IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 21, Issue 6, Ser. II (Nov - Dec 2019), PP 55-66
www.iosrjournals.org

Hardware Backed Security and TrustZone for Mobile Devices: Features and Use Cases

Adel R. Alharbi
Computer Engineering Department, University of Tabuk, Tabuk, 71491, Saudi Arabia

Abstract: *Rapidly expanding mobile application capabilities and features in smart phones demands increasing attention on a number of security concerns. Since operating systems are complex and vulnerable to many forms of attack, hardware backed security recently gained prominence due to the need for end-to-end security right from hardware. Advanced RISC Machines (ARM) TrustZone, an emerging hardware security technology that splits a system into a secure and normal world, defines security extensions to ARMv6 processors and higher, such as ARM11, CortexA8, A9, and A15. Security applications using TrustZone include protected applications such as Digital Rights Management, mobile payment applications, and etc. that run on the secure kernel.*
In this paper, first the mobile security and threats are discussed and then the solutions from Trusted Platform Module and Trusted Execution Environment are discussed and finally ARM TrustZone is discussed as a Trusted Execution Environment solution with details regarding its services such as Mobile Payment, Digital Rights Management & Enterprise Services.
Key Word*: TrustZone, Trusted Platform Module, Trusted Execution Environment, ARM, Mobile Security, Hardware Backed, Mobile Payment, Enterprise Services.*

Date of Submission: 24-12-2019 Date of Acceptance: 07-01-2020

I. Introduction

Today's mobile phone industry provides a wide range of services from voice services to data services to digital media services such as: digital music, video, and gaming to mobile commerce. A Smartphone that has the capability to be always online is going to be the natural expectation. In Mobile Market, rapidly advancing computing speed, n/w bandwidth and storage capacity, in turn rapidly expanding mobile application capabilities and features. This in turn demands increasing attention on a number of security concerns that result from the use of an open environment, rise in corporate environments, financial transactions, etc. not just to the end-user but to other customers [1].

Most tablets and mobile phones are depend on an ARM processor. ARM does not build the processor itself, but rather licenses chip suppliers designed by the processor. The manufacturers use this model and incorporate their specific characteristics to eventually create the individual chips. Qualcomm and Texas Instruments are examples of such chip manufacturers. Hardware manufacturers introduced hardware-based security features to enhance the security of solutions for mobile devices such as secure key storage.ARM TrustZone technology [2] is one of these features. It is a hardware-based solution that is integrated into the ARM processor cores enabling two execution environments to be run by the cores. These execution conditions are often called worlds: the normal world in which, for instance, Android operating system or some other operating system runs, and an unique protected environment whereby critical processes will function. It is possible for two worlds to run interlaced. In 2012, ARM confirmed that ARM TrustZone architecture would be integrated into each manufacturer's licensed processor design [3]. As a consequence, today The ARM TrustZone technology Processor is supplied with many smartphones.

On the other hand, operating systems are complex system that is very difficult to secure and this makes the devices to be vulnerable to many forms of attack. Hardware backed security recently gained prominence due to the need for end to end security right from hardware. TrustZone builds security into the hardware enabling the protection of memory and peripherals. It enables both secure operating system (secure operating system) and a normal operating system (normal operating system) to be run at the same time. Owing to its completeness in enabling both Rich operating system and secure operatingsystem to co-exist, this feature has received considerable attention of late.

In the first part of this paper, we provide overview on the security challenges facing mobile handsets and the need for hardware backed security including Trusted Platform Module and Trusted Execution Environment. This covers the differences between Trusted Platform Module and Trusted Execution

Environment and the standards involved and finally the available solutions of Trusted Platform Module and Trusted Execution Environment. Next, we describe the security features offered by ARM TrustZone, as Trusted Execution Environment and how it ensures end-to-end security right from secure boot to applications. As part of this, we also talk about the open source project called ”open virtual- ization” sponsored by Sierraware, to implement TrustZone and GlobalPlatform Application Program Interfaces. Finally we would discuss the details on the services such as Mobile Payment, Digital Rights Management & Enterprise Services using TrustZone. Furthermore, To make cross-referencing easier, the acronyms can be defined as shown in table 1.

