EDHOC

draft-ietf-lake-edhoc-19
draft-ietf-lake-traces-05

https://github.com/lake-wg/edhoc

IETF 116, Yokohama, LAKE WG, March 30, 2023

Since IETF 115

— edhoc-18
  — final wrap up from WGLC
— edhoc-19
  — directorate reviews (secdir, intdir, tsvart, genart)
  — shepherd review and Stephen's pre-last call review
— github master is essentially -19

— traces-04
  — updated to match edhoc-19
  — now both traces are checked

As always, details in https://github.com/lake-wg/edhoc

edhoc-17 → edhoc-18

Summary: edhoc-17 → edhoc-18

- Main changes:
  - PAD field removed
    - Padding realised as EAD with ead_label=0
  - EAD syntax revised
    - ead_value is now optional

```
ead = (
  ead_label : int,
  ? ead_value : bstr,
)
```

- Minor changes
  - Terminology alignment
  - Clarifications & nits

- See Change log

edhoc-18 → edhoc-19

Summary: edhoc-18 → edhoc-19

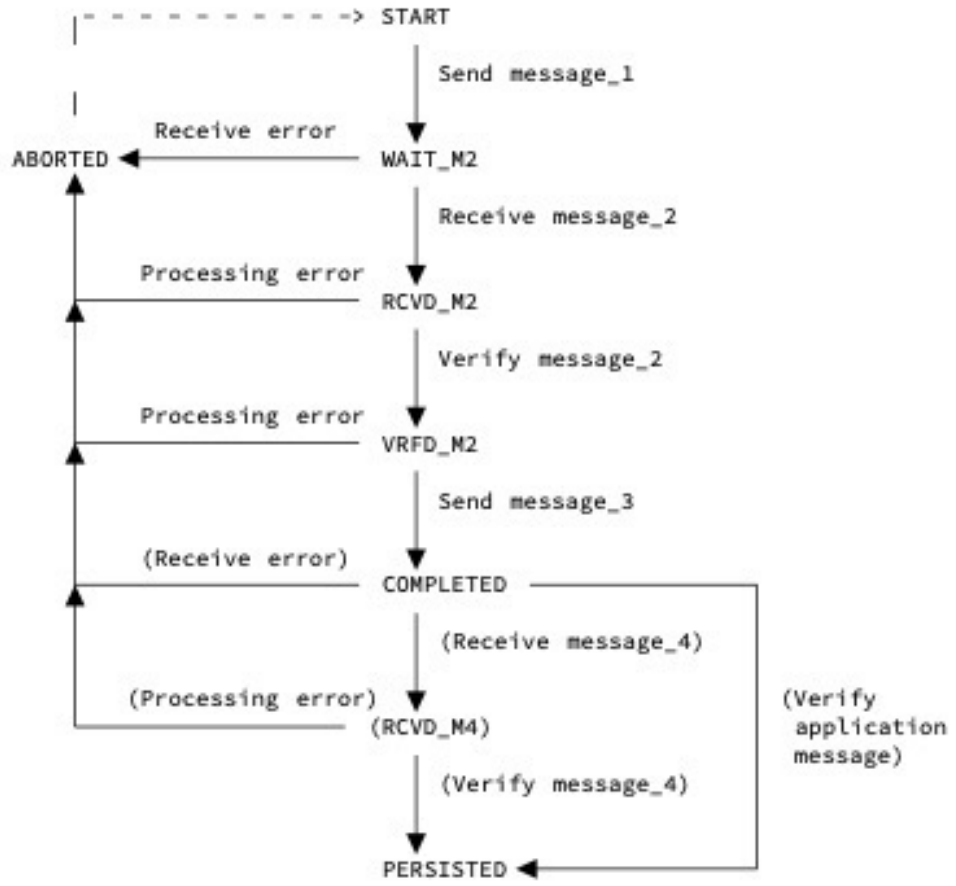
- Main changes
  - Appendix A - stricter normative text about the use of OSCORE
  - Appendix H (message correlation) removed, new section 3.4.1
  - New Appendix with example state machine

- Minor changes
  - Clarifications (many)

