WHITE PAPER

GlobalPlatform based Trusted Execution Environment and TrustZone® Ready

The foundations for trusted services

Rob Coombs/ATC-314 Date: October 31st, 2013

Foreword

Smart connected mobile devices are increasingly used for a wide range of business, financial and entertainment uses. These platforms combine customer downloaded applications, which may introduce malware, alongside corporate applications, financial tools and content protection schemes that enable the latest premium content. There is a clear need to build-in system wide security into modern Application Processors that can protect assets such as crypto keys and user credentials against software and hardware attack. This paper introduces ARM's four compartment security model and focuses on the ARM® TrustZone® technology based Trusted Execution Environment (TEE) as an important component of delivering secure services (including premium content) and applications. The paper then describes how a correctly implemented TrustZone® based TEE system can be "Secure by default", and robust against software attack.

Introduction

In 2012 ARM's silicon partners shipped 700 Million Cortex®-based applications processors into phones, tablets, DTV and other smart connected devices. Increasingly these devices are being used for high value use cases such as streaming of premium content, payment and handling of corporate or government information. To protect system assets from attack, modern ARM platforms use a combination of technologies, from the Cortex® core Hypervisor mode, to the TrustZone® based Trusted Execution Environment (TEE) and physically separate, tamper proof, secure elements secured with SecurCore® processors. This multi-layered or compartmentalized approach increases overall system security and provides the right level of protection, beyond the operating system, to the different assets within a mobile device.

Threat Landscape

Attacks on devices can come in many forms, from malware to social engineering, theft or physical loss of the device, or improperly secured devices either through misuse or by users jail-breaking their devices.

Attacks can be performed by many different methods, and malicious software can be installed by conventional means such as through a rogue app store, via social engineering or trojan, or by other attack vectors such as via the browser. When malware is present on a device, it then has the potential to escape its sandbox or process permissions, and any data held or input into the device can then become compromised.

Alternatively, if an attacker can gain physical access to the device, further attacks become possible. If the attacker can access the file system of the device, they can potentially steal data. If the data is encrypted, the attacker could copy the data off the device and perform an offline attack on the encryption. Whilst software attacks are often the main threat it is important to remember that physical attacks such as opening the device and probing the board become possible if the attacker possesses the phone.

In PC's, an even wider variety of attacks are possible, because unlike a System-on-Chip design, basic system components can be very accessible to probing, and peripherals with DMA access can be attacked and used to bypass software security measures, such as hypervisors [1][2].

The Four Compartment Security Model

To protect against these security threats, ARM offers a four compartment security model that provides increasing levels of security through a combination of reducing the attack surface and increasing isolation. A system designer will typically use a combination of the following to provide an appropriate level of protection of user and system assets:

1. **Normal World -** ***User mode/System mode - PL0/PL1 in ARMv7 or EL0/EL1 in ARMv8***

   Running processes or applications are isolated from each other by the operating system and the MMU. Each executing process has it's own addressable memory, isolated from other processes, along with a set of capabilities and permissions are administered by the operating system kernel which executes with System level privilege.

This is the normal operating state for application software together with the operating system and is often referred to as the 'Normal World', in contrast to the 'Trusted World' described below.

2. **Hypervisor Mode - *PL2 in ARMv7 or EL2 in ARMv8***

   The hypervisor allows multiple instances of the same or different operating systems to execute on the same processor as a virtual machine. Each virtual machine can be isolated from each other, and through use of a System MMU, such as the MMU-400, other bus masters can also be virtualised. This separation can be used to protect and secure resources and assets in one virtual machine from other virtual machines.

3. **Trusted World - *Secure state or EL3 in ARMv8***

   Using the TrustZone® security extensions, allows the system to be physically partitioned into the secure and non-secure components. This provides further isolation of assets and can be used to ensure that software operatiing within the normal operating system cannot directly access secure memory or secure peripherals.

4. **SecurCore**
   SecurCore processors enable physically separate, tamper proof ICs offering secured processing and storage protected against both physical and software attack.

The design of security architecture conventionally relies on two basic concepts: the principle of least privilege, and the partitioning of the system into protected compartments. For example, the TrustZone® based TEE is normally designed to maintain its isolation even if the Normal World has been compromised. A malicious hacker may take over the Normal World and spy on communications to the TEE, but the Trusted World will retain its integrity and confidentiality.

ARM considers that the hypervisor is able to offer a "protected" environment where the combination of the MMU and the hypervisor can protect the Guest OS and associated assets e.g. a media player or business email client.