Table no 1: Hardware & Mobile Security Acronyms

Abbreviation	Explanation
OS	Operating System
PC	Personal Computer
HD	High-Value
SD	Secure Digital
PIN	Personal Identification Number
OTP	One-Time-Password
TI	Texas Instruments
IT	Information Technology
CA	Conditional Access
VPN	Virtual Private Networking
CI	Integrated Circuits
I/O	Input/Output
TEE	Trusted Execution Environment
TPM	Trusted Platform Module
TCG	Trusted Computing Group
SE	Secure Element
ISO	International Standards Organization
RSA	Rivest-Shamir-Adleman
ARM	Advanced RISC Machines
AMD	Advanced Micro Devices
UI	User Interface
API	Application Program Interface
GP	GlobalPlatform
NTT	Nippon Telegraph and Telephone
SMC	Secure Memory Card
MMU	Memory Management Unit
IRQ	Interrupt Request
FIQ	Fast Interrupt Request
CRM	Customer Relationship Management
NFC	Near Field Communication
UICC	Universal Integrated Circuit Card
Intel TXT	Intel Trusted Execution Technology
SMX	Safer Mode Extensions
OTA	Over-The-Air
OMA	Open Mobile Alliance
DRM	Digital Rights Management
NDS	Nintendo Developer's System
SSO	Single-Sign-On
APB	Advanced Peripheral Bus
OMAP	Open Multimedia Applications Platform
BCAST	Broadcast
NS	Non-Secure
E-mail	Electronic Mail
L1	Level 1
L2	Level 2
EAL4	Evaluation Assurance Level 4
IEC	International Electrotechnical Commission
GSM	Global System for Mobile
UMTS	Universal Mobile Telecommunications System

II. Mobile Handset Challenges

In today’s market, smartphones exhibit dual usage requirements and they are:

- Handling increasing value data such as credentials from consumer banking.
- Increasing Software complexity.
  - Larger the code , lower the robustness and higher the bugs that hackers can exploit.
  - PC-like threats become more common in the mobile device.

The factors that increase the security concerns in mobile handsets are the following:

1) Use of Open Environment: This represents the ability for user to add or change applications at any time.
2) Privacy: This represents the ability to store personal information.
3) Content Protection: This represents DRM, or CA services to protect HD content.
4) Corporate Data: This represents a protected and fast connection to their workplace applications through VPN, secure data storage, and IT department remote device management.
5) Connectivity Protection: Weak point of network communication is the handling of the key that needs to be secured.
6) Financial Risks: Ticketing, remote payment, and proximity payment functionalities.

III. Android Operating System

Android OS is a Google-led Open Handset Alliance's operating system. It was issued in 2007 officially [4]. The operating system dependson a revamped Linux kernel to more easily adapt to a phone operating system.Although the Android OS as well as its software are open-source, the original code will only be published when a new final version is released.There is no open-minded on-going development. While most applications are developed in C++, Java is supported as well. The majority of Android OS services are written in C++ in addition to applications.The Android operating system can be used by anyone. This indicates that a large number of providers have exposure to Android mobile phones. Most manufacturers offer an Android experience which appears like or even behaves such as the phones created by the Open Handset of the Alliance, like the Nexus phones introduced by Samsung and LG in cooperation with Google.Others only use and change the experience and functionality of the Android OS as a reference [5].

A few Android OS features are necessary in view of protected key storage.The first of these is to store files on Android OS. Android's file system folder structure varies somewhat from the standard Linux OS:

- **/data** used to store data from all operating system applications and services.
- **/data/data** is where apps save their files. Every device will have its own repository, accessible exclusively through the targeteduser.
- **/sdcard** this is where the SD card is placed (if present in the system). External space on older Android devices is small although quicker, therefore programmers had to either store information locally or just on the SD card.Many modern Android devices have increased compact flash capacity because they don't need an SD card anymore.

The second feature is the allocation to each application and internal resources of different logical user IDs. This is unlike from a regular Linux system in which every user receives a user ID and utilizes the user ID allocated to the user for all applications it runs [5].

IV. Mobile Security Risk

Mobile security risks are primarily identified or at least traced back to the design or production facilities' physical security [6]. The different possibleattacks on mobile handsets are broken down into the following classes:

Hack attack

This is one in which a code attack can only be carried out by the software criminal. Types include downloaded malware and viruses from a physical or wireless connection to the smartphone, tablet, or any device. Unintentionally, the system user approves the program update, which further performs the assault, in many cases of a successful hack assault. It is because whether the malware appears to be a software application maybe the user intends to download specifically, or the user may not understand the alerts displayed by the operating environment. Figure 1 represents a typical hack attack, at first malware application identify bug/flaw that excited in the operating system in order to exploit flaw, then access secret to the financial application information [7].

