Telecommunication Systems (2020) 74:379–403
https://doi.org/10.1007/s11235-020-00656-y

Analytical study of hardware-rooted security standards and their implementation techniques in mobile

Naveeda Ashraf[1] · Ashraf Masood[1] · Haider Abbas[1] · Rabia Latif[2] · Narmeen Shafqat[1]

Published online: 25 March 2020

Abstract
Security of information in computers is of paramount importance. Considering the software security as inadequate, hardware rooted security standards were developed as Trusted Platform Module (TPM) 1.0 in 2003 and subsequently as TPM 2.0 in 2012. While trustworthy, these standards and their corresponding implementation in hardware as TPM chips were found to be inappropriate for mobile computing devices due to their small form factor, low computing resources, limited battery power and cost. Given these challenges, software derivative of TPM was devised for mobile devices as TPM Mobile. However, TPM Mobile was rarely implemented in real devices primarily due to lack of trust in its software nature. Another standard named as MTM also emerged as derivative of TPM but was never adopted widely due to physical limitations of the mobile devices that have been further constrained after introduction of Internet of Things. Subsequently, a software-cum-hardware combo implementation appeared in ARM-based mobile CPUs by the name of TrustZone as a trade-off between hardware and software. Although widely adopted ARM TrustZone has also been considered as inadequate vis-a-vis TPM standards. After conducting a comparative analysis of various security standards, this paper proposes mTPM, a comprehensive security standard. As such mTPM not only addresses prevalent information security requirements of mobile devices but also considers their physical constraints. mTPM primarily suggests an implementation of a security processor integrated within existing CPU, as stand-alone chip was considered infeasible for mobile devices. The detailed architectural model of mTPM has also been included as guidelines for uniformly secure implementation and standardization. In view of its advantages, mTPM is expected to find greater adoption and refinements over time.

Keywords ARM TrustZone · MTM · NIST · Roots of trust · TPM · Hardware-Rooted security

1 Introduction

Since the birth of mankind, efforts have been made to invent devices that 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 with the persistent miniaturization of computing resources and improvement in portable battery life, portable computers grew into popularity in last decade of twentieth century. The need of portable computing encouraged the manufacturers to integrate the computing resources into cellular phones and now in different wearable and IoT devices. 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 first step to standardize the security implementation and gave the specifications for a Trusted Platform Module (TPM). In 2006, the computers started to introduce embedded TPM chips

✉ Haider Abbas
haider@mcs.edu.pk

Naveeda Ashraf
naveedaashraf.msis14@students.mcs.edu.pk

Ashraf Masood
ashraf@mcs.edu.pk

Rabia Latif
rlatif@psu.edu.sa

Narmeen Shafqat
narmeen_shafqat@mcs.edu.pk

[1] National University of Sciences and Technology, Islamabad, Pakistan

[2] College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia

and built-in security. Since then the standards as well as computing devices were modified with enhanced security assurances to provide 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 access corporate data, resources and services to perform different activities. But unfortunately, mostly these mobile devices especially personally-owned devices are unable to provide strong security assurances to the organizations and end-users. Besides, 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 [1]. Enterprises have to accept these security risks present in the mobile devices because of several factors which include cost savings and employee's 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 (OS) and be further assisted by the system hardware.

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 raised many complications in which cost, size and power consumption constraints are the main concerns. Most of the contemporary solutions available in the market use virtualization technique to overcome the security concerns and mostly rely on ARM TrustZone technology. Qualcomm, Samsung 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. Therefore, the solutions available are ad-hoc, vendor specific and closed-form solutions. 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.

The first section of the paper discusses the National Institute of Sciences and Technology's (NIST) requirements and its architecture for hardware-rooted security in mobile devices. The second section highlights the specifications for the trusted mobile platform developed by Trusted Computing Group (TCG). This provides the baseline for the implementation of the functional security components and parameters for the hardware-rooted security in mobile embedded systems.

Section 3 focuses on the possible techniques to implement Trusted Platform Module (TPM) specifications duly modified to alleviate its shortcomings in Sect. 4. It describes various implementation techniques along with suggestions for optimum implementation methodology in Sect. 5. In Sect. 6 the contemporary solutions available in the market have also been highlighted with their implementation shortcoming in Sect. 7. Section 8 analyzes various light-weight cryptographic protocols which could be implemented in low computational power devices. Section 9 describes the ARM architecture in detail while highlighting the shortcoming in the implementation of TrustZone in Sect. 10.

In Sect. 11, a new hardware-rooted security solution has been proposed named mobile TPM (mTPM). It consists of two parts. First is the suggested modification of the limitations of the existing mobile standards. Second is the suggested security model which is based on the modifications in the shortcomings of Advanced RISC Machines (ARM) TrustZone security model. Some more enhancements to implement the modifications are also the part of this section.

Sections 12 and 13 discuss the implementation mechanism and feasibility of the proposed hardware-rooted security model mTPM. Moreover, it also analyzes the model for the compliance with the existing and modified standard and presents a comparative analysis of the security features inherited in ARM TrustZone and the proposed mTPM. Section 14 concludes the paper while focusing on the future work related to the research conducted on the topic.

2 NIST hardware-rooted security architecture

The previous discussion has highlighted the importance of the requirement of hardware security in mobile devices. However, many mobile devices are deficient in built-in secure hardware roots of trust. In 2012, NIST published Special Publication 800-164 and took the step to standardize the basic requirements to harden the core of mobile devices. According to it, the following steps should be taken by the IT industry to enable hardware-rooted security in the mobile devices [2].

2.1 Implement security capabilities

All mobile devices should provide subsequent 'three data security capabilities' to meet the standards of NIST;

(i) Device Integrity: Device integrity refers to the nonexistence of corruption in the hardware, firmware or software of a device. A mobile device provides an evidence of secure execution and device integrity if its configurations are shown to be in a trusted state.
(ii) Isolation: Isolation refers to the ability of the system to keep different data components and mechanisms separate from each other and, hence, control the flow of information from one entity to another. In mobile devices, isolation is required to ensure that no application interferes with the process of another application.
(iii) Storage Protection: Storage protection refers to preservation of the confidentiality and integrity of data integrity of data at rest, in transit, and upon access revocation. Protected storage primarily depends on encryption algorithms used to authenticate credentials of authorized users for integrity and the protection of data and the associated keys.

2.2 Verify security components

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 [3]. These security components are:

(i) Roots of Trust (RoTs): RoTs provide assurance of the trustworthiness of a mobile device. RoTs are trusted to perform security-critical functions like software verification, cryptographic key protection, and device integrity and device authentication. RoTs behave in a trusted and predictable manner since their errors cannot be identified. Hardware RoTs offer immutability, smaller attack surface, and more reliable behavior as compared to the software RoTs. Beside this, software RoTs offer the advantage of fast deployment on diverse platforms. In order to provide the security capabilities, devices should implement the subsequent RoTs specified in NIST guidelines:
    - Root of Trust for Storage (RTS)
    - Root of Trust for Verification (RTV)
    - Root of Trust for Integrity (RTI)
    - Root of Trust for Reporting (RTR)
    - Root of Trust for Measurement (RTM)
(ii) Application Programming Interface (API): The APIs expose the RoTs to the platforms so that OS and applications can have high level of security assurance. Mobile OS use the features offered by the RoTs through APIs to create and secure device integrity reports, verify and measure software and firmware, and protect locally stored authentication credentials, cryptographic keys, and other sensitive data.
(iii) The Policy Enforcement Engine (PEnE): PEnE is generally the part of mobile OS. It imposes policies on the device with the help of other device components and enables the maintenance, processing and management of policies on both the Information Owner's and device environments. In the case of conflict, this engine notifies the device owner and enforces a default policy which denies the unauthorized access of data until the error is resolved.

2.3 Mitigate risks of exposure

The last step is to mitigate the risk of revealing encryption keys. RoT for storage contains two classes of keys; Data encryption keys and Key encryption keys. While RTS protects the key encryption keys, data encryption keys are at risk of exposure as these are often decrypted for use outside the RoT. Isolation of applications from one another is one of the ways to mitigate this risk of exposure [4].

3 Trusted platform module mobile specifications

The increased utilization of diverse connected devices mainly mobile phones and tablets have fundamentally transformed our lifestyles. We can now 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 needs activated emergence of two key platform security technologies; Global Platform's Trusted Execution Environment and Trusted Computing Group's Mobile Trusted Module. These two technologies work together in a unified manner called TPM MOBILE to provide security and improved services to its consumers.

3.1 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. TPM v1.2 was standardized in 2003. Some of its salient features include strong cryptographic algorithms for hashing, authentication and authorization that were prevalent at the time of standardization such as SHA-1, RNG, RSA and HMAC. Moreover,

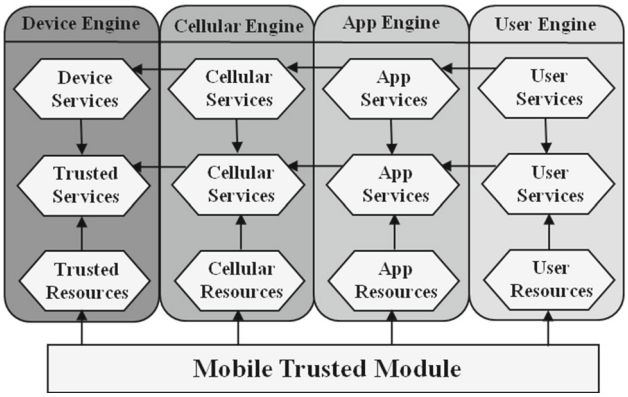

Fig. 1 Generic architecture of MTM

it had single storage hierarchy model to store data, keys and different mobile platforms [5].

