EAP-based Authentication Service for CoAP

Work in progress for:
draft-ietf-ace-wg-coap-eap-02

Rafael Marín-López, University of Murcia
Dan García-Carrillo, University of Oviedo

ACE Interim Meeting, June 08th, 2021

CoAP-EAP – Updates Summary 01 version

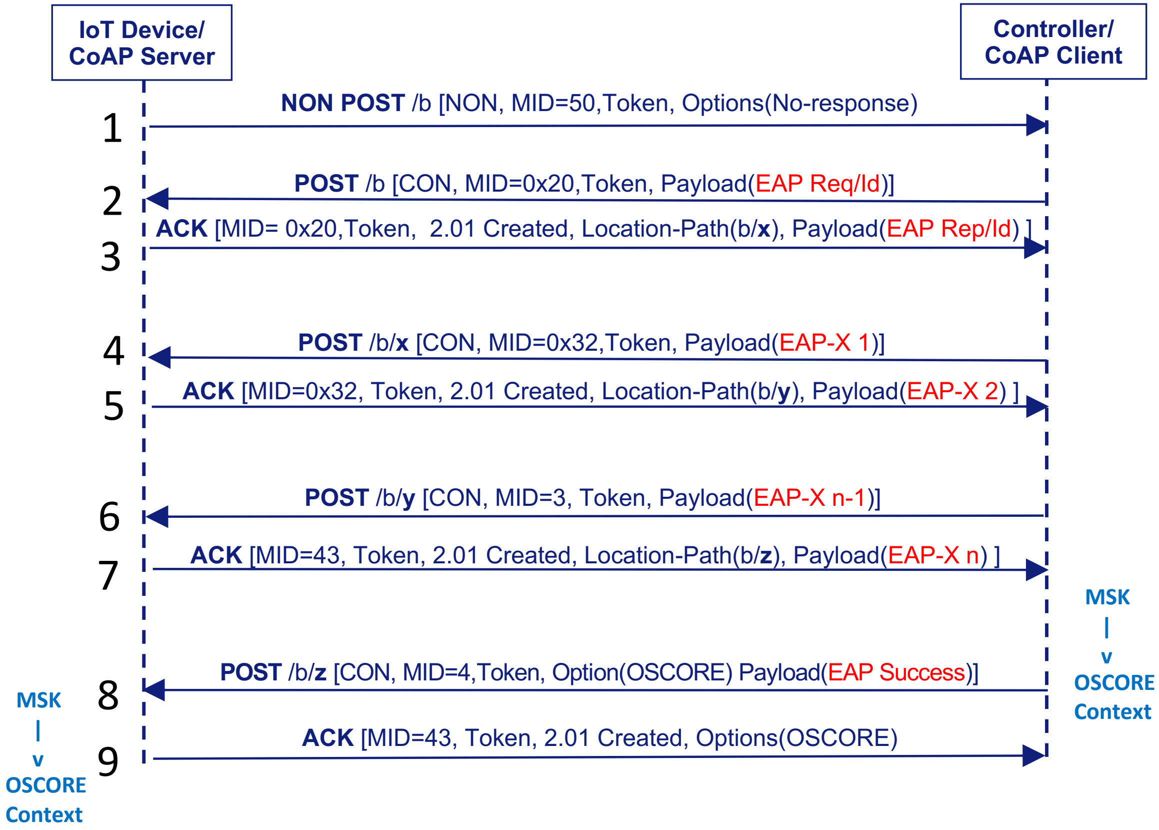
- Changed URI to conform to HATEOAS
- Added casuistic when a message is lost
- Added explanation on how the CoAP server and Client processes work.
- Writing and typo review

CoAP-EAP – Ordering guarantee following HATEOAS

- General Service URI **/b**
- Each step within the authentication creates a new resource with structure
  - /b/x
    - x -> Value representing the current step in the authentication process

x can be set to any value as long as it serves to specify univocally to both entities of the exchange the next step within the authentication process.