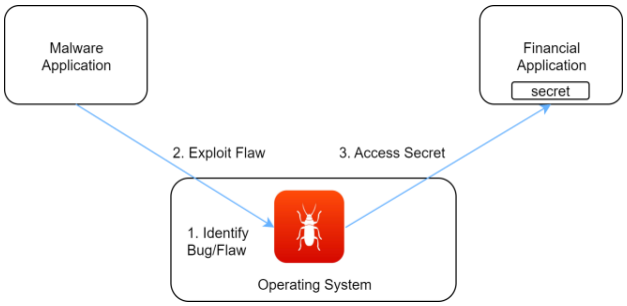

Figure 1 Hack Attack Illustration

Shack attack

Use equipment that a shop like Radio Shack might purchase, it is a low-budget hardware assault. In this situation, intruders have material access to the phone, but in the integrated circuit bundles not quite enough facilities but rather knowledge to attack. Intruders may scan bus lines, pins, and system signals using logic probes and network analyzers. Assailants could be also capable of conducting easy effective hardware assaults including requiring high or low voltage pins and bus lines, modifying memory equipment, or substituting hardware parts to malicious possibilities. Some of the current IC testing characteristics, including I/O border scan, JTAG debug, and BIST (built-in self-test) functions, could be used to manipulate a chip's functioning state. [8].

Lab attack

It is a high-budget hardware attack. In case, hackers have exposure to lab equipment along with microscopes, it is possible to perform limitless device reversing engineering. For any delicate portion of the model, including logic and memory, attackers must be thought to be able to reverse engineer transistor-level detail. Attackers can use lasers or other methods to modify the design engineer, add microscopic logic probing for surfaces of silicon steel, and bring crashes inside a running circuit. Hackers will track analog signals as well to conduct assaults including cryptographic key assessment, such as device energy consumption and electromagnetic emissions [9], [10].

Fab attack

It is the lowest level of intrusion when malignant software is embedded in the system list or configuration of an integrated circuit of the foundry or manufacturing plant. Chip validation cannot readily detect circuitry manufactured in the chip [9], [10].

V. Hardware Backed Security

The term "security" may be employed to contain several fundamental characteristics for a design that are very distinct. Each system integration would involve a separate set of defense assets, based on the type and significance of the protected property or resource; Security is all about attempting to prevent fraudulent activity [8], [11].

Need For Hardware Backed Security

The different aspects of realizing the need for hardware backed security shall be the following:

Acknowledge the difficulties

The security solutions provided by firewall, anti-virus and other products can be neutralized by malware that runs at the same or higher privilege levels.

Prepare for worst

The attacker should not be allowed to access the secure world even though root legal rights are obtained by the intruder in the normal OS. The possible solution and to guarantee this, we need end-to-end security, i.e. hardware backed security.

Different Solutions

The different solutions for hardware security are as the following:

External hardware security
Secure Element is an external hardware security module that yields a tamper-proof hardware/software/protocol combination that is capable of enabling smart card-grade applications. Typical implementations shall include UICC,embedded Secure Element, and removable memory cards.

Internal hardware security
There are two types of internal hardware security modules as explained in table 2.

Table no 2: Internal hardware security type comparison

Security Module	Advantage	Disadvantage
• The hardware block that handles cryptographic operations and key storage. The standards that are involved in this are TCG & ISO/IEC. Some of the members are Intel, AMD, HP and others [12] TCG defines a specification called TPM for defining this.	• It provides cost reduction and performance improvements over the smart card option.	• It is capable of protecting only the key but the resources that is protected by the cryptographic methods still can be accessed outside of the cryptographic module.
• The general purpose processing engine that is placed parallel to main processor. This prevents unauthorized access to sensitive resources.	• It also reduces significant cost and improves performance over the smart card option.	• It is less powerful than the main applications processor. • It consumes significant silicon area. • Inter-processor communication is harder.

Software virtualization
Multiple software platforms are separated using hypervisor that places each platform inside a virtual machine and provides communications mechanisms [13]. The specification defined for this by GP group is called TEE specification [14]. The advantage that there is no requirement for additional hardware as every processor within an MMU is enough to incorporate a virtualization approach. However, the disadvantage that it ignores the security issues like threats coming from the debug or test infrastructure. Debug and test visibility should be completely disabled for securing against such an attack.

TPM, SE, and TEE

TPM
TPM is the specification that describes a secure crypto-processor for storing cryptographic keys that protect data. This is also called TPM security device. It includes a set of features such as: protected capabilities (Key Storage & Crypto operations), and integrity measurement and reporting [15] The following are the different supported services offered by TPM security device:

1) Remote attestation: that enables a third party to validate whether or not the code has been updated. A hash-key overview of configuration of software and hardware is used to verify.
2) Binding: encrypting information using the TPM endorsing key (which is a RSA key that is burned into the device) or another key which comes down from the chip.
3) Sealing: this encrypts binding-like information, but it also defines a condition where the data should be decrypted.

Figure 2 represents the Transitive Trust in TPM, which allows a framework that boots from a fixed root of trust and extends the boundary of trusting to include software that was not native to trust origins. The target code is first evaluated in each expansion of the trust border before implementation control is transmitted.