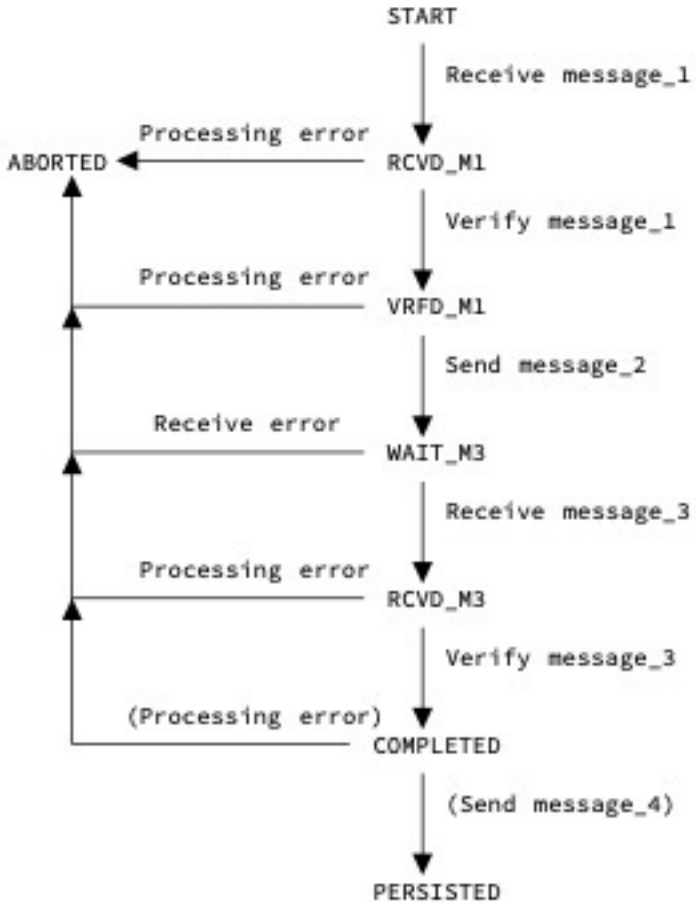
- See Change log

New Appendix: State Machine Example

Initiator State Machine



Responder State Machine



Recent Issues

Revisited error discussion (#402)

A. Brian requested error code to handle the case when a referenced credential is missing (#400)

— As discussed previously, we only pre-register error codes which can trigger autonomous action leading to successful completion of the protocol

— #400 matches this criterion
  — PR #401 is a proposed solution

Revisited error discussion (#402)

B. Marco propose to introduce an error code for EAD-related errors

For example:

— reuse the ead_label to indicate first EAD item that failed

— let the EAD application define ead_error

ERR_INFO = (
ead_label : int,
? ead_error : any,
)

— Do we need this, or should EAD applications register errors like any other applications?

Revisited error discussion (#402)

C. John’s comment about protected errors (next slide)

Protected error, EDHOC-exporter, and responder processing of message_3 (#375)

- After sending message_3 the Initiator can use EDHOC_Exporter
- When the responder receives message_3 the message might parse perfectly and the responder can use the exporter. The responder might still not want to talk to the Initiator.
- In general it is possible for the Responder to use the derived application keys to protect the EDHOC error message, but it is not clear if this should be done. This would corresponds to protected alert messages in, e.g., TLS.
- Error messages after EDHOC message_3 are **not** protected with OSCORE, when following draft-ietf-core-oscore-edhoc.
- PR #398 starting point, more clarifications needed

Security considerations for kccs and kcwt (#403)

- kcwt and kccs are new entries in the "COSE Header Parameters" registry defined in EDHOC
- kcwt contains a CBOR Web Token (CWT, RFC8392)
- kccs contains a CWT Claims set
  - Also defined in RFC8392
  - The abbreviation CCS is used in EDHOC
  - Default format for raw public keys

- Current draft is not elaborate much on kcwt and kccs
- In particular the use of kccs for transport of RPK means that there has to be some other mechanism for verifying the public key
- Security considerations need to be added

Next steps

- lake-edhoc waiting for AD review and Last Call
  - fix the few remaining issues during Last Call

- lake-traces ready for WGLC?