

Formal Analysis of draft-ietf-lake-ra-03

using SAPIC+

Elsa Lopez Perez, Inria

IETF-125 – 16/03/2026

draft-ietf-lake-ra

Outline of the presentation

1. Remote Attestation Over EDHOC
 - a. Studied Models
 - i. Background check model
 - ii. Passport Model
 - iii. Mutual Attestation
2. Claimed security properties
 - a. Remote Attestation Properties
 - b. EDHOC Properties
 - c. Combined Properties
3. Symbolic Analysis
 - a. Verification framework: SAPIC+
 - b. Modeling of properties
4. Results

Models studied under formal analysis

Case Name	Attester	Relying Party	Model	Studied
Unilateral 1	Initiator	Responder	Background Check	yes
Unilateral 2	Responder	Initiator	Background Check	no
Unilateral 3	Initiator	Responder	Passport	no
Unilateral 4	Responder	Initiator	Passport	yes

Also studied

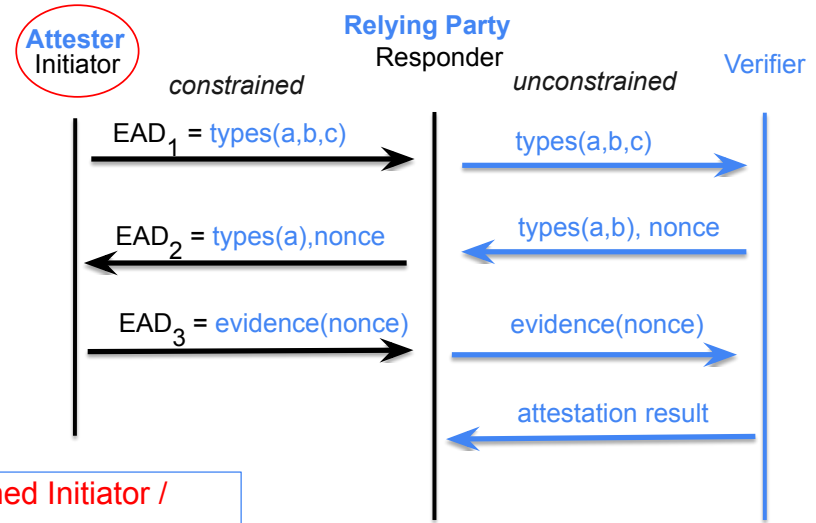
- Mutual attestation: Unilateral 1 + Unilateral 4

Remote Attestation Over EDHOC

Unilateral 1

- Background check model
 - **Attester:** IoT device.
 - **Relying Party:** Network service.
 - **Verifier:** Trusted Web Server

The EDHOC session is established between the Attester and the Relying Party.



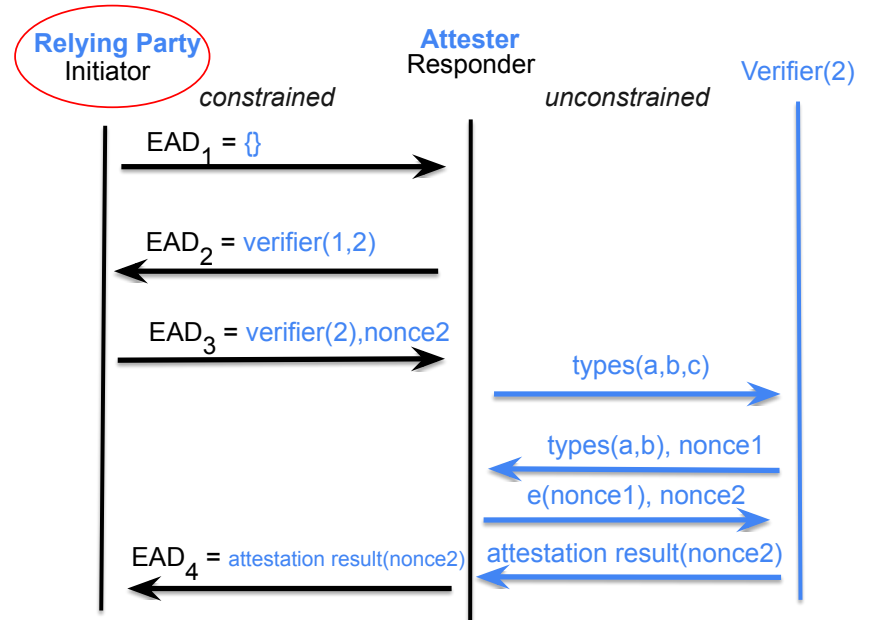
- Constrained Initiator / Attester
- Background-check model

