Tailoring TrustZone as SMM Equivalent

Tony C.S. Lo
Senior Manager
American Megatrends Inc.
Agenda

• Introduction
• ARM TrustZone
• SMM-Like Services in TrustZone
• Summary
Introduction
Introduction

- System Management Mode (SMM) was introduced on IA over 20 years ago
- Initially developed to handle power management and system critical events, it has evolved
  - SMM is used as a OS agnostic runtime firmware execution environment
  - Many OEM proprietary features require SMM
  - SMM is required to implement UEFI SecureBoot and NIST 800-147 secure flash on IA
  - SMM is even isolated from operating system access
Moving to New Architectures

• As OEMs look to move to other architectures like ARMv8-A, how do they create a secure platform feature set?

• Solution needs to be as flexible as SMM and offer the same or higher level of security
  – When possible, solution should leverage high-level PI SMM interfaces to simplify porting to new architectures

• A working solution can be built on top of ARM TrustZone
ARM TrustZone
• ARM TrustZone technology is available for many years.

• Various security applications on top of it:
  – Key protection
  – DRM
  – Electronic Payment
  – PIN Code Verification

• The ARM TrustZone architecture provides a hardware based security isolation enabling a secure world for
  – Trusted Code
  – Secure Interrupts
  – Secure Peripherals
Exception Levels Definitions

- **EL0**: The lowest exception level. Used to execute user application in Non-secure state.
- **EL1**: Privileged exception level. Used to execute operating systems, in Non-secure state.
- **EL2**: Hypervisor exception level. Used to execute hypervisor code, in Non-secure state.
- **EL3**: Secure Monitor exception level. Used to execute secure monitor code, which handles the transitions between Non-secure and Secure states. EL3 is in Secure state.
- **S-EL0**: Used to execute trusted application code in Secure state.
- **S-EL1**: Used to execute Trusted OS code in Secure state.
TrustZone Software

• ARM Trusted Firmware (ARM TF) is an open source reference implementation for EL3 software

• ARM TF intends to reduce duplicate effort by providing a single framework with:
  – EL3 Software
  – Multi Stage Authenticated Boot
  – PSCI (Power State Coordination Interface)
  – Trusted OS Interface
OP-TEE
(Open Source Portable - TEE)

• OP-TEE is an open source TrustZone based TEE solution

• OP-TEE act as one Secure Operating System which provides various API in secure world for trusted applications

• Available on GitHub
Typical Boot Sequence

- Secure Monitor Init
- Power On
- Prepare/Load Image
- Secure Monitor Init
- Normal World Bootloader
- Normal World OS Boot
- OS Runtime
- Secure World Init
- Normal World
- Secure World
Exception Levels

• Similar to IA, ARM provides different execution privilege levels
  – Traditional IA offers Ring 0 (Most Privileged) to Ring 3 (Least privileged)
  – ARMv8-A provides EL0 (Least Privileged) to EL3 (Most Privileged)

• Firmware and OS designers should make use of these ELs to isolate critical code from attacks by malicious software
Typical System Block Diagram

**Normal World**

- EL0: Guest OS (App, App)
- EL1: OS Kernel
- EL2: Hypervisor

**Secure World**

- S-EL0: Trusted Execution Environment
  - Trusted App
  - Trusted App
  - Trusted App
- S-EL1: Trusted OS Kernel
  - Trusted Drivers
- S-EL1 Payload Dispatch
- World-Switch Library
- ARM Trusted Firmware
- SoC Specific Firmware

- EL3: SMC Dispatcher
  - PSCI Core Interface
  - ARM System IP library
  - PSCI Platform
  - SoC SMC calls
Normal/Secure World Communication

• Normal world applications need a way to communicate with the secure world in certain cases
• Normal world application can generate exceptions to transfer control to monitor mode software, which performs context switching to switch to secure world
• The exceptions can be hardware or software based
  – SMC (Secure Monitor Call) is a software based exception
UEFI Security Implementation

Samples

- UEFI NVRAM Services are a runtime service that are trusted and secure services
  - TrustZone offers the opportunity for firmware developers to protect services like NVRAM
  - TrustZone offers the opportunity for hardware developers to limit access to critical hardware like SPI controllers by non TrustZone code

- To further secure platforms, each TrustZone piece of code should be developed to work at the lowest possible Exception Level
  - Only use EL3 when necessary, try to keep all code as S-EL1 or lower
SMM-Like Services in TrustZone
ARM vs IA

<table>
<thead>
<tr>
<th>Secure Memory Blocks</th>
<th>TrustZone</th>
<th>SMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure Memory Blocks</td>
<td>Secure Memory Region</td>
<td>SMRAM</td>
</tr>
<tr>
<td>Secure Mode</td>
<td>EL3/S-EL1/S-EL0</td>
<td>SMM</td>
</tr>
<tr>
<td>Enter Secure Mode via</td>
<td>SMC or Secure Interrupt</td>
<td>SMI</td>
</tr>
</tbody>
</table>

- Secure Memory Region: Can be one or multiple blocks.
- SMC: Secure Monitor Call
- Secure State: Exception Level of CPU
SMM Core/Services Integration

• On IA, once SMM is initialized, there needs to be a way to add code to this region
  – Many different OEM methods exist that make use of SW SMIs
• On ARM we need an equivalent!
  – Add SW provisioning interface within ARM TF to load SMM-like core/services
UEFI SMM Drivers/Protocols

• UEFI SMM Drivers/Protocols need TrustZone approaches:
  – UEFI SMM Drivers
    • SMM Core
    • SMM IPL
  – UEFI SMM Protocols
    • SMM Access
    • SMM Control
SMM as a Secure Payload

Normal World

SMM-Like Environment
- RAS Secure Driver
- Power Secure Driver
- BMC Secure Driver
- Secure Foundation

ARM TF
- SMC Dispatch Interrupt Forwarding
- Secure Monitor

Secure Payload
- Hardware Interrupts

SMC

S-EL1/0

EL3
Requirements

• UEFI SMI Services should be registered through ‘SmiHandleRegister’ function of SMST (System Management System Table)

• Secure memory region of TrustZone is protected before giving control to UEFI
  – The only way to access the secure memory region during UEFI is by switching to Secure World
UEFI SMM Services Invocation

• UEFI SMM Communication Protocol provides a way for UEFI drivers to invoke secure services in TrustZone.
OS Interface

• On IA systems, SMM is invoked by writing to an IO port
  – On some ARM based systems, an MMIO location can be used to invoke TrustZone services

• The UEFI specification was extended in 2.6 to include support non-IO based invocation of secure services
  – On ARM, SMM-like TrustZone Services can be invoked by OS agent
Invocation Path

UEFI ACPI Table adds one new field ‘Invocation register’ for Secure Services invocation.
Summary
Summary

• Despite the differences between SMM and TrustZone architectures, similarities allow TrustZone to be used as PI secure environment

• PIWG and ABST are the main groups that work on specifications regarding these topics
  – Interested parties are encouraged to join the conversation in PIWG and ABST!

• OEMs should pay attention to make sure their features easily migrate to new architectures
References

- **UEFI Specification 2.6**
- **UEFI Platform Initialization Specification 1.4**
- **ARM Trusted Firmware**
- **ARM Security Technology** – Building a Secure System using TrustZone Technology
- **Trusted Base System Architecture (TBSA)**
- **Trusted Board Boot Requirements (TBBR)**
- **TrustZone Media Protection Architecture (TZMP)**
Thanks for attending the UEFI Spring Plugfest 2016

For more information on the Unified EFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by