
# Object Security for CoAP

draft-selander-ace-object-security-01

Göran Selander, Ericsson
John Mattsson, Ericsson
Ludwig Seitz, SICS

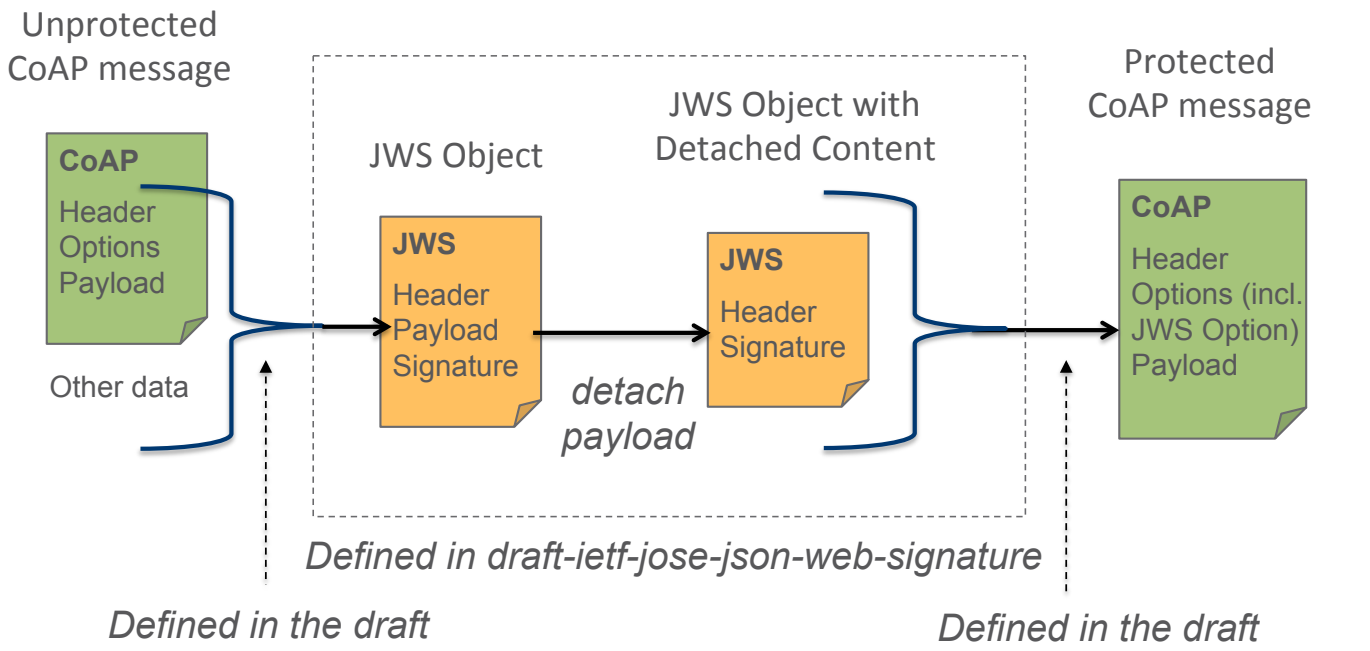
IETF 92 ACE WG, Dallas, March 24, 2015

# Background

- End-to-end security between endpoints; encryption by default
  - Intermediary nodes; proxying, storing-and-forwarding, caching, pubsub-brokering, . . .
- Object security as a complement to, or in conjunction with, DTLS
  - Suitable for constrained environments
- Version -00 focused on integrity and replay protection of CoAP messages

# Version -00
# Signature of CoAP message, using JWS

- Integrity and replay protection of CoAP message
- New CoAP Option
  Containing a JWS (signature) of the message
- Other data:
  Used in response to verify freshness




*Subset of CoAP Header, Options, Payload and other data wrapped in a JWS Object*

# Main changes in version -01

- Rewritten
- Renamed CoAP Option (“JWS” → ”Sig”)
- Added support for encryption (AEAD)
- Added a new CoAP Option (“Enc”) indicating encrypted message
- New parameter “mode”, allowing application specific profiling
  - Mode:COAP == Protection of CoAP message exchange.
  - Mode:APPL == Protection of payload only.
- Object format not restricted to JOSE (COSE or other formats can be applied)
  - Definition of generic “Secure Message”, customizable using “mode”
  - Estimates of Secure Message sizes with JOSE and COSE
  - Estimate of lower bound for Secure Message size

# End-to-end security considerations

- To protect against eavesdropping and manipulation of resource representations;
- To protect transport of authorization information ("access tokens");
- To protect from replaying old messages to be passed as a new message;