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 specifications are:

(i) The idea of secure boot is initiated. This means that the boot sequence is not only calculated, but also stops when non-approved software is detected.
(ii) 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.
(iii) It supports to run several parallel MTM instances of multiple stakeholders on the same device while still fulfilling the TCG specifications.

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 Fig. 1. Ideally, in a mobile platform, a single MTM hardware should be accessed by different engines with each engine as annotion of its own trusted services. Each mobile platform engine should support:

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

In 2012, TPM 2.0 was published which addressed many of the same use cases of TPM v1.2 and provided many similar features with enhanced security capabilities to provide a high level of security assurance for desktops and laptops. TPM 2.0 is not backward compatible with TPM v1.2 and hence not with MTM as well. TPM v2.0 provides stronger cryptographic algorithms than TPM v1.2 and also discards obsolete algorithms which were supported previously. Moreover, it provides a three-level hierarchy model for platform, storage and endorsement. Table 1 shows the major differences in specifications of both the policies.

Table 1 Comparison of TPM versions [5]

Specification	TPM v1.2/MTM	TPM v2.0
Algorithms	RSA, SHA-1	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

3.2 The GlobalPlatform TEE

The GlobalPlatform TEE identifies a consistent isolation environment for System-on-Chip (SoC) in which sensitive data, code and resources are executed separately from the main OS environment. This isolation is possible due to the hardware architecture. 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 [6].

3.3 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 in the processor starts its boot security. During the later stages of the boot, applications are verified crypto-

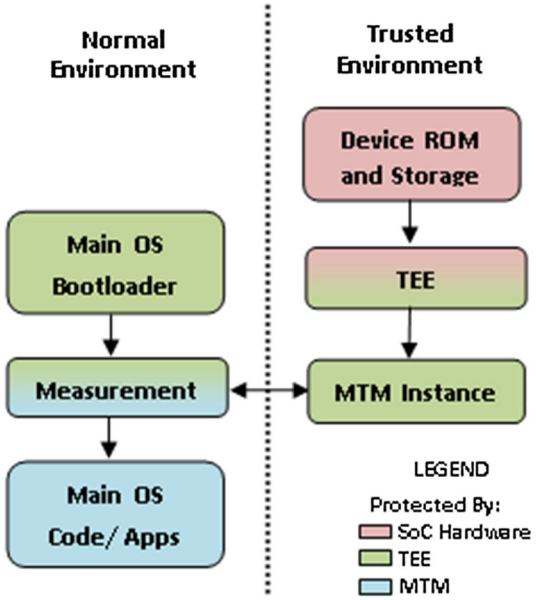

Fig. 2 TPM mobile boot process

graphically to make sure that authorized software is running on the device as shown in Fig. 2.

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. 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 [7].

The security model as illustrated above works because 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 on them, thus, ensuring trustworthy protection and portability across mobile devices.

4 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 [8]. TPM is considered to be the root of trust which enables secure attestation by providing secure cryptographic primitives for signatures and hashes. However, the similarities between MTM and TPM have raised many implementation concerns and respective challenges [9, 10]. The following are some of the salient observations related to MTM specifications:

(i) 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 has proven to be the 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 multi-stakeholder 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.

(ii) TCG enlightens the functional aspects of MTM while not focusing on the implementation technique required in developing such modules. This aspect has been left over for the manufacturers to define their strategies by themselves, which does not make one manufacturer'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.

(iii) 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.

(iv) The algorithms defined for security in MTM support cryptographic algorithms which require large computation power and resources. These are, thus, less suitable choices for low computing resource constraint processors. For example, 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.

(v) The implementation technique of 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.

(vi) 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 low 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 are discussed later where the solution implementation technique for the mobile devices has been proposed.

5 TPM implementation techniques

TPM is the basic component in the trusted computing devices which offers a hardware root of trust to ensure OS and application's 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 Fig. 3.

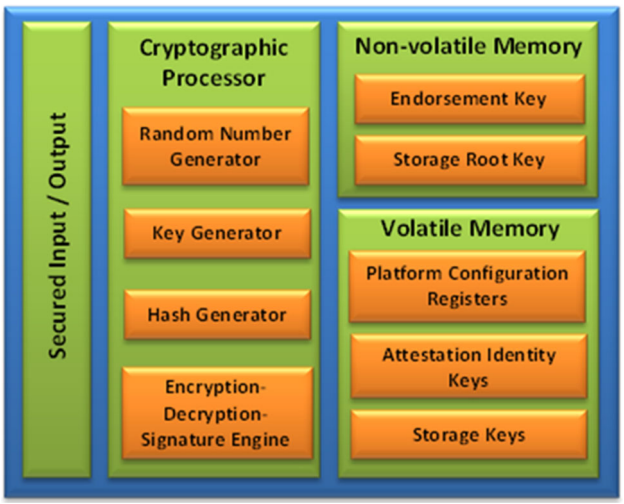

Fig. 3 Internal components of a TPM

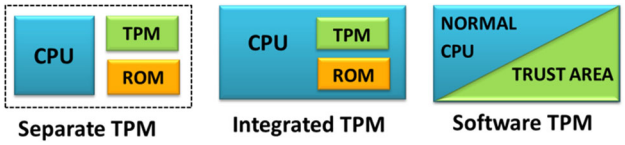

Fig. 4 Options for implementing TPM functionality

TPM functionality can be implemented in three different ways in the embedded system; as illustrated in Fig. 4. Each of these three implementation methods has their own pros and cons with respect to areas of interest such as cost, security and flexibility [11].

5.1 Separately mounted TPM

The first method is to mount a discrete TPM chip on the motherboard interconnected with the processor via a bus, used for data communication. This approach is a concrete example of compliant TCG specifications and is widely deployed in today's systems for trusted computing in desktop computers. Examples of separately mounted TPM include IBM's Secure-Blue technology or Texas Instruments' M-Shield mobile security technology. A discrete TPM chip soldered on the motherboard increases manufacturing cost, size, and weight of the 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.

5.2 Software TPM

The second method is software-TPM which executes 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. 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.

5.3 Integrated TPM

The third method is an on-chip deployment of the TPM module so as to make a single SoC act as a 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. The alternative idea is to integrate security features directly into the processor core through micro-architectural enhance-

ments. Hence, this provides the advantages of the previous method 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.

6 Contemporary implementation solutions

Some of the leading solutions provided by different manufacturers in the area of hardening mobile devices are highlighted in this section. All these solutions are implemented as an integrated TPM module implementation technique [12].

6.1 ARM TrustZone

ARM launched TrustZone in 2003. ARM TrustZone is a SoC-based approach that offers the security for a TEE running alongside the main OS. Applications referred to as Trusted Applications run on the TrustZone-protected TEE. TrustZone technology is incorporated tightly into ARM Cortex-A processors. TrustZone uses hardware-based system-wide security virtualization technique to create an isolated environment for Trusted Applications.

6.2 Qualcomm

Qualcomm, referred to as Snapdragon Security Solutions, is offering the security enabled in its Snapdragon family of processors. Snapdragon Security Solutions are based on three pillars of security: Secure MSM (composed of the primary components congruent with features enabled by ARM's Trust-Zone), Studio-access technology (provides security on the digital rights management controls of the system) and Enterprise and BYOD Security (Snapdragon-based solutions provide the APIs that mobile device management vendors can use as Qualcomm itself does not provide an end-to-end enterprise solution).

6.3 Samsung

In 2013, Samsung launched KNOX providing employees with the productivity needs of BYOD while, at the same time, protecting enterprise's data. 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 Trust-Zone technology to support embedded security. Samsung KNOX creates a separate secure area on the device for enterprise and corporate applications and data which are isolated from applications outside the container. Thus, KNOX provides a complete Mobile Device Management cloud-based solution for the BYOD scenarios [13].

6.4 MediaTek

MediaTek is known for its low-cost products and multi-core Central Processing Unit design, marketing the chipsets to other Smartphone vendors for use. MediaTek uses the same ARM CPU core designs and TrustZone technology as Qualcomm and Samsung, with its own modified closed-source TEE to communicate with the hardware. The lack of source-code prevents third-party patches for any security or hardware issues left unfixed by the company [14].

6.5 Intel

Similar to ARM, Intel has taken the approach of embedding security features into SoC. While ARM includes Trust-Zone as part of its Intellectual Processing (IP) cores that is 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 consistent security across all its processors through this embedded IP block. But this, too, is specific to only Intel processors which comprise less than 2% of the mobile market share [15].

6.6 Apple

Apple is considered to be one of the most secure solutions in the industry. It withholds a unique position in the industry when it comes to hardware/software integration as Apple designs its own chips and OS. Apple has designed security into its products from the silicon up. 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 [16].

7 Analysis of implementation solutions

According to Statista, nearly 85% of the market is captured by the Android smartphones. The chip brand distribution being used in these 85% of Android phones is shown in Fig. 5 (survey report by Antutu benchmark, Q2 of 2018). From the graph, it can be concluded that Qualcomm, Samsung and MediaTek capture almost 90% of the Android market.