TrustZone® and the Trusted Execution Environment

The TrustZone based Trusted Execution Environment provides a "Trusted World" where the security boundary is small enough to offer a route to certification and provable security. It is typically used for securing cryptographic keys, credentials and other secure assets. TrustZone® offers a number of system security features not available to the hypervisor: it can support secure debug, offer secure bus transactions and take secure interrupts directly into the Trusted World (useful for trusted input). There is an argument to restrict the amount of security functionality in the trusted world to limit the attack surface and make certification a practical proposition.

The TrustZone® security extensions work by providing the processor with an additional 'secure state' which allows secure application code and data to be isolated from normal operations, by only allowing execution of secure code or access to secure addresses, which could be data held in memory or secure peripherals, whilst in the 'secure state'. This allows basic system peripherals and resources to be

partitioned into those for secure use and those for normal operational use such as memories, IO controllers and peripherals. Conventionally, the Trusted World is used with its own dedicated secure operating system, called the Trusted Execution Environment (TEE), that works together with the conventional operating system, such as Linux or Android, to provide secure services. The interface from client applications to access the TEE, and for the use of trusted applications executing within the TEE are standardised by GlobalPlatform [4].

In a system that has both a hypervisor and a TrustZone® based TEE they can benefit mutually from their implementation: they are "better together". The TEE can enable integrity checking of the hypervisor or wider system, so if compromised, critical programs can be prevented from running. The TEE can benefit from the hypervisor by putting greater functionality into the hypervisor and keeping the amount of trusted code small.

A modern mobile device may have a number of secure elements owned by different parts of the value chain. The SIM card may be owned by the operator, the OEM may have its own SE and the OS may require access to a SE for holding keys or performing system integrity checks. Secure Elements have the benefit of tamper proof resistance to physical attacks. Because secure elements do not have access to an input method or display it can bebeneficial to establish secure communications with the secure element from the TEE. GlobalPlatform is working on the standardisation of an API for communication with a secure element from within the TEE.

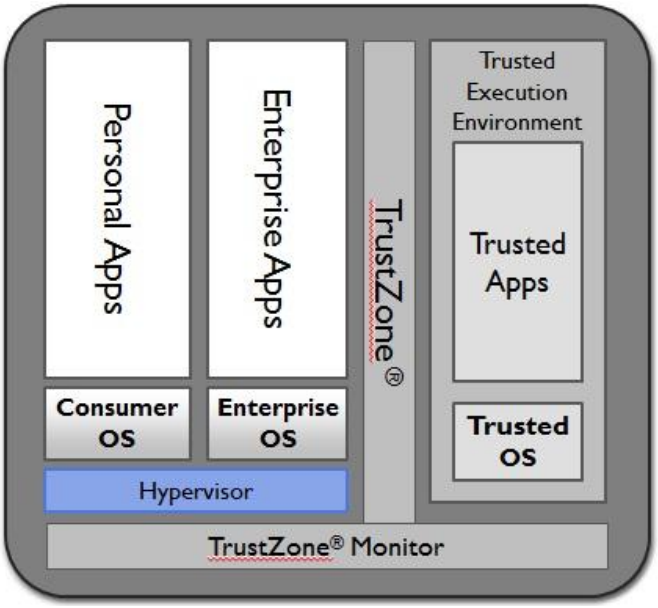

Fig 1. A modern application processor combining a TrustZone® based TEE and hypervisor

TrustZone® Media Protection & TrustZone® Ready

To enable the vision of any content on any screen requires that premium content is made available to consumers on mobile devices. For film studios to enable their valuable "early window" content requires that their assets are protected from video piracy. This requires a Digital Rights Management (DRM) scheme that can appropriately decrypt the content according to the user's rights. It is a common requirement that:

1. There is a trusted boot process that ensures system integrity
2. The DRM and its associated keys are isolated from the main operating system
3. Decrypted content is hardware protected from the main operating system

ARM proposes to its partners two different schemes for securing premium content, called TrustZone® Media Protection version 1 (TZMPv1) and version 2 (TZMPv2). These make use of a TrustZone® based TEE and system IP such as TrustZone® Address Space Controllers (TZC400) and System MMUs (such as MMU-400). Detailed documentation is available to partners as part of the TrustZone Ready Program that supports the integration of ARM's TrustZone technology into System-on-Chip designs.