Application layer protection

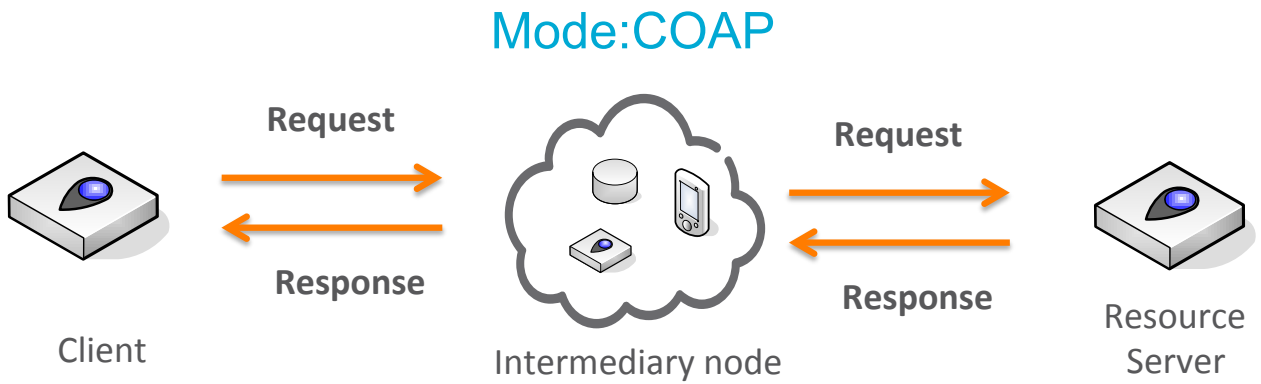
- To allow a client to verify that a response comes from a certain server and is the response to a particular request;
- To protect RESTful method used by the client, or response code by the server. E.g. if a forward proxy replaced the client requested GET with a DELETE then this must be detected by the server;
- To protect against eavesdropping of meta-data of request or response, including CoAP options such as Uri-Path and Uri-Query, which may reveal some information of what is requested etc

CoAP message protection

# Different modes

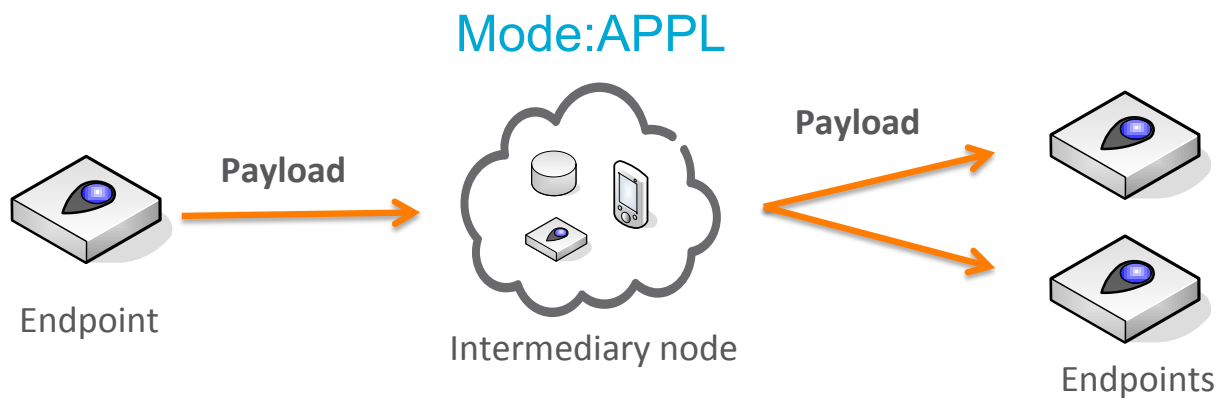
## Mode:COAP

- Point-to-point
- CoAP message
- Replay protection
- Challenge-response
- Forward proxy



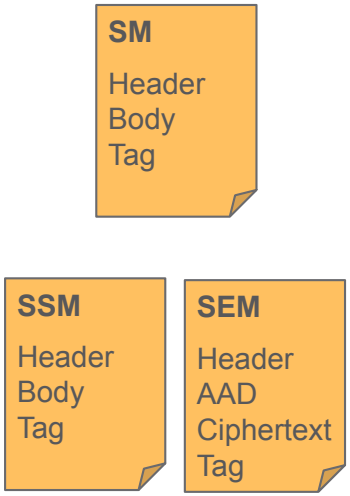

## Mode:APPL

- Point-to-multipoint
- Application layer data
- Replay protection
- Caching/PubSub



