OSCORE-capable Proxies

draft-ietf-core-oscore-capable-proxies-01

Marco Tiloca, RISE
Rikard Höglund, RISE

IETF 119 Meeting – Brisbane – March 20th, 2024
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. Define rules to escalate the protection of CoAP options
   › If possible, encrypt and integrity-protect an option originally defined as Class U or I for OSCORE

3. Explicitly admit a 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

› Focus on OSCORE, but the same applies “as is” to Group OSCORE
Since IETF 118

› Received comments from Christian Amsüss [1] and Göran Selander – Thanks!

› Submitted version -01 before the cut-off for IETF 119

› Summary of latest updates
  – Updated and added references
  – Various editorial fixes and readability improvements
  – Fixed notation in the examples of Appendix A
  – Onion CoAP [2] mentioned as use case
  – Considered also the CoAP options Proxy-Cri and Proxy-Scheme-Number [3]
  – Revised escalation of CoAP option protection
  – Revised processing of incoming requests

Details in the next slides

[1] https://mailarchive.ietf.org/arch/msg/core/9sPP9cAMDO5GFwZ4XeJng_bSnwQ/
Escalation of CoAP Option Protection

› Now listed as a point of update to RFC 8613

› Section 3.1 – Revised and simplified escalation rules, with inline examples
  – An outgoing message to protect includes an option OPT
  – OPT is originally defined as Class U or I for OSCORE
  – Should OPT be treated as if being of Class E instead?

› Same rationale as usual: encrypt and integrity-protect whenever it is possible
  – Three cases are defined, as “Any CoAP option OPT such that all the following conditions hold”
  – If there is a match, the option is treated as if being of Class E, otherwise as per its original Class
  – Added new state diagram in Appendix B; adapted version also in the next slide

› Unexpected but good side effect
  – When no proxies are involved, then Uri-Host and Uri-Port are encrypted
  – Backward compatible with endpoints that do not implement this update
Encryption of Class U/I Options

Sender OSCORE endpoint

I have an outgoing message M, which includes an option OPT.
I have to protect M for another OSCORE endpoint X

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

Did I add OPT to M?

Yes

• X is my next hop;
• My next hop is not supposed to be the immediately next consumer of OPT

No

This state can be reached only at a proxy

Is X a consumer of OPT?

No

Is X the immediately next consumer of OPT?

Yes

Does X need to access OPT before decrypting M or in order to decrypt M?

Yes

Process OPT as per its original Class U or I

No

Process OPT as Class E

• X is my next hop;
OR
• My next hop is not supposed to be the immediately next consumer of OPT

No

Yes

Yes

No

No

No

Yes

No

Yes

No

Yes

No

Yes
Authorization checks before OSCORE decryption

- Already required before proceeding with a forwarding; Christian proposed this addition
- Check if the Security Context is available and in an allow-list associated with the alleged sender
- Preserve location anonymity of an origin server, as warranted by a reverse-proxy in front of it

Göran: “authorization” is a particular case of something more general

- Revised: “authorized operation” → “acceptable operation”
- Both for a proxy to forward and for any OSCORE endpoint to decrypt an incoming request
- The endpoint decides based on its local configuration and/or authorization enforcement

For reverse-proxies

- Considered also the Uri-Host and Uri-Port options as Proxy-related options that influence the process
Comply with a special case at a forward-proxy, as noted by Christian

- If the request can be forwarded and the target URI authority points to the proxy itself, ...
- then the proxy has to directly consume the request, see Section 5.7.2 of RFC 7252

An endpoint SHOULD define the maximum number of OSCORE layers that it is able to apply (remove) when processing an outgoing (incoming) CoAP message

- Consistent with the application security requirements, also suggested by Christian
- Bounded by the maximum active OSCORE Security Contexts at the endpoint
- Bounded by the number of intermediate OSCORE endpoints explicitly set up
- At a receiving endpoint, the OSCORE decryption fails if the limit is reached
- Practical upper bound on the loop-based decryption of incoming messages

Updated state diagram in Appendix C; adapted version in the backup slides

- We did manage to squeeze in the additions suggested by Christian 😊
Next steps

› Closer look at:
  – Processing of the Hop-Limit option (RFC 8768)
  – Addition of an outer option, after producing the corresponding, encrypted inner option (e.g., Observe)

› Handling multiple responses to the same request, if also protected by a proxy
  – Same rationale and approach as in draft-ietf-core-oscore-groupcomm

› Extend the security considerations

› More examples of message exchanges in Appendix A
  – E.g., with a reverse-proxy, with a chain of proxies

› "OSCORE-in-OSCORE" named as "Matryoscore"?

› Comments and reviews are welcome!
Thank you!

Comments/questions?

https://github.com/core-wg/oscore-capable-proxies
Backup
Motivation

› A CoAP proxy (P) can be used between client (C) and server (S)
  – A security association might be required between C and P

› 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
Use cases

› Section 2.1, CoAP group communication through a proxy [4]
  – The proxy identifies the client before forwarding

› Section 2.2, Observe multicast notifications with Group OSCORE [5]
  – The client securely provides the Ticket Request to the proxy

› Sections 2.3 and 2.4, OMA Lightweight Machine-to-Machine (LwM2M)
  – The LwM2M Client uses the LwM2M Server as a proxy towards External Application Servers
  – The LwM2M Server uses the LwM2M Gateway as a reverse-proxy towards External End Devices

› Further use cases are listed in Section 2.5
  – Transport indication through trusted proxies – draft-ietf-core-transport-indication
  – CoAP performance measurements involving on-path probes – draft-ietf-core-coap-pm
  – EST over OSCORE through a CoAP-to-HTTP proxy – draft-ietf-ace-coap-est-oscore
  – OSCORE-protected “onion forwarding”, a la TOR – draft-amsuess-t2trg-onion-coap
  – Proxies as entry point to a firewalled network

Use cases

1. CoAP Group Communication with Proxies
   – draft-ietf-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 end-to-end security …
   – … C provides P with a Ticket Request obtained from S
   – That provisioning should be protected over C ↔ P
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
Use case 3 – 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: Am I a forward-proxy?
     - Yes: Is there the Proxy-Uri or Proxy-Cri Option?
       - Yes: Is forwarding this request an acceptable operation?
         - Yes: Consume the proxy-related options
         - No: Return 4.01
       - No: Is there the Proxy-Scheme or Proxy-Scheme-Number Option, together with the Uri-Host/Uri-Port Options?
         - Yes: Is forwarding this request an acceptable operation?
           - Yes: Consume the proxy-related options
           - No: Return 4.01
         - No: There is no Proxy-Scheme or Proxy-Scheme-Number Option, but there are Uri-Path and/or Uri-Host and/or Uri-Port Options
           - Yes: Am I a reverse-proxy using the indicated virtual addressing information for proxying?
             - Yes: Consume; OR decrypt and repeat
             - No: Determine if proxying or not
     - No: Determine if proxying or not
   - No: Return 5.05

2. Is there an OSCORE Option?
   - Yes: Is there an application?
     - Yes: Decrypt
     - No: Success?
       - Yes: Return 4.00
       - No: Decrypt
   - No: Return 4.01

3. Is the authority (host and port) of the request URI identifying me?
   - Yes: Forward
   - No: Forward

4. Is forwarding this request an acceptable operation?
   - Yes: Consume the proxy-related options and forward
   - No: Return 4.01

5. Is there an URI-Path Options?
   - Yes: Deliver to the application
   - No: Return 4.01

6. Is decrypting this request an acceptable operation?
   - Yes: Decrypt
   - No: Return 4.01

7. OSCORE error handling