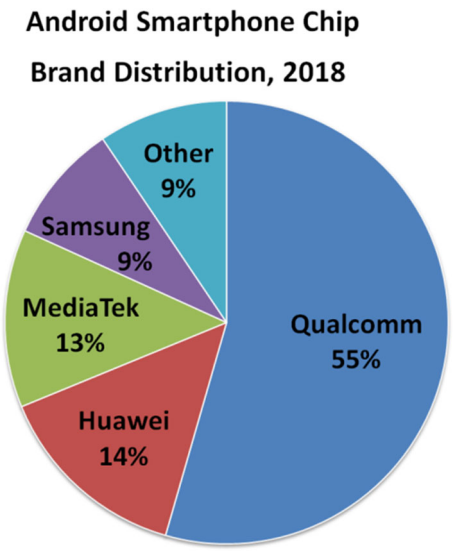

Fig. 5 Android smartphone chip brand distribution in Q2, 2018 [17]

All the contemporary solutions discussed above are implemented as integrated TPM module implementation technique using virtualization and they 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 Android 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 raised many complications, amongst which cost, size and power consumption constraints are the main concerns. As 99% of the solutions are based on ARM TrustZone technology, therefore, an integrated TPM implementation based on TrustZone architecture is expected to be a better solution. Our proposed solution also uses 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.

8 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 computing power processors meant for laptops, desktops, tablets and smartphones. On the contrary, these algorithms are too heavy for the power, processing and memory constraint environment of various wearables and IoTs which include Bluetooth, NFC, RFID and smart card systems. Therefore, lightweight ciphers were developed for such resource constraint devices providing reasonable security as conventional crypto algorithms but utilizing less power and memory due to smaller key size, smaller block size, less number of rounds and relatively simpler design architecture [18].

8.1 Lightweight block ciphers

Some of the lightweight block algorithms implemented widely and known for their high cryptographic strength and throughput are listed below in Table 2 along with their salient features [19–23].

The cipher with the maximum throughput and minimum memory usage is considered to be the better cipher. From the listed ciphers in the Table 2, the best cipher having the maximum throughput is Simon/Speck with 855 Kb/s of throughput [19]. After that, PRINCE and mCrypton seem to provide a better throughput of 533 Kb/s and 482 Kb/s respectively [23]. Figure 6 depicts their relative throughput graphically. Figures 7 and 8 illustrate the memory usage of RAM and ROM respectively. It depends on the design criterion of the developer that whether it uses more ROM space for the algorithm code and saves storage or trades RAM for higher processing. From the algorithms listed in Table 1, DESLX uses the maximum ROM space but Present utilizes maximum RAM space. Observing both the metrics simultaneously, Speck/Simon uses minimum RAM space and no RAM is utilized during processing. It stores its intermediate states and processing data in registers. Hence, Speck/Simon provides an optimal solution for maximum throughput and minimum storage utilization.

Out of the above listed light-weight ciphers, NIST has recommended DESL, SEA, TEA, SIMON and SPECK. Piccolo algorithm provides the best results for throughput and

Table 2 List of lightweight block cryptographic algorithms and their performance metrics [19–23]

S. no.	Lightweight algorithm	Key size (bit)	Block size (bit)	No. of rounds	Throughput (Kb/s @ 100 kHz)	Power consumed (μW)/bit	Memory utilized (bytes)	
							RAM	ROM
1	DESLX	184	64	16	44.4	1.6	112	16,816
2	HIGHT	128	64	32	188.2	–	18	3130
3	mCrypton	64	64	12	482.3	–	18	2726
		96					20	2834
		128					24	3108
4	Piccolo	80	64	25	237.04	4.42	79	2434
		128		31	193.9	2.78	91	2510
5	Present	80	64	31		2.78	142	4814
		128			200	3.67	142	4964
6	PRINCE	128	64	12	533.3	5.8		
7	SEA	96	96	93	103	3.218	24	2804
8	SIMECK	64	32	32	88.9	0.606	–	–
		96	48	36	120	0.875		
		128	64	44	133.3	1.162		
9	SPECK/SIMON	64	32	32	855	3.98	0	324
		72/96	48	36		3.32	0	556
		96/128	64	42/44		3.65	0	602
		128/192/25 6	128	68/69/7 2		4.20	0	1108
10	XTEA	128	64	64	57.1	19.5	11	1394

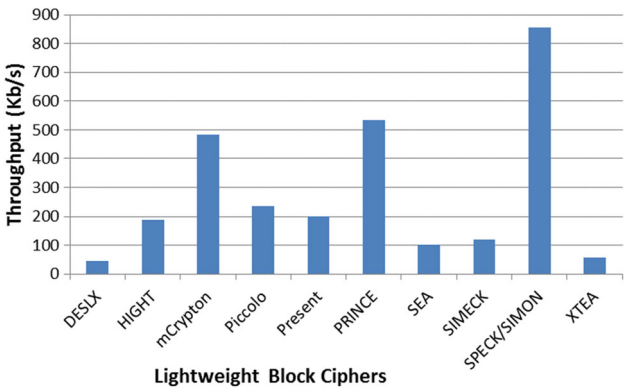

Fig. 6 Comparative analysis of throughput of popular lightweight block ciphers

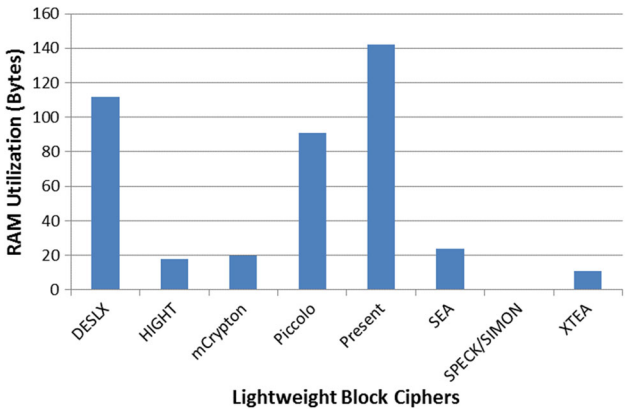

Fig. 7 Comparative analysis of RAM utilization of popular lightweight block ciphers

relative hardware size and is prioritized when implementing the algorithms in hardware. Simon and Speck are the algorithms made by NSA but have not been publically released. SPECK's hardware and SIMON's software implementation have proven to be among the best algorithms for resource constraint devices.

8.2 Lightweight hash functions

Table 3 shows the list of lightweight hash functions implemented widely along with their performance metrics [24, 29]. QUARK, PHOTON, DM-PRESENT and SPONGENT are the hashing functions standardized by NIST. As shown in Fig. 9 among the listed lightweight hashing functions QUARK seems to fulfill the tradeoff of high throughput, less power consumption and minimum memory usage. Whereas

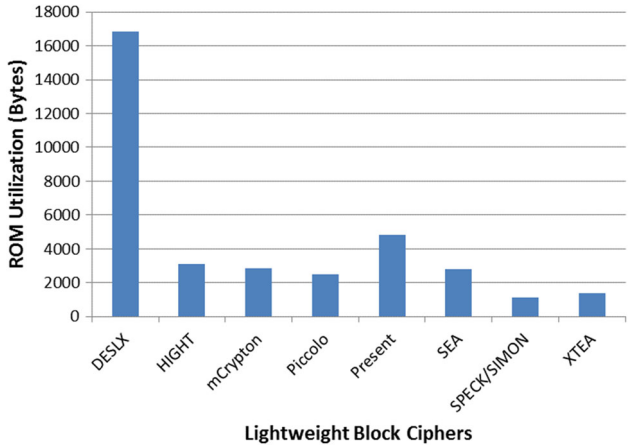

Fig. 8 Comparative analysis of ROM utilization of popular lightweight block cipher

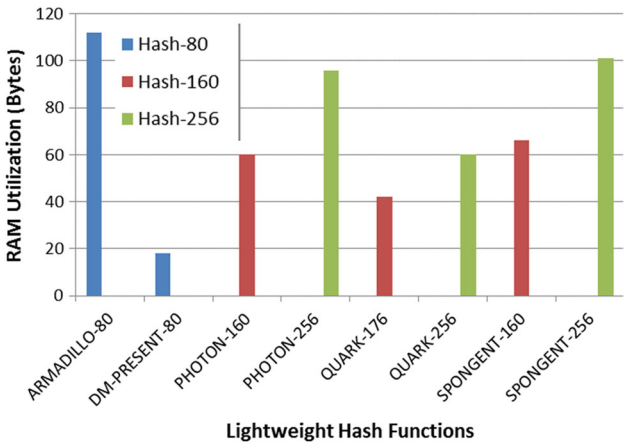

Fig. 9 Comparative analysis of RAM utilization of lightweight hash functions

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 where the suggested solution will be presented.