Remote Attestation Over EDHOC

Unilateral 4

- Passport model
 - **Attester:** Network Service
 - **Relying Party:** IoT device
 - **Verifier:** Web Server

The EDHOC session is established between the Attester and the Relying Party.



- Constrained Initiator / Relying Party
- Passport model

IETF-125 – 16/03/2026

draft-ietf-lake-ra

Remote Attestation Over EDHOC

Mutual Attestation

In mutual attestation mode, **both the IoT device and the network service undergo attestation** to verify each other's trustworthiness.

The **network service attestation** employs the **passport model**:

- **Attester:** Network Service.
- **Relying Party:** The IoT device.
- **Verifier:** Trusted web server.

The **IoT device attestation** employs the **background-check model**.

- **Attester:** The IoT device.
- **Relying Party:** Network Service
- **Verifier:** Trusted web server.

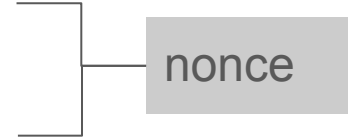
The EDHOC session is established between the Attester and the Relying Party.

Claimed Properties

RA Properties defined based on [1]

1. Evidence Freshness:

- a. Ensures that the evidence reflects the current state of the Attester, preventing the replayed or outdated evidence.



2. Attestation Result Freshness:

- a. Ensures that the attestation result is based on the latest available evidence, preventing the Relying Party from accepting stale attestation result.

3. Integrity of Evidence:

- a. Ensures that the evidence has not been altered in an unauthorized manner since it was created and transmitted.



4. Integrity of Attestation Result:

- a. Ensures that the attestation result has not been altered in an unauthorized manner since it was created and transmitted.

Claimed Properties

EDHOC Properties

1. **Mutual Authentication:**
 - a. Ensures that both parties verify each other's identity, preventing impersonation and man-in-the-middle attacks.

Claimed Properties

Combined Properties

1. **Channel binding:**
 - a. Ensures that the attestation is cryptographically bound to the authentication.
2. **Agreement of Parameters:**
 - a. Ensures that the Responder's view of attestation and authentication parameters at the end of the protocol matches those in the Initiator's view.
 - b. This property ensures **both** mutual **authentication** and **attestation** correctness.

Symbolic Analysis

- **Assumption:** We analyze the protocol under the well-established **Dolev–Yao (DY)** attacker model.
- The **attacker has full control of the communication**
 - Intercept and eavesdrop on messages
 - Modify or redirect messages
 - Replay previously observed messages
 - Interact with honest parties
- **Perfect Cryptography Assumption:** Cryptographic primitives are not breakable. Security failures must arise from the protocol design, not from breaking cryptography.
- **Security properties** formally expressed as **reachability** or **equivalences**

Verification Framework: SAPIC+

Advantages:

- **Unified Model:** A single SAPIC+ file can be used as input for multiple verifiers.
 - Proverif
 - Tamarin
- **Exploits strengths of different tools.**
- **Protocols** are described via the **applied pi-calculus** (similarly to ProVerif)
- Property specification use **first-order-logic** (similarly to Tamarin)

Advanced threat model?

We can add specific capabilities or "compromise scenarios" to the attacker's knowledge base.



