Trusted Firmware-M

Musca-B1 Secure Enclave Solution

Mark Horvath
2020-Aug-06
Agenda

• Goals of the Secure Enclave solution
• Limitations in the Musca-B1 board
• Flash layout and boot-flow
• Details of IPC message forwarding
Reference open source Secure Enclave solution

Secure Enclave is a separate subsystem next to an application core
We are running TF-M on it as another platform configuration of TF-M

Responsibilities:

• Provides the RoT in the system
• Secure boot-flow
• Provides PSA RoT services
  • Additional level of isolation for PSA RoT
  • PSA program defines PSA RoT (most trusted security domain) and Application RoT (for additional secure services) domains
Musca-B1 board

PSA development platform for IoT
Current TF-M architecture on Musca-B1

In case of PSA Isolation Level 2
Limitations from Secure Enclave point of view

- Secure Enclave does not have serial output
- In the Musca-B1 SoC SSE-200 always has access to important system assets (flash controllers, SCC controller, etc.)
- Desired boot-flow would be to start up Secure Enclave first, and then SSE-200 should be started up by Secure Enclave
  - The DAPLink FW releases the SSE-200 subsystem from reset first, it would require complex changes to modify the boot order
  - SSE-200 BL0 component imitates that Secure Enclave is the first system that starts to run
  - Without SSE-200 BL0 the boot flow can be treated as a valid reference solution
Boot chain

**Power On button pressed**
- DAPLink
  - Starts up SSE-200

**SSE-200**
- SSE-200 BLO starts up Secure Enclave, then enters wait state
- (Some functions are copied to Code SRAM, Eflash0 can be written by Secure Enclave)

**Secure Enclave**
- MCUboot image starts to run and authenticates all images
- If all images are valid control moves to Secure Enclave’s TF-M image

**Secure Enclave**
- TF-M image starts up
- SSE-200 virtually released from reset by sending VTOR address over MHU

**SSE-200**
- TF-M image starts up
- Synchronization with Secure Enclave over MHU

**SSE-200 and Secure Enclave are ready to communicate**
Musca-B1 flash layout with Secure Enclave

- **Reset**

**QSPI flash**
- Secure Enclave PS

**Eflash0**
- SSE-200 BLO
- Secure Enclave MCUBoot
- Secure Enclave ITS
- Secure Enclave NV counters

**Eflash1**
- Secure Enclave TF-M
- Concatenated SSE-200 Image (TF-M + NS binary)

**Code SRAM**
- SSE-200 BLO
- SE MCUBoot Eflash0 driver
- Shared are for IPC parameters

**Flash Area**
- Control flow
- SE
- SSE-200

- SSE-200 BLO just starts up SE by writing SCC registers, then waits for MHU message
- If SE finds all images intact it virtually starts up SSE-200 by sending an MHU message
Secure Enclave remaps

- Sys remap is used to access flash controllers from Secure Enclave
- Secure Enclave has its internal RAM
IPC message forwarding

• Reuses mailbox solution already available in Cypress port
• In the SSE-200 subsystem PSA Proxy Partition provides virtually all PSA RoT services
• PSA parameters are copied into the Code SRAM to be accessible by Secure Enclave (This copy can be eliminated if Secure Enclave can access all memory regions)
IPC message forwarding II

1. IPC msg
2. All parameters are copied
3. Trigger sent over MHU
4. TF-M processes the request

SSE-200 CPU0 NSPE
App

SSE-200 CPU0 SPE
PSA Proxy

Secure Enclave
TF-M

Code SRAM
NS mailbox and IPC parameters
IPC message forwarding III

SSE-200 CPU0 NSPE  

App

SSE-200 CPU0 SPE  

TF-M

PSA Proxy

Secure Enclave  

TF-M

5. TF-M writes result

Code SRAM

NS mailbox and IPC parameters

6. Trigger sent over MHU

8. IPC reply

7. Results copied from Code SRAM
IPC message forwarding IV

- If a request is sent by PSA Proxy control is given back to SPM while waiting for answer from Secure Enclave
- More PSA messages can be forwarded simultaneously
- Secure Enclave cannot process messages parallely, but that can change in the future
Planned schedule

- Start review by early September
- Merge solution at end of September
Thank You
Danke
Merci
Merci
Merci
Merci
Kiitos
Kiitos
감사합니다
धन्यवाद
شكرًا
ধন্যবাদ
תודה