IETF 113 TEEP Hackathon

March 21, 2022

Akira Tsukamoto (AIST)

Objective and Plan

- Objective
  - Find issues or consideration points when adding EAT and COSE in TEEP protocol implementation

- Plan
  - Try out SUIT handling in the implementation
  - EAT
    - EAT has “Passport Model" and the "Background-Check Model”
    - Pros and cons of above two in TEEP
  - COSE
    - CDDL format update
    - Which algorithms to be mandated or optional

Two EAT models from the EAT draft

- https://www.ietf.org/archive/id/draft-ietf-rats-architecture-01.html

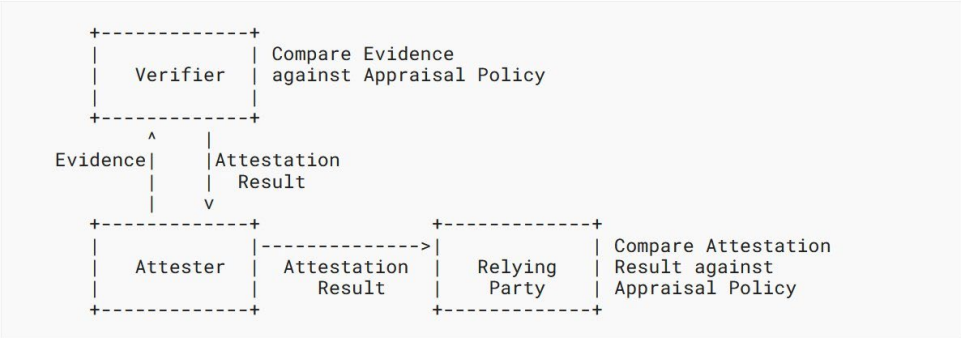


Figure 3: Passport Model

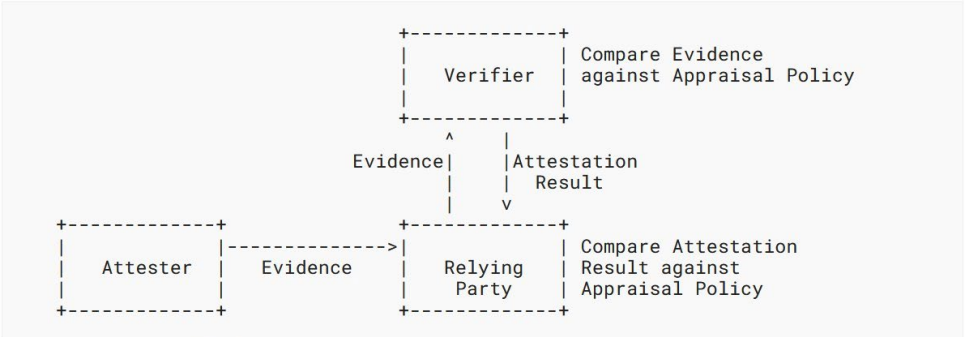
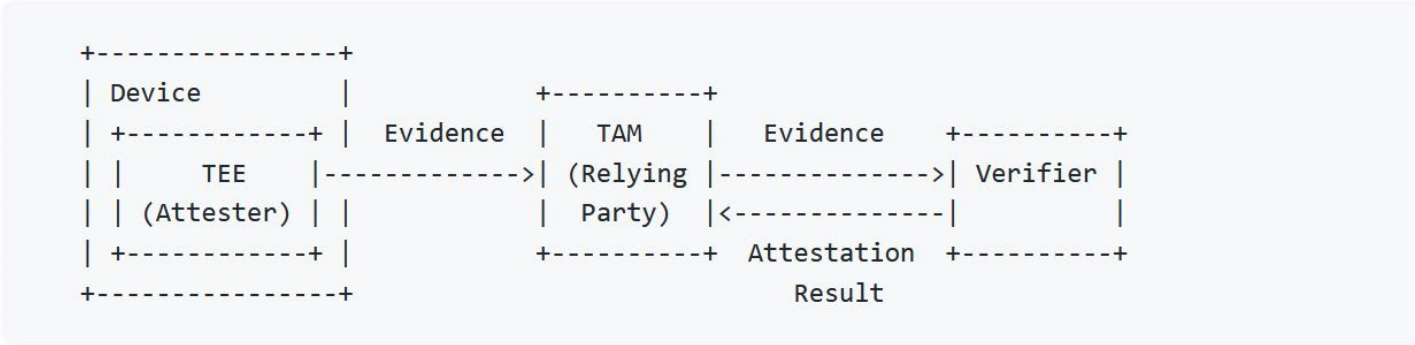


