Remote attestation over EDHOC
draft-song-lake-ra-00

Yuxuan SONG
Inria
Remote Attestation

Remote attestation is a security service to verify and confirm the integrity and trustworthiness of a remote device or system.

- **Evidence**: a set of Claims to demonstrate the integrity and security properties of its software or hardware.
- **Attestation result**: the output after evaluating the validity of Evidence
- **Relying Party**: the entity who consumes the Attestation result to reliably apply application-specific actions
IETF RATS architecture for Remote Attestation

RATS: Remote ATtestation procedureS [1]

- Background-check model

- Passport model

Just released version -00: Protocol

- Remote attestation procedure using EDHOC's EAD
Mapping to EDHOC

Initiator

EAD_1 = Attestation proposal

EAD_2 = Attestation request

EAD_3 = Evidence

Responder

Discussion point:
- Criticality of EAD labels?
- Initial idea:
  - EAD_1 is not critical
  - EAD_2 is critical
  - EAD_3 is critical
EAD_1

Attestation_proposal = bstr .cbor Proposed_EvidenceType

Proposed_EvidenceType = (content-format: [+ uint])

• **EAD_1** contains an array that contains all the supported evidence types by the Attester in decreasing order of preference.

• evidence type is indicated by `content-format`, which is an ID of the format type, from "CoAP Content-Formats" registry[1]
  - e.g., application/eat+cwt with an appropriate eat_profile parameter set
  - an example for PSA token is application/eat+cwt;
    eat_profile = "tag:psacertified.org,2023:psa#tfm" [2]

[1] https://www.iana.org/assignments/core-parameters/core-parameters.xhtml#content-formats
**EAD_2**

```python
Attestation_request = bstr .cbor Selected_EvidenceType
Selected_EvidenceType = (
    content-format:uint,
    nonce:bstr
)
```

- **content-format** is the selected evidence type by the Relying Party and supported by the Verifier

- **nonce** is 8 to 64 bytes long, generated by the Verifier and forwarded by the Relying Party
**EAD_3**

- **EAD_3** is a serialized EAT [1]

- Entity Attestation Token (EAT) provides an attested claims set that describes state and characteristics of an entity.

  - **eat_profile** claim indicates a EAT profile. It constrains the parameters that producers and consumers of a specific EAT need to understand.
    - e.g., the number and type of claims, the supported signature schemes, etc.
    - The value of the **eat_profile** is either an OID or URI.

- Verifier consumes the EAT and produces **attestation result** according to its appraisal policy.
Conclusion

Do remote attestation over EDHOC:

• Remote attestation procedure based on RATS architecture.
• Initial focus on background-check model.
• Definition of the EADs in three EDHOC messages.
Next steps

• Complete the TBDs in the draft (examples)
  • question from last LAKE interim meeting: Size estimates for EAD_3?

```
[psa-best-seed]/
/psa-nonce/  2397: h='a953a2d3a5a5a67a7a8a9a8b9a9caaeaf6c8b12b3b5b5c7b7b9ba9bcb9dhebf',
/psa-client-id/  18: h='13894f078190c7e7',
/psa-certificate-reference/
/psa-implementation-id/  394: 70e02,
/hdl/  398: "8604955272829-10018",
/iat/  396: h='aaaaaaaaa6b777h889b879f89c18b5c9f959f4d',
/iat-profile/  398: "6fa875f5f65627c7c5e48f92975296732d8ca7ad5e29b4d286e9cfc844818",  
/psa-security-lifecycle/
/psa-software-components/  266: 66,
/psa-verification-service-indicator/  2395: 11288,

2399: {
    /measurement-desc:
    6: "SHA256",
    /measurement-value:
    2: h='a953a2d3a5a5a67a7a8a9a8b9a9caaeaf6c8b12b3b5b5c7b7b9ba9bcb9dhebf',
    /signer-id/  398: "6fa875f5f65627c7c5e48f92975296732d8ca7ad5e29b4d286e9cfc844818",  
    /measurement-type/  1: "SPE",
    /version/  4: "1.6.0",
    },

6: "SHA256",
2: h='687d1a5b62a3b668c9c6a235d9d428976c316b8b5c799c5088522c1bf',
5: h='b150ca15c98c6b71a88827a1m4b3f7b16e99e9f0edfdeaa37c145ed1fccc809',
1: "NORM",
4: "8.0.0",
}
```

Figure: PSA token example generated by iat-verifier tool [1]

• the COSE Sign1 PSA token size is 453 bytes
• Preliminary implementation
• Any reviews, feedback and comments are greatly appreciated 😊
Thank you!

Yuxuan.song@inria.fr

https://github.com/ysong02/draft-song-lake-ra