Lightweight remote attestation over EDHOC for constrained IoT devices

RIOT Summit 2024

Yuxuan SONG
Inria Paris, Team AIO

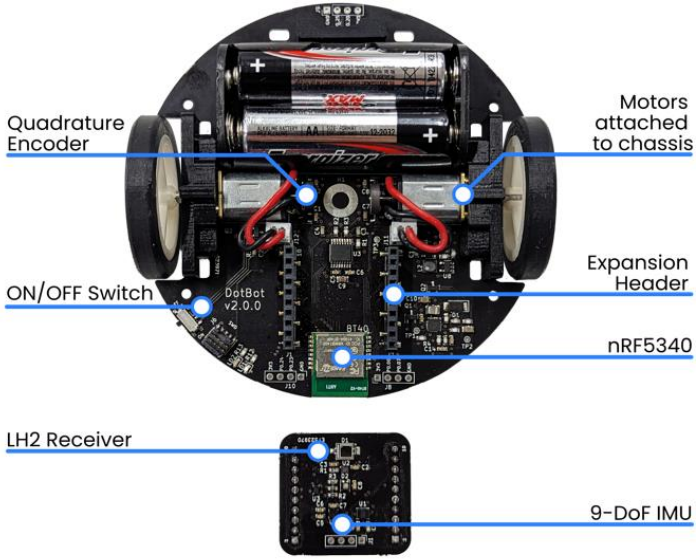
Outline

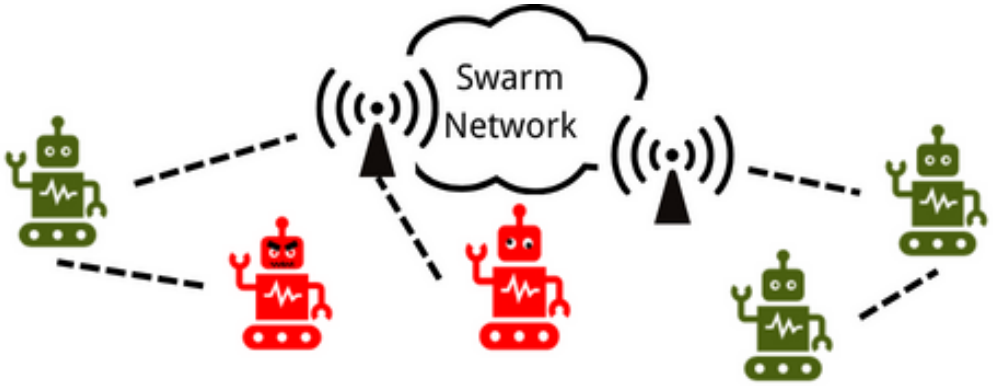
01. Context
02. Remote attestation
03. EDHOC: Ephemeral Diffie-Hellman Over COSE
04. Ongoing work at IETF: remote attestation over EDHOC
05. Implementation

01

V 1

V 2





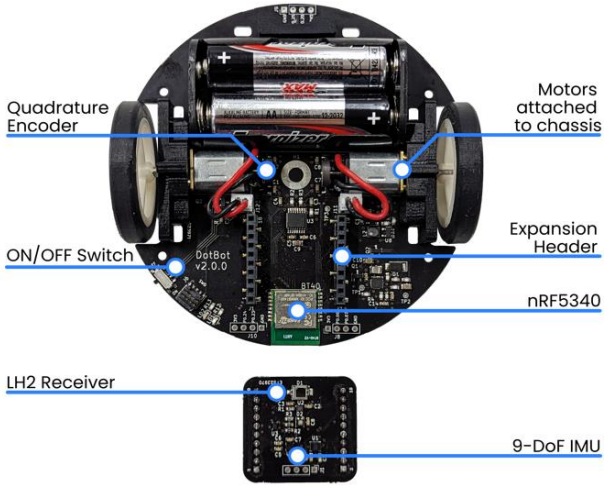
- DotBot platform
- Swarms of DotBots

Context: The DotBot Platform

- Low material costs (<35 EUR)
- Low-power wireless communication capabilities
- Supports indoor low-power localization
- Motor driver circuit
- Operates on 2x AA batteries

