EAP-based Authentication Service for CoAP

Changes for draft-ietf-ace-wg-coap-eap-04

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

ACE Interim Meeting, October 12th, 2021

Summary of main changes for v04

- Update on Flow independent of CON and NON (Clarification from last interim)
- Discovery
- Sending server resource in the first message
- Keeping OSCORE to confirm keys in CoAP-EAP
- Current flow of operation

CON and NON independence Clarification from last interim

- After a design meeting with Carsten and Christian, some clarifications were made regarding the use of CON or NON in CoAP-EAP
    - Reliability mechanism will be used using CoAP-EAP (CON, or TCP, etc.)

- No assumptions about piggybacking

Discovery of the EAP authenticator

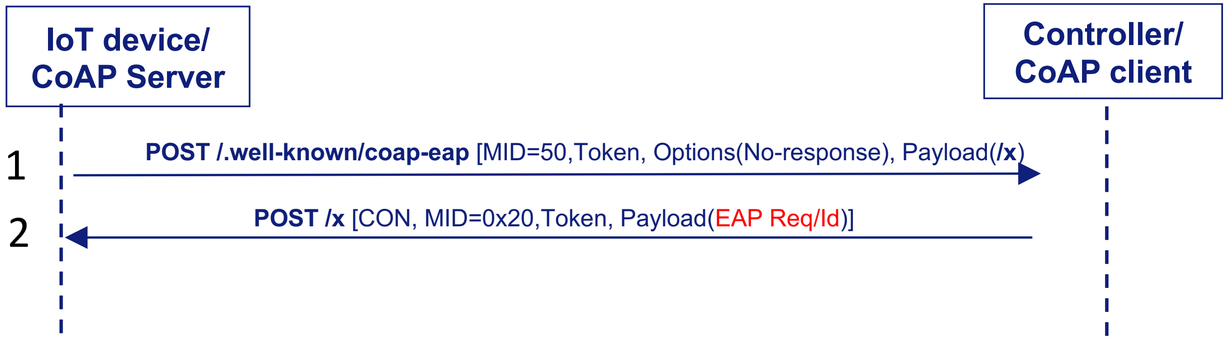
- Out of scope
- A brief discussion on this will be added to the next version - 04
  - First approach, to receive the IPv6 of the Border Router (e.g., RA) and send there the initial message

  Other approaches to be considered

  - DHCPv6 [RFC8415]
  - mDNS [RFC6762]

Sending the resource on the first message

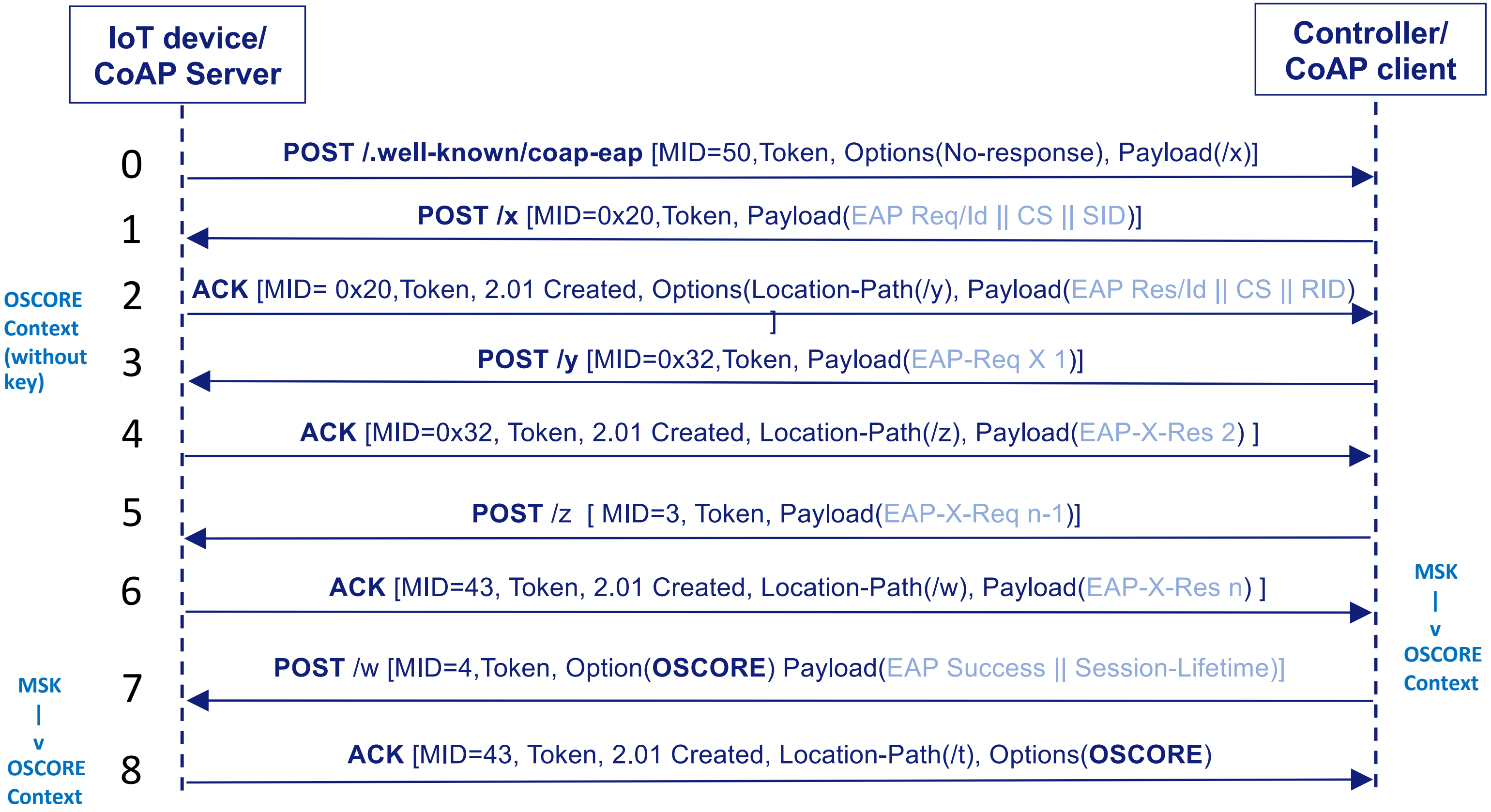
- Saves bytes over the air: well-known only sent once
- Avoids the CoAP server receiving unexpected well-known messages