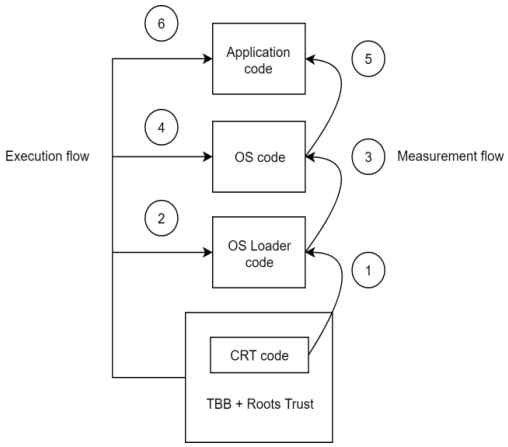

Figure 2 Transitive Trust enabled from a static root of trust to the system boot [16]

The different use cases of TPM are:
1) Platform integrity: TPM with BIOS forms a Root of Trust. TPM allow a secure storage and reporting of security relevant metrics.
2) Disk encryption: TPM can be used by encryption application.
3) Password protection: shorter or weaker passwords from user can be matched by hardware to a more harder passwords that prevent dictionary attack.

The different TPM solutions are:
1) ARM Secure Core.
2) TPMs from Broadcom and other vendors who meet ISO standards.

SE
SE is a tamper-proof hardware, software, and protocol combination embedded in a mobile handset that enables secure storage. SE is a necessity for the secure storage and execution of NFC-enabled applications, especially in the mode of card emulation [17]. NFC-enabled providers must ensure that the transaction takes place in a protected environment for users and service providers. Such defense is accomplished through the use of an SE which offers the security needed to serve different business models. There are currently several SE alternatives, nevertheless the most common ones are follow the options are shown in figure 3.

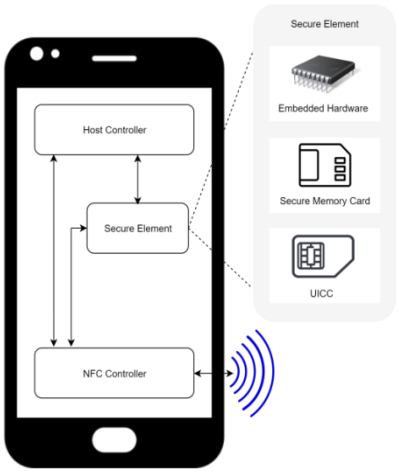

Figure 3SE Architecture

Figure 3 shows the options that should be included in the SE implementation as:
- Embedded Hardware: Embedding SE is a mobile phone non-removable part. During the production process, this chip is integrated into the mobile phone and must be modified after the product is shipped to the end user [18]. Certainly, this embedded SE chip cannot be used on other mobile phones. The time another person uses the mobile phone, it must be replaced or changed. The user must be customized to the SE of the new mobile phone.

- SMC: A removable SMC consists of memory, a built-in smart card component and a controller for smart cards. In other words, it is a memory card mix with a smart card. The SMC-based SE can host a large number of applications with the disposable property and a large capacity memory, and does not need to be reissued when the individual buys a new mobile phone [18].
- UICC: It is the physical smart card used for the application of the Subscriber Identity Module (SIM or USIM). Hence it is commonly referred to as a SIM (Subscriber Identity Module) or USIM (Universal Subscriber Identity Module). UICC-based SE is a disposable smart card used in GSM and UMTS networks for mobile terminals [18].

TEE

A "Secure World" that is physically and logically separated from "Normal World" is called TEE. Figure 4 shows the architecture of the TEE [19].

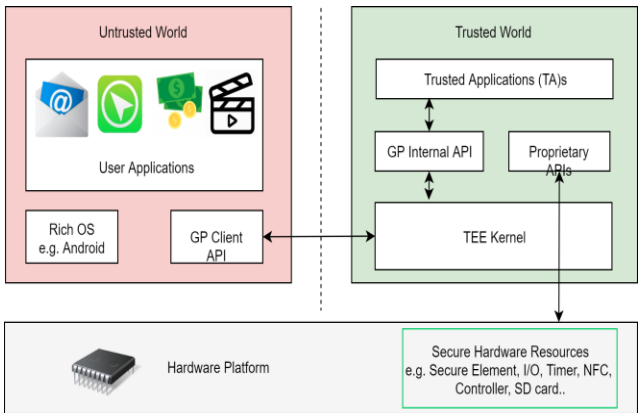

Figure 4 Architecture of the GP TEE

TEE encompasses several elements, secure execution of sensitive and critical applications inside a virtualized environment and secure boot ensures system software components are all in a well knownand trusted state [14]. The standards that are involved in TEE are GP [20] that includes Visa, MasterCard, NTT, ARM, and others. Note that after standard bodies involvement, ARM TrustZone APIs deprecated and all the newer implementations use GP TEE as the reference.