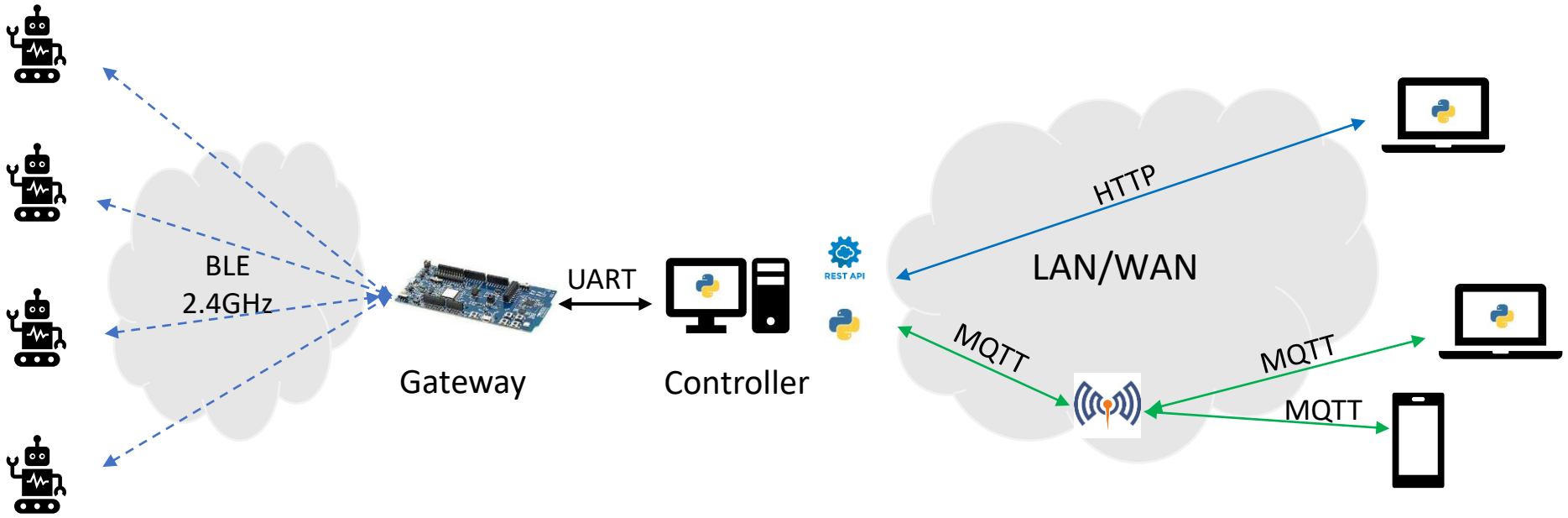
V 1

V 2



Context: Swarms of DotBots

A ‘‘swarm ’’ : A large number of entities that operate collectively in a coordinated manner.



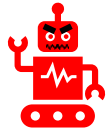
Swarm of DotBots: Coordinated group of DotBots, each DotBot uses BLE radio to communicate with the gateway. Gateway relays commands from the user.

Problem Statement

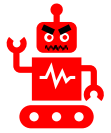
- How to ensure that ***ONLY*** robots with
  verified and trustworthy software and hardware configurations
  are allowed to join the swarm ?

