ARM TrustZone Technology

Course Description

This course is designed to give platform developers a complete overview of designing trusted systems with ARM TrustZone technology.
The course will introduce the privilege model and memory separation features of the v8-A architecture.
It will discuss platform and software requirements to allow such operations as secure boot, DRM or mobile payment.
The course discusses a complete trusted system including:

- Secure boot
- Secure monitor and EL3 Firmware
- Trusted kernel and applications
- Normal world OS drivers
- Platform design
- Memory protection

At the end of the course the participant will receive a certificate from ARM.

Course Duration

4 days

Goals

1. Understand the security need and how TrustZone address it versus traditional secure architectures
2. AArch32 and AArch64 architecture review
3. Become familiar with the secure world
4. Handle secure and non-secure interrupts
5. Understand the requirements from secure boot
6. Understand the secure monitor role and behavior
7. Manage memory system with TrustZone
8. Become familiar with the virtualization extension for increased security
9. Become familiar with TBBR and trusted firmware
10. Utilize the TrustZone software stack
11. Use TrustZone IPs for secure system
12. Understand what is secure debug and how to implement it
13. Become familiar with TrustZone ecosystem

Target Audience

Hardware and software system architects who need to understand the issues in developing trusted systems using ARM TrustZone

Prerequisites

- A working knowledge of the ARM application processor
- Knowledge of programming in C
- Experience of programming in assembler is useful but not essential
- Some knowledge of embedded systems

Course Material

ARM official course book

Agenda

Main Topics:

- Security & TrustZone Overview
- AArch32 Review
- AArch64 Review
- TrustZone Processor Architecture
- TrustZone Memory System Overview
- Virtualization Overview
- TrustZone Memory Management
- TrustZone Exception Handling
- TrustZone Secure Boot
- TBBR and Trusted Firmware
- TrustZone Software Stack
- TrustZone System Architecture
- TrustZone Debug
- TrustZone Ecosystem

Day #1

❖ Introduction to TrustZone Security

- Introduction to security
  - What is security?
  - The need for security
  - Security properties
  - Limitations and design considerations
  - Four domain security model

- Why TrustZone?
  - What are we guarding?
  - What are the threats?
  - Scale of attacks
  - Software attacks
  - Hardware attacks
  - System image verification
  - Signature key pair
  - Asymmetric key pairs
  - Symmetric keys

- Traditional “secure” architectures
- TrustZone architecture
  - The TrustZone solution
  - TrustZone hardware architecture
  - TrustZone definitions
  - TrustZone software architecture
  - Hardware architectural view
  - What TrustZone is…
  - TrustZone is not…

❖ Introduction to ARMv8-A Architecture

- What’s new in ARMv8-A?
- Privilege levels
- AArch32 registers
- AArch64 registers
- A32, T32 instruction set
- A64 instruction set
- AArch32 exception model
- AArch64 exception model
- AArch32 memory model
- AArch64 memory model

❖ **TrustZone Software Introduction**

- Introducing the secure world
  - Architecture versions
  - TrustZone architecture extensions
  - Software stack
  - Software stack with virtualization
  - Entering the Secure Monitr

- TrustZone memory system
  - Memory accesses
  - Secure cache extensions
  - Secure MMU/TLB extensions
  - Banked registers
  - Virtual address spaces
  - Accessing banked CP15 registers
  - Secure Configuration Register (SCR)
  - Non-Secure Access Control Register (NSACR)
  - Vector tables
  - Vector tables – Monitor mode

- TrustZone architecture
  - ARMv7-A
    - Privilege model
    - Processor modes
    - Register set
    - System configuration
  - ARMv8-A
    - Exception levels
    - Register set
    - System register access
    - Changing security state
    - Virtual address spaces
    - Vector tables
    - AArch32 when EL3 is AArch64
    - Privilege model when EL3 is AArch32

- Interrupt handling
  - Exception routing in ARMv7-A
  - Exception routing in ARMv8-A
  - Interrupt configuration

- Secure boot
  - Introduction to secure boot
  - Booting and the chain of trust

- Secure monitor
  - Secure Monitor code
  - Entering and exiting Secure Monitor
  - What security state am I in?