9 ARM TrustZone architecture

The ARM TrustZone is an integrated security architecture aiming to offer enhanced security as well as infrastruc-

Table 3 List of lightweight hashing functions and their performance metric [24–29]

S. no.	Lightweight algorithm	Digest size (bit)	Rate (bit/s)	Internal state size (bit)	Throughput (Kb/s @ 100 kHz)	Power consumed (μW)/bit	Memory utilized (bytes)	
							RAM	ROM
1	ARMADILLO	80	48	256	109	44	112	16,816
		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

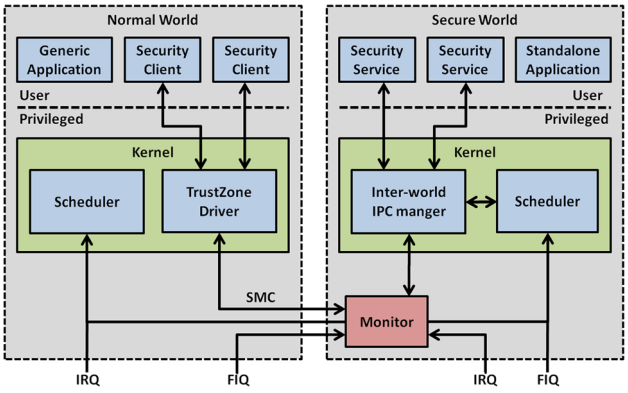

Fig. 10 Switching mechanism between two virtual modes

ture foundation for 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 [30].

9.1 Processor architecture

In ARM processors, each physical core is designed to provide two virtual cores; secure and non-secure, and a switching mechanism known as the 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 to the main system bus to access instructions or data. The non-secure world has open access to the non-secure system resources but is restricted to access the secure services. On the other hand, secure virtual processor has open access to all the resources as illustrated in Fig. 10.

The two virtual modes of a single physical processor context switch between the two worlds in a time-sliced fashion through monitor mode. The monitor mode is the part of the secure world. 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 [31].

The monitor mode software is designed and implemented by the SoC developers. Its main functions are 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

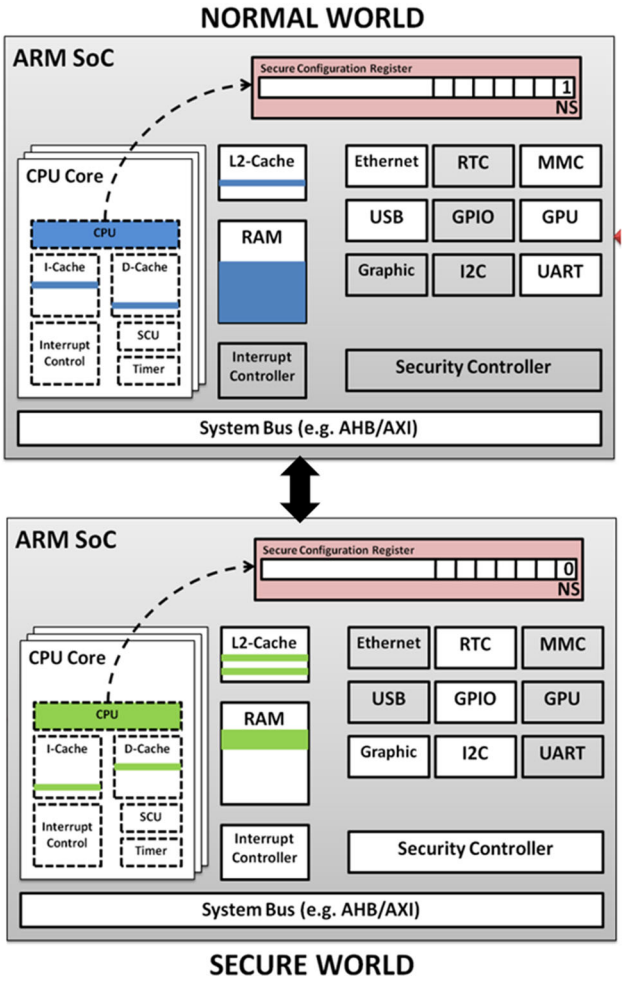

Fig. 11 ARM TrustZone virtual modes

is set to 1 for Normal world and set to 0 for the secure world. This is illustrated in Fig. 11.

9.2 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. The 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. However, when the NS bit is 1 and the processor is in the non-secure mode, it can access only the configured non-secure address space.

9.3 Software architecture

A dedicated secure world OS has 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. 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 and preventing one secure task from tampering with the memory space of another.

9.4 Booting a secure system

The most important and vulnerable instance during the life cycle of a secure system is boot time. This is the instance at which the attackers attempt to break the software while the system is powered down. 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 tampered. This is known as a 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 12 shows the schematic diagram of the secure boot sequence which takes place in TrustZone. Secure boot ensures that both 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 tampered flash images of the software.

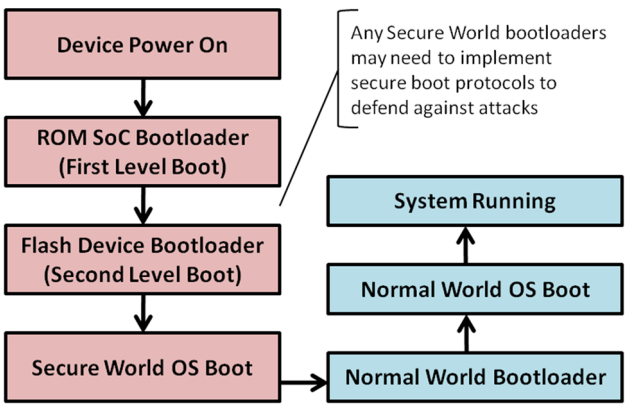

Fig. 12 Boot sequence of ARM TrustZone processors

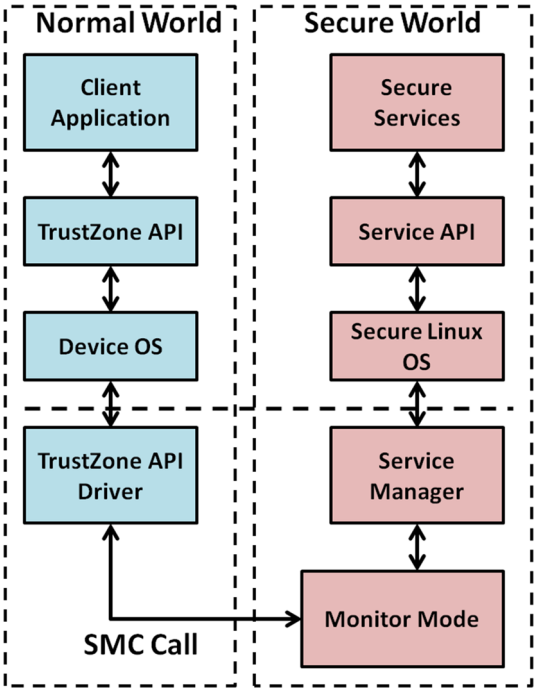

Fig. 13 ARM TrustZone secure access mechanism

9.5 TrustZone API

ARM TrustZone has developed its 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. Figure 13 shows the ARM TrustZone 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 the normal world. TZAPI is designed to be portable to almost any implementation of a secure environment [32].

10 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 dif-

ferent virtualized environments, some observations are made across all the vendor solutions and products while analyzing their core [33]. These are listed below:

10.1 Absence of secure storage

One of the extremely valuable features 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.

10.2 Absence of secure clock

Almost 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.

10.3 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.

10.4 Security provided through virtualization technique

Each of the physical processor cores in the ARM TrustZone processor design provides 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 contain the security advantages of a dedicated hardware TPM chip. Although ARM offers virtualization extensions, it is not mandatory for the vendor to apply these security extensions. As a result, many ARM-based SoC smart phones lack this security virtualization support and only operate in the normal world.

11 Proposed security model –mTPM

In this section, the proposed solution has been described. An effort has been made to offer a standardized mobile security that should address the limitations 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.

11.1 Suggested modifications in standard and mTPM

As described earlier, TCG specifications in MTM specify obsolete cryptographic primitives and their respective RoTs have been left open for the implementers. Therefore, the proposed model specifies all of them, such that it can be standardized for the entire industry. Accordingly, the proposed model gives following specifications:

(i) The Proposed Model implements 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 with TPM v2.0 to bring all the TPM manufacturers to a unified platform.
(ii) 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 constraints. Therefore, it is proposed that the security functions are integrated into a dedicated processor core of the main processor. This allows a flexible, cost-effective and low power consumption implemen-

tation. 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 are covered in the next sub-section.

(iii) With the advancement in technology and implementation of IoT networks, the smartphones have also become part of these networks and interact with low-power low-performance sensors and actuators. Since these sensors and actuators can only have lightweight cryptographic primitives, the smartphones should also have compatible lightweight cryptographic primitives. Therefore, mTPM proposes lightweight cryptographic primitives along with traditional cryptographic primitives as recommended by TPM v.2.0.

