Workgroup: RATS
Internet-Draft:
draft-fdb-rats-psa-endorsements-01
Published: 11 May 2022
Intended Status: Informational
Expires: 12 November 2022
Authors: T. Fossati Y. Deshpande H. Birkholz
Arm Ltd Arm Ltd Fraunhofer SIT

Arm's Platform Security Architecture (PSA) Attestation Verifier Endorsements

Abstract

PSA Endorsements include reference values, cryptographic key material and certification status information that a Verifier needs in order to appraise attestation Evidence produced by a PSA device. This memo defines such PSA Endorsements as a profile of the CoRIM data model.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 12 November 2022.

Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction
2. Conventions and Definitions
3. PSA Endorsements
   - 3.1. PSA Endorsement Profile
   - 3.2. PSA Endorsements to PSA RoT Linkage
   - 3.3. Reference Values
     - 3.3.1. Software Upgrades and Patches
   - 3.4. Attestation Verification Claims
   - 3.5. Certification Claims
   - 3.6. Endorsements Block List
4. Security Considerations
5. IANA Considerations
   - 5.1. CBOR Tag Registrations
   - 5.2. CoRIM Profile Registration
   - 5.3. CoMID Codepoints

Acknowledgements

References
- Normative References
- Informative References

Authors' Addresses

1. Introduction

PSA Endorsements include reference values, cryptographic key material and certification status information that a Verifier needs in order to appraise attestation Evidence produced by a PSA device [PSA-TOKEN]. This memo defines such PSA Endorsements as a profile of the CoRIM data model [CoRIM].

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The reader is assumed to be familiar with the terms defined in Section 2.1 of [PSA-TOKEN] and in Section 4 of [RATS-ARCH].

3. PSA Endorsements

PSA Endorsements describe an attesting device in terms of the hardware and firmware components that make up its PSA Root of Trust (RoT). This includes the identification and expected state of the

device as well as the cryptographic key material needed to verify Evidence signed by the device's PSA RoT. Additionally, PSA Endorsements can include information related to the certification status of the attesting device.

There are five types of PSA Endorsements:

- Reference Values (Section 3.3), i.e., measurements of the PSA RoT firmware;
- Attestation Verification Claims (Section 3.4), i.e., cryptographic keys that can be used to verify signed Evidence produced by the PSA RoT, along with the identifiers that bind the keys to their device instances;
- Certification Claims (Section 3.5), i.e., metadata that describe the certification status associated with a PSA device.
- Software Relations (Section 3.3.1), used to model upgrade and patch relationships between software components;
- Endorsements Block List (Section 3.6), used to invalidate previously provisioned Endorsements.

3.1. PSA Endorsement Profile

PSA Endorsements are carried in one or more CoMIDs inside a CoRIM.

The profile attribute in the CoRIM MUST be present and MUST have a single entry set to the uri http://arm.com/psa/iot/1 as shown in Figure 1.

```
/ corim-map / {
  / corim.profile / 3: [
    32("http://arm.com/psa/iot/1")
  ]
  / ... /
}
```

Figure 1: PSA IoT version 1, CoRIM profile

3.2. PSA Endorsements to PSA RoT Linkage

Each PSA Endorsement - be it a Reference Value, Attestation Verification Claim or Certification Claim - is associated with an immutable PSA RoT. A PSA Endorsement is associated to its PSA RoT by means of the unique PSA RoT identifier known as Implementation ID (see Section 3.2.2 of [PSA-TOKEN]).

In order to support PSA Implementation IDs, the CoMID type $class-id-type-choice is extended as follows:

```
; from draft-tschofenig-rats-psa-token
psa-implementation-id-type = bytes .size 32

tagged-implementation-id-type = #6.600(implementation-id-type)

$class-id-type-choice /= tagged-implementation-id-type
```

Besides, a PSA Endorsement can be associated with a specific instance of a certain PSA RoT - as in the case of Attestation Verification Claims. A PSA Endorsement is associated with a PSA RoT instance by means of the Instance ID (see Section 3.2.1 of [PSA-TOKEN]) and its "parent" Implementation ID.

