OSCORE-capable Proxies

draft-tiloca-core-oscore-capable-proxies-04

Marco Tiloca, RISE
Rikard Höglund, RISE

CoRE WG Interim Meeting, September 28th, 2022

Recap

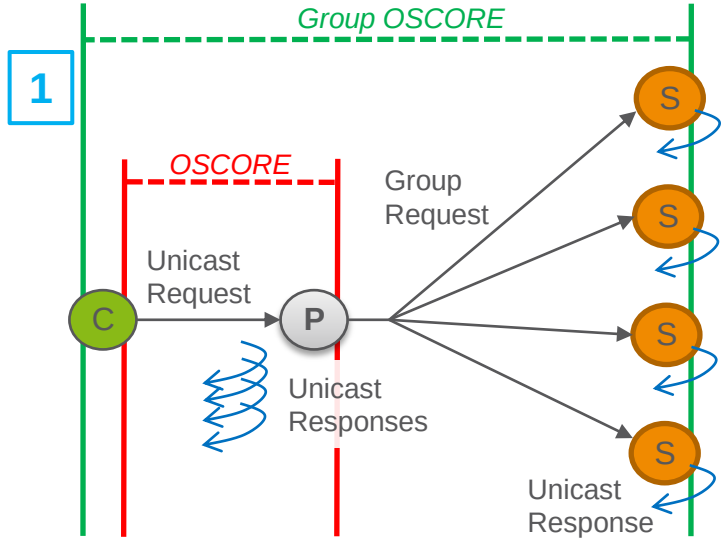
- **A CoAP proxy (P) can be used between client (C) and server (S)**
  - A security association might be required between C and P --- use cases in next slides
- **Good to use OSCORE between C and P**
  - Especially, but not only, if C and S already use OSCORE end-to-end
- **This is not defined and not admitted in OSCORE (RFC 8613)**
  - C and S are the only considered “OSCORE endpoints”
  - It is forbidden to double-protect a message, i.e., both over C ↔ S and over C ↔ P
- **This started as an Appendix of *draft-tiloca-core-groupcomm-proxy***
  - Agreed at IETF 110 [1] and at the June 2021 CoRE interim [2] to have a separate draft

[1] https://datatracker.ietf.org/doc/minutes-110-core-202103081700/

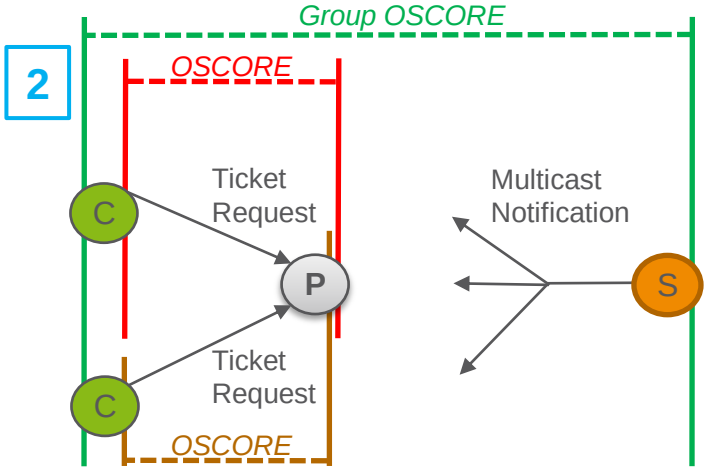
[2] https://datatracker.ietf.org/doc/minutes-interim-2021-core-07-202106091600/

Some use cases

1. **CoAP Group Communication with Proxies**
   - *draft-tiloca-core-groupcomm-proxy*
   - CoAP group communication through a proxy
   - P must identify C through a security association



