ensured using the device key in public key algorithms embedded in hardware in the secure dedicated memory. It will send the data after cryptographically binding it with the certificate.
- **ROTM:** All the cryptographic measurements will take place within the secure core, attested by ROTR and protected via ROTI. It will have the ability of reliable integrity measurements and establishes a ROT chain of transitive measurement components.
- **API and PEE:** The API and PEE will function in the propose model similarly as they were applied in TrustZone. The TZAPI and driver will be unaltered and used by the OS and applications as previously. This will make all the versions of the proposed model backward compatible to the higher layers irrelevant of the device OS and apps.
- **Secure Boot:** The secure boot will also be enabled in the proposed mTPM model as previously. But now the boot code will be stored in the secure ROM and the measurements will take place in the secure memory. Once the secure OS boots successfully it will boot the device rich OS and then the applications after due verification from respective key hierarchy. This is shown in the figure 5.3

- **Multiple Hierarchies:** ARM TrustZone provides a single hierarchy of storage architecture. This means a single device key burned in the OTP will be used by the manufacture, OS and application developers for integrity measurements. Although mobile phone manufacturers have used their own security solutions to provide keys for each level but those keys are stored in the normal storage locations and not authenticated by their respective higher level keys. mTPM suggests to deploy a four hierarchy key system for the mobile device environment. The device, manufacturers, OS and application hierarchies will use their own hierarchy key but generated and authenticated by its higher level hierarchy respectively. mTPM complies with the TPM v2.0 for multiple hierarchy system but TrustZone does not complies as it is based on MTM model which standardizes single hierarchy system.

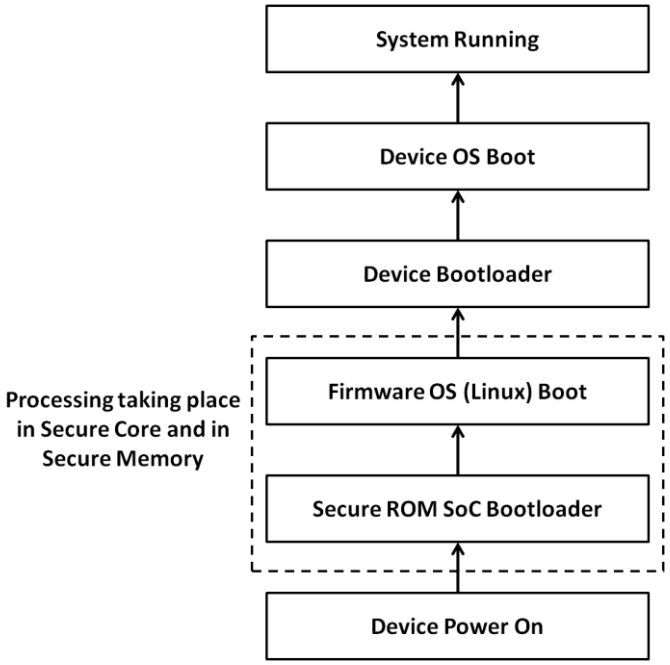

Figure 5.3: Secure Boot in the Proposed Model

The table 5.1 illustrates the comparative analysis of ARM TrustZone security solution and the proposed model mTPM. Different features have been compared and highlighted in the table.

Table 5.1: Comparative Analysis of Features Between ARM TrustZone and Proposed mTPM

FEATURES	ARM TrustZone	Proposed mTPM
Solution Type	Integrated TPM	Integrated TPM
Implementation technique	Each core virtually divided into secure mode and Normal mode timely sliced	Single core dedicated for secure mode and other cores always work in normal mode
Symmetric Algorithm	DES, DES 3	AES, SIMON, SPECK
Hashing Algorithm	SHA-1, MD5	SHA256, QUARK, SPONGENT
Digital Signature	RSA	RSA
Number of cores required for implementation	Any	Minimum 4
Number of Hierarchies	One (Device)	Four (Device, Manufacturer, OS, NULL)
Trusted Execution	✖	✔
ROTS	✖	✔
ROTI	✖	✔
ROTM	✖	✔
ROTV	✖	✔
ROTR	✖	✔
API	✔	✔
PEE	✔	✔
Secure Boot	✔	✔
Secure Entropy	✖	✔
Secure Clock	✖	✔
Secure Debug	✔	✔
Temper Detection	✖	✔

5.5 Advantages of the Proposed mTPM Security Model:

The proposed security model inherits the following advantages;