The different TEE solutions available are:

1) ARM TrustZone: to switch between the secure world and the normal world, ARM uses SMC.
2) Intel TXT: To switch between the secure world and the normal world, Intel uses SMX Instructions.
3) AMD secure execution environment.

Differences Between TEE and SE

The TEE provides a framework for security within the device, offering a layer of security between a typical Rich OS and a typical SE [14]. It provides robust, hardware-backed, scalable-consistent, OS-independent security. SE provides serious mobile security solutions. For example, most financial institutions (including credit cards and banks) use SE. But several use cases cannot be executed because of the performance, interaction, and user experience limitations of an SE. However, Together with TEE and SE can reach unprecedented levels of certified security. For example, the TEE can enable secure UIs and OTA credential provisioning.The major differences between TEE and SE can be described as it shown in the table 4.

Table no 3: The feature differences between TEE and SE

Feature	TEE	SE
Security	Better than Rich OS e.g.: Can resist software attacks like OS rooting, jail breaking, malware, and etc..	Better than TEE. (physical robustness and high tamper resistance against side channel attacks and therefore, it is certifiable at the highest security levels (EAL4 and above with Smart Card Protection Profile)
Processing Power	As high as Rich OS	Much Restricted
Memory Space	Similar to Rich OS	Much Restricted
Rich User Interface & Peripheral Connections	Similar to Rich OS	Restrictive
Leverage Rich OS Functionality	Leverage Rich OS Functionality while	Standalone

	maintaining adequate security	
Mobility	NA	Removable SEs (such as UICC or Micro SD) transferable from one device to another
Low Power Support	NA	NFC-enabled SEs can be used in low-power or no power modes

VI. Trust Zone

Protecting our systems and devices from attacks from hackers is a huge challenge. ARM's TrustZone architecture has a central role in tackling this problem [6], [13]. This section discuss about the main features provided by the TrustZone architecture and the security goals. It also gives an overview of the main elements of the ARM architecture.TrustZone is a virtualization technology (TEE) that enables a single physical processor core to safely and efficiently execute code from both the Normal world and the Secure world in a time-sliced fashion [2], [12]. Figure 5 represents the TrustZone architecture [22] that enables trusted PIN entry path using the following three steps:

1) From Trusted hardware peripheral input.
2) To Trusted Software.
3) To Trusted Authentication Service.

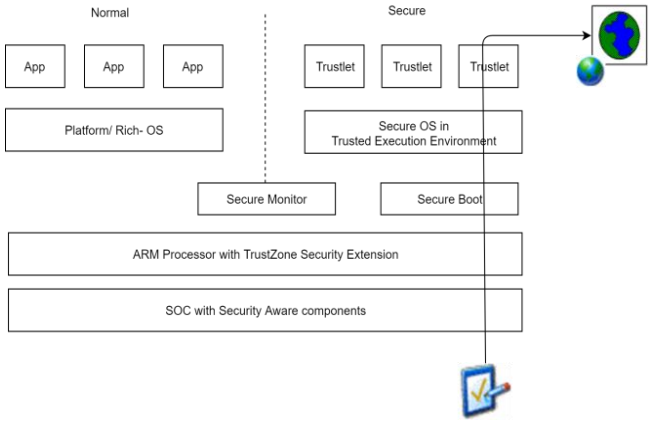

Figure 5 Trust Zone Architecture

ARM partnered with GP and defined the following specification:

- TEE Client API Specification [23].
- TEE Functional API Specification [24].
- TEE Internal API Specification [25], [26].

One of the hardware solution provider is Texas Instrument's OMAP chipsets, which integrates Trusted Foundations Software within M-Shield - under the name security middleware component. Another hardware solution provider is Qualcomm's Secure Shield (But TI's M-Shield is ahead in market). Alternatively, the different software side solution providers are Trusted Logic, Giesecke&Devrient, and Open Virtualization, Finally, trusted logic security middleware component software is completely integrated with Android 4.0 with secure boot and storage capability.

Security Goals

The primary security objective of TrustZone is to allow the confidentiality and integrity of almost any asset to be protected from specific attacks.

System Architecture

ARM TrustZone Technology offers, as mentioned earlier, a separation of the hardware in two worlds [5] as shown in Figure 6. Android OS or any other operating system is running in the normal world and delicate activities can be handled in the security world [27].

