

# Remote attestation over EDHOC draft-song-lake-ra-01

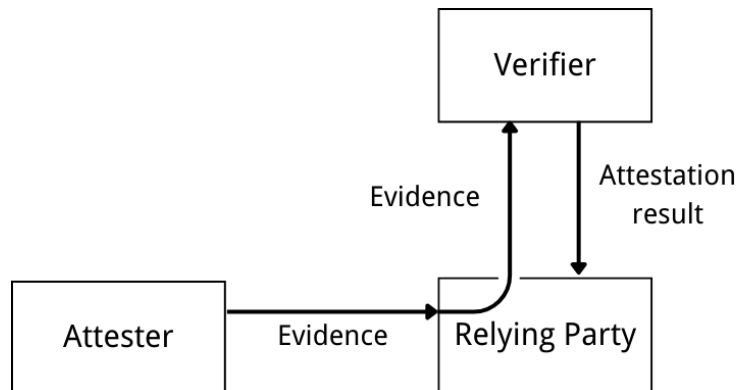
Yuxuan SONG

Inria

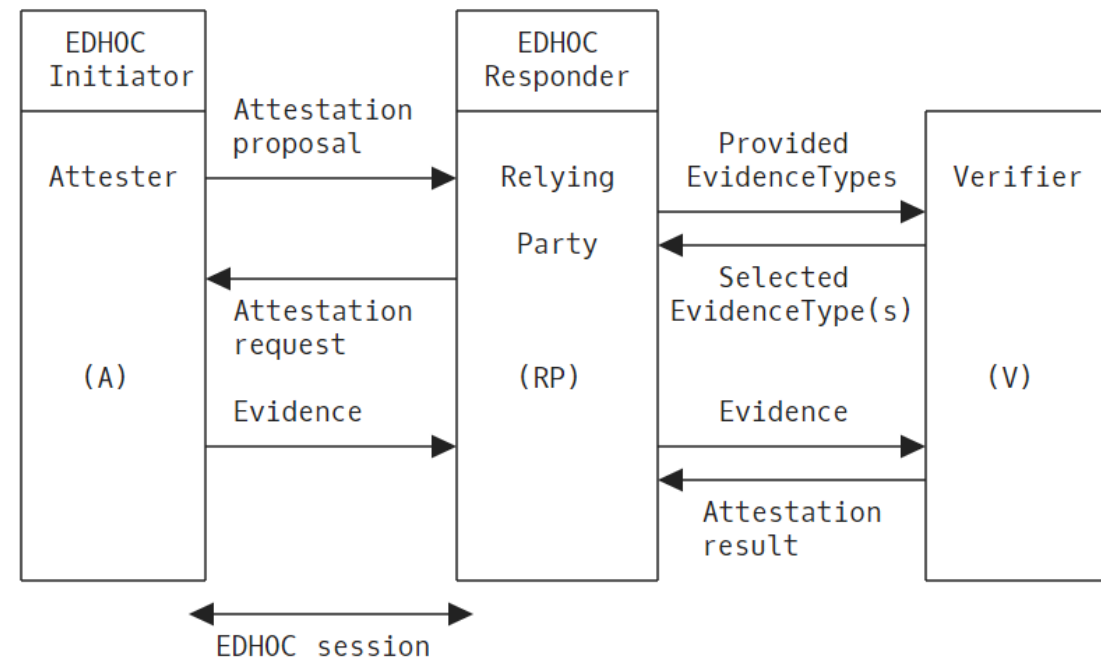
# Recap: version-00

- Forward attestation in Background-check model
  - Use case: An IoT device remotely attests its firmware version to a server for network access.
- Three EAD items: Attestation\_proposal, Attestation\_request, Evidence

- RATS Background-check model  
RATS: Remote ATtestation procedures [1]