- Implements an integrated security implementation solution with the advantage of a dedicated secure processing entity without incorporating an extra hardware
- Has the advantage of programmable flexibility as the core separation and functionality is handled in software.
- Exhibits a dedicated secure memory accessible only to the secure core for TPM functionality which will overcome the secure storage limitation of ARM TrustZone and will make the security implementation standardized.
- Exhibit high performance capability as the cores will be available all the time for processing (in contrary to TrustZone) and minimize the idle percentage of time during overall computing of the device.
- Will utilize less power while computing cryptographic algorithms required for encryption, decryption, hashing and signature verification as light weight algorithms will be used for processing.
- The cryptographic computing will take less time and will be less prone to errors as counter mode or GCM mode will be used possessing the capability of parallel computing.
- Secure Entropy and Secure Clock will increase the security of the system and will standardize the solution.
- Despite providing high security assurances and properties comparable to dedicated TPM, no higher level API modifications are required. This will make the newer versions of hardware chipsets (embedded with the proposed solution implementation) compatible to most of the available OS and applications.
- A unified security platform will be available to the all mobile manufactures with the open source embedded security software to develop their secure mobile devices.

Hence the proposed solution will bring all the mobile manufacturers on a single security platform (same as in static computing devices) providing a standardized, open

sources solution to them which is backward compatible to all the versions of operating systems and applications.

5.6 Conclusion

In this chapter we have discussed and analyzed the proposed mTPM model and concluded that it is practically feasible to implement the proposed model. Moreover the mTPM model fully complies with the available standards (NIST, MTM, TPM v2.0) and overcomes the shortcomings of ARM TrustZone technology. An effort has been done to comprehensively cover conceptual framework over existing standards and their corresponding implementation methodology. However, it is felt that the whole concept should subjected to physical testing and evaluation through fabrication of model SoC and development of its related monitor mode kernel software both for its security and performance analysis. Nevertheless, the contents of the chapter appear to fulfill the objective of presenting a security wise upgraded ARM TrustZone model with adequate justification of practical implementation along with theoretical compliance related standards.

CONCLUSION

The use of smart phones, tablets and wearable devices is on the rise at fast pace in enterprise, government and military The motivation of this research was based on the fact that the current software based security solutions in mobile devices are unable to provide adequate assurance to the users' data and applications Higher level of security for such applications can only be ensured through hardware mechanisms. This research has focus on securing the mobile device through hardware rooted security and adaptation into such low power and resource constraint devices

In **Chapter 2** we highlighted various hardware rooted mobile security standards available for the mobile manufacturers to develop secure mobile systems. Their relative comparison was also been carried out and it was concluded that NIST lists down the components and capabilities required in developing a secure mobile system whereas TPM Mobile also provides the TEE architecture to use these hardware rooted components efficiently on the upper layers of the mobile device architecture. Analysis of all these standards and their comparison was carried out. Moreover some of the shortcomings analyzed in MTM standard were also highlighted. At the end of the chapter concept of lightweight cryptography was introduced and different lightweight block ciphers and hashing algorithms are listed. There relative comparison with respect to throughput, power consumption and memory (RAM and ROM) was carried out. The best ciphers which fulfilled the tradeoff between greater throughput, less power consumption and minimum memory utilization were SIMON and SPECK among lightweight symmetric ciphers and QUARK among lightweight hash functions whereas SPONGENT and PHOTON are also a good option for implementation as they provide greater range of message digest with good characteristics. These results were used in the 4th chapter while proposing out mTPM security model.

In **Chapter 3** we had discussed the internal key components of a TPM and their implementation techniques with the TCG specifications related to them. It was concluded that integrated TPM techniques provides a better solution fulfilling the tradeoff between cost, size, power consumption and performance metrics of resource constraint devices. Different mobile security solutions made by the industry were discussed which included

ARM TrustZone, Qualcomm, Samsung, MediaTek, Intel, Apple and Boeing. Most of the vendor solutions were based on the security technology of ARM TrustZone with their own closed form security model implemented on its higher layers. Hence all the solutions were vendor specific. As ARM TrustZone was the widely deployed security architecture of most of the industrial mobile security solutions it was discussed and analyzed deeply. After its analysis many major shortcomings in the solution were highlighted which not only made the solution vulnerable to attacks but also made the solution non-standardize. Therefore all the solutions based on this technology were non standardized with respect to hardware rooted security. A list of exploits related to the TrustZone vulnerabilities was also presented. Hence it was concluded that most of the available market mobile security solutions are non-standardized, ad-hoc, vendor specific and closed form solutions. Hence a suitable solution was required to overcome the limitations in standards and ARM TrustZone.

