Lightweight Authorization using EDHOC

https://www.ietf.org/archive/id/draft-ietf-**lake-authz-01**.html

https://github.com/openwsn-berkeley/**lakers**

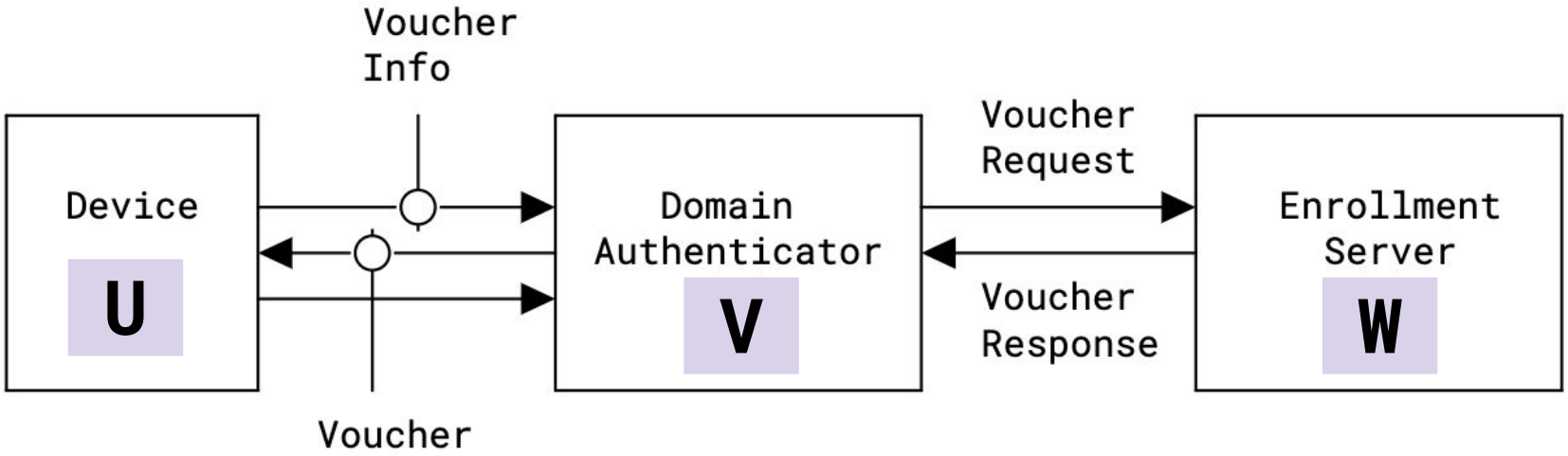
Geovane Fedrecheski, Inria

Contents

- Recap
- Implementation updates
- Draft updates
- Open issues / next steps

Recap: **Lightweight Authorization** using EDHOC

Also referred to as:
- **authz**
- **zero-touch** network join



Implementation: `lakers` Rust library (was `edhoc-rs`)

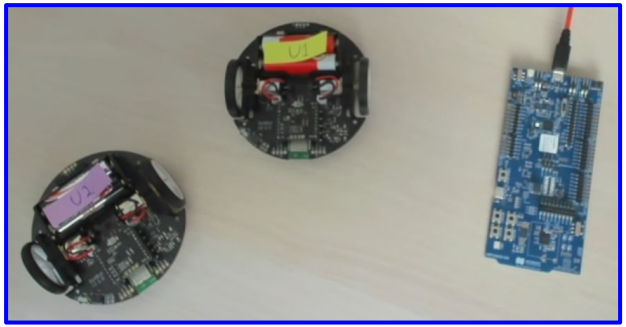
- EDHOC for microcontrollers (and more)
- **v0.5.1** at https://crates.io/crates/lakers

```
cargo add lakers
```

```
pip install lakers-python
```

```
wget https://.../lakers-c.zip
```

- **lake-authz-00 fully implemented**
  - device (**U**), authenticator (**V**), and server (**W**)
  - **demo in real hardware**



Draft Updates

Voucher can now carry ciphertext of COSE_Encrypt0

Previously it was just a MAC:

- Voucher = bstr .cbor EDHOC-Expand(PRK, info, length)

Now, it can contain encrypted data:

- Voucher = COSE_Encrypt0.ciphertext

(note that in AEAD the ciphertext contains a MAC)

Opaque info

Problem: U and W may need to exchange more information, e.g.:

1. U -> W: **context around U**, e.g. gateways nearby
2. W -> U: e.g. whether **V** is an **owner** or simply an **access-provider**
3. W -> U: **actionable error handling**, e.g. hints on allowed gateways

Solution: add `?OPAQUE_INFO: bstr`

carry AEAD-protected information between U and W

```
plaintext of ~~ENC_ID~~ ENC_U_INFO = (
  ID_U:          bstr,
  ?OPAQUE_INFO: bstr,
)
```

Example diff (case #1 above).

