linaro™ Arm Solutions at Lightspeed

Toward a Standard Device Attestation Token for Device Assignment

Mathieu Poirier
Thomas Fossati

Linux Plumbers 2025

The Problem

Coming up with an **architecture agnostic** way to represent device claims

This claims representation is meant to be embedded in a **platform specific** envelope

The Context

Device claims are collected from sysfs by a lead attestor running in the TVM

Devices claims are packaged according to the EAT Profile for Device Attestation

Packaged device claims are sent to 3rd party verifier for attestation

Motivations

Allowing 3rd party verification services to work with the same device claims representation

SPDM and TDISP are the same regardless of the architecture → claims representation should be the same

Represent information yielded by SPDM devices → nothing more, nothing less, no allegiance

Freedom to use the information as see fit based on use cases

Linaro has no commercial gain in this specification

More Details

Currently 7 (short) CDDL files

Concise Data Definition Language (CDDL) is defined here → very easy to understand

GitHub: https://github.com/rats-device-attestation/draft-poirier-rats-eat-da

Data Tracker: https://datatracker.ietf.org/doc/html/draft-poirier-rats-eat-da

We presented this work to several entities working on Confidential Computing

2 Emails were sent to the linux-coco mailing list

4 revisions, based on collected feedback

The Specification

spdm-claims.cddl:

```
da-token = {
  &(eat_profile: 265) => "tag:linaro.org,2025:device#1.0.0"
  &(eat_nonce: 10) => bytes .size 64
  &(eat_submods: 266) => {
    + device-name => $device-claims-set
  }
}

device-name = text .regexp "dev-[A-Za-z0-9]+"

$device-claims-set /= spdm-claims
$device-claims-set /= cxl-claims
$device-claims-set /= chi-claims
$device-claims-set /= pcie-legacy-claims
```

The Specification

pcie-legacy-claims.cddl:

```
pcie-legacy-claims = {
  &(eat_profile: 265) => "tag:linaro.org,2025:device-pcie-legacy#1.0.0"
  pcie-legacy-artefacts
  ? $$pcie-legacy-claim-extension
}

pcie-legacy-artefacts //= (
  &(artefacts-text: 3805) => pcie-type-0-1-config-space-text
  &(artefacts-bytes: 3806) => pcie-type-0-1-config-space-bytes
)

pcie-legacy-artefacts //= (
  &(artefacts-text: 3805) => pcie-type-0-1-config-space-text
)

pcie-legacy-artefacts //= (
  &(artefacts-bytes: 3806) => pcie-type-0-1-config-space-bytes
)

pcie-type-0-1-config-space-bytes = bytes .size 256

pcie-type-0-1-config-space = {
  &(vendorID: 1) => bytes .size 2
  &(deviceID: 2) => bytes .size 2
  ? &(command: 3) => bytes .size 2
  ? &(status: 4) => bytes .size 2
  ? &(revisionID: 5) => bytes .size 1
  ? &(classCode: 6) => bytes .size 3
  ? &(cacheLineSize: 7) => bytes .size 1
  ? &(latencyTimer: 8) => bytes .size 1
  ? &(headerType: 9) => bytes .size 1
  ? &(BITS: 10) => bytes .size 1
```

The Specification

spdm-claims.cddl:

```
spdm-claims = {
  &(eat_profile: 265) => "tag:linaro.org,2025:device-spdm#1.0.0"
  spdm-artefacts
  ? &(vca: 3804) => bytes
}

spdm-artefacts //= (
  &(measurements: 3802) => spdm-measurements
  &(certificates: 3803) => spdm-certificates
)

spdm-artefacts //= (
  &(measurements: 3802) => spdm-measurements
)

spdm-artefacts //= (
  &(certificates: 3803) => spdm-certificates
)
```

The Specification

spdm-certificates.cddl

```
spdm-certificates = {
  default-cert-slot => cert-chain
  ? aux-cert-slots => cert-chain
}

; ASN.1 DER-encoded certificates concatenated with no intermediate
; padding.
cert-chain = bytes

default-cert-slot = 0
aux-cert-slots = 1..7
```

spdm-measurements.cddl

```
 spdm-measurements = {
  + block-id => spdm-measurement
  ? "signature" => spdm-measurement-blocks-signature
}

block-id = 1..239
```

The Specification

spdm-measurement.cddl

```
spdm-measurement = {
  &(component-type: 1) => component-type
  measurement
}

measurement //= ( &(digest-measurement: 2) => digest-measurement )
measurement //= ( &(raw-measurement: 3) => raw-measurement )

component-type /= &(immutable-rom: 0)
component-type /= &(mutable-firmware: 1)
...
component-type /= &(informational: 9)
component-type /= &(structured-measurement-manifest: 10)

raw-measurement = bytes
digest-measurement = digest
digest = [
  alg: uint / text
  val: bytes
]
```

The Specification

spdm-measurement-block-signature.cddl

```
hash-algorithm-type /= &(tpm_alg_sha_256: 0)
hash-algorithm-type /= &(tpm_alg_sha_384: 2)
...
hash-algorithm-type /= &(tpm_alg_sm3_256: 64)

spdm-measurement-blocks-signature = {
  &(slot: 1) => 0..7, ; Slot of the certificate chain used to
                      ; authenticate the measurement.  Default
                      ; should be 0.
  &(requester-nonce: 2) => bytes .size 32,
  &(responder-nonce: 3) => bytes .size 32,
  &(combined-spdm-prefix: 4) => bytes .size 100,
  &(IL1: 5) => bytes, ; L1 (see comment on the right)
  &(base-hash-algo: 6) => hash-algorithm-type,
  &(signature: 7) => bytes
}
```

```
;
; See signature generation and verification algorithms for
; MEASUREMENTS messages on page 126.
;
; L1 = Concatenate(VCA, GET_MEASUREMENTS_REQUEST1,
;             MEASUREMENTS_RESPONSE1, ...,
;             GET_MEASUREMENTS_REQUESTn-1,
;             MEASUREMENTS_RESPONSEn-1,
;             GET_MEASUREMENTS_REQUESTn,
;             MEASUREMENTS_RESPONSEn)
;
```

Remaining Work

Add support for “spdm-challenge”:

- Very similar to the specification in spdm-measurement-blocks-signature.cddl
- Allows this specification to be used for CMA scenarios

Introduce support for new bus technology when needed:

- “cxl-claims”, “chi-claims”

Add a section to describe how the EAT Profile for Device Attestation binds with the CCA Attestation Token

- We want other architectures to do the same

Open Questions

How do we keep up with upcoming versions of the SPDM specification?

Is there a need to describe TDISP artefacts?

The only one that would make sense is the Interface Report but very HW oriented

Can you use this specification? If not, what needs to change?