IoT device/
CoAP Server
Controller/
CoAP client
1
POST /.well-known/coap-eap [MID=50,Token, Options(No-response), Payload(/x)
2
POST /x [CON, MID=0x20,Token, Payload(EAP Req/Id)]

Keeping OSCORE to confirm keys in CoAP-EAP

- After a design meeting with Christian we arrived to the conclusion that OSCORE can be maintained, as originally intended
    - An OSCORE message can be treated as alternate success indication
    - An OSCORE security context can be pre-defined, leaving the key to be completed after the EAP success is processed and the MSK is retrieved to complete security context
    - Recipient and Sender ID are now sent in Steps 1 and 2

Current flow of operation



IoT device/ CoAP Server

Controller/ CoAP client

0 **POST /.well-known/coap-eap** [MID=50,Token, Options(No-response), Payload(/x)]

1 **POST /x** [MID=0x20,Token, Payload(EAP Req/Id || CS || SID)]

2 **ACK** [MID= 0x20,Token, 2.01 Created, Options(Location-Path(/y), Payload(EAP Res/Id || CS || RID)]

3 **POST /y** [MID=0x32,Token, Payload(EAP-Req X 1)]

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

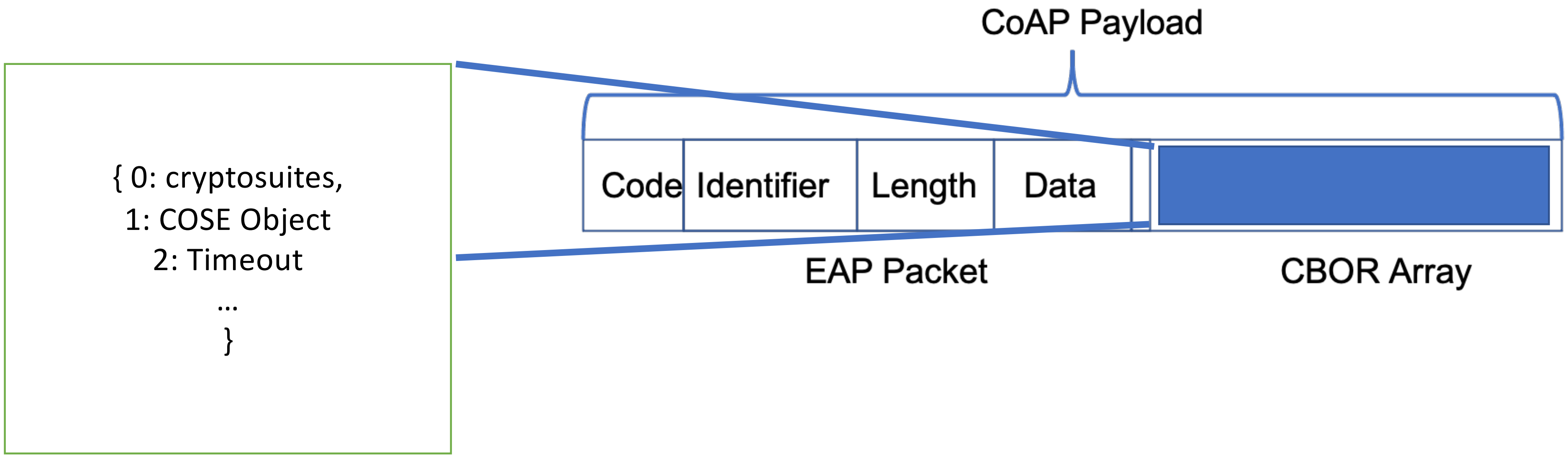
5 **POST** /z [MID=3, Token, Payload(EAP-X-Req n-1)]

6 **ACK** [MID=43, Token, 2.01 Created, Location-Path(/w), Payload(EAP-X-Res n)]

7 **POST** /w [MID=4,Token, Option(**OSCORE**) Payload(EAP Success || Session-Lifetime)]

8 **ACK** [MID=43, Token, 2.01 Created, Location-Path(/t), Options(**OSCORE**)

Tagged CBOR structure



{ 0: cryptosuites,
1: COSE Object
2: Timeout
…
}

CoAP Payload

Code Identifier Length Data

EAP Packet

CBOR Array

Extensible CBOR structure

THANK YOU