CoAP-EAP – Ordering guarantee w/ HATEOAS



IoT Device/ CoAP Server

Controller/ CoAP Client

1 **NON POST** /b [NON, MID=50,Token, Options(No-response)

2 **POST** /b [CON, MID=0x20,Token, Payload(EAP Req/Id)]

3 **ACK** [MID= 0x20,Token, 2.01 Created, Location-Path(b/**x**), Payload(EAP Rep/Id)]

4 **POST** /b/**x** [CON, MID=0x32,Token, Payload(EAP-X 1)]

5 **ACK** [MID=0x32, Token, 2.01 Created, Location-Path(b/**y**), Payload(EAP-X 2)]

6 **POST** /b/**y** [CON, MID=3, Token, Payload(EAP-X n-1)]

7 **ACK** [MID=43, Token, 2.01 Created, Location-Path(b/**z**), Payload(EAP-X n)]

MSK
|
v
OSCORE Context

8 **POST** /b/**z** [CON, MID=4,Token, Option(OSCORE) Payload(EAP Success)]

9 **ACK** [MID=43, Token, 2.01 Created, Options(OSCORE)

MSK
|
v
OSCORE Context

CoAP-EAP – Ordering guarantee w/ HATEOAS

- For each CoAP request (which contains an EAP Request) the process is

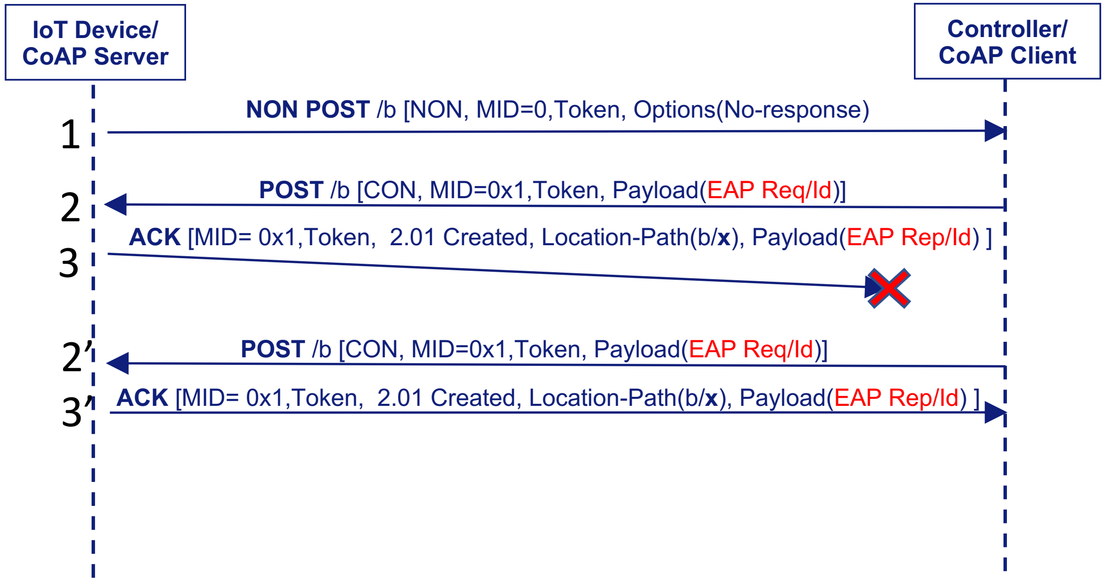
The CoAP server

- Receive the EAP payload and process it
- Send the content to the EAP state machine
- Receive the response from the EAP state machine
  - If everything goes as expected:
    - A new resource is created, /b/**y**
    - The previous resource /b/x is deleted
    - A response specifying the new resource is sent back

  - If an error occurs an error message is returned depending on the cause of the error.

