Secure boot & Attestation

Tamas Ban
2019.10.31

Compare against upstream MCUBoot

Feature	TF-M MCUBoot	Upstream MCUBoot
Multi image boot	✓	✓
RSA-2048 & RSA-3072	✓	✓
ECDSA various versions	✗	✓
Encrypted image support	✗	✓
Serial recovery	✗	✓
Boot data exchange	✓	✗
Rollback protection	✓	✗
Hardware key integration	✓	✗
Image RAM loading	✓	✗
NO-SWAP update feature	✓	✗

Upstream MCUBoot alignment

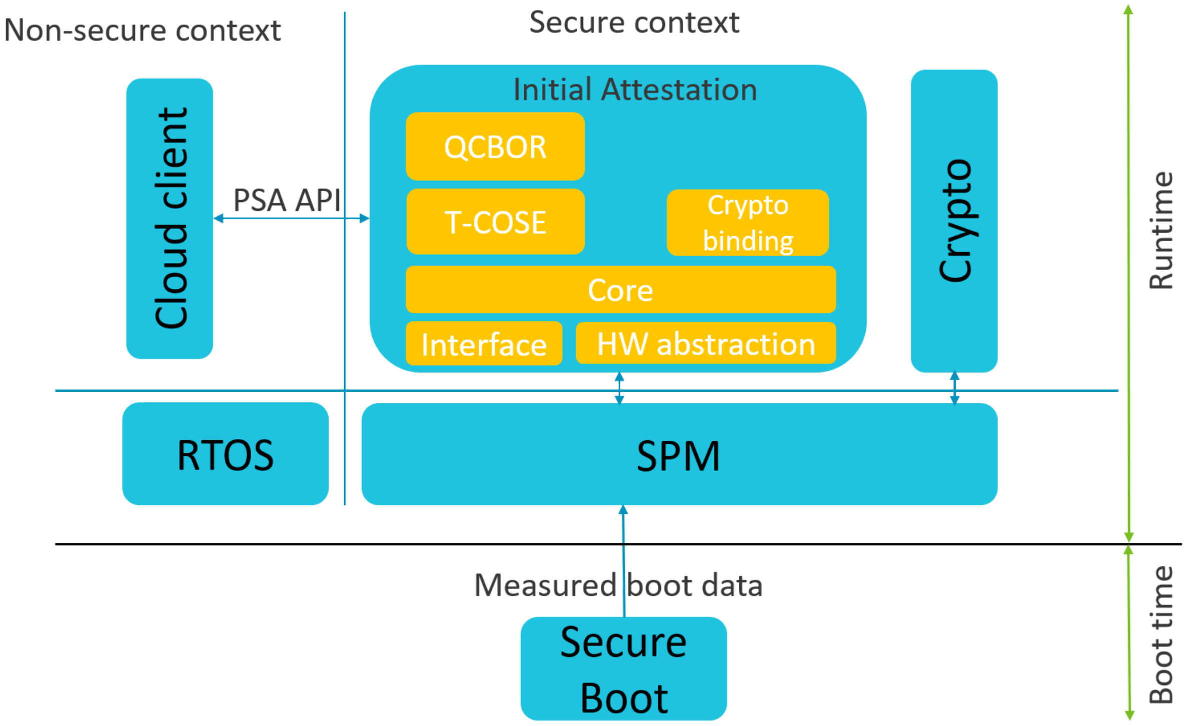
- Code sync is WIP: MCUBoot 1.4 release -> TF-M MCUBoot
- Build system integration
  - Use upstream MCUBoot as TF-M secure bootloader
- Upstreaming features (in proposed priority order):
  - Rollback protection
  - Boot data exchange
  - HW key integration
  - Others are not required for PSA certification
- Final goal is to use upstream MCUBoot as default secure bootloader for TF-M

Boot roadmap items

- Key revocation
- Might higher (>L2) PSA certification levels will require SW countermeasures against fault injection attacks
- Other ideas?

Initial attestation

- MCUBoot authenticates the firmware images and provide the boot record to runtime firmware to include it to attestation token
- Data exchange done in a shared RAM buffer
- Shared data structure follows the TLV approach
- Data can be already CBOR encoded at build time
- Attestation service collects data items, encode them to CBOR format and sign the token
- Extension of PSA attestation API:
  - tfm_get_initial_attest_public_key(...)


Non-secure context
Secure context
Cloud client
PSA API
Initial Attestation
QCBOR
T-COSE
Crypto binding
Core
Interface
HW abstraction
Crypto
Runtime
RTOS
SPM
Measured boot data
Secure Boot
Boot time