- Usage examples
- Cryptographic extensions
  - AES instructions

❖ **TrustZone Memory System Architecture**

- Introducing to memory system
  - Memory partitioning
  - Evolution of the ARM AMBA specifications
  - AMBA overview
  - AXI and ACE channels
  - TrustZone AXI hardware design
  - Hardware protection
  - TrustZone and AXI
  - TrustZone aware slave

- TrustZone fabric design
  - AXI interconnect basics
  - AXI slaves in TrustZone systems
  - TrustZone aware interconnect
  - Example: NIC-400
  - Peripheral register access
  - AXI designs with TrustZone
  - Memory wrappers
  - BP141 TrustZone adapter
  - BP147 TrustZone protection controller
  - TZC-380 TrustZone address space controller
  - TZC 400
  - MMU-40x
  - MMU-500
  - Securing peripherals

- TrustZone aware peripherals
  - Interrupt controller
  - Other masters – System Control Processor (SCP)
  - DMA controller
  - Power peripherals
  - Trusted key storage and counters
  - Trusted entropy source
  - Human interface

- ARM TBSA system
  - Example TBSA system
  - ARM Corelink and TrustZone

Day #2

❖ Introduction to ARMv8 Virtualization

- Introduction to virtualization
  - What and why
  - Types of virtualization
    - Type 1 and 2 hypervisors
    - Para vs full virtualization
- Virtualization in ARMv8-A
  - ARMv8-A virtualization
  - Stage 2 translation
  - Virtualizing exceptions
  - Virtualization of an interrupt
  - Virtualization and security

- Case study: Xen and KVM
  - Real world virtualization
  - Boot configuration before EL2
  - Paravirtualized drivers
  - Linux and KVM
  - KVM on foundation model
  - Xen hypervisor

❖ TrustZone Memory Management

- Introduction to memory management
  - Translation process
  - OS and application translation tables
  - Secure world translation tables v7-A
  - Secure world translation tables v8-A

- Caches and TLB
  - Translation tables and caches
  - Memory access from normal world
  - Memory access from secure world
  - Secure world access to shared memory
  - Memory physical attributes
  - Table walk security
  - Translation Lookaside Buffers (TLBs)

- Data cache maintenance

- DMA and other processors
  - Sharing memory with external masters
  - Cache coherent external masters

❖ **Memory Management for a Virtualized Environment**

- Introduction
  - Memory management
  - Translation regimes
  - Translation table walk
  - Stage 2 translation overhead

- Stage 2 translation table format
  - Second stage translation
  - VMID
  - Block attributes
  - Combining attributes

- Use cases
  - Legacy RTOS shipped alongside rich OS
  - BYOD (Bring Your Own Device)
  - Multiple OS's running, unaware of each other
  - Host OS to manage memory

- System MMU
  - Non-virtualized DMA driver
  - DMA driver in virtualized system
  - With a SMMU
  - What is a System MMU?
  - SMMU and multiple transaction streams
  - SMMU faults
  - Protected memory
  - Protected video system example

❖ **TrustZone Exception Handling**

- Introduction
  - TrustZone interrupts
  - Generic Interrupt Controller (GIC)
  - GICv2 and GICv3
  - Security configuration

- Interrupt routing when EL3 is AArch64
  - Exception routing
  - Normal execution – IRQ arrival
  - Secure interrupt
  - Yielding back to the rich OS
  - Interrupt security with GICv3

- Interrupt routing when EL3 is AArch32
  - Routing interrupts to Monitor mode
  - Exception vectors
  - Asynchronous exceptions – no trapping
  - Asynchronous exceptions – trap all
  - Software exceptions
  - Raising privilege levels
  - Secure interrupts – trap FIQ
  - Non-secure interrupts – trap IRQ
  - Masking interrupts
  - Secure exceptions
  - Monitor mode exceptions
  - Leaving Monitor mode
  - Unmasking exceptions
  - Interrupt Status Register (ISR)

Day #3

❖ **Device Virtualization**

- Virtualized device implementations
  - Device virtualization architecture
  - Virtualized device implementations
  - Emulated device driver
  - Para-virtualized device driver
  - Assigned device driver

- Device handling in ARM based systems
  - Device memory management
  - Hypervisor calls
  - Device interrupt handling
  - Interrupt routing to EL2
  - Virtual interrupt signaling (internal)

- GIC virtualization support
  - Virtual CPU interface registers mapping
  - GIC in a virtualized system

- List registers
- Virtual interrupt signaling (GIC)
- Virtualization in GICv4

❖ **TrustZone Secure Boot**

➢ Introduction
- Complex system example – ARMv8-A
- Secure boot
- What can secure boot achieve?
- Secure boot design
- Partitioned boot system

➢ Secure boot process
- Trust
- Chain of trust
- Booting and the chain of trust
- Boot example: load time memory
- Memory map before reset
- Boot example: reset to bootloader
- Secure bootstrap – boot level 1
- Boot example: bootloader operation
- Boot level 2 – secure RAM
- Boot example: normal world execution
- Boot failure
- Complex system verification
- Secure boot code update

➢ Secure boot design considerations
- Normal SMP boot
- Simple multi-core TrustZone solution
- Simple Trusted OS -SMP boot
- ARMv8-A boot sequence
- TBSA profile
- Example TBSA system
- Peripheral boot

❖ **TBBR and Trusted Firmware**

➢ Introduction
- TBBR functionality
- TBBR cryptographic strength

➢ TBBR boot process
- SoC setup and integrity check
- Firmware download and checking

