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

Yuxuan SONG

Inria

Remote Attestation

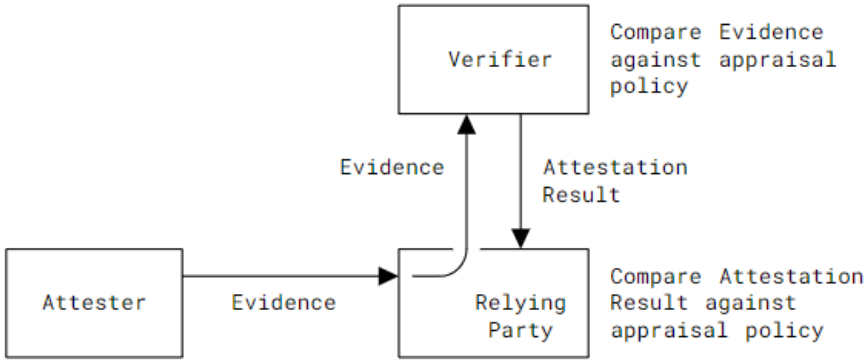
Remote attestation is a security service to verify and confirm the integrity and trustworthiness of a remote device or system.

- 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

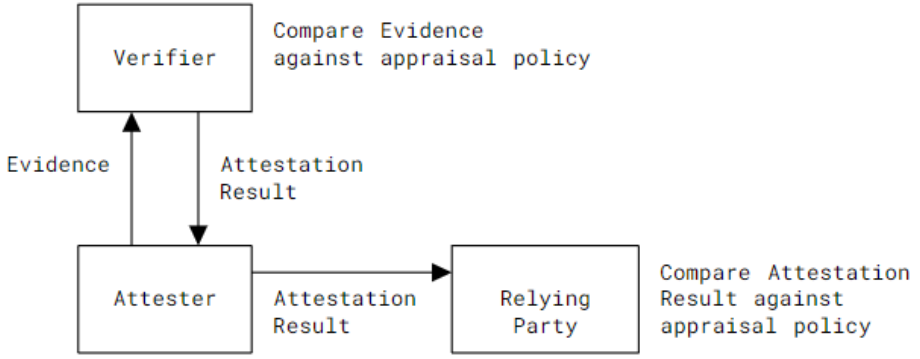
IETF RATS architecture for Remote Attestation

RATS: Remote ATtestation procedureS [1]