- *Symmetric Cryptographic Algorithms* After a literature survey carried out on lightweight block ciphers, NIST has recommended DESL, SEA, TEA, SIMON and SPECK. Table 2, illustrates the comparative analysis of lightweight block ciphers carried out in Sect. 8. Piccolo algorithm provides the best results of throughput and relative hardware size and is prioritized when implementing the algorithms in hardware (provides 237 Kb/s of throughput for 80 bits of key size utilizing 79 bytes of RAM and 2434 bytes of ROM). Simon and Speck are the algorithms made by NSA but have not been publically released. SPECK's hardware and SIMON's software implementation have proven to be among the best algorithms for resource constraint devices which provide the maximum throughput of 855 Kb/s while having minimum memory requirement. Hence after the analysis, following additional lightweight cryptographic primitives are proposed; SIMON/SPECK and PRESENT
- *Asymmetric Cryptographic Algorithms* mTPM proposes Elliptial Curve Cryptographic (ECC) algorithms for asymmetric cryptography.
- *Hashing Algorithms* Table 3 shows the list of lightweight hash functions implemented widely along with their performance metrics. QUARK, PHOTON, DM-PRESENT and SPONGENT are the hashing functions standardized by NIST. According to the comparative analysis carried out in this section (Sect. 8.1), among the listed lightweight hashing functions QUARK seems to fulfill the trade-off of high throughput (3.13 Kb/s for 256 digest size), less power consumption (4.35 μW/bit) and minimum memory usage (60 bytes of RAM and 1106 bytes of ROM). Whereas PHOTON and SPONGENT provide a wide range of digest size options for implementation (support {80, 128, 160, 224, 256} bits of digest size). Hence after the analysis, following additional lightweight cryptographic hash functions are proposed; QUARK, SPONGENT

(iv) Similar to the requirement for lightweight cryptographic primitives, there is also a requirement of suitable cryptographic mode of operation. The TPM uses symmetric encryption to encrypt authentication information and provide confidentiality in transport sessions. For transport sessions, the size of data is usually much larger than the nonce, so mechanism is needed to expand the key entropy to the size of the data. For this purpose, TMP has defined MGF1 function from PKCS#1 (RFC 8017) [34]. Though this function does key expansion without lowering the entropy of the nonce, it is computationally very heavy. After a literature survey of comparative analyses, it is suggested that "Counter Mode" is a lightweight function and considered appropriate for resource constraint mobile devices. The biggest advantage of the counter mode over most block cipher modes is the possibility to precompute key stream and non-error propagation across output blocks. With the increase of streaming data in triple play services over mobile data networks, packets arriving with errors or out of order can be effectively handled in real-time. In case Authenticated Encryption scheme is desired, many standardized schemes especially Galois/Counter Mode (GCM) provide suitable option.

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

11.2 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 and captures almost all of the market, it was considered to be the best choice, as minimum changes are required to adopt the proposed mTPM. ARM TrustZone uses hardware virtualization technique to implement security and thus shares the processor, memory and other hardware essentials between secure world and normal world of operation. Therefore, certain vulnerabilities have been reported to exploit/ crack the security system. The proposed mTPM actually works around ARM TrustZone to

mitigate its shortcomings and make it conceptually compliant with TPM 2.0.

The proposed implementation technique comprises of the security implementation changes in the integrated technology of ARM TrustZone and the additional security enhancements required supporting the integrated technique in compliance with TPM v2.0.

11.2.1 Dedication of security processor

The proposed mTPM implementation model is an integrated TPM like Trust-Zone but slightly different from it considering its 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 upon the selection of "World" i.e. 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 world permanently without switching their roles at anytime. This arrangement has several advantages:

(i) The dedicated core for TPM services truly complies with the TPM v2.0 requirements as Core 0 will never perform any other functions (for Normal World).
(ii) The integrated TPM processing element provides superior security as compared to an isolated hardware device as the bus for communication between Secure and Normal worlds is inside the main processor and inaccessible for interception externally.
(iii) The dedicated core is a programmable device and provides more programmable user flexibility (instruction set) then a hardware TPM chip. This will provide the flexibility of selecting and altering different cryptographic algorithms embedded in the core for security purposes and updated later on.
(iv) It will also not increase the die size of the SoC as no separate module is being integrated with the processor.
(v) It will overall decrease power consumption as the core operates only when cryptographic and mTPM services are needed.

11.2.2 Memory storage

RM TrustZone provides no guidelines as to how to manage the memory, since 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 variables 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. Keeping this aspect in view, mTPM architecture requires the following arrangement:

(i) A dedicated "Secure Memory" for storing cryptographic keys, Random Data Pool, Application level security parameters, and intermediate stage data under processing should be provided.
(ii) 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. The mTPM recommendations included this aspect to be configured for Core 0 to prevent any chance of exposure of secure world data to Normal world.
(iii) In addition to this, a One Time Pad (OTP) storage is required to program Encryption Keys for Application and OS provider. This storage should be fusible after a write operation to prevent read back at a later stage.
(iv) 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. As its name suggests, RPMB is a mechanism for storing data in an authenticated and replay-protected manner.

11.2.3 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 variables/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 has been defined earlier but the source of random number generator is defined as under:

(i) 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.
(ii) In case, TRNG is not available, then Random Numbers may be generated by sampling 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 a segment of the sampled data with the same analog signal. The resultant data may be stored in Secure Memory dedicated to secure world processing.

11.2.4 Secure clock

Similar to TPM, mTPM also requires a hardware Secure Clock (Sclk). The Sclk is required to be hardware tamper-proof and make Non-Volatile entries and never rolls backward in time. It is accessible to Secure world processing only for configuration. The Sclk is required to perform time-bound service refusal or time-bound authorization in the secure world. It is accessible to the Normal world only through monitor kernel for Read-only operation. This hardware device is an additional component beyond regular Real-time Clock (RTC). Despite this fact, ARM has introduced it in their high security processor such as Cortex M8. However, in case Sclk is not provided, the monitor mode should use tradition Real Time Clock (RTC) that should be available to both Secure and Normal worlds. However, in that case, it should only be used for time-bound lockout but not time-bound authorization.

11.2.5 Resource allocation and availability

Hiding memory and internal peripheral 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 configurability at monitor mode layer. In mTPM, all such parameters will be available for re-configuration at OS level for flexibility of hardware allocation especially for virtualization in security processes.

11.2.6 Cryptographic key hierarchy

Just like TPM 2.0, mTPM provides four hierarchies of Encryption Keys (for authorization/signing/attestation) and Storage Root Keys (SRK) (for encryption) namely Endorsement Hierarchy (EH), Storage Hierarchy (SH), Platform Hierarchy (PH) and Null Hierarchy (NH) for greater flexibility. These four hierarchies are intended to be used by platform manufacturers and the Storage and Endorsement hierarchies. The Null hierarchy will be used by Operating systems and OS-present applications. This arrangement will encourage

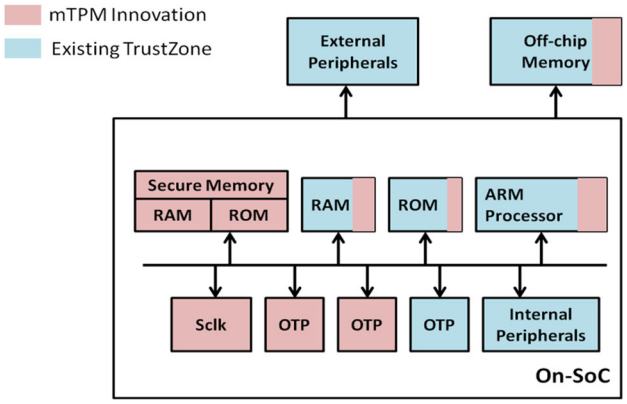

Fig. 14 mTPM and TrustZone combined SoC components

the vendors to make firmware/boot-loader controllers accessible to OS providers and end-user applications.

Figure 14 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. It should also 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 tampering i.e. the processor is tamper-proof
- Contains the intrusion detection capabilities and thus protects data disclosure. This could be achieved through the 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

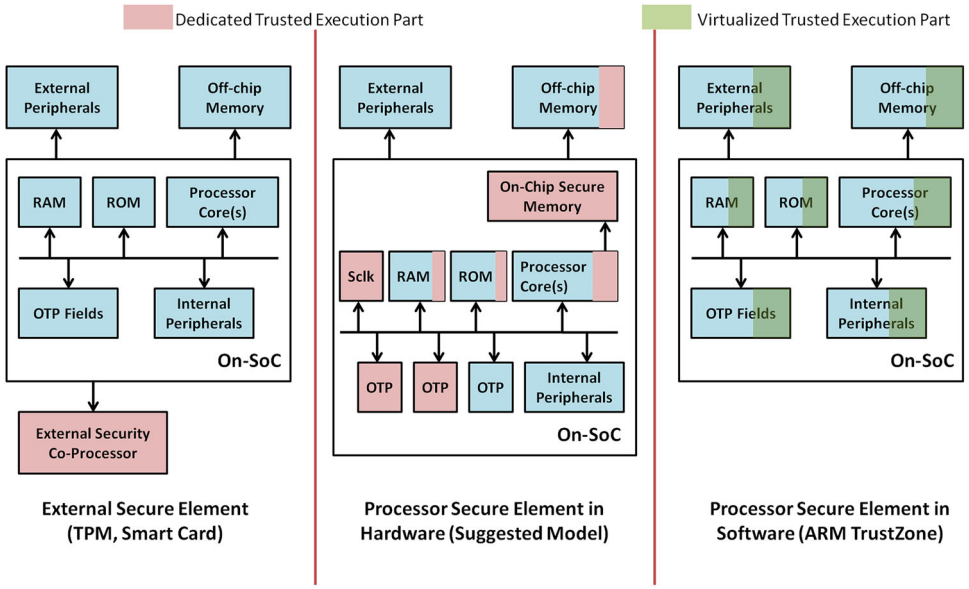

Fig. 15 TEE hardware realization alternatives

- Contains clear segregation in the 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 processed separately
- Contains its separate and segregated memory i.e. a separate RAM for non-volatile data at runtime (to store 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 15 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.