Figure 4: Background-Check Model

EAT in TEEP drafts

- TEEP is using background-check model
  - https://github.com/ietf-teep/architecture/blob/master/draft-ietf-teep-architecture.md



- TAM will be attesting with Verifier and not the Device
  - The Device does not have to have an implementation of talking with Verifier
  - Desirable for constraint devices
- How to support both in the future?
  - Maybe possible, if the EAT profile format has indication between Evidence or Attestation-Result
- Any reference implementation of Verifier to be used with TAM is helpful

EAT formats in CBOR and JSON

- https://github.com/ietf-teep/teep-protocol/issues/184
- EAT permits both JSON and CBOR
- Current description in the TEEP Protocol draft

```
query-response = [
    type: TEEP-TYPE-query-response,
    options: {
        ? evidence-format => text,
        ? evidence => bstr,
```

- Suggestion for the text string in evidence-format

  "application/jwt" MUST be present if the EAT format is JWT, and
  "application/cwt" MUST be present if the EAT format is CWT.

- Add types of evidence which could have bstr for CBOR and text for JSON

```
evidence = [
    bstr / text / nil
]
```

Selection of REQUIRED, RECOMMENDED, OPTIONAL in COSE

- https://github.com/ietf-teep/teep-protocol/issues/182
- https://github.com/ietf-teep/teep-protocol/issues/183
- Choices for implementing on real products
  - Preferred to match with SUIT and EAT requirements, since both will be using COSE
  - Recommending HSS-LMS for quantum resistance purpose
  - Include AES, RSA as options which are commonly used in the market, SHA-3 in the future?

```
;REQUIRED to implement:
   teep-cose-hash-algs /= cose-alg-sha-256    required in SUIT also

;REQUIRED to implement either of one:
   teep-cose-sign-algs /= cose-alg-es256
   teep-cose-sign-algs /= cose-alg-eddsa

;RECOMMENDED to implement:
   teep-cose-sign-algs /= cose-alg-hss-lms
```

```
;OPTIONAL to implement:
   teep-cose-hash-algs /= cose-alg-shake128
   teep-cose-hash-algs /= cose-alg-sha-384
   teep-cose-hash-algs /= cose-alg-sha-512
   teep-cose-hash-algs /= cose-alg-shake256
   teep-cose-sign-algs /= cose-alg-ps256
   teep-cose-sign-algs /= cose-alg-ps384
   teep-cose-sign-algs /= cose-alg-ps512
   teep-cose-sign-algs /= cose-alg-rsaes-oaep-sha256
   teep-cose-sign-algs /= cose-alg-rsaes-oaep-sha512
   teep-cose-encrypt-algs /= cose-alg-accm-16-64-128
   teep-cose-encrypt-algs /= cose-alg-a128gcm
   teep-cose-encrypt-algs /= cose-alg-a192gcm
   teep-cose-encrypt-algs /= cose-alg-a256gcm
   teep-cose-mac-algs /= cose-alg-hmac-256
```

RSA require certain key length

- How to specify RSA key length, along side with RSA in COSE?
  - Over 2048 bits required
  - Over 4096 bits recommended

```
teep-cose-sign-algs /= cose-alg-ps256
teep-cose-sign-algs /= cose-alg-ps384
teep-cose-sign-algs /= cose-alg-ps512
teep-cose-sign-algs /= cose-alg-rsaes-oaep-sha256
teep-cose-sign-algs /= cose-alg-rsaes-oaep-sha512
```

- Current IANA registration of COSE Algorithms
  - https://www.iana.org/assignments/cose/cose.xhtml

Summary

- Going through SUIT, EAT and COSE support on TEEP which were added after IETF 112
  - Plan to make the implementation public before IETF 114 with at least of SUIT support
- EAT background-check model and passport model for the TEEP
- Pros and cons, should make the draft generic to support both in the future?
- What cipher suite for COSE?
- Permitting RSA with a certain key length
- After 113, runnable code for COSE and EAT

A part of this hackathon presentation is based on results obtained from a project, JPNP16007,commissioned by the New Energy and Industrial Technology Development Organization (NEDO).