- **Compromise(k):** compromise of long term keys
- **LeakShare(x):** Ephemeral Key Leakage
- **LeakSKey(prk):** Session Key Leakage
- **LeakAttKey(ak):** Leakage of attestation key

IETF-125 – 16/03/2026

draft-ietf-lake-ra

Modeling of Properties

Freshness:

Freshness of evidence and result is captured by requiring that any two verification events referring to the same honest nonce must correspond to the same protocol execution.

lemma AttestationEvidenceFreshness:

```
All m pkI1 pkI2 pkR1 pkR2 nonce sig1 sig2
  cid1 cid2 pk_attkey1 pk_attkey2 y1 y2
  gxy1 gxy2 #i #j #k.
VerifiedNonceR(cid1, m, pkI1, pkR1, sig1,
  nonce, y1, gxy1, pk_attkey1)@i &
VerifiedNonceR(cid2, m, pkI2, pkR2, sig2,
  nonce, y2, gxy2, pk_attkey2)@j &
HonestNonce(nonce)@k
==>
(#i = #j) & (#k < #i)
```

Integrity:

It is expressed as a correspondence property: successful verification implies prior generation by an honest peer, unless a compromise event occurred.

lemma AttestationEvidenceIntegrity:

```
All cidR pkI pkR nonce sig y gxy #i #j.
VerifiedSignatureR(cidR, pkI, pkR, nonce,
  sig, y, gxy)@i & Honest(pkI)@j
==>
(Ex cidI ak x #t.
  AttestationGeneratedI(cidI, pkI, pkR, nonce,
    sig, ak, x, gxy)@t & t < i)
| (Ex ak #t. LeakAttKey(ak)@t)
| (Ex #t. Compromise(pkI)@t)
| (Ex #t. Compromise(pkR)@t)
```

Modeling of Properties

Channel Binding:

Successful attestation acceptance by the Responder implies that the same evidence was previously sent by the Initiator authenticated in that EDHOC session, unless a compromise event occurred.

```
lemma ChannelBinding:
  All cidR pkI pkR nonce sig gxy gx gy th3
    pk_attkey id_att #j.
  AttestationAcceptedR(cidR, pkI, pkR, nonce,
    sig, gxy, gx, gy, th3, pk_attkey, id_att
  )@j
==>
  (Ex cidI ak #i.
  SentByI(cidI, pkI, pkR, nonce, sig, gxy, gx,
    gy, th3, ak, pk_attkey, id_att)@i & i <
    j) | (Ex ak #t. LeakAttKey(ak)@t)
```






Agreement of parameters:

It ensures that whenever a party finishes, there exists a unique peer execution that agrees on the same authentication and attestation parameters

```
lemma AgreementParameters:
  All cidR method pkI pkR gxy y gx gy nonce
    sig dev_state id_att pk_attkey #i #j #k.
  FinishR(cidR, method, pkI, pkR, gxy, y, gx,
    gy, nonce, sig, dev_state, id_att,
    pk_attkey)@i &
  Honest(pkI)@j & Honest(pkR)@k
==>
  (Ex cidI x ak x #t.
  FinishI(cidI, method, pkI, pkR, gxy, x, gx,
    gy, nonce, sig, dev_state, id_att, ak,
    pk_attkey)@t & t < i)
| (Ex pkR #t. Compromise(pkR)@t & t < i)
| (Ex pkI #t. Compromise(pkI)@t & t < i)
| (Ex ak #t. LeakAttKey(ak)@t & t < i)
```

Results

Minimal Compromise Scenarios

Trace Property	Active + Capabilities	Result
Evidence Freshness	-	
Attestation Result Freshness	-	
Evidence Integrity	LeakAtKey(ak) & Compromise(k)	
Attestation Result Integrity	LeakAtKey(ak) & Compromise(k)	
Channel Binding		
Agreement of Parameters	LeakAtKey(ak) & Compromise(k)	

IETF-125 – 16/03/2026

draft-ietf-lake-ra

Results

Security Property	Method				
	0	1	2	3	4
Agreement of Parameters	✓	✓	✓	✓	✓
Evidence Freshness	✓	✓	✓	✓	✓
Result Freshness	✓	✓	✓	✓	✓
Evidence Integrity	✓	✓	✓	✓	✓
Result Integrity	✓	✓	✓	✓	✓
Mutual Authentication	✓	✓	✓	✓	✓
Channel binding	✗	✗	✗	✗	✗



- We analyzed the properties in all 5 authentication methods [1] [2]
- **Channel binding is not verified in any of the authentication methods**

