Object Security for Constrained RESTful Environments (OSCORE)

Motivation for OSCORE (former OSCOAP)

- CoAP (RFC 7252) is designed for proxy operations, e.g. to support intermittent connected endpoints
- These proxy operations require certain CoAP headers and options to be read by the proxy
  - Forwarding
  - Blockwise or other CoAP options between intermediate nodes
- This require selective protection of CoAP headers and options
- Not possible with DTLS on transport layer or application layer

OSCORE Recap

- An extension to CoAP (the Object-Security option)
- An application layer security protocol providing end-to-end security
  - Provides encryption, integrity and replay protection, and binds responses to requests
- Faithful to CoAP – supports options and proxy forwarding
  - Including Observe (RFC 7641), Blockwise (RFC 7959), Patch & Fetch (RFC8132), CoAP over TCP (work in progress), …
- Works wherever CoAP works (UDP, TCP, SMS, NB-IoT, 802.15.4 IE, ...)
- Secures end-to-end REST, e.g. translations between HTTP and CoAP
  - OCF requested end-to-end security for REST where one endpoint is HTTP and the other endpoint is CoAP, through an HTTP-CoAP proxy, or v.v. The change suggested by Dave Thaler has been included in the latest version of OSCORE.
- OSCORE works with CoAP over IP multicast.
  - OSCORE secures group communication with multiple senders and unicast reponses.
  - Fairhair alliance is considering OSCORE for their group communication use case.

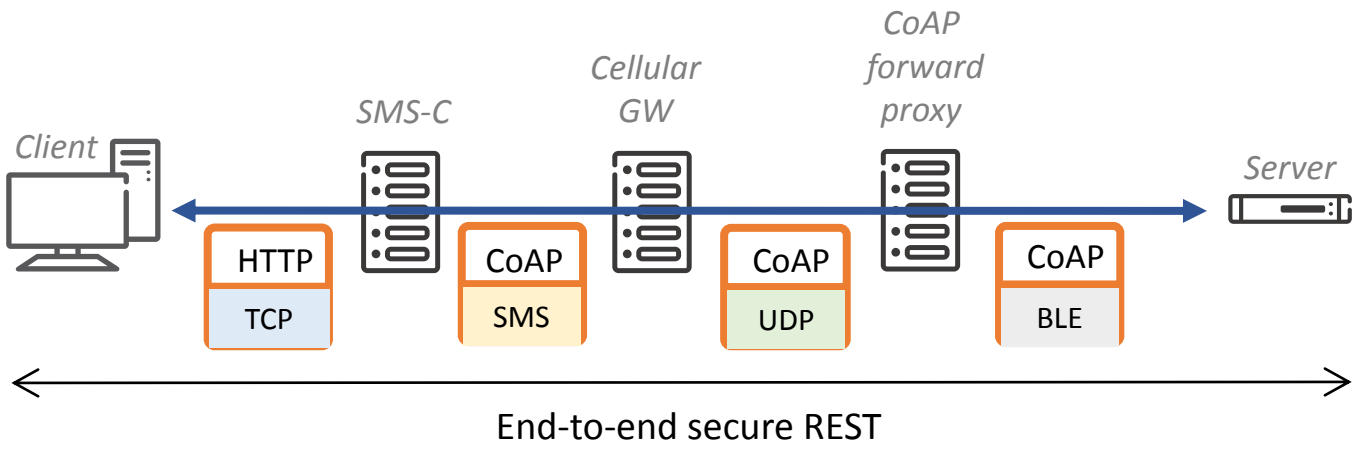
Standardization and Implementation

- Ready for WGLC in IETF CoRE: https://tools.ietf.org/html/draft-ietf-core-object-security-06
- Used by IETF 6tisch: https://tools.ietf.org/html/draft-ietf-6tisch-minimal-security
  - The compatibility with CoAP proxy operation provided by OSCORE is used e.g. in IETF 6tisch WG to support security through stateless proxies in multi-hop low power networks based on IEEE 802.15.4e.
- Considered for inclusion in OCF, OMA DM LwM2M, Fairhair Alliance
- Several implementations: C, Java, Python
- Open source for Contiki, Californium, OpenWSN
- Interop testing at IETF Hackathon, ETSI F-Interop: https://ericssonresearch.github.io/OSCOAP/

Status Summary

- OSCORE is stable (CoRE working group last call)
- It has been independently implemented and interop tested
- Now works even when HTTP endpoints are used (thanks Dave!)