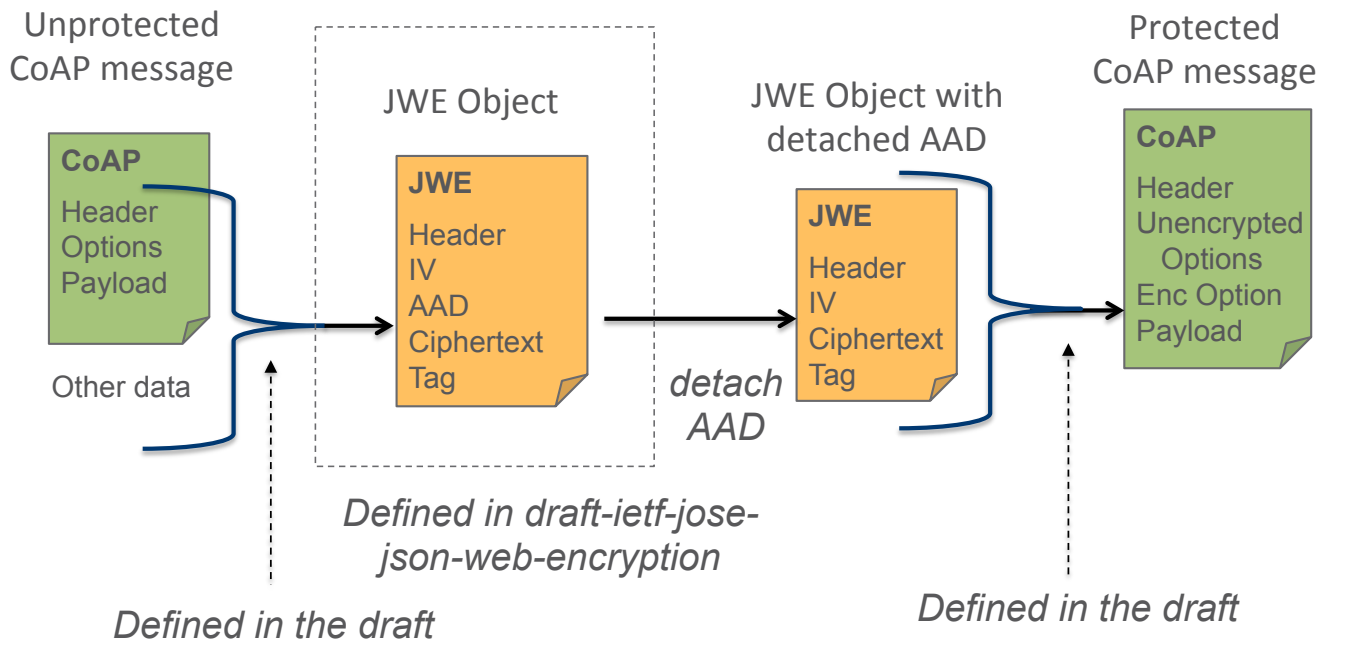
# Secure Message

- A Secure Message (SM) consists of Header, Body and Tag
  - Generalization of JOSE, COSE, …
  - SEM, SSM analogously to JWE, JWS respectively
- Header
  - **Algorithm**: Cipher suite. Similar to “alg” in JWS, “enc” in JWE
  - **Key Identifier**: Identifies sender security context/key(s). Syntax similar to JOSE “kid”
  - **Sequence Number**: Enumerating messages signed with a key identified by the Key Identifier. New JOSE Header Parameter
  - **Mode**: Application specific message format, content and processing. New JOSE Header Parameter
- Purpose of SM is as placeholder for an optimized format yet-to-be-defined, extended with the new header parameters.



# Encryption of CoAP Message, using SEM=JWE

- Encryption, integrity and replay protection of CoAP message
- New CoAP Option called “Enc”: Indicating presence of an encrypted object (here a JWE)
- Other data: Used in response to verify freshness




*Subset of CoAP Header, Options, Payload and other data wrapped in a JWE Object*

# Message Sizes

› JWS

Header: {"alg":"HS256", "kid":"a1534e3c5fdc09bd", "seq":"00000142", "mod":"0”}

› JWE

- Header: {"alg":"dir", "kid":"a1534e3c5fdc09bd", "enc":"A128GCM", "mod":"0"}
- IV contains sequence number

| Scheme | Header | MAC | Over-head |
|---|---|---|---|
| **JWS** | 90 B | 43 B | 135 B |
| **COSE** | 35 B | 32 B | 70 B |
| **Lower bound** | 12 B | 16 B | 28 B |

| Scheme | Header | IV | MAC | Over-head |
|---|---|---|---|---|
| **JWE** | 86 B | 16 B | 22 B | 127 B |
| **COSE** | 40 B | 12 B | 16 B | 70 B |
| **Lower bound** | 12 B | 0 B | 8 B | 20 B |

“Lower bound” is estimated with CSM format (Appendix C).

# Implementation (work in progress)

- Variant of 00-version
  - Mode:COAP
  - CSM
- Erbium REST Engine for Contiki
- TI CC2538 (32 bit processor, 32 Kbyte RAM, 512 Kbyte flash)
  - AES_CCM_8 in software
- First measurements: The impact on required RAM, flash, and on processing is essentially due to crypto – not due to added code for message parsing, etc.


# Thank you!
# Questions?

# Reading hints

- End-to-end security considerations (Sec. 2-3)
  - End-to-end security between endpoints in the presence of intermediary nodes.
- Message format (Sec. 4, App. B-C)
  - Encryption, integrity protection and replay protection.
  - Focus on AEAD in this version of the draft.
- CoAP layer protection (Sec. 5.1, App. A)
  - New CoAP Options
  - Client-server challenge-response protocol.
- Application layer protection (Sec. 5.2)
  - Point-to-point/multipoint, e.g. caching, publish-subscribe.
- Examples (App. D)

# Lower bound for SM (CSM)

- Binary format, tailor-made to estimate minimum size of SM
- Header
  - M = Mode
  - ALG = Encoding of the ciphersuite
  - Variable length of KID and SEQ, encoded in KL and SL
  - Fixed fields: 2 bytes
  - Total header: ≥ 4 bytes
- Body
  - Duplicated content detached
- Tag
  - Size of (truncated) MAC