Attack due to a lack of cryptographic binding between the attestation evidence and the authentication session

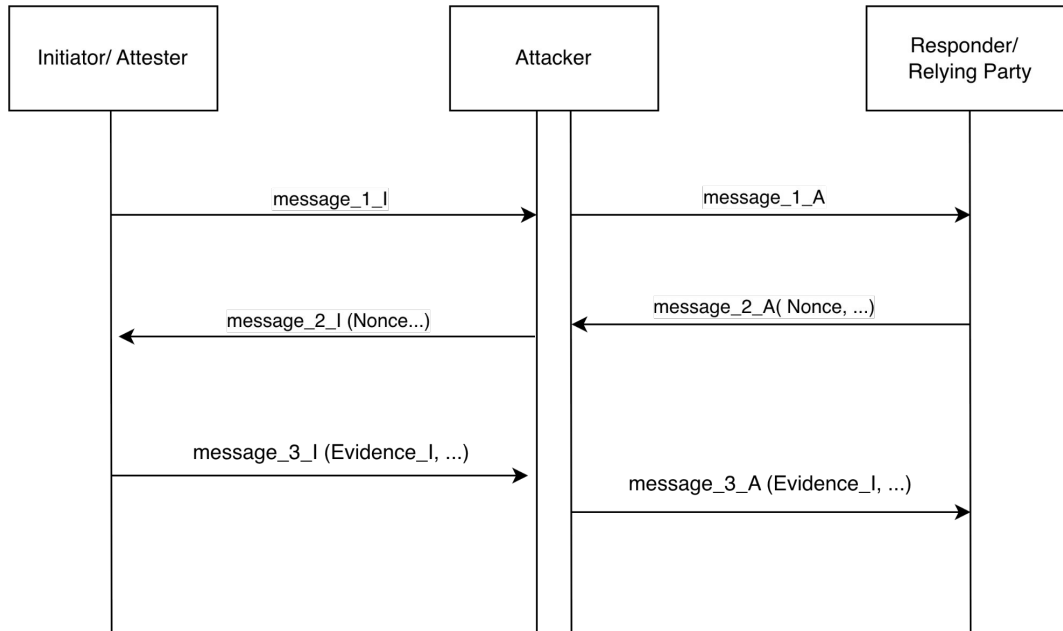
Attack can happen even without leakage of the attestation key!!

[1] <https://github.com/ElsaLopez133/edhoc-ra-formal-analysis>

[2] [ElsaLopez133/edhoc-psk-attestation-formal-analysis](https://github.com/ElsaLopez133/edhoc-psk-attestation-formal-analysis): Formal analysis of remote attestation in EDHOC-PSK

Results

Description of the Attack



Description of the attack:

1. **Compromise of authentication keys**
2. MitM establishes sessions with both endpoints and obtains a valid attestation evidence from an Honest Attester.
3. Attacker presents the evidence within a different session, as if it were produced for that authentication channel.
4. If evidence is not cryptographically bound to the session, verification succeeds.

Results

Proposed Mitigation

Security Property	Method					Fix
	0	1	2	3	4	
Agreement of Parameters	✓	✓	✓	✓	✓	✓
Evidence Freshness	✓	✓	✓	✓	✓	✓
Result Freshness	✓	✓	✓	✓	✓	✓
Evidence Integrity	✓	✓	✓	✓	✓	✓
Result Integrity	✓	✓	✓	✓	✓	✓
Mutual Authentication	✓	✓	✓	✓	✓	✓
Channel binding	✗	✗	✗	✗	✗	✓

Proposed mitigation:

Include the hash of the first two messages in the digitally signed attestation evidence.

This hash contains information linked to the authentication session, including the ephemeral shared secret that is session unique.

Next Steps

Case Name	Attester	Relying Party	Model	Studied
Unilateral 1	Initiator	Responder	Background Check	yes
Unilateral 2	Responder	Initiator	Background Check	no
Unilateral 3	Initiator	Responder	Passport	no
Unilateral 4	Responder	Initiator	Passport	yes

Thank you!

IETF-125 – 16/03/2026

draft-ietf-lake-ra