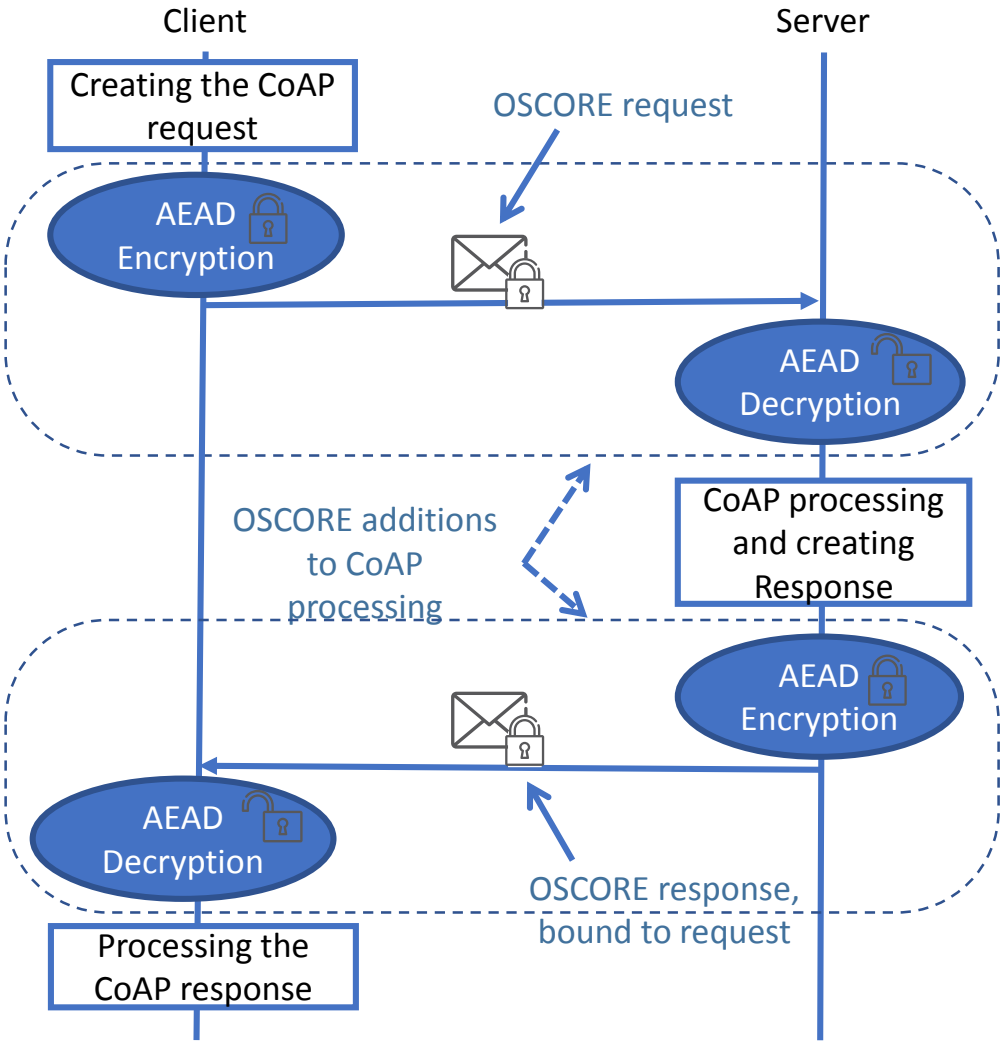
OSCORE Example



NOTE: CoAP/OSCORE may be used to secure operations and responses transported on top of arbitrary application layer protocols.