The two most significant Platform Definition Documents or PDDs , which are part of the TrustZone® Ready Program are the:

- Trusted Base System Architecture (TBSA)
- Trusted Board Boot Requirements (TBBR)

These documents describe what is required rather than "how to do it" and should form a useful checklist for system designers. TBSA-Client focuses on Content Protection, Payment and Enterprise (for the client) use cases.

Characteristics of a Secure Platform

So, how can a TrustZone based TEE be used to secure the platform, and provide a means to provide strong security?

In the UK the Government's National Technical Authority for Information Assurance (IA) provides openly published recommendations on how to secure computing environments and platforms and how to design systems to be "secure by default" [3]. We refer to it here as it is a publicly available source of best practice.

This paper examines each of those recommendations, and presents how use of the TrustZone security extensions, as well as a Trusted Execution Environment, can be used to create a robust and secure platform that meets the highest standards, with little effect on design or manufacturing costs. We aim to show how correct use of TrustZone can provide secure isolation of secure assets, which cannot be accessed or attacked by conventional means. ARM's TrustZone technology can be used to defend

against software attacks and with its correct use and configuration can be used to defend against many physical attacks as well.

The following sections discuss point by point how using ARM TrustZone technology can be used to secure ARM based platforms so that they are “Secure by Default”.

Processor Security Controls Limit Access and cannot be Bypassed

Legacy systems can often provide unnecessary modes of operation which when misconfigured or left enabled, can allow security controls to be bypassed. An example of this cited by CESG, is the ‘System Management Mode’ (SMM) of x86 architectures.

The TrustZone® security extensions have a monitor mode which provides a single point of entry into the Trusted World which cannot be bypassed. It acts as gatekeeper and can control the interaction between the two worlds, and is the only permissible way to access the Trusted World. There is no other means to execute secure application or system code in the secure state. The mechanisms to enter into the secure state and monitor mode are well defined, and verified during the design of the core. There will be no other means to enter the secure state, and without entering the secure state no secure assets can be accessed.

If malicious software were to attempt an attack on the Trusted World, it must first bypass the monitor mode, which with appropriately configured page tables will not itself be able to address any secure assets or secure executable code.

Direct Memory Access (DMA) is Limited and Controlled

The TrustZone® security extensions place an additional bit, known as the NS (or Non-secure) bit on the AXI system bus. These bits indicate if a transaction can be secure, and are used to indicate the processor state when the transaction was requested. For instance, when executing within the Trusted Execution Environment the transaction will be secure, and non-secure while in normal operation and executing within Android, Linux or any other conventional operating system.

This additional bit on the bus effectively partitions the system into secure or non-secure parts, and has the effect that secure peripherals or memory cannot be accessed from the Normal World. Other bus masters apart from the CPU can optionally have the capability of making a secure transaction, or alternatively can be restricted to being only part of the Normal World. Peripherals can be statically configured to be secure or non-secure or by using an ARM TrustZone® Protection Controller can be dynamically configured, so as to be accessible by the Trusted World only or Normal World as required. Such configuration is obviously a secure operation. This can be useful for instance, for when changing a touch controller normally used for input into the Normal World to secure input only, where you temporarily do not want the Normal World to be able to access the peripheral.

So, by use of the NS bit, bus masters can be configured to be to restricted to non-secure transactions only, preventing their access to secure assets, meeting the requirement of limiting and controlling DMA.

DMA from External Devices is Additionally Protected

It is strongly recommended by ARM that the NS bit if taken off chip, forces all transactions from external masters to be non-secure and so secure peripherals, whether RAM, fuses, or IO, can only be controlled by on-chip bus masters. Once exposed off chip, with access to the rest of the AXI system bus, if an attacker could force the line to the secure state, they could force a secure address access.

By this mechanism, external devices cannot access secure assets on-chip, meeting CESG's requirement.

Central Processor Access From Other Processing Elements is Minimised and Controlled

As stated above, only bus masters which have been designed to be secure can access Trusted World assets.

Processes Consuming Platform Resources can be Identified and Controlled

Trusted applications executing within the TEE can potentially access the address space of the normal world, which allows trusted applications to potentially act in a supervisory role of the Normal World.

Potentially, resource allocation can be securely controlled and supervised by a trusted application running within the TEE, or alternatively, a trusted application can be used to monitor and supervise the OS in arbitration of system resources.

Debug Functionality Does Not Compromise Security

ARM recommends that JTAG and trace debug be disabled in production devices. If present, JTAG could potentially compromise security by allowing inspection of memory or arbitrary code execution, and trace could compromise security by leaking important information to an attacker.

In many designs this is commonly configured by a fuse at time of manufacture to differentiate production from development parts.