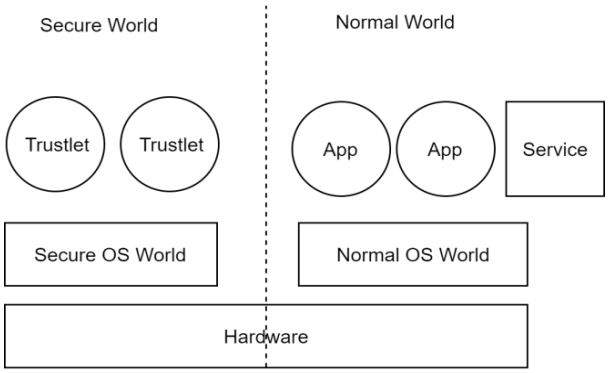

Figure 6 TrustZone's separation of hardware into two worlds

ARM TrustZone Technology's primary characteristics is the security bit on the communication bus. As depict in figure 7, the ARM processor has an AXI-bus, which is used by the primary processor to communicate with peripherals. There are two types of buses. AMBA3 AXI system bus: It ensures that no secure world resources can be accessed by the components in the normal world and this enables strong security perimeter between the two. On the main system bus, an extra control signal is added for each of the read and write channels. These security bits are also called NS, or NS bits where low represents secure and high represents non secure. AMBA3 APB peripheral bus: It enables secure environment for peripherals, such as interrupt controllers, timers, and user I/O devices. A secure interrupt controllers and timers enable a non un-interruptible monitoring of the system, a secure clock shall enable robust DRM, and a secure keyboard shall enable secure entry for a user password. In fact, debug hardware shall also be controlled using security bits [28].

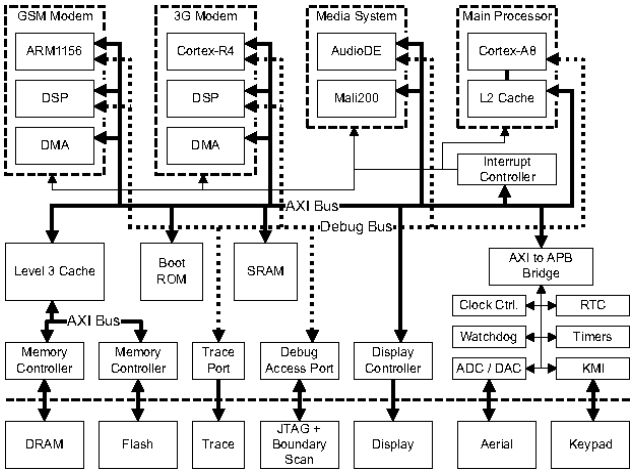

Figure 7 Architecture of the ARM and its AXI bus [27]

Processor Architecture

ARM processors with security extensions are ARMv6 processors and higher, like ARM11, CortexA8, A9, and A15 [29]. Each physical processor core in these designs provide 2 virtual cores, one Non-secure and one secure. A mechanism that is used to context switch between these virtual cores is called monitor mode. A software executing a dedicated instruction called secure monitor call, or some hardware exceptions (that include IRQ, FIQ, external Data Abort, and external Pre-fetch Abort), can trigger entry to monitor [30].

MMU & Cache & Interrupts

- MMU : It provides two virtual MMUs where one is used for each virtual processor.
- Caches : L1, L2 and beyond, processor caches are extended with additional security bit to record the security state of the transaction accessing the memory. Itsupports presence of both the type of data (data marked with both security states) inside the caches thereby removing the need for a cache flush while switching between the two worlds and also thereby enabling high performance.
- Interrupts: IRQ and FIQ interrupts are trapped directly to the monitor. This enables creation of flexible model to secure interrupt sources as there is no intervention from code in either world.

Booting a Secure System
Secure boot sequence enables security checks to be done before the normal world OS take over. A chain of trust that is generated from a root of trust that cannot be easily tampered with is called Secure Boot Sequence.

Open Virtualization Project
This project is the first open source implementation for ARM TrustZone. This project follows the ARM TrustZone specifications from GP. It provides comprehensive solution from secure boot to application management and supports all ARM architectures including ARM11, Cortex-A8, A9, and A15. It can be customized for resource constrained platforms and also easy to integrate with Linux, Android and other mobile platforms. Figure 8 represents the system architecture with open virtualization [31].

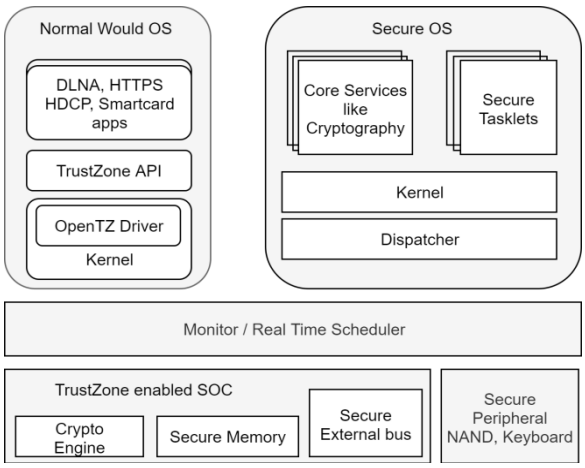

Figure 8 Open Virtualization Project

VII. TrustZone Use Cases

The different use cases of TrustZone [1] are the following:
1) Content (Premium Content Protection, Advertising Enforcement, Cloud Based Services).
2) Payment Service/Bank (Secure PIN Entry for Payments, Remote User Authentication, Mobile Banking).
3) Enterprise (Anti Malware, Data Security and Storage, ID/SSO Management).
4) Commerce (Voucher/Coupons, ID Management, and Location Based Services).