✓ Up-to-date firmware version

✗ Old firmware version



✗ Compromised firmware version



✗ Tampered firmware version

02

- Remote Attestation
- RATS architecture
- Ongoing work at IETF: attested TLS

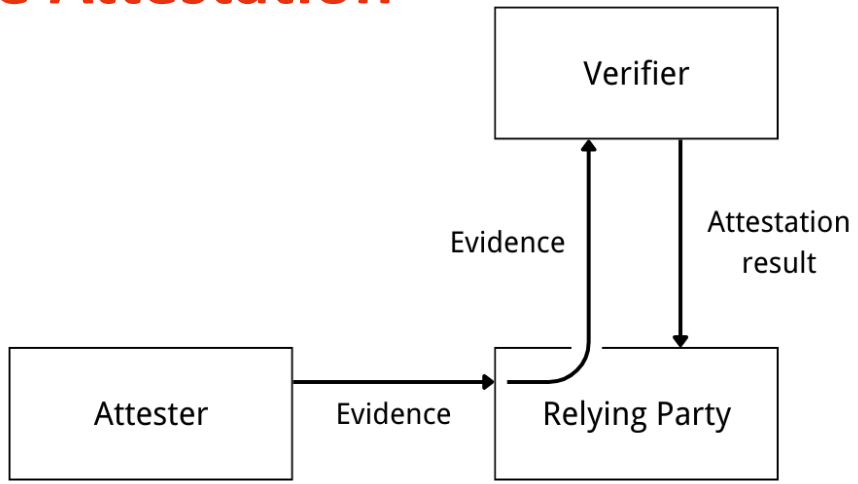
Remote Attestation

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

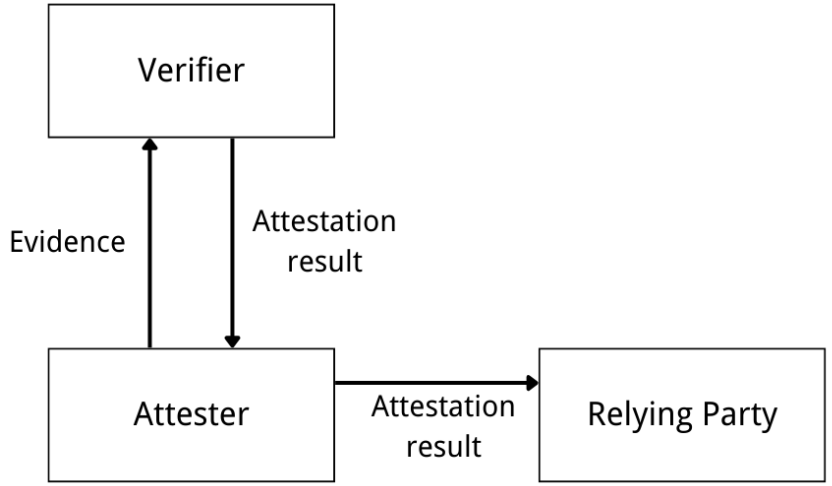
- Evidence: a set of Claims to demonstrate the integrity and security properties of its software or hardware.
- Attestation result: the output after evaluating the validity of Evidence
- Relying Party: the entity who consumes the Attestation result to reliably apply application-specific actions

IETF RATS architecture for Remote Attestation

- Background-check model



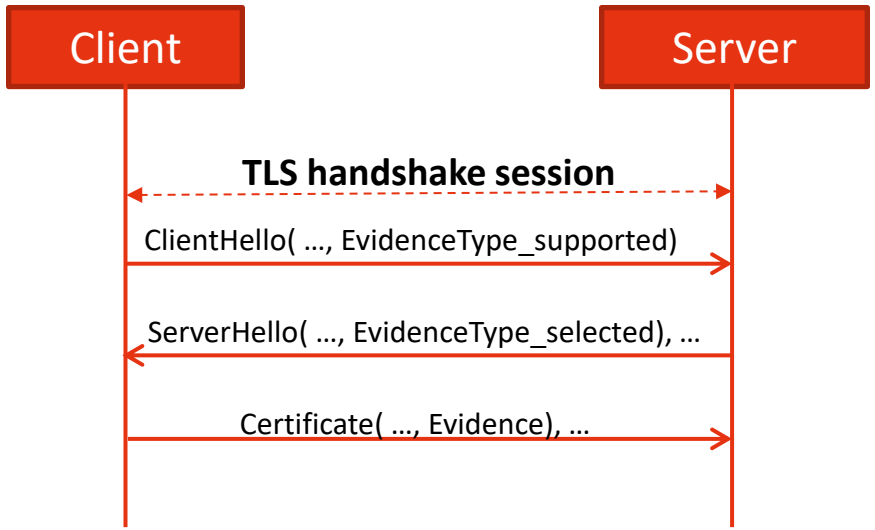
- Passport model



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

Ongoing work at IETF: attested TLS

- Remote attestation in TLS (Transport Layer Security)
  - Evidence and Attestation results are allocated as new TLS extensions
  - Remote attestation is done alongside with TLS handshake sessions



Limitation: High memory requirements and high energy consumption of TLS for low-power constrained networks[1].

[1]Fedrecheski, Geovane, Mališa Vučinić, and Thomas Watteyne. "Performance Comparison of EDHOC and DTLS 1.3 in Internet-of-Things Environments." IEEE Wireless Communications and Networking Conference. 2024.