- Software image execution
- Normal world bootloader

- Certificate overview
  - Trusted SoC firmware certificates
  - Structure of an X.509v3 certificate
  - TBBR extensions
  - Firmware certificate creation
  - Generic firmware certificate creation example
  - Certificate use for verification
  - Generic firmware verification example
  - Structure of BL2 content certificate
  - Authenticating the BL2 firmware image
  - Key certificates
  - Certificate chaining

- Other boot considerations
  - Debug and test
  - System control processor boot
  - SCP mailbox

- ARM Trusted Firmware
  - Introduction
  - Bootloader stages
  - ARM Trusted Firmware architecture
  - Using the ARM Trusted Firmware

❖ **TrustZone Software Stack**

- TrustZone software architecture
- Normal world software
  - Possible driver architecture
  - Driver interfaces
  - Example
  - Connectionless protocol
  - World shared memory
  - Managing WSM
  - Handling client crashes

- Secure world software
  - Simple/single function secure world
  - Complex secure world
  - Blocking design
  - Non-blocking design

- Case study: MobiCore
  - Complex secure world kernel
  - MobiCore TEE – API map
  - MobiCore interfaces
  - MobiCore – driver API
  - Normal world OS driver API and MCI
  - Normal world to secure world change
  - Notification queues
  - MobiCore control protocol interface
  - MCP command transfer
  - Trusted applications
  - MobiCore TEE – trustlet API functions
  - Application to trustlet communication and vice versa

Day #4

❖ TrustZone System Architecture

- Introduction
  - Design considerations
  - Example system for discussion
  - Initial view of system

- Multi-core systems
  - Multi-processor operation
  - Multi-processor issues
  - Single processor execution
  - Isolated processor
  - Trusted SMP kernel
  - Per-processor secure execution
  - Local thread SMP
  - Rich OS considerations

- Peripherals
  - What is a trusted peripheral?
  - Non-secure peripherals
  - Secure or trusted peripherals
  - Secure only peripherals
  - Trusted counters
  - Firmware trusted counters
  - World switchable peripherals
  - Peripheral world switching
  - Switching worlds – requirements
  - Switching worlds – control

- Secure aware peripherals
- TrustZone aware interrupt controller
- TrustZone operational system
- System design requirement
- Partitioned system

- Memory configuration
  - Where to put things in memory?
  - Static memory systems
  - Switchable memory systems
  - Stack and Heap

- Physical security issues
  - Cost vs caution
  - PCB layout considerations
  - Obscuring PCB signals
  - “Sheltering” vulnerable devices
  - Covering PCB devices
  - Sharing packaging
  - Sharing substrate

❖ **TrustZone Debug**

- Introduction
  - Types of debug interaction
  - Debug infrastructure
  - ARM debug architecture
  - Debug in trusted systems

- Core debug features
  - Invasive core debug
  - Debug events
  - Core debug memory security
  - Monitor debug mode
  - Debug options
  - Non-invasive core debug
  - Performance Monitoring Unit (PMU)
  - PMU security configuration

- System level debug
  - Other debug components
  - Debug authentication

❖ **Virtual Machine Management**

- Making one processor look like another
  - Virtual machine and virtual core
  - Feature registers
  - ID registers
  - Cache topology
  - Memory attribute overrides
  - IMP DEF features
- Interaction with secure state
  - SMC trapping
  - Shared memory
- VM Switching
  - Virtual core vs processor vs machine
  - Lazy context switch
  - Power management & the guest
  - Power management and EL3
- Miscellaneous
  - HVC
  - Generic timer support for virtualization
  - Virtual count and timer
  - PMU
  - Applications without OS

❖ **TrustZone Ecosystem**

- Introduction
  - TrustZone development
  - Fragmented development
  - ARM TrustZone ecosystem
- Trusted Base System Architecture (TBSA)
  - TBSA introduction
  - Example of TBSA rules
  - TBSA client use case definitions
- TrustZone Media Protection
  - TZMP
  - Hierarchy of trust
  - Hardware isolation
  - Virtualization

- SMC calling convention
  - Function type
  - SMC argument passed
  - Power state coordination interface
  - PSCI control
  - Call flow example

- TrustZone development
  - Current state of TrustZone development
  - Global platform TEE specifications
  - Trusted OS available
  - N3 TEE and TEEi
  - TrustZone development boards
  - ARM Fast Models

- TrustZone certification
  - “Certification” comes in many flavors
  - Self-certification
  - Self-certification with liability
  - Higher levels of certification
  - Security certification
  - Common criteria
  - Evaluation Assurance Level (EAL)
  - Core security evaluation
  - Certification for TrustZone systems
  - TrustZone system protection profiles
  - TrustZone applications
  - Application security evaluation