QWES Token

draft-mandyam-rats-qwestoken

Giri Mandyam

Qualcomm Wireless Edge Services (QWES)

- Announced in 02/18
  - https://www.qualcomm.com/news/releases/2018/02/14/qualcomm-technologies-announces-introduction-qualcomm-wireless-edge
- Attestation a key service
  - COSE-based

QWES Token and EAT - Commonalities

- Common Claims
  - OEM ID
  - Nonce

QWES Token and EAT - Differences

- DevID (Device ID)
  - Unlike EAT ueid, DevID can be assigned specific to service provider
- HWVer – chip (SoC) version
- Context
  - Context of attestation, e.g. authentication, certificate issuance
- PKHash – hash of public key provisioned by OEM

QWES Token and EAT – Differences (cont.)

- SPID (Service Provider ID)
  - Service provider is different from OEM or device manufacturer, e.g. financial institutions
- QSEEversion – TEE version
- FWversion – version of FW used for bootstrapping device
- CSR – certificate signing request
  - Potentially saves a round trip

QWES Token and EAT - Differences

- Security state – replication of security-impacting partition in one-time programmable (OTP) memory
  - OTP settings can be used to manage security state of device (e.g. debug locking)
- App hash
  - Hash of invoking application
  - Suitable for TEE model, where an application running in rich execution (user) space can convey attestation token

Consideration for RAts/EAT

- Should security OTP be sent directly, or the information conveyed via representative claims?
- Should context be explicitly expressed?
- FW/HW versions – common way of representation?
- Device identifiers – should they be universal or specific to service provider?

Consideration for RAts/EAT (cont.)

- TEE considerations
  - Concept of service provider comes into play (TEEP architecture)
  - What models do we contemplate
    - SP app in user space – is it's integrity important? If so, how is its state conveyed?
      - Hash of binary?
    - Does SP have to own TA in which attestation token is created?