These identifiers are typically found in the subject of a CoMID triple, encoded in an environment-map as shown in Figure 2.

```
/ environment-map / {
  / comid.class / 0 : {
    / comid.class-id / 0 :
      / tagged-impl-id-type / 600(
        h'61636d652d696d706c656d656e746174
          696f6e2d69642d303030303030303031'
      ),
      / comid.vendor / 1 : "ACME Ltd.",
      / comid.model /  2 : "Roadrunner 1.0"
  },
  / comid.instance / 1 :
    / tagged-ueid-type / 550(
      h'01
        4ca3e4f50bf248c39787020d68ffd05c
        88767751bf2645ca923f57a98becd296'
    )
}
```

Figure 2: Example PSA RoT Identification

Optional vendor and model can be specified as well. Together, they are interpreted as a unique identifier of the product that embeds the PSA RoT. Consistently providing a product identifier is RECOMMENDED.

3.3. Reference Values

Reference Values carry measurements and other metadata associated with the updatable firmware in a PSA RoT. When appraising Evidence, the Verifier compares Reference Values against the values found in

the Software Components of the PSA token (see Section 3.4.1 of [PSA-TOKEN]).

Each measurement is encoded in a measurement-map of a CoMID reference-triple-record. Since a measurement-map can encode one or more measurements, a single reference-triple-record can carry as many measurements as needed, provided they belong to the same PSA RoT identified in the subject of the "reference value" triple. A single reference-triple-record SHALL completely describe the updatable PSA RoT.

The identifier of a measured software component is encoded in a psa-swcomp-id object as follows:

```
psa-swcomp-id = {
  psa.measurement-type => text
  psa.version => text
  psa.signer-id => psa.hash-type
}

psa.hash-type = bytes .size 32 / bytes .size 48 / bytes .size 64

psa.measurement-type = 1
psa.version = 4
psa.signer-id = 5
```

The semantics of the codepoints in the psa-swcomp-id map are equivalent to those in the psa-software-component map defined in Section 3.4.1 of [PSA-TOKEN]. The psa-swcomp-id MUST uniquely identify a given software component within the PSA RoT / product.

In order to support PSA Reference Value identifiers, the CoMID type $measured-element-type-choice is extended as follows:

```
tagged-psa-swcomp-id = #6.601(psa-swcomp-id)

$measured-element-type-choice /= tagged-psa-swcomp-id
```

and automatically bound to the comid.mkey in the measurement-map.

The raw measurement is encoded in a digests-type object in the measurement-values-map. The digests-type array MUST contain at least one entry. The digests-type array MAY contain more than one entry if multiple digests (obtained with different hash algorithms) of the same measured component exist.

The example in Figure 3 shows a CoMID a PSA Endorsement of type Reference Value for a firmware measurement associated with Implementation ID acme-implementation-id-000000001.

```
/ concise-mid-tag / {
  / comid.tag-identity / 1 : {
    / comid.tag-id / 0 : h'3f06af63a93c11e4979700505690773f'
  },
  / comid.triples / 4 : {
    / comid.reference-triples / 0 : [
      [
        / environment-map / {
          / comid.class / 0 : {
            / comid.class-id / 0 :
              / tagged-impl-id-type / 600(
                h'61636d652d696d706c656d656e746174
                  696f6e2d69642d303030303030303031'
              ),
              / comid.vendor / 1 : "ACME Ltd.",
              / comid.model /  2 : "Roadrunner 1.0"
          }
        },
        [
          / measurement-map / {
            / comid.mkey / 0 : 601({
              / psa.measurement-type / 1 : "PRoT",
              / psa.version /          4 : "1.3.5",
              / psa.signer-id /        5 : h'acbb11c7e4da2172
                                           05523ce4ce1a245a
                                           e1a239ae3c6bfd9e
                                           7871f7e5d8bae86b'
            }),
            / comid.mval / 1 : {
              / comid.digests / 2 : [
                / hash-alg-id / 1, / sha256 /
                / hash-value /  h'44aa336af4cb14a8
                                  79432e53dd6571c7
                                  fa9bccafb75f4882
                                  59262d6ea3a4d91b'
              ]
            }
          }
        ]
      ]
    ]
  }
}
```

