Arm’s Confidential Compute Architecture Reference Attestation Token

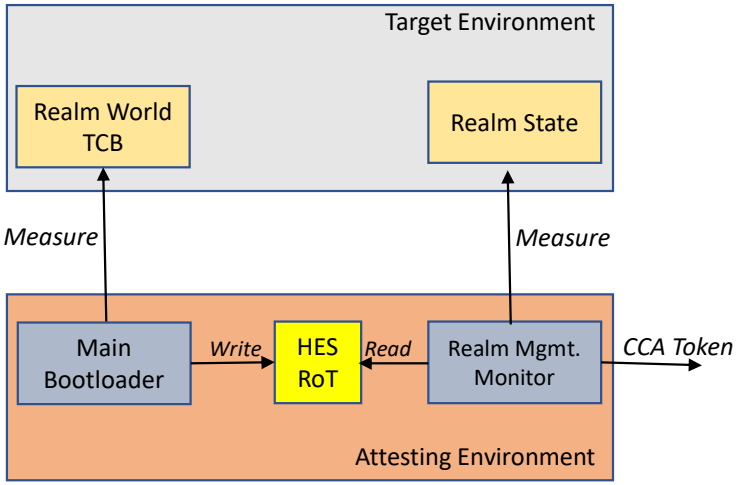
Simon Frost (Arm), Thomas Fossati (Linaro) and Giri Mandyam (Mediatek)

Introduction

- "Confidential Computing is the protection of data in use by performing computation in a hardware-based, attested Trusted Execution Environment."
  - From "A Technical Analysis of Confidential Computing", The Confidential Computing Consortium, November 2022, https://confidentialcomputing.io/wp-content/uploads/sites/10/2023/03/CCC-A-Technical-Analysis-of-Confidential-Computing-v1.3_unlocked.pdf.
- Focus of confidential computing (CC) is the protection of data at rest, in transit and in use
- For data in use executing in the context of a TEE, attestation takes on added importance
  - Allows for remote verification of computational workloads
- Arm's Confidential Computing Architecture (CCA) provides a reference design for CC implementation systems-on-chip (SoC's) with embedded TEE's
  - https://developer.arm.com/documentation/den0125/0300

Attestation for ARM CCA

- CCA requires the following primary subsystems in a CC implementation
  - RoT, that provides HW Enforced Services (HES) for securing the CCA environment
  - Realm – an execution environment for specific CC workloads/processes
  - Realm World – provides isolation and a security state for realms
- Realm World is separate from the existing Arm-defined security domains
  - Non-secure (user space) and Secure (Trustzone)



- HES Rot forms Platform Token
- Realm mgmt. monitor (RMM) forms Realm Token
- RMM retrieves Platform Token from HES RoT and forward it along with Realm Token
  - Collective message is CCA Token
- Both tokens follow EAT format

Attestation Models Supported in CCA

Delegated Model

- Completion of signing is delegated from HES RoT to RMM
- RMM obtains Realm Attestation Keypair (RAK) from RoT and uses it to sign the Realm Token
- Realm code obtains CCA Platform and Realm token from RMM and sends both to Verifier
  - Hash of RAK Public Key used as eat-nonce for Platform Token
  - Realm Token uses challenge data provided by verifier

Direct Model

- RMM creates claim set and hashes it
- Hash provided to RoT as eat-nonce to create Platform Token
- Platform Attestation Keypair (PAK) is the only attestation signing keypair

Attestation Messaging

- Platform and Realm Tokens are presented in a CBOR Message Wrapper (CMW) collection (delegated model)

```
cca-platform-token = bstr .cbor COSE_Sign1_Tagged
cca-realm-delegated-token = bstr .cbor COSE_Sign1_Tagged

cca-token-collection = {
   44234 => cca-platform-token          ; 44234 = 0xACCA
   44241 => cca-realm-delegated-token
}
```

- Direct model presents Realm claim set along with Platform Token in CMW

Next Steps

- Will continue to refine contribution and register new claims as defined in current draft in CWT claims registry
  - Please see “IANA Considerations” in draft for listing of new claims
- Invite Rats Working Group members to provide input for future versions of draft
- Also looking for case studies and any gap analysis related to CC attestation
  - Additional claims that would be required for verifiers/relying parties