11.3 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 the device. In Fig. 16, the left figure shows the ARM TrustZone 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 has been used in our model.

In the case of multi-core architecture of TrustZone, in order to process the secure world command, the monitor

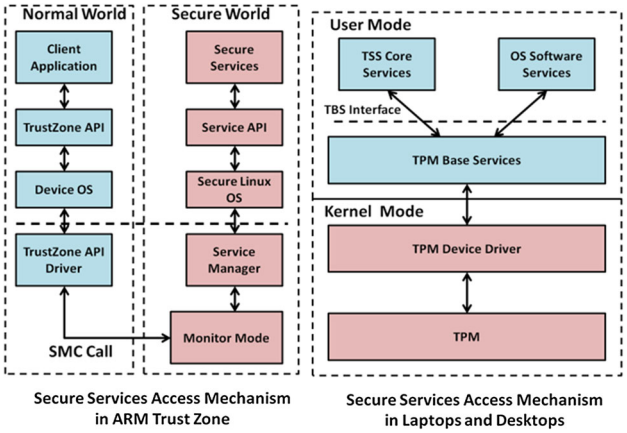

Fig. 16 Secure services access mechanism

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 operate in normal world, then 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 Fig. 17.

To implement a static computing TPM access mechanism, some changes are required in the TrustZone access mechanism which also complements our modified integrated model. In the proposed security model, the entire upper layer APIs including TZAPI will function in the same way as they did in ARM TrustZone. 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

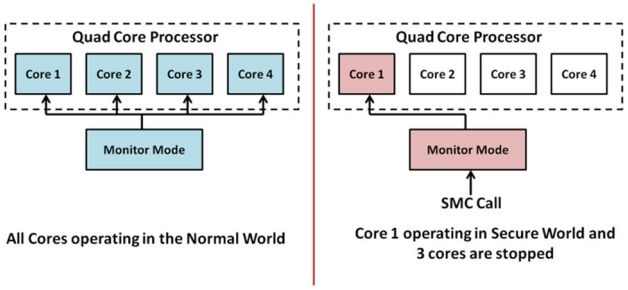

Fig. 17 ARM TrustZone secure service execution mechanism

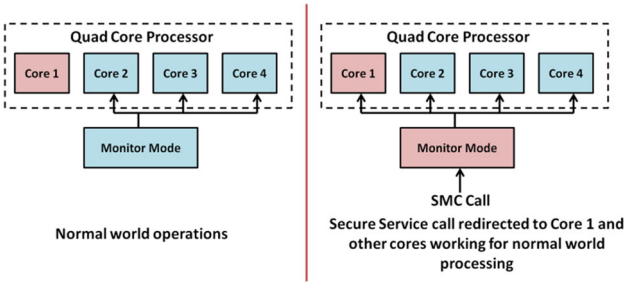

Fig. 18 mTPM security operation access mechanism

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 Fig. 13) and normal world operation towards all other processors which may work normally even during the secure operations. Figure 18 illustrates this phenomenon.

12 Proposed mTPM: proposed model implementation feasibility

In the previous section, a new mobile security model – mTPM was proposed. mTPM is a mobile security model that targets almost all the smartphones in the industry which include Qualcomm, Samsung, MediaTek, and Huawei. Nearly 99% of these industrial solutions are based on ARM TrustZone 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 with the existing technology. Moreover, the new model not only complies with the existing standards but also suggests modifications in the mobile standards and implements them in its model. In this section, the analysis of mTPM model and its implementation feasibility has been carried out.

In the last section, a security model mTPM has been proposed. The extent to which it is feasible to implement on hardware is discussed in this section.

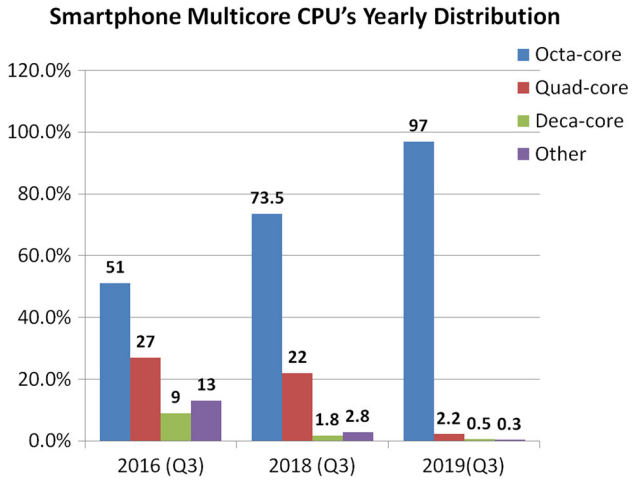

Fig. 19 Yearly market share of multi-core processors

12.1 Implementation on multi-core processors

The proposed security model 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 device, the more feasible it is 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 for mobile phones changed its research approach and the 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 has become a trend and a mobile device characteristic. Figure 19 shows graph of the yearly market share of different multi-core processors, (surveys carried out by Antutu [35–37]). The graphs reveal that 97% of the market inherits 4 or more cores in the mobile phones. Furthermore, from the market statistics of 2016, 51% of the market comprised of octa-core processors, whereas in 2019, 97% of the market comprises of the octa-core processors and almost 99.5% of the mobile devices use 8 or more cores. Hence it can be deduced from the graphs data that the mobile market industry is diverting towards using more and more multi-core processing technology.

A majority number of mobile phones exhibit more than 4 cores and almost more than 97% possesses 8 cores. As the model specifies to isolate a core for secure processing, the number of cores available for normal processing decreases. 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 devices possess more than 8 cores, hence, the security model can be implemented on most of the mobile devices.