03

RFC 9528
Ephemeral Diffie-Hellman Over COSE (EDHOC)

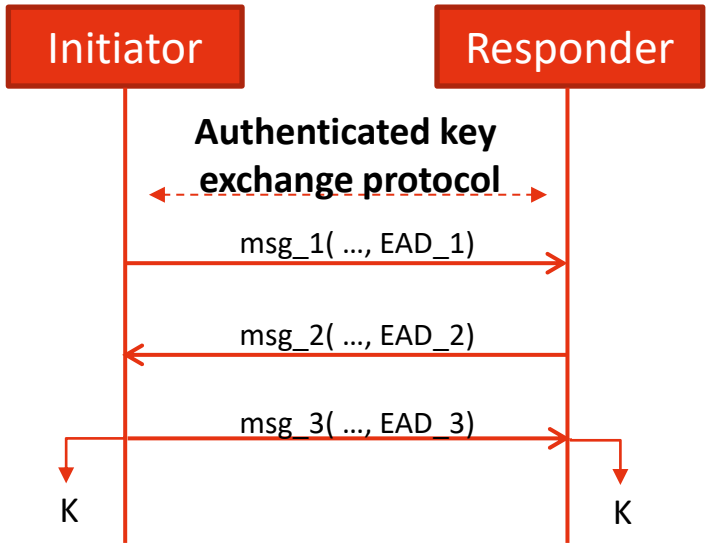
Abstract

This document specifies Ephemeral Diffie-Hellman Over COSE (EDHOC), a very compact and lightweight authenticated Diffie-Hellman key exchange with ephemeral keys. EDHOC provides mutual authentication, forward secrecy, and identity protection. EDHOC is intended for usage in constrained scenarios, and a main use case is to establish an Object Security for Constrained RESTful Environments (OSCORE) security context. By reusing CBOR Object Signing and Encryption (COSE) for cryptography, Concise Binary Object Representation (CBOR) for encoding, and Constrained Application Protocol (CoAP) for transport, the additional code size can be kept very low.

EDHOC: Ephemeral Diffie-Hellman Over COSE

EDHOC: Ephemeral Diffie-Hellman Over COSE

- Standardized by the IETF LAKE working group
- A lightweight authenticated key exchange protocol [1]



- Two entities: Initiator and Responder
- 3 messages, 2 round trips
- EAD: External Authorization Data

Flight	#1	#2	#3	Total
DTLS 1.3 – RPKs, ECDHE	185	454	255	894
DTLS 1.3 – Compressed RPKs, ECDHE	185	422	223	830
DTLS 1.3 – Cached RPK, PRK, ECDHE	224	402	255	881
DTLS 1.3 – Cached X.509, RPK, ECDHE	218	396	255	869
DTLS 1.3 – PSK, ECDHE	219	226	56	501
DTLS 1.3 – PSK	136	153	56	345
EDHOC – X.509s, Signature, x5t, ECDHE	37	115	90	242
EDHOC – RPKs, Signature, kid, ECDHE	37	102	77	216
EDHOC – X.509s, Static DH, x5t, ECDHE	37	58	33	128
EDHOC – RPKs, Static DH, kid, ECDHE	37	45	19	101

EDHOC and DTLS 1.3 overhead comparison[2]

[1] https://www.ietf.org/archive/id/draft-ietf-lake-edhoc-22.html
[2]Ambrosin M, Conti M, Lazzeretti R, et al. Collective remote attestation at the Internet of Things scale: State-of-the-art and future challenges[J]. IEEE Communications Surveys & Tutorials, 2020, 22(4): 2447-2461.