OSCORE Processing

- Addition to CoAP
- Uses Authenticated Encryption with Additional Data (AEAD)
- AES-128-CCM-8 mandatory to implement
- Protection of CoAP messages using the COSE format (COSE_Encrypt0)
- Replay protection
- Handling partial loss of security context



OSCORE protection 1(3)

- CoAP has a Messaging Layer and a Request/Response Layer (RFC 7252)
- The messaging layer handles UDP and asynchronous interactions
- CoAP over TCP replaces the messaging layer with a framing mechanism
- OSCORE only protects the request/response layer and therefore works with both UDP and TCP

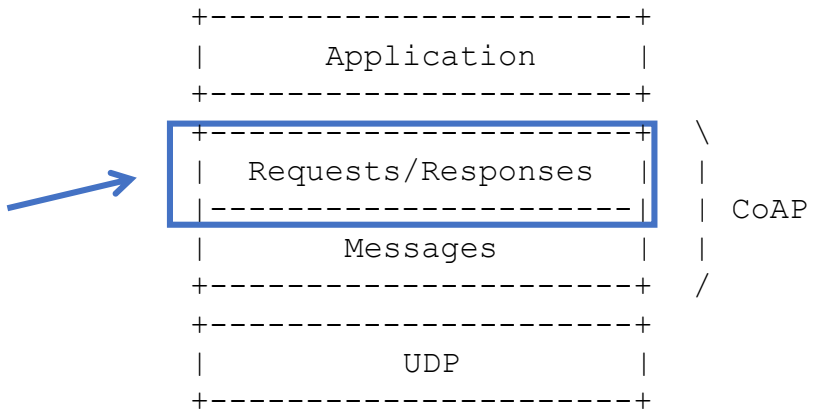
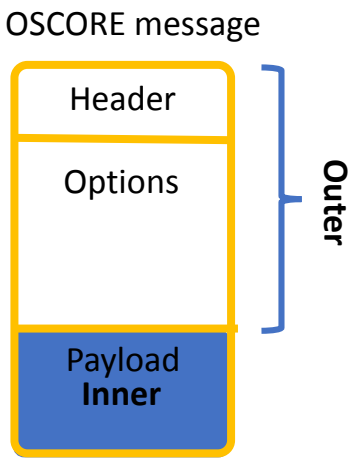


