Generic Support for ARM TrustZone

Linux Plumbers’14 BoF, Düsseldorf, Germany

by
Javier González:
jgon@itu.dk
javierg@xilinx.com
What is TrustZone?

ARM TrustZone Architecture

[Diagram showing TrustZone Architecture]

Untrusted Area
(e.g., Normal World)

Trusted Area
(e.g., Secure World)

Normal App

Secure Task

Client

Server

Memory

Peripherals

Hardware

Secure Space

Kernel Space

User Space

root

root

Cortex-A / ARM1176 Processor with ARM TrustZone Technology

Secure Element (SecurCore)

Embedded OS

Open Applications

Applications Requiring Secure OS Support

TrustZone API

TZ Driver

Secure OS

Monitor

What is TrustZone?

What is TrustZone?

• Extra bit in the AMBA3 AXI Advanced Peripheral Bus (APB)
  - NS-bit: secure/non-secure

• Memory partitioning: secure and normal memory
  - Secure world memory assigned at boot-time

• Secure peripherals *on-the-fly*
  - Potentially any peripheral attached to the AXI-bus
  - In reality this is platform-specific
  - Typically: APB, SDIO, QSPI, USB, Ethernet, DMA, etc.

• A Trusted Execution Environment (TEE)
  - Execution environment separated by hardware
  - Different software stacks
  - Same processor as non-secure (normal) world (SMP & AMP)
  - Secure world triggered by a SMC call (secure monitor)
TrustZone history

- TrustZone was introduced already in 2004
- It has been a very closed technology
  - Driven by Banking (e.g., Visa, Mastercard), DRM (e.g., Netflix), and other *offload* use cases (e.g., signing keys)
- Support for QEMU from 2013 (Johannes Winter), now Linaro has taken over ([http://www.linaro.org/blog/core-dump/arm-trustzone-qemu/](http://www.linaro.org/blog/core-dump/arm-trustzone-qemu/))
- Supported in Cortex-A processors
- Supported in development boards, disabled in commercial products
  - Fully supported: Xilinx Zynq, Nvidia Tegra 3, Freescale i.MX53, ARM Versatile Express, …
TrustZone today

• Number of open source projects leveraging TrustZone in different platforms

![Logos of various open source projects leveraging TrustZone](image)

• Widely used API for user space applications

GlobalPlatform™

• No low level API that can be used by kernel submodules (and reused to expose Global Platform’s and others in user space) yet
TrustZone today

• What I have:
  - open_session, close_session, write_secure, read_secure interface for the kernel (same as TPM device driver is implemented)
  - List of input / output arguments (arbitrary number)
    - Global Platform imposes 4 arguments
  - Ported Sierraware’s Open Virtualization driver to implement this interface, also as a char device.
  - Modified Global Platform’s user space interface to use the driver and do user space ABI unit testing

• What we should have in mind (coming for ARMv8)
  - ARM Trusted Firmware: ARM, Xilinx, Linaro, and others
  - SMC calling convention
TrustZone discussion

• I want to push this mainstream for other people to:
  - Use TrustZone:
    - TPM use cases in ARM (MTPM) - IMA is a good example
    - Implement their own *secure system calls*
  - Port more drivers for other implementations
• Is open / close / read / write the right interface? Generic enough?
  - It is for my use cases but…
• Is it time to introduce *secure_read* and *secure_write* syscalls?
  - Would serve TPM, TrustZone, Secure Element (SE), Smart Card
• Common device list for *secure_processors*??
• Other things you might find interesting :)