Figure 3: Example Reference Value

3.3.1. Software Upgrades and Patches

In order to model software lifecycle events such as updates and patches, this profile defines a new triple that conveys the following semantics:

- *SUBJECT: a software component
- *PREDICATE: (non-critically / critically) (updates / patches)
- *OBJECT: another software component

The triple is reified and used as the object of another triple, psa-swrel-triple-record, whose subject is the embedding environment.

```
comid.psa-swrel-triples = 5

$$triples-map-extension //= (
  comid.psa-swrel-triples => [ + psa-swrel-triple-record ]
)

psa.updates = 1
psa.patches = 2

psa-swrel-rel = [
  type: psa.updates / psa.patches
  security-critical: bool ; true means it's a fix for a security bug
]

sw-rel = [
  new: psa-swcomp-id ; identifier of the "new" firmware
  rel: psa-swrel-rel ; patches, updates and the security flag
  old: psa-swcomp-id ; identifier of the "old" firmware
]

psa-swrel-triple-record = [
  environment-map
  sw-rel
]
```

An example of a security critical update involving versions "1.3.5" and "1.4.0" of software component "PRoT" within the target environment associated with Implementation ID acme-implementation-id-000000001 is shown in Figure 4.

```
/ concise-mid-tag / {
  / comid.tag-identity / 1 : {
    / comid.tag-id / 0 : h'3f06af63a93c11e4979700505690773f'
  },
  / comid.triples / 4 : {
    / comid.psa-swrel-triples / 5 : [
      [
        / environment-map / {
          / comid.class-id / 0 :
          / tagged-impl-id-type / 600(
            h'61636d652d696d706c656d656e746174
              696f6e2d69642d303030303030303031'
          ),
          / comid.vendor / 1 : "ACME Ltd.",
          / comid.model /  2 : "Roadrunner 1.0"
        },

        / sw-rel / [
          / new / {
            / psa.measurement-type / 1 : "PRoT",
            / psa.version /          4 : "1.4.0",
            / psa.signer-id /        5 : h'acbb11c7e4da2172
                                           05523ce4ce1a245a
                                           e1a239ae3c6bfd9e
                                           7871f7e5d8bae86b'
          },

          / rel / [
            / type / 1, / psa.updates /
            / security-critical / true
          ],

          / old / {
            / psa.measurement-type / 1 : "PRoT",
            / psa.version /          4 : "1.3.5",
            / psa.signer-id /        5 : h'acbb11c7e4da2172
                                           05523ce4ce1a245a
                                           e1a239ae3c6bfd9e
                                           7871f7e5d8bae86b'
          }
        ]
      ]
    ]
  }
}
```

Figure 4: Example Critical Software Upgrade

3.4. Attestation Verification Claims

An Attestation Verification Claim carries the verification key associated with the Initial Attestation Key (IAK) of a PSA device. When appraising Evidence, the Verifier uses the Implementation ID and Instance ID claims (see Section 3.2) to retrieve the verification key that it SHALL use to check the signature on the Evidence. This allows the Verifier to prove (or disprove) the Attester's claimed identity.

Each verification key is provided alongside the corresponding device Instance and Implementation IDs (and, possibly, a product identifier) in an attest-key-triple-record. Specifically:

* The Instance and Implementation IDs are encoded in the environment-map as shown in Figure 2;

* The IAK public key is carried in the comid.key entry in the verification-key-map. The IAK public key is a PEM-encoded SubjectPublicKeyInfo [RFC5280]. There MUST be only one verification-key-map in an attest-key-triple-record;

* The optional comid.keychain entry MUST NOT be set by a CoMID producer that uses the profile described in this document, and MUST be ignored by a CoMID consumer that is parsing according to this profile.