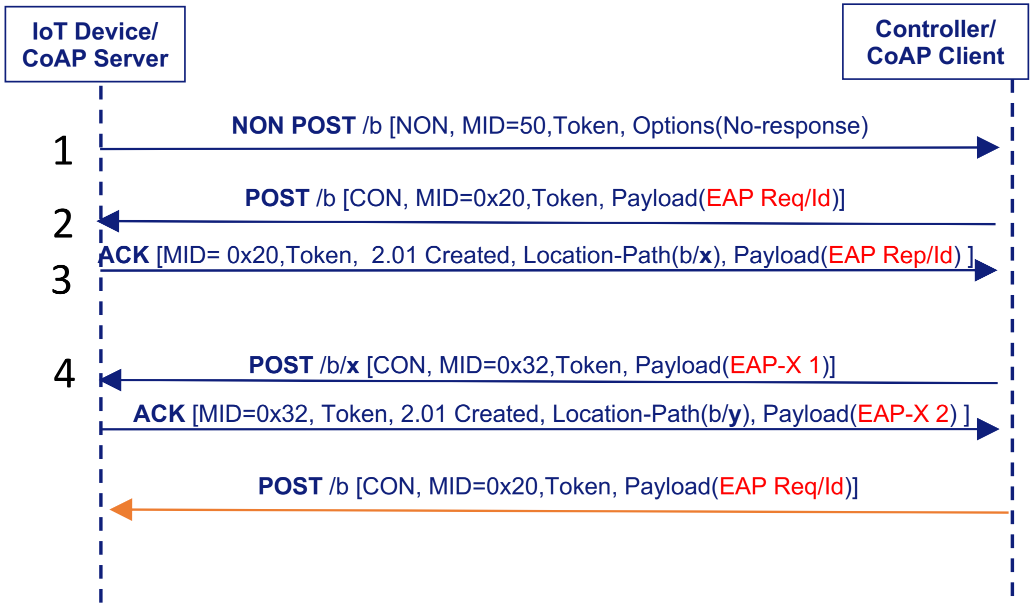
CoAP-EAP – Ordering guarantee w/ HATEOAS

- Casuistic when messages are lost
  - If the piggybacked response with a new resource is lost
    - The CoAP client will continue to retransmit until the response arrives
    - The CoAP server will recognize the message as retransmission and resend the message


IoT Device/
CoAP Server
Controller/
CoAP Client
1
NON POST /b [NON, MID=0,Token, Options(No-response)
2
POST /b [CON, MID=0x1,Token, Payload(EAP Req/Id)]
3
ACK [MID= 0x1,Token, 2.01 Created, Location-Path(b/x), Payload(EAP Rep/Id)]
2’
POST /b [CON, MID=0x1,Token, Payload(EAP Req/Id)]
3’
ACK [MID= 0x1,Token, 2.01 Created, Location-Path(b/x), Payload(EAP Rep/Id)]

CoAP-EAP – Ordering guarantee w/ HATEOAS

- Casuistic when an old message arrives



IoT Device/
CoAP Server

Controller/
CoAP Client

1 **NON POST** /b [NON, MID=50,Token, Options(No-response)

2 **POST** /b [CON, MID=0x20,Token, Payload(EAP Req/Id)]

3 **ACK** [MID= 0x20,Token, 2.01 Created, Location-Path(b/**x**), Payload(EAP Rep/Id)]

4 **POST** /b/**x** [CON, MID=0x32,Token, Payload(EAP-X 1)]

ACK [MID=0x32, Token, 2.01 Created, Location-Path(b/**y**), Payload(EAP-X 2)]

POST /b [CON, MID=0x20,Token, Payload(EAP Req/Id)]

IF managed at CoAP engine

- If the CoAP engine takes care of it, as the server recognizes de old message it can send a stored copy

Then the client would recognize MSGID <
and that he got the response already, dropping it

IF managed at Application

- If the control in the server goes up to the application, it generates a 4.04 not found since its deleted

Then the client would recognize the MSGID <
and that he got the response already, dropping it

THANK YOU