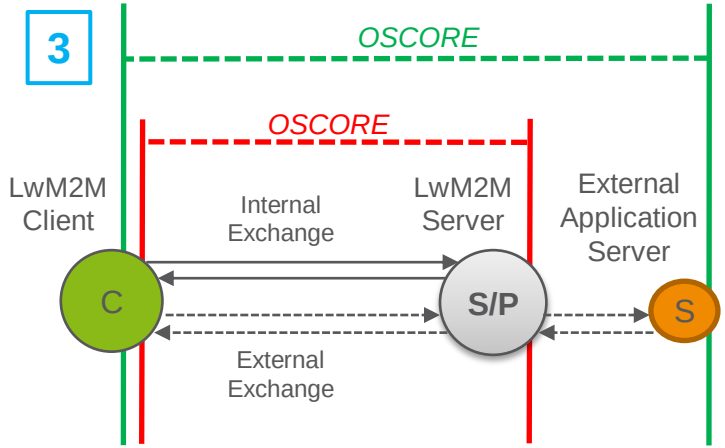
2. **CoAP Observe Notifications over Multicast**
   - *draft-ietf-core-observe-multicast-notifications*
   - If Group OSCORE is used for e2e security …
   - … C provides P with a Ticket Request obtained from S
   - That provisioning should be protected over C ↔ P



Some use cases

3. LwM2M Client and external Application Server

- From the *L2wM2M Transport Binding* specification:
  - OSCORE can be used between a LwM2M endpoint and a non-LwM2M endpoint, via the LwM2M Server
- The LwM2M Client may use OSCORE to interact:
  - With the LwM2M Server (LS), as usual; and
  - With an external Application Server, via LS acting as proxy



More use cases are discussed in the draft

Contribution

- **Twofold update to RFC 8613**

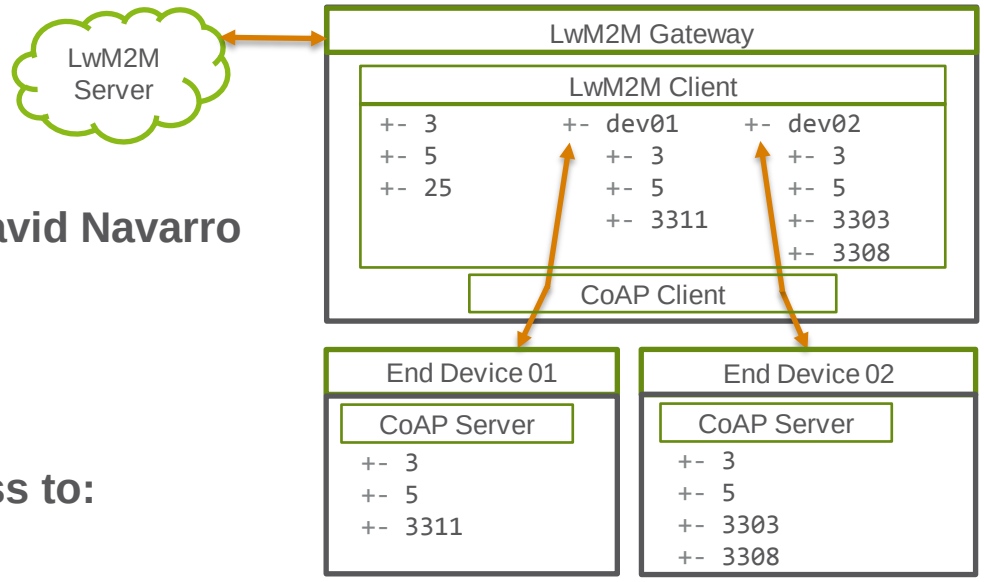
1. **Define the use of OSCORE in a communication leg including a proxy**
   - Between origin client/server and a proxy; or between two proxies in a chain
   - Not only an origin client/server, but also an intermediary can be an “OSCORE endpoint”

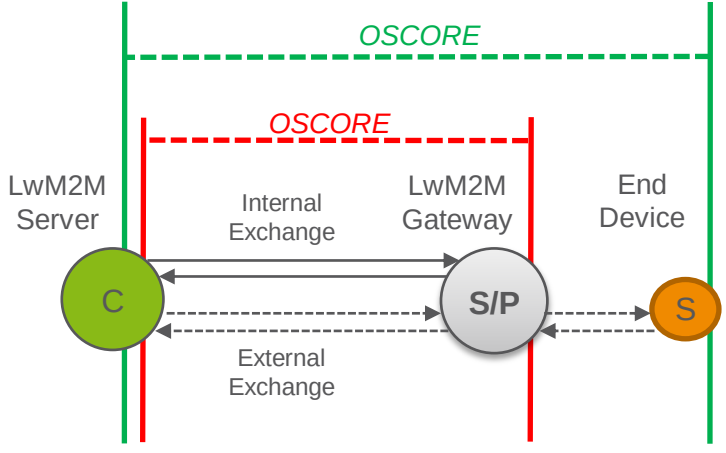
2. **Explicitly admit nested OSCORE protection – “OSCORE-in-OSCORE”**
   - E.g., first protect end-to-end over C ↔ S, then further protect the result over C ↔ P
   - Typically, at most 2 OSCORE “layers” for the same message
     - 1 end-to-end + 1 between two adjacent hops
   - Possible to seamlessly apply 2 or more OSCORE layers to the same message
     - Building block for “OSCORE-protected Onion Forwarding”, see Appendix B