04

Remote attestation over EDHOC

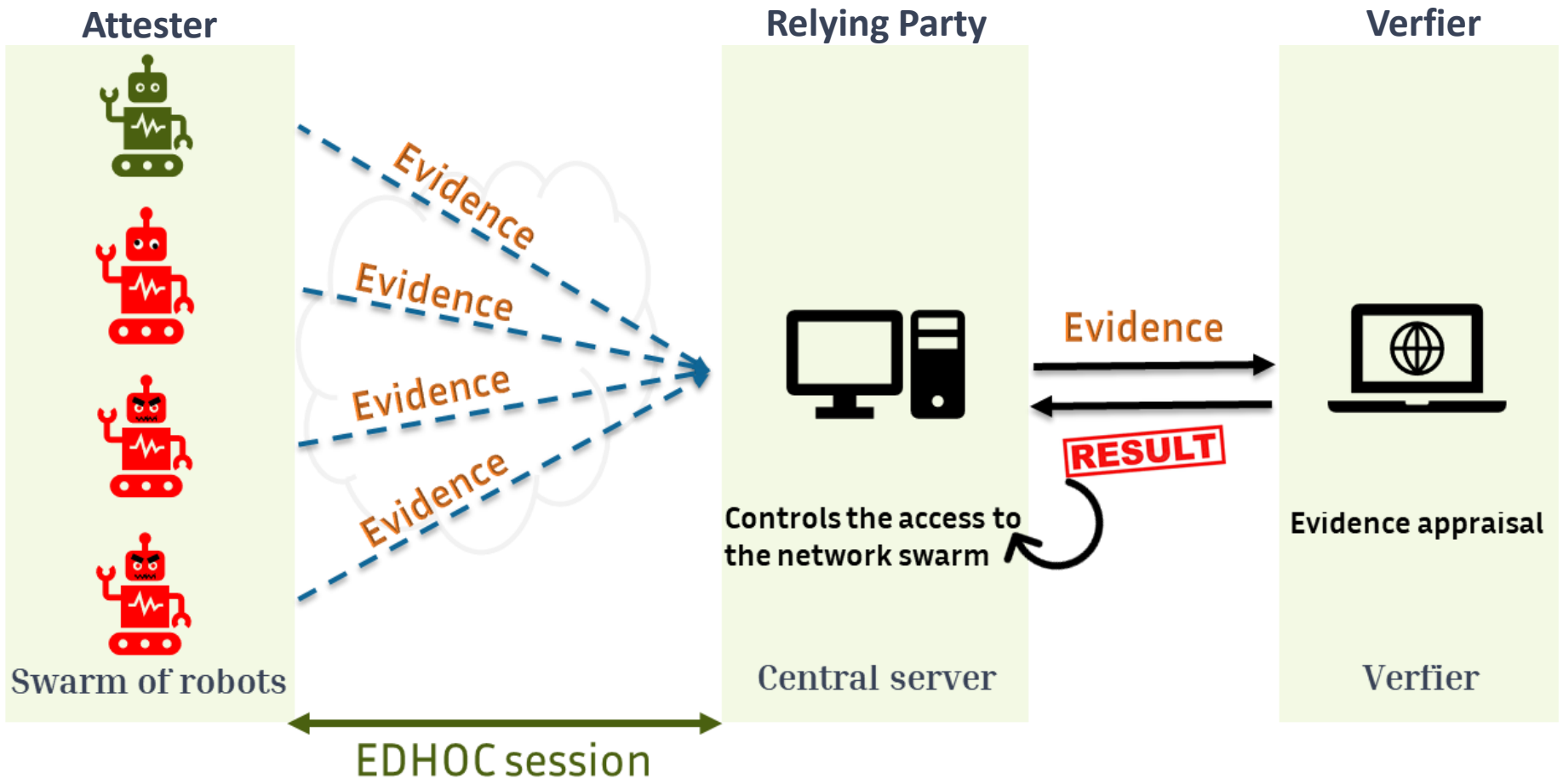
Abstract

This document specifies how to perform remote attestation as part of the lightweight authenticated Diffie-Hellman key exchange protocol EDHOC (Ephemeral Diffie-Hellman Over COSE), based on the Remote ATtestation procedureS (RATS) architecture.

Ongoing work at IETF LAKE:

Internet-Draft: Remote attestation over EDHOC

Mapping the previous swarm of robots to remote attestation



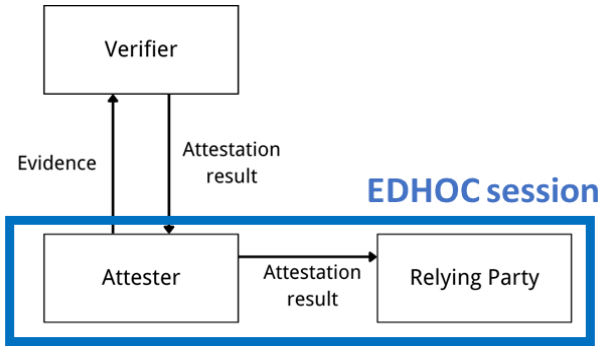
Internet-Draft: Remote attestation over EDHOC

The specification defines:

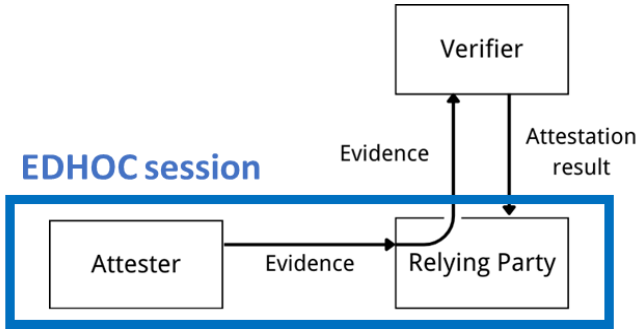
- Forward remote attestation
- Reverse attestation
- Mutual attestation