The example in Figure 5 shows the PSA Endorsement of type Attestation Verification Claim carrying a secp256r1 EC public IAK associated with Instance ID 4ca3...d296.

```
/ concise-mid-tag / {
  / comid.tag-identity / 1 : {
    / comid.tag-id / 0 : h'3f06af63a93c11e4979700505690773f'
  },
  / comid.triples / 4 : {
    / comid.attest-key-triples / 3 : [
      [
        / environment-map / {
          / comid.class / 0 : {
            / comid.class-id / 0 :
              / tagged-impl-id-type / 600(
                h'61636d652d696d706c656d656e746174
                  696f6e2d69642d303030303030303031'
              ),
              / comid.vendor / 1 : "ACME Ltd.",
              / comid.model /  2 : "Roadrunner 1.0"
          },
          / comid.instance / 1 :
            / tagged-ueid-type / 550(
              h'01
                4ca3e4f50bf248c39787020d68ffd05c
                88767751bf2645ca923f57a98becd296'
            )
        },
        [
          / verification-key-map / {
            / comid.key / 0 :
              "MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgA
              ETl4iCZ47zrRbRG0TVf0dw7VFlHtv18HInY
              hnmMNybo+A1wuECyVqrDSmLt4QQzZPBECV8
              ANHS5HgGCCSr7E/Lg=="
          }
        ]
      ]
    ]
  }
}
```

Figure 5: Example Attestation Verification Claim

3.5. Certification Claims

PSA Certified [PSA-CERTIFIED] defines a certification scheme for the PSA ecosystem. A product - either a hardware component, a software component, or an entire device - that is verified to meet the security criteria established by the PSA Certified scheme is warranted a PSA Certified Security Assurance Certificate (SAC). A SAC contains information about the certification of a certain product (e.g., the target system, the attained certification level,

the test lab that conducted the evaluation, etc.), and has a unique Certificate Number.

The linkage between a PSA RoT -- comprising the immutable part as well as zero or more of the mutable components -- and the associated SAC is provided by a Certification Claim, which binds the PSA RoT Implementation ID and the software component identifiers with the SAC unique Certificate Number. When appraising Evidence, the Verifier can use the Certification Claims associated with the identified Attester as ancillary input to the Appraisal Policy, or to enrich the produced Attestation Result.

A Certification Claim is encoded in an psa-cert-triple-record, which extends the $$triples-map-extension socket, as follows:

```
comid.psa-cert-triples = 4

$$triples-map-extension //= (
  comid.psa-cert-triples => [ + psa-cert-triple-record ]
)

psa.immutable-rot = 1
psa.mutable-rot = 2

psa-rot-descriptor = {
  psa.immutable-rot => psa-implementation-id-type
  psa.mutable-rot => [ * psa-swcomp-id ]
}

psa-cert-triple-record = [
  psa-rot-descriptor
  psa-cert-num-type
]

psa-cert-num-type = text .regexp "[0-9]{13} - [0-9]{5}"
```

- The Implementation ID of the immutable PSA RoT to which the SAC applies is encoded as a tagged-impl-id-type in the psa.immutable-rot of the psa-rot-descriptor;

- Any software component that is part of the certified PSA RoT is encoded as a psa-swcomp-id (see Section 3.3) in the psa.mutable-rot of the psa-rot-descriptor;

- The unique SAC Certificate Number is encoded in the psa-cert-num-type.

A single CoMID can carry one or more Certification Claims.