- Background-check model



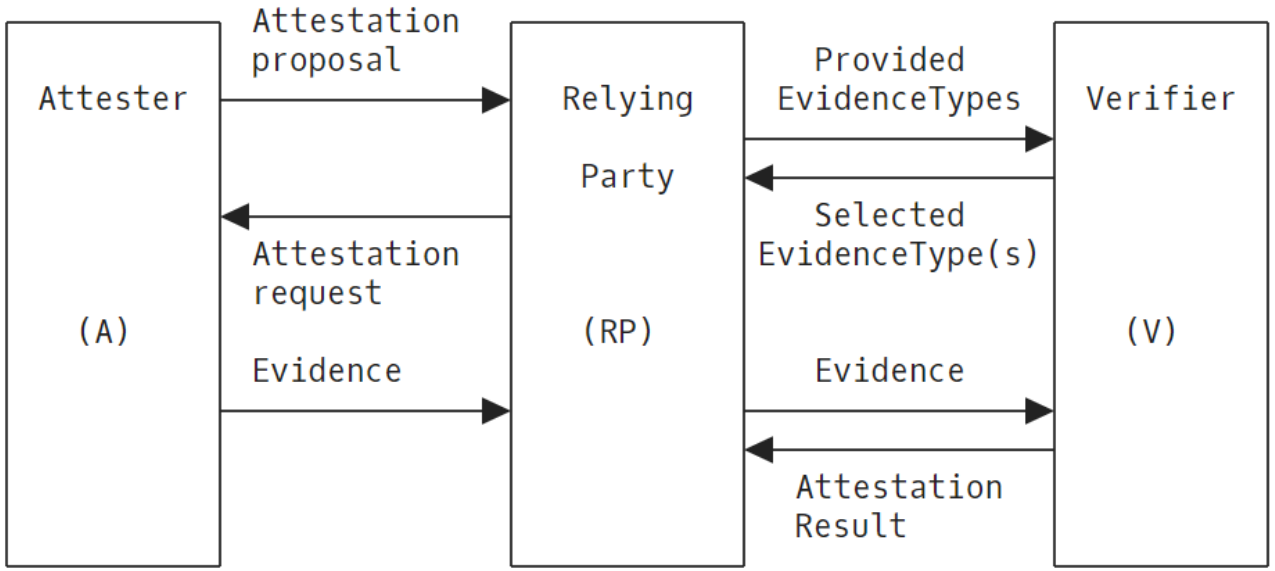
- Passport model



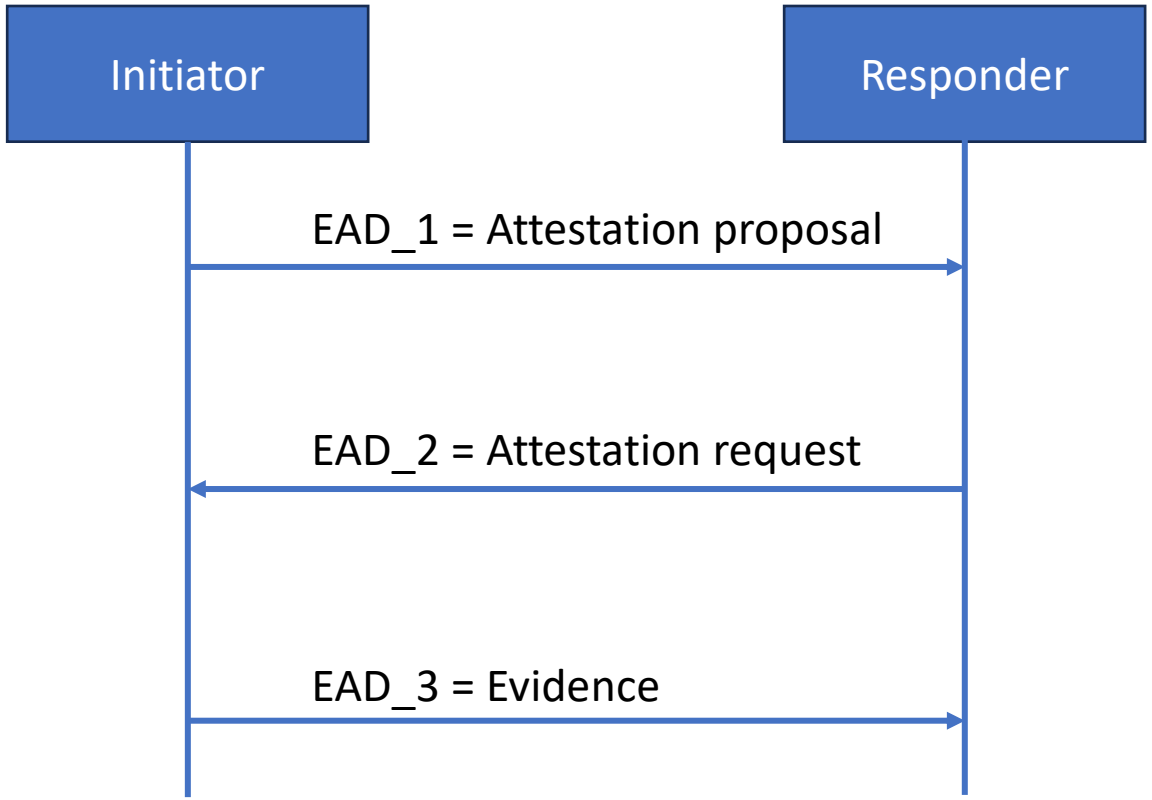
[1]https://www.rfc-editor.org/rfc/rfc9334.html#name-background-check-model

Just released version -00: Protocol

- Remote attestation procedure using EDHOC's EAD



Mapping to EDHOC



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
[2] https://datatracker.ietf.org/doc/draft-tschofenig-rats-psa-token/

EAD_2

```
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.

[1]https://datatracker.ietf.org/doc/draft-ietf-rats-eat/

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-boot-seed/                    2397: h'a0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebf',
  /eat_nonce/                        10: h'1385b9708109c7fb',
  /psa-client-id/                    2394: 3002,
  /psa-certificate-reference/        2398: "0604565272829-10010",
  /psa-implementation-id/            2396: h'aaaaaaaaabbbbbbbbbbbbcccccccccccccccddddddddddddddd',
  /ueid/                             256: h'01fa58755f658627ce5460f29b75296713248cae7ad9e2984b90280efcbcb50248',
  /eat_profile/                      265: 66,
  /psa-security-lifecycle/           2395: 12288,
  /psa-software-components/          2399: [
                                       {
                                     /measurement-desc/  6: "SHA256",
                                     /measurement-value/ 2: h'e33ea1e002d2fe794d1a1679db58bb6a23a8f659bb77f89c458cecf9d5995ffd',
                                     /signer-id/         5: h'bfe6d86f8826f4ff97fb96c4e6fbc4993e4619fc565da26adf34c329489adc38',
                                     /measurement-type/  1: "SPE",
                                     /version/           4: "1.6.0",
                                          },
                                          {
                                                         6: "SHA256",
                                                         2: h'087d13c68f32aaafb8c4fc0a2253445432009765e216fb85c398c9580522c1bf',
                                                         5: h'b360caf5c98c6b942a4882fa9d4823efb166a9ef6a6e4aa37c1919ed1fccc049',
                                                         1: "NSPE",
                                                         4: "0.0.0",
                                          },
                                       ],
  /psa-verification-service-indicator/ 2400: "www.trustedfirmware.org",
}
```

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 ☺

[1]https://git.trustedfirmware.org/TF-M/tf-m-tools.git/tree/iat-verifier

Thank you!

Yuxuan.song@inria.fr

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