- message flow



# Changelog: draft-song-lake-ra-01

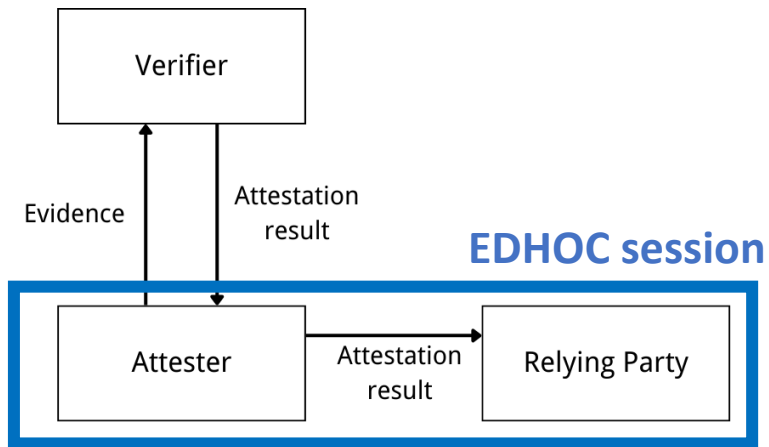
- 1. Introduction
- 2. Conventions and Definitions
- 3. Problem Description
- 4. Assumptions
- 5. The Protocol
  - 5.1. Overview
  - 5.2. External Authorization Data 1
  - 5.3. External Authorization Data 2
  - 5.4. External Authorization Data 3
- 6. Security Considerations
- 7. IANA Considerations
  - 7.1. EDHOC External Authorization Data Registry
- 8. References
  - 8.1. Normative References
  - 8.2. Informative References
- Appendix A. Example: Remote Attestation Flow
- Appendix B. Example: Firmware Version
- Acknowledgments
- 23/07/2024
- Author's Address



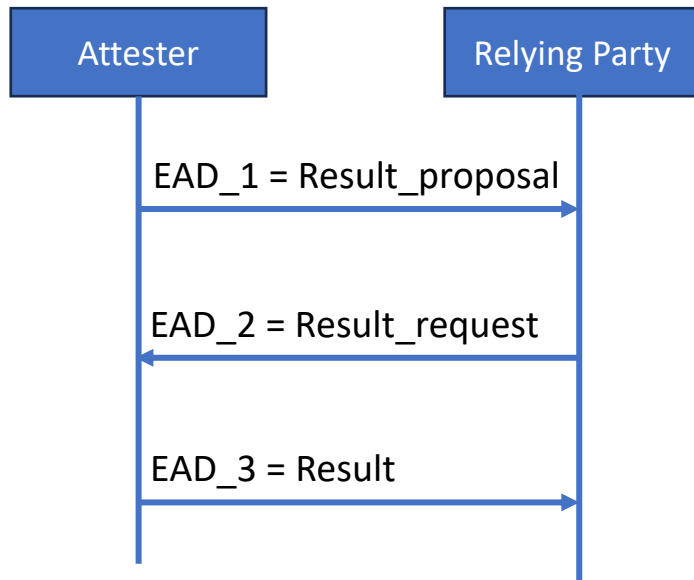
- Different message flows
  - forward attestation
  - reverse attestation (new)
  - mutual attestation (new)
- Consider two RATS architectures
  - Background-check model
  - Passport model (new)
- **New** EAD items for passport model
- **New** EDHOC Error "Attestation failed"

- 1. Introduction
- 2. Conventions and Definitions
- 3. Problem Description
- 4. Assumptions
- 5. The Protocol
  - 5.1. Forward remote attestation
    - 5.1.1. Background-check model
  - 5.2. Reverse attestation
    - 5.2.1. Background-check model
    - 5.2.2. Passport model
  - 5.3. Mutual attestation
    - 5.3.1. Background-check model -- Background-check model
    - 5.3.2. Background-check model -- Passport model
  - 5.4. External Authorization Data (EAD) items
    - 5.4.1. Attestation\_proposal
    - 5.4.2. Attestation\_request
    - 5.4.3. Evidence
    - 5.4.4. Result\_proposal
    - 5.4.5. Result\_request
    - 5.4.6. Result
- 6. Error Handling
  - 6.1. EDHOC Error "Attestation failed"
- 7. Security Considerations
- 8. IANA Considerations
  - 8.1. EDHOC External Authorization Data Registry

# Attestation in passport model



## mapping to EDHOC:



## Process over EDHOC:

The Attester proposes Verifier identities from which it can relay the attestation result.