Features:

- less memory and energy consumption compared to TLS
- network authentication in parallel with attestation
  - attestation items are carried in EAD field over EDHOC



Passport model



Background-check model

Table of Contents

Forward remote attestation

- Use case: An IoT device needs to be attested for onboarding check

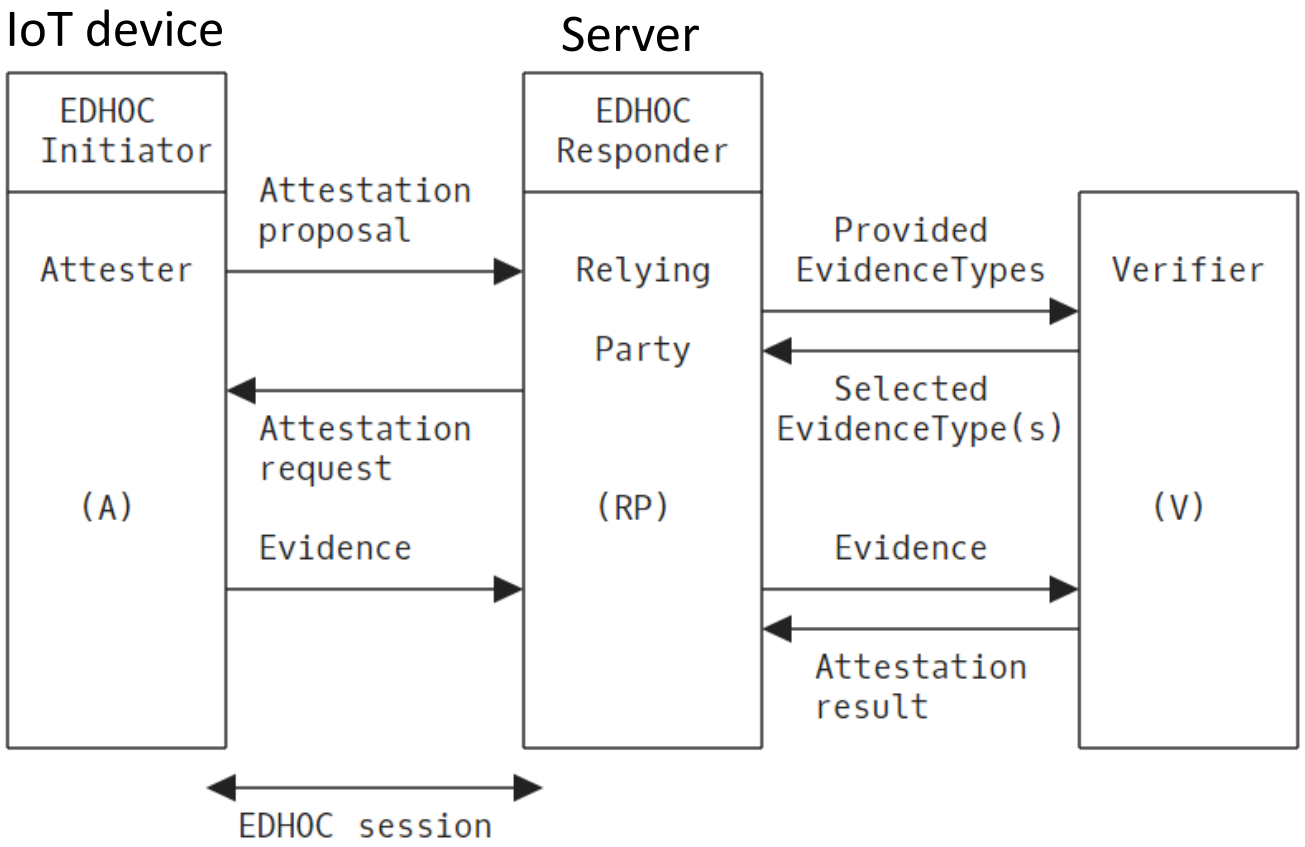


Table of Contents

Reverse attestation

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

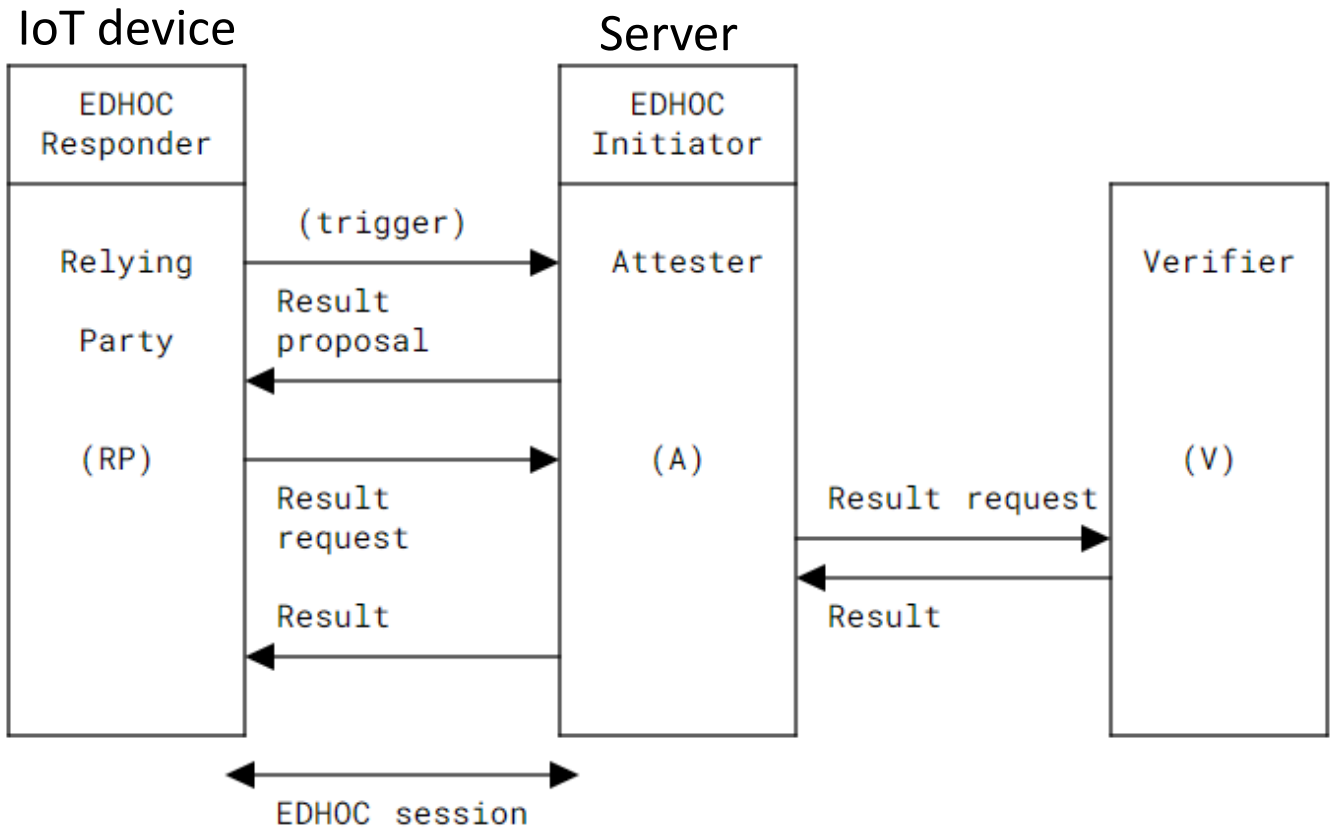
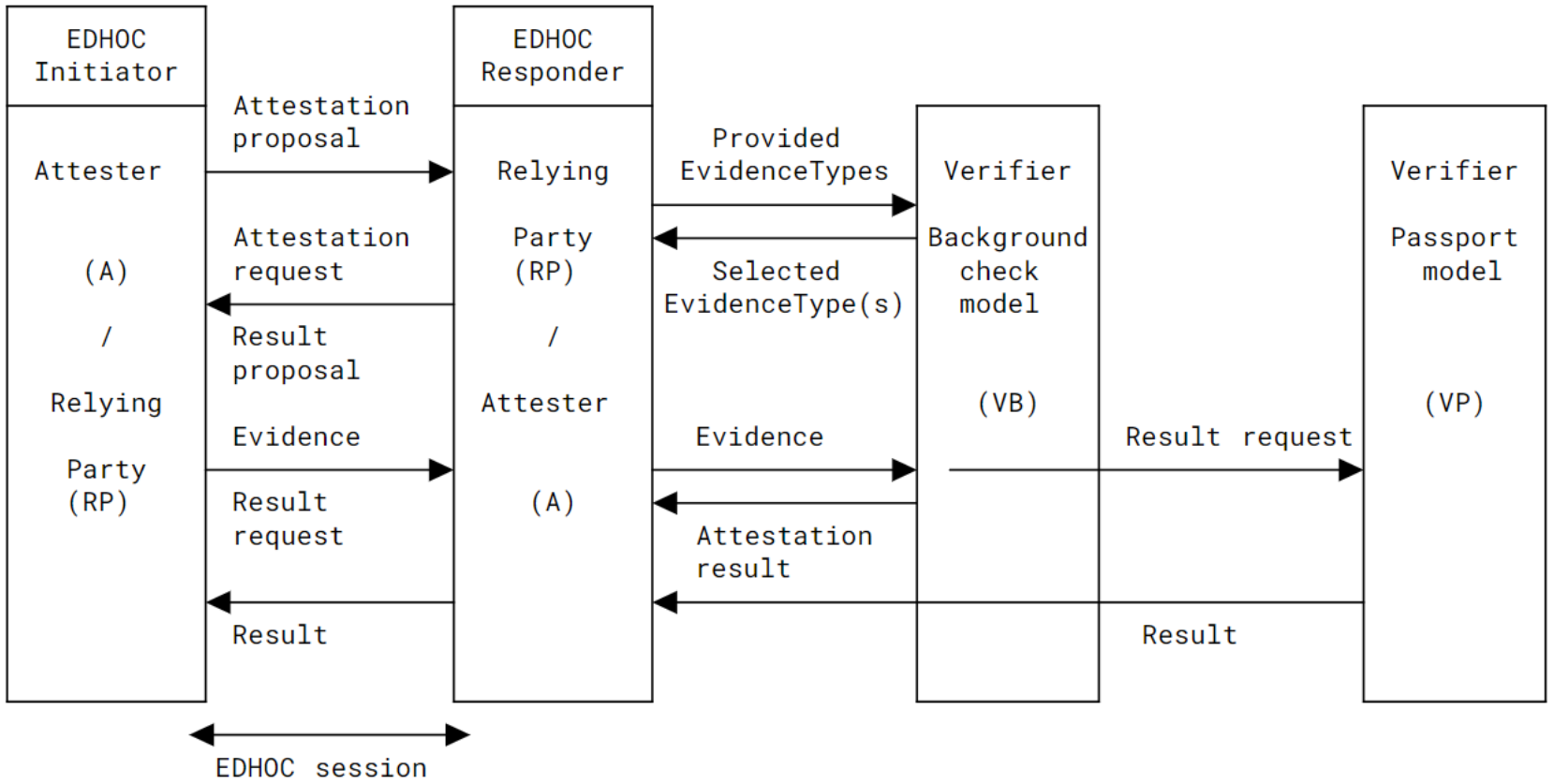