- **Focus on OSCORE, but the same applies “as is” to Group OSCORE**

Since v -01

- **Added new use case suggested by David Navarro**
- **Use of the LwM2M Gateway**
- **Provide the LwM2M Server with access to:**
  a) Resources at the LwM2M Gateway
  b) Resources at external End Devices, through the LwM2M Gateway, via dedicated URI paths
- **In case (b), the LwM2M Gateway acts, at its core, as a reverse-proxy**





Since v -01

Revised definition of “proxy-related options”

- Proxy-URI Option
- Proxy-Scheme Option together with any of the Uri-* Options

Forward-proxying

- Uri-Path Options, if present not together with Proxy-Scheme

Reverse-proxying

Since v -01

Revised set of CoAP options to encrypt, as if they were of class E for OSCORE

- Let's say that an outgoing message is being protected for an OSCORE endpoint *X*
  - The sender endpoint is applying the *i*-th OSCORE layer, to be consumed by *X*
  - The following options are encrypted, regardless of their original class for OSCORE
- OSCORE Option, when present before encryption
  - That is, added when applying the previous OSCORE layer
- EDHOC Option, when NOT intended to *X*
- Options intended to *X*, but not relevant for pre-decryption processing or for removing the *i*-th layer --- This prevents from encrypting the EDHOC Option when intended to *X*
  - Proxy-Uri, Proxy-Scheme, Uri-Host, Uri-Port
  - Listen-To-Multicast-Notifications
  - Multicast-Timeout, Response-Forwarding, Group-ETag