The Relying Party selects a trusted Verifier identity in common and gets the attestation result.

## EAD items:

### Result\_proposal

```
Result_proposal = bstr .cbor Proposed_VerfierIdentity
Proposed_VerfierIdentity = [ + VerifierIdentity ]
```

```
VerifierIdentity = {
  label => values
}
```

similar to ID\_CRED, from COSE Header Param. registry

### Result\_request

```
Result_request = bstr .cbor Request_structure
```

```
Request_structure = {
  selected_verifier: VerifierIdentity
}
```

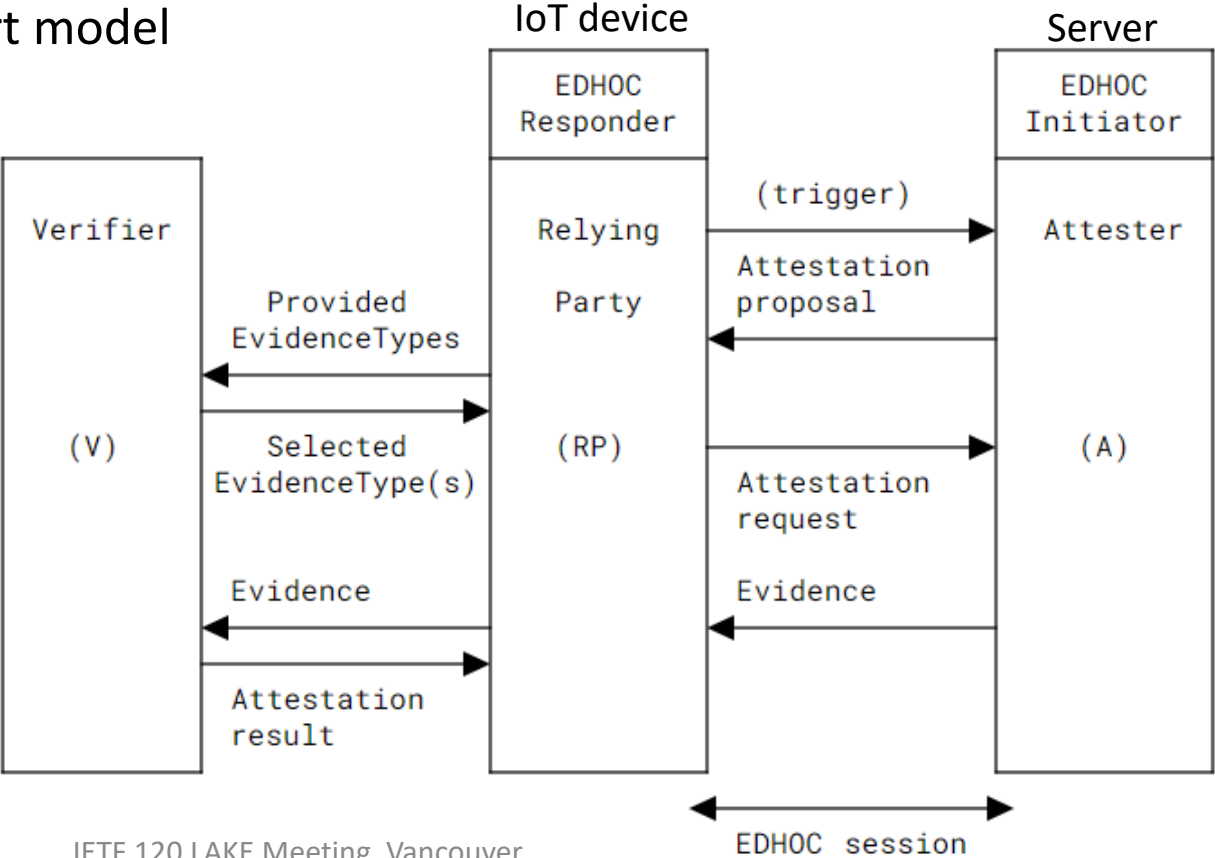
Result is a serialized EAT (Entity Attestation Token).