Table of Contents

Mutual attestation

- Use case: Both the device and server need to be attested



05

Implementation

Attester: attestation service

Verifier: verification process

Evidence generation

Signed attestation token

- in COSE_Sign1 structure
- overall size: 227 bytes

```
COSE_Sign1 = [
    Headers,
    payload : bstr / nil,
    signature : bstr
]
```

Verification process:

- parse COSE_Sign1
- signature check
- nonce check
- evidence check

```
/*payload*/
{
/eat-nonce/                 10: h'a29f62a4c6cdaae5',
/ueid/                      256: 'bbb',
/measurements/              273: [
  /CoAP Content-Format ID/        [ 258,
  /evidence in CoSWID/              {
                                      0: 'tagID'            /tag-id/
                                      12: 0                 /tag-version/
                                      1: "DotBot firmware"  /software-name/
                                      2: {                  /entity/
                                          31: "Attester"    /entity-name/
                                          33: 1             /role, must be "tag-creator" which is 1/
                                          },
                                      3: {                  /evidence/
                                          17: [             /file/
                                               {
                                                24: "partition0-nrf52840dk.bin",   /fs-name/
                                                20: (size of file),                /size in bytes/
                                                7: [                               /hash of file/
                                                    1,                             /alg SHA-256/
                                                    h'06294f6806b9c685eea795048579cfd02a0c025bc8b5abca42a19ea0ec23e81a'
                                                    ]                              /hash value/
                                               }
                                              ]
                                          }
                                    }
                                  ]
                                ]
},
```

Thank you !

yuxuan.song@inria.fr