12.2 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 the 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, then it is recommended to 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 0 is reserved and no process runs on the core except specified.
- To assign a specific task to a specific core, the following commands 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 is

```
$ 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. 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. Similar 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 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 known worldwide.

Since mTPM functions are implemented through commands at kernel level, they are not directly available to the application developer. A kernel mode API will be developed and integrated in the Android OS by the mobile device manufacturer for provisioning of mTPM resources at application layer. In this way, rooting of the device will not affect the mTPM functionality. Analogy to this effect can be drawn from the integration of Titan M (a discrete hardware TPM styled IC developed by Google and provided in its Pixel 3 smartphones in Oct 2018 [38]). In spite the major change in implementation, Titan M is accessed automatically by a single Android keystore system. For this purpose, Google has developed a Strongbox API and integrated in Android 9 and onward [39]. Beginning with Android 9 (API level 28), StrongBox Keymaster API was introduced for those devices which include a secure chip (like Titan M on Google Pixel 3). If someone is running an application on Android 28 or above, you just need to invoke the `setIsStrongBoxBacked(True)` method to let Android know that you want to use it if it's available on the device. Availability of Titan-M chip is verifiable through the call `KeyInfo.isInsideSecurityHardware() if` returns true value.

12.3 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) instead of 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 Non-Uniform Memory Access (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 is 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 is 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 is

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

By using the above set of commands, we can allocate 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.

12.4 Percentage usage of a core in a multi-core processor architecture

From the above section, it can be concluded that it is possible to isolate a core and dedicate a memory to that core only for secure functions. However, it is expected to overall degrade the performance of CPU. That is why, the time multiplexed mode has been devised by ARM [30]. However, the degradation is dependent upon the CPU load and is generally application specific. In order to get estimation about the performance degradation of the CPU after implementing mTPM, CPU profiling of Samsung Note 5 running Exynos 720 octa-core processor was done. The profiling data was collected using an open-source tool named Workload Automation tool 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 profiling results revealing the percentage usage time of each core while execution of different applications have been depicted in Fig. 20. The bar charts represent percentage time that all the cores are used in 90 s duration while executing various popular applications. The lowermost chart depicts the percentage usage time of cores while web browsing the Facebook using the 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. From the chart an important aspect is observed 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 chart shows the graph while working on MS Word document. The chart clearly depict that 45% of the all the cores are idle and less than 5% of the time, all eight cores are being used for processing. The uppermost graph depicts the percentage of time that 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.

From CPU profiling data bar charts, one can deduce that not all the cores of an octa-core processor are loaded at any given time (as all eight cores are used only 5% of the time) and some of the cores mostly remain idle. If a core is dedicated for secure processing (as suggested in mTPM) and its processes are shifted to the remaining seven cores, performance degradation may take place only occasionally. As the cores' usage is mostly underutilized, the overall performance of the mobile device is expected to be affected just marginally. Moreover, as the number of the cores in a processor is increasing with upcoming technology, reduction in performance degradation of the processor is expected in coming days. However, coupled with the advantage of enhanced security by mTPM, marginal performance degradation of the processor appears to be acceptable.

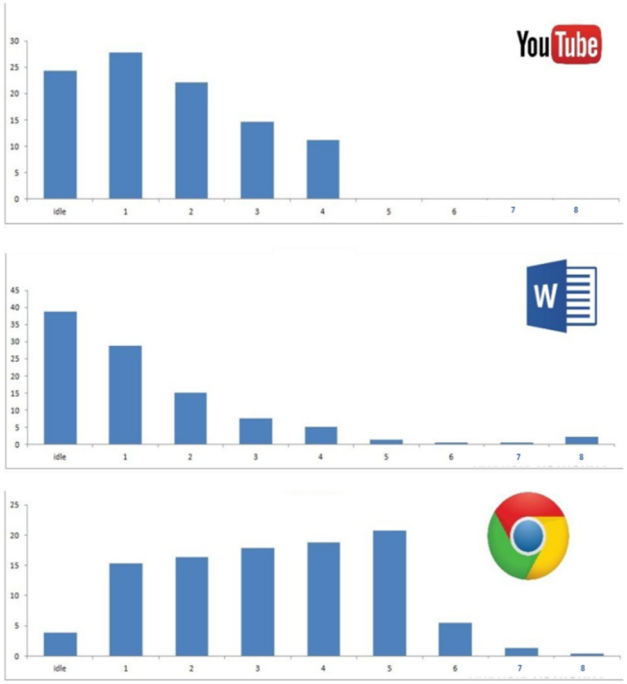

Fig. 20 Percentage of time that the number of cores is used in processing

12.5 Power consumption of a core in a multi-core processor architecture

As described in the previous section that more the number of cores better will be the computing performance and less will be the power consumed. The innovation of big little architecture in the multi-core technology has not only increased the computing performance of the device but has also decreased the power consumption of the processor. This has directly increased the battery life of the mobile devices.

If we dedicate a single core and memory for secure operations, then less number of cores will be functioning for normal operations. Moreover, as functions of the secure core require high computing arithmetic operations including cryptographic operations, hence it will consume more power than before. The power consumption can be reduced by two methods. Firstly, by using lightweight cryptographic functions, the

memory, time and the power utilized by the secure functions will significantly improve as these functions normally utilize one third of the resources and power as compared to the traditional cryptographic functions such as AES, DES, and SHA (also depicted in Tables 2 and 3). Secondly, the time and power consumed can be reduced by storing the iteratively computed results in the secured memory or by using algorithms requiring less computation which also require some data to be stored previously in the secured memory. This will reduce the power consumption of the secure core but, on the other hand, it will require more secured memory area. More the allocated secured memory area for secure operation less will be the memory left for normal operations. Hence, this will result in the reduced performance of the normal operations but, on the other hand, it will reduce the power consumed by the secure processor.

It is expected that after dedicating a single core and allocating secure memory for secure operations, the power consumed will be more than the previous power consumption of the same device with no secure mode operations. But on the other hand, it will also enhance the security. This increase in the power consumption will directly affect the battery life of the mobile device. The increase in power consumption can be reduced by using greater number of cores and large memory devices and using lightweight cryptographic algorithms which will indirectly decrease the power consumption of the normal core processing and will increase the overall power performance. However, the numerical data for the extent of degradation is subject to experimentation.

13 Proposed 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 (available open-source on the Internet) 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. Else, an alternative eMMC module can be used as a secure storage area of the secure processor. This dedicated memory provides a secure repository for the cryptographic keys and other security parameters and fulfills the requirement of ROTS. Moreover, in TPM v2.0, dedicated memory is the primary component of the TPM which has been satisfied in the proposed model.
- *ROTV* The suggested verification algorithms used in digital signatures are lightweight algorithms and can run in the secure core. The dedicated memory can be used to process and store data and no red data is allowed to 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 tamper-proof locations required to store and process measurements and assertions will be provided by the secure core and its private memory. No measurements or assertion records will be available outside the core. As the processor, since SoC is considered to be tamper-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 to perform reliable integrity measurements and establish a ROT chain of transitive measurement components.
- *API and PEE* The API and PEE will function in the proposed model similar to that applied in the 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 with 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 enabled. 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 the respective key hierarchy. This is shown in Fig. 21.
- *Multiple Hierarchies* ARM TrustZone provides a single hierarchy of storage architecture. This means that a single device key burned in the OTP will be used by the manufacturer, OS and application developers for integrity measurements. Although mobile phone manufacturers have used their own security solutions to provide keys for each level, those keys are stored in the normal storage locations and not authenticated by their respective higher level keys. mTPM suggests deploying 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 comply with it, as it is based on MTM model which standardizes single hierarchy system.

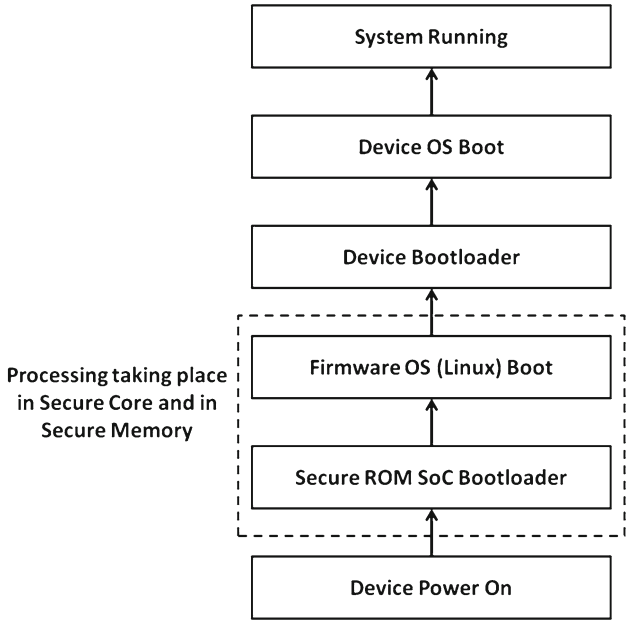

Fig. 21 Secure boot in the proposed model

Table 4 illustrates the comparative analysis of ARM TrustZone security solution and the proposed model mTPM. Different features have been compared and highlighted in the table.

14 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 extra hardware.
- Programmable flexibility as the core separation and functionality are 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.
- Exhibits high-performance capability as the cores will be available all the time for processing (in contrary to TrustZone) and minimizes the idle percentage of time during overall computing of the device.
- Utilizes less power while computing cryptographic algorithms required for encryption, decryption, hashing and signature verification as light-weight algorithms will be used for processing.
- Takes less time and is less prone to errors as the counter mode or GCM mode will be used which possesses 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 with most of the available OS and applications.
- A unified security platform will be available to all the mobile manufacturers with an 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-source

Table 4 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 is virtually divided into secure mode and Normal mode; timely sliced	Single-core is 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
# of cores required for implementation	Any	Minimum 4
Number of hierarchies	One (Device)	Four (Device, Manufacturer, OS, NULL)
Trusted execution	No	Yes
ROTS	No	Yes
ROTI	No	Yes
ROTM	No	Yes
ROTV	No	Yes
ROTR	No	Yes
API	Yes	Yes
PEE	Yes	Yes
Secure boot	Yes	Yes
Secure entropy	No	Yes
Secure clock	No	Yes
Tamper Detection	No	Yes

solution to them which is backward compatible with all the versions of OS and applications.

15 Future directions

Although there is a significant development in mobile processor technology, no manufacturer has integrated TPM functionality or even its equivalent. Google has recently developed a dedicated TPM chip Titan M and integrated in their Pixel 3 and 3XL smartphones [38]. However, despite its availability and integration in Android 9, no other vendor has integrated this chip in their smartphones. Probably, the reasons can be the continuous price war, efforts to reduce power consumption, space requirement and difficulty in integration, just to name the few. This tendency has placed a question mark on the viability of a dedicated TPM chip in smartphones. In this scenario, mTPM appears to the best choice in providing a TEE environment without any additional hardware. When an API is made available for its integration, probably it is expected to be widely utilized by many vendors.