Mobile Payment Use Cases
There are different transaction types in mobile payments [32]:
1) Transactions with a remote merchant or entity: This type involves an online transaction where the user's money is transferred to another entity to pay for goods or a services. This includes sensitive Operations like user authentication and transaction validation.
2) Transactions that are a proximity payment at a merchant site [17]: NFC enables quick processing of transactions. As these transactions happen generally offline, they require a security component called Secure Element (eg:- UICC) in the handset eg:- Transaction validation.

Threats and Risks in Mobile Payments
- Retrieval of user passwords and PIN codes for user authentication and transaction validation.
- Modification of transaction essential data, such as transaction amount.
- Generation of transactions without user validation.

Threat Protection using TEE
- User Authentication: (Secure Keypad & Secure Display) Trusted User Interface securely collect a user password or PIN code that will be verified via on a remote server, or within a SE.
- Transaction Validation: This ensures that the information is displayed accurately. It prevents transaction validation without the explicit authorization of the user (e.g. through a user PIN code).
- Transaction Processing: This is isolated from any unt-rusted software attack as it is executed part of TEE.

DRM Use Case

Smartphones, tablets, and other devices enable users to enjoy high quality content, such as music, video, books, and games. Content provided by these businesses require content protection mechanisms to shield their businesses from illegal copying or distribution [33]. DRM protections are required for the following use cases:

- Prevent digital duplication (e.g. watermarking).
- Control access to the multimedia content when being used (e.g. Microsoft PlayReady or OMA DRM).
- Conditional access systems that control broadcast content reception and usage (e.g. Nagra, NDS, Irdeto, Viaccess, and OMA BCAST).

TEE provides benefits for several cases such as: storing keys, credentials, and certificates. Another benefit is executing the critical software of the on device content protection system. It is also executing the critical content protection functions and/or securely delegating to the SE and its secured access.

Enterprise

When the end-user uses a mobile device to access email, intranet, and corporate documents, there is a need for reliable, end-to-end security that ensures that corporate data is protected when stored on the device and ensures that corporate network authentication data (i.e. cryptographic certificates and keys) is not misused [34]. In this case, there are several TEE benefits as follows:

1) E-mail manager or CRM application , sensitive functionalities, such as encrypted storage and controlled access to e-mail or customer information.
2) VPN authentication, secure VPN credential provisioning and reliable authentication cryptographic calculation.
3) User to enter a password prior to accessing the encrypted corporate data and connecting to the corporate network.
4) OTP application e.g. secure authentication token for logging on to corporate network.

Moreover, TEE provides an execution environment with a speed of execution comparable to the Rich OS. Hence, security features required to support the corporate use case will not significantly impact the overall user experience.

VIII. Conclusion

This paper analyzed the background on the requirement for hardware backed security for handling mobile handset challenges and then the different solutions for the hardware security and finally the benefits of TrustZone due to its costs and features. It also discusses about the powerful security achieved when TrustZone is combined with secure element which would prevent lab attacks. In the later sections, the architecture and internal structure of TrustZone based system is discussed and the paper ends with the different use cases of TrustZone due to its rich feature support.