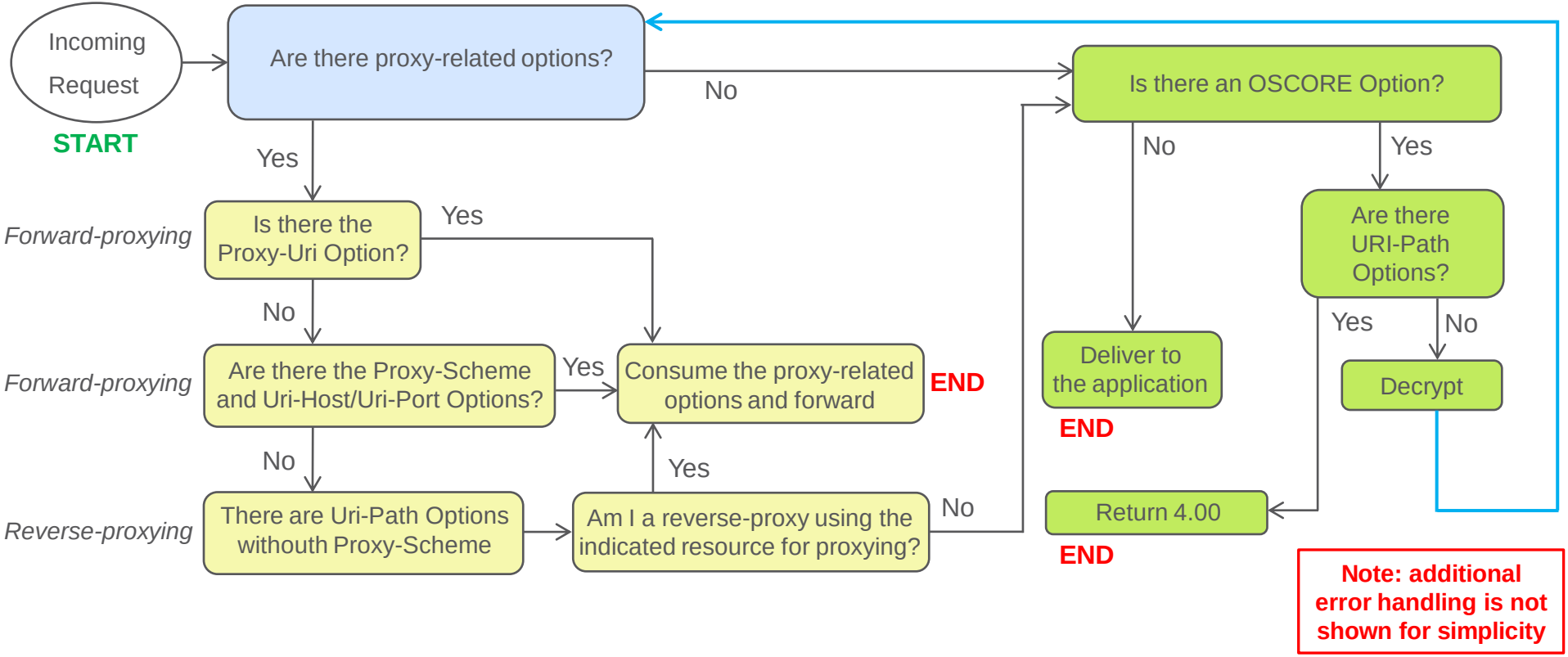
Since v -01

Revised message processing

- **Updated processing of incoming requests**
  - Some simplifications, based on new definitions of options to encrypt
  - Covered also the case related to reverse-proxying
  - Algorithm presented as three steps to navigate (including jumping and looping back)
    - 1) Is this about proxying? ; 2) Perform proxying ; 3) Consume or decrypt

- **Anything else has remained the same**
  - Processing of outgoing requests
  - Processing of outgoing responses
  - Processing of incoming responses

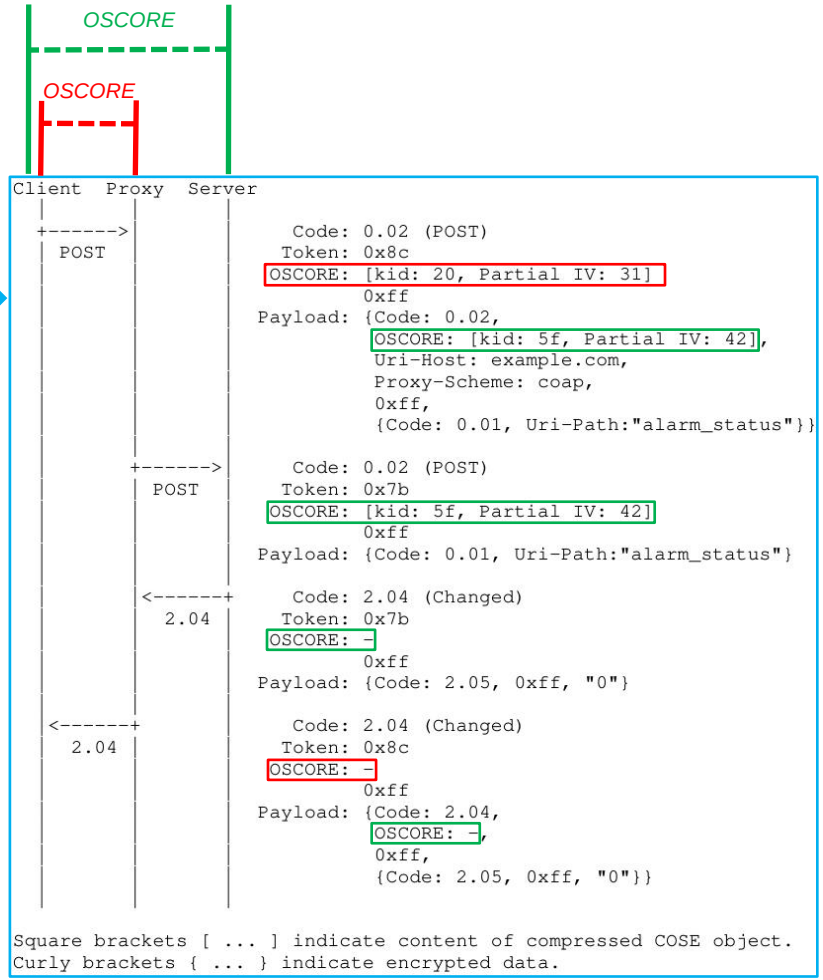
Processing an incoming request



Since v -01

Added examples (Appendix A)