# Reverse attestation

- Use case: A server attests remotely to gain the device's trust and retrieve its sensitive data.
- over reverse EDHOC message flow

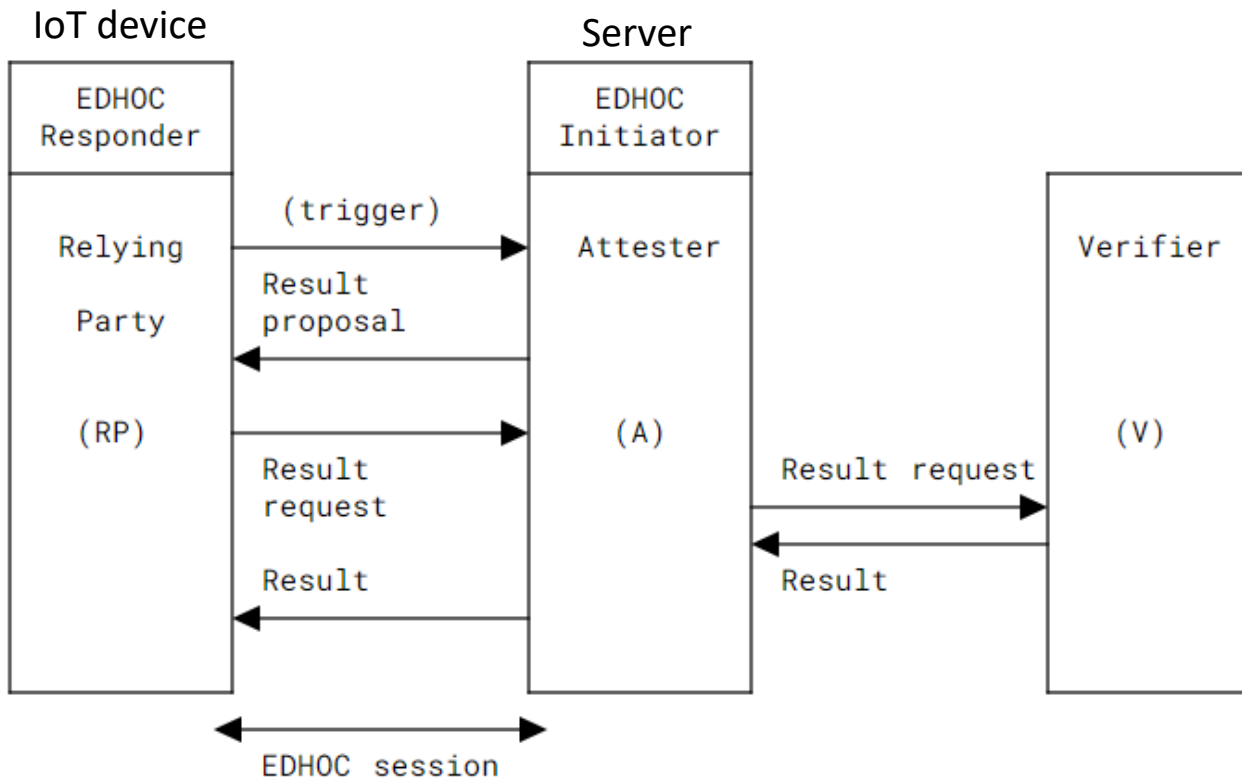
**Seeking input from the WG:** the feasibility of both the background-check model and the passport model

In background-check model:



# Reverse attestation

In passport model:



## Seeking input from the WG:

What kind of freshness is expected for the attestation result?