I/O Control

All IO peripherals can be selectively chosen, either dynamically or as part of the design of the IC, to be secure or non-secure as required. This allows secure input, secure display or secure storage, which are inaccessible to the Normal World.

For instance, a touch controller and display controller could be accessible by the Normal World. A trusted application, by use of the ARM TrustZone® Protection Controller, could dynamically change them to be

secure periherals, at the request of Normal World software which can suitably suspect the Normal World device drivers.

The trusted application could then receive a secure PIN entry, and display secure text, both of which could not be physically accessed by software running in the Normal World. The trusted application could then reset the state of the peripherals before returning them back to the Normal World.

Alternatively, an interface to a storage device could be exclusively held by the Trusted World, or the Trusted World could alternatively encrypt its data with keys held only by it, and then store that data in the Normal World.

In addition, paths for video and audio can similarly be configured by the Trusted World to be secure, providing secure decoding and display for premium content.

In this way, I/O from the Trusted World can be properly secured and isolated from potentially malicious software in the Normal World.

Device Identity

ARM recommends that each System-on-Chip design retains its own Hardware Unique Key, stored in fuses. This can be used to derive and secure other keys which can be used to uniquely identify the device.

The trusted applications executing in the TEE can securely authenticate themselves, by use of other keys stored in the TEE. This can also be used to protect other world readable secure identification assets which cannot be allowed to change, such as serial numbers for IMEI or MAC addresses.

This allows strong authentication at the application layer, or when network drivers are placed in the Trusted World, this can also allow secure authentication lower down the network stack and prevent spoofing of unauthorised identification numbers, such as for instance, spoofing of MAC addresses on a device.

Secure Credential Storage

The Trusted World can be used to securely store and manage private keys. The actual encryption or decryption can then occur within the Trusted World at the request of applications in the Normal World, without ever compromising the security of the keys.

Measured/Verified Boot

The Trusted Base Boot Requirements specification (TBBR) recommends methods to construct a Trusted Boot sequence, and trusted boot is an essential requirement of a TEE.

In addition, a Trusted Application executing within the Trusted Execution Environment can provide a secure means to store measurements to provide virtual TPM functionality. This can be used in booting the Normal World operating system, to secure it’s boot process.

Secure Update/Recovery

Secure Firmware update mechanisms are specified as part of the Trusted Base Boot Requirements specification (TBBR) [7].

Control Flow Integrity

A secure platform can defend itself against attempts to subvert the order of execution of commands, and can retain the integrity of the code it is attempting to execute. There is no substitute in this, for well written, robust, defensive software.

To support this, the MMU of the ARM Cortex® core provide a XN/eXecute Never bit, to ensure data pages cannot be executed, and in addition, the ARM C Compiler/DS-5, provides stack checking functionality for protecting the integrity of system and application software to support and help software development achieve this objective.

Security Primitives

Finally, a secure platform should reasonably be expected to provide basic security primitives, and can be beneficial to platform security.

The Trusted Base System Architecture [6], recommends a series of primitives as a minimum for the TEE, such as an entropy source for cryptographic operations and generating key material, fuses to defend against rollback attack and so on.

The ARMv8 architecture also provides additional instructions for AES and SHA-1 and SHA-256, to provide an optimal, reliable and secure solution for basic operations.

Conclusion

This paper demonstrates how ARM's TrustZone® technology together with its virtualisation extensions and SecurCore™ processor cores can provide a strong, robust and secure base for System-on-Chip designs that simply cannot be matched by a PC based design. It shows how this security can be used meaningfully for business and consumer applications.

And finally it describes how the combination of the TrustZone® Architecture, the Security recommendations for System-on-Chip designs provided in TrustZone® Ready and how a GlobalPlatform Trusted Execution Environment can be a secure foundation that can protect against the main threat facing networked mobile devices: software attack.

REFERENCES

[1] Programmed I/O accesses: a threat to Virtual Machine Monitors?
Loïc Duflot & Laurent Absil – PacSec2007

[2] Subverting the Xen hypervisor
Rafal Wojtczuk – Black Hat 2008

[3] CESG: Secure By Default
http://tinyurl.com/mztb7y8

[4] GlobalPlatform – TEE System Architecture
http://www.globalplatform.org/specificationsdevice.asp

[5] ARM Security Technology – Building a Secure System using TrustZone Technology
PRD29-GENC-009492C

[6] Trusted Base System Architecture
DEN0007*

[7] Trusted Base Boot Requirements
DEN006*

**Available with NDA and dropzone access*