1. **OSCORE used for C↔S and C↔P**
   - Pre-established Security Contexts
2. **OSCORE used for C↔S and P↔S**
   - Pre-established Security Contexts
3. **OSCORE used for C↔S, C↔P and P↔S**
   - Pre-established Security Contexts
4. **OSCORE used for C↔S and C↔P**
   - Security Contexts established with EDHOC
   - https://datatracker.ietf.org/doc/draft-ietf-lake-edhoc/



Since v -01

- **Added Section 4 on cacheability of OSCORE-protected responses**
  - Use of the approach defined in [3], based on OSCORE Deterministic Requests
  - A proxy looks for a cache hit, using the exact request to forward
  - A proxy caches the exact response to forward back

Before a possible, further encryption

- **Added Appendix B – "OSCORE-protected Onion Forwarding"**
  - Case in point for protecting a message with 2+ OSCORE layers
  - Kind-of mimicking the message protection in Tor, but using OSCORE
  - Currently a list of raw bullet points, to be better elaborated/presented
  - To be considered: later extract this content to be a separate Experimental draft

[3] https://datatracker.ietf.org/doc/draft-amsuess-core-cachable-oscore/

Summary and next steps

- **Proposed update to RFC 8613**
  - Define the use of OSCORE in a communication leg including a proxy
  - Explicitly admit nested OSCORE protection – “OSCORE-in-OSCORE”

- **Next steps**
  - Expand on possible corner cases, as dictated by the semantics of specific options
  - Add guidelines on establishment of Security Contexts – The detailed method is out of scope
  - Revised processing of incoming responses – Following pending updates to Group OSCORE
  - Add more examples: use of EDHOC optimized workflow; use of a reverse-proxy
  - Look into CoAP header compression from RFC 8824. Need for any adaptations?

- **The core mechanics is stable – Comments and input are welcome!**

Thank you!

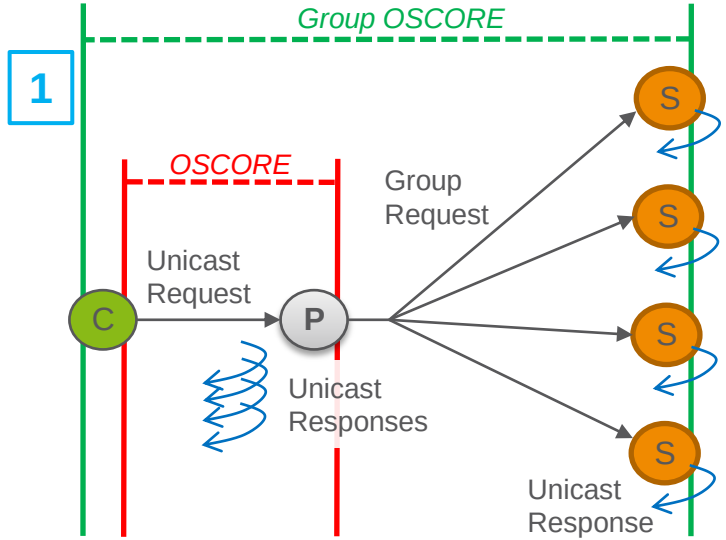
Comments/questions?

https://gitlab.com/crimson84/draft-tiloca-core-oscore-to-proxies

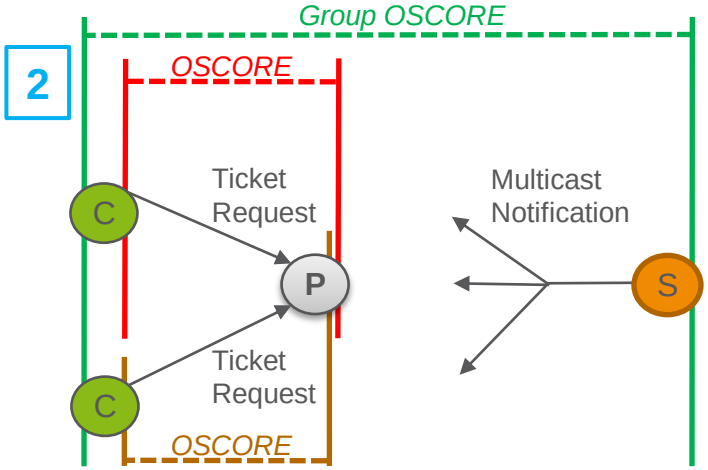
Backup

Some use cases

1. **CoAP Group Communication with Proxies**
   - *draft-tiloca-core-groupcomm-proxy*
   - CoAP group communication through a proxy
   - P must identify C through a security association