1. Fresh attestation result with nonce

a new remote attestation is performed after receiving Result request

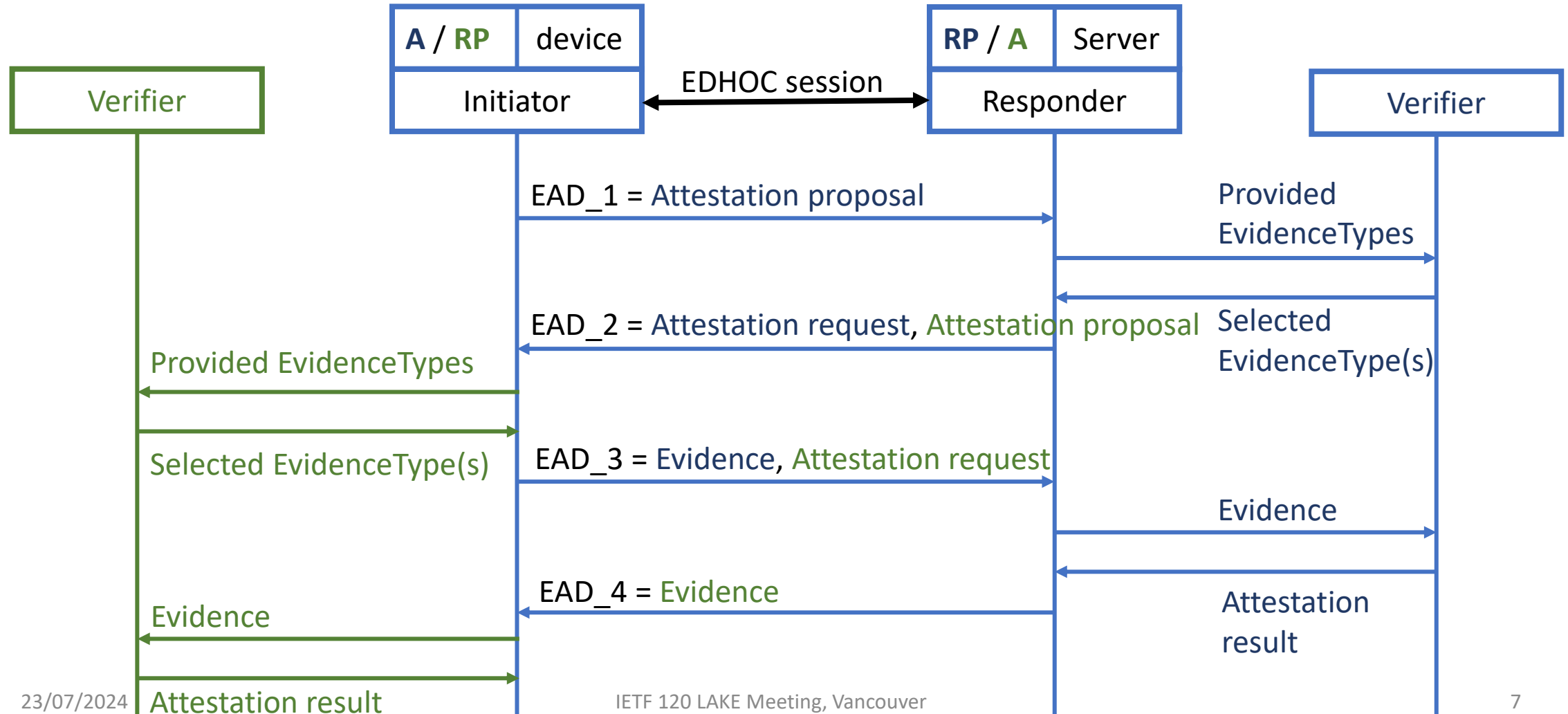
2. Pre-stored attestation result with timestamp

the attestation result is stored at the Verifier with a timestamp indicating the generation time

# Mutual attestation:

Background-check — **Background-check** (seeking input from the WG on feasibility)