References

[1]. W. H. W. Hussin, P. Coulton, and R. Edwards, "Mobile ticketing system employing trustzone technology," in International Conference on Mobile Business (ICMB'05). IEEE, 2005, pp. 651–654.
[2]. B. Ngabonziza, D. Martin, A. Bailey, H. Cho, and S. Martin, "Trustzone explained: Architectural features and use cases," in 2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC). IEEE, 2016, pp. 445–451.
[3]. J. Mick, "Arm to bake on-die security into next gen smartphone, tablet, pc cores, april 2012," 2012.
[4]. O. H. Alliance, "Industry leaders announce open platform for mobile devices. press release," 2007.
[5]. T. Cooijmans, J. de Ruiter, and E. Poll, "Analysis of secure key storage solutions on android," in Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices. ACM, 2014, pp. 11–20.
[6]. X. Yan-Ling, P. Wei, and Z. Xin-Guo, "Design and implementation of secure embedded systems based on trustzone," in 2008 International Conference on Embedded Software and Systems. IEEE, 2008, pp. 136–141.
[7]. J. Krause, "Hack attack," ABAJ, vol. 88, p. 51, 2002.
[8]. T. Alves, "Trustzone: Integrated hardware and software security," White paper, 2004.
[9]. S. Skorobogatov, "Physical attacks and tamper resistance," in Introduction to Hardware Security and Trust. Springer, 2012, pp. 143–173.
[10]. S. P. Skorobogatov, "Semi-invasive attacks: a new approach to hardware security analysis," 2005.
[11]. M. Tehranipoor and C. Wang, Introduction to hardware security and trust. Springer Science & Business Media, 2011.
[12]. J. Grossschadl, T. Vejda, and D. Page, "Reassessing the tcgspecifi- cations for trusted computing in mobile and embedded systems," in 2008 IEEE International Workshop on Hardware-Oriented Security and Trust. IEEE, 2008, pp. 84–90.
[13]. J. Winter, "Experimenting with arm trustzone–or: How i met friendly piece of trusted hardware," in 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Commu- nications. IEEE, 2012, pp. 1161–1166.
[14]. H. Chai, Z. Lu, Q. Meng, J. Wang, X. Zhang, and Z. Zhang, "Teei-a mobile security infrastructure for tee integration," in 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications. IEEE, 2014, pp. 914–920.
[15]. M. A. Mat Isa, A. Abu Talib, J.-L. AbManan, and S. Rasidi, "establishing trusted process in trusted computing platform,," 01 2010.
[16]. T. TCG, "Specification architecture overview, specification revision 1.4, 2nd august 2007."

[17]. M. Badra and R. B. Badra, "A lightweight security protocol for nfc-based mobile payments," Procedia Computer Science, vol. 83, pp. 705–711, 2016.
[18]. V. Coskun, K. Ok, and B. Ozdenizci, Near field communication (NFC): From theory to practice. John Wiley & Sons, 2011.
[19]. C. Shepherd, R. N. Akram, and K. Markantonakis, "Establishing mutually trusted channels for remote sensing devices with trusted execution environments," in Proceedings of the 12th International Conference on Availability, Reliability and Security. ACM, 2017, p. 7.
[20]. G. Platform, "The trusted execution environment: Delivering en-hanced security at a lower cost to the mobile market," White Paper February, 2011.
[21]. "Globalplatform tee protection profile v1.2.1," https://globalplatform.org/specs-library/tee-protection-profile- v1-2-1/, accessed: 2019-11-13.
[22]. G. Platform, "Globalplatform card remote application manage- ment over http card specification v2. 2amendment b version 1.1. 1.3," Document Reference: GPC SPE 011, 2014.
[23]. "Globalplatform tee client api specification v1.0," https://globalplatform.org/specs-library/tee-client-api- specification/, accessed: 2019-11-13.
[24]. "Globalplatform tee system architecture v1.2," https://globalplatform.org/specs-library/tee-system- architecture-v1-2/, accessed: 2019-11-13.
[25]. "Globalplatform tee internal core api specification v1.2.1," https://globalplatform.org/specs-library/tee-internal- core-api-specification-v1-2/, Accessed: 2019-11-13."
[26]. T. GlobalPlatform, "Internal api specification v1. 0," GlobalPlat- form, December, 2011.
[27]. A. Arm, "Security technology-building a secure system using trustzone technology," ARM Technical White Paper, 2009.
[28]. A. Vasudevan, E. Owusu, Z. Zhou, J. Newsome, and J. M. McCune, "Trustworthy execution on mobile devices: What security proper- ties can my mobile platform give me?" in International Conference on Trust and Trustworthy Computing. Springer, 2012, pp. 159–178.
[29]. "Programmer's guide for armv8-a. version: 1.0," https://cs140e.sergio.bz/docs/ARMv8-A-Programmer-Guide.pdf, accessed: 2019-11-14.
[30]. C. Lesjak, D. Hein, and J. Winter, "Hardware-security technologies for industrial iot: Trustzone and security controller," in IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society. IEEE, 2015, pp. 002 589–002 595.
[31]. M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: what it is, and what it is not," in 2015 IEEE Trust-com/BigDataSE/ISPA, vol. 1. IEEE, 2015, pp. 57–64.
[32]. M. Pirker and D. Slamanig, "A framework for privacy-preserving mobile payment on security enhanced arm trustzone platforms," in 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications. IEEE, 2012, pp. 1155– 1160.
[33]. W. H. W. Hussin, R. Edwards, and P. Coulton, "E-pass using drm in symbian v8 os and trustzone: Securing vital data on mobile devices," in 2006 International Conference on Mobile Business. IEEE, 2006, pp. 14–14.
[34]. P. Vanessa, "Security of mobile banking and payments. sans insti- tute," 2012.

Adel R. Alharbi. " Hardware Backed Security and TrustZone for Mobile Devices: Features and Use Cases." IOSR Journal of Computer Engineering (IOSR-JCE) 21.6 (2019): 55-66.