16 Conclusion

In the paper, an effort has been made to comprehensively analyze the existing mobile device security standards. The analysis has revealed various shortcomings in the current standards along with their corresponding implementation in the form of ARM Trustzone technology In order to overcome these limitations, we have proposed a new security model mTPM as an upgrade to ARM Trustzone, for the provision of an effective TEE in Android devices. The proposed model envisages primarily the dedicated hardware implementation enabled and configured at the kernel level by the device manufacturers. An API integrated in Android OS, will make it accessible for use to the application developers. The mTPM model has been shown to comply with the current security standards (NIST, MTM, TPM v2.0) and effectively overcomes the limitations of ARM TrustZone technology. However, it has been felt that the model should be subjected to physical testing and evaluation through the fabrication of model SoC and development of its related monitor mode kernel software both for its security and performance analysis. The contents of the paper 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.

Acknowledgements This research is supported by the Higher Education Commission (HEC), Pakistan through its initiative of National Center for Cyber Security for the affiliated lab "National Cyber Security Auditing and Evaluation Lab" (NCSAEL), Grant No. 2(1078)/HEC/M&E/2018/707.

References

1. Sallam, A., The new era of mega trends: Hardware rooted security. Hardware, Security, and Emerging Solutions, Citrix Solutions. Retrieved from https://www.citrix.com/articles-and-insights/trends-and-innovation/jan-2015/the-new-era-of-mega-trends-hardware-rooted-security.html.
2. NIST SP 800-164 (2012). Guidelines on Hardware Rooted Security in Mobile Devices. Retrieved from https://csrc.nist.gov/publications/detail/sp/800-164/draft.
3. NIST SP 800-124 Revision 1, Guidelines for Managing and Securing Mobile Devices in the Enterprise. Retrieved from https://csrc.nist.gov/publications/detail/sp/800-124/rev-1/final.
4. NIST SP 800-111, Guide to Storage Encryption Technologies for End User Devices. Retrieved from https://csrc.nist.gov/publications/detail/sp/800-111/final.
5. TCG Published (2011). "Mobile Trusted Module 2.0 Use Cases", Specification Version 1.0, 4 March 2011.
6. Großschadl, J. (2008). Reassessing the TCG specifications for trusted computing in mobile and embedded systems. In *IEEE international workshop on hardware-oriented security and trust* (pp. 84–90). IEEE Computer Society.
7. Asokan, N. (2014). Mobile trusted computing. *Proceedings of the IEEE, 102*(8), 1189–1206.
8. TCG Published. (2008). TCG mobile reference architecture. Specification version 1.0, Revision 5.
9. Kim, Mooseop, Hongil, Ju, Kim, Youngsae, Park, Jiman, & Park, Youngsoo. (2010). Design and implementation of mobile trusted module for trusted mobile computing. *IEEE Transaction on Consumer Electronics, 56*(1), 134–140.
10. Markantonakis, K., & Mayes, K. (2014). *Secure smart embedded devices, platforms and applications* (pp. 71–94). Berlin: Springer.
11. McGill, K. N. (2013). Trusted mobile devices: requirements for a mobile trusted platform module. *Johns Hopkins Technical Digest, 32*(2), 544–554.
12. Dickson, F. (2014). Hardening android: Building security into core mobile devices. *Secure Networking in Frost and Sullivan, 2*(4), 19–21.
13. Samsung Knox Security Solution, Samsung Electronics Whitepaper, Version 2.2, May 2017.
14. MT2502ASOCProcessor Technical Brief (September 2014) MediaTek Corporation, Version 1.0.
15. Enhanced Security Features for Applications and Data In-use (2019). Intel SGX Product Brief.
16. iOS Security, iOS 12.3 (2019) Apple Corporation Whitepaper.
17. Antutu Benchmark Report (2018). Global Android Smartphone User Preferences for Q2.
18. NIST IR 8114 (2017). Report on lightweight cryptography. Retrieved from https://nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8114.pdf.
19. Beaulieu, R., Shors, D., Smith, J. (2015). Simon and speck: Block ciphers for the internet of things. NSA document.
20. Cazorla, M., Marquet, K., & Minier M. (2013). Survey and benchmark of lightweight block ciphers for wireless sensor networks. In *2013 international conference on security and cryptography (SECRYPT)* (pp. 1–6).
21. Lara-Nino, C. A., & Morales-Sandoval, M. An evaluation of AES and present ciphers for lightweight cryptography on smartphones. https://doi.org/10.1109/conielecomp.2016.7338557.
22. Hosseinzadeh, J., & Bafghi, A. (2017). Evaluation of lightweight block ciphers in hardware implementation: A comprehensive survey. *IEEE International Conference on New Research Achievements in Electrical and computer Engineering*, *1*(1), 1–7.

23. Rinne, S., Eisenbarth, T., Paar C. (2007). Performance analysis of contemporary light-weight block ciphers on 8-bit microcontrollers. In *Speed 2007*.
24. Hammad, B. T., Jamil, N., Rusli, M. E., & Reza, M. E. (2017). A survey of lightweight cryptographic hash function. *International Journal of Scientific and Engineering Research, 8*(7), 806–814.
25. Badel, S., Dağtekin, N., Nakahara Jr., J., Ouafi, K., Reffé, N., Sepehrdad, P., & Vaudenay, S. (2010) "ARMADILLO: A multi-purpose cryptographic primitive dedicated to hardware. In *Cryptographic hardware and embedded systems, CHES 2010* (pp. 398–412). Springer.
26. Koyama, T., Sasaki, Y., & Kunihiro, N. (2012). Multi-differential cryptanalysis on reduced DM-PRESENT-80: Collisions and other differential properties. In *Information security and cryptology–I-CISC* (pp. 352–367). Springer.
27. Berger, T. P., D'Hayer, J., Marquet, K., Minier, M., & Thomas, G. (2012). The GLUON family: A lightweight hash function family based on FCSRs. In *Progress in cryptology-AFRICACRYPT* (pp. 306–323). Springer.
28. Guo, J., Peyrin, T., & Poschmann, A. (2011). The PHOTON family of lightweight hash functions. In: *Advances in cryptology–CRYPTO* (pp. 222–239). Springer.
29. Abdelraheem, M. A. (2012). Estimating the probabilities of low-weight differential and linear approximations on PRESENT-like ciphers. In *Information security and cryptology–ICISC 2012* (pp. 368–382). Springer.
30. Building a Secure System using TrustZone Technology (2009). ARM security technology whitepaper. Retrieved from http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf.
31. Smc Calling Convention- System Software on ARM Platforms (2016). ARM Security Technology.
32. TrustZone API Specification version 3.0 (2009). ARM Security Technology. Retrieved from http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.prd29-genc-009492c/index.html.
33. Raj, H., Saroiu, S., Wolman, A., Aigner, R., Cox, J., England, P., Fenner, C., Kinshumann, K., Loeser, J., Mattoon, D., Nystrom, M., Robinson, D., Spiger, R., Thom, S., & Wooten, D. (2016). fTPM: A software-only implementation of a TPM Chip. In *Proceedings of 25th USENIX security symposium*.
34. RFC 8017 (2016). PKCS #1: RSA Cryptography Specification Version 2.2", ISSN: 2070-1721.
35. Aututu Global Phone Users' Preference Report for Q1 2019 (2019). Antutu Benchmark Report, doc/117726.
36. Antutu Report: Global Android Smartphone User Preferences for Q2 2018 (2018). Antutu Benchmark Report, doc/115174.
37. Top 10 Global Popular Phones and User Preferences, Q3 2016 (2016). Antutu Benchmark Report, doc/107641.
38. Building a Titan: Better security through a tiny chip (2018). Android Developers Platform Blog by Google Inc., Retrieved 10 October, 2019 from https://android-developers.googleblog.com/2018/10/building-titan-better-security-through.html.
39. Android Keystore System (2019). Android developers platform documentation by Google Inc., Retrieved 10 October, 2019 from https://developer.android.com/training/articles/keystore.html.

Naveeda Ashraf received her Bachelor's degree in Electrical (Telecommunication) in 2010 from National University of Sciences and Technology (NUST). In 2018, she received her MS in Information Security degree with distinction (Gold medalist) from NUST. Currently she works as a freelancer and as a visiting faculty lecturer at different universities of Pakistan. Her Research interests include Wireless Network Security, Cyber Security, Ethical Hacking, Penetration Testing and Blockchain Applications.

Ashraf Masood is on the faculty and former Dean of Military College of Signals (NUST), Islamabad. He received his BS and BE (Telecommunication Engineering) degrees in 1980 and 1985 both with distinction (Gold Medalist). He did his MS and PhD in Electrical Engineering from Michigan State University, East Lansing, USA in 1990 and 1992 respectively. His research interests are in information security with focus on cryptographic system engineering. He has also designed and developed several IT security products for diverse applications. In acknowledgement to his IT R&D accomplishments, he was conferred with several distinguished awards including National IT Excellence Award for two consecutive years in 2004 and 2005. During his 35 years of academic career, he has supervised several tens of MS and PhD thesis research work with over 40 research publications to his credit. Currently, he is also CEO of an IT Company, Principal Advisor to a consortium of IT companies, Vice Chair of Open Source Foundation of Pakistan and Patron-in-Chief of two magazines.

Haider Abbas received the M.S. degree in engineering and management of information systems and the Ph.D. degree in information security from KTH, Sweden, in 2006 and 2010, respectively. He is currently heading the National Cyber Security Auditing and Evaluation Lab with MCS-NUST. He is a Cyber Security Professional who took professional trainings and certifications from the Massachusetts Institute of Technology, USA; Stockholm University, Sweden, IBM, and EC- Council. He is serving as an Associate Editor for a number of international journals, including the IEEE Journal of Biomedical And Health Informatics, the Journal of Network and Computer Applications, Electronic Commerce Research, IEEE access, Neural Computing and Applications and Cluster Computing. He has also won many awards and has received several research grants for ICT-related projects from various research funding authorities and working on

scientific projects in U.S., Europe, Saudi Arabia, and Pakistan. He is the principal advisor for several graduate and doctoral students with the National University of Sciences and Technology, Pakistan, Al-Farabi Kazakh National University, Kazakhstan, the Florida Institute of Technology, USA, and Manchester Metropolitan University, U.K.

Rabia Latif received her MS in Information Security (2010) and PhD in Information Security (2016) from National University of Sciences and Technology, Pakistan. She is currently working as Assistant Professor in the College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia. Her Research interest includes Cloud Computing Security, Healthcare Data Security, Web Security, Cyber Security and Network Security. Her professional career consists of activities ranging from Conference Chair, Technical Program Committee Member and reviewer for several international journals and conferences.

Narmeen Shafqat is affiliated with Department of Information Security, National University of Sciences and Technology (NUST), Pakistan as Lecturer. She is currently pursuing PhD in Cyber Security at NorthEastern University, Boston, MA under the prestigious Fulbright Scholarship. She received her Bachelor's degree in Electrical (Telecommunication) and MS in Information Security in 2013 and 2016 respectively from NUST. She has acquired several international security trainings namely Information Design Assurance Red Team (IDART) from Sandia National Laboratories, USA, Cyber Executive Training from Monterrey Institute of International Studies, MIIS, USA, Decentralized IoT Systems and Security (DISS-NDSS) etc. Her research interests mainly focus on Wireless Network Security, IoT Fingerprinting, Computer Systems Security and Cyber laws.