Figure 1: Abstract Layering of CoAP

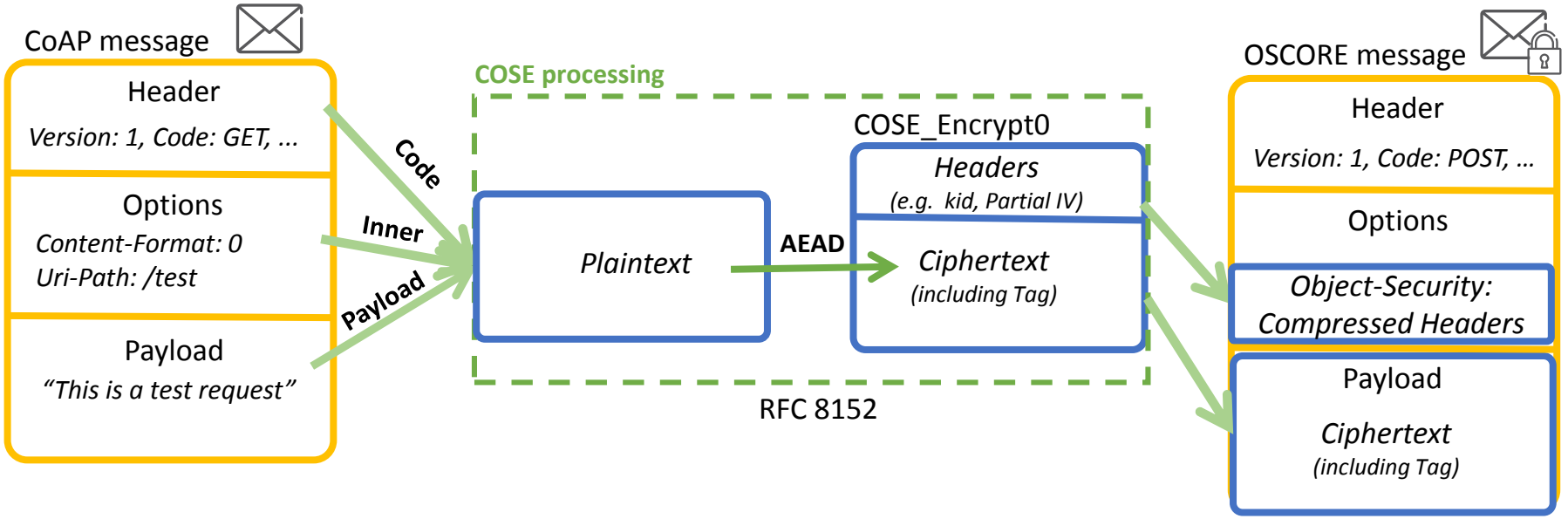
RFC 7252

OSCORE protection 2(3)

- CoAP (RFC7252) is designed for proxy operations (e.g. store and forward to sleepy devices, translations, etc.) and specifies how CoAP proxies operate
- E2E security and proxy operations are in certain cases conflicting
- OSCORE supports CoAP proxy operations as long as it is not prevented by end-to-end security
- To support end-to-end security as well as proxy operations, OSCORE enables CoAP message fields to be Inner or Outer
  - Inner message fields are encrypted and integrity protected, and transported in the CoAP Payload. These message fields are intended for the other endpoint.
  - Outer message fields are unprotected and transported in CoAP Header or Option of the OSCORE message. These are intended for a proxy.



Generating the OSCORE Message



OSCORE protection 3(3)

- CoAP Payload is Inner
- CoAP Options are Inner unless needed by a proxy, see table. The Outer options are used:
  - x – to support proxy forwarding
  - x – to support fragmentation of OSCORE messages ("Outer Blockwise")
  - x ("Observe"/"Max-Age") – by legacy proxies
- CoAP Header fields are Outer; "Code" is also Inner
  - The Code of the original CoAP message is encrypted
  - The OSCORE message has a dummy Code
- New options define its OSCORE processing
  - By default Inner

No.	Name	E	U
1	If-Match	x	
3	Uri-Host		x
4	ETag	x	
5	If-None-Match	x	
6	Observe		x
7	Uri-Port		x
8	Location-Path	x	
11	Uri-Path	x	
12	Content-Format	x	
14	Max-Age	x	x
15	Uri-Query	x	
17	Accept	x	
20	Location-Query	x	
23	Block2	x	x
27	Block1	x	x
28	Size2	x	x
35	Proxy-Uri	x	x
39	Proxy-Scheme		x
60	Size1	x	x

E = Encrypt and Integrity Protect (Inner)
U = Unprotected (Outer)

The Object-Security Option

No.	C	U	N	R	Name	Format	Length	Default
TBD	x				Object-Security	(*)	0-255	(none)

C = Critical, U = Unsafe, N = NoCacheKey, R = Repeatable

(*)

- Partial IV† is set to Sequence Number
- kid† is set to Sender ID
- Context Hint is used by the CoAP server find the relevant security context

† Defined in COSE (RFC 8152)

Key establishment

- OSCORE profile for ACE
  https://tools.ietf.org/html/draft-seitz-ace-oscoap-profile
- TLS 1.3 handshake over CoAP
  https://tools.ietf.org/html/draft-mattsson-ace-tls-oscore
- EDHOC – key exchange built on COSE, CBOR and CoAP
  - PSK, RPK, X.509 Certificates
  - 50-75% message overhead for PSK/RPK compared to TLS 1.3

  https://tools.ietf.org/html/draft-selander-ace-cose-ecdhe

Group OSCORE

- OSCORE extends to group communication
- Securing RFC7390
- Used e.g. for building automation