

OSCORE-capable Proxies

draft-ietf-core-oscore-capable-proxies-06

Marco Tiloca, RISE
Rikard Höglund, RISE

IETF 125 Meeting – Shenzhen – March 20th, 2026

Scope: update RFC8613 and RFC8768

- › **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”
- › **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
- › **Explicitly admit nested OSCORE protection**
 - For example, first protect end-to-end over $C \leftrightarrow S$, then further protect the result over $C \leftrightarrow P$
 - 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
- › **Explicitly define the Hop-Limit Option (RFC 8768) to be Class U for OSCORE**

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

“Recent” updates

- › **Last time presented: version -02 at IETF 120 (July 2024)**
- › **-02 → -03**
 - Completed content about the CoAP Hop-Limit Option
 - › Clarified motivation for being Class U for OSCORE, in Section 1 “Introduction”
 - › Added related security considerations, see Section 7.2 “Hop-Limit Option”
 - OSCORE-capable proxies have to understand CoAP options in outgoing messages that they protect
- › **-03 → -04**
 - Use cases moved to Appendix A
 - Message processing: deviations from RFC 8613 collected in Section 2.1
 - Clearer reasoning around inner/outer CoAP options in Section 2.2 (escalation of option protection)
 - More security considerations on membership of OSCORE groups
- › **-04 → -05**
 - Minor fixes and editorial improvements

Updates since version -05

› Editorial

- Clearer phrasing and readability improvements
- Updated references to Internet Drafts and to LWM2M specifications
- Removed use of normative language when the behavior is inherited and not new
- Removed unnecessary references to sections of RFC 7252

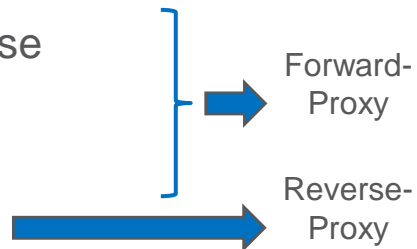
› Improved Section 6 “CoAP Header Compression with SCHC”

- Building on *draft-ietf-schc-8824-update*, intended to obsolete RFC 8824
- Fixed generalization of Outer SCHC Compression (i.e., after all OSCORE protections)
- Explicit distinction between Inner and Outer SCHC Compression Rules

Updates since version -05

› Covered the new Uri-Path-Abbrev Option, see *draft-ietf-core-uri-path-abbrev*

- Added to the set of “Proxy-related options”
- Possibly together with Proxy-Uri or Proxy-Cri --- Exotic narrow case
- Possibly together with Proxy-Scheme or Proxy-Scheme-Number, but without the mutually exclusive Uri-Path
- Possibly together with Uri-Host and/or Uri-Port, without Uri-Path



› Accordingly updated

- Section 1.1 “Terminology” – Definition of “Proxy-related options”
- Section 2.4 “Processing of an Incoming Request”
- State diagram in Appendix D on processing incoming requests

Updates since version -05

- › **Explicit requirement: Partial IV in non-first responses to the same request**
 - That was already the intention, but not all possible cases were already covered
 - Thanks to Christian Amsüss for pointing out!

- › **Possible cases**
 - The response is protected with Group OSCORE → All ok already
 - The response is protected with OSCORE and Observe is used → All ok already
 - The response is protected with OSCORE and Observe is not used
 - › This document effectively creates this case, which was not covered yet

- › **Message processing: new Sections 2.5.1 and 2.6.1 “Partial IV in the OSCORE Option”**
 - Response sender: it must include the Partial IV in non-first responses to a request
 - Response recipient: for a given request, it must only accept at most one response without Partial IV from each source OSCORE endpoint

Updates since version -05

- › **Review of version -04 from Christian Amsüss [1] – Thanks a lot!**
 - Comments seem to refer to v -01 of the WG document and the last version of the individual submission

- › **Clarified “consumer” in Section 2.2 “Protection of CoAP Options”**
 - *..., a recipient endpoint is denoted as "consumer" of an option OPT if the endpoint is meant to have access to OPT for processing it as appropriate.*

- › **Section 2.4 “Processing of an Incoming Request”**
 - Fixed use of error codes at the origin server
 - No OSCORE Option present and no application found
 - › Use 4.04 (Not Found) instead of 4.00 (Bad Request)
 - OSCORE Option present together with Uri-Path or Uri-Path-Abbrev
 - › Removed text about using 4.00 (Bad Request)
 - › Per RFC 8613, present Class E options are removed before starting any OSCORE processing

Updates since version -05

› **Appendix A “Use Cases”**

- Two use cases have been better explained and brought up as dedicated subsections
- A.5 “Access Control to a Proxy”
 - › Usable only by authenticated and authorized clients
- A.6 “Access Control to the Origin Server” (via a proxy)
 - › Based on client’s authenticated identity
 - › To unauthenticated clients, the proxy can be a hard firewall or a rate-limiter [2]

› **Policies for (source-based) decryption of incoming requests at origin servers**

- Reorganized text to be more modular, better positioned, and more explanatory
- Shortened 2.4.0: just mention about such policies, with a forward pointer to Section 2.4.1
- New Section 2.4.1, explaining the rationale and practical use of such policies
- Section 7.1: updated related security considerations

› **... And many good editorial suggestions**

Open point from Christian's review

- › **When should a proxy *try* to establish a Security Context with an origin server?**
 - Any guidance to give? Anything that could be learned from SVCB Resource Records (RR)?
- › **Discovering if the server supports EDHOC (RFC 9528)**
 - That should be possible by using the toolbox defined in *draft-ietf-lake-app-profiles*
- › **Discovering if the server supports nested OSCORE layers**
 - SVCB RR seem a good approach. Is this also applicable to reverse DNS lookup?
 - › The Proxy-Uri/Uri-Host Option can include a literal IP address instead of a registered name ...
 - Can we rely on target attributes in a link to a new well-known resource?
- › **Deciding if trying to establish a Security Context**
 - If the proxy supports OSCORE and knows that the server supports nested OSCORE layers ...
 - The proxy SHOULD establish OSCORE with the server whenever possible
 - If the proxy supports EDHOC and knows that the server supports EDHOC ...
 - The proxy SHOULD use EDHOC to establish a Security Context with the server

Implementation

› **Eclipse Californium (Java) – Lucas Åhl**

- <https://github.com/Toflowz/californium/tree/for-Edhoc>
- Tested with the Uri-Proxy Option or with the Proxy-Scheme Option plus the Uri-* Options
- Tested with OSCORE over Client-Proxy and/or Client-Server and/or Proxy-Server
- Tested also using Observe (RFC 7641) and key agreement with EDHOC (RFC 9528)

› **Contiki-NG – Manassanan Vongprai**

- <https://github.com/depuf/contiki-ng/tree/develop>
- Tested on Zolertia FireFly nodes
- Tested with the Uri-Proxy Option
- Tested with OSCORE over Client-Proxy and Client-Server

Next steps

- › **Address remaining points from Christian's review**
- › **Handling replay and local ordering of multiple responses to the same request**
 - Same rationale and approach as in Group OSCORE, see *draft-ietf-core-oscore-groupcomm*
- › **Examples of message exchange with a chain of proxies**
- › **Comments and reviews are welcome!**

Thank you!

Comments/questions?

<https://github.com/core-wg/oscore-capable-proxies>