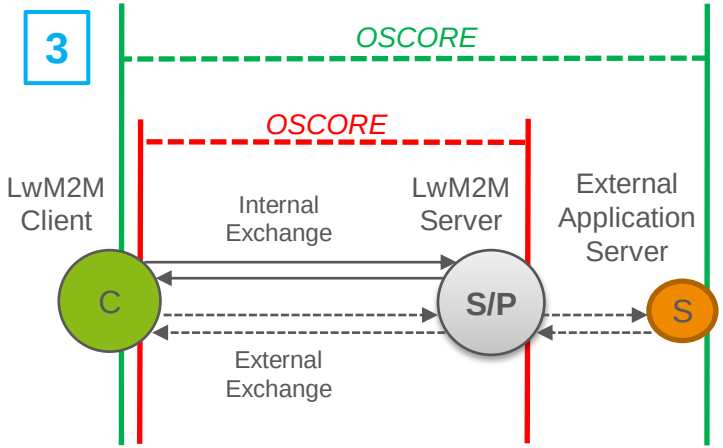
2. **CoAP Observe Notifications over Multicast**
   - *draft-ietf-core-observe-multicast-notifications*
   - If Group OSCORE is used for e2e security …
   - … C provides P with a Ticket Request obtained from S
   - That provisioning should be protected over C ↔ P



Some use cases

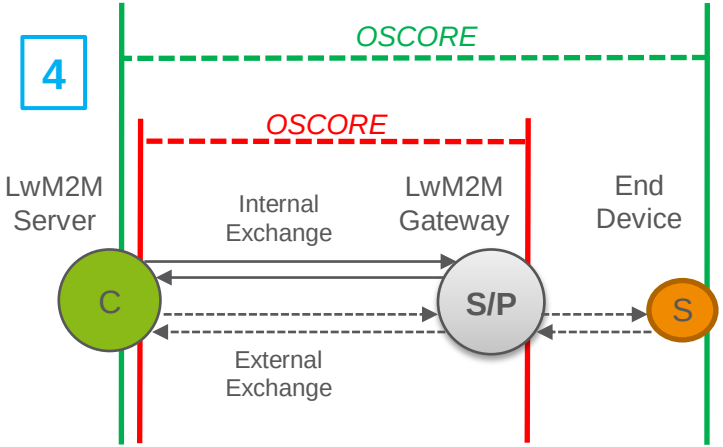
3. LwM2M Client and external Application Server

- From the *L2wM2M Transport Binding* specification:
  - OSCORE can be used between a LwM2M endpoint and a non-LwM2M endpoint, via the LwM2M Server
- The LwM2M Client may use OSCORE to interact:
  - With the LwM2M Server (LS), as usual; and
  - With an external Application Server, via LS acting as proxy



4. Use of the LwM2M Gateway

- It provides the LwM2M Server with access to:
  - a) Resources at the LwM2M Gateway
  - b) Resources at external End Devices, through the LwM2M Gateway, via dedicated URI paths
- In case (b), the LwM2M Gateway acts, at its core, as a reverse-proxy



Some use cases

- **OMA LwM2M Client and External Application Server**
  - *Lightweight Machine to Machine Technical Specification – Transport Binding*

> *OSCORE MAY also be used between LwM2M endpoint and non-LwM2M endpoint, e.g., between an Application Server and a LwM2M Client via a LwM2M server. Both the LwM2M endpoint and non-LwM2M endpoint MUST implement OSCORE and be provisioned with an OSCORE Security Context.*

  - The LwM2M Client may register to and communicate with the LwM2M Server using OSCORE
  - The LwM2M Client may communicate with an External Application Server, also using OSCORE
  - The LwM2M Server would act as CoAP proxy, forwarding traffic outside the LwM2M domain

Processing an incoming request