In **Chapter 4** a new security model for mobile device namely mTPM was presented. Its salient features include dedicating a core for secure processing, dedicating a secure memory accessible only to and via the secure core, usage of lightweight cryptographic primitives for secure computations, provision of secure clock to the secure core, provision of secure entropy by the RNG, and incorporating a four level hierarchy system. As it has been quite elaborated, mTPM provides all the security provisions of TPM 2.0 for mobile devices including but not limited to resource constraint wearable, IoT sensors and actuators

In **Chapter 5** we had discussed and analyzed the proposed mTPM model and concluded that it is practically feasible to implement the proposed model. Moreover the mTPM model fully complies with the available standards (NIST, MTM, TPM v2.0) and overcomes the shortcomings of ARM TrustZone technology. An effort has been done to comprehensively cover conceptual framework over existing standards and their corresponding implementation methodology. The contents of the chapter appear to fulfill the objective of presenting a security wise upgraded ARM TrustZone model with adequate justification of practical implementation along with theoretical compliance related standards.

It is hoped that the proposed mTPM model will provide a unified, vendor neutral and standardized security platform for the mobile device manufacturers and will contribute towards a secure mobility environment.

FUTURE WORK

The three objectives of the research work to be carried out were stated at the start of the documentation which included identifying and understanding the hardware rooted security standards individually and their comparative analysis, analysis of the industrial security solutions of mobile devices, and proposal of a new hardware security solution for mobile devices. All the objectives and aims have been fulfilled during our research work and documented. Based on current work, following tasks can be taken up for future research work.

1. This thesis has laid the foundation by adequately pointing out shortcoming of the existing standards and industrial practices for their implementation in diverse customized form. It is believed that now a new set of standards can be developed exclusively for mobile devices that are compliant to well established TPM 2.0 on one side and physical limitations of mobile device SoCs on the other side, and thereby, bringing the OS and Applications development on the uniform and standard form.
2. The implementation strategy of the proposed mTPM has been developed around well practiced ARM TrustZone. The feasibility of the mTPM has been justified in segments at various layers as reference hardware platform was unavailable at the bottom. Various aspects have been picked up from best industrial practices from hardware for static platforms, underlying Linux operating system, boot-loader implementations and guidelines for other mobile platforms. However, it is felt that the proposed mTPM model should subjected to physical testing and evaluation through fabrication of model SoC and development of its related monitor mode kernel software both for its security and performance analysis in a holistic manner.
3. The current research work primarily focuses on Android based mobile devices and that too based on ARM SoCs only. Similar level of research is also required for the analysis other mobile devices platform, such as Apple, blackberry, etc. who manufacture their devices right from the hardware of the device with their customized OS and enterprise certified applications with the objective to arrive at a uniform security standard for entire family of mobile devices.

4. So far, the scope of mobile devices has been restricted to smartphones, tablets and to some extent wearable devices. However, the advent of IoT has extended the scope of mobile device definition to an array of sensors, actuations, smart cards, smart tokens and RFID/Bluetooth/NFC devices that are severely resource constraint but operating around diverse nature of processing and communication protocols (including Wi-Fi, ZigBee, Bluetooth, MQTT and XMPP etc). Therefore they all gave varying nature of security service implementation; most of them are quite non-standardized. It is, therefore, required that processing and communication capabilities all such devices may be accounted for while developing and standardized security mechanism is developed for IoT. Therefore, a separate research work may be undertaken on compilation of the requirements and resources of contemporary IoT devices in an integrated form.
5. While this thesis has given a proposed solution in the form of mTPM, certain recommendations have been given to improve the security of current ARM TrustZone technology in the existing hardware using certain technical actions at firmware level including Linux kernel configuration, boot-loader and use of various peripherals. Using these techniques, the security of current devices may be improved to near mTPM without any hardware modification. An independent project on this subject may be undertaken to implement a reference design to be followed by vendors such to provide enhanced level of security to the end user applications while staying within the scope of existing hardware.

REFERENCES

[1] Mooseop Kim, Hongil Ju, Youngsae Kim, Jiman Park and Youngsoo Park, "Design and Implementation of Mobile Trusted Module for Trusted Mobile Computing", ieee transaction on consumer electronics, vol. 56, No. 1, February 2010

[2] Johann Großschadl, "Reassessing the TCG Specifications for Trusted Computing in Mobile and Embedded Systems",", IEEE transaction on consumer electronics, vol. 56, No. 1, February 2010

[3] NIST SP 800-164, Guidelines on Hardware Rooted Security in Mobile Devices (draft), 2012

[4] NIST SP 800-111, Guide to Storage Encryption Technologies for End User Devices [SP800-111].