Error Handling

New: EDHOC Error "Access denied"

ERR_CODE	ERR_INFO Type	Description
TBD3	error_content	Access denied

1. Defined as generic error for EDHOC
2. Then, for lake-authz specifically,
   we define an encrypted `REJECT_INFO`

```
error_content = (
  REJECT_TYPE:   int,
  ?REJECT_INFO: bstr,
)
```

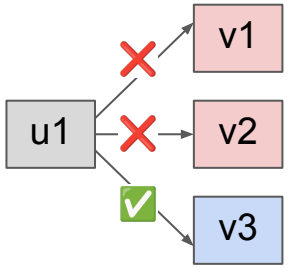
```
plaintext of REJECT_INFO = (
    OPAQUE_INFO:  bstr,
)
```

`OPAQUE_INFO` contains actionable information about the error (example in next slide)

Enrollment hints

Problem:

- device tries to join via "wrong gateway" several times, until it eventually gets authorized



Consumes time ⏳ and energy 🔋

Proposal:

- U may tell W which gateways it has discovered
  - `u_hints`: within `OPAQUE_INFO` in EAD_1
- W may tell U which gateways it should use
  - `v_hints`: within `OPAQUE_INFO` in error case

Open questions:
* what should be the gateway identifier? e.g. MAC address, SSID, PAN ID, LoRaWAN devaddr

Both `u_hints` and `v_hints` are optional

Goal is to speed up enrollment when many Vs are available

Examples

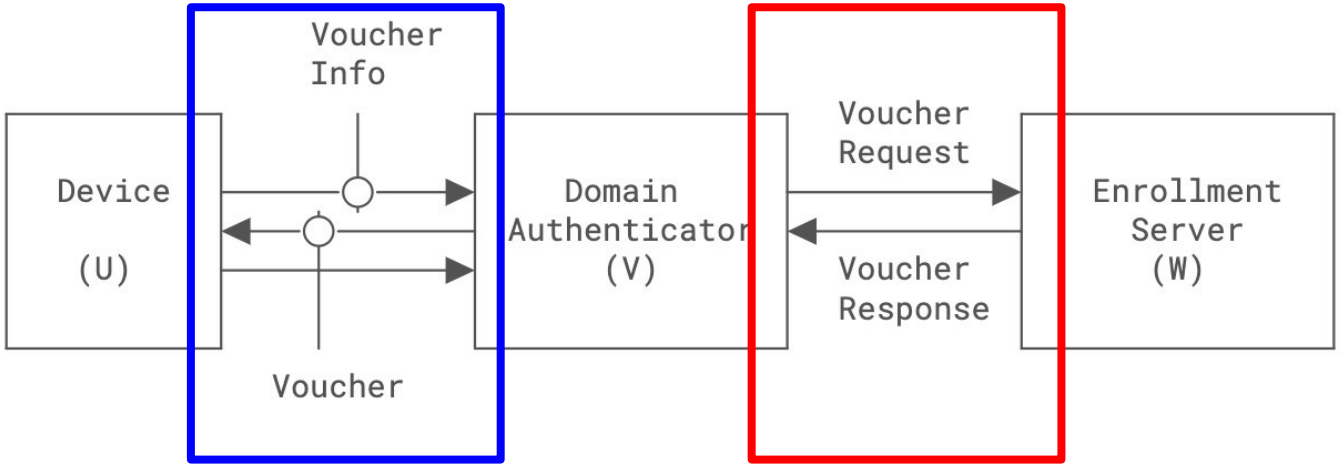
1. Minimal:
    a. Simple, successful execution
2. Wrong gateway
    a. Demonstrate error case with wrong gateway
    b. Using REJECT_INFO field of the EDHOC error Access Denied

```
> plaintext of REJECT_INFO contains a
> list of suggested gateways = [h'3963C9D05C62']
```

