OSCORE-capable Proxies

draft-tiloca-core-oscore-capable-proxies-06

Marco Tiloca, RISE
Rikard Höglund, RISE

CoRE WG Interim Meeting, June 7th, 2023
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 work 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

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 (from David Navarro)
   - 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
**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
   - But 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**
Recent updates

› Previous presentation: version -04 at the 2022-09-28 CoRE interim meeting

› Then two close submissions of -05 and -06 around IETF 116

› The proxy has to check whether forwarding a decrypted request is ok
  – For example, leveraging the OSCORE Security Context used for decryption
  – Input from Christian during the CoRE interim meeting on 2022-09-28 – Thanks!

› Clarified corner case where the proxy does not forward a valid request
  – For example, if including the Listen-To-Multicast-Notifications Option [1]

Processing an incoming request

START

Incoming Request

Are there proxy-related options?

Yes

Is there the Proxy-Uri Option?

Yes

Are there the Proxy-Scheme and Uri-Host/Uri-Port Options?

Yes

Is forwarding this request an authorized operation?

Yes

Consume the proxy-related options and forward

No

Return 4.01

END

No

Return 4.00

END

No

Are there URI-Path Options?

Yes

Deliver to the application

No

Decrypt

END

END

No

Is there an OSCORE Option?

Yes

Are there URI-Path Options?

No

Return 4.01

END

END

Note: additional error handling is not shown for simplicity

Forward-proxying

No

Am I a reverse-proxy using the indicated resource for proxying?

Yes

Determine if proxying or not

Proxying

No

Decompress; OR decrypt and repeat

NEW!

Reverse-proxying

There are Uri-Path Options without Proxy-Scheme

Yes

Determine if proxying or not

Proxying

No

Consume; OR decrypt and repeat
Recent updates

› OSCORE protection of CoAP options
  - If a CoAP option is originally defined as class U or I for OSCORE ...
  - ... when should it be protected as if it was of class E?

› Improved general rules
  - Now better covering corner cases and class I options
  - A good sanity check was the Request-Hash option [2]
    › Processed as class I in responses
    › Expected to be elided from responses, but still possible to send it on the wire

› Current rule formulation – Section 3.1
  - Three main cases, all formulated as “Any CoAP option such that ...”
  - Multiple examples provided for each case
  - The rationale is always to encrypt as many options as possible
Encryption of Class U/I Options

I have an outgoing message M, which includes an option OPT.

Did I add OPT to M?

OPT is of class U or I. How do I process it with OSCORE?

Is OPT intended to be consumed by X?

Does X need to access OPT before decryption or to perform decryption?

Have I added OPT after a previous encryption to an endpoint different than X, where OPT was treated as class U/I?

Have I added OPT before a previous encryption to an endpoint different than X, where OPT was treated as class U/I?

This can happen only at a proxy 

Sender OSCORE endpoint

Process OPT as class E

Process OPT as per its original class U or I
Recent updates

› New Section 5 – Guidelines on establishing OSCORE Security Contexts
  – Generally agnostic of the used establishment method

› For OSCORE
  – Guidelines for the client using EDHOC [3], first with the proxy, then with the origin server
  – Reference to the possible, optimized EDHOC workflow [4]

› For Group OSCORE
  – Expected between origin client and servers; they rely on the Group Manager
  – If a proxy uses Group OSCORE, it must not be in the same group of the origin endpoints

Recent updates

› Revised notation in the examples of message exchange – Appendix A
  – Easier to see what is encrypted
  – Easier to see which OSCORE Security Context is used

› New example with EDHOC [3] and the EDHOC + OSCORE request [4]
  – See Appendix A.5 – The improvement is bigger than intuitively expected
  – Use EDHOC between C and P as well as between C and S (through P)
  – How many messages to: i) complete EDHOC with P and with S; and ii) exchange data with S?
  – Without optimization (Appendix A.4): 16 messages
  – With optimization (Appendix A.5): 10 messages

Open point #1

› Appendix B “OSCORE-protected Onion Forwarding”
  – The origin client protects its request first for the origin server, ...
  – ... then for the last proxy, then for the second from last proxy, ... , then for the first proxy
  – That can become something similar to TOR, but using OSCORE

› At a high level, the use case is described earlier in the draft
  – The text in Appendix B is currently a collection of notes and directions
  – The foundation looks promising, but it is unlikely to be ready and used any time soon

› Proposal
  – Remove Appendix B from the current Internet Draft
  – Reuse its content for a separate, Experimental Internet Draft building on the current one

Thoughts?
Open point #2

› Revising the rules on protecting CoAP options put light on RFC 8798
  – For the Hop-Limit option, no OSCORE processing and class is defined

› From RFC 8613, Section 4.1
  – *Options that are unknown or for which OSCORE processing is not defined SHALL be processed as Class E (and no special processing).*

› That is, Hop-Limit has to be treated as a class E option
  – If the origin client adds the option, encrypting it is not desirable (even w/o OSCORE with proxy)
  – You would have an inner option and outer option, with the inner one not useful

› Proposal
  – In this Internet Draft, define that the Hop-Limit Option is of class U (i.e., update RFC 8798)
  – Per the protection rules: the option is unprotected end-to-end, but protected for the next proxy

Thoughts?
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 step: submit version -07 before the IETF 117 cut-off
  – More, newly identified use cases: -core-coap-pm and -ace-coap-est-oscore
  – High-level use of SCHC header compression (see RFC 8824 & draft-tiloca-schc-8824-update)
  – New appendix with ASCII-art figure for the processing of incoming requests
  – Address the two raised open points (see slides 13 and 14)
  – Minor editorial fixes

› Version -07 should be ready for considering a WG Adoption Call

› The core mechanics are already stable – Comments are welcome!
Thank you!

Comments/questions?

https://gitlab.com/crimson84/draft-tiloca-core-oscore-to-proxies
Backup
Some use cases – LwM2M

› OMA LwM2M Client and External Application Server

  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

1. Are there proxy-related options?
   - Yes: Is there the Proxy-Uri Option?
     - Yes: Am I a forward-proxy?
       - Yes: Return 5.05
       - No: Is forwarding this request an authorized operation?
         - Yes: Consume the proxy-related options and forward
         - No: Return 4.01
     - No: Return 4.00
   - No: Is there an OSCORE Option?
     - No: Are there URI-Path Options?
       - Yes: Deliver to the application
       - No: Decrypt
     - Yes: Success? (Yes)
     - No: OSCORE error handling

2. Forward-proxying
   - Is there the Proxy-Uri Option?
     - Yes: Are there the Proxy-Scheme and Uri-Host/Uri-Port Options?
       - Yes: Am I a reverse-proxy using the indicated resource for proxying?
         - Yes: Consume; OR decrypt and repeat
         - No: End
       - No: End
     - No: End
   - No: End

3. Reverse-proxying
   - There are Uri-Path Options without Proxy-Scheme
     - Determine if proxying or not

CoRE WG Interim Meeting | 2023-06-07 | Page 18