[5] NIST SP 800-124 Revision 1, Guidelines for Managing and Securing Mobile Devices in the Enterprise [SP800-124]

[6] NIST IR 8114, "Report on Lightweight Cryptography", March 2017

[7] R. Beaulieu, D. Shors, J. Smith, "Simon and Speck: Block Ciphers for the Internet of Things", NSA document, 9 July 2015

[8] R. Beaulieu, D. Shors, J. Smith, "The Simon and Speck Block Ciphers on AVR 8-bit Microcontrollers", NSA, 2014

[9] Mickael Cazorla, Kevin Marquet and Marine Minier, "Survey and Benchmark of Lightweight Block Ciphers for Wireless Sensor Networks", **ieee** international conference SECRYPT, 2013

[10] Lara-Nino, Carlos Andres, Morales-Sandoval Miguel, "An evaluation of AES and present ciphers for lightweight cryptography on smartphones", ieee conference on Electronics, Communications and Computers (CONIELECOMP), 2016

[11] Jaber Hossein Zadeh , Abbas Ghaemi Bafghi, "Evaluation of Lightweight Block Ciphers in Hardware Implementation: A Comprehensive Survey" , 2016

[12] Soren Rinne, Thomas Eisenbarth, and Christof Paar, "Performance Analysis of Contemporary Light-Weight Block Ciphers on 8-bit Microcontrollers", 2011

[13] www.cryptolux.org/index.php/Lightweight_Block_Ciphers, last visited 30 November, 2017

[14] C. Manifavas, George Hatzivasilis, K. Fysarakis, and K. Rantos, “Lightweight Cryptography for Embedded Systems - A Comparative Analysis”, 21 march 2014

[15] Baraa Tareq Hammad, Norziana Jamil, Mohd Ezanee Rusli and Muhammad Reza Z`aba, “A survey of Lightweight Cryptographic Hash Function”, International Journal of Scientific & Engineering Research Volume 8, Issue 7, July-2017

[16] www.cryptolux.org/index.php/Lightweight_Hash_Functions, last visited 30 November 2017

[17] Microsoft documentation, “fTPM: A Software-only Implementation of a TPM Chip”

[18] Andreas Fitzek master’s thesis, “Development of an ARM TrustZone aware operating system ANDIX OS”, Graz University of Technology, April 2014

[19] ARM Security Technology whitepaper, “Building a Secure System using TrustZone Technology”, 2009

[20] ARM Security Technology, “Smc Calling Convention- System Software on ARM Platforms”, 2016

[21] ARM Security Technology, “TrustZone API Specification version 3.0”, 20 february 2009

[22] N. Asokan , “Moile Trusted Comuting”, ieee Vol. 102, No. 8, August 2014

[23] Trusted Computing Group. https://www.trustedcomputinggroup.org

[24] Trusted Computing Group white paper, “TPM mobile with trusted execution environment for comprehensive mobile device security”, June 2012

[25] TCG mobile reference architecture specification version 1.0, https://www.trustedcomputinggroup.org/specs/mobilephone/tcg-mobilereference-architecture-1.0.pdf, June 2007

[26] Roger L. Kay, “The Future of Trusted Computing,” GovSec 2005

[27] TCG documentation, “Mobile Trusted Module 2.0 Use Cases Specification”, Version 1.0, 4th March, 2011

[28] Kathleen N. McGill, “Trusted Mobile Devices: Requirements for a Mobile Trusted Platform Module”, Johns Hopkins Apl Technical Digest, Volume 32,

Number 2, 2013

[29] Konstantinos Markantonakis, Keith Mayes, “Secure Smart Embedded Devices, Platforms and Applications”, chapter 4, pg 71-94, springer 2014

[30] T. R. Halfhill. ARM dons armor: Trust-Zone security extensions strengthen ARMv6 architecture. Microprocessor Report, 17(34):20–23, Aug. 2003.

[31] T. R. Halfhill. IBM offers chip-level security: Secure Blue technology aims to make security ubiquitous in SoCs. Microprocessor Report, 20(19):1–4, May 2006.

[32] Frost and Sullivan, “Hardening Android: Building Security in the Core of Mobile Devices”, May 2014

[33] Ahmed Sallam, “The new era of mega trends: hardware rooted security”, January 2015

[34] Jesus Molina, Houcheng Lee, Sung Lee, Zhexuan Song, “A Mobile Trusted Platform Module (mTPM) Architecture”, 2012

[35] Samsung Knox whitepaper, “ Samsung Knox Security Solution”, June 2015

[36] Samsung Knox whitepaper, “An Overview of the Samsung Knox Platform”, June 2015

[37] Samsung KNOX white paper:, “Mobile Malware and Enterprise Solution”, May 2015

[38] MediaTek documentation, “MT2502A SOC Processor Technical Brief”, 8 September 2014

[39] Ross Anderson, Mike Bond, “Cryptographic Processors-A Survey”, IEEE proceedings, 2010, page 100-115

[40] Lubos Gaspar, “Crypto-processor-architecture, programming and evaluation of the security”, April 2014

[41] Apple whitepaper, “iOS Security - whitepaper”, March 2017