The example in Figure 6 shows a Certification Claim that associates Certificate Number 1234567890123 - 12345 to Implementation ID acme-implementation-id-000000001 and a single "PRoT" software component with version "1.3.5".

```
/ concise-mid-tag / {
  / comid.tag-identity / 1 : {
    / comid.tag-id / 0 : h'3f06af63a93c11e4979700505690773f'
  },
  / comid.triples / 4 : {
    / comid.psa-cert-triples / 4 : [
      [
        / psa-rot-descriptor / {
          / psa.immutable-rot / 1 :
            h'61636d652d696d706c656d656e746174
              696f6e2d69642d303030303030303031',
          / psa.mutable-rot / 2 : [
            / psa-swcomp-id / {
              / psa.measurement-type / 1 : "PRoT",
              / psa.version /          4 : "1.3.5",
              / psa.signer-id /        5 : h'acbb11c7e4da2172
                                             05523ce4ce1a245a
                                             e1a239ae3c6bfd9e
                                             7871f7e5d8bae86b'
            }
          ]
        },
        / psa-cert-num-type / "1234567890123 - 12345"
      ]
    ]
  }
}
```

Figure 6: Example Certification Claim with `supplement` Link-Relation

3.6. Endorsements Block List

This is work in progress. It may change or be removed in the future.

The following three "blocklist" claims:

- *reference-blocklist-triple
- *attest-key-blocklist-triple
- *cert-blocklist-triple

are defined with the same syntax but opposite semantics with regards to their "positive" counterparts to allow invalidating previously provisioned endorsements from the acceptable set.

4. Security Considerations

TODO

5. IANA Considerations

5.1. CBOR Tag Registrations

IANA is requested to allocate the following tag in the "CBOR Tags" registry [IANA.cbor-tags], preferably with the specified value:

Tag	Data Item	Semantics
600	tagged bytes	PSA Implementation ID (Section 3.2 of RFCTHIS)
601	tagged map	PSA Software Component Identifier (Section 3.3 of RFCTHIS)

Table 1: CoRIM CBOR Tags

5.2. CoRIM Profile Registration

IANA is requested to register the following profile value in the TODO CoRIM registry.

Profile Value	Type	Semantics
http://arm.com/psa/iot/1	uri	The CoRIM profile specified by this document

Table 2: PSA profile for CoRIM

5.3. CoMID Codepoints

IANA is requested to register the following codepoints to the "CoMID Triples Map" registry.

Index	Item Name	Specification
4	comid.psa-cert-triples	RFCTHIS
5	comid.psa-swrel-triples	RFCTHIS

Table 3: PSA CoMID Triples

Acknowledgements

TODO

References

Normative References

[CoRIM] Birkholz, H., Fossati, T., Deshpande, Y., Smith, N., and W. Pan, "Concise Reference Integrity Manifest", Work in Progress, Internet-Draft, draft-birkholz-rats-corim-02, 26 January 2022, <https://www.ietf.org/archive/id/draft-birkholz-rats-corim-02.txt>.

[IANA.cbor-tags] IANA, "Concise Binary Object Representation (CBOR) Tags", <https://www.iana.org/assignments/cbor-tags>.

[PSA-TOKEN] Tschofenig, H., Frost, S., Brossard, M., Shaw, A., and T. Fossati, "Arm's Platform Security Architecture (PSA) Attestation Token", Work in Progress, Internet-Draft, draft-tschofenig-rats-psa-token-09, 7 March 2022, <https://www.ietf.org/archive/id/draft-tschofenig-rats-psa-token-09.txt>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

Informative References

[PSA-CERTIFIED] "PSA Certified", 2021, <https://www.psacertified.org>.

[RATS-ARCH] Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote Attestation Procedures Architecture", Work in Progress, Internet-Draft, draft-ietf-rats-architecture-15, 8 February 2022, <https://www.ietf.org/archive/id/draft-ietf-rats-architecture-15.txt>.

Authors' Addresses

Thomas Fossati
Arm Ltd

Email: thomas.fossati@arm.com

Yogesh Deshpande
Arm Ltd

Email: yogesh.deshpande@arm.com

Henk Birkholz
Fraunhofer SIT

Email: henk.birkholz@sit.fraunhofer.de