Open issues / Next steps

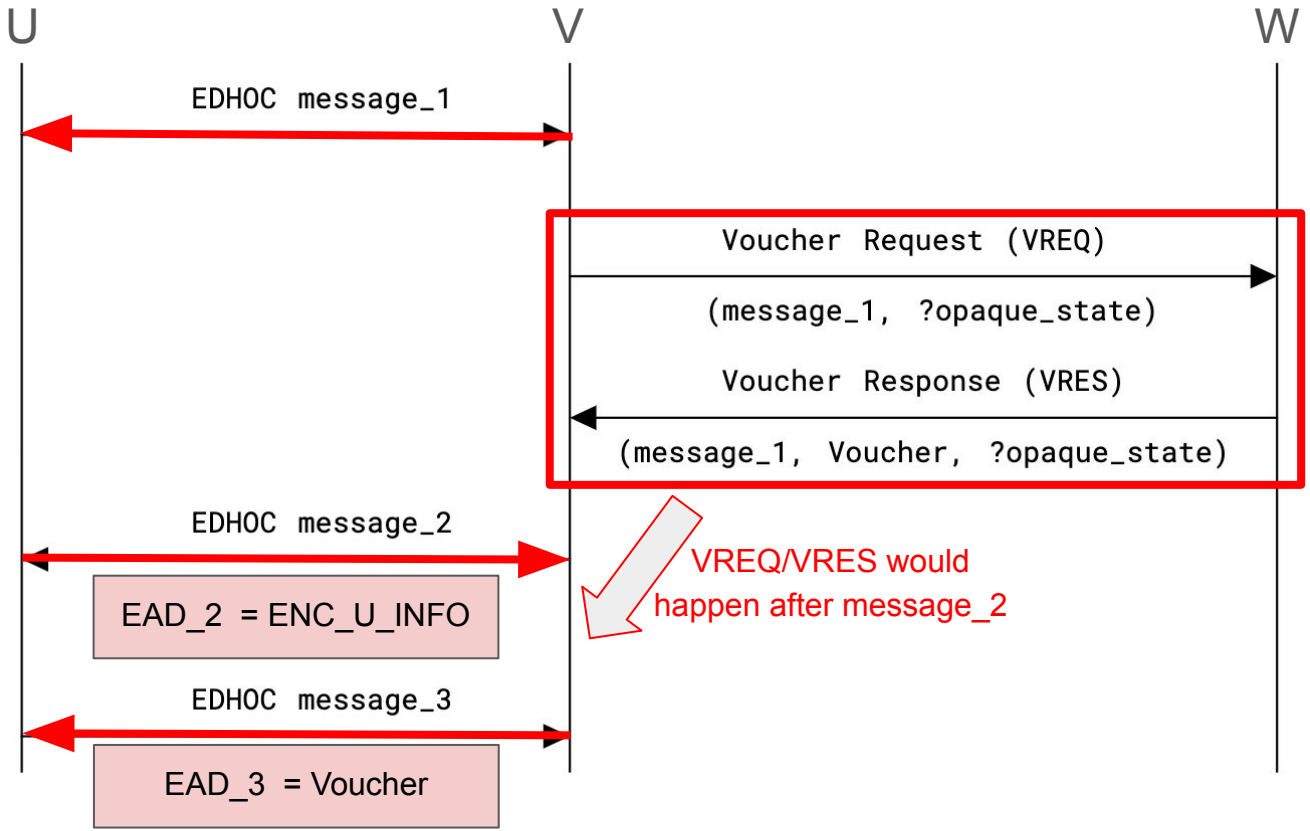
#19: explain error in the VREQ/VRES protocol

the "Access Denied" error is defined for this interaction



to-do: explain it here

#21: authz with inverted EDHOC roles



#23, #24, #25: make OPAQUE_INFO less opaque?

```
plaintext of EAD_1 = (
    ID_U:            bstr,
    ?OPAQUE_INFO:    bstr,
)
```

#23

```
plaintext of EAD_1 = (
    ID_U:       bstr,
    ?nearby_vs: net-ids,
)
```

```
network-ids = {
 / mac    /   1: [+bstr],
 / pan-id /   2: [+bstr],
 / ipv4   /   3: [+bstr],
 / ipv6   /   4: [+bstr],
 / TBD    / TBD: [+bstr],
}
```

```
plaintext of Voucher = (
    ?OPAQUE_INFO: bstr
)
```

#24

```
plaintext of Voucher = (
  role-of-v:        uint,
  directives-for-u: bstr,
)
```

```
plaintext of REJECT_INFO = (
    OPAQUE_INFO:     bstr,
)
```

#25

```
plaintext of REJECT_INFO = (
  suggested-vs:              net-ids,
  additional-recovery-info:  bstr,
)
```

Current

New (?)

Stale issues

#2: Add EAD_3 field?

- After implementing it, I have found no need for an EAD_3
  - unless #21 (inverted I and R) goes ahead, but it’s a separate issue
- Suggested action: close issue

#3: Voucher is not bound to U

- Voucher is bound to EDHOC session via transcript hash, hence no problem
- Suggested action: close issue

Final remarks

- Implementation “closely” following draft
- U and W can now share more information via `OPAQUE_INFO`
  - but can we improve it?
- New EDHOC Error “Access Denied”
  - and in lake-authz, allows addressing scalability issues (many Vs)
- Open issues / next steps

Thank you!

https://www.ietf.org/archive/id/draft-ietf-**lake-authz-01**.html

https://github.com/openwsn-berkeley/**lakers**

geovane.fedrecheski@inria.fr