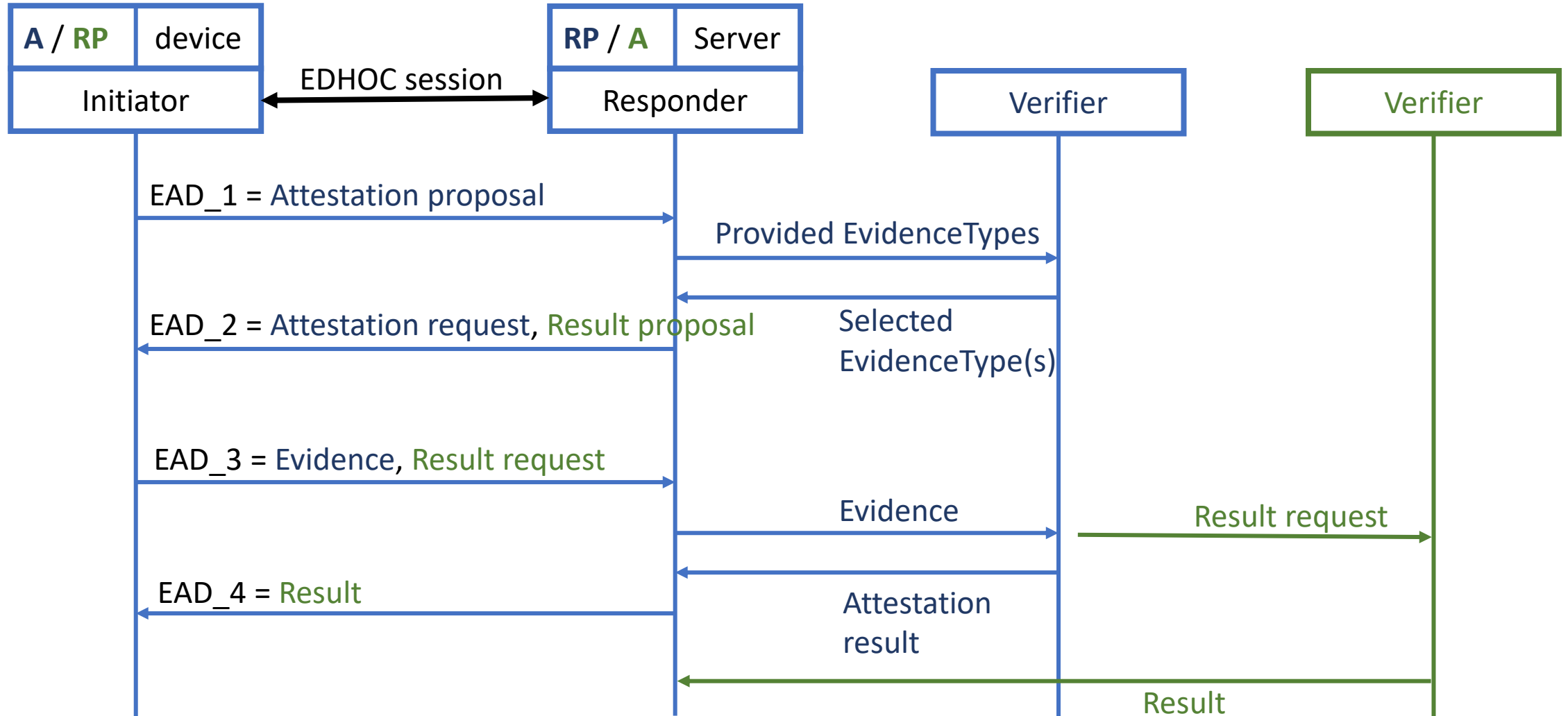
- for devices with no connectivity constraints



# Mutual attestation:

Background-check — **passport** (seeking input from the WG on feasibility)

- for constrained devices with connectivity problem





# New error: Attestation failed

ERR_CODE	ERR_INFO Type	Description
TBD7	attestation	Attestation failed

To indicate to the receiver that the remote attestation failed after the evidence is sent.

The error will be sent in two cases:

1. Verifier evaluates the evidence and generates a negative attestation result
2. Relying Party cannot establish a sufficient level of trust to proceed with decision-specific actions.

Relying Party generates the error, the application layer decides how to handle the error message.

# Conclusion

- Defined three attestation flows
  - forward attestation
  - reverse attestation
  - mutual attestation
- Applied on two RATS architectures
  - Background-check model
  - Passport model
- EDHOC error: Attestation failed
- Looking for inputs from the WG
  - freshness type of attestation result, reverse attestation, mutual attestation, or any other advice.

# Thank you!

Open for more discussions and collaborations: [yuxuan.song@inria.fr](mailto:yuxuan.song@inria.fr)

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

Welcome any comments and advice 😊